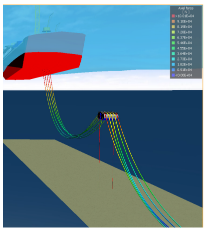
Flexcom

Flexcom User Manual

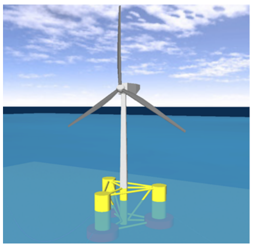
Table of Contents

Part I

1 Flexcom

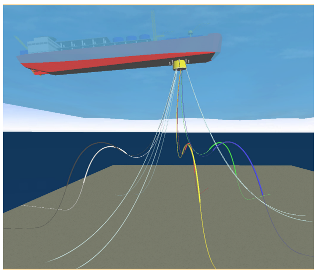


Flexcom is a premium structural analysis software package which has underpinned the engineering design on some of the world's most demanding offshore projects. Developed by Wood's software solution specialists, the program is designed and maintained by engineers who are immersed in the offshore energy industry. Flexcom's design philosophy is based on the provision of advanced computational techniques to provide confidence in the engineering design, coupled with a user-friendly interface which facilitates optimum productivity. The software uses an industry-proven finite element formulation, incorporating a hybrid beam-column element with fully coupled axial, bending and torque forces. Flexcom is a highly versatile software package, capable of simulating risers, mooring lines, umbilicals, CALM buoys, offloading lines, pipelines, installation processes, offshore wind turbines and wave energy devices. With over 30 years' experience, Flexcom has been delivering advanced engineering solutions to all the major operators, contractors, construction companies, manufacturers, and

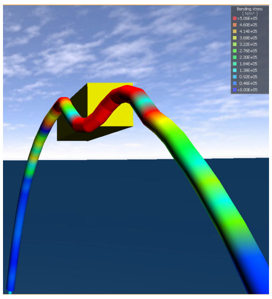


engineering consultants.

BENEFITS OF USING FLEXCOM



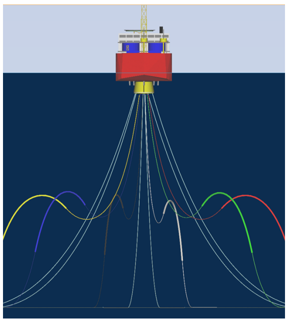
- **Supreme confidence in the engineering design**: Flexcom's renowned computational technique is widely acknowledged as best-in-class
- **One stop solution**: Flexcom's versatile nature makes it suitable for use in a variety of scenarios, ranging from FEED studies, detailed engineering design, fatigue life assessment, structural installation and decommissioning
- **Quality assurance**: Mathematical equations are fully supported within the software, allowing users to develop and validate template models, streamlining the QA process. Flexcom's keyword style input also readily lends itself to QA inspection
- **Advanced visualisation**: Powerful visual aids such as colour contouring of stresses provide the user with an intuitive visual representation of engineering data
- **Trust and reliability**: Flexcom is bolstered by a premium technical support service, with over 90% of queries fully resolved within one working day
- **Cost effective solution**: Affordable cost structure and very flexible licensing options. Plus our user interface does not require a licence, maximising the number of licences available for numerical simulation.



ADVANCED USER EXPERIENCE

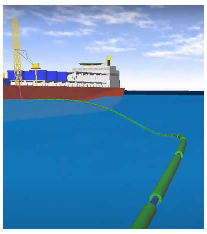
- Quick and easy model building via the Line component and automatic mesh-creation facility
- Fully integrated keyword editor facilitates rapid data specification, with helpful input prompts and command auto-completion
- Parameters and mathematical equations are supported, effectively providing spreadsheet type functionality within the Flexcom environment
- Streamlined post-processing enables assembly of results from a large load case matrix into a single spreadsheet
- Enhanced visualisation of engineering data such as stress contouring and 3D surface plotting
- Direct interface to Microsoft Excel, with optional access to VBA code which enables power users to develop specialised post-processing tools
- User defined plug-ins allow power users to create intelligent models which automatically adjust to instantaneous conditions which develop over time during dynamic simulations
- Dedicated video creation studio which allows users to create customised videos of finite element models and showcase innovative engineering designs. It's even possible to create a fully immersive virtual reality experience using the Unity Plug-In.

SIMULATION CAPABILITIES

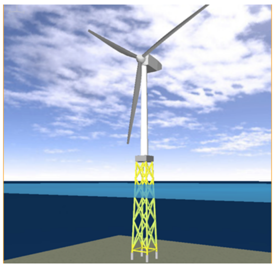


- Static, quasi-static, time and frequency domain analysis, plus Fatigue life assessment
- Modal analysis, interfacing with Shear7 (enabling the assessment of fatigue damage caused by vortex induced vibration)
- Recognised industry leader in finite element solution techniques. Up to 10 integration points per element to ensure precise distribution of applied forces. Third order shape functions used to predict solution variations within each element
- Several non-linear material models, including hysteretic bending effects and plastic hardening. Plus pipe-in-pipe, pipe-on-pipe and multiple riser bundle configurations
- Fully coupled aero-hydro-structural modelling of floating offshore wind turbines via coupling with OpenFAST
- Detailed internal fluid and slug flow models, including centrifugal, Coriolis and dynamic pressure effects
- Range of contact modelling options, including contact surfaces and line clashing
- Wake interference, including Huse and Blevins' formulations, plus a generic user-defined wake model
- User defined plug-ins which allow power users to adjust the constitutive finite element matrices and directly control the numerical solution at run-time

- Financial appraisals of Wave Energy Converters via coupling with ExceedenceFinance, a third party techno-financial software product.



FURTHER INFORMATION



Refer to Getting Started for further information on getting to grips with Flexcom, building your own models and running your own simulations. In order to help novice users to up-skill and improve their proficiency with Flexcom, we have compiled a series of online tutorial videos.

Check out our extensive video library for illustrations of Flexcom's modelling capabilities and broad range of applications.

Learn about our flexible licensing options and get your hands on a free trial version of the software.

We have over 2,600 followers on LinkedIn and we encourage everyone to sign up. We'll keep you update with news of latest releases and new modelling features as they become available. You'll also find some very interesting simulation videos which are shared on our forum.

Enjoy working with Flexcom!

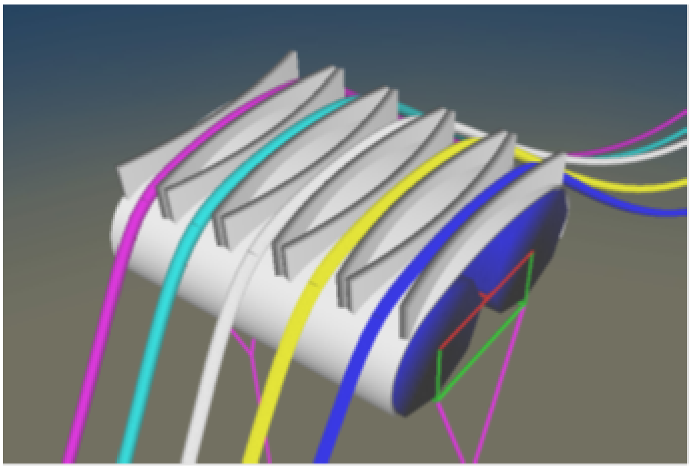
PROMOTIONAL VIDEO

1.1 Getting Started

MODEL BUILDING

Model building is very straightforward with Flexcom. Slender structures such as risers, pipelines and mooring lines may be built quickly and easily using the Lines modelling feature, while the automatic mesh-creation facility readily accommodates mesh sensitivity studies. Large floating structures (such as FPSOs, semi-subs and drill-ships) are modelled using the standard Vessel feature (displacement-driven simulation) while smaller floating structures are treated as Floating Bodies (force-driven simulation). Environmental Parameters, such as ocean depth and water density, along with a Seabed definition, typically completes the minimum required model specification.

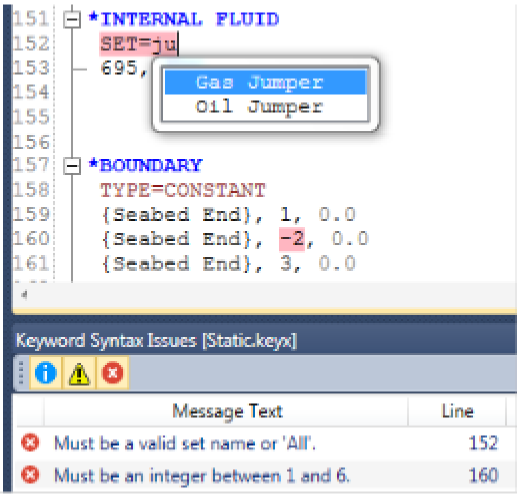
The Model View continually updates to reflect changes to the model, and provides a preview of both the overall configuration and the distribution of finite elements.



Sample Mid-Water Arch Model

DATA ENTRY

Flexcom is designed to simplify the quality assurance process. The keyword style input readily lends itself to QA inspection by project managers. The integrated Keyword Editor facilitates rapid data specification, aiding the user with helpful input prompts and command auto-completion. Any inaccurate specifications are detected by automatic syntax checking and data validation. Beginner or inexperienced users may prefer to use the Table Editor. Data is entered in tabular format, with the various rows and columns clearly labelled.

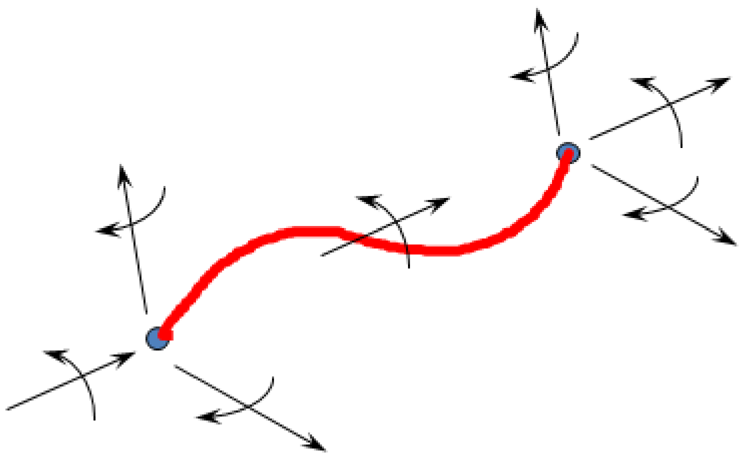


Helpful Input Prompts and Automatic Data Validation

Pre-defined input Parameters form an integral part of a typical model specification. Once a parameter is altered, all dependent parameters are automatically updated, with consequent savings for model setup effort. Environmental simulation is also seamless, as one parameterised master template keyword file may be used to generate all the required input files (e.g. to accommodate a matrix of varying wave periods and headings).

FINITE ELEMENT SOLUTION

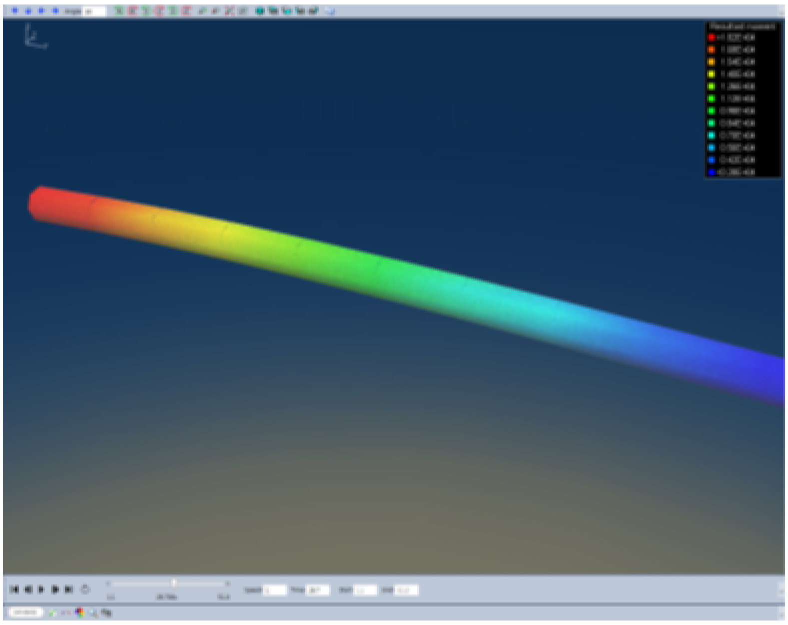
Flexcom's advanced computational technique provides supreme confidence in the engineering design. The software uses an industry-proven Finite Element Formulation, incorporating a hybrid beam-column element with fully coupled axial, bending and torque forces. Up to 10 integration points are used within each finite element to ensure a precise distribution of applied forces. Third order shape functions are used to predict solution variable (e.g. moment, curvature) variations within each element.



Advanced Computational Technique

POST-PROCESSING

Element forces may be presented in traditional plot format, graphically illustrated via colour contouring, or automatically collated in Microsoft Excel. These versatile post-processing options deliver tangible benefits for engineering teams, as pertinent results may be extracted quickly and easily.



Stress Contouring

ONLINE TUTORIAL VIDEOS

In order to help novice users to up-skill and improve their proficiency with Flexcom, we have compiled a series of online tutorial videos.

PROGRAM DOCUMENTATION

Flexcom is equipped with a fully integrated documentation system. One centralised help system encompasses all of the various sub-components, such as data inputs, background theory, and sample models. It also incorporates extensive cross-referencing for effortless browsing. The documentation is logically broken up into the following categories:

- Installation contains information on basic System Requirements and Network Licensing.
- User Interface introduces the various concepts behind Flexcom's user interface, and software user-experience in general. It presents detailed information on the various sub-components, including the concept of a Project Workspace, Model Building (via the Keyword Editor or Table Editor), Running Analyses (explaining Analysis Jobs and Work Sequence), and Results Examination (via the powerful Model View and Plotting features).

  - Flexcom's main user interface also incorporates two specific modules, Flexcom Wind and Flexcom Wave, which are specifically designed to create models for floating wind turbines and wave energy conversion devices, respectively.

- Data Inputs provides a complete reference guide to all data inputs used in Flexcom.

- Theory provides a comprehensive background on the modelling and analytical capabilities of the software. It includes a section on Flexcom's industry-proven Finite Element Formulation, an advanced computational technique which provides supreme confidence in the engineering design. It also discusses the comprehensive range of Postprocessing options available in Flexcom.

- Examples describes a set of examples which illustrate some of the range of applications for which Flexcom may be used. Sample models cover areas such as Top Tensioned Risers, Steel Catenary Risers, Flexible Risers, Mooring Systems, Offloading Systems, Pipelines, Hybrid Riser Systems and Installation Analysis.

- References lists all the technical publications referenced throughout the help system.

KNOWLEDGE BASE

In addition to the comprehensive documentation we also host an online Knowledge Base for Flexcom as well as all of Wood desktop software products.

This web based resource contains answers to frequently asked questions on the use of our products. It essentially provides best practice advice on how to use the products and troubleshoot potential issues encountered.

Content is continually added to the Knowledge Base over time and so it represents a valuable reference for engineers who are actively using our software.

To access the Knowledge Base users are asked to register, via the link, with their company email address and individual password. We will approve new users as soon as possible after registration.

Once access has been granted, users are then free to navigate through all the provided content, search for answers on a particular topic, and vote on the most helpful answers.

1.2 Tutorial Videos

In order to help novice users to up-skill and improve their proficiency with Flexcom, we have compiled a series of online tutorial videos. If you are beginning to work with Flexcom for the very first time, it is recommended that you start at the beginning, and work your way along through each video in the beginner section (and beyond if you have time). We hope to add more tutorial videos over time, expanding the range of topics covered, and creating a library which will be helpful reference for experienced users also. Note that the closing section on migrating from earlier versions is intended for existing users of the software who have recently upgraded, or are about to upgrade, to a new version of the software. So if you are a new user, simply disregard this final section.

BEGINNER LEVEL

1. General Software Overview
2. User Interface I
3. Model Building
4. Unit Systems
5. Parameters and Equations
6. User Interface II
7. Dynamic Analysis
8. Post-Processing
9. Model View
10. Plotting Module
11. User Interface III

INTERMEDIATE LEVEL

12. 3D Seabed Profile Generator
13. Pre-Processor Scripting I

14. Pre-Processor Scripting II

15. Summary Post-Processing & Collation

16. Code Checking

17. Excel Add-in

18. AVI Studio

19. Flexcom Wind (FWTs)

20. Flexcom Wave (WECs)

ADVANCED LEVEL

21. User-Defined Element

22. User-Solver Variables

23. Custom Post-Processing (VBA)

24. Custom Post-Processing (DARs)

25. Unity Plug-in

MIGRATING FROM OLDER VERSIONS

26. Running Flexcom on the cloud

27. Flexcom 8.10 Highlights

28. Flexcom 8.6 Highlights

29. Flexcom 8.4 Highlights

30. Units Importer

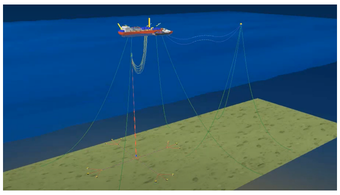
31. Project Importer

32. Auxiliary Body Converter

1.3 Video Library

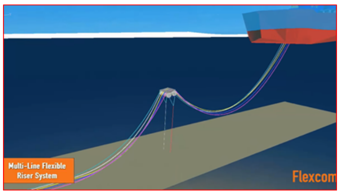
ENGINEERING DESIGN

These videos provide sample illustrations of Flexcom's modelling capabilities and broad range of applications.

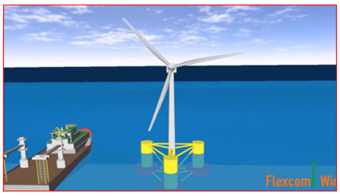


Offshore production system. This system is based on a free standing bundled hybrid tower concept, which is typical of deep water production systems deployed in West of Africa field developments. Oil is transferred to the FPSO using a set of flexible jumpers, and from there to the CALM buoy via two offloading lines which hang in a wave type configuration. Both the FPSO and the CALM buoy are held on station using chain and steel wire mooring lines. Several flowlines are located on the seafloor and connected to the vertical tower using an arrangement of rigid spools.

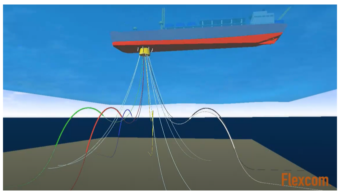
Although this model is very complex and contains over 3000 elements, it is very easy to assemble in Flexcom using the line modelling feature. Flexcom also provides a high accurate pipe-in-pipe modelling feature which has underpinned the engineering design of some of the world's most complex riser bundle configurations. Flexcom has a detailed internal flow model, catering for centrifugal, coriolis and dynamic pressure effects. The internal fluid contents may also vary over time, in order to accurately simulate intermittent slug flow.



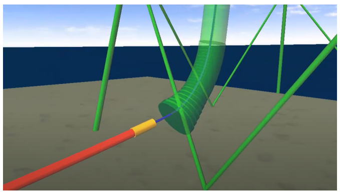
Flexible riser system. The model contains 5 flexible risers, which pass over a tethered mid-water arch in a lazy-S configuration, and is typical of the shallow water production systems deployed in North Sea field developments. The mid-water arch itself is modelled as a rigid frame structure, which supports an arrangement of contact surfaces. Uplift is provided by two large buoyancy tanks which keep the tethers in tension, and help to stabilise the entire system. Flexcom's renowned computational technique provides high levels of confidence in the engineering design. Although the time domain simulation is relatively detailed, including intermittent contact with curved surfaces, the entire simulation may be completed in just a few minutes. See Example C02 - Multi-Line Flexible System for further information.



Floating Offshore Wind. This is one of the reference models used in the international research project OC4. OC4 is sponsored by the International Energy Agency and coordinated by NREL. It benchmarks a variety of simulation software for modelling of offshore wind turbines, both fixed and floating. Results from the Flexcom model show very good agreement with published data, in terms of platform motions, mooring line tensions, tower base moments, nacelle accelerations etc. Aerodynamic outputs such as generator power and rotor speed also show close agreement with other modelling tools. See Example L01 - OC4 Semi-Submersible for further information.

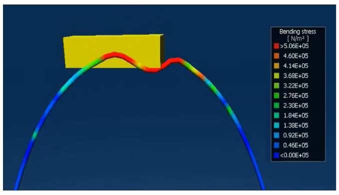


Turret disconnect. This is an operation which is performed in harsh environments caused by inclement weather conditions. The model includes a turret moored FPSO, 6 mooring lines, 2 production risers, 2 water injection risers, a gas lift line and an umbilical. The turret itself is modelled as a frame type structure, with rigid connections linking it to each of the attached lines. The turret is disconnected from the FPSO and allowed to sink into the ocean. Contact with the circular vessel moonpool is also monitored using an arrangement of contact surfaces positioned around the turret. See Example C03 - Turret Disconnect for further information.

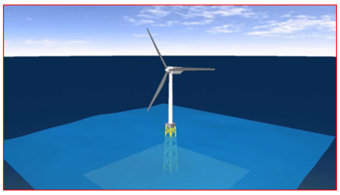


J-Tube pull-in. The J-Tube is a curved conductor which is rigidly attached to a fixed platform, and through which a flexible riser is installed. One end of the flexible is attached to a winching cable which is inside the J-tube. The winching process is modelled using Flexcom's winch element feature, which allows paying-out or reeling-in operations to be simulated. The interaction between the flexible and J-tube is modelled using Flexcom's pipe-in-pipe modelling facility. Sliding connections are used in this model, with the software continually monitoring and updating the connected nodes as the analysis progresses. See Example H02 - J-Tube Pull-In for further information.

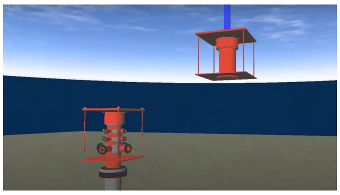
Independent studies have shown a strong correlation between Flexcom and general purpose finite element software for this type of analysis. Furthermore, the authors conclude that the solution provided by Flexcom offers greater efficiency, both in terms of ease of model set-up, and significantly reduced computational times.



Dropped object. This model contains a flexible riser arranged in a steep wave configuration, which is connected to an FPSO. A support vessel is docked alongside the FPSO, and a freight container accidentally drops from the crane and sinks down into the ocean. Contact between the falling container and the flexible riser is modelled using Flexcom's line clashing feature. Clashing is a complex and highly non-linear phenomenon, so the use of relatively small time steps is essential, in order to accurately capture the moment of impact, and the subsequent structural response. Flexcom automatically reduces the solution time step in anticipation of contact, and once the components have separated after impact, the time step begins to increase again. This approach facilitates a robust and accurate contact model, while also ensuring an efficient simulation. In order to accurately capture the deformation of the flexible riser, relatively short elements are used in the contact region. Numerical damping is typically used to simulate energy dissipation during the collision. See Example J01 - Dropped Object and Recovery for further information.

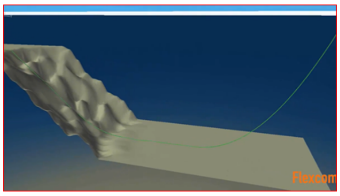


Fixed offshore wind. This is one of the reference models used in the international research project OC4. OC4 is sponsored by the International Energy Agency and coordinated by NREL. It benchmarks a variety of simulation software for modelling of offshore wind turbines, both fixed and floating. Results from the Flexcom model show very good agreement with published data, in terms of shear forces and bending moments, and structural deflections in the jacket structure. See Example L03 - OC4 Jacket for further information.



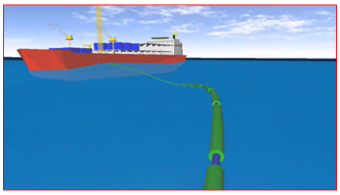
Drilling Riser. This is a deep-water riser, typical of deepwater exploration systems deployed in regions such as the Gulf of Mexico. The model simulates an emergency disconnect scenario, following failure of the dynamic positioning system on the drillship. In such situations, the riser is quickly disconnected from the wellhead to prevent any potential damage. See Example A01 - Deepwater Drilling Riser for further information.

Although Flexcom can provide an approximate simulation of the disconnect event, a more comprehensive simulation may be performed using DeepRiser DeepRiser, which specialises in the analysis of top-tensioned risers. DeepRiser provides a detailed hydro-pneumatic tensioner, which models all the major hydraulic and pneumatic components. It can model each tensioning cylinder independently, and can simulate the behaviour of the anti-recoil control system. It also incorporates an advanced fluid flow model, based on a finite volume numerical technique, to model the drilling mud exiting from the base of the riser.

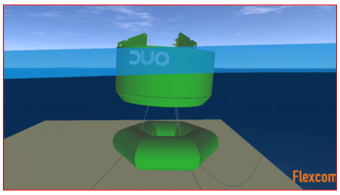


Pipe Laying. This is a complex installation procedure, as the pipeline is laid down over a steep subsea precipice. The winching process is modelled using Flexcom's winch element feature, which allows paying-out or reeling-in operations to be simulated. Flexcom possesses a powerful seabed modelling facility which can handle complex topographies with ease. The user simply specifies an arbitrary cloud of data points, and Flexcom's triangulation algorithm automatically generates a continuous seabed profile.

Although this simulation was performed using Flexcom, Wood also offer a dedicated installation software package called PipeLay. PipeLay provides a comprehensive set of technical features which are specifically designed for installation analysis, including rapid generation of fixed and floating stinger profiles, and a range of options for modelling rollerbox support contact. PipeLay also facilitates automatic adjustment of support elevations to achieve specified curvature limits, saving time for installation engineers.



Floating hose. This hose is used to offload oil from a semi-submersible platform to an FPSO. Floating hose models are extremely sensitive to wave loading, as the entire model is located in the water surface region. Flexcom provides a detailed buoyancy formulation, which accounts for the partial submergence of the hose. The submerged volume of each element is computed based on the instantaneous water surface elevation and the local penetration angle. Flexcom also automatically increases the number of integration points within every surface piercing element. This ensures a precise distribution of the applied hydrodynamic forces. See Example E03 - Floating Hose for further information.



Wave energy converter. This device is known as 'DUO', developed by Pure Marine Gen. Its innovative nature has led to it being shortlisted as a finalist in the U.S. Department of Energy's Wave Energy Prize. Empirical results from model-scale tank tests have shown a strong correlation with numerical simulations performed using Flexcom. The software uses coupled analysis to model WECs, including high and low frequency wave forces, added mass, radiation damping, and viscous drag.

Flexcom can also simulate the power take-off using a combination of non-linear spring and damper elements, presenting the designer with key information regarding power output and energy generation. According to Dr Paul Brewster, C.T.O. at Pure Marine Gen, "*the advances in simulation capabilities that Flexcom provides have enabled the development and validation of DUO, and the step-change improvements in performance make wave energy a commercial reality*".

More interesting videos are also available on the Flexcom channel on YouTube.

1.4 Licensing Options

OVERVIEW

Flexcom has very flexible licensing options, to suit various organisational requirements. There are actually 5 different methods by which Flexcom may be licensed from Wood :

- Ownership
- Monthly Rental
- Annual Subscription

- Token-based
- Cloud-based

Flexcom's user interface does not require any license and so tasks like model building and post-processing can be done without occupying a license seat. The license is only required when performing the numerical simulation. This open system is significantly different to most licensing models on the market and affords great flexibility to the user base. It also accommodates new or junior team members who wish to gain Flexcom experience while not utilising a valuable license seat that could otherwise be used for analysis.

LICENSE MODELS

Ownership

A license is purchased outright by the customer so that the software can be used for an unlimited period. In addition to the initial purchase cost, the customer also pays an annual fee (beginning in Year 2) in order to receive technical support and software upgrades over time. This model is typically favoured by organisations where Flexcom is used extensively and consistently over long periods of time.

Monthly Rental

The customer rents the software on a month-by-month basis. Once the rental period expires, the software ceases to operate. Rental models can be suitable for well-defined projects of fixed scope. There is a breakeven point in terms of project duration beyond which it becomes more economical to opt for full ownership. When upgrading a rental license to full ownership, a significant discount is available based on a percentage of the total rental fees paid during the 12 months preceding the purchase date.

Annual Subscription

This model is conceptually similar to a monthly rental, with two notable exceptions. Firstly, the minimum subscription period is 12 months (as opposed to 1 month) and secondly, the cost of acquiring a license for 12 months via annual subscription is much lower than the cumulative total of 12 individual monthly rentals. This model offers an ideal combination of license flexibility and cost efficiency.

Token-Based

The token-based license represents a ‘pay-per-use’ model. It facilitates organisations who do not have enough workload to warrant a monthly rental license and who only need to use Flexcom on an ad-hoc basis. With this model, users purchase a number of tokens upfront and these tokens are consumed subsequently as simulations are performed. The rate of token consumption is dependent on the type of simulation being performed (e.g. dynamic time domain simulations cost more tokens than static analyses). Token-based licensing is not recommended for significant or prolonged usage of Flexcom, in which case the monthly rental, annual subscription or perpetual options (in ascending order of commitment) become more cost effective. Refer to token-based licenses for further information.

Cloud-Based

A cloud-based license involves the delivery of a software license, plus associated computational power and storage facilities, over the internet to offer a convenient, immediate and scalable means of running Flexcom. This ‘software-as-a-service’ (SaaS) model allows customers to pay for Flexcom exactly when and where required, helping to lower operational costs and spread out capital investments over time. It can appeal to a broad range of customer types:

- Companies who use Flexcom intensively can use cloud computing to supplement their desktop licenses and quickly scale up their computational power to meet busy project workloads and/or short deadlines.
- Companies who tend to rent Flexcom on a project-by-project basis may find the cloud solution a more practical and cost-effective model than the traditional rental models. If software hours can be related to engineering hours based on prior experience, detailed cost estimates can be provided to engineering clients up-front.
- Companies who are gradually phasing out traditional PC/server hardware and migrating all their systems to the cloud.

Refer to Flexcom-on-the-Cloud for further information on the cloud-based license.

SUMMARY

The following table presents a useful summary of the various license models and should assist you in selecting the one which best meets your requirements.

License Model	Pros	Cons	Sample Customer Profile
Ownership	• Software operates for an infinite period • Most cost effective license model in long term	• Significant up-front purchase cost • Additional annual fee in order to receive technical support and program upgrades	• Organisations who use Flexcom consistently over several years
Monthly Rental	• Low up-front cost • Payments spread evenly over several months • Option to upgrade to full ownership, including some level of discount	• Software ceases to operate after rental period has expired • Expensive means of licensing if used long term	• Ideal for short duration engineering projects of fixed scope, especially if software costs can be included in tenders/bids
Annual Subscription	• Much lower entry cost than full ownership • Much more cost-effective than monthly rental	• Software ceases to operate after annual subscription period has expired	• Organisations who use Flexcom consistently over several years

Licens e Model	Pros	Cons	Sample Customer Profile
		• Ownership model may work out slightly cheaper for very long-term usage	
Token-based	• Low up-front cost • Usage based rather than time based • Charges relate directly to usage levels • License period is very flexible as tokens do not expire	• Expensive means of licensing if software is used regularly and/or intensively	• Companies who use Flexcom very occassionally
Cloud-based	• Low up-front cost • Usage based rather than time based • Charges relate directly to usage levels	• Could be considered cost inefficient if suitable hardware is already available on site and is under-utilised	• Companies who use Flexcom intensively and who need to supplement their desktop licenses during busy periods

License Model	Pros	Cons	Sample Customer Profile
	• Harness the power of cloud computing • Reduce dependency on local hardware	• A detailed cost-benefit analysis is recommended to assess whether cloud computing or local hardware offers the best value in the long term. Each organisation is unique in this respect and the ultimate decision will depend on a variety of factors	• Companies who are gradually phasing out traditional PC/server hardware in favour of cloud solutions

1.5 New Features!

WELCOME!

Welcome to Flexcom 2022.1, on general release from August 2022. This release incorporates several significant new features, including a new truss element that offers enhanced speed and robustness when modelling structures with low/zero bending stiffness such as mooring lines, and several enhancements for modelling floating offshore wind turbines. Consistent with modern software trends, we have adopted a new version numbering system starting from this version. The major number now reflects the year of release, with minor and maintenance numbers having the same significance as previously.

We receive many helpful suggestions from our users and our focus is on delivering new features in the areas where you need them most. We welcome your continued feedback on Flexcom, it is an essential part of our software development process. So please feel free to contact me directly.

Aengus Connolly, Flexcom Product Manager

RECENT SOFTWARE UPGRADES

- Flexcom 2022.1August 2022
- Flexcom 8.13 October 2021
- Flexcom 8.12 November 2018 This release was issued to selected users only to facilitate extended testing.
- Flexcom 8.10 July 2018 Watch Flexcom 8.10 eSeminar
- Flexcom 8.9 December 2017 This release was issued to selected users only to facilitate extended testing.
- Flexcom 8.7 June 2017 This release was issued to selected users only to facilitate extended testing.
- Flexcom 8.6 August 2016 Watch Flexcom 8.6 eSeminar
- Flexcom 8.4 July 2015 Watch Flexcom 8.4 eSeminar

RETURNING USERS

If you have used Flexcom in the past, but not recently, you may be interested in a quick summary of the main structural changes which have been made to the software in recent years. Refer to Returning Users for further details.

KNOWN SOFTWARE FAULTS

Our policy is to provide complete transparency to our Flexcom user community regarding any known software errors or limitations. Refer to Known Software Faults for further information. All of these issues will be rectified in the next maintenance release of the software. In the interim however, workarounds are suggested where feasible.

1.5.1 Working from Home & COVID-19

Following the global COVID-19 pandemic, we realise that many engineers are now working from home part-time. Our objective as software providers in Wood is to enable our customers to work effectively and successfully from home, continuing to deliver value to their end clients. Many engineers are already equipped with company laptops, high-speed internet access and VPN connections to corporate networks. Due to the rapid onset of the virus however, not all engineers working from home may be equipped with the same high quality hardware/software as they are accustomed to from their workplace.

If your engineers are currently experiencing any difficulty is using Flexcom from home, Wood are happy to offer assistance. For example, we can offer web-hosted licenses which would enable all engineers with an internet connection to run Flexcom from their home computers, even if they cannot connect to the network license server in the office. These additional licenses would be supplied free-of-charge in most instances, helping to alleviate some of the pressure being experienced by your IT departments.

If your remote engineers require high powered computing solutions, we also offer our cloud computing service. This software-as-a-service model means that your engineers would have access to high-powered machines on the cloud, with only the most basic requirement of an internet browser on their local machine. While not completely free-of-charge, we can provide this service effectively at very competitive rates.

Please contact us if you require any assistance.

1.5.2 Flexcom 2022.1

Consistent with modern software trends, we have adopted a new version numbering system starting from this version. The major number now reflects the year of release, with minor and maintenance numbers having the same significance as previously.

Highlights include…

- **Truss Element:** The new Truss Element is designed specifically for modelling structures which have very low levels of structural bending stiffness (such as mooring chains) and is essentially a simplified version of the standard beam-column element employed by Flexcom. It has 3 translational degrees of freedom at each node, and deforms only in the axial direction (it does not deform in bending or torsion). As it does not solve for nodal rotations, the connection at each node is essentially a pure hinge. The axial force penalty term is retained making the truss element a 7-DOF hybrid finite element with two end nodes.

- **Wind Field Generator:** The Wind Field Generator app allows you to create wind data files which characterise the wind field as a function of space and time. It acts a user-friendly interface to the TurbSim software which does not have a Window-based GUI of its own. It allows you to run batches of TurbSim wind field simulations to generate all the wind data files required to support your design load cases.

- **Flexcom Wind GUI:** The architecture of Flexcom Wind, our dedicated wind turbine model building tool, has been updated. It now operates in a similar manner to our DeepRiser software, whereby a library of components can be built up over time, and you can pick and choose the components which you wish to include in any individual model. We hope to further expand the model building functionality, which is currently limited to semi-submersible platforms, in the next program release.

- **Soil Modelling:** Flexcom can now model T-z curves, which are analogous to the P-y modelling feature which has been available for some time. T-z curves represent the soil-structure resistance in the axial direction and are a useful addition for the analysis of top tensioned risers, which typically run through and below the mudline.

- **Rotational Damper:** A rotational damper element is now available, in addition to the traditional (translational) damper. It exerts a moment which is inversely proportional to the relative rotational velocity of its end nodes.

- **Display of Element Convected Axes in Model View:** You can now view the local convected axes for each element in the Model View. By way of background, Flexcom uses a convected coordinate axes technique for modelling finite rotations in three dimensions. Each element of the finite element discretisation has a convected axis system associated with it, which moves with the element as it displaces in space.

See the Flexcom 2022.1 Newsletter for full details on all these new features.

1.5.3 Flexcom 8.13

The content of Flexcom 8.13 is very much guided by user feedback. We received many helpful suggestions from our users and our focus has been on delivering new features in the areas where you need them most.

Highlights include…

- **Offshore Wind**. Aerodynamic modelling features are now available in Flexcom following successful completion of the software validation process. See Wind Turbine Modelling and Validation for further information. We've also updated our aerodynamic solver (developed by NREL) from FAST 8.16 to OpenFAST V2.6.0 (May 2021).
- **Flexcom-on-the-Cloud**. Flexcom is now available on the cloud via our AppStream platform. See Flexcom-on-the-Cloud for further information.
- **Numerical Solver**.
  - Display of iteration progression in the Model View for non-converged solutions.
  - Default numerical integration scheme is Generalised-α, controlled via optimised coefficient for the spectral radius at infinity (ρ∞). See *TIME STEPPING for a description of the new inputs.
  - Critical Euler load monitored in elements experiencing compression. See Compression and Buckling for further information.
- **Post-processing**
  - Reduced database sizes, by storing data at 1, 2 or 4 points per element. This option has the potential to greatly reduce the disk space, particularly for large load case matrices. *DATABASE CONTENT is used to control the number of data storage points.
  - ISO-13628-7 (completion/workover riser systems) now supported. See Code Checking for further information.
  - Seabed reaction forces available. Both time history (*TIMETRACE) and envelope (*STATISTICS) plots are available.
  - Nodal reaction forces available in local axis system (see *TIMETRACE). This can be very useful when checking interface loads in the vessel axis system.

    - Flexcom 8.13 informs you about the completion status of all simulations included in post-processing and results collation.

    - General improvements to Excel output. Node and element label data is now included, for ease of reference to any lines used during model building. The overall layout of data has also been improved for better user experience.

- **New Examples**. Flexcom 8.13 includes some interesting new examples, including:

    - Floating Wind: OC4 Semi-sub

    - Floating Wind: UMaine VolturnUS-S Semi-sub

    - Fixed Wind: OC4 Jacket

    - Wave measurement buoy

- **Pipe-in-Pipe: VIV Fatigue**. Flexcom offers a novel approach to estimate the VIV response of an inner pipe of a pipe-in-pipe system. See VIV Induced Fatigue of Pipe-in-Pipe Systems for further information.

- **User Experience**. Apart from the above, Flexcom 8.13 also provides a series of minor enhancements which will improve user experience. See Flexcom 8.13 Newsletter for further details.

1.5.4 Flexcom 8.12

Building on Flexcom 8.7 (which was subsequently superseded by Flexcom 8.10), Flexcom 8.12 delivers additional functionality in the area of floating offshore wind. Specifically it addresses two key limitations of Flexcom 8.7, namely the ability to model dynamic control systems and turbulent wind inflow.

- **Control Systems**: Flexcom now interfaces to the FAST module ServoDyn, providing a link between a user-generated control dynamic link library (DLL) and solution variables provided by AeroDyn and Flexcom. Flexcom implements variable rotor speed control by allowing the user to modify generator torque. When operating above the rated wind speed, blade pitch control is typically employed to shed additional power. Refer to Coupling between Flexcom and ServoDyn for further information.

- **Wind Field**: Flexcom now interfaces to the FAST module InflowWind, so dynamic and turbulent wind loading can be specified. InflowWind processes wind inflow data and supports several wind file formats including uniform, binary TurbSim full-field, binary Bladed-style full-field and HAWC formatted binary full-field wind files. It also has its own internal calculated steady wind and supports arbitrary wind directions. Refer to InflowWind Overview for further information.

1.5.5 Flexcom 8.10

This is a major release of the software which provides a number of significant technological advancements.

Guided by feedback from our global user base, we are proud to deliver a number of practical advances in key areas. Flexcom now provides a metal plasticity modelling feature for the first time. In response to the growing interest in marine renewable energy, we have also extended Flexcom's range of modelling capabilities to include floating offshore wind turbines and wave energy converters.

Highlights include…

- **Metal Plasticity:** Flexcom 8.10 sees the introduction of a metal plasticity model, capable of predicting plastic deformation and capturing residual strain, for the first time in Flexcom. Refer to Linear Elastic with Plastic Hardening material model for further information.

- **User-Defined Element:** This new feature allows you to directly change elemental properties while a simulation is in progress. See User Defined Element for further information.

- **Unity Plug-In:** The Unity Plug-In allows users to create more advanced visualisations than is possible within Flexcom itself. Refer to Unity Plug-In for further information.

- **Hydrodynamic Data Importer:** Flexcom 8.10 now provides a utility program allows you to automatically import characteristic data relating to a floating body from a range of well-known hydrodynamic simulation packages, including WAMIT, ANSYS Aqwa and NEMOH. Refer to Hydrodynamic Data Importer for further information.

- **Summary Wave Scatter:** This feature estimates simulation results for seastate combinations which you have not actually simulated using an extrapolation technique on simulated adjacent seastates. Refer to Summary Wave Scatter for further information.

- **Web Hosted & Token Based Licensing:** Flexcom now provides an option to use a web hosted licensing system.

- **New Examples:** Flexcom 8.10 includes some interesting new examples, including a steel pipe installation with plastic deformation, and a marine riser with an internal landing string deployment.

- **User Experience/Miscellaneous:** Apart from the above, Flexcom 8.10 also provides a range of more minor but highly practical additions which contribute to improved user experience. See here for further details.

1.5.6 Flexcom 8.9

In response to the growing interest in wave energy, Flexcom 8.9 sees the launch of a dedicated module, Flexcom Wave. Full details on all of the new features is contained in the Flexcom 8.9.1 Newsletter.

Floating Dual-Body Point Absorber

Highlights include…

- **New Module**: A dedicated *Flexcom Wave* module is provided with Flexcom 8.9. Software inputs are defined in engineering terms and logically grouped into familiar components, such as floating body, mooring and power take-off etc. Refer to Flexcom Wave for further information.

- **Sample Devices**: Flexcom Wave is accompanied by two sample devices: a Floating Dual-Body Point Absorber and a Submerged Tether-Moored Point Absorber. If you have a specific device concept which is not currently accommodated by Flexcom Wave, you can always Migrate to Flexcom itself.

- **Design Environment**: Flexcom Wave is complemented by ExceedenceFinance, a financial appraisal product which examines commercial feasibility, to form a fully-integrated design environment.

  - ExceedenceFinance contains databases of metocean data for various geographical locations around the world, and you can quickly download wave resource information for your chosen location, and automatically Import from Exceedence into Flexcom Wave.

  - You can also Export to Exceedence, sending the power matrix computed by Flexcom's detailed engineering model back into ExceedenceFinance, and perform financial appraisals based on realistic power data for a given wave energy converter and ambient environment.

- **Simulation Results**: Flexcom Wave provides you with various options for examining key parameters typically associated with wave energy converters, such as annual energy production, electrical power matrix, floating body motions and mooring fairlead tensions.

- **Hydrodynamic Data Importer**: This is a very helpful utility program which allows you to automatically import characteristic data relating to a floating body from a range of well-known hydrodynamic simulation packages, including WAMIT, ANSYS Aqwa and NEMOH. Refer to Hydrodynamic Data Importer for further information.

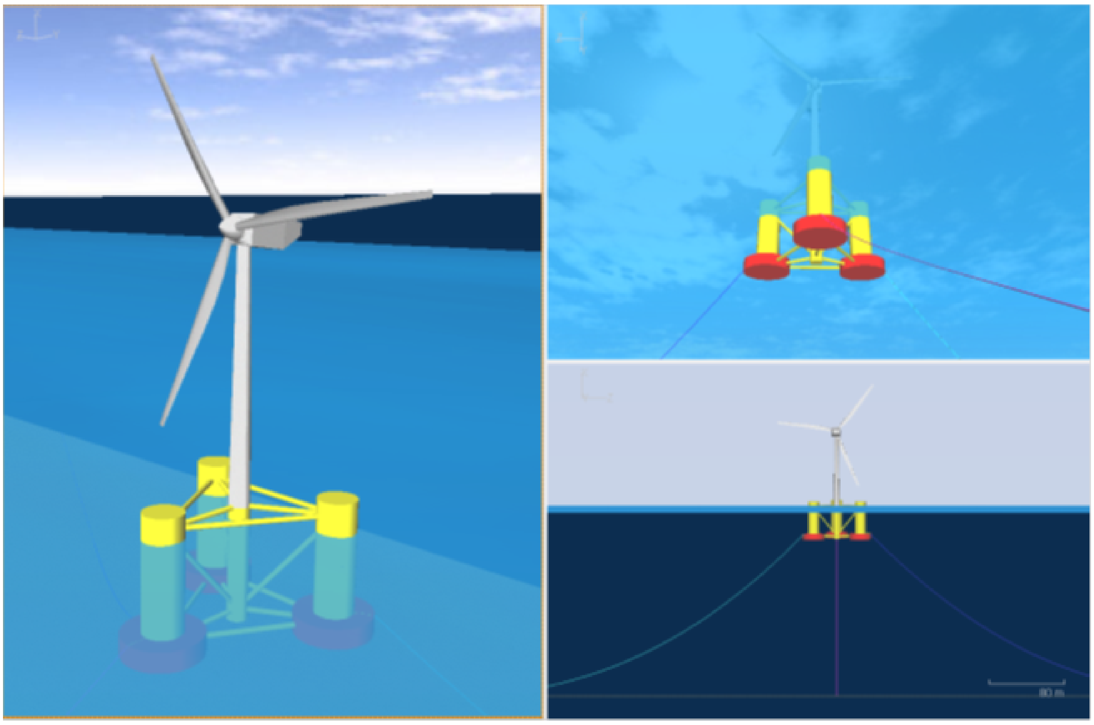
1.5.7 Flexcom 8.7

In response to the growing interest in floating offshore wind, Flexcom 8.7 sees the introduction of a dedicated module, Flexcom Wind. Full details on all of the new features is contained in the Flexcom 8.7.1 Newsletter.

Highlights include…

- **New Module**: A dedicated *Flexcom Wind* module is provided with Flexcom 8.7. Software inputs are defined in engineering terms and logically grouped into familiar components, such as turbine, tower, platform and so on. Refer to Flexcom Wind for further information.

- **Aerodynamic Model**: A trusted aerodynamic model delivered via a fully integrated coupling with the industry-recognised FAST software. Refer to Wind Turbine Modelling for further information.



Floating wind turbine modelled using Flexcom

1.5.8 Flexcom 8.6

The overarching theme of this new version is enhanced user experience. Having listened to feedback from our global user base, we have focused on providing a range of improvements in various areas of the program. Full details on all of the new features is contained in the Flexcom 8.6.1 Newsletter. Rather than following the written documentation, you may prefer to watch the Tutorial Video on YouTube.

Highlights include…

- **User Experience**: Flexcom 8.6 provides the best user experience of the entire Flexcom 8 release series, thanks to a highly responsive keyword editor. Refer to Keyword Editor for further information.

- **3D Plotting**: 3D Plotting is an advanced feature which provides enhanced visualisation of engineering data. For example, you can plot maximum effective tension as a function of both wave period and incident wave heading in a 3-dimensional space. Refer to Summary Collation Plot for further information.

- **Model Building**: A new feature has been added which allows you to model repeated line sections quickly and easily. Refer to Modelling Repeating Sub-Sections for further information.

- **User Solver Variables**: You can directly augment the global force vector to simulate an arbitrary time-varying load. It is even possible to directly modify the constitutive finite element matrices should you have very specialised modelling requirements. Refer to J04 - User Solver Variables for further information.

- **Run-Time Performance**: Flexcom 8.6 has shown to be approximately 5% faster on average than Flexcom 8.4 for time domain simulations.

- **Pipe-in-Pipe Models**: It's now possible to alternate between fixed and sliding connections, affording you greater modelling options. Refer to Sample Applications for further information.

- **Damper Element**: Velocity-dependent damper elements allow you to simulate more advanced control systems. Refer to Damper Elements for further information.

- **Additional Outputs**: Additional outputs provide greater transparency regarding internal computations which are carried out at run time. Refer to *PRINT for further information.

1.5.9 Flexcom 8.4

OVERVIEW

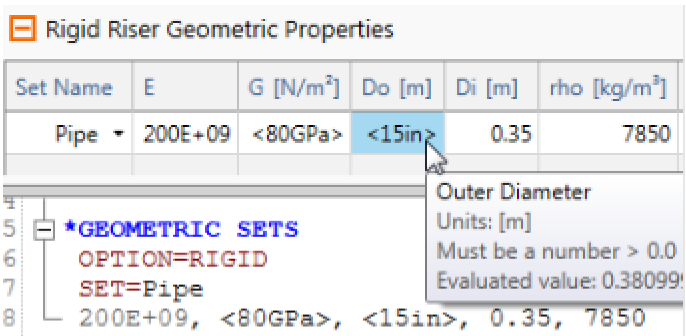
Flexcom 8.4 is another exciting new release in the Flexcom 8 series. This latest version provides several important new features along with a range of practical additions which contribute to improved user experience. Full details on all of these new features is contained in the Flexcom 8.4.1 Newsletter. Rather than following the written documentation, you may prefer to watch the Tutorial Video on YouTube.

Usability	Technical Features	Other Items of Note

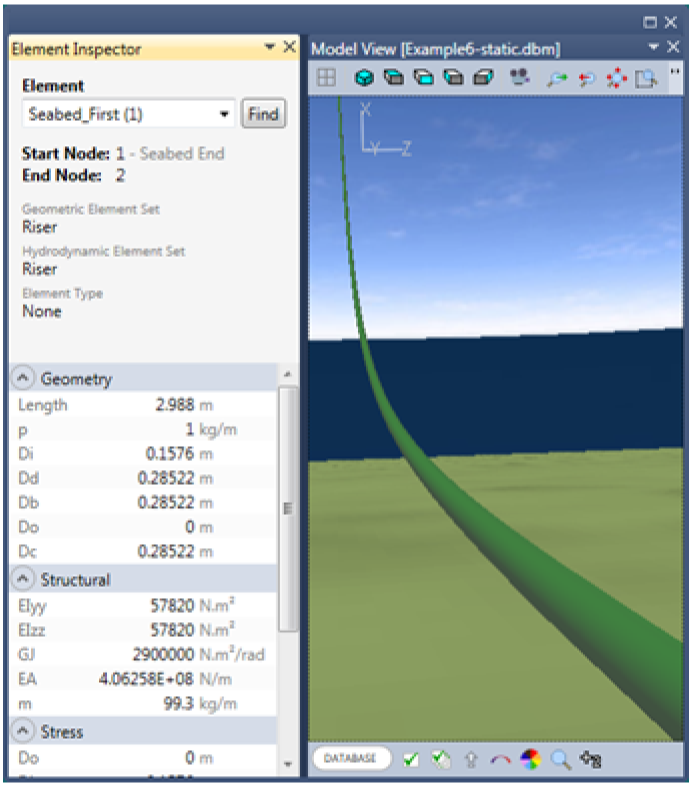
Fully integrated unit systems	Powerful 3D seabed modelling facility	Quad-Core Processing
Enhanced model view	Solution criteria automation	Miscellaneous Features
Flexcom 64-bit version	Streamlined interface with Shear7	
New documentation system	API-2RD code checking	
Reorganised examples set	Standard soil models	

USABILITY

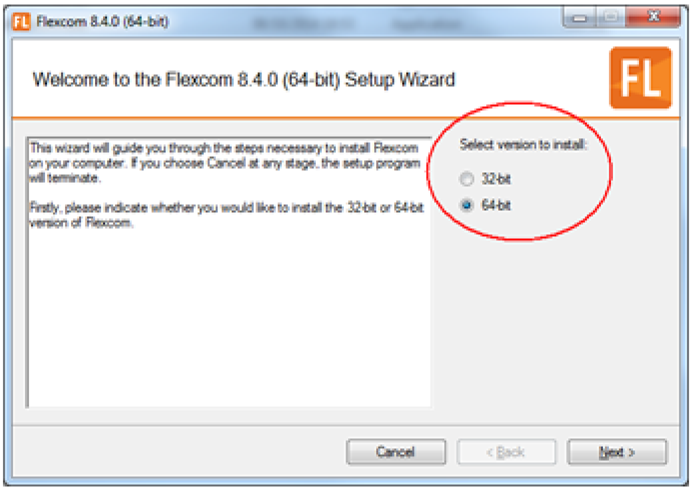
- **Fully integrated unit systems**, including standard SI and imperial options. Intrinsic conversion functions even allow you to combine various units, giving you the power and flexibility to specify inputs any way you choose. Refer to Units Systems for further information.



- **Enhanced model view**, including multiple simultaneous views, saved camera positions, a measuring tape control, and a new "inspector" feature which allows you to examine material properties directly from the structural animation. Ranges/extremes may now be explicitly defined for stress colouring. Refer to Set Number and Layout of Panes, Element Inspector, Save and Restore Camera Positions, Show Measuring Tape Control and Control Element Colouring for further information.

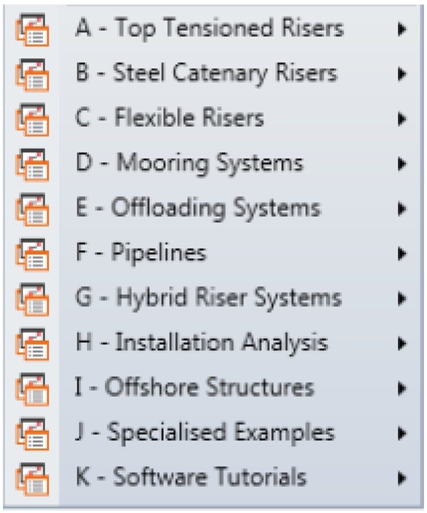


- **Flexcom 64-bit version**, designed specifically for 64-bit operating systems. The latest version offers improved run-time performance on 64-bit machines. Specifically, Flexcom 8.4 is approximately 14% faster than Flexcom 8.3.



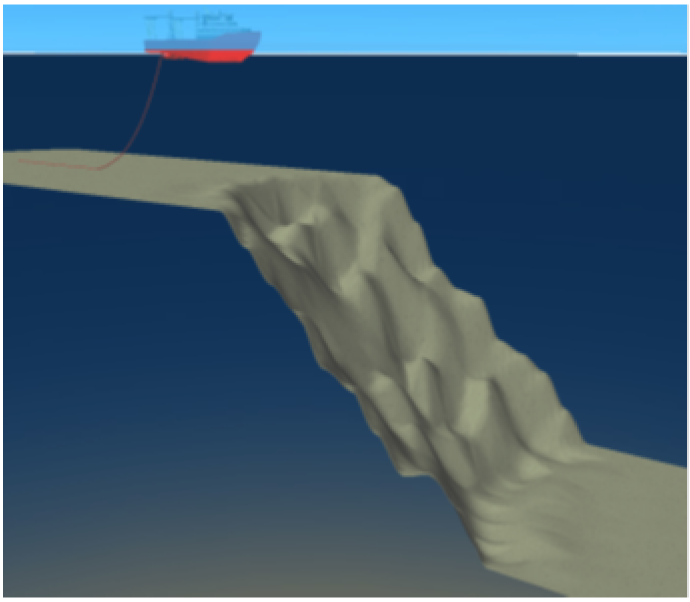
- **New documentation system**, fully integrated into the software. It also incorporates extensive cross-referencing for effortless browsing. The documentation is also available online at https://flexcom.fea.solutions/index.html.

- **Reorganised examples set** for ease of access - refer to Examples for further information. There are also some interesting new examples, including C03 - Turret Disconnect, E03 - Floating Hose, G03 - Rigid Spool, H04 - Pipe Laying and J01 - Dropped Object.

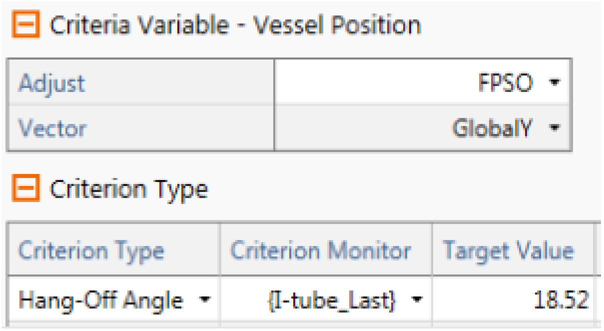


TECHNICAL FEATURES

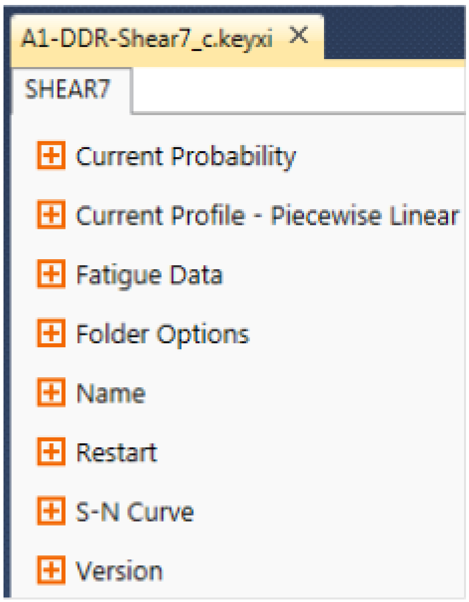
- **Powerful 3D seabed modelling facility.** Flexcom can handle complex seabed topographies with ease. You simply specify an arbitrary cloud of data points, and Flexcom's triangulation algorithm automatically generates a continuous profile via cubic spline interpolation. Refer to 3D Seabed Profile for further information, and H04 - Pipe Laying for a sample model.



- **Solution criteria automation** – you can now specify a required constraint (such as maximum bending moment in the touchdown zone), and Flexcom automatically adjust the model configuration to satisfy the specified criteria. Refer to Solution Criteria for further information.

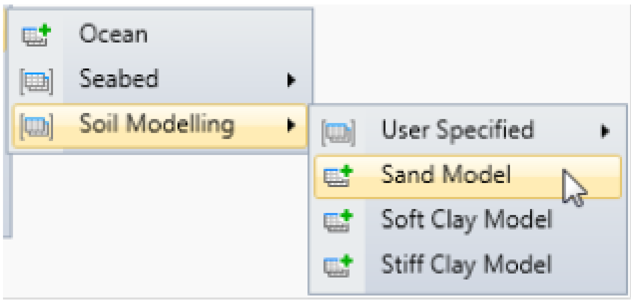


- **Streamlined interface with Shear7**, allowing you to define all the required inputs through Flexcom, providing a seamless integration of the modal and VIV fatigue analysis stages. Refer to SHEAR7 Interface for further information.



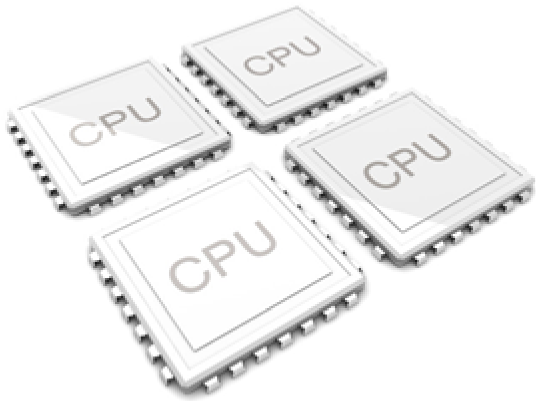
- **API-2RD code checking**, allowing you to automatically check your riser designs against the very latest codes of practice. Refer to Code Checking for further information.

- **Standard soil models** tailored towards drilling riser analysis, including sand, soft clay, stiff clay etc. Refer to Soil Modelling for further information.



QUAD-CORE PROCESSING

- **4-core processing comes as standard** with Flexcom 8.4. Effectively this means that existing Flexcom users who are using standard licence contracts can now double their processing power at no additional cost.



MISCELLANEOUS FEATURES

Aside from the major new features listed above, Flexcom 8.4 also provides numerous helpful additions, including...

- **Table Editor**. Several usability enhancements, including....

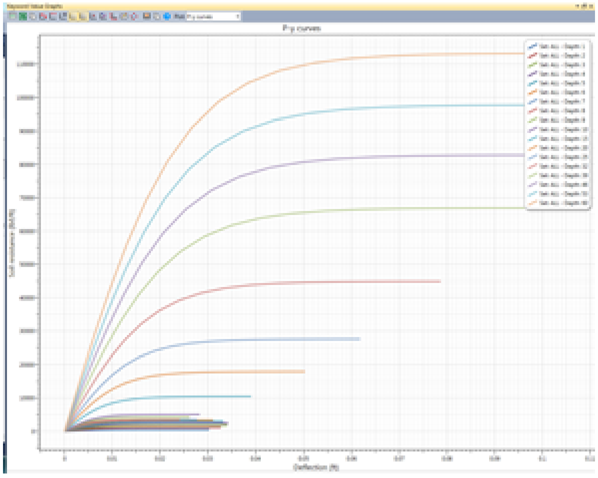
  ○ Clearly labelled units

  ○ Prompting for defined items (such as element set names)

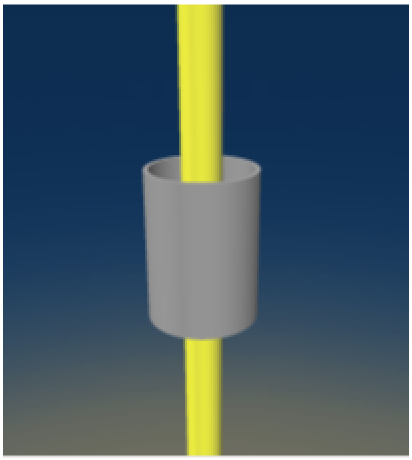
  ○ Undo/redo function

  ○ Tables listed in alphabetical order for ease of reference

- **Curve Preview**. Any non-linear relationships (e.g. stress-strain curves, P-y curves) may be previewed and visually inspected within the Flexcom user interface. Refer to Input Value Graphs for further information.

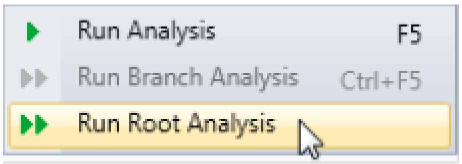


- **Curved Guide Surfaces**. Contact may now be modelled with both the outer (convex) or inner (concave) surface of a curved guide surface. This may be useful for modelling a component such as a diverter housing in a drilling riser analysis. Refer to Cylindrical Guide Surfaces for further information.



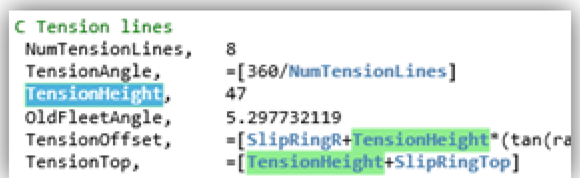
- **Set Length Timetrace**. Element set lengths may now be plotted as a function of time, which can be very helpful for monitoring winch payout. Refer to *TIMETRACE for further information.

- **Run Root**. The new *Run Root Analysis* option allows you quickly run an analysis, without having to concern yourself with input file dependencies. The software automatically runs any preceding analyses in a restart chain, before executing the required simulation.

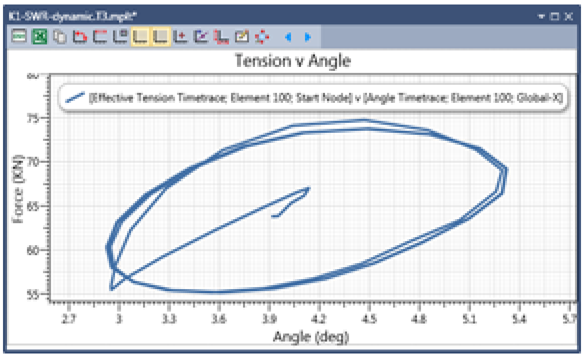


- **CSV Output**. The Plotting feature has a facility to export data from individual plots to Excel or CSV files. When working with a large number of plot files, perhaps through a script, this export procedure can get tedious. With Flexcom 8.4 you can now request an additional CSV file to accompany all plot files. Refer to Spreadsheet Output for further information.

- **Smart Select**. When selecting text in the Keyword Editor, all other instances of that selected text in the file are also highlighted. This is very useful when inspecting a model to quickly identify where named items are used. For example, simply select an element set name to quickly see all the places in the file where that element set is used. This is also very useful for tracking parameters used in equations. Similarly, all instances of the search text are highlighted when using Find in the Keyword Editor (CTRL+F).



- **Variable-Variable Plots**. These plots have been improved for Flexcom 8.4 and it is now possible to combine any two timetrace plots to create a variable-variable plot such as tension-angle.



- **64-bit Excel**. The Flexcom Excel Add-in is now compatible with both 32-bit and 64-bit versions of Microsoft Excel.

1.5.10 Returning Users

If you have used Flexcom in the past, but not recently, here's a quick summary of the main structural changes which have been made to the software in recent years:

- **Model Building**: Model building is very quick nowadays via the Lines modelling feature and mesh generation is automatic. There is an integrated Keyword Editor which facilitates rapid data specification, assisting you with helpful input prompts and command auto-completion. Any inaccurate specifications are detected by automatic syntax checking and data validation.
- **Pre-Processor Scripting**: Flexcom supports pre-defined parameters and mathematical equations, similar to computational products such as Excel and MatLab. Once a parameter is altered, all dependent parameters are automatically updated, with consequent savings for model setup effort. You can create highly parameterised models which can be easily varied subsequently (e.g. one template model could be used to examine mooring lines of different length or azimuth angle). Environmental simulation is also straightforward, as one template simulation file can be used to create all the input files required to simulate an entire wave scatter diagram.
- **Finite Element Solution**: While Flexcom has always used an advanced computational technique, power users can also use User Defined Elements to directly change elemental properties while a simulation is in progress. You can also use the simulate arbitrary loads which are dependent on instantaneous solution conditions using the User Solver Variables features. Advanced materials models such as Metal Plasticity model are now available, capable of predicting plastic deformation and capturing residual strain.

- **Animation Module**: The Model View has been enhanced considerably in recent versions, now offering multiple simultaneous views, saved camera positions, a measuring tape control, and an Element Inspector which allows you to examine material properties directly from the model view. Stress contouring offers you an intuitive visual representation of the structural stresses. Flexcom also includes a Unity Plug-In which may be used to create a fully immersive virtual reality experience from a Flexcom structural animation.

- **Post-Processing**: Recent versions of Flexcom make it very easy for you to retrieve solution outputs in a convenient format. The Summary Collation feature allows you to automatically assemble outputs from a range of load cases into a single spreadsheet. You can also plot the variation of any summary postprocessing output against any key driving parameters. For example, you can create a 3D plot of maximum effective tension as a function of both wave period and incident wave heading. Custom post-processing is available via an Excel Add-In, VBA code, or custom DLLs.

- **General User Experience**: Many years ago, user experience with Flexcom was limited due to an ageing user interface. However since Flexcom 8.1 was introduced in 2011, it now has a very modern user interface and offer a very positive user experience. The latest user interface is built using the latest software development technologies, including .NET Framework and Windows Presentation Foundation. In the past, Flexcom did not have any intrinsic unit system, with the onus placed on the user to employ a consistent set of units throughout the model. However since Flexcom 8.4 was introduced in 2015, it now offers SI and imperial unit systems. It also has an in-built unit conversion system which allows the user complete flexibility when defining inputs. For example a length input may be specified in metres, centimetres, millimetres, feet, inches etc.

A few points in relation to software licensing are noteworthy also:

- **License Models**: Flexcom has very flexible licensing options, including Ownership, Annual Subscription, Monthly Rental and Token-based license models. Nowadays you can even harness the power of cloud computing via the Cloud-based license model.

- **User Interface**: Flexcom's front end does not require any license and so tasks like model building and post-processing can be done without occupying a license seat. This open system is significantly different to most licensing models on the market and affords great flexibility to software users.

- **Cost**. Flexcom is highly cost-competitive nowadays. Many years ago, Flexcom tended to be relatively expensive to license, given that it was a pioneering software which matured more rapidly than its competitors. Nowadays Flexcom's license costs are comparable to other products in its field. Attractive discount schemes are available in many instances - please contact us for further information.

1.5.11 Known Software Faults

INTRODUCTION

The article provides information regarding known software faults in the latest version of the software on general release. Our policy is to provide complete transparency to our Flexcom user community regarding any known software errors or limitations. All of these issues will be rectified in the next maintenance release of the software. In the interim however, workarounds are suggested where feasible. The material should be self-explanatory, but feel free to contact us directly if you have any further questions.

All issues are classified according to a severity guide, which is as follows.

1. Blocker: A fault that affects all or the majority of areas of application of the program that cannot be temporarily solved by a workaround.

2. Critical: A fault that affects a widely used area of application of the program that cannot be temporarily solved by a workaround.

3. Major: A fault that affects a lightly used area of application of the program, or one that represents an unusual combination of input data that cannot be temporarily solved by a workaround.

4. Minor: A fault that affects any area of application of the program that can be temporarily solved by a workaround.

5. Trivial: A fault that represents only a very minor inconvenience to the user that does not ne a workaround.

FAULT LIST

Refer to the following articles for details on known faults in various versions of Flexcom.

- Flexcom 8.10.2

- Flexcom 8.10.4
- Flexcom 8.13.2
- Flexcom 8.13.3

1.5.11.1 Flexcom 8.10.2

OVERVIEW

The following faults are present in Flexcom 8.10.2. More detailed information on each fault is provided in the following sections.

No.	Issue	Severity
1	Hydrostatic pressure and slug flow	**Critical**
2	Run-time statistics for rotational velocity and acceleration	Minor
3	Orientation of pipe-in-pipe connections	Minor
4	Database revision number	Minor
5	Line meshing for repeating sub-sections	Minor
6	Job execution service fails to start analysis process on Windows 10 due to limited user account privileges	Minor
7	Job execution service fails to start analysis process due to blocked port	Minor
8	Analysis job appears paused if Flexcom project is located on a mapped network drive	Minor
9	Sliding Pipe-in-Pipe Connections	**Major**

10	Use of equations in the *P-y keyword	Minor
11	RAO conversions from WAMIT	Minor
12	AVI Studio unaware of nodal displacement scaling option	Minor
13	Keyword editor very slow for models which contain a large number of lines	Minor
14	Information messages relate specifically to nodes and elements even if a model has been constructed with Lines	Minor
15	Network licensing client app issues warning messages about unlicensed modules	Minor
16	Equivalent nodes in pipe-in-pipe models	Major
17	User-Defined Wave Spectrum in Frequency Domain	Minor

DETAILED INFORMATION

Issue 1: Hydrostatic pressure and slug flow

- Related Topics: Hydrostatic Pressure, Slug Flow, Buoyancy Forces
- Description: This issue could possibly be viewed as a software improvement rather than a program fault, in the sense that the software behaviour is consistent with the program documentation. However, given its potential significance for engineering design, it is being categorised as a critical issue in order to raise awareness amongst the Flexcom user community. This issue is quite specific, so a detailed explanation is provided in the following bullet points.

- The hydrostatic pressure differential due to slugs (e.g. gas) is usually relatively minor when compared with the hydrostatic pressure associated with the typically heavier surrounding internal fluid (e.g. oil). In situations where slugs are relatively short in length, any inaccuracy in the current numerical modelling approach is minimal. For longer slugs however, the inaccuracy in hydrostatic pressure, particularly for non-horizontal elements, can lead to incorrect buoyancy forces being modelled by Flexcom.

- Let's consider an extreme case for illustrative purposes. Imagine you are modelling a vertical pipe which is supported at one end using a fixed boundary condition. When the pipe is full of internal fluid, the weight of the fluid is correctly modelled, and the reaction force at the support location is consistent with expectations. Now imagine that a very long slug is passed through the pipe – sufficiently long that the pipe becomes fully filled with slug material. In this case, the weight of material within the pipe remains consistent with the fluid density rather than the slug density. Naturally the reaction force at the support location is incorrect.

- The hydrostatic pressure within an element is based on the internal fluid definition and any local changes in hydrostatic pressure caused by the presence of slug flow is neglected by Flexcom 8.10.2 (and all previous versions). The program operation is consistent with expectations, as the software documentation clearly states that… "*It is important to note that the hydrostatic pressure term is computed solely on the basis of the internal fluid definition and is unaffected by presence of any slug flow. It would not be numerically feasible for Flexcom to attempt to compute and sum up local variations in hydrostatic pressure in consecutive elements for different fluid densities and different elevations*".

- The gravitational and inertial forces for an element containing fluid or slug are modelled correctly in Flexcom 8.10.2. For any element which is either partially or wholly filled with a slug, the number of integration points for that particular element is temporarily increased to the maximum number (which is equal to 10, assuming the user is not already using this number). The composition of the internal fluid is then determined on an integration point-by-point basis – each integration point is assumed to govern a section of the element in the region of that point. This ensures that the composition of the entrained fluid is accurately quantified, which means that relatively long elements may still be used without compromising the accuracy of the slug flow model.

    - Buoyancy forces are modelled in Flexcom using a summation of all the pressure forces acting on a particular element. In the case of a horizontally aligned element, the pressure forces acting on the element end faces cancel each other out so there is no net horizontal force. This means that for a horizontal element, whether it contains fluid or slug or a mixture of both, the buoyancy forces are modelled correctly. In the case of a vertically aligned element, the pressure forces acting on the element faces produce a net vertical force. Where the element is filled with fluid, the pressure differential correctly simulates the weight of entrained fluid. However, if the element is slug filled (or partially filled), the effective weight modelled is incorrect as it remains consistent with the surrounding fluid. In the case of an element of arbitrary orientation, the degree of inaccuracy is related to its angular orientation with respect to the horizontal plane.

- Workaround: No feasible workaround exists.

Issue 2: Run-time statistics for rotational velocity and acceleration

- Related Topics: Database Files

- Description: If you request the storage of run-time statistics via the `*DATABASE CONTENT` keyword, any subsequent results for nodal rotational velocity and acceleration will be incorrect. Specifically, the values are presented in terms of radians (e.g. radians/sec) but they are labelled as degrees (e.g. degrees/sec).

- Workaround: You can manually apply a scale factor to convert from radians to degrees, or simply suppress the storage of run-time statistics and allow the post-processor to compute the required statistics.

Issue 3: Orientation of pipe-in-pipe connections

- Related Topics: Pipe-in-Pipe Contact Modelling

- Description: The orientation of a pipe-in-pipe connection may be computed incorrectly if the primary node forms part of an element which does not belong to the primary element set. A situation where this might occur would be one where a vertical outer pipe, containing a vertical inner pipe, is also connected to a horizontal pipe via a T-piece fitting. When determining the local orientation for a pipe-in-pipe connection, Flexcom searches for an element which contains the primary node. As soon as it finds such an element, it assumes this to be the primary element, even if there are other more suitable elements. Although the problem is generally quite rare, it has the potential to dramatically alter a model, and would not be immediately obvious to a Flexcom user.

- Workaround: You can avoid the problem by ensuring that the primary elements (i.e. the vertical pipe in the hypothetical case mentioned above) are defined before any additional elements (i.e. the horizontal pipe). Feel free to contact us should you have any concerns about a particular model you are working on.

Issue 4: Database revision number

- Related Topics: Motion Database File Structure, Database Access Routines, VBA, Excel-Add-in

- Description: All database files created by Flexcom are marked with a database revision number. This is a unique identifier which denotes the contents and storage format of the database file. When a new software release adds new information to the database file, the database revision number is incremented. This ensure that any user-defined custom post-processing routines remain compatible with database files created by different versions of Flexcom. When Flexcom 8.10 was released, the motion database file was extended to include vessel velocity and acceleration terms. Due to an oversight however, the database revision number was not incremented accordingly. In theory, this means that any DBM files created by Flexcom 8.10.1 and Flexcom 8.10.2 are incompatible with user-defined custom post-processing routines. In practical terms however, you are unlikely to encounter this fault, as it only manifests itself when then vessel velocities and accelerations (parameter No.20) are explicitly requested for storage under the `*DATABASE CONTENT` keyword.

- Workaround: No action is required unless you have explicitly set the *Vessel Velocity and Acceleration* storage option to *Yes*, in which case you should reset it to *No*.

Issue 5: Line meshing for repeating sub-sections

- Related Topics: Meshing Algorithm, Modelling Repeating Sub-Sections

- Description: If you are modelling a line which is composed of repeating sub-sections, the mesh density generated by the software may not be entirely consistent with expectations. Specifically, if you explicitly specify a *Min. Element Length at End* for any section, Flexcom inadvertently overwrites the *Min. Element Length at Start* for the next section with this value. The issue is very minor, and it is highly unlikely to be even noticed the vast majority of users.

- Workaround: Leave all the inputs for *Min. Element Length at End* blank, in which case they will default to the corresponding *Min. Element Length at Start* values. In practical terms, both inputs are likely to be very similar (if not identical) in the vast majority of cases.

Issue 6: Job execution service fails to start analysis process on Windows 10 due to limited user account privileges

- Related Topics: Job Execution Service

- Description: The execution service fails to start on a Windows 10 operating system if the *Local Service* account does not have full access to the "ServiceProfiles\LocalService" in the Windows installation folder. This happens because all services running under the *Local Service* account are required to store their configuration files in that folder.

- Workaround: Ask your local IT personnel to grant full access rights to the %Windows%\ServiceProfiles\LocalService folder for the *Local Service* account.

Issue 7: Job execution service fails to start analysis process due to blocked port

- Related Topics: Job Execution Service

- Description: If port 8000 is locked by another application, the execution service fails to initialise fully, which in turn means that the finite element engine fails to start. From a user perspective, it appears that there is a connectivity issue between user interface and the execution service (i.e. traffic light icon appears red). The service will be shown as running in *Windows Task Manager*, but in the *System Application Log* a message will appear stating that the service failed to start. If *Debug Logging* is enabled in Flexcom, a communication exception will be recorded, stating that port 8000 is being used by another application.

- Workaround: Resolve the conflict by closing the other application, freeing up port 8000 for use by Flexcom.

Issue 8: Analysis job appears paused if Flexcom project is located on a mapped network drive

- Related Topics: Analysis Jobs

- Description: If you are running a Flexcom project over a mapped drive, any new analysis jobs will appear paused in the Analysis Status View. You may see an error relating to low disk space, but this is misleading. A sample project on a mapped drive might appear in Windows Explorer as follows… U:\Flexcom\Project.fcproj. The same project stated with the full network path might appear as follows… \\Server Name\Drive Name\Folder Name\Flexcom\Project.fcproj.

- Workaround: Use the full network path when opening the Flexcom project, rather than the mapped drive location.

Issue 9: Sliding Pipe-in-Pipe Connections

- Related Topics: Sliding Connections

- Description: Sliding pipe-in-pipe connections may be set-up incorrectly by Flexcom 8.10.1 and Flexcom 8.10.2. The algorithm which selects the appropriate secondary node from the user-specified secondary pipe set no longer functions correctly following a restructuring process undertaken during the development of Flexcom 8.10.1. This fault relates only to models with both sliding pipe-in-pipe connections and pipe-in-pipe sections. Specifically, inactive pipe-in-pipe connections (e.g. if an inner pipe is being inserted into an outer pipe, many of the connections will be initially inactive) do not become active subsequently to reflect changes in the physical configuration which occur due to relative sliding (e.g. when the inner pipe is fully inserted and contained within the outer pipe) if the sliding element set referenced under the *PIP CONNECTION keyword is also defined as an inner pipe-in-pipe section under the *PIP SECTION keyword. If you are modelling a sliding pipe-in-pipe scenario, it is strongly advised that you perform some spot-checks on the connected node pairs using the *PRINT keyword.

- Workaround: If you create a duplicate element set, this will allow you to reference different element set names in the *PIP SECTION and *PIP CONNECTION keywords, and thereby avoid the software fault. The sample keyword inputs below provide an illustrative example. Strictly speaking, this fault could be classified as 'minor' given that a feasible workaround exists (consistent with the fault severity definitions provided above). However, given its potential significance for engineering design, it is being categorised as a 'major' issue in order to raise awareness amongst the Flexcom user community.

```
C Create a duplicate element set for the inner pipe

*ELEMENT SETS

SET=InnerPipeHydrodynamics

SUBSET=InnerPipe

C Reference the duplicate inner pipe element set in the *PIP SECTION keyword

*PIP SECTION

OUTER=OuterPipe, INNER=InnerPipeHydrodynamics

C Reference the original inner pipe set in the *PIP CONNECTION keyword

*PIP CONNECTION

GEN={OuterPipe_Start}, {OuterPipe__End}, 1, SET=InnerPipe,
CURVE=PowerLaw
```

Issue 10: Use of equations in the *P-y keyword

- Related Topics: *P-y, Parameters, Equations

- Description: If you reference pre-defined parameters or use equations when defining the *P-y keyword, the pre-processor fails to process the relevant data and simply passes character expressions to the finite element engine to which they appear meaningless. For example, if you specify a depth term of ‘DEPTH==[Mudline-10]’, where the mudline is located at X=0, then the pre-processor should evaluate the expression and pass ‘DEPTH=-10’ to the analysis engine. Instead it simply passes ‘DEPTH==[Mudline-10]’, and this is interpreted as ‘DEPTH=0’ by the analysis engine which is incapable of processing the equation.

- Workaround: Do not use pre-defined parameters or use equations when defining the *P-y keyword. Always specify variables explicitly in numerical form.

Issue 11: RAO conversions from WAMIT

- Related Topics: RAO Conversions, *VESSEL,INTEGRATED, *RAO
- Description: When converting RAO data from WAMIT to Flexcom, Flexcom incorrectly converts the RAOs for the rotational degrees of freedom (yaw, roll and pitch). Rotational RAOs from WAMIT have units of radians per metre/foot, but the Flexcom conversion fails to take this into account and imports the coefficients as degrees per metre/foot, with performing the necessary adjustment.
- Workaround: Manually convert the RAO data from WAMIT to Flexcom.

Issue 12: AVI Studio unaware of nodal displacement scaling option

- Related Topics: AVI Studio, Scale Nodal Displacements
- Description: The AVI studio does not store any user-defined settings for nodal displacement scaling in the Model View. Hence all videos created with the AVI studio display real-life displacement values which remain unscaled.
- Workaround: No workaround exists, but the issue is not very important.

Issue 13: Keyword editor very slow for models which contain a large number of lines

- Related Topics: Keyword Editor, Line Generation Report, Lines
- Description: The Line Generation Report is slowing down the Keyword Editor unnecessarily. If your model contains a lot of lines, this report contains quite a lot of information which must be computed and presented by the Flexcom user interface. However, this information only needs to be computed when the Line Generation Report is being displayed. The window is normally switched off as typically you will not be interested in such a level of detail. It is hidden by default when the software is first run and remains hidden unless you have explicitly enabled it. In Flexcom 8.10.2 and earlier versions, the user interface computes this information at all times, regardless of whether it is required or not, and this is placing an unnecessary burden on resources.
- Workaround: Rather than using the keyword editor in the normal fashion, use the 'Open as Text Document' option, or simply use any standard text editor software.

Issue 14: Information messages relate specifically to nodes and elements even if a model has been constructed with Lines

- Related Topics: Running Analyses, Lines
- Description: Lines are a relatively recent addition to Flexcom which have proven very popular with software users. They provide an automatic mesh creation facility so that the user need not concern themselves with any particular node and element numbering scheme (as would have been the case in earlier versions of the software). Behind the scenes, Flexcom translates the user-defined lines into the fundamental building blocks of nodes, elements and cables. However, any error or warning messages issued by the program, typically during the pre-processing stage, still relate specifically to nodes and elements, which makes it awkward for a user to quickly understand the source of any problem relating to input data.
- Workaround: Examine the Line Generation Report or the Model View to relate the node and element numbers back to the lines from which they are derived.

Issue 15: Network licensing client app issues warning messages about unlicensed modules

- Related Topics: Network Licensing Client
- Description: If you are using a network license, the Network Licensing Client app continually pops into the foreground, alerting you about unlicensed software modules, regardless of whether or not the module is required for your current simulation. For example, if you are running a standard Flexcom simulation, the client app can issue a warning such as “You do not currently have a network license for Flexcom Wind”.
- Workaround: No workaround is necessary as the warning messages may simply be ignored.

Issue 16: Equivalent Nodes in Pipe-in-Pipe Models

- Related Topics: Sliding Connections

- Description: As noted in Hydrodynamic Forces, drag forces and hydrodynamic inertia on inner pipe-in-pipe elements are modelled as terms on the left hand side of the equations of motion, capturing the required coupling between the outer node's velocity/acceleration and the inner node loading. In Flexcom 8.10.1 and 8.10.2, token connections of zero stiffness are automatically inserted by the software where inner nodes have no physical connection (i.e. no contact stiffness) to the outer pipe. In situations where the outer and inner pipes have been connected using Equivalent Nodes, these additional connections can lead to unexpected behaviour in the finite element solution.

- Workaround: The problem can be avoided by redefining the element sets listed under `*PIP SECTION` to ensure that any equivalenced nodes do not form part of the section definitions. Strictly speaking, this fault could be classified as 'minor' given that a feasible workaround exists (consistent with the fault severity definitions provided above). However, given its potential significance for engineering design, it is being categorised as a 'major' issue in order to raise awareness amongst the Flexcom user community.

Issue 17: User-Defined Wave Spectrum in Frequency Domain

- Related Topics: User-Defined Wave Spectrum, Frequency Domain Analysis

- Description: If you apply a user-defined wave spectrum in a frequency domain simulation, an internal error can occur within the program and the application may terminate without issuing any explanation.

- Workaround: Use one of the standard wave spectrum definitions instead, or perform a time domain simulation with the original user-defined spectrum.

1.5.11.2 Flexcom 8.10.4

OVERVIEW

The following faults are present in Flexcom 8.10.4. More detailed information on each fault is provided in the following sections.

No.	Issue	Severity

1	Floating body hydrostatic stiffness omitted from modal analysis	Minor
2	Shear7 interface includes coatings in strength diameter	Minor
3	Wave zone feature not available	Minor
4	Criteria feature incompatible with frequency domain simulations	Minor
5	Incorrect data echo for point load DLL	Minor
6	WAMIT data import	Minor
7	Summary collation ignores nodal velocities and accelerations during collation process	Minor
8	Flexcom Wind: Morison inertial loads applied by default to a floating substructure which may be undesirable	Minor
9	Flexcom Wind: For parked conditions, blade pitch angle reverts to 0° from 90° without control system model	Minor
10	Flexcom Wind: Tower drag	Major
11	Mode shapes presented in OUT file	Minor

DETAILED INFORMATION

Issue 1: Floating body hydrostatic stiffness omitted from modal analysis

- Related Topics: *FLOATING BODY
- Description: When Flexcom assembles the stiffness matrix for Modes, the hydrostatic stiffness terms are inadvertently overlooked, so the modal solver is effectively unaware of the floating body. Note that the issue does affect static or time domain simulations.
- Workaround: You could manually insert node springs or spring elements in order to compensate for the missing terms in the modal simulation.

Issue 2: Shear7 interface includes coatings in strength diameter

- Related Topics: *COATINGS, SHEAR7 Interface

- Description: If you apply an external coating to an element, the coating diameter is passed to Shear7 as the strength. Coatings do not affect structural strength so the base diameter should be passed instead.

- Workaround: Manually edit the Shear7 input file before running the Shear7 simulation. Or simply avoid the use of coatings altogether by manually increasing the structural mass using standard geometric properties.

Issue 3: Wave zone feature not available

- Related Topics: *WAVE-GENERAL

- Description: If you specify a customised wave zone, the program ignores your input and uses the default wave zone size. This means that the wave zone always extends by half a wavelength below the mean water line.

- Workaround: No workaround exists but typically none is necessary.

Issue 4: Criteria feature incompatible with frequency domain simulations

- Related Topics: *CRITERIA

- Description: If you use the criteria feature to obtain a certain model configuration in a static analysis, and then perform a frequency domain dynamic analysis as a restart, post-processing results from the frequency domain may be unavailable.

- Workaround: Use the criteria feature to obtain the desired model configuration, then manually recreate the required model adjustments in a separate/unrelated static analysis, before proceeding to frequency domain dynamics.

Issue 5: Incorrect data echo for point load DLL

- Related Topics: *LOAD

- Description: Input data echo for point loads applied via DLL may be incorrect. The node number quoted refers to the index/order of the node in the overall meshing scheme rather than the user defined node number. The issue is purely cosmetic and does not affect the simulation in any way.

- Workaround: None necessary.

Issue 6: WAMIT data import

- Related Topics: Hydrodynamic Data Importer
- Description: Data is imported from WAMIT files for one wave heading only.
- Workaround: Manually import the relevant RAO data, or repeatedly import a reduced set of data from a modified WAMIT data file.

Issue 7: Summary collation ignores nodal velocities and accelerations during collation process

- Related Topics: Summary Postprocessing Collation
- Description: The summary collation tool is capable of post-processing nodal displacements but ignores velocities and accelerations.
- Workaround: Manually collate the required information from the summary output (SUM) files.

Issue 8: Flexcom Wind: Morison inertial loads applied by default to a floating substructure which may be undesirable

- Related Topics: Flexcom Wind
- Description: For models built with the Flexcom Wind module, elements representing the floating substructure have a normal inertia coefficient of 1.0 (it is actually the added mass coefficient, which is typically set to zero by the user, plus 1.0). This means that during any dynamic simulations, there is an additional inertia term stemming from Morison's equation which is undesirable. Given that users are typically using the radiation-diffraction approach to represent the majority of wave induced loads, they are using Morison's equation to simulate viscous drag effects only. Therefore, the model should have non-zero drag coefficients only, with any added mass and inertia coefficients all set to zero.
- Workaround: Manually set the relevant inertia terms to zero in the main Flexcom keyword file.

Issue 9: Flexcom Wind: For parked conditions, blade pitch angle reverts to 0° from 90° without control system model

- Related Topics: Flexcom Wind

- Description: For parked conditions, the blade pitch angle is typically initialised to 90° via the *AERODYN DRIVER keyword. If you are not modelling a control system (which would be very unusual), the blade pitch angle reverts to 90° at the beginning of the simulation.

- Workaround: Use ServoDyn to control the blade pitch angle at 90°, in addition to initialising it using the AeroDyn driver keyword.

Issue 10: Flexcom Wind: Tower drag

- Related Topics: Flexcom Wind

- Description: Tower drag loads are based on relative velocity between the wind and the tower. Due to an error in he AeroDyn code, the computation uses tower displacement terms rather than tower velocities.

- Workaround: None.

Issue 11: Mode shapes presented in OUT file

- Related Topics: Modal Analysis

- Description: Occasionally the mode shape presented in the OUT file can differ from the Shear7 input file, due to a quirk in the normalisation process.

- Workaround: None, but none is necessary, as the mode shapes contained in the Shear7 input file are correct.

1.5.11.3 Flexcom 8.13.2

OVERVIEW

The following faults are present in Flexcom 8.13.2. More detailed information on each fault is provided in the following sections.

No.	Issue	Severity
1	Wind turbine simulations fail due to a licensing issue	Major

2	*FLOATING BODY keyword ignores node label information, leading to incorrect application of hydrodynamic loading	Minor
3	Program crash when reduced database storage is used in combination with mooring line elements	Minor
4	Sliding pipe-in-pipe connections may be assigned an incorrect orientation vector	**Major**

DETAILED INFORMATION

Issue 1: Wind turbine simulations fail due to a licensing issue

- Related Topics: Network Licensing
- Description: It is not possible to run wind turbine simulations due to a fault in the licensing code.
- Workaround: None. You must upgrade to a later version.

Issue 2: *FLOATING BODY keyword ignores node label information, leading to incorrect application of hydrodynamic loading

- Related Topics: Floating Body
- Description: The *FLOATING BODY keyword accepts nodal inputs, allowing you to identify the centre of gravity, centre of buoyancy, centre of hydrodynamic forces (also known as the force RAO reference point) and centre of drag. These inputs may be specified as node numbers or node labels. If you specify node labels, the program ignores the CoB, CoF and CoD labels, and applies all forces at the centre of gravity location. This includes hydrostatic stiffness, force RAOs, added mass, radiation damping and (concentrated) viscous drag.
- Workaround: Specify node numbers rather than node labels.

Issue 3: Program crash when reduced database storage is used in combination with mooring line elements

- Related Topics: Customised Database Contents

- Description: If your model contains elements with Geometric Properties in Mooring Line Format, and you request reduced database storage via the Customised Database Contents option, the simulation can fail due to a database writing error.

- Workaround: Request full database storage via the Standard Database Contents option.

Issue 4: Sliding pipe-in-pipe connections may be assigned an incorrect orientation vector

- Related Topics: Sliding Connections

- Description: Sliding pipe-in-pipe connections may be set-up incorrectly by Flexcom 8.13.2 and Flexcom 8.13.3. The algorithm which selects the appropriate primary and secondary elements does not function correctly in all cases, following a restructuring process undertaken during the development of Flexcom 8.13. This fault relates only to models with sliding pipe-in-pipe connections and it is usually encountered when there are several elements which share the same node (e.g. when several elements meet at a branch/intersection point in the model). If the program selects an inappropriate element, the connection stiffness can be applied in an incorrect direction.

- Workaround: There is no workaround, so you are advised to upgrade to Flexcom 8.13.4 or later. If it is not possible for you to upgrade, you could perform some spot-checks on the connected node pairs using the *PRINT keyword. If you are satisfied that the connections are set-up correctly for your particular model, you may continue to use Flexcom 8.13.2 or Flexcom 8.13.3.

1.5.11.4 Flexcom 8.13.3

OVERVIEW

The following faults are present in Flexcom 8.13.3. More detailed information on each fault is provided in the following sections.

No.	Issue	Severity
1	*FLOATING BODY keyword ignores node label information, leading to incorrect application of hydrodynamic loading	Minor

2	Sliding pipe-in-pipe connections may be assigned an incorrect orientation vector	Major
3	Snapshot plot of axial strain displays one spurious point for the first element in the set	Minor
4	Job execution service can report simulations as "Terminated" ( ) even though they have completed successfully.	Minor
5	Network licensing client app can fail to acquire available license	Minor
6	Hydrodynamic Data Importer does not read QTF data from ANSYS Aqwa files	Minor

DETAILED INFORMATION

Issue 1: *FLOATING BODY keyword ignores node label information, leading to incorrect application of hydrodynamic loading

- Related Topics: Floating Body
- Description: The *FLOATING BODY keyword accepts nodal inputs, allowing you to identify the centre of gravity, centre of buoyancy, centre of hydrodynamic forces (also known as the force RAO reference point) and centre of drag. These inputs may be specified as node numbers or node labels. If you specify node labels, the program ignores the CoB, CoF and CoD labels, and applies all forces at the centre of gravity location. This includes hydrostatic stiffness, force RAOs, added mass, radiation damping and (concentrated) viscous drag.
- Workaround: Specify node numbers rather than node labels.

Issue 2: Sliding pipe-in-pipe connections may be assigned an incorrect orientation vector

- Related Topics: Sliding Connections

- Description: Sliding pipe-in-pipe connections may be set-up incorrectly by Flexcom 8.13.2 and Flexcom 8.13.3. The algorithm which selects the appropriate primary and secondary elements does not function correctly in all cases, following a restructuring process undertaken during the development of Flexcom 8.13. This fault relates only to models with sliding pipe-in-pipe connections and it is usually encountered when there are several elements which share the same node (e.g. when several elements meet at a branch/intersection point in the model). If the program selects an inappropriate element, the connection stiffness can be applied in an incorrect direction.

- Workaround: There is no workaround, so you are advised to upgrade to Flexcom 8.13.4 or later. If it is not possible for you to upgrade, you could perform some spot-checks on the connected node pairs using the *PRINT keyword. If you are satisfied that the connections are set-up correctly for your particular model, you may continue to use Flexcom 8.13.2 or Flexcom 8.13.3.

Issue 3: Snapshot plot of axial strain displays one spurious point for the first element in the set

- Related Topics: *SNAPSHOT

- Description: If you request a plot of axial strain, the plot will contain a large and spurious value for the first node of the first element. All other axial strain data is correct barring this point, but the plot looks totally incorrect as a result.

- Workaround: Manually edit the relevant file (e.g. MPLT, TAB, XLSX), replacing the incorrect data point with an adjacent one.

Issue 4: Job execution service can report simulations as "Terminated" even though they have completed successfully

- Related Topics: Job Execution Service

- Description: Occasionally the job execution service can report simulations as Terminated () even though they have completed successfully. This is caused by a delay in the communication services between the user interface and the finite element engine. The problem only occurs sporadically, but seems to be most prevalent in large batch runs, when the machine is fully loaded and all CPUs are busy. It can be a source of inconvenience as a large analysis job can stall halfway through, as it can prevent files in a restart chain from starting if earlier ones are deemed to have failed.

- Workaround: One way to reduce the likelihood of the problem occurring is to keep the user interface open at all times. Flexcom reserves one CPU for the user interface while it remains open, and this helps to alleviate congestion in the communication services. Some users close the user interface while a large batch run is in progress in an effort to make all CPUs available for simulations, but this is not recommended. Where the problem does occur, you should manually re-run any simulations which are marked as "Terminated", and then run all dependent analyses in the restart chains.

Issue 5: Network licensing client app can fail to acquire available license

- Related Topics: Network Licensing Client

- Description: If you are using a network license, the Network Licensing Client app can occasionally fail to acquire a license, even though there are free seats available. Where this happens, your Flexcom simulation will fail to run. A related issue is that the app can launch several instances of itself, whereas only one is required, and these can be seen in the Windows taskbar (near the Windows system clock). The problem seems to affect web-hosted licenses in particular.

- Workaround: Contact Wood if you experience issues and we will provide you with a new Client app which you can install on your machine.

Issue 6: Hydrodynamic Data Importer does not read QTF data from ANSYS Aqwa files

- Related Topics: Hydrodynamic Data Importer

- Description: Any QTF data contained in .LIS files generated by ANSYS Aqwa is ignored during the data importation process.

- Workaround: Manually import the relevant QTF data. Contact Wood if you need assistance.

1.6 Installation

This section contains information on:

- System Requirements describes the general guidelines that outline the minimum recommended requirements for your machine in order to successfully run Flexcom.

- Software Installation guides you through the various stage of the software installation process.
- Network Licensing provides information about setting up and running network licensing for Flexcom.
- Downloads provides a range of useful download links associated with Flexcom.
- Flexcom-on-the-Cloud discusses the possibility of accessing Flexcom from the cloud using just a web browser (no local software installation is required in this case).

1.6.1 System Requirements

SUPPORTED OPERATING SYSTEMS

The following versions of Windows are supported:

- Windows 10 (64 bit)
- Windows Server 2019 (64 bit)

Note also that while other Windows operating systems are not officially supported (i.e. they are not included in the standard test plan implemented by Wood prior to each new release), the software may also be compatible with such systems.

Naturally we advise users to install the version of Flexcom which matches the local operating system. For example, if you have a 64-bit machine, you should install the 64-bit version of Flexcom.

MINIMUM SYSTEM REQUIREMENTS

The following general guidelines outline minimum recommended requirements for your machine in order to successfully run Flexcom:

- Microsoft Windows operating system.
- Intel dual-core processor or equivalent. Various components of the user interface perform different operations simultaneously in the background while the program is running, so having more than one processing unit will improve the overall responsiveness of the program.

- The analysis engine of Flexcom is computationally intensive and performs best on CPUs with strong floating point maths performance. Generally speaking, the higher the processor speed, the faster the program will run.

- Hard disk with at least 10GB free space. The program installation itself only requires about 300MB, but once you begin working with the software, the size of results database files may quickly become significant.

- 2GB of RAM. Flexcom is not a particularly memory intensive application, but available amounts lower than 2GB are not recommended.

- Internet Explorer is a prerequisite for viewing the context-sensitive help. If you do not already have a copy, it may be downloaded free of charge from the Microsoft web site at www.microsoft.com.

- Any Microsoft DirectX 9.0 compatible graphics card. This is mandatory in order for the structural animation/preview to be displayed correctly.

- A USB port, for a security device. Strictly speaking this is not essential, as your license agreement may pertain to a network rather than standalone license.

- To use the Flexcom Add-in for Excel, either the 32-bit or 64-bit version of Microsoft Excel 2007 or later must be installed.

OPTIMAL HARDWARE CONFIGURATION

The above guidelines represent the very minimum specification for a machine which is running Flexcom. If you are considering purchasing a new machine, you should ideally aim for specifications such as the following:

- 8 Core CPU. The more CPUs and or Cores you have available to you, the more simulations you can perform in parallel. The standard Flexcom license is currently 4 Core, but many organisations have 8 Core licenses. If you are unsure about your current license agreement, please contact us for clarification. Note also that one Core is generally reserved for Flexcom's user interface, and if you intend running other applications also, 8-Cores is a recommended minimum.

- 8GB of RAM. Depending on the simulations you run, Flexcom can be memory intensive. To ensure that Flexcom runs smoothly and allows you to simultaneously use other applications we recommend having plenty of RAM.

- Fast speed processor with higher amounts of cache. We recommend an Intel Core-i7 or equivalent. This will ensure that Flexcom simulations run as quickly as possible.

- Fast SSD drive. On a machine with a powerful CPU and high amounts of RAM, the disk becomes a bottleneck that limits the performance. In Flexcom, some types of simulations require intensive disk access and having an SSD will make sure those simulations run faster, utilizing the full potential of the CPU and RAM on the machine.

- 1TB hard drive. Disk space tends to fill up quite quickly if you are a power user. Although relatively inexpensive (in case of HDD) to purchase from the outset, additional disk storage can be difficult to retrofit later if required. You can always purchase an external hard drive subsequently if required, but access speeds will be lower than with a local drive.

- Windows 10 (64-bit), or later operating system.

CLOUD COMPUTING

Note also that Flexcom can be accessed directly from the cloud. With this model, the Flexcom software license, plus associated computational power and storage facilities, is delivered as a single coherent package via the internet. This is an alternative model which reduces dependency on local hardware. Refer to Flexcom-on-the-Cloud for further details on the potential benefits of adopting this approach.

1.6.2 Software Installation

There are potentially two parts to the software installation.

- Firstly you must install Flexcom on each of the users' computers.

- Secondly, if your office environment is using a network license arrangement which is supported by a NetHASP hardware dongle, then you must also install License Management Software software on the server machine.

If your office is using a standalone license (e.g. a HASP hardware dongle), or indeed a network license which is supported by a web hosted license, the second step is not required.

1.6.2.1 Installing Flexcom

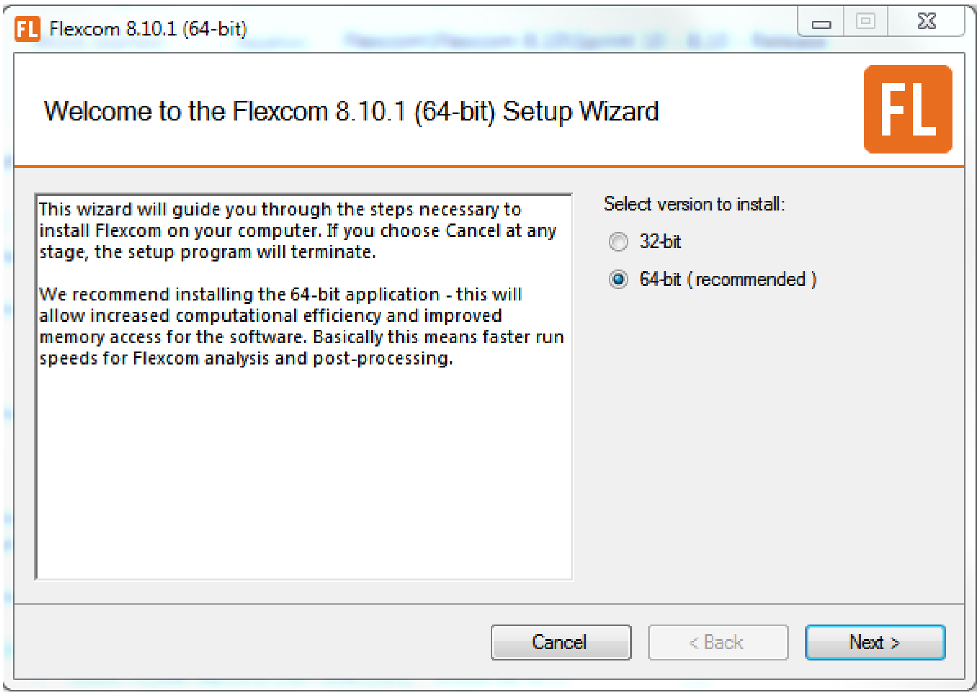
ADMINISTRATOR PRIVILEGES

The program installation requires administrator privileges. If you do not have administrator rights on your machine, you should contact your Systems Administrator before attempting to install the software.

INSTALLING FLEXCOM

To install Flexcom, please perform the following steps:

1. Download the latest version of Flexcom
2. Save the ZIP file to a temporary folder on the hard drive, and unzip the contents.
3. Right click on 'InstallFlexcom.exe' and select 'Run as administrator' to launch the Setup Wizard.
4. Select the correct version of Flexcom to suit your operating system, whether it is 32-bit or 64-bit. Most machines come with 64-bit operating systems as standard nowadays, so the majority of Flexcom users should opt for the latter to avail of increased computational efficiency and improved RAM access for the software.



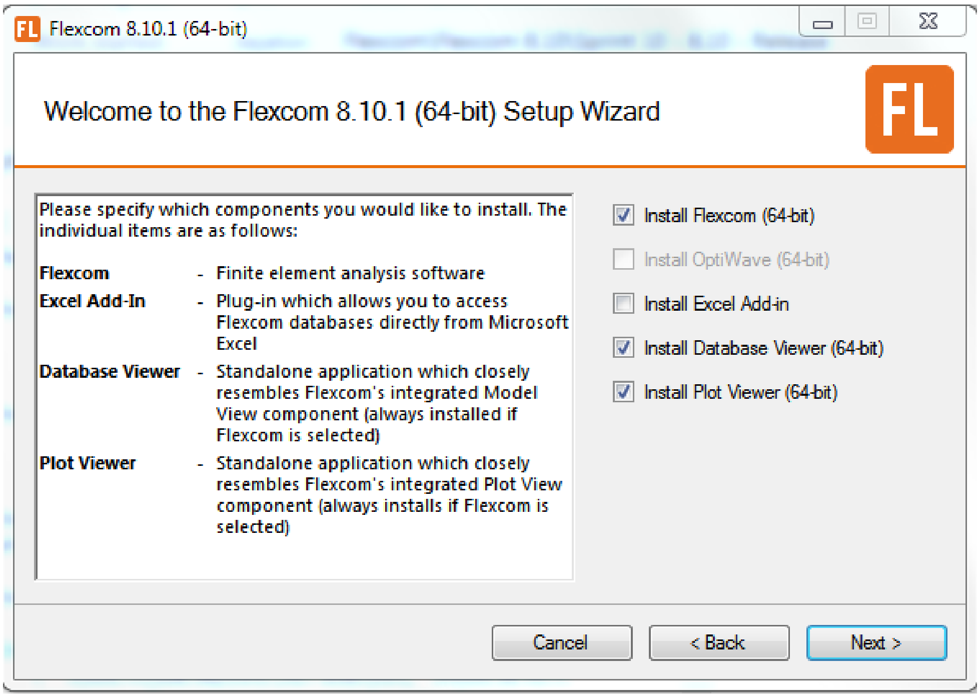
5. Select the components which you wish to install...

a. Flexcom

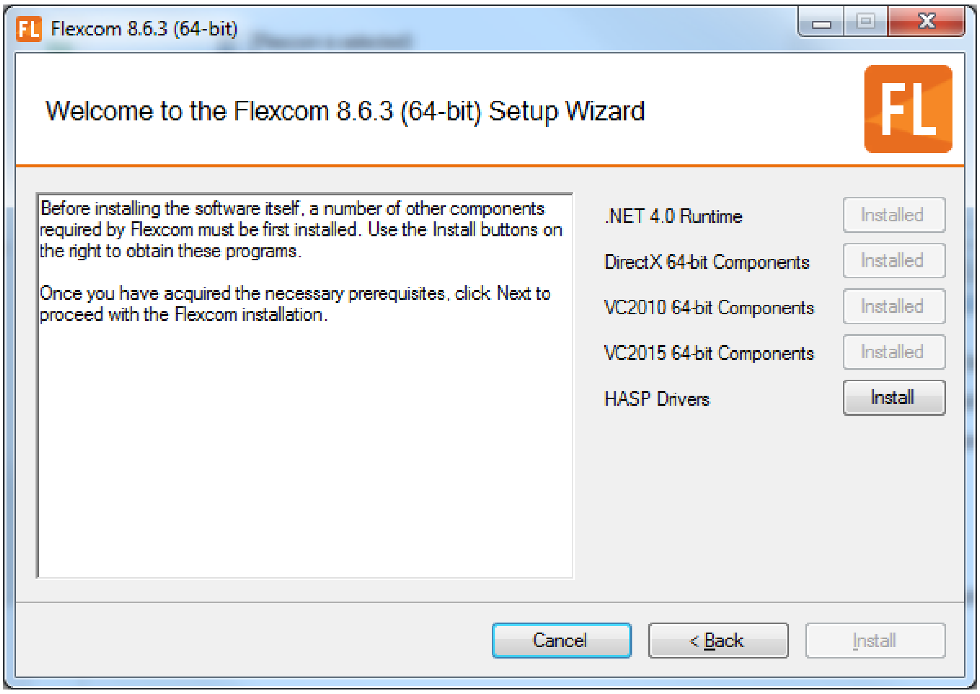
b. Excel Add-In

c. Database Viewer (a standalone application which closely resembles Flexcom's integrated Model View)

d. Plot Viewer (a standalone application which closely resembles Flexcom's integrated Plotting module)



6. Flexcom has dependencies on a number of other components, and each of these must be installed on your machine before you can proceed with the Flexcom installation. Click each of the install buttons in turn (and follow the subsequent installation instructions) until each component has been successfully installed. If a component is already present, the corresponding install button will appear greyed-out.



The vast majority of Flexcom installations run smoothly. Occasionally however, the installer cannot proceed past the prerequisites screen. Even after you have installed a particular component, the relevant button remains continuously enabled, and so the main installer cannot proceed any further. Should this happen, and you are sure that you have successfully installed all the dependencies, a useful workaround is to use the CTRL+ALT+N key combination to enable the 'Next' button. This will allow you to proceed directly to the next stage of the installation process. Note however that unless all the relevant components have been installed, Flexcom may not run properly afterwards.

7. Specify the location where you would like to store the set of illustrative examples which accompanies Flexcom.

8. Choose shortcut locations to quickly launch the software.

9. Specify the location where you would like to install the software executables on the hard drive.

10. Press confirm to proceed with the installation.

11. After the software installation has fully completed, you will need to store a license file (entitled 'MCScode') in your local Flexcom installation directory. The exact location will depend on the options selected during the installation process, but will typically be something like... "*C:\Program Files\Wood\Flexcom\Version 2022.1.1\bin*". If you have not already received your licence file, you should Contact Us.

SILENT INSTALLATION

If you are installing Flexcom on a range of computers (e.g. an IT technician performing a group update), you might like to perform a silent installation from the command line. Flexcom uses a standard installation procedure so many of the features which you would expect from a Windows installer pack are available. Typically you would append the /quiet switch to the command line so that the installer runs without displaying a user interface. In this case, no prompts, messages, or dialog boxes are displayed to the user. Further information on command line options is provided in this Windows article… https://docs.microsoft.com/en-us/windows/desktop/msi/standard-installer-command-line-options. Assuming that the pre-requisite software packages have already been installed, and this will be typically be the case if the computer already has or had some version of Flexcom installed on it, then you can bypass the ‘InstallFlexcom.exe’ program and proceed directly to the ‘Flexcom.x64.msi’ file (which installs a 64-bit version of Flexcom).

1.6.2.2 Installing License Management Software

INTRODUCTION

There are two possible modes of network licensing:

1. A NetHASP hardware dongle is placed on a network PC that then becomes the *License Server* for the network.

2. Web hosted licensing which is similar to 1 except that the licence server is on the cloud removing the additional requirement to host and maintain a server machine. If you are using this option, then you do not need to install any network licencing software.

NETHASP HARDWARE DONGLE

To install Flexcom Network Licensing software for the NetHASP hardware dongle approach, please perform the following steps:

1. Download the Network License Manager application and unzip the contents.

2. Run the NetworkLicenceManager.exe Installer Wizard.

3. In the Wizard you will see two buttons on the right:

- *Sentinel HASP Drivers*

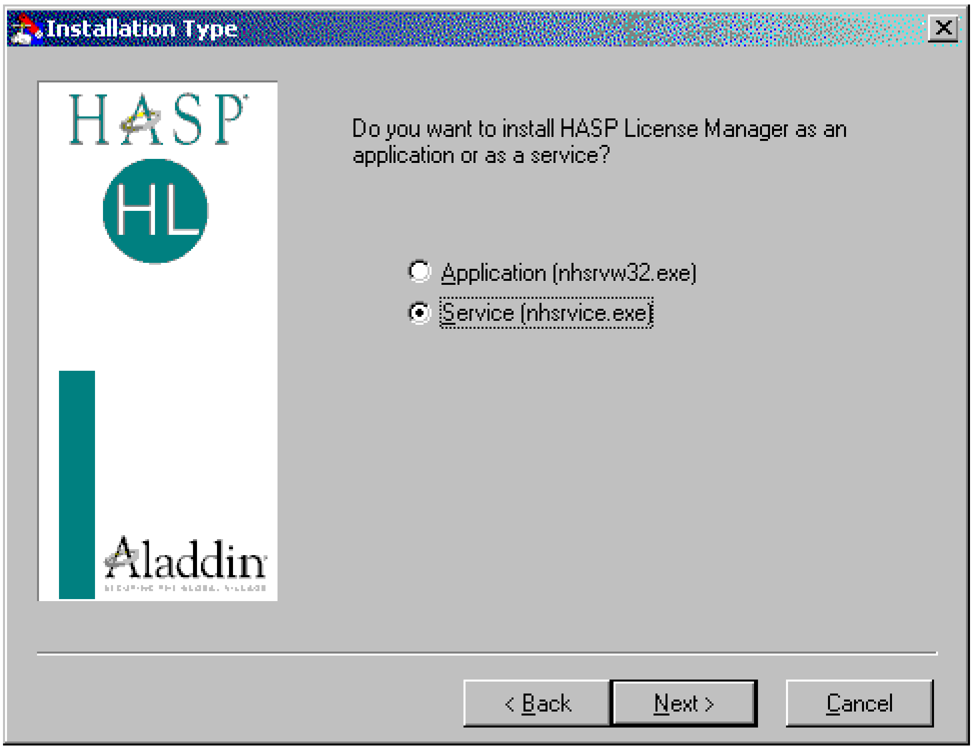
- *NetHASP License Manager*

These buttons will either state *Install* or *Installed* depending on whether the applications in question are already present on your network license server. Note that for the Installed case the button will be disabled thus preventing any unnecessary reinstallation.

4. If the Sentinel HASP Drivers button is labelled Install then you must click on it as the installation of the associated drivers is compulsory.

5. To install the license manager you should click on the Install button for the *NetHASP License Manager*. Note that you should not install it unless you have a red-coloured network dongle and an appropriate license file, as it will not work. Also, it should only be installed on the license server computer.

6. The main setup screen for the licence manager is similar to that shown below. You should select *Service* setup.



License Manager Setup

For details on how to run the License Manager, please refer to Network Licensing Manager.

1.6.3 Network Licensing

INTRODUCTION

A single-user version of Flexcom, without network licensing, requires a local software protection device (dongle) attached directly to the computer. The Flexcom licence may be shared around an office only by physically transferring the software protection device from one machine to another.

Network licensing allows multiple users of Flexcom on a network to share a pool of licenses. There are two possible modes of network licensing:

1. A NetHASP hardware dongle, programmed to permit a fixed number of users, is placed on a network PC that then becomes the *License Server* for the network. Users may get a network license from this server to run Flexcom and release it once finished, freeing the license for another user. The hardware dongle does not need to be connected to a dedicated server machine, although this approach is recommended. It may be connected to any PC on the network, including one used to run Flexcom.

2. Web hosted licensing which is similar to option 2 except that the licence server is on the cloud removing the additional requirement on the user to host and maintain the licence server. A consistent network connection is necessary for the web hosted licensing system. In the case of Ownership, Annual Subscription & Monthly Rental licences, if the connection to the license server is interrupted, a grace period of one hour is allowed for re-connection. Token-based licensing cannot work without a network connection.

FURTHER INFORMATION

Further information is contained in the following sections:

- Network Licensing Client describes the *Network Licensing Client* applet, which allows users to obtain and release Flexcom network licenses.
- How Licensing Works describes how licensing works internally, including the licensing client's automatic and manual modes.
- NetHASP Hardware Dongle describes the NetHASP (physical dongle) approach to network licensing, and contains information on the *Network License Manager* software, the *NetHASP Monitor* module and a section on *NetHASP License Troubleshooting*.
- Web Hosted Licensing describes the new "Licensing as a Service" (LaaS) offered to users from Flexcom 8.10.1 onwards.

1.6.3.1 Network Licensing Client

This section describes the Network Licensing Client applet, which allows users to obtain and release Flexcom network licences.

The Network Licensing Client is a small applet that allows the user to attach to a Network License Server and obtain a licence for Flexcom. It is installed automatically along with the rest of the Flexcom installation.

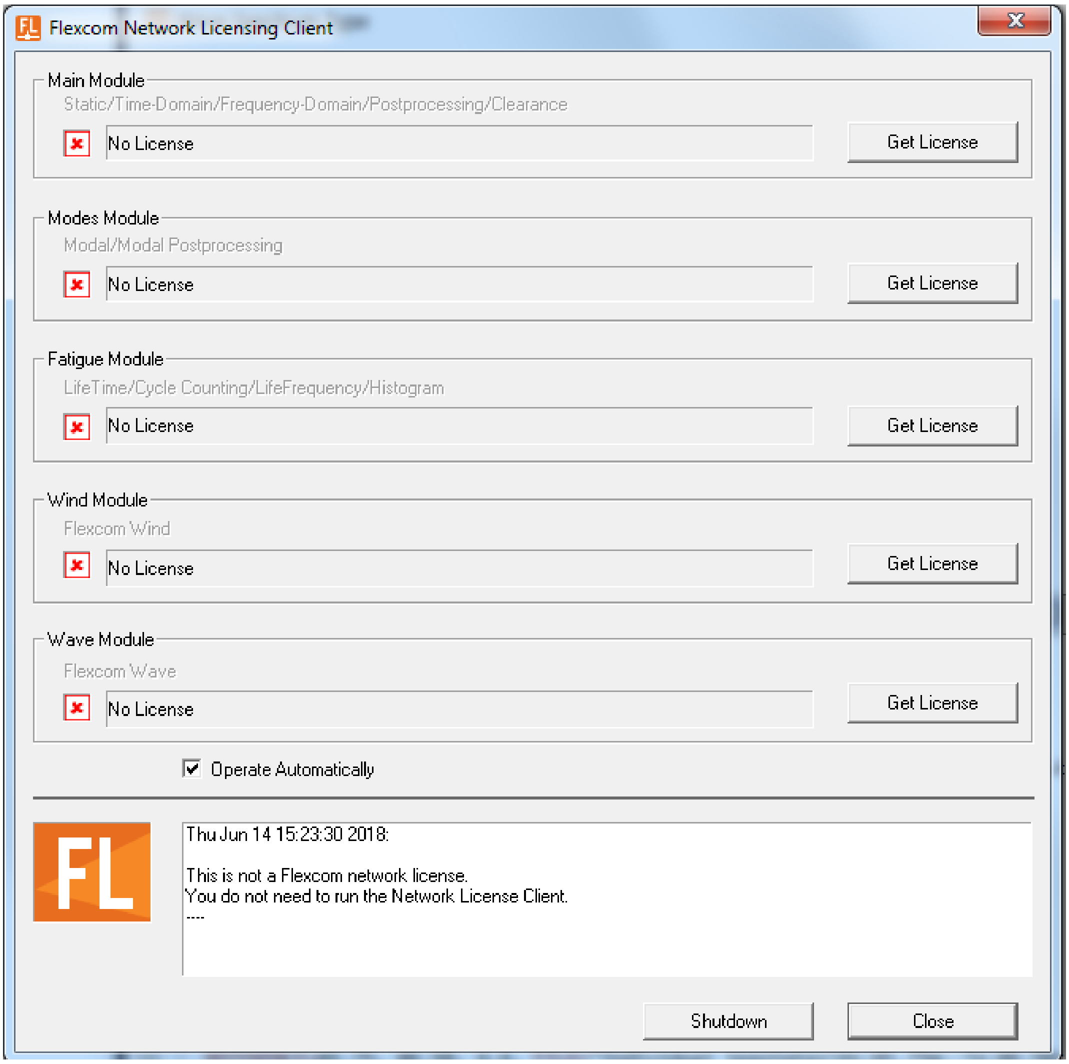
If your Flexcom software licence agreement allows you to use network licensing, you will have been issued with a network version of the software licensing file (MCScode), and the Network Licensing Client should start automatically when you start Flexcom.

When the Network Licensing Client starts, it displays a small icon in the notification area of the Windows taskbar, as shown below.

Network Licensing Client Icon

This icon is similar to the Flexcom program icon overlain with one of three symbols. A red ✗, as shown above, indicates that you do not currently have any licences. A green ✓, on the other hand, indicates that you have obtained a licence for Flexcom from the License Server. A black question-mark symbol indicates that the Network Licensing Client is searching for a licence.

Double-clicking on the icon displays the Network Licensing Client dialog, as shown below.



Network Licensing Client

Your current licence status is listed on the upper section of the dialog. The name of the license server from which your license was obtained is also listed; in the picture above, it is a server called ‘localhost’. There is also a button to allow you to manually get or release a license. Refer to How Licensing Works for further details.

There is a check box that allows you to select whether or not the Network Licensing Client will operate automatically. It can operate either manually or automatically; these two modes of operation are described in the following section. For most users, Operate Automatically is the more convenient mode.

On the lower section of the dialog, there is a status area in which messages are displayed, with a time stamp. If you have a problem getting a licence, you should open the dialog and check to see if an error message is displayed in the status area.

At the bottom of the dialog, there are two buttons, Shutdown and Close. Shutdown releases any licences you may have at present, and terminates the Network Licensing Client. Close simply closes the dialog; you can open it again by double-clicking on the icon in the Windows taskbar.

1.6.3.2 How Licensing Works

You can choose whether to have the Network Licensing Client operate automatically or manually, by selecting or de-selecting the Operate Automatically option. Automatic operation is more convenient for most users. However, ‘power users’ may prefer to use manual operation, for example to hold on to a licence indefinitely.

AUTOMATIC OPERATION

With automatic operation, the Network Licensing Client requires no user intervention. When Flexcom requires a licence, it automatically requests one. If there are no licences available, or if there is some other error, it will display a message to inform you. A number of points regarding the acquirement of licenses in automatic mode are noteworthy:

1. When you launch the Flexcom User Interface (UI), no license is requested by the software. This affords you full access to all the features provided by the UI, and avoids any locking of licenses unnecessarily. For example, you may build a model using either the Table Editor or Keyword Editor, availing of the structure preview facility while the model is under development, and you have full access to the comprehensive context-sensitive help. It is also possible to examine results from analyses which have been performed previously.
2. As soon as you proceed to performing an actual finite element analysis, Flexcom will automatically request a license from the network server, and obtain one assuming there is a free license available.
3. Once you have successfully obtained a license, it remains assigned to you for the entire duration of your current Flexcom session (you manually release it using the Network Licence Client). This prevents other users in the network from accessing your license and preventing you from performing further finite element analyses.

4. When you shut down the Flexcom UI, the software automatically releases any licences assigned to you and shuts down the Network Licensing Client, assuming that all your finite element analyses have completed. If there are any analysis runs still in progress, and you wish to close the UI, leaving these analyses to run in the background, then any licences obtained during the session will remain assigned to you. You must free these licences manually yourself at a later stage – otherwise you will prevent other users in the network from accessing them. You should also manually shut down the Network Licensing Client at this point. Before the UI closes, you will be issued with an appropriate warning message to this effect.

MANUAL OPERATION

With manual operation, Flexcom cannot request licences automatically. Instead, you must use the Get License and Release License buttons. Neither can Flexcom shut down the Network Licensing Client automatically; you must do this manually using the Shutdown button. However, if Flexcom needs a licence, or if the Network Licensing Client is no longer required and may be shut down, a message is displayed in the status area advising you of this fact.

The Network Licensing Client should start automatically when you run Flexcom. However, you can also start it manually from the Top Menu Bar within the UI, or by selecting it from the Flexcom program group in the Windows Start Menu.

HOW LICENSING WORKS INTERNALLY

The logic, as currently implemented in Flexcom 8.13, is as follows:

1. The user interface or the finite element engine calls a function in the licensing DLL.
2. The code in the DLL determines from the MCSCode file whether or not you are using a network license.
3. For a network license, the client application is requested (it is started if necessary) and the "get" command is sent to the client to acquire the license for the requested module.
4. The client application then checks if it can connect to the server and if it can it acquire the license. The server can be a physical server in the case of a NetHASP hardware dongle, or an online platform in the case of a Web-Hosted license.
5. When the license is acquired, the session duration that is configured on the licensing server is used as a license expiry timeout (this is currently configured at 10 minutes).

6. Every 5 minutes (on a timer) the client application does the following:

    - checks if the finite element engine has requested license release (analysis run is finished)
    - checks if the license session has expired
    - checks if it is less than 30 seconds to the license session expiration

7. If the license release is requested, the client will continue running and retain the license until the license session expires.

8. If the license is not released and the license session is about to expire (less than 30 seconds to expiry), the client will reacquire the license - this resets the license session to the configured duration (10 minutes).

The above logic guarantees that the license is retained for a minimum of the license session expiry timeout. This prevents frequent license requests to the license server and ensures that such requests happen about once every 10 minutes, regardless of the number of simulations which are in progress.

Web hosted license logic differs very slightly from other modes of network licensing (NetHasp and SRM). For the web hosted set-up, the client app waits for the session expiry before shutting down, so if there is a number of analyses running in succession it prevents the client app from stopping and restarting too often. This reduces the risk for license instability when working on slow or intermittent internet connections.

Token licensing is slightly different from the others. Here an additional check is performed on the available tokens. If sufficient tokens exist, a consume request is sent. It runs integrated in the product via the license DLL rather than via the client app.

1.6.3.3 NetHASP Hardware Dongle

This section describes the NetHASP (physical dongle) approach to network licensing, and contains information on the *Network License Manager* software, the *NetHASP Monitor* module and a section on *NetHASP License Troubleshooting*.

Further information is contained in the following sections:

- Network License Manager describes the *Network License Manager* software, which must run on the Flexcom *Licence Server* machine. Note that this Network License Manager is only applicable to the NetHASP hardware dongle approach, which is discussed in the proceeding section.
- NetHASP Monitor describes the NetHASP *Monitor* module, a software program which allows you to browse all the NetHASP dongles on your network.
- NetHASP License Troubleshooting describes steps that are useful for troubleshooting any issues with Flexcom network licensing.

Network Licensing Manager

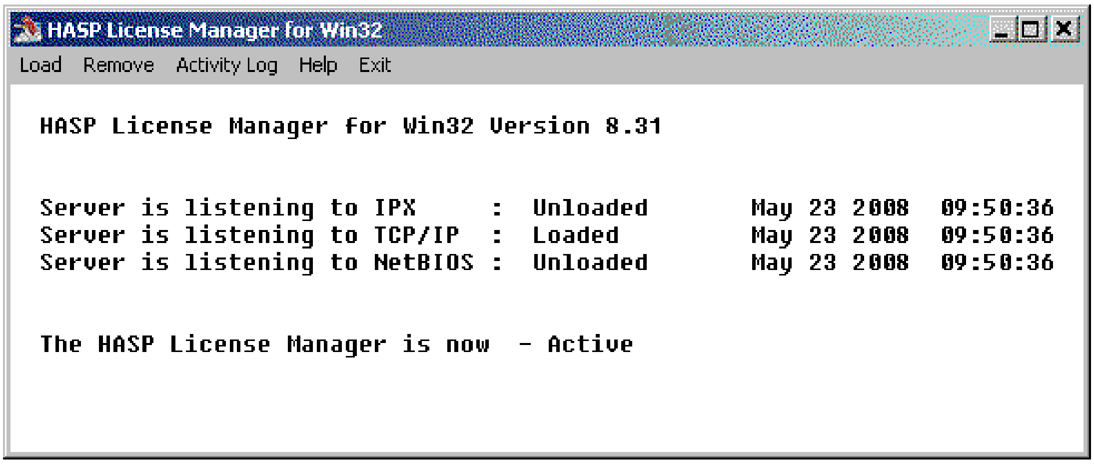
This section describes the Network License Manager software, which must run on the Flexcom Licence Server machine. Note that this Network License Manager is only applicable to the NetHASP hardware dongle approach.

The computer with the NetHASP hardware dongle attached becomes the Flexcom Licence Server for the network. The Licence Server machine must run the Network License Manager program to allow other computers on the network to access the hardware dongle.

Make sure to choose the option to install the application as a service during setup. Please note that the program’s entry is ‘Hasp Loader’ in the services list.

Alternatively, the Network License Manager program may be run from the Windows Start Menu.

The Network License Manager displays a small icon in the notification area of the Windows Taskbar, except before login on a server where it is running as a service. You can double-click on the icon to open a dialog showing its current status, as shown below.



HASP Licence Manager

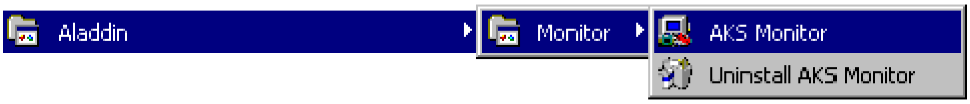
The Network License Manager should normally be left running at all times. Shutting it down logs out any attached clients and will cause any Flexcom runs in progress at the time to terminate prematurely. If clients are logged in when an attempt is made to shut down the Network License Manager, a warning will be displayed and confirmation requested before the program exits.

Very occasionally, a licence may become “suspended”. For example, if a computer crashes, or if a computer loses its connection with the local network, while it has a Flexcom network licence, the license may be deemed nominally in use although the machine is not actually accessing it any more. If that happens, the Network License Manager will eventually free the suspended licence after several hours of inactivity. If the licence is required urgently, the only way to free it quickly is to shut down the Network License Manager. This will log out all network clients, so care should be taken that other Flexcom users are notified properly before the program is shut down.

NetHASP Monitor

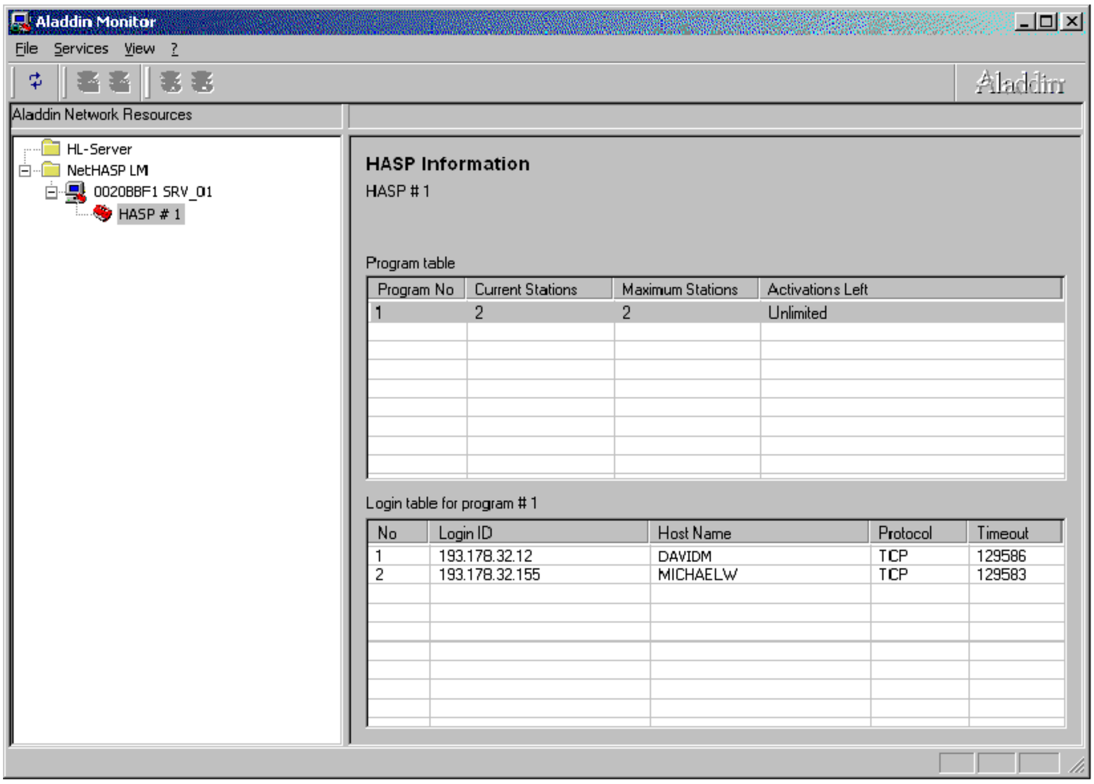
An optional module, which you may find useful as a user of Flexcom Network Licensing, is the Aladdin NetHASP Monitor. This software allows you to browse all the NetHASP hardware dongles on your network (not just Wood ones). You may then see how many licences are currently available for any particular product and which machines currently have product licences issued to them.

To install this software, simply run AKSMON32.EXE from the EXTRAS\MONITOR folder on the Flexcom installation CD-ROM and follow the on-screen instructions. A shortcut to the program will then appear in the Windows Start Menu, as shown below.



AKS Monitor Shortcut

Run the software by clicking on the AKS Monitor shortcut. A sample main program window is shown below. The options presented by this window are now illustrated by means of an example session.



AKS Monitor

In the bar on left-hand-side of the program, open the NetHASP LM folder. In this example there is only one entry here, for the server called SRV_01, which we select. You should choose the entry which corresponds to the server your Flexcom network dongle is attached to.

Under the server entry, it is possible that there may be several dongles listed. However, normally there will be just one. If there is more than one, you will have to ask your system department for assistance in identifying which is the Flexcom dongle. Here there is only one entry, HASP #1, and we select that.

On the right-hand-side of the program, the information for this dongle is shown. We can see that only Program 1 (Flexcom) is listed and that all its licences (2) are allocated. These licences are currently in use at the computers called DAVIDM and MICHAELW. See the table below for a full list of Wood program numbers.

It is worth noting that other programs for which you are licensed, if unlisted, have, by implication, all their licences available.

Base Product	Program/Module	Number
Flexcom	Flexcom	1
	Modes	2
	Fatigue	4
	Wind	6
	Wave	10
DeepRiser	DeepRiser	12
	Recoil	13
	Optima	21
PipeLay	PipeLay	30
	OptiLay	20
Miscellaneous	Stiffener	25
	Crosscom	40

Network Licensing Program Numbers

NetHASP License Troubleshooting

The following steps are useful for troubleshooting any issues with network licensing which uses the NetHASP system (physical red dongle).

PART1: ON THE NOMINATED SERVER MACHINE

1. Is the red network dongle connected to the main server machine? The red dongle must be plugged into the nominated server machine – with an IP address matching that provided to Wood.

2. Is the dongle illuminated? If not, this could indicate a problem with the USB port (see Server Point 3) or incorrect drivers (see Server Point 4).

3. Is the dongle inserted into a standard USB port? If the dongle is inserted into a SuperSpeed USB port (typically labelled 'SS' followed by the normal USB ‘branch’ symbol), it may not be recognised. If so, try using a standard USB port instead. If the dongle becomes illuminated, proceed directly to Step 5. If not, continue to Step 4.

4. If the drivers have not been installed correctly, or if they have somehow become corrupted, they may need to be reinstalled. To reinstall the drivers, follow these steps:

   a. Unplug the dongle from the machine

   b. Open the Sentinel Downloads website

   c. Download "Sentinel HASP/LDK - Command Line Run-time Installer" and unzip the files to the local hard drive

   d. Open a command prompt window as an administrator (search for "cmd" on the Windows Start menu, right-click on the executable and select "Run as administrator")

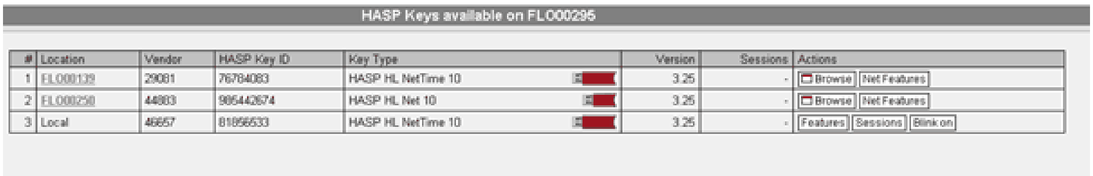
   e. Navigate to the location where you unzipped the file (you can use the command 'cd foldername' to change directory in a command prompt window; the command 'cd..' allows you to move up one directory level)

f. Type the command 'haspdinst.exe –i' and press Enter

g. Once the process completes successfully, the device drivers should now be fully installed

h. Plug the dongle back into the machine again. At this point the dongle should become illuminated. If not, please Contact Us for further instructions.

5. Is the Network License Manager running on the server machine? To check this, navigate to http://localhost:1947 within an internet browser on the server machine. Click ‘Sentinel Keys’ from the menu on the left of the page. As per the screenshot below, you should see a HASP key attached with Vendor ID 46657 and Location ‘Local’.

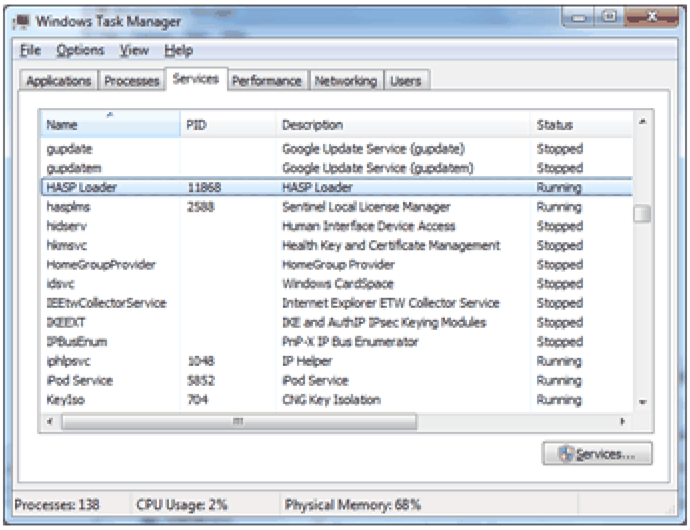


HASP Keys available on FLO00295

#	Location	Vendor	HASP Key ID	Key Type	Version	Sessions	Actions
1	FLO00139	29081	76784083	HASP HL NetTime 10	3.25	-	Browse Net Features
2	FLO00250	44883	985442674	HASP HL Net 10	3.25	-	Browse Net Features
3	Local	46657	81856633	HASP HL NetTime 10	3.25	-	Features Sessions Blink on

6. Can you not see the HASP key? If the HASP key is not present or you cannot see the page, the Network License Manager may not be installed correctly.

a. Reopen the Network License Manager Installer software, and use the Wizard to reinstall the *NetHASP License Manager* as per the instructions in the help article. Note in particular that the software should be installed as a service (rather than an application) and the setup should be run with administrator permissions.



7. Does the nominated server IP address match the MCSCode file on the client machine?

   a. Open the client MCSCode file in a text editor like Notepad++. You should see text similar to the text below. Take note of the IP address specified in the file (example IP address [123.456.78.90] highlighted in yellow).

   b. Obtain the actual IP address for the server machine. Instructions to find your IP address can be found on the Microsoft site: http://windows.microsoft.com/en-ie/windows/find-computers-ip-address#1TC=windows-7. Compare this with the IP address stored in the license file - both should be identical.

```
[Flexcom]
01/04/2014
01/04/2015
01-02-03-04-05-06
Company Name
12345
4
2014
Company Name
Company Address 1
Company Address 2

Company Name
1 2 1
FREECOM
AdfdfgPq?[Y1WK=~?!8fgjn?>$|EYN-1MGyU\f)Lh)pd%0$<$^£%=t*\$p)p(7
[Network]
1
12345, ServerName, 123.456.78.90
=I)!u8o+DUNkdjidgn9£%DY£$YS£*FHW^DGE&dfEz1zOogLF|RS7K)Rh6W}b@W
```

8. Is port 475 available for communication? Check that port 475 on the server machine is available and is not blocked by Firewall. This is the port which the license manager uses to communicate. If the port is blocked, can it be opened? If not, please use the following steps to change the port number.

   a. The port can be set in the license manager configuration file 'nethasp.ini'. To configure the nethasp.ini file, add the following to the [NH_TCPIP] section: NH_PORT_NUMBER = and set the TCP/IP port number. The default number is 475. Replace it with a port number that is open (do not use any brackets around the number).

b. Then on the license manager server, in the license manager installation directory should be a 'nhsrv.ini' file. Open it using a standard text editor and change the following line under the [NHS_IP] section: NHS_IP_portnum = 475. Change the number from 475 to match the number you put in the 'nethasp.ini' file, then re-start the license manager. Make sure that any firewall installed on your NetHASP License Manager host allows incoming connections on that port.

9. Does the server machine have more than one red/network NetHASP dongle attached to it? For example, it is possible (although unusual) to have two separate network licenses of Flexcom under your software contract, or you may have other third-party software which also requires a NetHASP dongle on the server machine. If this is the case, you will need to move the Flexcom dongle to another machine and inform Wood of the new IP address.

PART 2: ON THE CLIENT MACHINE

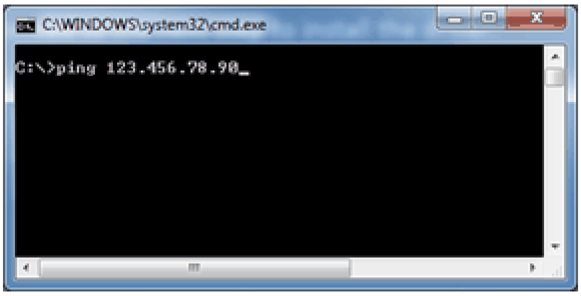
1. Is the MCSCode file placed in the Bin folder? The MCSCode file that you received from Wood should be placed in the Bin folder of the Flexcom installation directory on the client machine. For example, C:\Program Files\Wood\Flexcom\Version 2022.1.1\bin.

MCSCode

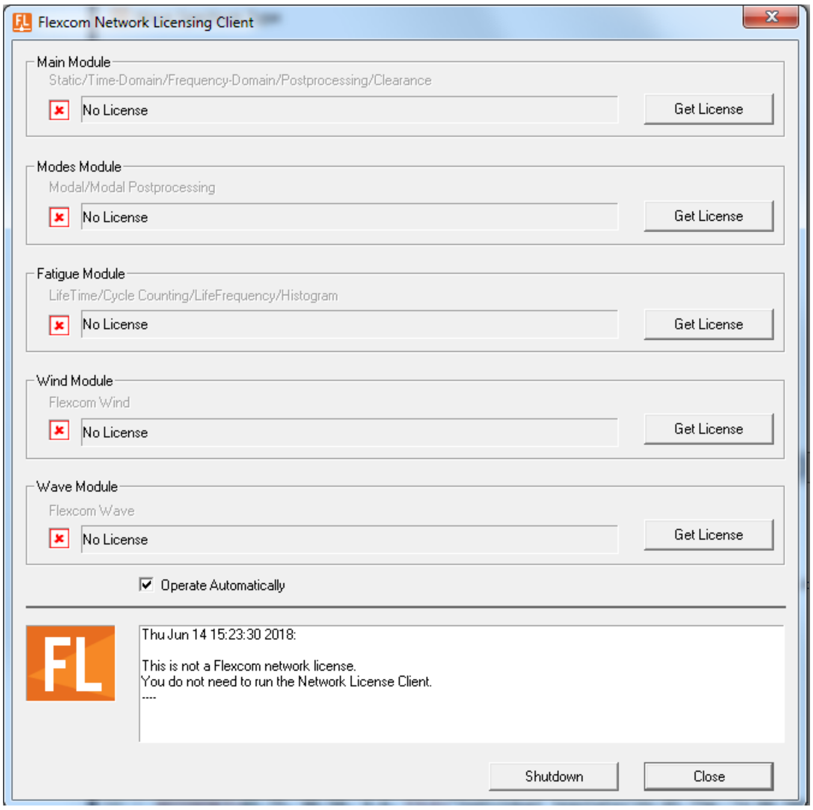
2. Can you see the license server machine on the network? Let's assume for illustrative purposes that the server IP address is 123.456.78.90. You can check that you can connect to the server machine by either:

   a. Navigating to http://123.456.78.90:1947 in your internet browser or

   b. Pinging the server by opening a command prompt in Windows, and typing ‘ping 123.456.78.90’ at the prompt, where ‘123.456.78.90’ is the actual IP address of your server machine, as provided to Wood.



3. Can you get a license manually? Open the Flexcom Network Licensing Client app by selecting *Flexcom Main Menu* > *Licensing* > *Show Network License Client*. Click the first *Get License* button for Flexcom. Do you get error message(s) in the text box or do you successfully retrieve a license from the server? Any error message would be useful to help us diagnose the problem. The entire contents of the text box should be pasted into your email to us - note that the box can have scroll bars.



4. Could there still be some problem with the communication across the network? To eliminate this, let's try moving the network dongle to the client machine. In this way, the same machine will act as both the client/user and the server/host from a licensing perspective.

   a. On the user machine, follow the steps listed in Network License Manager Installer.

b. Unplug the network dongle from the server machine, and plug it into the client machine. The dongle should illuminate. If not, this could indicate a problem with the USB port (see Server Point 3) or incorrect drivers (see Server Point 4).

c. Obtain the IP address for the user machine. Instructions to find your IP address can be found on the Microsoft site: http://windows.microsoft.com/en-ie/windows/find-computers-ip-address#1TC=windows-7. Email this IP address to Wood. Our technical support team will create a temporary license file which will use the IP address of the user machine (rather than the server machine).

d. When you receive the new license file, save it into the local Flexcom installation folder on the client machine.

e. Try to get a licence with the Network Licensing Client app on the user machine

IS YOUR PROBLEM RESOLVED NOW?

The troubleshooting tips above should have helped to resolve the problem. If you are still experiencing issues, please Contact Us and provide us with as much information as possible i.e. let us know what happened at each of the various steps, including screenshots or text output as appropriate.

1.6.3.4 Web Hosted Licensing

INTRODUCTION

Flexcom 8.10.1 onwards provides an option to use a web hosted licensing system, also known as "Licensing as a Service" (LaaS). This system offers additional flexibility to users by enabling ready to use licensing entitlements with less set-up overheads compared to traditional dongles. Several licensing types are available with LaaS:

- Ownership
- Monthly Rental
- Annual Subscription
- Token-Based

Web hosted licensing is where entitlements are stored and checked via an internet connection, and so it does not require the use of an on-site dongle to run the software. Note however that it does require a relatively stable internet connection.

One of the added benefits of web hosted licensing is that it offers the ability to use the software on any PC which has an internet connection, and so a user can access their licenses even when they are outside the company office or network.

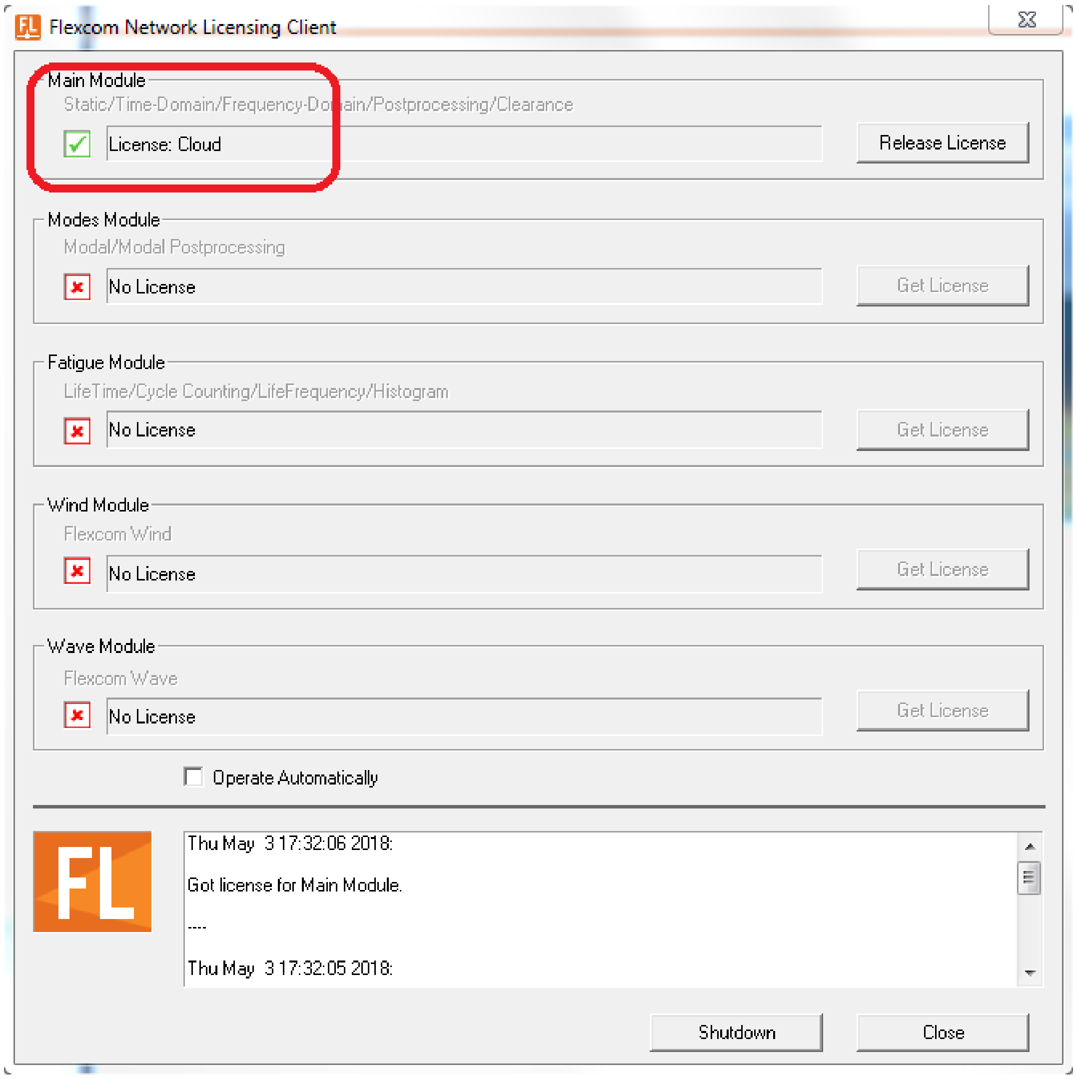
The web hosted license provider for Flexcom is NetLicensing and more information can be found on their website.

Note that cloud-based licensing, which involves the delivery of a software license, plus associated computational power and storage facilities, is a distinct license model for acquiring Flexcom and should not be confused with web-hosted licensing.

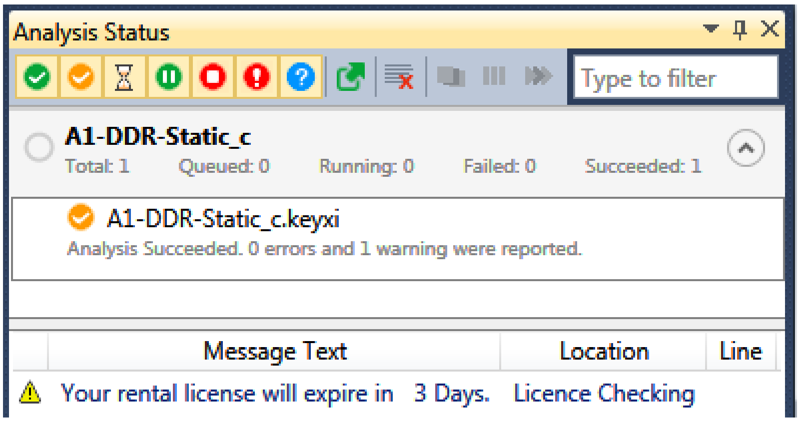
OWNERSHIP, RENTAL & SUBSCRIPTION LICENSES

Under web hosted licensing, these license model provide floating seats. A floating seat is one that can be used anywhere as long as the seat is available and the user has an internet connection. While in use, the seat is not available to other users. A floating seat is held by a user until Flexcom exits.

To elaborate a bit further, a license file is provided to the user which points to the web hosted license server and the Network Licensing Client will get the license from the server as shown below.



A rental model will get the license from the web hosted server in the same manner as the perpetual, and will also issue a warning in the Analysis Status message window when the rental licence has less than 7 days remaining as shown below.



A stable internet connection is recommended for such licenses. However the system is also designed to tolerate some level of intermittent connectivity.

TOKEN-BASED LICENSES

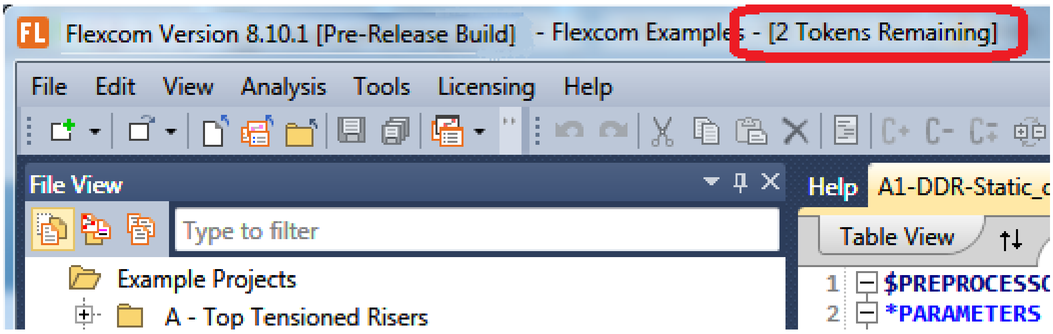
The web hosted licensing system also provides a Pay-Per-Use (token based) license model to facilitate users who do not need a full time rental license and may only need to use Flexcom on an ad-hoc basis. With this model users must purchase a number of tokens upfront and then these tokens are consumed as the user performs simulations in Flexcom. In other words, a certain number of tokens will be consumed on the successful completion of each Flexcom simulation (if a simulation fails for any reason, no token consumption occurs).

The rate of token consumption is dependent on the type of simulation being performed, as shown in the table below.

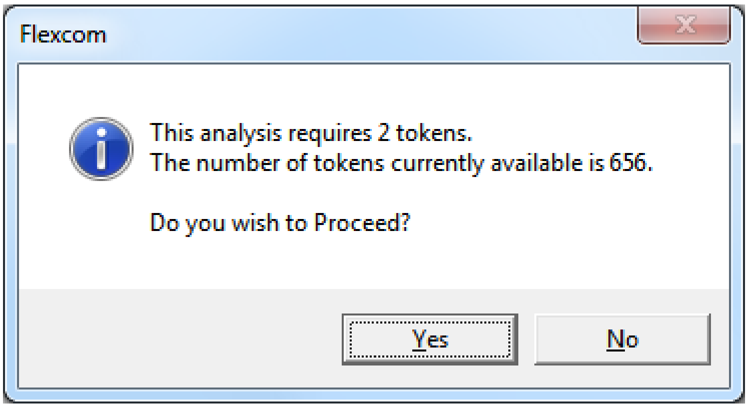
Simulation Type:	**Cost in Tokens**
Static analysis types (static analysis, including quasi-statics)	1
Basic dynamic analysis types (including regular wave time domain analysis, all frequency domain simulations, and modal analysis)	5
Complex dynamic analysis types (i.e. random sea time domain)	20
Inclusion of wind turbine modelling	+1

Post-processing (including database, timetrace, summary, summary collation, clearance & interference, code checking, fatigue analysis and all custom post-processing)	0
Automatic file generation based on template variations	0

The quantity of remaining tokens can be viewed on the title bar of the user interface as shown below.



The token usage and remaining token amount will be reported by the Analysis Status View as shown below.



Important Notes

- Token-based licenses require an active (and stable) internet connection. If the user is not online, it will not be possible to perform any simulations as token consumption cannot take place.

- Users should be aware that large Analysis Jobs can consume a significant amount of tokens. The software issues an information message at the beginning of each job which alters the user regarding the likely level of token consumption for an entire job. If you anticipate running a large number of simulations on a regular basis, then it may be more cost effective to acquire a monthly rental, annual subscription or ownership license.

1.6.4 Downloads

This section provides a range of useful download links associated with Flexcom.

LATEST VERSION

- Latest version This link is always up-to-date with the latest version on general release. This is currently Flexcom 2022.1.1 (August 2022).

Our policy is to provide complete transparency to our Flexcom user community regarding any known software errors or limitations. Refer to Known Software Faults for further information. All of these issues will be rectified in the next maintenance release of the software. In the interim however, workarounds are suggested where feasible.

DEMO VERSION

- Demo version The demo version is ideal for engineers who have a passing interest in Flexcom and would like to learn more about it. A series of online Tutorial Videos are available also.

OLDER VERSIONS

- Flexcom 8.13.3 November 2021
- Flexcom 8.10.4 July 2019
- Flexcom 8.6.4 January 2018
- Flexcom 8.4.4 March 2016
- Flexcom 8.3.7 October 2014
- Flexcom 7.9.7 August 2010

ANCILLARY TOOLS

- Database Viewer Standalone program which offers the same functionality as Flexcom's integrated Model View
- Project Importer Dedicated conversion tool which allow you to migrate your input files from Flexcom 7 to Flexcom 8 format
- RBC Helpful tool for creating rotational boundary conditions in DOFs 4, 5 & 6 (as they do not in the most general case represent individual rotations about the global X, Y & Z axes). Refer to Application of Rotational Constraints for further information.

LICENSING AND SECURITY

- HASP_Info Returns information about your NetHASP hardware dongle, such as dongle ID number and licensed software modules.
- HASP_Info_alt Alternative version of HASP_Info, which is compatible with old parallel port type dongles.
- RUS_Customer Allows you to reprogram your NetHASP hardware dongle under the guidance of Wood personnel (e.g. to add or remove licenses).
- Network License Manager Allows you to install the network license manager software on the server machine which hosts the NetHASP hardware dongle. This is not relevant for web hosted licensing.

RESOLVING INSTALLATION ISSUES

The vast majority of Flexcom installations complete successfully. Occasionally some users have difficulty in getting Flexcom to run successfully after it has been installed. You should contact our technical support support team if you require assistance, who will provide guidance regarding any additional steps which you will need to undertake manually.

- Flexcom 8.10.2 DLLs
- Flexcom 8.10.4 DLLs

1.6.5 Flexcom-on-the-Cloud

OVERVIEW

Flexcom can now be accessed directly from the cloud. This means that the Flexcom software license, plus associated computational power and storage facilities, is delivered as a single coherent package via the internet. This ‘software-as-a-service’ model is becoming increasingly popular, as it allows customers to pay for Flexcom exactly when and where required, helping to lower operational costs and spread out capital investments over time.

ADVANTAGES

Some advantages of the cloud solution are as follows:

- Cost
    - Reduced capital expenditure on expensive hardware
    - Efficient management of software costs via an hourly-based licensing system – pay only for what you use
    - Costs automatically spread out over the duration of a project/year, avoiding the need for a lump sum purchase
- Instant access
    - Run Flexcom whenever and wherever you want, from any computer with internet access
    - No IT issues
    - No local software installation setup required – Flexcom runs within an internet browser
- Cloud storage
    - Persistent cloud storage space to archive your work. There is no limit to how much data you can archive
    - Secure storage area – only you can view your files
- Flexibility
    - Variety of CPU options available (e.g. 8xCPU, 32xCPU, etc.)

    - Simply start a new session to access more (or less) computing power as required

- Scalability

    - Users no longer compete for software licenses or hardware resources, improving overall productivity

- Compatibility

    - Works alongside existing desktop licenses

    - Prepare and fine-tune structural models locally, then execute large batches of computationally intensive simulations on the cloud

FURTHER INFORMATION

- Please contact Wood via sw.support@woodplc.com if you would like to arrange a demonstration of Flexcom-on-the-Cloud via video-link.

- IT personnel may be interested in learning about setting-up the cloud platform to meet their company's requirements.

- Engineers should read the step-by-step guide on using the cloud platform, or simply watch the tutorial video on running Flexcom on the cloud.

- Other frequently asked questions are covered in the Security and Reliability section.

1.6.5.1 Setting-up the Cloud Platform

FLEET SIZE

The first step is for your organisation to inform Wood of the required fleet size. A fleet is defined as a group of identical instances which will be used to serve your company’s requirements. The term ‘instance’ refers to an active user session on the cloud. The number of active instances in a fleet is automatically scaled up or down depending on active demand, so that there is never any unused capacity. Put simply, the maximum number of instances in a fleet determines how many user sessions can be running concurrently.

NOMINATED USERS

Your organisation will need to provide Wood with a list of nominated software users who are authorised to use the cloud platform. Each user is identified by their name and email address. Once Wood have established the cloud platform, all designated users within your organisation will receive an introductory e-mail from ‘AWS AppStream’ (Amazon Web Services) with instructions on how to log on to the cloud portal. A ‘designated user’ in this context is defined in terms of a unique email address, although it is possible for several engineers to use the same log-on credentials.

OFFICE LOCATIONS

It is also important to inform Wood of the geographical locations around the world where your nominated users are located. To ensure optimal performance of the cloud platform and avoid latency issues, the virtual machines will need to be spun-up on servers which are physically located in the same region as the software user.

MACHINE TYPES

The following machines types will be provided as standard on the cloud platform. If you require an alternative configuration, you should discuss this with Wood's technical staff in advance.

- 4 x CPU
- 8 x CPU
- 16 x CPU
- 32 x CPU

DATA STORAGE TYPES

The following types of data storage are available on the cloud platform. It is important that you understand the role and operation of each storage type.

- 'Temporary/Transient' - this is effectively the local hard drive on an active cloud machine. Also known as ephemeral storage, it reverts to its initial state once the user session has finished. In other words, any data stored here is deleted once the user session ends. It has a maximum storage capacity of 150 GB, and a maximum lifespan of 96 hours (equal to the maximum session duration). There is no cost associated with temporary storage, as the costs are built into the hourly rates for computational time on the cloud platform. It is important to note that any data contained in temporary storage must be transferred to another storage area before the active user session is terminated - otherwise it is lost permanently and cannot be recovered. If the user is transferring large amounts of data, sufficient time should be allowed for the file transfer process to fully complete before the active session duration approaches the maximum timeout (96 hours). For efficiency, the use of automated file transfer scripts is recommended to periodically transfer data while a Flexcom batch run is in progress, rather than performing one large file transfer manually following the final completion of the Flexcom batch run.

- 'Permanent/Cold' - this is everlasting storage which is associated with each user account. Any data saved here persists after the user has logged off from the cloud portal. The size of the storage area, and its lifespan, are both unlimited. This is the most economical means of data storage as the storage costs (which are charged per GB per hour) are very low. It is important to note that active Flexcom simulations cannot be hosted within the cold storage area - these must be performed in another storage area with the data transferred into cold storage afterwards. Data stored in permanent storage is private to the user, and cannot be viewed by other users.

- 'Persistent/Hot' - this is an active storage area which also persists once after the user has logged off from the cloud portal. It is effectively a hybrid version which combines the best features from temporary and cold storage into a single offering. The size of the storage area is finite (1TB is a popular option), although its lifespan is unlimited. It is a more expensive means of storage than cold storage (there is a fixed charge per month, which depends on the storage capacity) but it alleviates any concerns regarding potential data loss from temporary storage. It also has the advantage that data stored in hot storage is visible to all nominated users of the cloud platform (i.e. it is effectively a shared drive). The storage capacity should be carefully chosen by IT personnel in consultation with the engineering manager.

Storage Type	Max. Capacity	Max. Lifespan	Cost	Access
Temporary/ Transient	150 GB	96 hours	Free	Single user
Permanent/ Cold	Unlimited	Unlimited	Low (charged per GB per hour)	Single user
Persistent/ Hot	Finite (e.g. 1 TB)	Unlimited	Moderate (fixed amount per month)	Multiple users

Data Storage Types

Although it is optional, we recommend including 1TB of hot storage, based on experience with existing customers. Once you have decided on the amount of hot storage you require, you should inform Wood, who in turn will advise you about the associated monthly cost.

FLEXCOM

You should also inform Wood of the versions of Flexcom which you wish to use on the cloud platform. Strictly speaking, Wood is only obliged to provide technical support on the most recent version, but engineering teams sometimes need to use an older version. For example when a previous engineering project is extended or reactivated, and where the end client is insisting on software version consistency throughout the entire project lifecycle.

Given that web hosted licensing was first introduced in Flexcom 8.10.1, this is the earliest version of Flexcom which is available for use on the cloud platform. The range of available versions will naturally grow over time.

SOFTWARE APPLICATIONS

In addition to Flexcom, the following applications will be made available as standard:

- Windows Explorer

- 7-Zip (free and open-source file archiver)

These applications are essential to the software user for uploading and downloading files and organising online storage. Refer to the step-by-step guide on using the cloud platform for further information on these topics.

You should also inform Wood of any other software applications which you wish to use on the cloud. Responsibility for licensing of these applications resides with your organisation, and Wood may request confirmation of appropriate licenses before adding further applications. Sample applications of interest might include…

- Microsoft Excel
- Notepad++ (free text editing software)
- Adobe Acrobat Reader
- Command Prompt

COSTS

The cloud platform is costed as follows:

- Basic Subscription Fee: in order to cover initial set-up costs, account management, and cloud standby readiness, there is a (small) fixed subscription fee per month.
- Usage Fees: Hourly usage fees are computed based on (i) whether you are running a generic machine or one which includes Flexcom and (ii) the number of processors on that machine. These charges take effect once the machine has been spun-up on the cloud, and then cease whenever the session is terminated by the user. Flexcom machines naturally cost more than generic machines, as the hourly rate includes a license fee for using Flexcom itself, plus a fee for using the computational power provided by Amazon Web Services. Generic machines allow users to perform a variety of non-Flexcom related tasks (e.g. uploading, downloading, zipping, custom pre- and post-processing etc.) at a much lower hourly charge rate.
- Storage Fees: Permanent/Cold storage is charged per GB per hour, and is relatively low cost. Persistent/Hot storage comes at a fixed charge per month which depends on the storage capacity.

For up-to-date information on monetary costs, please contact Wood.

CHECKLIST FOR IT PERSONNEL

Please assemble the following information before contacting Wood:

- ✓ Fleet size - maximum number of user sessions which can be running concurrently
- ✓ Nominated users - list of nominated software users who are authorised to use the cloud platform, along with their email addresses
- ✓ Office locations - list of cities where your nominated software users are physically located
- ✓ Machine types - list of machine types available to the software user to provide different levels of processing power
- ✓ Storage required - amount of hot storage required to support your engineering projects
- ✓ Flexcom - list of Flexcom versions required by the engineering team
- ✓ Software Applications - list of any other software applications which are also required
- ✓ Costs - IT personnel should have a good understanding of all associated costs, and discuss these with the engineering manager

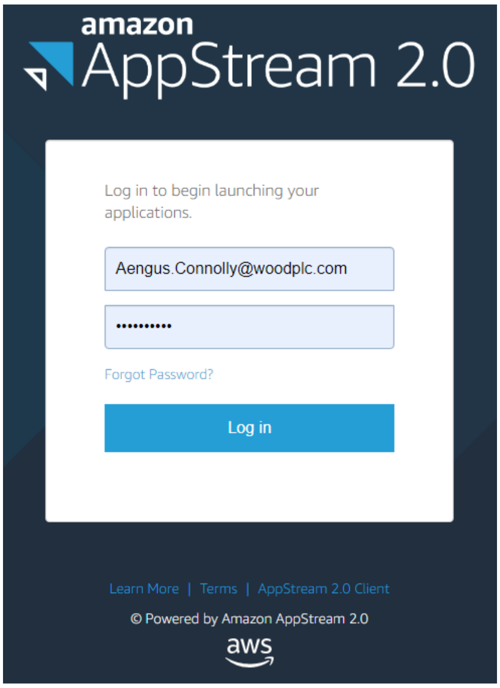
1.6.5.2 Using the Cloud Platform

The following is a step-by-step guide for software users and these instructions should be circulated amongst the local engineering team. Running Flexcom on the cloud is straightforward and requires little or no training. However, there are several aspects which are worthy of further discussion.

INITIATING A SESSION

1. Begin by launching an internet browser. Any of the standard browsers (e.g. Chrome, Firefox, Internet Explorer etc.) may be used, provided it is a relatively recent version which supports HTML5.

2. Log on to the Wood portal using your unique email address and password.



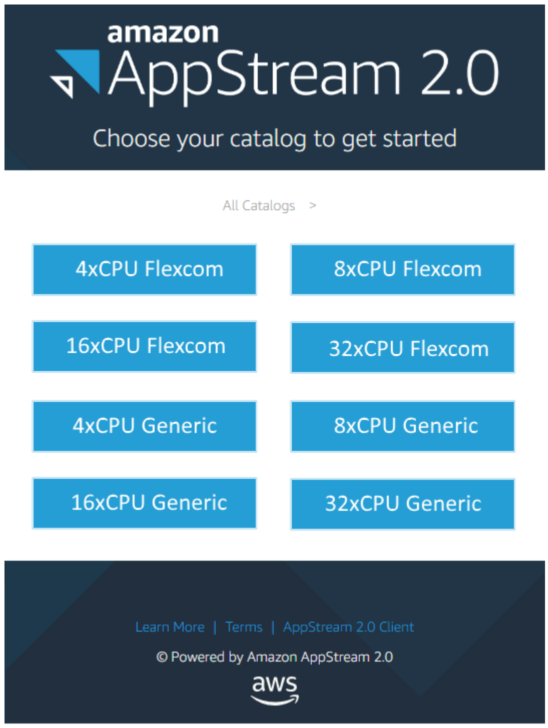
3. Select the machine specification which you require. Please note the following:

   a. Flexcom machines naturally cost more than generic machines, as the hourly rate includes a license fee for using Flexcom itself, plus a fee for using the computational power provided by Amazon Web Services. Generic machines allow you to perform a variety of non-Flexcom related tasks (e.g. uploading, downloading, zipping, custom pre- and post-processing etc.) at a much lower hourly charge rate.

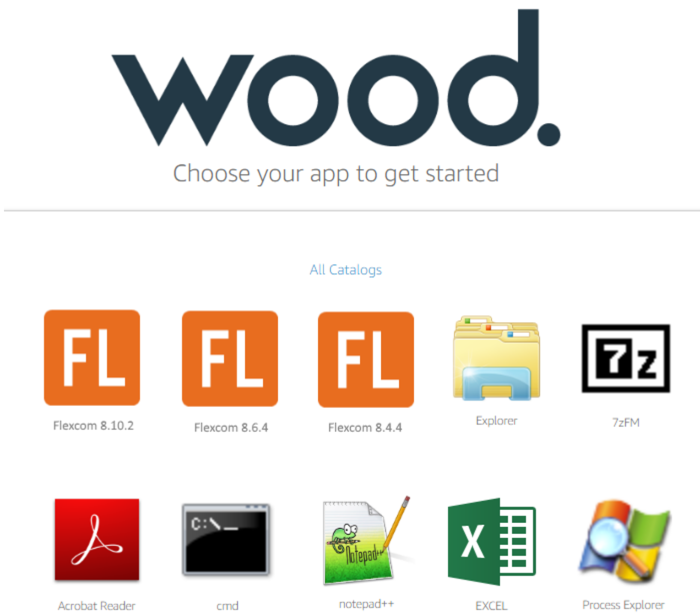
   b. Higher specification machines cost per more per hour than lower specification ones.

   c. Usage charges take effect as soon as the machine has been spun up on the cloud, and persist until the session is terminated.

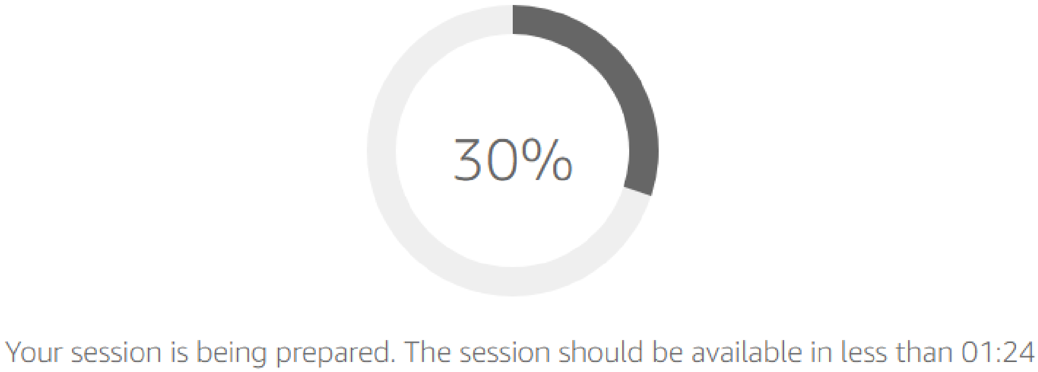
   d. Your organisation is being billed per hour regardless of whether you are using the virtual machine or not. So do not create a machine if you do not intend utilising it.



4. Select the application which you wish to begin using initially. Once the cloud session has become established online, you will be able to launch other applications as well, so it doesn't really matter which application you select at this point.



5. Now you must wait while for your chosen machine is being spun up on the cloud. Boot times are quite quick, normally it takes somewhere between 1 and 2 minutes.



WORKING IN THE CLOUD ENVIRONMENT

Once Flexcom is up and running, simply use it exactly as you would do on your own desktop machine. If you have never used a cloud platform before now, just think of it in terms of logging on remotely to another machine in your company's offices via remote desktop.

Menu Bar

On the top right of your screen, you will see a series of buttons…

From left to right, these buttons represent:

- Launch app – allows you to launch other software applications
- Switch windows – allows you to bring a different window into the foreground, similar to the Alt-TAB key combination on a normal desktop computer
- My files – allows you to upload and download files between your desktop and cloud machines
- Clipboard – allows you to copy and paste data between your desktop and cloud machines
- Settings – allows you to change some general settings (rarely required)

- Enter fullscreen mode – allows you to switch into full screen viewing mode. Press the ESC key to revert to normal window mode

Online Data Storage

The following types of online data storage are available on the cloud platform. It is important that you understand the role and operation of each storage type.

- ‘Temporary/Transient’ - this is effectively the local hard drive on an active cloud machine. Also known as ephemeral storage, it reverts to its initial state once the user session has finished. In other words, any data stored here is deleted once the user session ends. It has a maximum storage capacity of 150 GB, and a maximum lifespan of 96 hours (equal to the maximum session duration). There is no cost associated with temporary storage, as the costs are built into the hourly rates for computational time on the cloud platform. It is important to note that any data contained in temporary storage must be transferred to another storage area before the active user session is terminated - otherwise it is lost permanently and cannot be recovered. If the user is transferring large amounts of data, sufficient time should be allowed for the file transfer process to fully complete before the active session duration approaches the maximum timeout (96 hours). For efficiency, the use of automated file transfer scripts is recommended to periodically transfer data while a Flexcom batch run is in progress, rather than performing one large file transfer manually following the final completion of the Flexcom batch run.

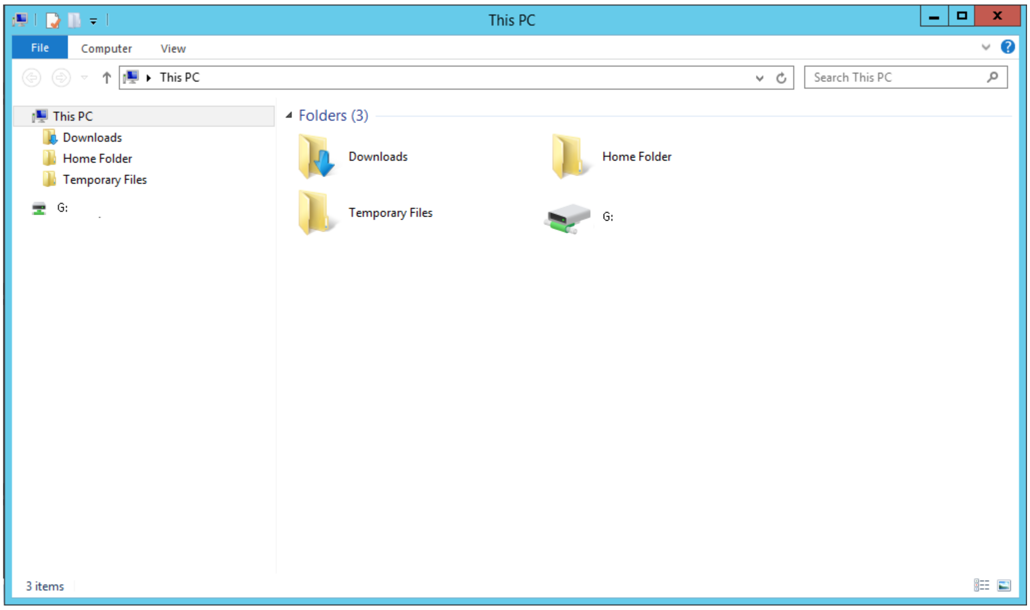
- 'Permanent/Cold' - this is everlasting storage which is associated with each user account. Any data saved here persists after the user has logged off from the cloud portal. The size of the storage area, and its lifespan, are both unlimited. This is the most economical means of data storage as the storage costs (which are charged per GB per hour) are very low. It is important to note that active Flexcom simulations cannot be hosted within the cold storage area - these must be performed in another storage area with the data transferred into cold storage afterwards. Data stored in permanent storage is private to the user, and cannot be viewed by other users.

- 'Persistent/Hot' - this is an active storage area which also persists once after the user has logged off from the cloud portal. It is effectively a hybrid version which combines the best features from temporary and cold storage into a single offering. The size of the storage area is finite (1TB is a popular option), although its lifespan is unlimited. It is a more expensive means of storage than cold storage (there is a fixed charge per month, which depends on the storage capacity) but it alleviates any concerns regarding potential data loss from temporary storage. It also has the advantage that data stored in hot storage is visible to all nominated users of the cloud platform (i.e. it is effectively a shared drive). The storage capacity should be carefully chosen by IT personnel in consultation with the engineering manager.

Storage Type	Max. Capacity	Max. Lifespan	Cost	Access
Temporary/ Transient	150 GB	96 hours	Free	Single user
Permanent/ Cold	Unlimited	Unlimited	Low (charged per GB per hour)	Single user
Persistent/ Hot	Finite (e.g. 1 TB)	Unlimited	Moderate (fixed amount per month)	Multiple users

Data Storage Types

Using the 'Launch app' button, you can open Windows Explorer. This allows you to view the entire contents of your cloud machine, similar to viewing your hard disk on your local machine.



Here you will see high level folders called 'Temporary Files' and 'Home Folder', and drive which is labelled G: or similar. These areas correspond to the storage types described above. Note that you can create as many sub-folders as you wish to organise your data.

- Temporary Files -> Temporary/Transient storage
- Home Folder -> Permanent/Cold storage
- G: drive -> Persistent/Hot storage

Please note the following points:

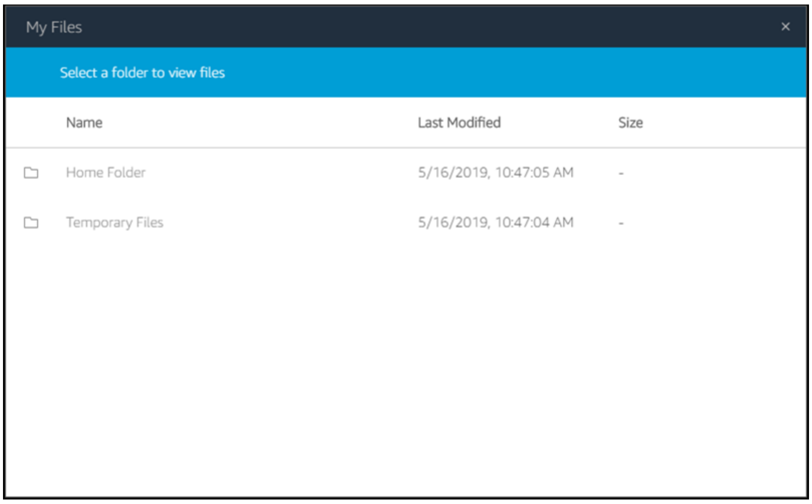
- The maximum capacity of 'Temporary Files' is 150 GB. You should check the maximum capacity of 'G: drive' using Windows Explorer and remain conscious of this limit also.
- If you are working in 'Temporary Files', it is imperative that you save your data to 'Home Folder' or 'G: drive' so that it persists after you have logged out. Alternatively, you may download it to your desktop computer. For efficiency, the use of automated file transfer scripts is recommended to periodically transfer data while a Flexcom batch run is in progress, rather than performing one large file transfer manually following the final completion of the Flexcom batch run.
- The maximum time limit for each user session is 96 hours. If you anticipate that a Flexcom batch run will take more than 96 hours to complete, then you should sub-divide it up into a number of smaller batches.

- The session will automatically terminate after 96 hours, if you have not manually terminated it previously, regardless of what computations are in progress at that point. All data contained in 'Temporary Files' will be automatically deleted at that point.

- Once an engineering project has completed, you should consider an efficient archiving scheme. For example, you could store all Flexcom data files for some time after completion of the project in permanent/cold storage. Beyond that point, you may wish to delete bulky output files and only retain the input files. Flexcom provides a 'Clean Directory' option which could be useful in this context.

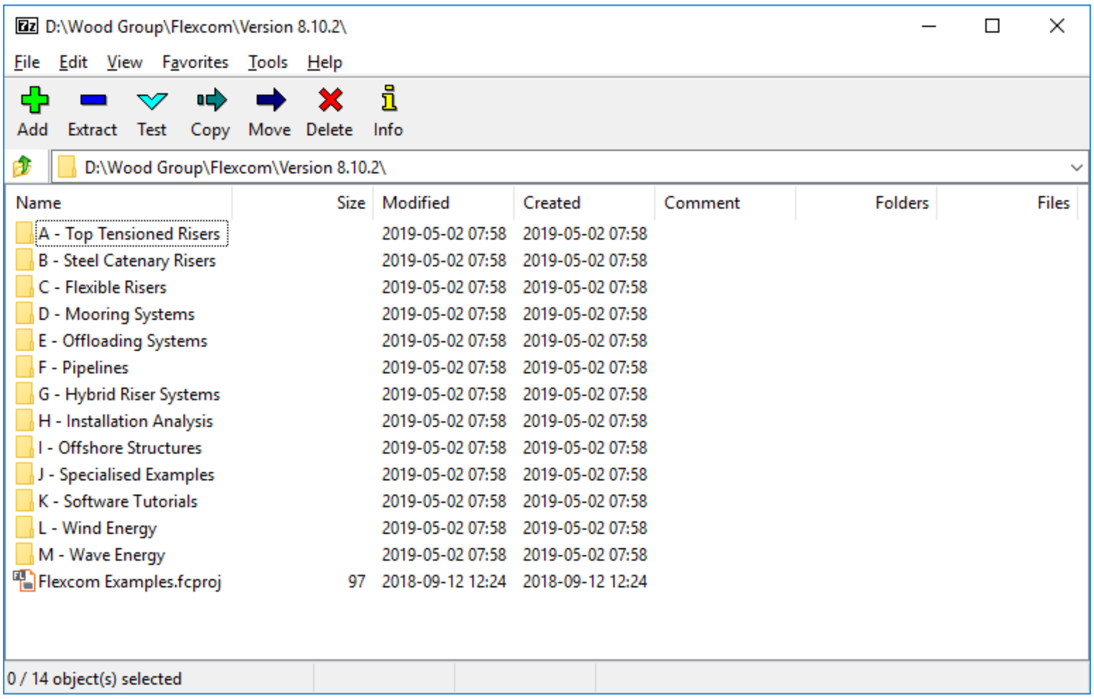
UPLOADING AND DOWNLOADING FILES

File transfer is facilitated by the 'My files' window.



To upload files to your cloud machine, click on 'Temporary Files' to move the focus into that folder (or one of its sub-folders), then simply drag files into the 'My files' window from Windows Explorer on your desktop machine (this is particularly easy if you have 2 monitors attached to your desktop computer). Alternatively you may click the 'Upload Files' button on the top right of the 'My files' window, and this will allow you to browse your local hard disk.

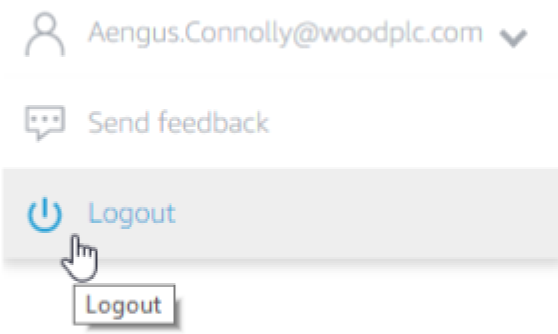
As it is only possible to upload (or download) one file at a time, you should use the 7-Zip file archiver to create a single compressed ZIP file on your desktop machine before uploading to the cloud. 7-Zip is an open source tool which is free to use. If you do not already have it installed on your computer, you may download it from the 7-Zip website.



To download files to your desktop machine, firstly zip the required files/folders into a single ZIP file, then simply click on the relevant file name in the 'My files' window. The file will then be transferred into the 'Downloads' folder on your desktop machine.

CLOSING A SESSION

After you have finished running your Flexcom simulations and have saved your data, you should ensure to log out and terminate the session.



Please note the following:

- It is your responsibility to log out. Your organisation will continue to be billed per hour if you forget to log out.

- Simply closing your internet browser does not constitute logging out.

BACKUPS

Ideally it should happen very rarely, cloud service providers can experience unplanned outages, service disruptions and outright downtime. Naturally this lies outside of Wood's control, it is a good idea to have a contingency plan in place. You should always maintain a copy of your Flexcom input files on your local desktop machine. If the cloud server were to collapse during an active user session and the temporary storage lost completely, you would still be able to reproduce the Flexcom output files in a subsequent cloud session.

1.6.5.3 Security and Reliability

As cloud computing is still relatively new, many prospective users often express concerns over the security and reliability of the cloud platform. This section should help to alleviate some of those concerns, but if you have any further questions, please contact Wood via sw.support@woodplc.com.

DATA SECURITY

- Any data which you upload to the cloud platform is encrypted in transit between your web browser and the cloud machine.
- The data on the cloud machine itself is not encrypted, but when you terminate your session, the machine is destroyed and any data added or created by you on the local drives is destroyed. The storage associated with an active session is known as ephemeral storage, because it reverts to its initial state once the user session has finished. The maximum time limit for each user session is 72 hours, after which point the session will automatically terminate, and all data on the cloud machine is automatically deleted at that point.
- Data stored in a 'Home Folder' is encrypted at rest and persists after you have logged off from the portal. Each user account comes with its individual persistent storage, which is not visible to other users.

INFORMATION AVAILABLE TO Wood

- The systems administrator in Wood has administrative privileges on the cloud platform, and therefore has access to the files which all users have placed in their Home Folders. This is a necessary prerequisite associated with establishing the cloud platform for each customer organisation. However, Wood undertakes never to view or download any customer data under any circumstances.

- Each customer organisation retains the intellectual property rights over their data. Wood fully respects these ownership and confidentiality rights. The situation is similar in some respects to the technical support service provided for desktop software licenses - customers are often required to submit sample input files to allow our technical support team to investigate issues with software models. Data received in this manner is treated confidentially and securely by Wood.

- The cloud service provider actively monitors the machines in your fleet (how many are in use, how many are available etc.). The number of hours spent on active sessions is also continually monitored. This is necessary to facilitate accurate billing.

- Wood's systems administrator can see who is logged on from your organisation, but cannot see what any individual user is doing on their cloud machine.

- Wood undertakes never to perform any form of data analytics whatsoever on customer files.

CONTINGENCY

Ideally it should happen very rarely but cloud service providers can experience unplanned outages, service disruptions and outright downtime. Naturally this lies outside of Wood’s control, but we have contingency plans in place.

- The cloud platform service provided to your organisation is covered by a service level agreement between Wood and the cloud service provider. The cloud service provider targets a monthly up-time percentage of at least 99.9%.

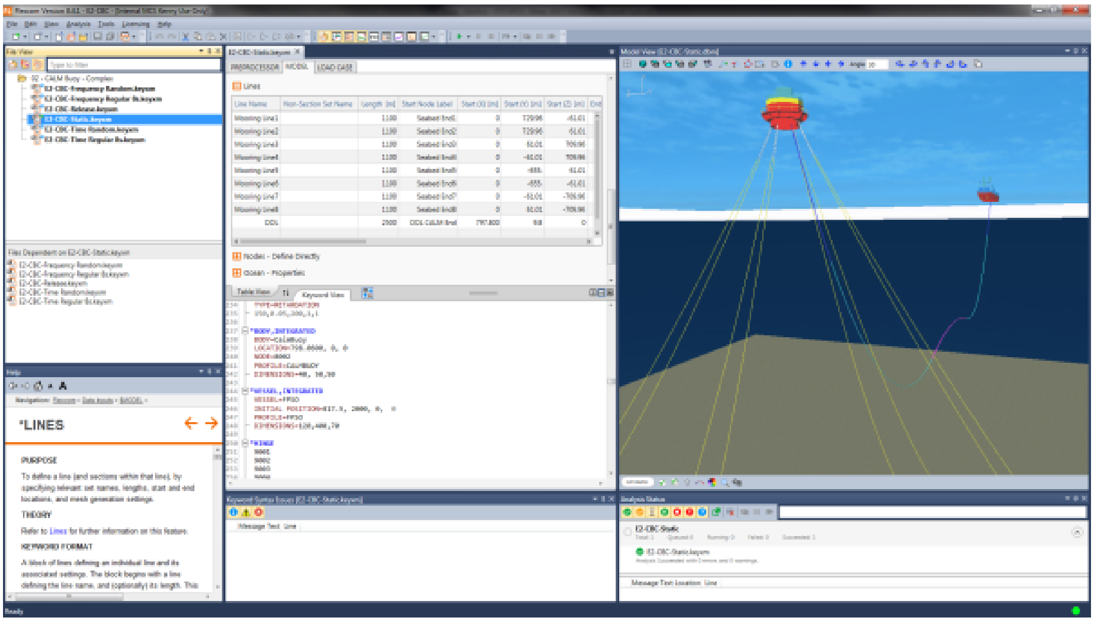
- Regarding the permanent storage, the 'Home Folders' are designed for high durability. The files are redundantly stored on multiple devices across a minimum of three data centres in a geographical region. This service is designed to sustain concurrent device failures by quickly detecting and repairing any lost redundancy, and they also regularly verify the integrity of your data using check-sums. The cloud service provider assumes responsibility for this aspect.

- In the unlikely event of prolonged unavailability of the cloud server, Wood can provide alternative solutions:
    - As a first option, Wood would look to establish a similar capability in a different region of the cloud if possible.
    - Alternatively, replacement desktop licenses may be provided at short notice. For example, if you have 5 authorised users on a cloud platform which offers a 32-CPU machine, Wood can provide a temporary replacement 5-user network license with 32-CPUs enabled. The license entitlements can be authorised via an internet connection, avoiding any time consuming set up of on-site security dongle. Note however that the customer organisation would need to have the necessary hardware available locally to replicate, or partially replicate, the computational power available via the cloud platform. Replacement desktop licenses have a one-week duration after which point they will expire.
- Finally, software users are advised to maintain a copy of their Flexcom input files on a local desktop machine. If the cloud server were to collapse during an active user session and the temporary storage lost completely, the user would still be able to reproduce the Flexcom output files in a subsequent cloud session.

1.7 User Interface

OVERVIEW

Flexcom’s user interface has been created using the latest software development technologies, including .NET Framework and Windows Presentation Foundation (WPF). The .NET Framework is an integral Windows component that supports the development of the most recent generation of applications. It includes a large library of functions and supports several programming languages, which allows developers to combine their own source code with the existing framework in an integrated development environment. WPF is a graphical system produced by Microsoft for rendering user interfaces in Windows-based applications. At its core lies a resolution-independent, vector-based rendering engine that is designed to take advantage of modern graphics hardware. These leading-edge technologies underpin the power and versatility of Flexcom’s next-generation user interface.



User Interface

FURTHER INFORMATION

Further information on the various components within Flexcom's user interface, including details instructions regarding their use, is contained in the following sub-sections.

- Project Workspace
- Model Building
- Running Analyses
- Job Execution Service & Troubleshooting
- Analysis Preview
- Results Examination

Flexcom's main user interface also incorporates two specific modules, Flexcom Wind and Flexcom Wave, which are specifically designed to create models for floating wind turbines and wave energy conversion devices, respectively.

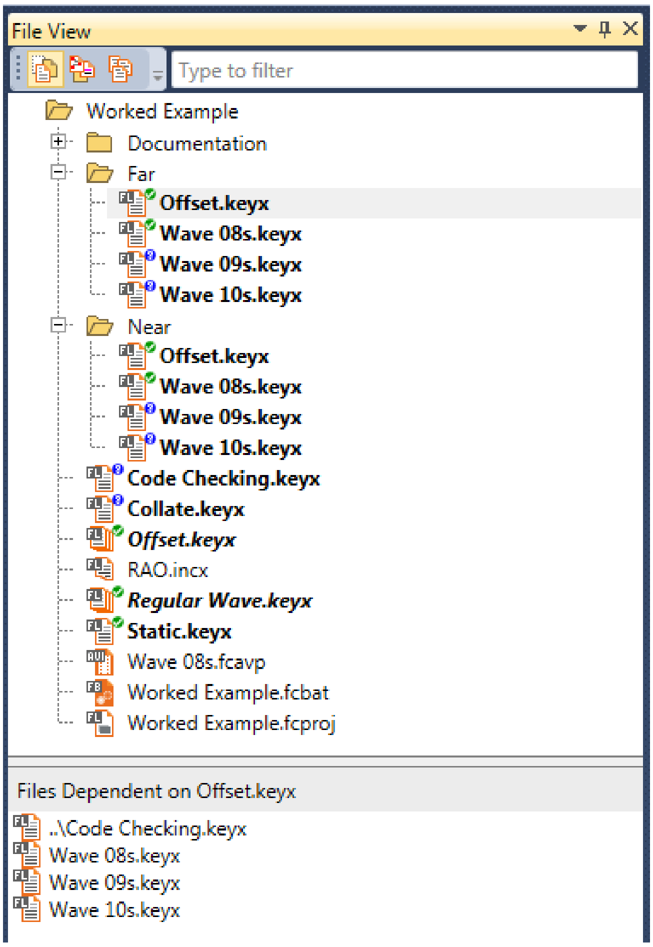
1.7.1 Project Workspace

OVERVIEW

Flexcom 8 is an integrated engineering environment, with all the necessary tools available 'in one box'. This environment is highly project-focused, making it easy to see the interdependencies between keyword files. This contrasts with Flexcom 7 and earlier versions of the software, where an individual keyword file was effectively a standalone entity, and the input file hierarchy in terms of storage, interdependency and so on was completely your own responsibility, with the software offering little in an overall organisational context.

The idea of a Flexcom project is best illustrated by means of a simple example. Consider a sample analysis of a steel catenary riser. A typical load case matrix might consist of (i) an initial static analysis, (ii) near and far offset analyses and (iii) multiple dynamic analyses with varying regular wave periods for both offset cases. Each individual analysis step would traditionally have a corresponding keyword file, representing a standalone Flexcom analysis. In a project-focused environment, each of these individual keyword files is collated into a single project, as illustrated in the figure below, which shows a sample File View from a project.

The top level folder contains the project file Worked Example.fcproj that identifies the folder and all of its sub-folders as a Flexcom project. Project files have the file extension .fcproj (an acronym for FlexCom PROJect). The top level folder also contains the keyword file for the initial static analysis, which represents the starting point for all subsequent restart analyses. The top level folder contains sub-folders for near and far offset conditions. Each sub-folder contains an offset analysis keyword file, which restarts directly from the initial static configuration. Each sub-folder also contains three regular wave dynamic analysis files which restart from the respective offset analyses.



Sample Project Environment

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Keyword File Interdependencies and Status discussed the hierarchy within a project workspace, and the designation of file status via symbolic icons.
- File View describes the panel used to display the files that make up a project, their hierarchy, and the status of the corresponding analyses.

1.7.1.1 Keyword File Interdependencies and Status

The primary advantage of the project structure approach is that it displays the hierarchy within the entire project, so that the keyword file interdependencies are readily evident at a glance. In the sample project environment shown in the Sample Project Environment, the far offset file is currently selected (shown highlighted in blue), and the lower portion of the File View shows a list of the dynamic regular wave analyses which restart from it.

Another powerful feature of the project structure is that it provides you with an indication of the status of all the analyses in the project, which is does via the icons (such as) which are attached to the file names. This allows you to see at a glance which ones have completed successfully, and which (if any) require further attention. There are numerous icons or symbols used for this purpose, and their significance is discussed in more detail in File View. For information though, the significance of the symbols in the Sample Project Environment is as follows:

- The initial static analysis has completed successfully ()
- The far offset analysis completed successfully, but the keyword file for the initial static preceding it has since been modified, so the results are now obsolete and potentially misleading (*)
- The far 8s regular wave analysis failed to complete successfully ()
- The remaining analyses have not yet been run ()

The designation of files as obsolete is particularly helpful in terms of project Quality Assurance.

1.7.1.2 File View

The File View displays a list of all the items included in the project, and functions in a similar manner to Windows Explorer. By default all files are shown, but you can filter the list by choosing to view known files or keyword files only. Known files are ones whose file extension is recognised by Flexcom, and would include keyword (.KEYX extension), output (.OUT extension), plot (.MPLT extension) etc.

It also displays the hierarchy within the entire project, so that the keyword file interdependencies are readily evident at a glance. When a file is selected in the upper section of the File View, the lower section shows a list of the analyses which restart from it.

The File View also provides an indication of the status of all the analyses in the project. This functionality allows you to see at a glance which ones have completed successfully, and which (if any) require further attention. The symbols used are as follows:

- Running. This indicates that the analysis is currently in progress.
- Running, with Warnings. This indicates that the analysis is currently in progress, but has some associated warnings. Inspection of relevant output messages is recommended.
- Paused. This indicates that the analysis is in progress but has been paused.
- Successful. This indicates that the analysis completed successfully, without any warnings or errors.
- Successful, with Warnings. This indicates that the analysis completed successfully, but with some warnings. Inspection of relevant output messages is recommended.
- Failed. This indicates that the analysis failed to complete successfully. Inspection of relevant output messages is essential, with a view to successfully rerunning a modified analysis.
- Terminated. This indicates that the analysis has terminated prematurely, for example if the process was cancelled by some external source such as the Windows Task Manager.
- Licensing Issue. This indicates that you do not have the appropriate license to perform the analysis. Contact the technical support service at sw.support@woodplc.com if required.
- Unknown. This indicates that the analysis status is unknown. Typically this means that the analysis has not yet been run, or perhaps the analysis was stopped by the user before it had reached its completion.

- * Obsolete. This indicates that the analysis has been run previously, but the keyword data of this or preceding analysis has since been modified, and its results are now obsolete and potentially misleading.

- Pending. This indicates that the analysis has been added to a run-queue, via a Branch Run for example, so it is scheduled to commence presently.

1.7.2 Model Building

DATA ENTRY

Two data entry methods are supported by Flexcom. Beginner or inexperienced users may prefer to use the Table Editor - data is entered in tabular format, with the various rows and columns clearly labelled. More experienced users tend to favour the Keyword Editor - working directly with the keyword entries is the quickest way of inputting data.

DATA VALIDATION

Flexcom provides a number of helpful features to ensure the integrity of input data. Keyword Syntax Issues provides a helpful display of any warnings or error messages associated with the input data, while Input Value Graphs allows you to preview and visually inspect any non-linear relationships in the input data. Moreover, the Model View provides a live structure preview during model building. During model building, as you add new components to the model via the Table Editor or the Keyword Editor, the Model View is automatically updated.

FINITE ELEMENT MESH

Nodes and Elements represent the most basic building blocks of a finite element model. Nodes represent the cornerstones of the finite element solution, while elements are used to define the finite element mesh connectivity. However, the majority of model building in Flexcom is performed using Lines, which provide an automatic mesh creation facility to greatly expedite the model creation process. The Line Generation Report presents a summary of the finite element discretisation data for all lines in the model.

SEABED MODELLING

Flexcom can handle complex Seabed Profiles with ease. You simply specify an arbitrary cloud of data points, and Flexcom's triangulation algorithm automatically generates a continuous profile via cubic spline interpolation. The Seabed Utility is a standalone application which allows you to generate compiled seabed bathymetry data files that can later be referenced by an analysis keyword file.

HYDRODYNAMIC DATA IMPORTER

The Hydrodynamic Data Importer is a very helpful utility program which allows you to automatically import characteristic data relating to a Floating Body from a range of well known hydrodynamic simulation packages into Flexcom.

1.7.2.1 Data Entry

OVERVIEW

When you double-click on a particular file in the File View, its contents are shown in the main display area. This area is also used to display plot files and text output files. When a keyword file is being displayed, the area is divided into two sections, with the Table Editor shown in the upper section by default, and the Keyword Editor below it.

Beginner or inexperienced users may prefer to use the Table Editor. Data is entered in tabular format, with the various rows and columns clearly labelled. More experienced users tend to favour the Keyword Editor - working directly with the keyword entries is the quickest way of inputting data.

If you prefer to position the Keyword Editor on top, you can alternate the positions using the  button). Either (or both) sections may be used to define the finite element model and the loading conditions to which it is subjected. The two input sections operate in tandem via Data Synchronisation – as soon as you add or change an entry in the Table Editor, the alteration is immediately reflected in the Keyword Editor, and vice versa. Furthermore, the Model View is continually updated based on the information you specify.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Keyword Editor behaves very much like a standard text editor but has some additional helpful features, including colour coding of entries, syntax prompting and command autocompletion.
- Table Editor facilitates data input in rows and columns in a similar manner to a spreadsheet.
- Data Synchronisation discusses the Automatic and Manual options for data synchronisation between Table Editor and Keyword Editor.

Keyword Editor

OVERVIEW

Existing Flexcom users who preferred to work directly with keyword files pre-Flexcom 8 will find the new Keyword Editor intuitively very useful. It behaves very much like a standard text editor but has lots of additional helpful features, such as:

- Colour Coding
- Data Validation
- Syntax Prompting
- Auto-Completion
- Smart Select
- Quick Find

USER EXPERIENCE

The majority of users will find the Keyword Editor highly responsive, with no time lag between user command and software response. Our vision is to provide a very helpful data editing tool, with speed and responsiveness comparable to a generic text editor. Some reduction in performance is inevitable for very large keyword files, due to the complexity of the predictive features listed above. Here are some helpful tips to help you maximise your experience with the software.

- Use external file references for very bulky keyword data. Coupled analysis simulations for example, require large amounts of input data to define added mass, radiation damping, force RAOs etc. Referencing external files removes clutter from the main keyword deck, and also reduces keyword processing workload thereby further improving user experience. See below for a list of helpful external file references.

- Close any unnecessary data windows if you are not using them. Examine the View Menu or View Toolbar to check which windows are currently active. The Model View is particularly resource heavy, especially for models with a large number of elements (thousands or tens of thousands). This will help to speed things up, as the user interface has to do a lot less processing.

- Try to store files locally. Accessing files stored on a network drive can be very slow, especially for big projects with lots of files.

- Do not create a file with lots of unrecognised lines in one block. The most common reason for such cases is an automatic file generator which does not insert a space prefix for data lines (i.e. keyword data starts in the first column, right at the beginning of each line).

- Avoid using #INCLUDE directives in your keyword file. They can significantly slow down the keyword editor and result in poor user experience. Moreover, this is an obsolete feature which is no longer supported.

HELPFUL EXTERNAL FILE REFERENCES

$MODEL

- *ADDED MASS, FILE= option

- *HYDRODYNAMIC COUPLING, FILE= option

- *P-Y, FILE= option
- *RADIATION DAMPING, FILE= option
- *VESSEL,INTEGRATED, RAO= option

$LOAD CASE

- *CURRENT COEFF, FILE= OPTION
- *FORCE RAO, FILE= option
- *QTF, FILE= OPTION
- *QTF CALIBRATION FB, FILE= OPTION
- *RAO, FILE= OPTION
- *WIND COEFF, FILE= OPTION

Colour Coding

- Section and keyword names appear in blue font (e.g. *OCEAN)
- Mandatory input data appears in black font
- Optional input data appears in grey font
- Comment lines appear in green font (e.g. C... Environment Specification)
- Input tags (i.e. predefined portions of a keyword which immediately precede an equals sign) appear in red font. For example...

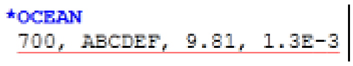
```
*COATINGS
 SET=Outer Pipe
 TYPE=EXTERNAL
 NAME=Insulation, THICKNESS=0.08, DENSITY=900
```

Data Validation

Incorrect input data appears highlighted by a red background. For example...

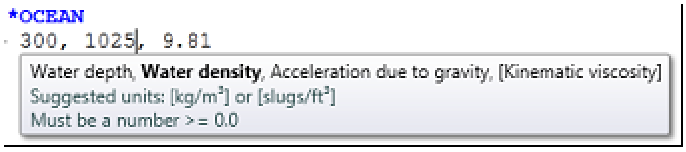
```
*OCEAN
 700, -1025, 9.81, 1.3E-3
```

Unrecognised input data lines are highlighted by a red underline. For example...



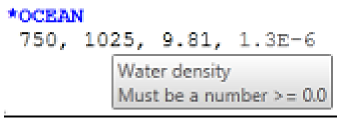
Syntax Prompting

Informative prompts appear while you type input data. For example...



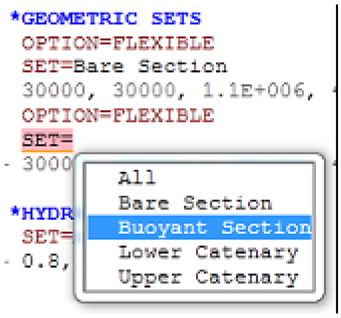
If the prompt disappears (it is not always displayed, so as not to be unnecessarily obtrusive) and you would like to reinstate it, simple press the CTRL+SPACE key combination.

When you hover the mouse cursor over a particular parameter, a descriptive tag appears. For example...



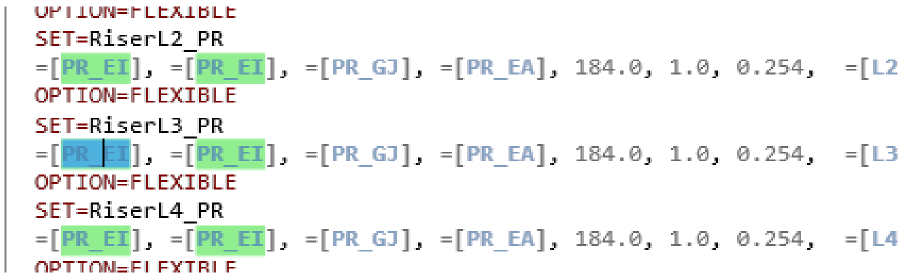
Auto-Completion

As you type, lists of possible entries are provided. You may select from the list using the Arrow and Return keys, or simply pressing the Tab key to quickly select the default value. For example...



Smart Select

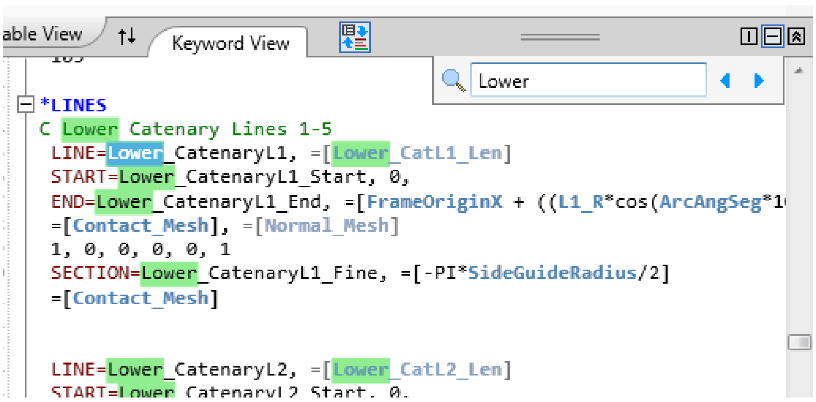
When any text is selected in the Keyword Editor, all other instances of that selected text in the file are also highlighted. This is very useful when inspecting a model to quickly identify where named items are used. For example, simply select an element set name to quickly see all the places in the file where that element set is used. This is also very useful for tracking parameters used in equations.



Smart Select used to see all instances of a non-linear bending stiffness curve

Quick Find

To find text quickly hit CTRL+F to bring up the Find box. Click the left and right arrows to cycle through the matches in the Keyword Editor. Furthermore, all instances of the text you enter is highlighted in the file as you type in the Find box.



Keyword Editor Find

Table Editor

Existing Flexcom users who built models using the pre-Flexcom 8 user interface will find the Table Editor intuitively familiar. The layout and contents of each data entry table closely resembles that of the data entry dialogs in the old user interface. The main difference is that in the old user interface, a data entry dialog for every single input was always present, leading to a very large number of buttons on screen. This made the interface appear quite cluttered. With the new approach, each table is created by request, so the project only contains those parameters which are directly related to your model.

You can add new tables by right clicking anywhere in the Table Editor to invoke the context sensitive menu options and then select Add Feature. The features are grouped together into logical categories, and sub-categories as appropriate. You can add new sections (e.g. Load Case, Database Postprocessing etc.) by right clicking in the header area at the top to invoke the context sensitive menu options and then select Add Section.

The data tables are typically minimised by default for neatness, but you can expand the tables by clicking on the button to facilitate data entry or modification. Similarly, a table may be compressed again by clicking on the button.

Options are provided to allow you to:

- copy a cell (you select a cell entry by double clicking on it)
- copy a row (you select a row by clicking on it)
- copy several rows (you select multiple rows by clicking on the relevant ones while holding down the CTRL key)
- copy the entire table, by selecting all rows

You can use the copy and paste facility to transfer data to and from other Windows programs such as Microsoft Excel. Data may be exchanged with any application which accepts columns of tab-separated text.

Data Synchronisation

Automatic synchronisation of data between Table Editor and Keyword Editor is the default program operation. However, in a small minority of cases, the transfer of data from one medium to the other can be quite time consuming, and you may want to temporarily disable the synchronisation. For example, if you are editing a very large table of vessel RAO data in the Table Editor, it is more efficient to specify all the required data there fully first, before transferring the data to Keyword Editor afterwards. (Note that the use of include files for RAO data is still strongly recommended.) The symbol on the separator between Table Editor and Keyword Editor is available for this purpose. When the symbol shows, Automatic synchronisation is enabled. Clicking on it to change it to disables automatic synchronisation and changes instead to Manual mode. In this mode, you are prompted to manually synchronise views; for example, if you make a change to the keyword file, an option to Click to update Table View ... appears in the Table Editor.

When operating in Manual mode, the Table Editor and Keyword Editor are mutually exclusive entities – as soon as you modify input data in either editor, the other input conduit becomes read only, and will remain in this state until such time as you manually update. Finally, when you switch from Manual to Automatic mode, data is synchronised immediately.

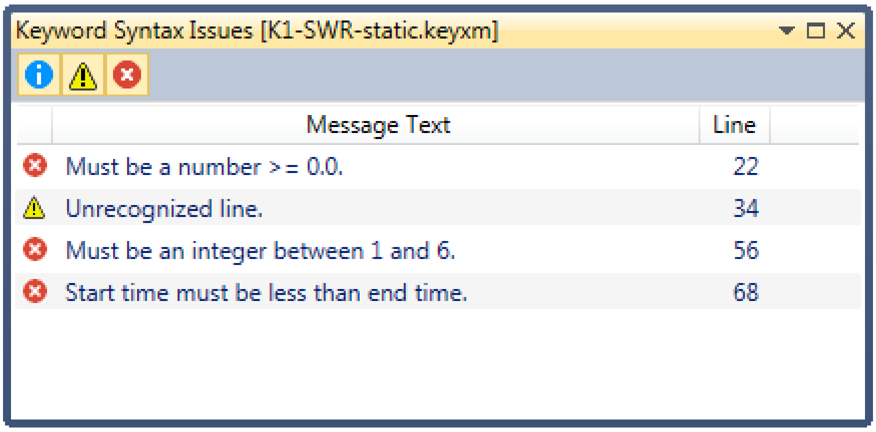
1.7.2.2 Data Validation

Flexcom provides a number of helpful features to ensure the integrity of input data, including...

- Keyword Syntax Issues provides a helpful display of any warnings or error messages associated with the input data.
- Input Value Graphs allows you to preview and visually inspect any non-linear relationships in the input data (e.g. stress-strain curves for non-linear materials, P-y curves for soil modelling).
- The Model View provides a live structure preview during model building. During model building, as you add new components to the model via the Table Editor or the Keyword Editor, the Model View is automatically updated.

Keyword Syntax Issues

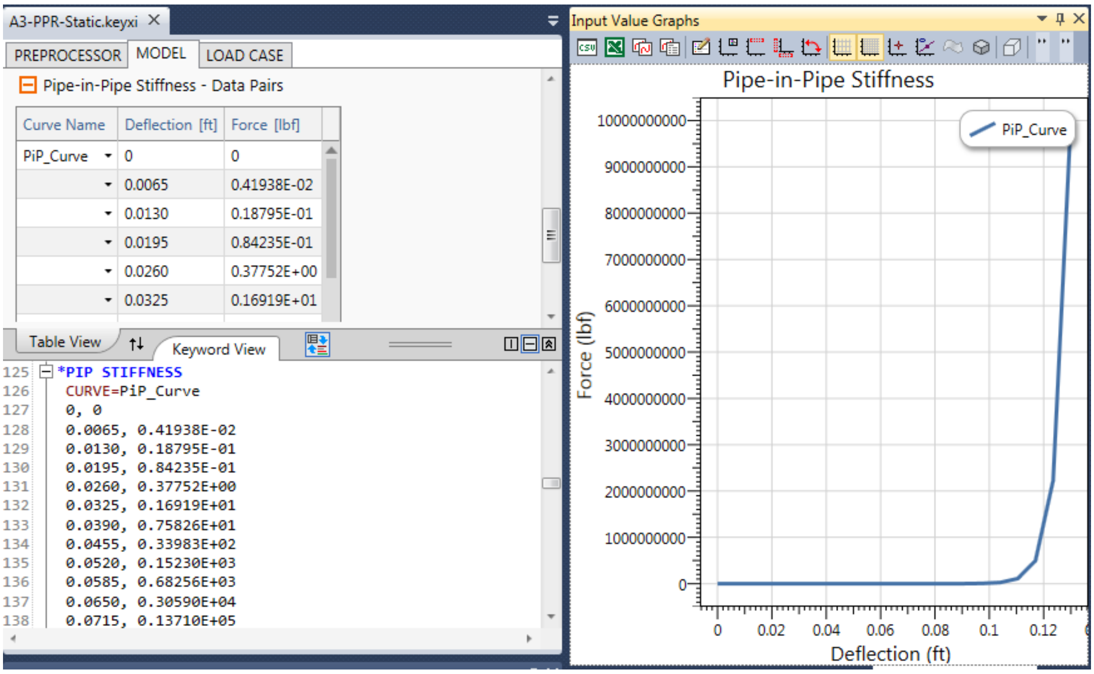
The Keyword Syntax Issues area is used to display any warnings or error messages associated with the input data. Where input constraints are violated (for example, if a negative value is specified for g, the acceleration due to gravity), appropriate error/warning messages are presented in the Keyword Syntax Issues area, and the offending line numbers are presented. For ease of use, double-clicking on a particular message takes you directly to the relevant line in the keyword data. You may then take appropriate corrective action to rectify any inconsistencies in the input data before proceeding to run the analysis. Some sample keyword syntax issues are shown below.



Sample Keyword Syntax Issues

Input Value Graphs

Any non-linear relationships (e.g. stress-strain curves for non-linear materials, P-y curves for soil modelling) in your file may be previewed and visually inspected within the Flexcom user interface. To invoke this feature, press the Input Value Graphs option under the View menu, and then select the parameter of interest from the Plot drop-down list. If you edit the curve values in the Keyword Editor or the Table Editor, the preview plot will also update automatically.



Input Value Graph showing a preview of a pipe-in-pipe stiffness curve

1.7.2.3 Finite Element Mesh

Nodes and Elements represent the most basic building blocks of a finite element model. Nodes represent the cornerstones of the finite element solution, while elements are used to define the finite element mesh connectivity. However, the majority of model building in Flexcom is performed using Lines, which provide an automatic mesh creation facility to greatly expedite the model creation process. The Line Generation Report presents a summary of the finite element discretisation data for all lines in the model.

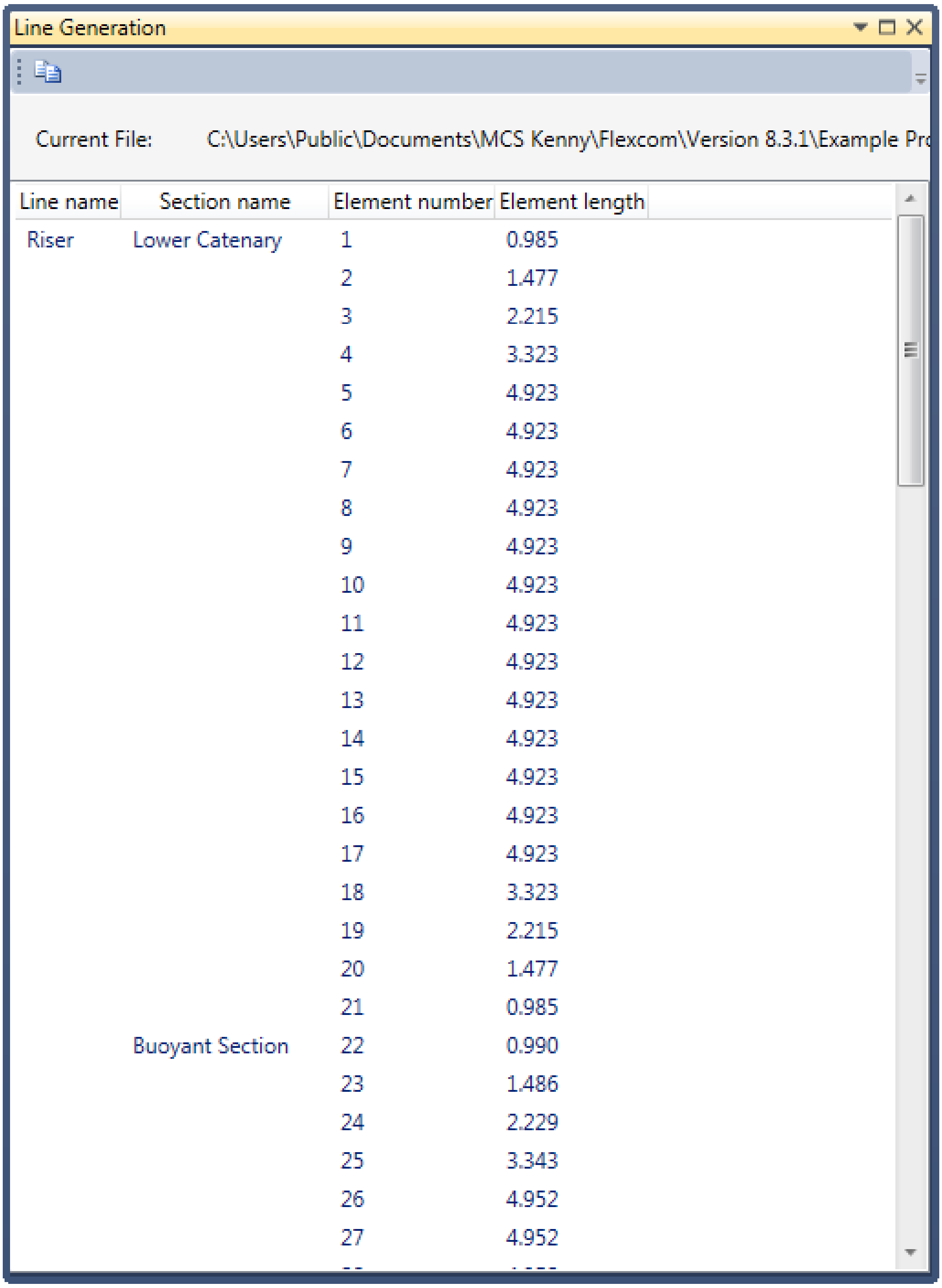
Line Generation Report

The Line Generation Report presents a summary of the finite element discretisation data for all lines in the model. It allows you to visually inspect each line, examine the list of element which it contains, their corresponding lengths, and the line sub-sections to which they belong. This data can be copied and pasted into a spreadsheet and should be quite useful from a user perspective for QA/verification purposes.

The Line Generation Report has the following features:

-  (Copy). The Copy button allows you to copy the contents of the Line Generation table to the clipboard.

- Current File. As you may have several keyword files open simultaneously, the current file lists the name and full path of the keyword file which is currently in focus.
- Line Name. The name of the line, as specified under the *LINES keyword.
- Section Name. The name of the line sub-section, as specified under the *LINES keyword.
- Element Number. The number of the element in the finite element discretisation.
- Element Length. The length of the element, typically in metres or feet, depending on the system of units employed in the model.


Line Generation

Current File: C:\Users\Public\Documents\MCS Kenny\Flexcom\Version 8.3.1\Example Pro

Line name	Section name	Element number	Element length
Riser	Lower Catenary	1	0.985
		2	1.477
		3	2.215
		4	3.323
		5	4.923
		6	4.923
		7	4.923
		8	4.923
		9	4.923
		10	4.923
		11	4.923
		12	4.923
		13	4.923
		14	4.923
		15	4.923
		16	4.923
		17	4.923
		18	3.323
		19	2.215
		20	1.477
		21	0.985
	Buoyant Section	22	0.990
		23	1.486
		24	2.229
		25	3.343
		26	4.952
		27	4.952

Sample Line Generation Report

1.7.2.4 Seabed Utility

The Seabed Utility standalone application allows the user to generate compiled seabed bathymetry data files that can later be referenced by an analysis keyword file. The Seabed Utility can be launched from the Tools menu of Flexcom. Either 2D or 3D seabed bathymetry data files can be generated and are referred to using the `*SEABED PROFILE` keyword. Both seabed geometry types are described in 2D Profile and 3D Profile.

A Flexcom seabed file has the file extension .FCSBD. Compiled seabed files contain a series of 2D or 3D user-specified points as well as rendering and contact information which have been pre-processed for use in Flexcom analyses. Precompiling the seabed data for use in analyses has the following advantages:

1. The compilation process does not have to be performed repeatedly in every analysis, saving analysis time.

2. The Flexcom user interface is able to show a true preview of the interpolated seabed profile that the analysis is actually employing.

The Seabed Utility can can be used to create 2D and 3D seabed profiles as described in the following subsections:

- 2D Seabed File describes how to use the Seabed Utility to create a 2D seabed file.
- 3D Seabed File describes how to use the Seabed Utility to create a 3D seabed file.

Note that the Seabed Utility standalone application is only relevant in the context of Elastic Seabed definitions. Plain text-format inputs are still used to define arbitrary seabed profiles for Rigid Seabed definitions.

2D Seabed File

A new 2D seabed file can be created by selecting the appropriate option from the main menu or the program toolbar.

The screenshot below shows the six areas on the 2D seabed screen:

1. A button to import data points.

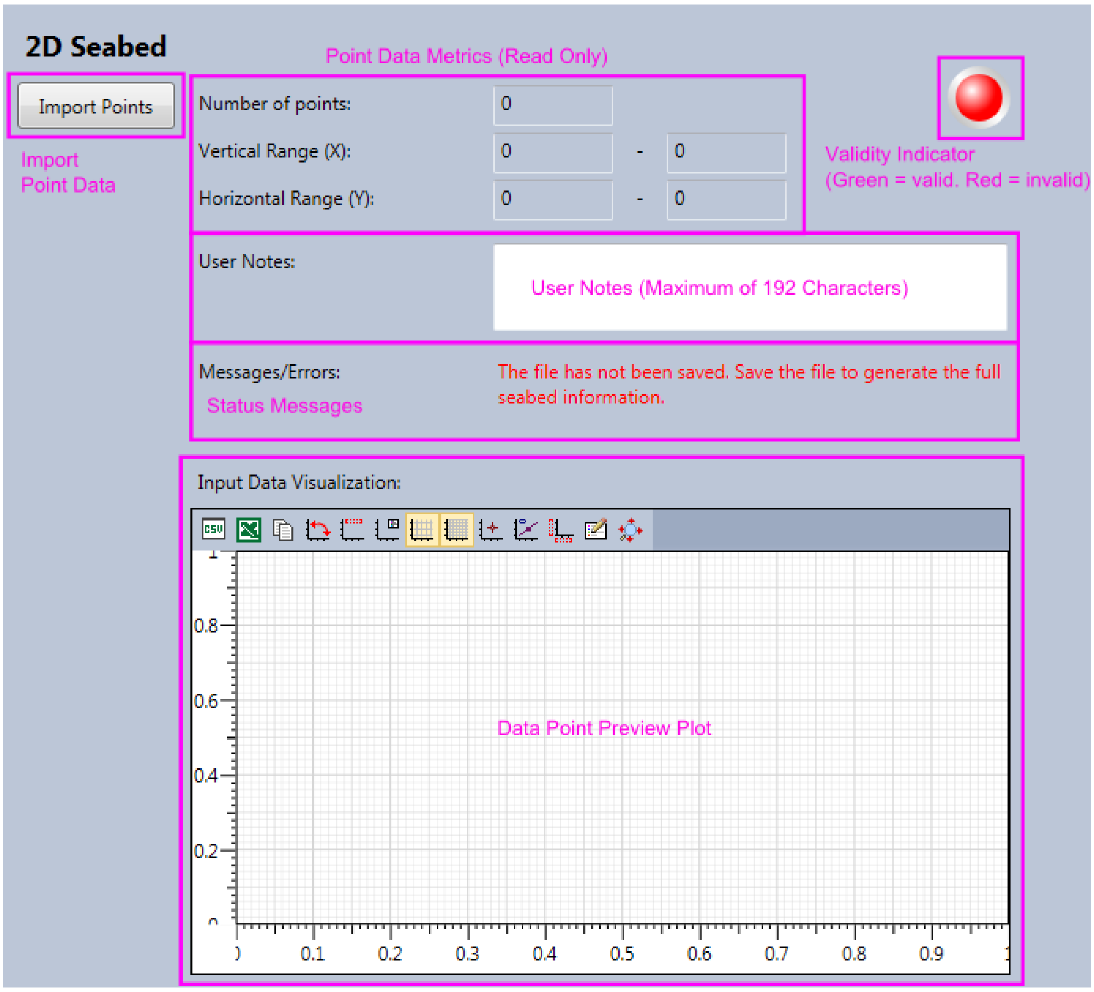
2. Metrics on imported data points; count and ranges.

3. An area to enter and edit user notes.

4. A validity indicator which shows if the seabed file is valid for using in a Flexcom analysis.

5. An area where status messages are displayed.

6. A plot preview of the imported data points.



Creating a 2D seabed requires the input of a series of (X, Y) data points. Pressing the “Import Points” button allows you to select a text-based file and offers flexible import options, very much in the manner of Excel’s Text Import Wizard. The order of the data point components in the text file must conform to the Flexcom global coordinate system where the X axis represents elevation and the Y axis represents horizontal displacement.

For a 2D Seabed, the only other user option which can be set is the User Comments. This block, limited to 192 characters, can be used to record any information you wish about the file.

On the upper-right of the display, the coloured indicator shows if the file is currently valid for use in a Flexcom analysis. For a new file, or a file just edited, this will appear red to indicate the file is not currently valid as it has not yet been compiled. Once the file is saved, provided there is no error in the compilation, the indicator should turn to green. If it is still red, an error message will appear under Messages/Errors. Alternatively, hovering over the indicator will show the current error as a tooltip. Review your input data first and if necessary contact software support for further assistance.

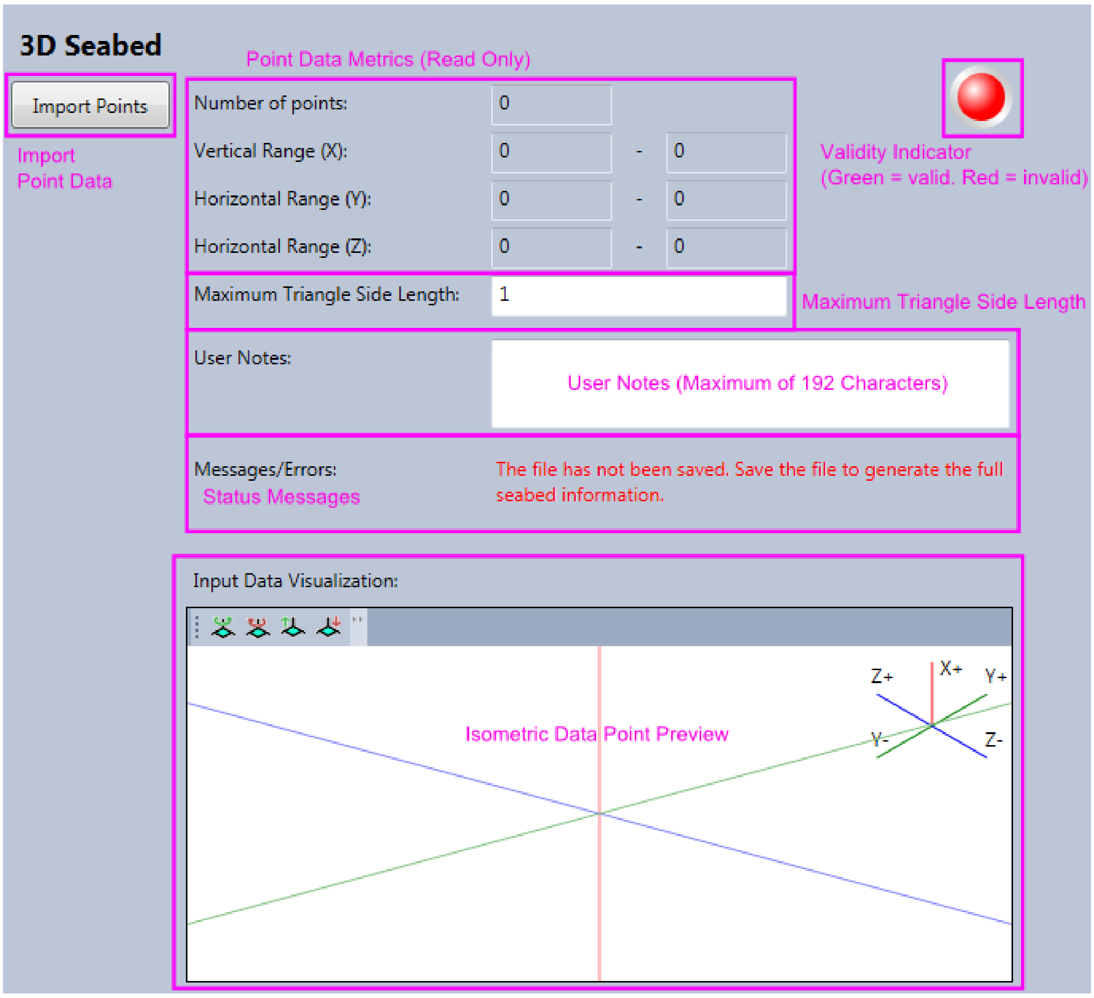
The data point preview plot operates like all other plots in Flexcom. For further information refer to Plotting.

3D Seabed File

A new 3D seabed file can be created by selecting the appropriate option from the main menu or the program toolbar.

The screenshot below shows the seven areas on the 3D seabed screen:

1. A button to import data points.
2. Metrics on imported data points; count and ranges.
3. An entry for the maximum triangle side length to use when discretising the data for visualisation purposes.
4. An area to enter and edit user notes.
5. A validity indicator which shows if the seabed file is valid for using in a Flexcom analysis.
6. An area where status messages are displayed.
7. A 3D isometric preview of the imported data points.



Creating a 3D seabed requires the input of a series of (X, Y, Z) data points. Pressing the "Import Points" button allows you to select a text-based file and offers flexible import options, very much in the manner of Excel's Text Import Wizard. The order of the data point components in the text file must conform to the Flexcom global coordinate system where the X axis represents elevation and the Y axis and Z axis represent orthogonal horizontal displacement.

For the rendering of a nonlinear 3D Seabed, you may wish to change the maximum triangle side length. This value is automatically set, based on the input data, to give a reasonably detailed nonlinear seabed rendering mesh while at same time ensuring that the seabed file size is not excessively large. This value should, ideally, be less than the horizontal distance between the closest pair of input data points to give a realistic 3D rendering of the nonlinear surface. If this value is too large then the rendered nonlinear seabed will appear unrealistic and contact nodes may appear to penetrate or lie above the seabed surface. If this value is too small, the compilation time may be excessively long and the compiled file size extremely large. It is important to note that this input value has an effect on the 3D model display only and has no bearing on the internal analysis seabed contact computations.

You can also enter User Comments. This block is limited to 192 characters and can be used to record any information you wish about the file.

On the upper-right of the display, the coloured indicator shows if the file is currently valid for use in a Flexcom analysis. For a new file, or a file just edited, this will appear red to indicate the file is not currently valid as it has not yet been compiled. Once the file is saved, provided there is no error in the compilation, the indicator should turn to green. If it is still red, an error message will appear under Messages/Errors. Alternatively, hovering over the indicator will show the current error as a tooltip. Review your input data first and if necessary contact software support for further assistance.

The isometric data point preview offers a simple isometric projection of the input data points to aid in checking the input data. Points are slightly depth faded to improve comprehensibility; points closer appear darker and those further away appear lighter. The vertical rotation of the isometric plot can be toggled between the four pre-set viewing angles using the first two toolbar buttons. The isometric tilt can also be varied using the second two toolbar buttons.

1.7.2.5 Hydrodynamic Data Importer

OVERVIEW

The Hydrodynamic Data Importer is a very helpful utility program which accompanies Flexcom which can be launched from the *Tools >> Model Building* menu. It allows you to automatically import characteristic data relating to a Floating Body from a range of well-known hydrodynamic simulation packages. Prior to the simulation of floating bodies in Flexcom, important hydrodynamic data is typically sourced externally from a radiation-diffraction program, and the importer aims to streamline the data transfer process, minimising both user effort and the potential for errors in data specification. Flexcom, like most other programs, has its own set of conventions regarding the specification of input data. The importer understands the output file formats and conventions used in several third-party software packages, and can automatically read in the relevant input data, perform the necessary conversions, and create output files which are readily accessible by Flexcom as standard input files.

Further information is provided in the following sections:

- Using the Importer outlines the basic operation of the converter tool, and discusses its relevant inputs and outputs.
- WAMIT outlines the WAMIT data format and the relevant conversions necessary for Flexcom.
- ANSYS Aqwa outlines the ANSYS Aqwa data format and the relevant conversions necessary for Flexcom.
- NEMOH outlines the NEMOH data format and the relevant conversions necessary for Flexcom.
- OrcaWave outlines the OrcaWave data format and the relevant conversions necessary for Flexcom.

Using the Importer

REQUESTED INPUTS

The data importer accepts a range of inputs as outlined below. Apart from the data file name and source type (i.e WAMIT, Aqwa etc.), the remainder of the inputs are optional. If any of these parameters are explicitly specified via the importer, the corresponding value in the data file is ignored. If any value is not explicitly specified here, the corresponding value is read from the data file. Where a required parameter is not available from either source, the program will issue an error message.

Input:	Description
Input File Name:	The name and location of the input file from which the hydrodynamic data is to be sourced. This should either be the WAMIT formatted *.out file or the AQWA *.lis file or the NEMOH Nemoh.cal file.
Input File Source:	The name of the program which created the input file. The options are WAMIT, AQWA, NEMOH and ORCAWAVE.
Gravity:	The acceleration due to gravity. This entry is only relevant to WAMIT.
Water Density:	The mass per unit volume of the fluid in which the body is situated, typically seawater. This entry is only relevant to WAMIT.
Length Scale:	The length scale which is used to convert the input data into dimensional form. This entry is only relevant to WAMIT.
Excitation Forces:	The source of the wave excitation data, either *Haskind* (the default) or *Diffraction*. This entry is only relevant to WAMIT.
Output Directory:	The path to the folder where Flexcom will route all data files generated during the conversion process. The path is optional and defaults to the current working directory.

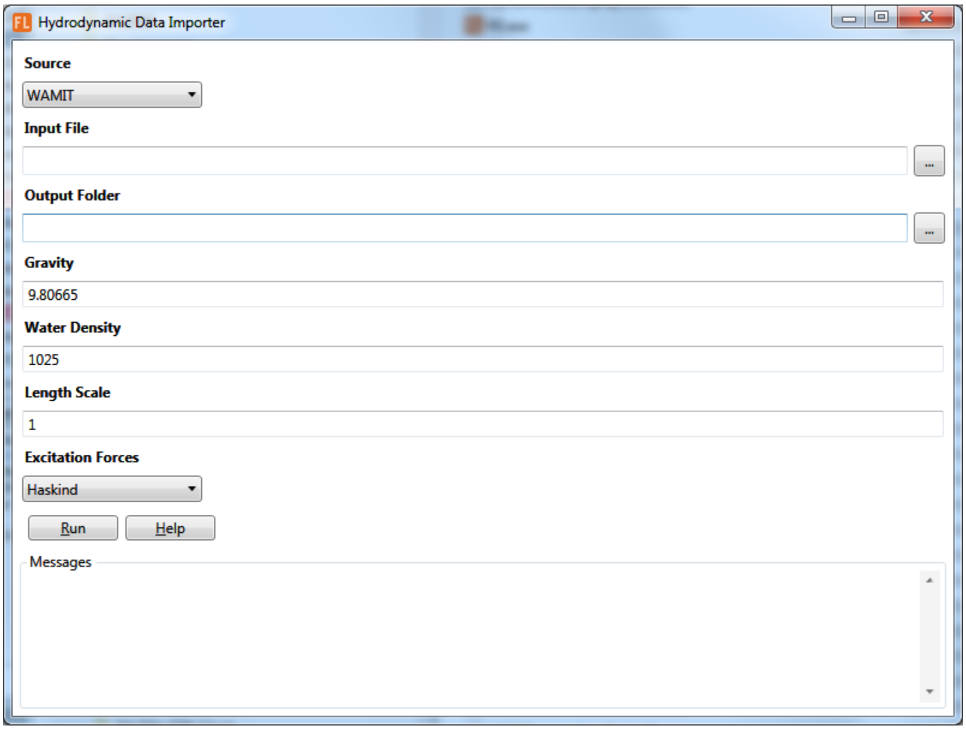
PERFORMING THE DATA IMPORT

User Interface

The user interface is shown below and is straight forward to use. The *source* type of the input data, either WAMIT, AQWA or NEMOH, is selected from a drop-down menu. Paths are provided to the source file and output directory. As mentioned above, the output path is optional and defaults to the current working directory.

The interface for the WAMIT data import differs from AQWA and NEMOH in that it provides inputs for *water density, gravity* and *length scale* which are used for converting the non-dimensional WAMIT data. The inputs default to 1025kg/m^3 for *water density*, 9.80665m/s^2 for *gravity* and 1.0 for *length scale*. Another WAMIT only option on the interface is the source of the wave excitation data, either *Haskind* (the default) or *Diffraction,* which is available to choose from a drop-down menu.

Once you have selected the data file and specified any optional parameters, simply press the *Run* button, The data importer then automatically reads in the relevant input data, performs the necessary conversions, and creates output files which are readily accessible by Flexcom as standard input files.



Hydrodynamic Data Importer Interface

Running from the Command Line

If you wish to by-pass the data importer's user interface, and run the program from a command prompt window or another external program, you simply call the data importer's engine directly, pass it the name and location of the data file which you wish to import, and optionally provide some additional command line arguments. The data importer's engine is called 'HydrodynamicDataImporter.exe' and it is located in your local installation directory. The location varies depending on which version of Flexcom you have installed, but will be typically something like... "C:\Program Files\Wood\Flexcom\Version 2022.1.1\bin\HydrodynamicDataImporter.exe". Note that if the file path contains spaces, you must enclose the character expression in quotation marks.

Mandatory arguments:

- -input *Input File Name and Path*. The name and location of the input file from which the hydrodynamic data is to be sourced. The file path is optional and defaults to the current working directory.
- -source *Input File Source*. The name of the program which created the input file. The options are WAMIT, AQWA or NEMOH (Case insensitive).

Optional arguments:

- -gravity *Acceleration due to Gravity*. This entry is only relevant to WAMIT.
- -wdensity *Water Density.* This entry is only relevant to WAMIT.
- -lscale *Length Scale.* The length scale which is used to convert the input data into dimensional form. This entry is only relevant to WAMIT.
- -excite *Excitation Source*. The source of the wave excitation data, either Haskind (the default) or Diffraction. This entry is only relevant to WAMIT.
- -output *Output Directory*. The path to the folder where Flexcom will route all data files generated during the conversion process. The path is optional and defaults to the current working directory.

GENERATED FILES

The data importer generates a range of data files which may be accessed by Flexcom in the standard fashion.

- Hydrostatic_Stiffness.dat. This file contains Hydrodynamic Stiffness data which is used to model the effects of buoyancy on the floating body. These inputs may be copied into the *FLOATING BODY keyword in your Flexcom model.

- Displacement_RAOs.dat. This file contains the Displacement RAO data which are used in the application of first order (wave frequency) motions. This file is used in conjunction with *VESSEL, INTEGRATED and *RAO keywords in your model.

- Force_RAOs.dat. This file contains Force RAO data from which high frequency First-Order Wave Loads are derived. This file name may be referenced directly from the *FORCE RAO keyword in your Flexcom model.

- Added_Mass.dat. This file contains Added Mass data from which Wave Radiation Loads are derived. This file name may be referenced directly from the *ADDED MASS keyword in your Flexcom model.

- Radiation_Damping.dat. This file contains Radiation Damping data from which Wave Radiation Loads are derived. This file name may be referenced directly from the *RADIATION DAMPING keyword in your Flexcom model.

- Newmans_QTFs.dat. This file contains Quadratic Transfer Function (QTF) data from which low frequency Second-Order Wave Drift Loads are derived. This file name may be referenced directly from the *QTF keyword in your Flexcom model.

WAMIT

This section outlines how the relevant data is obtained from WAMIT and converted into Flexcom. The WAMIT data is generally provided in non-dimensional form (denoted by overscored items in the various equations below).

GENERAL DATA

The following parameters are read from the the header block of the WAMIT file.

- Water Density (ρ)

- Acceleration due to Gravity (g)

- Length Scale (L)

HYDROSTATIC STIFFNESS

The relevant Hydrodynamic Stiffness data is located immediately after the "Hydrostatic and gravitational restoring coefficients:" tag line, as shown in the following example.

```
Hydrostatic and gravitational restoring coefficients:
C(3,3),C(3,4),C(3,5):   1823.9      0.0000      -6032.3
C(4,4),C(4,5),C(4,6):               19920.       0.0000       0.0000
       C(5,5),C(5,6):                           0.13182E+07   0.0000
```

The WAMIT data is converted to Flexcom using the following relations.

$$C_{33} = \overline{C}_{33}\rho g L^2$$

$$C_{34} = \overline{C}_{34}\rho g L^3$$

$$C_{35} = \overline{C}_{35}\rho g L^3$$

$$C_{44} = \overline{C}_{44}\rho g L^4$$

$$C_{45} = \overline{C}_{45}\rho g L^4$$

$$C_{46} = \overline{C}_{46}\rho g L^4$$

$$C_{55} = \overline{C}_{55}\rho g L^4$$

$$C_{56} = \overline{C}_{56}\rho g L^4$$

Symmetry is assumed between the off-diagonal terms of hydrostatic stiffness matrices, so Flexcom assumes that $C_{ij} = C_{ji}$ for all i & j values, with the exception of C_{46} and C_{56} (i.e. C_{64} and C_{65} are always set to zero).

ADDED MASS AND RADIATION DAMPING

The relevant Added Mass and Radiation Damping data is located immediately after the "ADDED-MASS AND DAMPING COEFFICIENTS" tag line, as shown in the following example.

```
ADDED-MASS AND DAMPING COEFFICIENTS
 I     J         A(I,J)         B(I,J)

 1     1   1.855784E+02   1.238282E+02
 1     3  -1.372237E+02  -7.201145E+01
 1     5   2.581553E+04   1.686343E+04
 2     2   1.685065E+03   1.521512E+03
 2     4  -5.329087E+03  -4.116092E+03
```

```
   2     6  -1.659771E+03  -1.476585E+03
   3     1  -1.706160E+02  -3.923212E+01
   3     3   9.427506E+03   4.065720E+03
   3     5  -3.385432E+04  -3.088962E+04
   4     2  -5.548521E+03  -5.356068E+03
   4     4   1.321490E+05   1.908522E+04
   4     6  -2.830587E+04  -2.364873E+04
   5     1   2.614973E+04   1.905146E+04
   5     3  -3.224300E+04  -3.099698E+04
   5     5   6.146746E+06   3.397135E+06
   6     2  -1.698846E+03  -1.461858E+03
   6     4  -2.379884E+04  -1.789628E+04
   6     6   8.810688E+05   6.358418E+05
```

The WAMIT data is converted to Flexcom using the following relations.

$$A_{ij} = \overline{A}_{ij} \rho L^k$$

$$B_{ij} = \overline{B}_{ij} \rho \omega L^k$$

where:

k = 3 for (i,j = 1,2,3)

k = 4 for (i = 1,2,3),(j = 4,5,6) or (i = 4,5,6),(j = 1,2,3)

k = 5 for (i,j = 4,5,6)

DISPLACEMENT RAOS

The relevant data is located after the "RESPONSE AMPLITUDE OPERATORS" tag line as shown in the following example:

```
     RESPONSE AMPLITUDE OPERATORS

   Wave Heading (deg) :      0

      I    Mod[RAO(I)]    Pha[RAO(I)]

      1    5.55722E-01            -90
      2    1.34657E-03            -85
      3    1.51839E+00           -179
      4    5.81387E-05           -100
      5    2.03162E-02             90
      6    1.98620E-01              1
```

The WAMIT data is converted to Flexcom using the following relations.

$$\zeta_i = \overline{\zeta}_i L^n$$

Where:

n = 0 for (i = 1,2,3)

n = 1 for (i = 4,5,6)

FORCE RAOS

The Hydrodynamic Data Importer provides you with an option of selecting either the Haskind or Diffraction excitation forces, and the relevant Force RAO data is located immediately after the "HASKIND EXCITING FORCES AND MOMENTS" or "DIFFRACTION EXCITING FORCES AND MOMENTS" tag lines, as shown in the following example.

```
       HASKIND EXCITING FORCES AND MOMENTS

     Wave Heading (deg) :     180

        I      Mod[Xh(I)]      Pha[Xh(I)]

        1    3.453370E+01            130
        2    5.291137E-06             92
        3    7.375123E+01            -70
        4    2.295947E-05            -24
        5    4.068666E+03            127
        6    6.939258E-05            -88
```

The WAMIT data is converted to Flexcom using the following relation.

$$X_i = \overline{X}_i \rho g L^m$$

where:

m = 2 for (i = 1,2,3)

m = 3 for (i = 4,5,6)

QTF DATA

WAMIT provides QTF data in two different forms. The full QTF data includes QTF coefficients for all possible combinations of wave frequency and wave direction, whereas the abbreviated version contains only the diagonal terms of the full QTF matrix. Given that Second-Order Wave Drift Loads in Flexcom are based on Newman's (1974) approximation only, the relevant QTF data is located immediately after the "SURGE, SWAY & YAW DRIFT FORCES (Momentum Conservation)" tag line, as shown in the following example.

```
 SURGE, SWAY & YAW DRIFT FORCES (Momentum Conservation)

  Wave Heading (deg) :    180       180

     I       Mod[F(I)]    Pha[F(I)]

     1    1.01572E+00         180
     2    1.01946E-07         180
     6    1.11558E-06         180
```

The WAMIT data is converted to Flexcom using the following relation.

$$F_i = \overline{F}_i \rho g L^k$$

where:

k = 1 for (i = 1,2,3)

k = 2 for (i = 4,5,6)

GENERAL CONVERSIONS

- The order of translational RAOs is changed from WAMIT format (Surge, Sway, Heave) to Flexcom format (Heave, Surge, Sway).
- The order of rotational RAOs is changed from WAMIT format (Roll, Pitch, Yaw) to Flexcom format (Yaw, Roll, Pitch).The definition of wave heading is transformed such that a zero degree heading, which is incident on the stern in the WAMIT format, is incident on the bow in the Flexcom format.
- Wave period values in seconds (WAMIT format) are changed to wave frequency values in Hertz (Flexcom format).
- A positive phase angle in WAMIT format is converted from leading to lagging to conform to Flexcom format.

FURTHER INFORMATION

Further information may be obtained from the WAMIT program documentation which is available via the WAMIT website.

ANSYS Aqwa

This section outlines how the relevant data is obtained from ANSYS Aqwa and converted into Flexcom.

GENERAL DATA

General parameters such as water depth, water density and acceleration due to gravity located immediately after the "G L O B A L P A R A M E T E R S" tag line, as shown in the following example.

```
* * * * G L O B A L   P A R A M E T E R S
        - - - - - - - - - - - - - - - - -

WATER  DEPTH  . . . . . . . . . . . . . . .

DENSITY  OF  WATER  . . . . . . . . . . . .

ACCELERATION  DUE  TO GRAVITY . . . . . . .
```

HYDROSTATIC STIFFNESS

The relevant Hydrodynamic Stiffness data is located immediately after the "STIFFNESS MATRIX" tag line, as shown in the following example.

```
                                    STIFFNESS MATRIX
                                    ----------------


          X            Y             Z            RX
      -----------------------------------------------------------

 X    0.0000E+00   0.0000E+00    0.0000E+00   0.0000E+00   0.

 Y    0.0000E+00   0.0000E+00    0.0000E+00   0.0000E+00   0.

 Z    0.0000E+00   0.0000E+00    2.6376E+05   3.0612E-03  -5.

RX    0.0000E+00   0.0000E+00    3.0612E-03   7.7807E+07   1.

RY    0.0000E+00   0.0000E+00   -5.5629E-02   1.2615E-03   7.

RZ    0.0000E+00   0.0000E+00    0.0000E+00   0.0000E+00   0.
```

ADDED MASS AND RADIATION DAMPING

The relevant Added Mass and Radiation Damping data is located immediately after the "ADDED MASS" and "DAMPING" tag lines, as shown in the following example.

```
                    WAVE PERIOD =  20.000   WAVE FREQUENCY =  0.3142

                                                 ADDED  MASS
                                                 -----------

              X            Y             Z             RX
          ---------------------------------------------------------

   X      1.0767E+07    0.0000E+00   -1.3308E+06    0.0000E+00    1.

   Y      0.0000E+00    1.4552E+08    0.0000E+00    3.1150E+08    0.

   Z     -1.1467E+06    0.0000E+00    2.1351E+08    0.0000E+00   -5.

  RX      0.0000E+00    3.0255E+08    0.0000E+00    7.4387E+09    0.

  RY      1.4697E+09    0.0000E+00   -5.6818E+08    0.0000E+00    6.

  RZ      0.0000E+00   -4.1127E+08    0.0000E+00   -9.4037E+09    0.

                                                 DAMPING
                                                 -------

              X            Y             Z             RX
          ---------------------------------------------------------

   X      6.9450E+05    0.0000E+00   -1.8941E+05    0.0000E+00    1.

   Y      0.0000E+00    4.3042E+06    0.0000E+00    3.5709E+06    0.

   Z     -7.4658E+04    0.0000E+00    4.1545E+07    0.0000E+00   -1.

  RX      0.0000E+00    3.3832E+06    0.0000E+00    3.1599E+06    0.

  RY      1.5838E+08    0.0000E+00   -1.5698E+08    0.0000E+00    3.

  RZ      0.0000E+00   -7.8452E+06    0.0000E+00   -2.8377E+07    0.
```

DISPLACEMENT RAOS

The relevant Dicplacement RAO data is located after the "R.A.O.S-VARIATION WITH WAVE PERIOD/FREQUENCY" tag line as shown in the following example.

```
                                           R.A.O.S-VARIATION WITH WAVE
PERIOD/FREQUENCY

                               -----------------------------------------
---
      PERIOD FREQ  DIRECTION          X                 Y
  Z                RX                 RY                RZ
       ------ ----- ---------
------------------------------------------------------------------------
-----------------------------------
        (SECS)(RAD/S)(DEGREES)    AMP      PHASE    AMP      PHASE    AMP
     PHASE    AMP      PHASE    AMP      PHASE    AMP      PHASE

------------------------------------------------------------------------
----------------------------------------------------------

       125.66 0.050   180.00    6.2575  -90.00    0.0000  119.41
1.0047    0.00    0.0000  115.75    0.0945   90.00    0.0146    0.00
        51.63 0.122             2.5620  -90.00    0.0000  150.55
1.0304    0.00    0.0000   -8.63    0.2706   90.00    0.0024   -0.06
        32.49 0.193             1.6099  -90.00    0.0000  141.59
1.0944    0.00    0.0000  -29.45    0.5835   90.00    0.0009   -0.27
        23.70 0.265             1.1956  -90.00    0.0000  127.99
1.2557    0.00    0.0000  -41.72    1.2995   90.00    0.0004   -1.00
        18.66 0.337             1.0624  -89.98    0.0000   80.69
1.8408    0.03    0.0001  -55.94    3.7969   90.02    0.0002   -4.17
```

FORCE RAOS

The relevant Force RAO data is located immediately after the "FROUDE KRYLOV + DIFFRACTION FORCES-VARIATION WITH WAVE PERIOD/FREQUENCY" tag line, as shown in the following example.

```
                               FROUDE KRYLOV + DIFFRACTION FORCES-VARIATION WITH WA
                               ---------------------------------------------------
       PERIOD   FREQ   DIRECTION          X                 Y                 Z
       ------   -----  --------- ---------------------------------------------------
       (SECS)  (RAD/S) (DEGREES)    AMP      PHASE    AMP      PHASE    AMP      PH
       ---------------------------------------------------------------------------

        20.00   0.314       0.00  8.55E+06  -91.49  2.63E+00   27.63  4.58E+07  -11
        17.00   0.370             8.87E+06  -95.13  2.18E+00   -2.05  3.40E+07  -17
        14.00   0.449             6.82E+06 -109.73  2.83E-01   83.66  1.57E+07  -28
        12.00   0.524             3.32E+06 -156.15  3.59E-01  127.59  2.41E+06 -103
        10.00   0.628             5.28E+06  102.43  8.24E-01   60.76  7.39E+06  148
         8.00   0.785             3.46E+06  -30.14  6.69E-01  173.29  3.54E+06  -29
         6.00   1.047             1.34E+06   10.98  2.68E-01  123.57  8.36E+05   97
```

QTF DATA

ANSYS Aqwa provides QTF data in two different forms. The full QTF data includes QTF coefficients for all possible combinations of wave frequency and wave direction, whereas the abbreviated version contains only the diagonal terms of the full QTF matrix. Given that [Second-Order Wave Drift Loads]() in Flexcom are based on [Newman's (1974)]() approximation only, the relevant [QTF]() data is located immediately after the "W A V E - D R I F T L O A D S F O R U N I T W A V E A M P L I T U D E * * 2" tag line, as shown in the following example.

```
FORCES    FREQUENCY    DIRECTION (DEGREES)
------    ---------    -------------------
DUE TO  (RADIANS/SEC)    0.0       20.0       40.0       60.0       80.0       90
------ -------------- -------------------------------------------------------------
DRIFT
-----
ROLL(RX)
            0.314   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            0.370   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            0.449   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            0.524   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            0.628   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            0.785   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            1.047   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            1.257   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            1.571   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            2.094   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
PITCH(RY)
            0.314   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            0.370   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            0.449   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            0.524   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            0.628   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            0.785   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            1.047   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            1.257   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            1.571   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
            2.094   0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E+00  0.00E
YAW(RZ)
            0.314   3.46E-05 -2.35E-03  1.59E-03 -4.15E-04  6.58E-04 -7.53E
            0.370   1.54E-04 -5.74E-05 -2.93E-03 -4.46E-03 -3.50E-04 -6.78E
            0.449  -1.03E-05 -2.96E-03 -3.91E-03  9.73E-04 -1.38E-03  1.43E
            0.524   1.71E-04 -6.45E-03 -3.57E-03 -6.71E-03 -5.78E-03  3.27E
            0.628   2.02E-03 -5.76E-02 -9.29E-02 -7.14E-02 -3.15E-02 -1.85E
            0.785  -2.39E-05  1.92E-02  1.40E-02  7.49E-03  4.90E-03 -4.92E
            1.047   2.80E-04 -1.10E-02 -1.29E-02 -3.02E-02 -8.78E-03  5.78E
            1.257   1.64E-02 -4.61E-02 -4.92E-02 -9.01E-02 -1.85E-02  1.28E
            1.571  -7.84E-03  8.18E-02  1.37E-01  1.30E-01  4.57E-02 -6.59E
            2.094   1.92E-02 -5.31E-01 -3.68E-03  7.27E-01  5.21E-01 -6.05E
```

GENERAL CONVERSIONS

- The order of translational RAOs is changed from ANSYS Aqwa format (Surge, Sway, Heave) to Flexcom format (Heave, Surge, Sway).
- The order of rotational RAOs is changed from ANSYS Aqwa format (Roll, Pitch, Yaw) to Flexcom format (Yaw, Roll, Pitch).
- The definition of wave heading is transformed such that a zero degree heading, which is incident on the stern in the ANSYS Aqwa format, is incident on the bow in the Flexcom format.
- The units of wave frequency are changed from radians/sec (ANSYS Aqwa format) to Hertz (Flexcom format).

FURTHER INFORMATION

Further information may be obtained from the ANSYS Aqwa program documentation which is available via the ANSYS Aqwa website.

NEMOH

This section outlines how the relevant data is obtained from NEMOH and converted into Flexcom.

GENERAL DATA

NEMOH is slightly different from both WAMIT and AQWA in that not all the relevant information is contained in one single formatted output file. This requires the NEMOH output file and folder structure to be intact and as follows:

- **Nemoh Directory**
    - '*Nemoh.cal*' : **Output file**
    - '*mesh*' : **Directory**
        - '*KH.dat*' : **Hydrostatic stiffness data**
    - '*results*' : **Directory**
        - '*ExicationForce.tec*' : **Force RAO data**

□ '*RadiationCoefficients.tec*' : **Added mass and radiation damping coefficients**

The Nemoh.cal file is the main input file to the Hydrodynamic Data Importer. This is parsed first and the 'Description of floating bodies' section is read for the number of degrees of freedom and number of resulting generalised forces.

```
Nemoh.cal
--- Environment -------------------------------------------------------------
1025.0 ! RHO ! KG/M**3 ! Sea water density
9.81 ! G ! M/S**2 ! Gravity
0. ! DEPTH ! M ! Water depth (0 for deep water)
0. 0. ! XEFF YEFF ! M ! Wave measurement point
--- Description of floating bodies ------------------------------------------
1 ! Number of bodies
--- Body 1 ------------------------------------------------------------------
Meshes\Cylinder.dat ! Name of mesh file
500 280 ! Number of points and number of panels
6 ! Number of degrees of freedom
1 1. 0. 0. 0. 0. 0. ! 1 for translation, 2 for rotation followed by direction
1 0. 1. 0. 0. 0. 0. ! Sway
1 0. 0. 1. 0. 0. 0. ! Heave
2 1. 0. 0. 0. 0. -7.5 ! Roll about gravity centre
2 0. 1. 0. 0. 0. -7.5 ! Pitch about gravity centre
2 0. 0. 1. 0. 0. -7.5 ! Pitch about gravity centre
6 ! Number of resulting generalised forces
1 1. 0. 0. 0. 0. 0. ! Force along x axis
1 0. 1. 0. 0. 0. 0. ! Force along y axis
1 0. 0. 1. 0. 0. 0. ! Force along z axis
2 1. 0. 0. 0. 0. -7.5 ! Moment force along x axis about gravity centre
2 0. 1. 0. 0. 0. -7.5 ! Moment force along y axis about gravity centre
2 0. 0. 1. 0. 0. -7.5 ! Moment force along z axis about gravity centre
0 ! Number of lines of additional information
```

HYDROSTATIC STIFFNESS

The relevant Hydrodynamic Stiffness data is located in the '*KH.dat*' file in the 'mesh' directory.

The relevant data is parsed from lines 3, 4 & 5:

```
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
  0.0000000E+00  0.0000000E+00 -0.1093750E+00  0.0000000E+00  0.3255625E+03
  0.0000000E+00  0.0000000E+00  0.0000000E+00 -0.8896512E+08  0.0000000E+00
  0.0000000E+00  0.0000000E+00  0.3255625E+03  0.0000000E+00 -0.8921333E+08
  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00  0.0000000E+00
```

ADDED MASS AND RADIATION DAMPING

The relevant Added Mass and Radiation Damping data is located in the '*RadiationCoefficients.tec*' file in the '*results*' directory. The relevant data is parsed from each line according to the degrees of freedom requested in the Nemoh.cal file.

```
VARIABLES="w (rad/s)"
"A    1    1" "B    1    1"
"A    1    2" "B    1    2"
"A    1    3" "B    1    3"
Zone t="Motion of body    1 in DoF   1",I=    20,F=POINT
  0.3000000E-01  0.6992227E+06  0.5643575E+01 -0.2606135E+03 -0.5823575E-02 -
  0.3442105E+00  0.7568660E+06  0.1010863E+05 -0.3684811E+03 -0.1307687E+02 -
  0.6584210E+00  0.8639779E+06  0.1124563E+06 -0.1008213E+04 -0.4749821E+03 -
  0.9726315E+00  0.7227339E+06  0.6939101E+06  0.2187267E+04 -0.7428232E+03 -
  0.1286842E+01  0.1599056E+05  0.1923467E+06  0.5095169E+02  0.1706791E+04
```

FORCE RAOS

The relevant Force RAO data is located in the '*ExicationForce.tec*' file in the '*results*' directory. The relevant data is parsed from each line according to the degrees of freedom requested in the Nemoh.cal file.

```
VARIABLES="w (rad/s)"
"abs(F    1    1)" "angle(F    1    1)"
"abs(F    1    2)" "angle(F    1    2)"
"abs(F    1    3)" "angle(F    1    3)"
Zone t="Diffraction force - beta =   0.000 deg",I=    20,F=POINT
  0.3000000E-01  0.5227926E+05 -0.1570717E+01  0.4654428E+04  0.3140393E+01
  0.3442105E+00  0.6040670E+06 -0.1559181E+01  0.6800956E+06 -0.3135197E+01
  0.6584210E+00  0.1166256E+07 -0.1502177E+01  0.3596944E+07 -0.2886035E+01
  0.9726315E+00  0.1698730E+07 -0.1172833E+01  0.3424150E+07 -0.1516424E+01
  0.1286842E+01  0.5728897E+06 -0.2830505E+00  0.1075440E+07 -0.1654646E+01
```

GENERAL CONVERSIONS

- The order of translational RAOs is changed from NEMOH format (Surge, Sway, Heave) to Flexcom format (Heave, Surge, Sway).
- The order of rotational RAOs is changed from NEMOH format (Roll, Pitch, Yaw) to Flexcom format (Yaw, Roll, Pitch).
- The definition of wave heading is transformed such that a zero degree heading, which is incident on the stern in the NEMOH format, is incident on the bow in the Flexcom format.

- The units of wave frequency are changed from radians/sec (NEMOH format) to Hertz (Flexcom format).

FURTHER INFORMATION

- Full details regarding NEMOH's modelling capabilities and its underlying theory are available from the NEMOH website.
- Refer to Installing and Running NEMOH if you would like to install and run the software on your own computer.

Installing and Running NEMOH

PREFACE

Before you begin, it is recommended that you open the NEMOH website and the read the articles on 'Running NEMOH' and creating a 'Mesh'. This article provides some useful guidance which supplements that provided on the official NEMOH website. NEMOH is run from a command prompt window as it does not currently have a dedicated user interface. So you may find that it is not very user-friendly, but provided you follow some basic guidelines, you should be able to run the program without much difficulty.

INSTALLATION

NEMOH is an open source software developed by researchers at Ecole Centrale de Nantes which is freely available for download. The product currently does not have a dedicated installer pack, so you simply download the ZIP file, unzip it to extract the program executables, and store these on your local hard drive. In order to make the product easy to find, we suggest that you create a sub-folder in your standard program directory (e.g. 'C:\Program Files\Nemoh v2.03' or similar). If you do not have administrator privileges on your computer, you may wish to choose a generic folder which over you have full access (e.g. your 'desktop' or 'my documents' folders).

Meshmagick is a useful meshing tool which is compatible with NEMOH. It enables mesh visualisation so we recommend that you install it along with NEMOH. We recommend that you install Meshmagick via Python, as this will allow you to use the command line options programmatically via Python scripts if desired. Meshmagick is currently written in Python 2.7 so this needs to be installed before installing Meshmagick itself. Installation instructions are available online, but the main points are summarised here for convenience...

- Navigate to the Anaconda download page using your internet browser.
- Click on Downloads, then click on Windows.
- Under the section 'Python 2.7 version', click on '64-Bit Graphical Installer' to install the 64-bit version (or the 32-bit version if your local operating system happens to be 32-bit).
- Run 'Anaconda2-2019.03-Windows-x86_64.exe'. Select the default options during installation, and also tick the box which allows the installer to add Anaconda to the Windows system path (this allows your computer to recognise Python via the Windows Path, which is useful when running Python scripts from the command prompt).

Next you can proceed to the installation of Meshmagick itself.

- Open an Anaconda command prompt window. You'll find this on the Windows start menu by clicking Windows->Anaconda2 (64-bit)->Anaconda Prompt, or similar. Or simply use the Windows search bar to locate 'Anaconda Prompt'.
- In the Anaconda command prompt, change the version of conda to 4.2, by typing: **conda install conda=4.2**. Type y to accept the changes when prompted.
- Close the current Anaconda command prompt and then reopen a new one.
- Now install Meshmagick by typing: **conda install -c frongere meshmagick**. Type y to accept the changes when prompted.
- You can check if the installation has completed successfully by typing: **meshmagick -h**. This should show the Meshmagick command line help.

WORKING FOLDER

Create a new folder somewhere on your local hard drive to act as your working directory for running your NEMOH simulations. NEMOH requires that the working directory contains 4 files and 2 sub-folders (otherwise it will not run, and the command line errors may not offer much in the way of explanation). So the contents of your working directory must include:

- Mesh (folder)
  - MeshFileName.dat (where 'MeshFileName' is the name of your pre-mesh file)
- Results (folder)

- This folder can be empty, but must be present nonetheless

- ID.dat
- Input.txt
- Mesh.cal
- Nemoh.cal

INPUT FILE FORMATS

ID.dat

This file is simply used to identify the location of the working folder. It consists of a text expression to identify the path to the working folder, with another variable used to state the length of this text expression. The most straightforward approach is to identify the current folder as the working folder. So the file contents would read as follows.

```
1     ! Number of characters in working folder name
.     ! Working folder name
```

Input.txt

This file contains some solution parameters for controlling the operation of NEMOH. The following illustrates some default values as taken from an example provided with NEMOH.

```
--- Calculation parameters -----------------------------------------------
0                        ! Indiq_solver    ! -   ! Solver (0) Direct Gauss (1)
20                       ! IRES            ! -   ! Restart parameter for GMRES
5.E-07                   ! TOL_GMRES       ! -   ! Stopping criterion for GMRE
100                      ! MAXIT           ! -   ! Maximum iterations for GMRE
1                        ! Sav_potential   ! -   ! Save potential for visualiz
```

Mesh.cal

This file controls the meshing program which accompanies NEMOH. A sample file which you may use as a starting point is as follows. The variables are fairly self-explanatory. Note that the z coordinate of the centre of gravity is measured from the mean water line downwards.

```
MeshFileName.dat      ! Name of pre-mesh file
 0                    ! 1 if a symmetry about (xOz) is used. 0 otherwise
 0 0                  ! Possible translation about x axis (first number) and
 0 0 -1               ! Coordinates of gravity centre
 144                  ! Target for the number of panels in refined mesh
 2
 0
 1
 1025                 ! Water density (kg/m3)
```

```
 9.81                 ! Gravity (m/s2)
```

MeshFileName.dat

This is the pre-mesh file which NEMOH's meshing program can refine for you. The format is as follows.

```
Nn                        Number of nodes (integer)
Np                        Number of panels (integer)
x     y     z             List of nodal coordinates (real numbers).
.     .     .
.     .     .
.     .     .
N1    N2    N3    N4      List of panel connectivities (integers).
.     .     .
.     .     .
.     .     .
```

Notes:

- NEMOH's convention is that the z axis is vertical, while the x and y axes are horizontal.
- The z coordinate values are measured from the mean water line downwards (i.e. any nodes below the water line should have a negative z value). It is not necessary to include any part of the body which lies above the water line, but we recommend that you include the full body, as no further mesh creation will be required if you decide to alter the body's draft subsequently.
- N1 to N4 are the indices of the 4 nodes which form each quadrilateral panel. The nodes should be numbered anti-clockwise when the panel is viewed from the fluid domain towards the body.
- You can model triangular panels also, but you must still specify 4 nodes. For example, if your nodes are numbered 1, 2 & 3, then the panel will be defined as 1, 2, 3, 1.

Nemoh.cal

This file controls the operation of NEMOH. A sample file which you may use as a starting point for a single body is as follows. The variables are fairly self-explanatory.

```
--- Environment ------------------------------------------------------------
1025.0                   ! RHO             ! KG/M**3   ! Fluid specific volume
9.81                     ! G               ! M/S**2    ! Gravity
25.                      ! DEPTH           ! M         ! Water depth
0.    0.                 ! XEFF YEFF       ! M         ! Wave measurement poir
--- Description of floating bodies -----------------------------------------
1                        ! Number of bodies
--- Body 1 -----------------------------------------------------------------
```

```
.\Mesh\MeshFileName.dat.dat   ! Name of mesh file
276   72                      ! Number of points and number of panels
6                             ! Number of degrees of freedom
1 1. 0.     0. 0. 0. 0.       ! Surge
1 0. 1.     0. 0. 0. 0.       ! Sway
1 0. 0. 1. 0. 0. 0.           ! Heave
2 1. 0. 0. 0. 0. -0.5         ! Roll about CdG
2 0. 1. 0. 0. 0. -0.5         ! Pitch about CdG
2 0. 0. 1. 0. 0. -0.5         ! Yaw about CdG
6                             ! Number of resulting generalised forces
1 1. 0.     0. 0. 0. 0.       ! Force in x direction
1 0. 1.     0. 0. 0. 0.       ! Force in y direction
1 0. 0. 1. 0. 0. 0.           ! Force in z direction
2 1. 0. 0. 0. 0. -0.5         ! Moment force in x direction about CdG
2 0. 1. 0. 0. 0. -0.5         ! Moment force in y direction about CdG
2 0. 0. 1. 0. 0. -0.5         ! Moment force in z direction about CdG
0                             ! Number of lines of additional information
--- Load cases to be solved -----------------------------------------------
21    0.2   3.0         ! Number of wave frequencies, Min, and Max (rad/s) ->
1     0.    0.          ! Number of wave directions, Min and Max (degrees)
--- Post processing -------------------------------------------------------
1     0.1   10.         ! IRF                          ! IRF calculation (0 fc
0                       ! Show pressure
181.  0.    180.        ! Kochin function              ! Number of directions
0     2     1000. 2.    ! Free surface elevation       ! Number of points in x
```

Notes:

- The name of the mesh file refers to the file which is created by NEMOH's meshing program, not the pre-mesh file which you created.
- The number of points and panels again refers to the final mesh file, which are most likely different from those specified in the pre-mesh file.

RUNNING NEMOH

NEMOH does not have a user interface in the conventional sense and so must be ran from a command prompt window. Begin by taking the following steps.

- Open a command prompt window by selecting it from your Windows start menu. Alternatively use the Windows search bar to search for 'Command Prompt'.
- Navigate to your working folder by using the 'cd' (change directory) command. For example, type something like **cd "C:\My Working Folder"** and press return to execute the command. Note that you must include quotation marks if the path to your working folder contains spaces.

- Make a note of the location of the NEMOH program executables on your local hard drive. This will typically be something like 'C:\Program Files\Nemoh v2.03'.

There are 4 steps involved in running NEMOH.

1. Run the meshing program, 'Mesh.exe'.

   - e.g. type something like **"C:\Program Files\Nemoh v2.03\Mesh.exe"** and press return to execute the command.

   - Assuming the program runs correctly, the 'Mesh' sub-folder of your working folder will now contain the final mesh file. If your pre-mesh file is named 'MeshFileName.dat', then the final mesh file will be named 'MeshFileName.dat.dat'.

   - Note the number of nodes and panels in the final mesh file - you will need to specify these in the 'Nemoh.cal' file before running NEMOH itself.

   - Examine the mesh visually using Meshmagick. To do this you should open an Anaconda prompt window (use the Windows search bar to locate 'Anaconda Prompt'), navigate to the 'Mesh' sub-folder of your working folder, type something like **meshmagick Description_Full.tec --show** and press return to execute the command. Make sure that the mesh looks correct and that the panel normal vectors are pointing outwards from the body into the fluid domain (once Meshmagick is open, you can press 'N' to display the normal vectors).

2. Run the pre-processor, 'preProcessor.exe'.

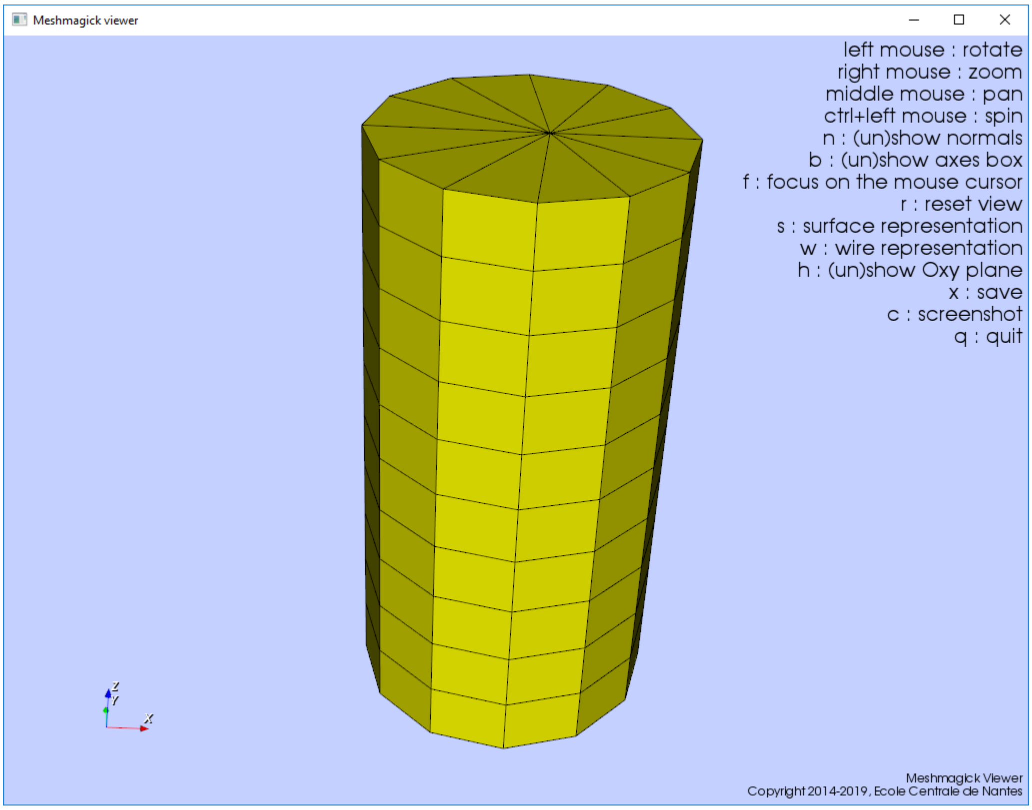
   - e.g. type something like **"C:\Program Files\Nemoh v2.03\preProcessor.exe"** and press return to execute the command.

3. Run the simulation solver, 'Solver.exe'.

   - e.g. type something like **"C:\Program Files\Nemoh v2.03\Solver.exe"** and press return to execute the command.

4. Run the post-processor, 'postProcessor.exe'.

   - e.g. type something like **"C:\Program Files\Nemoh v2.03\postProcessor.exe"** and press return to execute the command.



Sample Mesh Displayed by Meshmagick

HYDRODYNAMIC DATA IMPORTER

Once you are satisfied with the hydrodynamic simulation performed by NEMOH, you should use the Hydrodynamic Data Importer to convert its output data into a format which is readily accessible by Flexcom. The importer produces the following 4 files.

- Hydrostatic_Stiffness.dat. This file contains Hydrodynamic Stiffness data which is used to model the effects of buoyancy on the floating body. These inputs may be copied into the `*FLOATING BODY` keyword in your Flexcom model.
- Force_RAOs.dat. This file contains Force RAO data from which high frequency First-Order Wave Loads are derived. This file name may be referenced directly from the `*FORCE RAO` keyword in your Flexcom model.
- Added_Mass.dat. This file contains Added Mass data from which Wave Radiation Loads are derived. This file name may be referenced directly from the `*ADDED MASS` keyword in your Flexcom model.

- Radiation_Damping.dat. This file contains Radiation Damping data from which Wave Radiation Loads are derived. This file name may be referenced directly from the `*RADIATION DAMPING` keyword in your Flexcom model.

ORCAWAVE

This section outlines how the relevant data is obtained from OrcaWave and converted into Flexcom.

GENERAL DATA

After OrcaWave calculations completes, results are presented in an Excel-compatible (.xlsx or .xls) spreadsheet (read-only) with tabs displaying tables of results. Depending on the model features, results tables can vary but at the moment the Flexcom importable results include:

- Hydrostatics
- Added mass and damping
- Load RAOs (Diffraction and Haskind)
- Displacement RAOs
- Drift loads (i.e. mean drift/Newman QTF loads)

HYDROSTATIC STIFFNESS

The relevant Hydrodynamic Stiffness data is located in the "Hydrostatics" sheet, as shown in the following example.

Hydrostatics

OrcaWave 11.1b: L02 OC4 Semi-sub.owr (modified 9:13 AM on 29-Mar-21 by OrcaWave 11.0b)

Body1			
Volume (m^3)	13371.6769		
Mass (te)	14072.72		
Centre of gravity (m)	-0.0102	0.0	-9.8902
Centre of buoyancy (m)	-0.0029	0.0	-13.2

Water plane area (m^2)	362.9425					
Centre of floatation (m)	-0.0026	0.0				
Water plane moment Lxx (m^4)	140720.018					
Water plane moment Lyy (m^4)	140736.5219					
Water plane moment Lxy (m^4)	0.0					
Hydrostatic stiffness matrix	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	3648.2313	0.0	9.3774	0.0
	0.0	0.0	0.0	1003220.7241	0.0	-1022.1772
	0.0	0.0	9.3774	0.0	1003050.4034	0.0
	0.0	0.0	0.0	0.0	0.0	0.0

OrcaWave units of the 3x3 block (heave/roll/pitch) of a stiffness matrix are:

where F and L denote force and length units, respectively.

ADDED MASS AND RADIATION DAMPING

The relevant Added Mass and Radiation Damping data is located in the "Added Mass" and "Damping" sheets, as shown in the following examples.

Added mass

OrcaWave 11.1b: L02 OC4 Semi-sub.owr (modified 9:13 AM on 29-Mar-21 by OrcaWave 11.0b)

	Added mass for period 4.0s					
	DOF					
	Body1					
	Surge	Sway	Heave	Roll	Pitch	Yaw
	1	2	3	4	5	6
1	4882.704	0.0	0.0296	0.0	-69125.2298	0.0
2	0.0	4882.2406	0.0	69115.9084	0.0	-78.8585

3	0.217	0.0	14237.3282	0.0	55.3233	0.0
4	0.0	69449.6297	0.0	7013994.7597	0.0	-335.0741
5	-69459.8917	0.0	57.4877	0.0	7014791.5568	0.0
6	0.0	-69.8588	0.0	-228.274	0.0	3909576.9848

Orcawave units of the 3x3 blocks of an added mass matrix are:

where M and L denote mass and length units, respectively.

Damping

OrcaWave 11.1b: L02 OC4 Semi-sub.owr (modified 9:13 AM on 29-Mar-21 by OrcaWave 11.0b)

	Damping for period 4.0s					
	DOF					
	Body1					
	Surge	**Sway**	**Heave**	**Roll**	**Pitch**	**Yaw**
	1	**2**	**3**	**4**	**5**	**6**
1	4996.1118	0.0	-0.0355	0.0	-20339.2577	0.0
2	0.0	4998.8144	0.0	20344.3668	0.0	13.8139
3	-0.0293	0.0	41.8528	0.0	0.2819	0.0
4	0.0	20179.2339	0.0	109372.0665	0.0	28.3653
5	-20173.2918	0.0	0.2174	0.0	109362.935	0.0
6	0.0	19.0343	0.0	80.2397	0.0	2713638.3917

Orcawave units of the 3x3 blocks of a damping matrix are:

where F, L and T denote force, length and time units, respectively.

LOADS RAOS

The relevant Load RAO data is located in the "Load RAOs (Haskind)" and "Load RAOs (diffraction)" sheets, as shown in the following examples.

Load RAOs

(Haskind)

OrcaWave 11.1b: L02 OC4 Semi-sub.owr (modified 9:13 AM on 29-Mar-21 by OrcaWave 11.0b)

Wave heading	Period	Surge 1 Ampl.	Surge 1 Phase	Sway 2 Ampl.	Sway 2 Phase	Heave 3 Ampl.	Heave 3 Phase	Roll 4 Ampl.	Roll 4 Phase	Pitch 5 Ampl.	Pitch 5 Phase	Yaw 6 Ampl.	Yaw 6 Phase
deg	s	kN/m	deg	kN/m	deg	kN/m	deg	kN.m/m	deg	kN.m/m	deg	kN.m/m	deg
0.0	4.0	1552.4421	211.3981	0.0	175.5111	130.136	346.9145	0.0	89.1349	5825.3401	351.2653	0.0	310.5234
0.0	5.0	3493.7735	319.2215	0.0	287.7877	1284.7804	100.4475	0.0	258.8618	35449.0062	135.4807	0.0	77.3428
0.0	5.5	4782.5462	304.4005	0.0	276.786	1261.0172	81.6468	0.0	88.3865	47582.7958	117.985	0.0	41.208
0.0	6.0	5560.8616	311.0242	0.0	208.4548	1329.5763	64.8041	0.0	74.9774	36575.3179	129.4346	0.0	209.3405
0.0	6.5	4161.7693	335.306	0.0	185.2142	1246.0818	60.1747	0.0	92.6651	15605.9629	224.5723	0.0	195.6393
0.0	7.0	2686.5091	1.4962	0.0	68.327	1061.2895	70.0728	0.0	254.0211	36614.7318	274.3315	0.0	143.1399
0.0	7.5	2322.7932	31.6488	0.0	92.0851	952.5394	96.9725	0.0	230.9493	54141.9751	276.3595	0.0	199.2684
0.0	8.0	2651.48	53.3302	0.0	245.5759	1075.1032	128.0482	0.0	23.5721	66870.8485	273.6068	0.0	187.5389
0.0	8.5	3139.6208	65.0601	0.0	279.3458	1312.666	148.6555	0.0	29.2393	75620.9705	271.0759	0.0	298.3511
0.0	9.0	3560.5083	71.6825	0.0	161.9588	1504.6342	160.4658	0.0	173.0365	80606.3272	269.4561	0.0	225.7732
0.0	9.5	3857.9931	75.8745	0.0	54.622	1602.2108	167.3956	0.0	12.0696	82295.0763	268.5831	0.0	33.1998
0.0	10.0	4034.0402	78.7938	0.0	22.0804	1612.3891	171.641	0.0	138.6234	81406.7942	268.1934	0.0	276.0653

Load RAOs (diffraction)

OrcaWave 11.1b: L02 OC4 Semi-sub.owr
(modified 9:13 AM on 29-Mar-21 by OrcaWave 11.0b)

		DOF Body1 Surge 1		Sway 2		Heave 3		Roll 4		Pitch 5		Yaw 6	
Wave heading	Period	Ampl.	Phase	Ampl.	Phase	Ampl.	Phase	Ampl.	Phase	Ampl.	Phase	Ampl.	Phase
deg	s	kN/m	deg	kN/m	deg	kN/m	deg	kN.m/m	deg	kN.m/m	deg	kN.m/m	deg
0.0	4.0	1616.8744	214.1596	0.0	16.7086	150.0771	345.112	0.0	22.5955	5672.1165	352.5142	0.0	316.869
0.0	5.0	3524.1749	319.5696	0.0	86.8202	1340.3075	101.4042	0.0	78.27	35983.1986	136.0502	0.0	322.0869
0.0	5.5	4817.7314	304.5872	0.0	262.2988	1299.0218	82.4297	0.0	264.3236	48062.2536	118.0086	0.0	333.9087
0.0	6.0	5597.8071	311.2136	0.0	328.0616	1362.7948	65.2784	0.0	269.0802	36500.1845	129.4093	0.0	334.3401
0.0	6.5	4184.4214	335.5126	0.0	235.9602	1276.9539	60.4538	0.0	166.8476	15695.3181	227.9083	0.0	339.7247
0.0	7.0	2699.859	1.7116	0.0	13.3804	1089.489	70.3063	0.0	265.1193	37706.313	275.4727	0.0	177.8591
0.0	7.5	2336.2079	31.8325	0.0	264.4379	981.068	97.2262	0.0	106.6543	55507.0326	277.0072	0.0	172.8533
0.0	8.0	2668.3127	53.4463	0.0	75.4998	1110.8245	128.2463	0.0	319.0491	68401.9758	274.0474	0.0	282.0246
0.0	8.5	3160.0953	65.1287	0.0	324.0476	1358.5961	148.7891	0.0	339.0898	77268.4413	271.3971	0.0	264.0059
0.0	9.0	3584.0397	71.7235	0.0	254.1191	1559.5003	160.5641	0.0	45.3165	82318.2355	269.702	0.0	168.0255
0.0	9.5	3883.7893	75.8995	0.0	153.202	1663.5002	167.4734	0.0	192.6021	84019.3291	268.78	0.0	303.4827
0.0	10.0	4061.3097	78.8093	0.0	43.6621	1677.8165	171.7044	0.0	345.993	83098.6425	268.3574	0.0	44.3969

DISPLACEMENTS RAOS

The relevant Displacement RAO data is located in the "Displacement RAOs" sheet, as shown in the following example.

Displacement RAOs

OrcaWave 11.1b: L02 OC4 Semi-sub.owr (modified 9:13 AM on 29-Mar-21 by OrcaWave 11.0b)

		DOF Body1 Surge 1		Sway 2		Heave 3		Roll 4		Pitch 5		Yaw 6	
Wave heading	Period	Ampl.	Phase	Ampl.	Phase	Ampl.	Phase	Ampl.	Phase	Ampl.	Phase	Ampl.	Phase
deg	s	m/m	deg	m/m	deg	m/m	deg	rad/m	deg	rad/m	deg	rad/m	deg
0.0	4.0	0.036	42.9234	0.0	13.6117	0.002	167.0425	0.0	206.8585	0.0003	56.0602	0.0	137.6823
0.0	5.0	0.114	148.019	0.0	127.5982	0.0313	281.8211	0.0	71.2249	0.0001	200.4452	0.0	263.2594
0.0	5.5	0.1825	139.7146	0.0	106.3758	0.0376	263.0062	0.0	271.3626	0.0004	181.0228	0.0	244.6457
0.0	6.0	0.2201	145.9374	0.0	35.6026	0.0478	246.2256	0.0	252.7589	0.0013	147.2827	0.0	69.6527
0.0	6.5	0.197	158.3398	0.0	14.4375	0.0535	241.4094	0.0	225.6261	0.0024	138.2799	0.0	19.4367
0.0	7.0	0.1623	173.9745	0.0	259.7283	0.054	250.8968	0.0	80.194	0.0035	130.2097	0.0	332.225
0.0	7.5	0.1489	197.2771	0.0	264.9008	0.0575	277.4601	0.0	62.8947	0.0044	122.6131	0.0	24.11
0.0	8.0	0.1683	221.8133	0.0	51.1194	0.0767	308.5903	0.0	204.1939	0.0052	116.4683	0.0	8.4917
0.0	8.5	0.2123	239.0602	0.0	94.4823	0.1096	329.5841	0.0	249.5526	0.0058	111.8355	0.0	108.6892
0.0	9.0	0.2665	249.4851	0.0	331.7835	0.1457	341.7805	0.0	0.2374	0.0062	108.3735	0.0	47.1169
0.0	9.5	0.3222	255.8386	0.0	267.6795	0.1789	348.9019	0.0	181.2306	0.0064	105.7172	0.0	213.4493

0.0	10.0	0.3755	259.8887	0.0	171.8863	0.2072	353.1391	0.0	331.8753	0.0065	103.5971	0.0	96.2774

QTF DATA

Given that Second-Order Wave Drift Loads in Flexcom are based on Newman's (1974) approximation only, the relevant QTF data is located in the "Mean drift loads (PI)" sheet, as shown in the following example.

Mean drift loads (PI)

OrcaWave 11.1b: L02 OC4 Semi-sub.owr (modified 9:13 AM on 29-Mar-21 by OrcaWave 11.0b)

Wave heading		Period	DOF Body 1 Surge		Sway		Heave		Roll		Pitch		Yaw	
			1		2		3		4		5		6	
(1st order wave 1)	(1st order wave 2)		Ampl.	Phase	Ampl.	Phase	Ampl.	Phase	Ampl.	Phase	Ampl.	Phase	Ampl.	Phase
deg	deg	s	kN/m^2	deg	kN/m^2	deg	kN/m^2	deg	kN.m/m^2	deg	kN.m/m^2	deg	kN.m/m^2	deg
0.0	0.0	4.0	60.1882	0.0	0.0	180.0	2.0855	0.0	0.0	180.0	202.2284	0.0	0.0	180.0
0.0	0.0	5.0	80.2867	0.0	0.0	0.0	8.3322	0.0	0.0	180.0	340.0095	0.0	0.0	180.0
0.0	0.0	5.5	60.2783	0.0	0.0	0.0	9.4414	0.0	0.0	180.0	212.5962	0.0	0.0	0.0
0.0	0.0	6.0	25.7206	0.0	0.0	0.0	18.2964	0.0	0.0	0.0	136.7775	0.0	0.0	180.0
0.0	0.0	6.5	32.8004	0.0	0.0	0.0	28.2411	0.0	0.0	0.0	160.82	0.0	0.0	0.0
0.0	0.0	7.0	45.3586	0.0	0.0	0.0	35.3925	0.0	0.0	180.0	278.1227	0.0	0.0	0.0
0.0	0.0	7.5	41.3604	0.0	0.0	0.0	39.2664	0.0	0.0	0.0	348.6755	0.0	0.0	180.0
0.0	0.0	8.0	28.0587	0.0	0.0	0.0	38.8332	0.0	0.0	180.0	328.4822	0.0	0.0	180.0

0.0	0.0	8.5	15.218	0.0	0.0	180.0	34.8606	0.0	0.0	180.0	250.3351	0.0	0.0	180.0
0.0	0.0	9.0	7.0199	0.0	0.0	0.0	29.5961	0.0	0.0	180.0	165.8212	0.0	0.0	180.0
0.0	0.0	9.5	2.8697	0.0	0.0	0.0	24.8258	0.0	0.0	0.0	100.6511	0.0	0.0	180.0
0.0	0.0	10.0	1.0288	0.0	0.0	0.0	21.1913	0.0	0.0	0.0	57.0976	0.0	0.0	0.0

GENERAL CONVERSIONS

- The order of translational RAOs is changed from OrcaWave format (Surge, Sway, Heave) to Flexcom format (Heave, Surge, Sway).
- The order of rotational RAOs is changed from OrcaWave format (Roll, Pitch, Yaw) to Flexcom format (Yaw, Roll, Pitch).
- Loads RAO units are converted (typically from kN/m) to N/m (forces) and (typically from kNm/m) to Nm/m (moments).
- Loads QTF units are converted (typically from kN/m^2) to N/m^2 (forces) and (typically from kNm/m^2) to Nm/m^2 (moments).
- Displacements RAO units (rotational degrees of freedom) are converted from rad/m to deg/m.
- Phase angles are converted (typically from radians) to degrees.
- The definition of wave heading is transformed such that a zero degree heading, which is incident on the stern in the OrcaWave format, is incident on the bow in the Flexcom format.
- The phase angle convention in OrcaWave is that positive phase angles are "leading". The OrcaWave phase angles are converted (multiplying by -1) to conform to Flexcom format (i.e. converted from leading to lagging).
- Wave period/frequency values, if not already in Hertz, are converted (from seconds or radians/second) to wave frequency values in Hertz.

FURTHER INFORMATION

Further information may be obtained from the OrcaWave program documentation which is available via the OrcaWave website.

1.7.2.6 Wind Field Generator

OVERVIEW

The Wind Field Generator app allows you to create wind data files which characterise the wind field as a function of space and time. It acts a user-friendly interface to the TurbSim software which does not have a Window-based GUI of its own. It allows you to run batches of TurbSim wind field simulations to generate all the wind data files required to support your design load cases. Several of Flexcom's wind turbine examples, such as Example L04 - UMaine VolturnUS-S IEA15MW, utilise TurbSim binary wind-field definitions files. With the file extension BTS (denoting Binary TurbSim), these files tend to be very large, so they are not supplied with Flexcom as it is not practical to include them in the installation package. Instead, you can readily generate the BTS files yourself with the Wind Field Generator app. For further information on TurbSim itself, refer to TurbSim Overview.

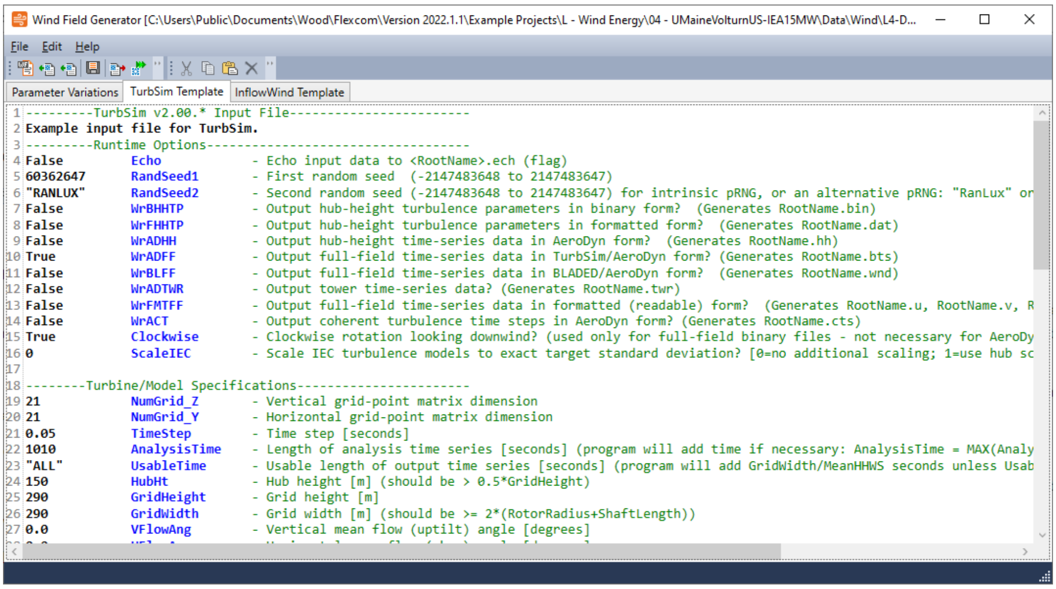
GENERATING WIND FILES

TurbSim V2.6.0 (May 2021) comes pre-installed with the Flexcom installation.

Follow these steps to generate your TurbSim wind data files:

1. Launch the tool via *Tools->Model Building->Wind Field Generator*.

2. Import a sample TurbSim input file via *File->Import TurbSim Template*. Likewise import a sample InflowWind input file via *File->Import InflowWind Template*. These files will act as a base templates for the generated IN and DAT files. Example L04 contains sample files if you do not already have any of your own. These may be found in a location such as 'C:\Users\Public\Documents\Wood\Flexcom\Version 2022.1.1\Example Projects\L - Wind Energy\04 - UMaineVolturnUS-IEA15MW\Data\Wind'. The contents of the template files are displayed on the TurbSim Template and InflowWind Template tabs which neatly colour the inputs for ease of viewing. You may also make edits to these template files here if you wish.



Sample TurbSim Template

3. Define the required wind load cases on the *Parameter Variations* tab. The first line is automatically populated from the template file. The table supports copy and paste, so many users will prefer to set up their wind definitions in Excel, and then copy and paste the data into the wind field generator app. However this window is very quick to navigate once you know the basic keystrokes:

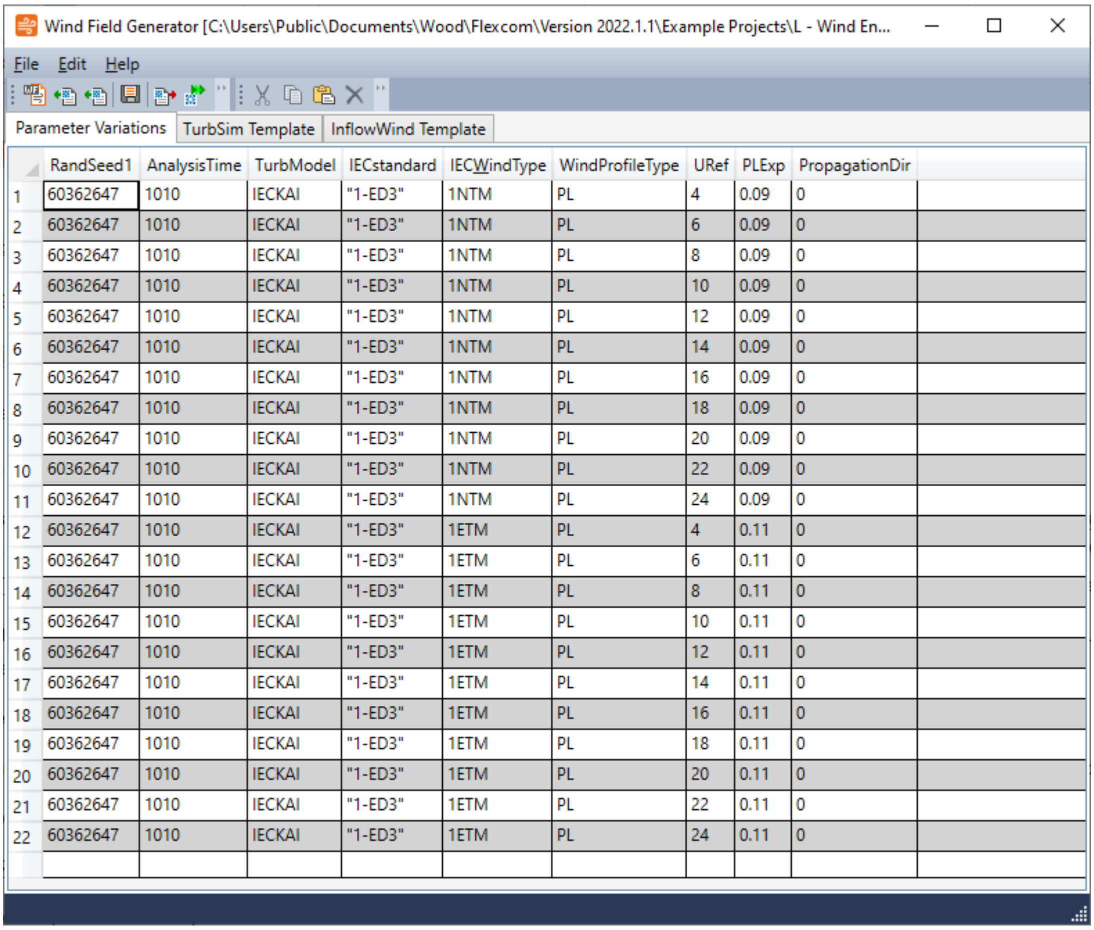
    a. Use the arrow or TAB keys to move between cells.

    b. To insert a new row, move to the bottom of the table, type data in a cell which you want to vary, and press ENTER. Data from the previous row is automatically copied into the new row, apart from the cell which you have just edited.

    c. Rows may be selected using SHIFT+UP or SHIFT+DOWN.

    d. CTRL+C and CTRL+V are used to copy and paste respectively.

    e. Rows may be deleted using DEL.



	RandSeed1	AnalysisTime	TurbModel	IECstandard	IECWindType	WindProfileType	URef	PLExp	PropagationDir
1	60362647	1010	IECKAI	"1-ED3"	1NTM	PL	4	0.09	0
2	60362647	1010	IECKAI	"1-ED3"	1NTM	PL	6	0.09	0
3	60362647	1010	IECKAI	"1-ED3"	1NTM	PL	8	0.09	0
4	60362647	1010	IECKAI	"1-ED3"	1NTM	PL	10	0.09	0
5	60362647	1010	IECKAI	"1-ED3"	1NTM	PL	12	0.09	0
6	60362647	1010	IECKAI	"1-ED3"	1NTM	PL	14	0.09	0
7	60362647	1010	IECKAI	"1-ED3"	1NTM	PL	16	0.09	0
8	60362647	1010	IECKAI	"1-ED3"	1NTM	PL	18	0.09	0
9	60362647	1010	IECKAI	"1-ED3"	1NTM	PL	20	0.09	0
10	60362647	1010	IECKAI	"1-ED3"	1NTM	PL	22	0.09	0
11	60362647	1010	IECKAI	"1-ED3"	1NTM	PL	24	0.09	0
12	60362647	1010	IECKAI	"1-ED3"	1ETM	PL	4	0.11	0
13	60362647	1010	IECKAI	"1-ED3"	1ETM	PL	6	0.11	0
14	60362647	1010	IECKAI	"1-ED3"	1ETM	PL	8	0.11	0
15	60362647	1010	IECKAI	"1-ED3"	1ETM	PL	10	0.11	0
16	60362647	1010	IECKAI	"1-ED3"	1ETM	PL	12	0.11	0
17	60362647	1010	IECKAI	"1-ED3"	1ETM	PL	14	0.11	0
18	60362647	1010	IECKAI	"1-ED3"	1ETM	PL	16	0.11	0
19	60362647	1010	IECKAI	"1-ED3"	1ETM	PL	18	0.11	0
20	60362647	1010	IECKAI	"1-ED3"	1ETM	PL	20	0.11	0
21	60362647	1010	IECKAI	"1-ED3"	1ETM	PL	22	0.11	0
22	60362647	1010	IECKAI	"1-ED3"	1ETM	PL	24	0.11	0

Sample Parameter Variations

4. By default, the Parameter Variations tab displays only a small subset of all possible entries in the TurbSim input file. These include some of the most commonly used inputs (e.g. IEC turbulence type, mean wind speed etc.). However you can enable many more columns via *File->Settings*.

5. Define a naming convention for the generated files via *File->Settings*. As you may have a large number of wind files, it is important to assign each one a meaningful name for clarity. The *File Name Pattern* input gives you complete flexibility in terms of file names:

    a. Plain text is applied unchanged to all files.

    b. {ParameterName} allows you to reference parameter values.

    c. {#} allows you to reference parameter row numbers.

    d. If you need to insert a bracket, you can use {{ or }}.

A sample naming convention might read as follows: {TurbModel}-{IEC_WindType}-{URef}-{RandSeed1}-{PropagationDir}. The resulting file names will reflect the main features of each wind profile (see sample BTS file names below).

6. Generate the wind files using *File->Generate Input Files & Run TurbSim*. You will be prompted for a folder where the generated files are to be stored. File generation is a two-step process:

    a. TurbSim input files (*.IN) and corresponding InflowWind input files (*.DAT) are created. These are text files created by the Wind Field Generator app and typically produced in seconds.

    b. TurbSim wind field data files (*.BTS) are created. These are large binary files produced by TurbSim, each one can take several minutes to produce.

Name	Date modified	Type	Size
IECKAI-1ETM-4-60362647-0.bts	05/08/2022 08:08	BTS File	52,238 KB
IECKAI-1ETM-6-60362647-0.bts	05/08/2022 08:10	BTS File	52,238 KB
IECKAI-1ETM-8-60362647-0.bts	05/08/2022 08:10	BTS File	52,238 KB
IECKAI-1ETM-10-60362647-0.bts	05/08/2022 08:10	BTS File	52,238 KB
IECKAI-1ETM-12-60362647-0.bts	05/08/2022 08:10	BTS File	52,238 KB
IECKAI-1ETM-14-60362647-0.bts	05/08/2022 08:10	BTS File	52,238 KB
IECKAI-1ETM-16-60362647-0.bts	05/08/2022 08:10	BTS File	52,238 KB
IECKAI-1ETM-18-60362647-0.bts	05/08/2022 08:10	BTS File	52,238 KB
IECKAI-1ETM-20-60362647-0.bts	05/08/2022 08:10	BTS File	52,238 KB
IECKAI-1ETM-22-60362647-0.bts	05/08/2022 08:10	BTS File	52,238 KB
IECKAI-1ETM-24-60362647-0.bts	05/08/2022 08:10	BTS File	52,238 KB
IECKAI-1NTM-4-60362647-0.bts	05/08/2022 08:08	BTS File	52,238 KB
IECKAI-1NTM-6-60362647-0.bts	05/08/2022 08:08	BTS File	52,238 KB
IECKAI-1NTM-8-60362647-0.bts	05/08/2022 08:08	BTS File	52,238 KB
IECKAI-1NTM-10-60362647-0.bts	05/08/2022 08:08	BTS File	52,238 KB
IECKAI-1NTM-12-60362647-0.bts	05/08/2022 08:08	BTS File	52,238 KB
IECKAI-1NTM-14-60362647-0.bts	05/08/2022 08:08	BTS File	52,238 KB
IECKAI-1NTM-16-60362647-0.bts	05/08/2022 08:08	BTS File	52,238 KB
IECKAI-1NTM-18-60362647-0.bts	05/08/2022 08:08	BTS File	52,238 KB
IECKAI-1NTM-20-60362647-0.bts	05/08/2022 08:08	BTS File	52,238 KB
IECKAI-1NTM-22-60362647-0.bts	05/08/2022 08:08	BTS File	52,238 KB
IECKAI-1NTM-24-60362647-0.bts	05/08/2022 08:08	BTS File	52,238 KB

Sample BTS Files

7. If you wish to QA the input files before running TurbSim, you can select *File->Generate Input Files* instead.

8. Save your project workspace before exiting the Wind Field Generator tool.

SAMPLE PROJECT

Example L04 - UMaine VolturnUS-S IEA15MW requires TurbSim BTS files to support the wind turbine simulations. You will find a sample project file for the app in the Data->Wind subfolder of the project, it is called L4-DLCs-Wind.wfproj. It is already pre-populated with the relevant wind speeds and turbulence models etc., so all you need to do is select the *File->Generate Input Files and Run TurbSim* command.

1.7.3 Running Analyses

SINGLE ANALYSIS

There are several methods to initiate a single Flexcom analysis, including:

- Selecting *Run Analysis* from the *Analysis* menu on the *Top Menu Bar*
- Right-clicking on the relevant keyword file in the File View to invoke the context-sensitive menu options, and then selecting the Run Analysis option
- Right-clicking in the Keyword Editor to invoke the context-sensitive menu options, and selecting *Analysis* and then the *Run Analysis* option

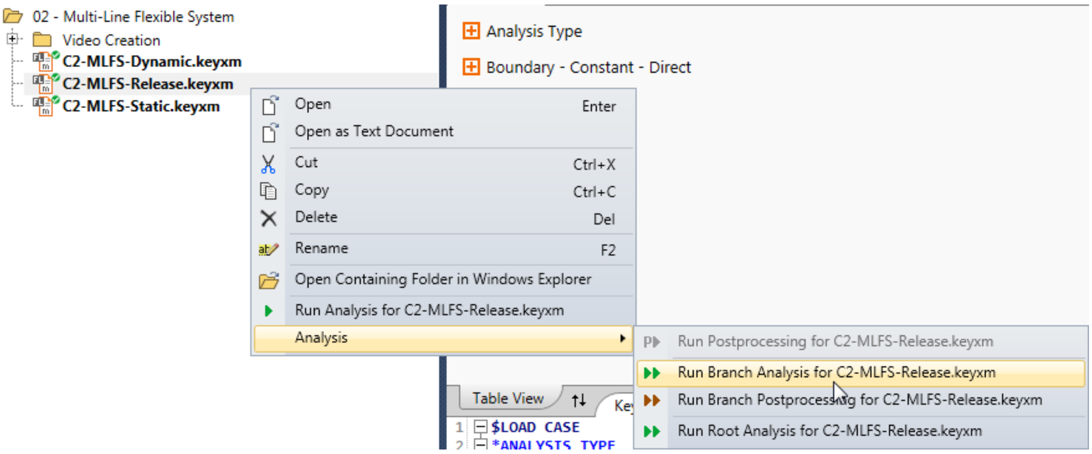
MULTIPLE ANALYSES

Flexcom also provides a range of options for performing a sequence of analyses without user intervention. Incidentally, the first four options are all available from the File View component.

- *Run Folder* causes Flexcom to execute all analyses in the selected folder and its subfolders.
- *Run Selected* causes Flexcom to execute all selected analyses, and analyses within any selected folders. You can quickly select several files and/or folders by pressing and holding the CTRL key, before using the left mouse button to click on the relevant entries. Pressing and holding the SHIFT key, in conjunction with the left mouse button, allows you to quickly select a range of entries.
- *Run Branch* causes Flexcom to run the selected analysis, and all analyses that are dependent on it in the project tree structure.

- *Run Root* is conceptually similar to *Run Branch*, but it searches for preceding analyses, and executes all analyses in the project tree structure up to and including the selected analysis.

- *Run Batch* allows you to build up a list of keyword files which you wish to analyse. In order to invoke the functionality, you must first create a batch file. Batch files have the file extension .fcbat (an acronym for FlexCom BATch). This method running is more complex than the others, and so is discussed separately in Batch Files.



Sample Branch Run

RUNNING A FLEXCOM ANALYSIS OUTSIDE THE FLEXCOM USER INTERFACE

Occasionally some users wish to by-pass the Flexcom user interface, and run a Flexcom simulation from a command prompt window or another external program. In order to do this, you simply call Flexcom's pre-processing module, give it the name and location of the Flexcom keyword file which you wish to analyse, and optionally provide some additional command line arguments.

Flexcom's pre-processing module is called 'fl3.exe' and it is located in your local installation directory. The location varies depending on which version of Flexcom you have installed, but will be typically something like... "*C:\Program Files\Wood\Flexcom\Version 2022.1.1\bin\fl3.exe*". Note that if the file path contains spaces, you must enclose the character expression in quotation marks.

The name and location of the Flexcom keyword file naturally depends on the simulation which are interested in performing. Let's suppose you wish to run one of the standard Flexcom examples set. The relevant text might resemble the following... "*C:\Users\Public\Documents\Wood\Flexcom\Version 2022.1.1\Example Projects\A - Top Tensioned Risers\01 - Deepwater Drilling Riser\Connected\A1-DDR-Static_c.keyxi*". Note again how quotation marks are used.

There are also a number of optional arguments which you can supply to Flexcom to indicate certain preferences. All arguments are optional and can be used in any order. In reality, only the first option will be of any interest to the majority of users. The remainder are typically used by Wood's technical support team.

- */p* - runs post-processing only (i.e. skips finite element simulation and proceeds directly to post-processing)
- */f "File Name"* - saves a copy of the pre-processed keyword file to a specified location (this may be useful for power users who wish to see how Lines have been meshed into explicit Nodes and Elements)
- */z* - does not run simulation or post-processing at all (perhaps useful if you are only interested in examining the pre-processed keyword file)
- */g* - simulates running from graphical user interface (i.e. simulation does not output results to console window)
- */s* - hides analysis console window
- */v* - shows a message box which contains the command line used to run the analysis control module
- */nw* - suppresses warning messages which occur during keyword pre-processing (this may be useful if a particular model issues a large number of similar warning messages which you are satisfied to accept)

Let's suppose you wish to post-process one of the standard Flexcom examples set. The entire command line might resemble the following... *"C:\Program Files\Wood\Flexcom\Version 2022.1.1\bin\fl3.exe" "C:\Users\Public\Documents\Wood\Flexcom\Version 2022.1.1\Example Projects\A - Top Tensioned Risers\01 - Deepwater Drilling Riser\Connected\A1-DDR-Static_c.keyxi" "/p".* When a finite element simulation is launched from the user interface under normal circumstances, the very same calling procedure is followed by the user interface. In this case however, the pre-processing module is called automatically in the background so the entire process appears seamless to the user.

If you would like to run a list of simulations in series, you can assemble a suitable batch file which contains a series of commands, like the following:

- set Flexcom="C:\Program Files\Wood\Flexcom\Version 2022.1.1\bin\fl3.exe"
- set Folder=C:\Users\Public\Documents\Wood\Flexcom\Version 2022.1.1\Example Projects\A - Top Tensioned Risers\01 - Deepwater Drilling Riser\Connected\
- call %Flexcom% "%Folder%A1-DDR-Offcur_c"
- call %Flexcom% "%Folder%A1-DDR-Modal_c"
- call %Flexcom% "%Folder%A1-DDR-Shear7_c"
- call %Flexcom% "%Folder%A1-DDR-Reg_c-td"
- call %Flexcom% "%Folder%A1-DDR-Reg_c-fd"
- call %Flexcom% "%Folder%A1-DDR-Ran_c-td"
- call %Flexcom% "%Folder%A1-DDR-Ran_c-fd"

Note that if there are any restart dependencies between the files, the simulations must be listed in the correct order of execution (e.g. initial static analyses, followed by vessel offsets, followed by currents, followed by dynamics etc.).

If you have a very large number of simulations to execute, performing the analyses sequentially is not a very efficient method as it only utilises one of the available CPUs on your computer. If you would like to run several simulations in parallel, you could use a more advanced batch file, such as the one provided for Running TurbSim. This batch file is designed to execute a number of processes in parallel, controlled via the 'maxProc' entry which should be less than the number of available CPUs. Note also that the list of files to be executed must still respect the correct order with respect to restart dependencies.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Analysis Jobs
- Analysis Status View
- Work Sequence
- Batch Files

1.7.3.1 Analysis Jobs

An analysis job is typically comprised of several individual analyses, although a job may equally consist of a single analysis run only. For example, if you run an individual keyword file by pressing the F5 key while editing the keyword file, this action commences an analysis job which is comprised of a single analysis only. Flexcom also provides a range of options for performing a sequence of analyses without user intervention. These include Run Folder, Run Selected, Run Branch, Run Root and Run Batch.

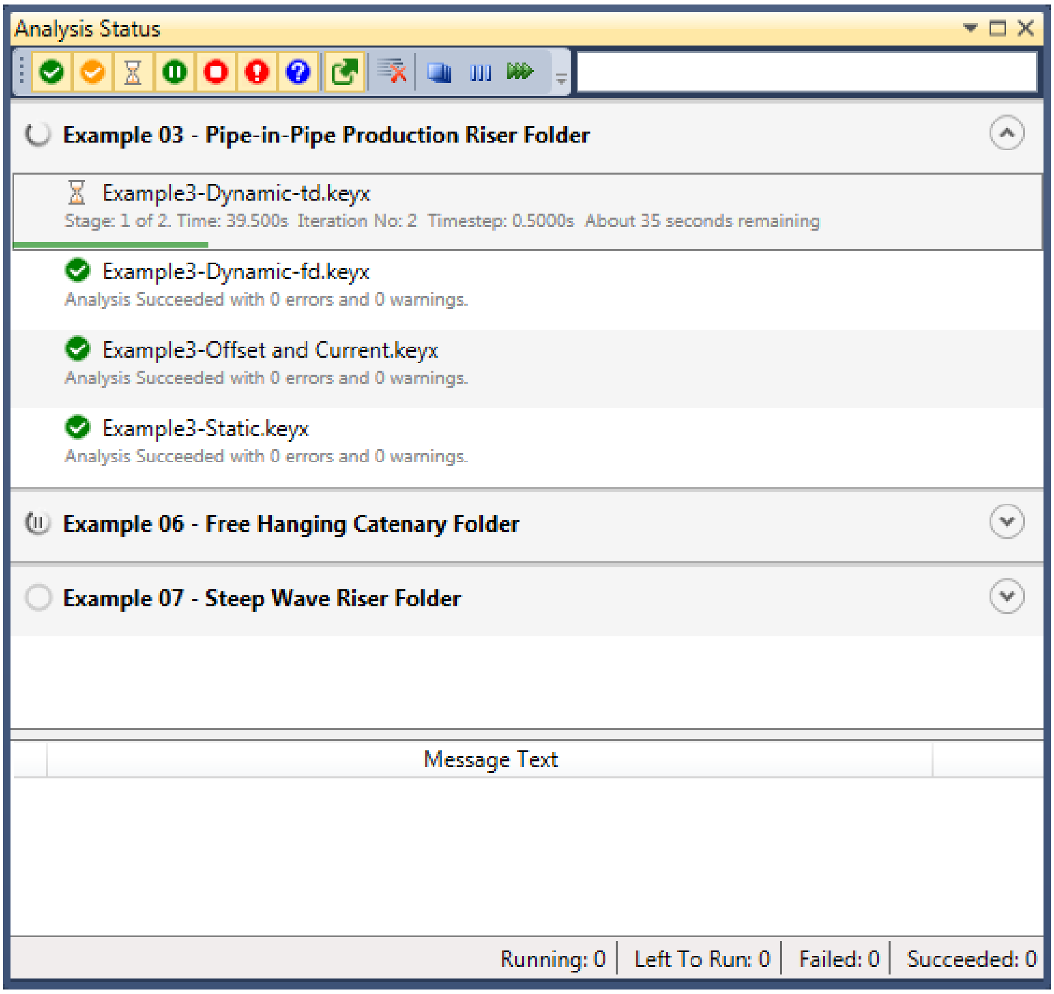
When you initiate a job from the File View, or any other means, the job appears in the Analysis Status View. It is assigned an appropriate name by default, depending on how the job was initiated. For example, this could be an individual file name, a file name followed by Branch Analysis or Root Analysis, or a folder name.

In the image below, there are three jobs as follows:

- Example 03 - Pipe-in-Pipe Production Riser. This job is currently in progress, as indicated by the rotating circle icon to the left of the job name. The job is shown in expanded view mode (which is controlled via the arrow control to the right of the job name), such that information pertaining to its descendant analyses is visible also.

- Example 06 - Free Hanging Catenary. This job is currently paused, as indicated by the pause icon to the left of the job name. The job is shown in contracted view mode, such that information pertaining to its descendant analyses is hidden.

- Example 07 - Steep Wave Riser. This job is fully complete, as indicated by the static circle icon to the left of the job name. The job is shown in contracted view mode, such that information pertaining to its descendant analyses is hidden.



Sample Analysis Jobs

1.7.3.2 Analysis Status View

OVERVIEW

The Analysis Status View has a number of functions, including:

- Providing a high level summary of all Analysis Jobs and their current status.

- Presenting more detailed information regarding the progress of individual analyses within a particular analysis job.

- Offering options to manipulate the current Work Sequence. For example, individual analyses or entire analysis jobs may be paused, restarted or terminated if so desired. It is also possible to reprioritise events – certain jobs or analyses may be designated as higher priority than others.

ANALYSIS JOBS

The current status of all analysis jobs is summarised by the Analysis Status View, using the following icons.

- Running. The rotating circle icon indicates that the analysis job is currently in progress.

- Paused. The rotating circle icon containing the pause symbol indicates that the analysis job is in progress but has been paused.

- Completed. The static circle icon indicates that the analysis job has completed.

PROGRESS OF INDIVIDUAL ANALYSES

More detailed information regarding the progress of individual analyses within a particular analysis job is also available. A progress bar indicator allows you to monitor the progress of an analysis while it is running, and an approximate estimate of remaining CPU time is also provided in the case of time domain dynamic analyses. If you select any particular analysis by clicking on it, the lower area of the Analysis Status view displays any warnings or error messages which appear at run time. In the Sample Analysis Jobs image, Example-3-Dynamic-td.keyx is approximately 20% complete, and has an estimated completion real-time of 35 seconds.

STATUS ICONS

The status icons which accompany each individual analysis allow you to see at a glance which ones have completed successfully, and which (if any) require further attention. The symbols used are as follows:

- Running. This indicates that the analysis is currently in progress.

- Running, with Warnings. This indicates that the analysis is currently in progress, but has some associated warnings. Inspection of relevant output messages is recommended.

- Paused. This indicates that the analysis is in progress but has been paused.

- Successful. This indicates that the analysis completed successfully, without any warnings or errors.

- Successful, with Warnings. This indicates that the analysis completed successfully, but with some warnings. Inspection of relevant output messages is recommended.

- Failed. This indicates that the analysis failed to complete successfully. Inspection of relevant output messages is essential, with a view to successfully rerunning a modified analysis.

- Terminated. This indicates that the analysis has terminated prematurely, for example if the process was cancelled by some external source such as the Windows Task Manager.

- Licensing Issue. This indicates that you do not have the appropriate license to perform the analysis. Contact the technical support service at sw.support@woodplc.com if required.

- Unknown. This indicates that the analysis status is unknown. Typically this means that the analysis has not yet been run, or perhaps the analysis was stopped by the user before it had reached its completion.

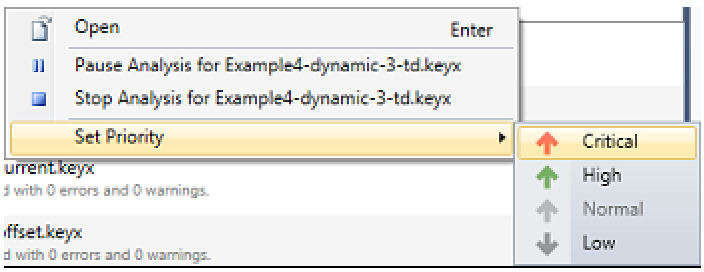
- Obsolete. This indicates that the analysis has been run previously, but the keyword data of this or preceding analysis has since been modified, and its results are now obsolete and potentially misleading.

1.7.3.3 Work Sequence

By default, each analysis job assumes the same priority as any other analysis jobs which are currently in progress. This means that Flexcom will evenly distribute the active analysis jobs among the available processing units on your machine.

In terms of the individual analyses themselves within a given analysis job, Flexcom automatically compiles a default run sequence. The program is aware of the hierarchy within the project workspace, so it recognises the order in which analyses must be executed. For example, a restart analysis cannot be not initiated until the preceding analysis has completed. However, within this framework, there may be several combinations which are equally valid. For example, if the analysis job contains 12 vessel offset analyses in different directions, all of which restart from an initial static analysis, they are entered into the run-queue simply on the basis of alphabetical order.

The default event sequencing provided by the program may be acceptable in the majority of cases, but the Analysis Status view offers some additional options to manipulate the current work sequence. For example, individual analyses or entire analysis jobs may be paused, restarted or terminated if so desired. It is also possible to re-prioritise events – certain job or analyses may be designated as higher priority than others. Should you wish to obtain results from a particular analysis urgently, you can manually instruct the program to treat the analysis as higher priority. Simply right-click on the relevant job or analysis, and select the desired priority level, as shown below.



1.7.3.4 Batch Files

The Batch run facility allows you to build up a list of keyword files which you wish to analyse. In order to invoke the functionality, you must first create a batch file; one method to do this is to invoke File -> New File from the Top Menu Bar and then selecting the Batch option. Batch files have the file extension .fcbat (an acronym for FlexCom BATch).

The operation of the Batch run facility is fairly self-explanatory – you define various rules and criteria which determine whether a keyword file should be included in or excluded from the batch run. Use the and buttons at the top to add new rules or remove existing ones. Once a rule is selected in the Rules section, its definition may be viewed or modified in the Criteria section underneath. Some additional points to note are as follows:

- Include rules are evaluated as logical TRUE or FALSE constants. If more than one Include rule is defined, they are combined in a similar manner to logical OR operators. Specifically, if a keyword file matches any of the specified Include rules then it is included in the file list (assuming for simplicity that there are no exclude rules defined).
- Each rule is defined using one or more criteria. Criteria may be defined in terms of directory, file names (wildcard operators may be used) or file creation/modification dates. If more than one criterion is defined, the criteria within that rule are combined in a similar manner to logical AND operators. Specifically, a keyword file must match all of the specified criteria within a rule for that particular rule to be satisfied.
- Exclude rules are evaluated as logical TRUE or FALSE constants, and effectively behave in a similar manner to logical NOT operators. If more than one Exclude rule is defined, they are combined in a similar manner to logical OR operators, and the overall result is applied as a NOT operator. Specifically, if a keyword file matches any of the specified Exclude rules then it is excluded in the file list.
- Where both Include and Exclude rules are defined, overall Include and Exclude results are evaluated using logical OR operators as mentioned already. The Exclude result is then inverted using the NOT operator, and an overall final TRUE/FALSE result is obtained by combining the Include and Exclude results using the logical AND operator. Specifically, if a keyword file matches any of the specified Include rules, and it does not match any of the specified Exclude rules, then it is included in the file list.

An option is provided to preview the list of keyword files before initiating the actual run series. This is facilitated by the File View, which only displays the keyword files pertaining to the Batch run, rather than all the keyword files included in the project workspace.

When you perform a Batch run, a list of analyses is compiled initially and each keyword file in the list is then entered into a “run queue”. For maximum efficiency, Flexcom determines how many processing units are available on your machine, and endeavours to continually match the number of analyses in progress with the number of available processing units.

Note also that the program is aware of the hierarchy within your project workspace, so it recognises the list of analyses which should be performed, and the order in which they should be executed (so a restart analysis is not initiated until the preceding analysis has completed).

1.7.4 Job Execution Service & Troubleshooting

OVERVIEW

There is a clear distinction between the graphical user interface and the finite element engine in Flexcom – these are, and always have been, separate programs in their own right. In all versions up to and including Flexcom 8.2, the user interface directly controlled the finite element engine. So for example, when you press the Run button to start an analysis, the user interface program invokes the finite element analysis program, providing the keyword data as input. During run-time, the interface and engine programs are in direct communication, so that the interface can display relevant progress information, and inform the user when the analysis is complete.

With the advent of Flexcom 8.3, there is an intermediary entity which manages the interaction between the interface and engine programs. When you initiate a run, the user interface now passes an execution command to an execution service, which in turn invokes the finite element engine. This approach has a number of important advantages. Firstly, the use of an intermediary service affords the user far greater control over the execution order of analyses, as noted above. Secondly, the analysis jobs may proceed as normal in the absence of the user interface. So you may close the user interface in order to free up resources, and restart it again later if desired. If your machine happens to shut down unexpectedly (e.g. via crash or manual reboot) while analysis jobs are in progress, the execution service re-launches itself upon restart and simply picks up where it left off.

In practice, the operation of the job execution service should be completely seamless, such that you need not even be aware of its existence. When the service is operating correctly, a green traffic light icon appears on the lower right hand corner of the user interface, as shown below. Strictly speaking, this feature is not contained within the Analysis Status view – it is actually part of the Status Bar to ensure it remains visible at all times – but as is inherently linked to the Analysis Status view, it is described here for completeness.

RESTARTING THE JOB EXECUTION SERVICE

Occasionally the user interface may lose connectivity with the service and stop receiving analysis status updates. Should this happen, you can manually restart the execution service. To reboot the service, you can right click on the traffic light icon to invoke the context-sensitive menu, and select 'Restart Flexcom Services' from the list of options as shown. It may also be beneficial to 'Stop Flexcom Services', wait a few seconds, then 'Start Flexcom Services' again. If the 'Reset Flexcom Service' option is chosen from the context menu on the traffic light then the database will be reset (in addition to the services being restarted), removing everything from the Analysis Status View.

In rare circumstances, the services icon will appear green but it is not possible to run any Flexcom simulations. This problem typically manifests itself by the job status appearing to be permanently stalled (i.e. the cursor continually displays the rotating circle icon in the analysis status view). Should this happen, restarting the job execution service as suggested above may not fix the problem. Sometimes Flexcom does not have sufficient permissions to stop and start services even if it is running it administrator mode, and in this case you should try using the Windows Services app directly.

If you find yourself in this predicament, try the following steps...

1. Pause the job, and wait for the software to confirm it is paused.
2. Stop the job, again wait until you receive a stop confirmation.
3. Stop Flexcom services by right clicking on the traffic light icon. Wait a few seconds.
4. Start Flexcom services.
5. Now try running the job again. If it runs fine, disregard the remaining points. If not, proceed to step 6.
6. Follow steps 1-2 again.
7. Type ‘services’ into Windows start menu search bar, and when you see ‘Services’, right click on it and select ‘Run as Administrator’.

8. Examine the list of services and look for the ‘Flexcom Execution Engine’ service which corresponds to the version you are running. Once you find it, right click and select 'Stop Service'.

9. If you are running and older version of Flexcom (e.g. 8.6.4 or earlier), look for ‘Flexcom Watcher Service’ for that version, and stop the service also.

10. Now try running job again. If it still fails to run, please contact us. We will be able to deploy debugging tools to get Flexcom up and running again for you.

LOCAL FIREWALL SETTINGS

If you are persistently experiencing communication problems between the user interface and the execution service, it is likely that communication is being blocked by a local firewall. Users in this predicament will either see a red traffic light icon at all times (even after following the restart procedure discussed above), or analysis jobs will appear to hang indefinitely (denoted by a rotating circle icon), or perhaps analysis runs will remain in the unknown state (denoted by a blue question mark icon). You should contact your local IT department and ask them to check if firewall settings are blocking Flexcom communications. Typically this may be caused by anti-virus software or organisational security policies but the good news is that the problem may be readily overcome by adding some exceptions to these generic rules.

Flexcom's installer pack adds all the required rules and exceptions to Windows Firewall during the software installation process. However if a third party firewall is installed, then a local IT person will have to make some configuration changes themselves. For example, McAfee firewall is known to cause such issues, and Flexcom users have successfully added exceptions to the McAfee settings to prevent it from interfering with the software. The following network communication capabilities are required for Flexcom to work.

For the following processes...

- MCS.Flexcom.exe
- MCS.Flexcom.Wind.exe
- MCS.Flexcom.Wave.exe
- AnalysisExecutionEngine.exe
- MCS.Flexcom.AnalysisLauncher.exe

- client.exe
- fl3.exe

...the following networking access is required:

- In/Out UDP communications to/from localhost, port 3702
- In UDP communications to 239.255.255.250 from localhost, port 3702
- In/Out TCP communications to/from localhost, port 8000 (note that if this port number is used by another application, it is possible to configure Flexcom to use different port number via the *Flexcom Settings* dialog)

Several different versions of Flexcom may exist side-by-side on a single machine (users often wish to revert to an earlier version if some re-work is required to an old engineering project after its completion). In order to facilitate this, Flexcom's installer pack includes the version number in the default installation path. In the following examples, X.X.X is the version number being installed.

Recent versions of Flexcom are installed to:

- C:\Program Files\Wood\Flexcom\Version X.X.X\Bin (e.g. C:\Program Files\Wood\Flexcom\Version 8.10.4)

Versions of Flexcom prior to 8.9 were installed to:

- C:\Program Files\Wood Group\Flexcom\Version X.X.X\Bin (e.g. C:\Program Files\Wood Group\Flexcom\Version 8.6.4)

Versions of Flexcom prior to 8.6 were installed to:

- C:\Program Files\MCS Kenny\Flexcom\Version X.X.X\Bin (e.g. C:\Program Files\MCS Kenny\Flexcom\Version 8.4.1)

Your local IT personnel should ensure to use wildcards where applicable to account for variations described above.

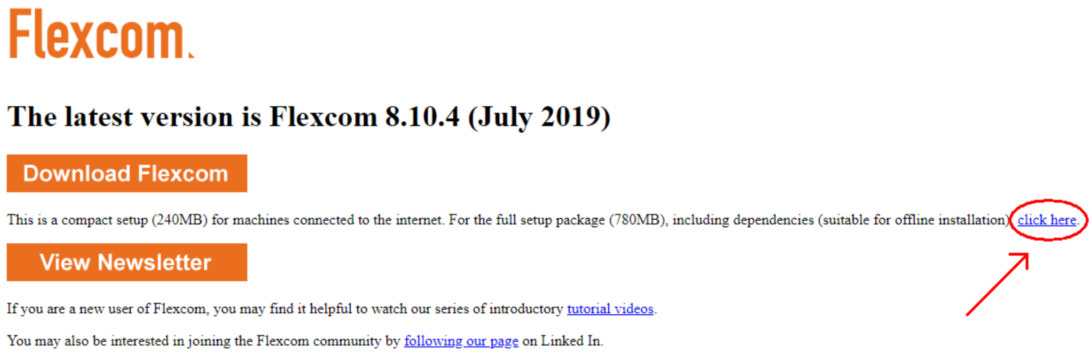
APPLICATION ERROR FROM ACM

Occasionally we have received reports from users that Flexcom fails to run any simulation. The analysis control module (the pre- and post-processing part of the finite element program) issues an error message similar to the following:

- The application was unable to start correctly (0xc000007b). Click OK to close the application.
- ACM stopped working. Error code '0xC0000138'.

Although unrelated to the job execution service, the issue is covered here as this page is designed to help users to overcome any issues with simulation execution. This problem actually relates to redistributables - these are libraries of software components which Flexcom relies on to operate correctly. The redistributables are either not installed correctly or have somehow become corrupted after a successful installation. Should you encounter this problem, follow these steps to rectify this issue.

1. This process requires administrator privileges. If you do not have local admin rights on your machine, contact your IT department for assistance.
2. Close any Flexcom applications which you have running.
3. Check if your machine is 64-bit or 32-bit.
4. Open the Windows Control Panel and look for an option to uninstall programs, typically found under 'Programs and Features'. The layout of options will depend on which version of Windows you have. Open the list of programs which are installed on your machine.
5. Using the search bar in the top right hand corner of your screen, search for “Visual Fortran” in the list of installed programs.
6. Remove/uninstall any programs called "Intel(R) Visual Fortran Redistributables on Intel(R)64" or similar. Remove all versions, both 32 and 64-bit.
7. Download the Flexcom installer pack. Please ensure to download the full offline version of the installer, as shown below.



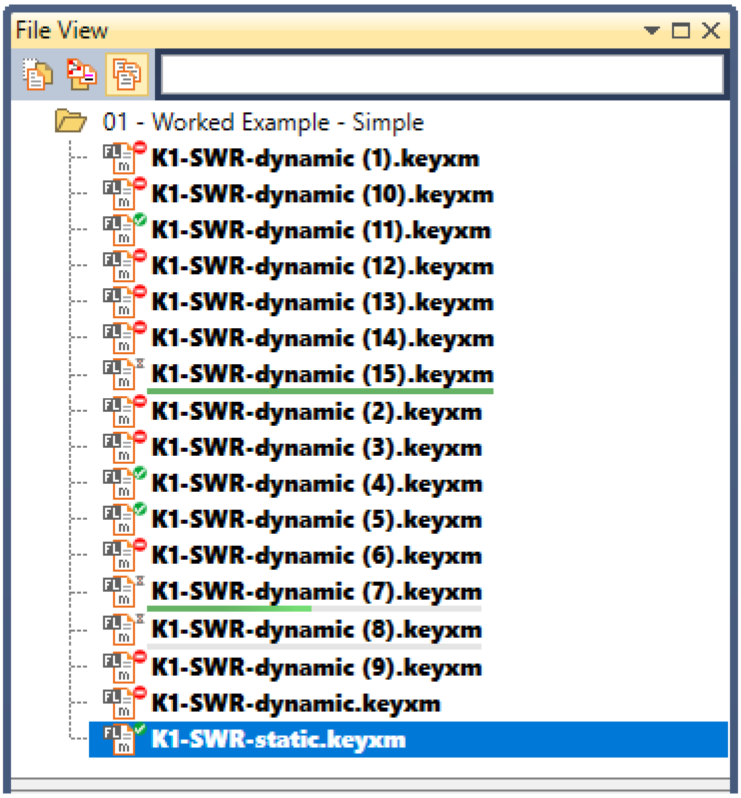
8. Unzip the installer pack to some location on your local hard drive.

9. Using Windows Explorer, open the "Prerequisites" sub-folder of the pack, then the "IVFRedist" sub-folder.

10. Install the Intel(R) Visual Fortran Redistributables, by clicking on the relevant MSI file, depending on whether your machine is 64-bit or 32-bit.

SIMULATIONS UNEXPECTEDLY MARKED AS TERMINATED

Occasionally the job execution service can report simulations as Terminated ( ) even though they have completed successfully. These are known as '*false failures*' by Flexcom power users. This is caused by a delay in the communication between the user interface and the finite element engine. The problem only occurs sporadically, but seems to be most prevalent in large batch runs, when the machine is fully loaded and all CPUs are busy. It can be a source of inconvenience as a large analysis job can stall halfway through, as it can prevent files in a restart chain from starting if earlier ones are deemed to have failed. Most users will never experience this issue, but if you are experiencing problems, some helpful advice is given here.



False Failures

The Execution Engine is designed to detect situations where an analysis process has stopped unexpectedly, or was terminated deliberately, and in such cases the analysis is marked as Terminated ( ). In some cases this detection may be false and occur a result of a major slowdown in communication between an analysis process and the Execution Engine. This may happen when an analysis job with thousands of analyses is running and the CPU utilisation on the machine is 100% or close to it. In such cases, even though the analysis is marked as Terminated, the related .OUT file for the analysis is present and contains "Successful Flexcom Analysis" message at the end of it.

CPU / Core Reservation

The best way to reduce the likelihood of the problem occurring is to reserve a CPU core or "logical processor" for the user interface and its associated tasks (see notes). This helps to alleviate congestion in the communication channels and rectifies the issue in the vast majority of cases where it occurs.

1. If you are running Flexcom 8.13.3 or earlier, then you can achieve this by doing either of the following:

a. Keep the Flexcom user interface open (especially during batch runs) - Flexcom automatically reserves one logical processor for the user interface while it remains open.

b. Use Flexcom Settings to limit the number of analysis processes that can be started simultaneously to at least one less than the number of logical processors available on the machine. This setting will remain in effect even if you close Flexcom user interface. You can find the number of available cores in Windows Task Manager (see notes).

2. If you are running a later version, then you don't have to worry, Flexcom is already configured to keep one logical processor for handling user interface and communications. It is possible to override this setting via the Flexcom Configuration File if you wish, but it is not recommended if you're running large batches of analyses.

Communication Time Settings

Advanced users may wish to change communication timings via the Flexcom Configuration File. You should first try to reduce the analysis update time setting *PreKeyUpdateInterval* to reduce the frequency of communications between analysis processes and the Execution Engine. If that does not help you can increase the analysis disconnect time-out setting *PreKeyClientDisconnectWaitTimeout* to make sure that an analysis process waits long enough for the Execution Engine to receive and process the communication messages. If changing those settings does not help, please contact software support.

Notes

- A CPU (sometimes referred to as a socket) is a physical processor unit inside a computer, there can be one or more CPUs installed in a computer.
- A Core is a part of the CPU that is responsible for running a process (e.g. Flexcom Analysis), a CPU typically has multiple Cores to facilitate multiple processes in parallel.
- A Logical Processor is an ability of a Core to run more than one process at a time (typically two processes per Core) and is a feature present in most of the more powerful CPUs.

- When you open the Performance tab in Windows Task Manager you can see the configuration of your machine and how many CPUs, Cores, and Logical Processors are present.

1.7.4.1 Flexcom Configuration File

The Flexcom Configuration File contains advanced settings relating to Flexcom and is intended to give power users more control over how the program operates. The vast majority of users will never need to modify this file, or may not even be aware that it exists. This file is called FlexcomGlobal.config and is located in the C:\ProgramData\Wood\Flexcom\Version 2022.1.1. It contains a range of user-configurable settings are shown below.

```xml
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <appSettings>
        <add key="DatabaseDir" value="C:\ProgramData\Wood\Flexcom\Version 8.13.3\" />
        <add key="ExamplesDir" value="C:\Users\Public\Documents\Wood\Flexcom\Version 8.13.3\Example Projects\" />
        <add key="MaxProcessesLimit" value="0" />
        <add key="MinFreeDiskSpace" value="1048576000" />
        <add key="GenerateCrashReports" value="True" />
        <add key="DebugOutput" value="False" />
        <add key="ReserveCPUCore" value="True" />
        <add key="PreKeyUpdateInterval" value="3000" />
        <add key="PreKeyClientConnectWaitTimeout" value="20000" />
        <add key="PreKeyClientDisconnectWaitTimeout" value="300000" />
    </appSettings>
</configuration>
```

Flexcom Configuration File

The file contains pairs of keys and corresponding values. The significance of each key is outlined in the following table.

Key	Function
DatabaseDir	The location where the Flexcom usage database is stored. This database contains information about executing or already executed analyses and analysis jobs (such as state of the execution job queue, individual analysis status and messages, etc.). This is typically C:\ProgramData\Wood\Flexcom\Version 2022.1.1 or similar.
ExamplesDir	The location where the Flexcom examples are stored. This is typically C:\Users\Public\Documents\Wood\Flexcom\Version 2022.1.1\Example Projects or similar.

Key	Function
MaxProcessesLimit	The maximum number of analysis processes which may be executed in parallel. You can set this value to 0 for an unlimited amount. The actual number of analyses that can be executed in parallel is limited by the smallest of the following values: (i) this setting, (ii) the number of Logical Processors (see notes) available on the machine, (iii) the commercial license agreement. See also ReserveCPUCore setting below. Default value is 0.
MinFreeDiskSpace	The minimum free disk space in MB. Flexcom will not start a new analysis process when available free disk space on the disk where the keyword file is located is below this number. If this happens the analysis execution job containing the analysis will pause automatically. Once the disk space is freed up you can resume the execution job. Default value is 1000.
GenerateCrashRepo rts	A logical value which allows Flexcom to generate an error report in the event of a user interface crash. These crash reports provided specialised information which enables the technical support team to fix bugs which are difficult to reproduce. Default value is True.
DebugOutput	A logical value which allows Flexcom to generate log files. These files contain detailed information about Flexcom processes and enable the technical support team to assist users who are having specialised issues with the software. Default value is False.

Key	Function
ReserveCPUCore	A logical value which allows Flexcom to automatically reserve one Logical Processor (see notes) for the user interface and its associated tasks. This helps to keep the user interface usable and responsive when the machine is heavily loaded with finite element simulations. It also helps to alleviate other potential issues (see False Failures for further details). Available in Flexcom versions newer that 8.13.3. Default value is True.
PreKeyUpdateInterval	The analysis progress update interval, in milliseconds, specifies how often the analysis progress information is updated. Only change this setting if you have False Failures problems. Available in Flexcom versions newer that 8.13.3. Default value is 3000 which is 3 seconds.
PreKeyClientConnectWaitTimeout	The connection wait time, in milliseconds, specifies how long Execution Engine, will wait for analysis progress information to come from analysis process before considering it a failure. Only change this setting if you have False Failures problems. Available in Flexcom versions newer that 8.13.3. Default value is 20000 which is 20 seconds.
PreKeyClientDisconnectWaitTimeout	The disconnect wait time, in milliseconds, specifies how long an analysis process will wait for Execution Engine to acknowledge that it has received all progress updates for that analysis. If the analysis process finishes before receiving such acknowledgement, Flexcom will consider it a failed analysis. Only change this setting if you have False Failures problems.

Key	Function
	Available in Flexcom versions newer that 8.13.3. Default value is 300000 which is 5 minutes.

If you are running Flexcom 8.13.3 or earlier, then you should keep the user interface open at all times (especially during batch runs), as Flexcom automatically reserves one CPU for the user interface while it remains open. Some users close the user interface while a large batch run is in progress in an effort to make all CPUs available and maximise throughput of finite element simulations, but this is not recommended.

If you are running a later version, then you do not need to worry, as Flexcom automatically keeps one CPU on standby, unless you specifically request otherwise, which again is not recommended. In other words, even if you close the user interface, this does not mean that all CPUs are available for finite element simulations.

Notes

- A CPU (sometimes referred to as a socket) is a physical processor unit inside a computer, there can be one or more CPUs installed in a computer.
- A Core is a part of the CPU that is responsible for running a process (e.g. Flexcom Analysis), a CPU typically has multiple Cores to facilitate multiple processes in parallel.
- A Logical Processor is an ability of a Core to run more than one process at a time (typically two processes per Core) and is a feature present in most of the more powerful CPUs.
- When you open the Performance tab in Windows Task Manager you can see the configuration of your machine and how many CPUs, Cores, and Logical Processors are present.

1.7.5 Analysis Preview

Flexcom provides a number of helpful options for previewing wave loading and associated vessel responses, before commencing a finite element simulation.

- Wave Spectrum Plot provides a graphical preview of the wave spectrum for analyses which include a random seastate, such as Pierson-Moskowitz, Jonswap, Ochi-Hubble or Torsethaugen.

- RAO Response Plot option is a useful feature for verifying that vessel RAO data has been input correctly into Flexcom. It graphically presents the vessel response as a function of incident wave heading and frequency, for each of the local Vessel Degrees of Freedom.

- Water Elevation Plot provides a preview of the water surface elevation, at any point in the wave field, as a function of time.

- Vessel Response Plot provides a preview of the vessel response to the applied wave loading as a function of time, in each of the local Vessel Degrees of Freedom. This provides confidence that the specified RAO data does in fact lead to the expected vessel motions.

1.7.5.1 Wave Spectrum Plot

The *Wave Spectrum Plot* provides a useful graphical preview of a wave spectrum, before embarking on a 3-hour time domain simulation of a random seastate. Spectral Density is plotted as a function of frequency for any wave spectrum defined in the input data. If more than one wave spectrum is included in the input data, the name of the desired spectrum may be selected from the *Plot* drop-down list.

Note that the *Wave Spectrum Plot* operates in a synchronised manner with the relevant input data. For example, when any data pertaining to wave spectra is altered in the keyword file, the plot updates automatically.

Flexcom presents a wide range of wave specification options. There are five options for defining a random seastate in terms of a wave spectrum, as follows:

- Pierson-Moskowitz describes the Pierson-Moskowitz wave spectum model, and presents the various formats for input data specification.

- Jonswap describes the Jonswap wave spectum model, and presents the various formats for input data specification.
- Ochi-Hubble outlines the Ochi-Hubble wave spectum model.
- Torsethaugen outlines the Torsethaugen wave spectum model.
- User-defined describes a facility whereby you can input an arbitrary wave spectrum directly in terms of a series of data pairs.

Refer to Wave Loading for further details.

1.7.5.2 RAO Response Plot

OVERVIEW

The *RAO Response Plot* feature is a useful means of verifying that vessel RAO data has been input correctly into Flexcom. In the most general case, RAO magnitudes and phase angles vary with both incident wave heading and wave frequency, so the complete specification of RAO data for any given vessel can be quite extensive. Furthermore, Flexcom has its own set of conventions regarding the specification of RAO data, and the program supports RAO Conversions from third-party software.

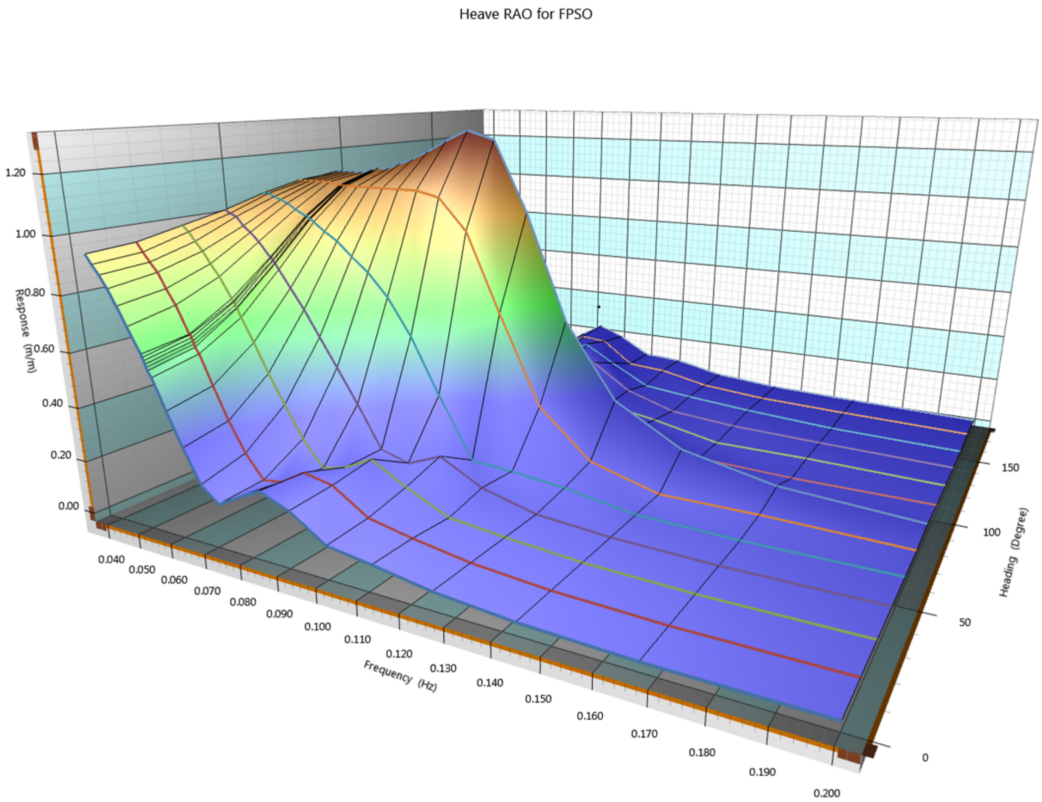
The *RAO Response Plot* feature graphically presents responses for each of the local Vessel Degrees of Freedom. Heave is shown by default, but the other directions may be examined using the *Plot* drop-down list. RAO magnitudes are generally of most interest, but it is also possible to examine RAO phase angles. If more than one vessel is included in the input data, the name of the desired vessel may be selected from the *Vessel* drop-down list.

2D & 3D DISPLAY OPTIONS

In 2D mode, vessel response is presented as a function of wave frequency in a 2-dimensional space. If RAO data is specified at several wave headings, separate data series are superimposed on the same plot, each using a different colour, as designated by the plot legend.

In 3D mode, vessel response is presented as a function of both wave frequency and wave heading in a 3-dimensional space. Viewing data in 3D space provides enhanced visualisation and is particularly suited to RAO data.

The following sample plot presents heave RAO data (m/m) as a function of wave heading (degrees) and wave frequency (Hz).



Sample 3D Plot

You can alternate between 2D and 3D views by pressing the button.

VESSEL SPECIFICATION

A vessel may be defined using either one of two keywords, `*VESSEL` or `*VESSEL,INTEGRATED`. While the former is perfectly valid, it has effectively been superseded by the latter, which is now the recommended option. *VESSEL,INTEGRATED is a more comprehensive keyword and has several advantages over its predecessor. Most importantly it accepts RAO data also, thereby eliminating the need for a separate `*RAO` keyword – hence the 'integrated' nature of the new keyword. Further details are available in Basic Vessel Concepts.

In general, when performing an analysis where RAOs are specified as a function of wave heading, it is always recommended to examine the time dependent motions of floating vessels. This provides confidence that the specified RAO data does in fact lead to the expected vessel motions. Flexcom also provides a related feature, the Vessel Response Plot option, to readily facilitate such an examination.

1.7.5.3 Water Elevation Plot

Flexcom provides an option to preview the water surface elevation as a function of time, before commencing a time domain finite element simulation. By default, the wave elevation is presented at the origin {Y=0, Z=0}, but any arbitrary point in the wave field may also be selected. Specifically, you can explicitly specify an alternate location via the *Y* and *Z* inputs in the *Water Elevation Plot*. Note that these terms are specified in the global XYZ axis system.

Note that the *Water Elevation Plot* operates in a synchronised manner with the relevant input data. For example, when any data pertaining to wave loading is altered in the keyword file, the plot updates automatically.

Flexcom presents a wide range of wave specification options. There are three regular wave options as follows:

- Regular Airy describes linear Airy wave theory.
- Stokes V describes Stokes V wave theory, and presents the various options for the calculation of vessel response.
- Dean's Stream describes Dean's Stream wave theory, and presents the various options for the calculation of vessel response.

There are five options for defining a random seastate in terms of a wave spectrum, as follows:

- Pierson-Moskowitz describes the Pierson-Moskowitz wave spectum model, and presents the various formats for input data specification.
- Jonswap describes the Jonswap wave spectum model, and presents the various formats for input data specification.

- Ochi-Hubble outlines the Ochi-Hubble wave spectum model.
- Torsethaugen outlines the Torsethaugen wave spectum model.
- User-defined describes a facility whereby you can input an arbitrary wave spectrum directly in terms of a series of data pairs.

Refer to Wave Loading for further details.

The *Water Elevation Plot* operates like all other plots in Flexcom. For further information refer to Plotting.

1.7.5.4 Vessel Response Plot

OVERVIEW

Flexcom provides the option to view the response of the vessel in the *Vessel Response Plot*. The response can be shown for any of the local Vessel Degrees of Freedom, depending on the option selected in the *Plot* drop-down list. By default, the response is presented at the vessel reference point, but any arbitrary point on the vessel (e.g. riser hang-off location, tip of crane arm) may also be selected.

If you are interested in examining the vessel response at some location other than the vessel reference point, you can explicitly specify an alternate location via the *X Offset*, *Y Offset* and *Z Offset* inputs in the *Vessel Response Plot*. These offsets are relative terms defined with respect to the vessel reference point, and are specified in the local vessel XYZ axis system.

Note that the *Vessel Response Plot* operates in a synchronised manner with the relevant input data. For example, when any data pertaining to vessel RAOs or wave loading is altered in the keyword file, the plot updates automatically.

A vessel may be defined using either one of two keywords, `*VESSEL` or `*VESSEL,INTEGRATED`. While the former is perfectly valid, it has effectively been superseded by the latter, which is now the recommended option. *VESSEL,INTEGRATED is a more comprehensive keyword and has several advantages over its predecessor. Most importantly it accepts RAO data also, thereby eliminating the need for a separate `*RAO` keyword – hence the ‘integrated’ nature of the new keyword. Further details are available in Basic Vessel Concepts.

The *Vessel Response Plot* operates like all other plots in Flexcom. For further information refer to Plotting.

1.7.6 Results Examination

Flexcom provides a variety of outputs in order to facilitate examination of finite element results. These include Structural Animations, with associated Stress Contouring, Graphical Output in the form of plot files, text-based Tabular Output, and Spreadsheet Output in the form of an Excel workbooks.

The relevant components of the Flexcom user interface are as follows.

- Plotting presents an overview of the *Plotting* facility, which allows you to examine Flexcom plot files (.MPLT file extension). In addition to standard features like zooming and panning different areas for detailed inspection, the *Plotting* facility also allows you to superimpose several plots on top of each other.

- Model View describes the structure display facility. This feature provides both a structure preview facility available during model building, and a way of viewing an animation of the structure response after an analysis has completed. The Model View provides a control system capable of allowing precise inspection of models, allowing you to move freely and examine any region of the model in detail.

- Element Inspector is a very useful utility feature which complements the Model View. It allows you to quickly select an element and examine its geometric, structural, stress and hydrodynamic properties where available as well as start/end nodes. This is very useful feature in terms of Quality Assurance, allowing models to be readily inspected by peer or managerial review.

- AVI Studio presents an overview of the dedicated video creation studio which accompanies the main software. The AVI studio allows you to create customised videos of your finite element models. Video files are an effective means of showcasing innovative designs, and serve as valuable promotional material for engineering teams.

- Unity Plug-In describes the Unity gaming engine plug-in which enables you to transfer results from a Flexcom database into the Unity environment, allowing you to create more advanced visualisations based on the motions derived from the finite element simulation.

1.7.6.1 Plotting

PLOT FILES

Plot files have the file extension .MPLT. You open, save and close plot files in the usual manner of Windows applications. Standard options are provided on the top menu bar and also via toolbar shortcuts.

PLOTTING TOOLBAR

The toolbar at the top of the Plotting facility provides useful options for customising plots, including exporting to Excel, manipulating title, axis and legend entries, viewing data tooltips etc. Refer to Plotting Toolbar for a full list of all the relevant commands.

ZOOMING

The quickest and easiest option for zoom control is to use the Mouse Scroll Wheel. Rolling the wheel upwards zooms in, while rolling downwards zooms out. Further options are also available for zooming individual axes in 2D Plots.

PANNING

In 2D plots, you can pan the view by holding down the Left Mouse Button, and then moving your mouse in the required direction.

In 3D plots, you can pan the view by holding down the Right Mouse Button, and then moving your mouse in the required direction.

ROTATING

In 3D plots, you can rotate the view by holding down the Left Mouse Button, and then moving your mouse in the required direction. Alternatively you can use the standard Rotate Buttons on the Plotting Toolbar. Rotating is not relevant for 2D plots.

SUPERIMPOSING PLOTS

Should you wish to superimpose two or more plots to facilitate comparisons, you may:

- Open two or more plots together by selecting multiple entries in the File Open dialog, in which case you will be presented with a superimpose option
- Open a single plot first, and then add other plots to it, by selecting the relevant name of another plot in the File View area and dragging it onto the Plotting area

Plotting Toolbar

The toolbar at the top of the Plotting facility provides useful options for customising plots.

Button	Function	2D	3D
	Export plot data to CSV format (Comma-Separated Values)	✓	✓
	Export plot data to Microsoft Excel	✓	✓
	Copy Plot Image	✓	✓
	Copy Series Data to Clipboard	✓	✓
	Edit Series. This button provides several options for customising the display of the data series. You may rename or delete series, adjust the colour and thickness of the lines, adjust the marker types etc.	✓	✓
	Control Chart Legend	✓	✓
	Edit Chart Title	✓	✓
	Edit Axis Titles	✓	✓
	Swap Axes	✓	✓
	Show/Hide Major Grid Lines	✓	✓
	Show/Hide Minor Grid Lines	✓	
	Show/Hide Crosshair	✓	✓

	Show/Hide Tooltips	✓	✓
	Zoom All	✓	✓
	Previous Plot File	✓	✓
	Next Plot File	✓	✓
	Alternate between 2D and 3D display	✓	✓
	Show/Hide surfaces on 3D plot (if available)		✓
	Perspective/Orthographic View		✓
	Front View		✓
	Side View		✓
	Top View		✓
	Positive Rotation about X		✓
	Negative Rotation about X		✓
	Positive Rotation about Y		✓
	Negative Rotation about Y		✓
	Positive Rotation about Z		✓
	Negative Rotation about Z		✓

Plotting Toolbar Functions

2D Plots

OVERVIEW

2D plots are the most common in Flexcom. One (dependent) variable is plotted against another (independent) variable in a 2-dimensional space. In the vast majority of cases, the independent variable (typically plotted on the horizontal axis) is one of the following:

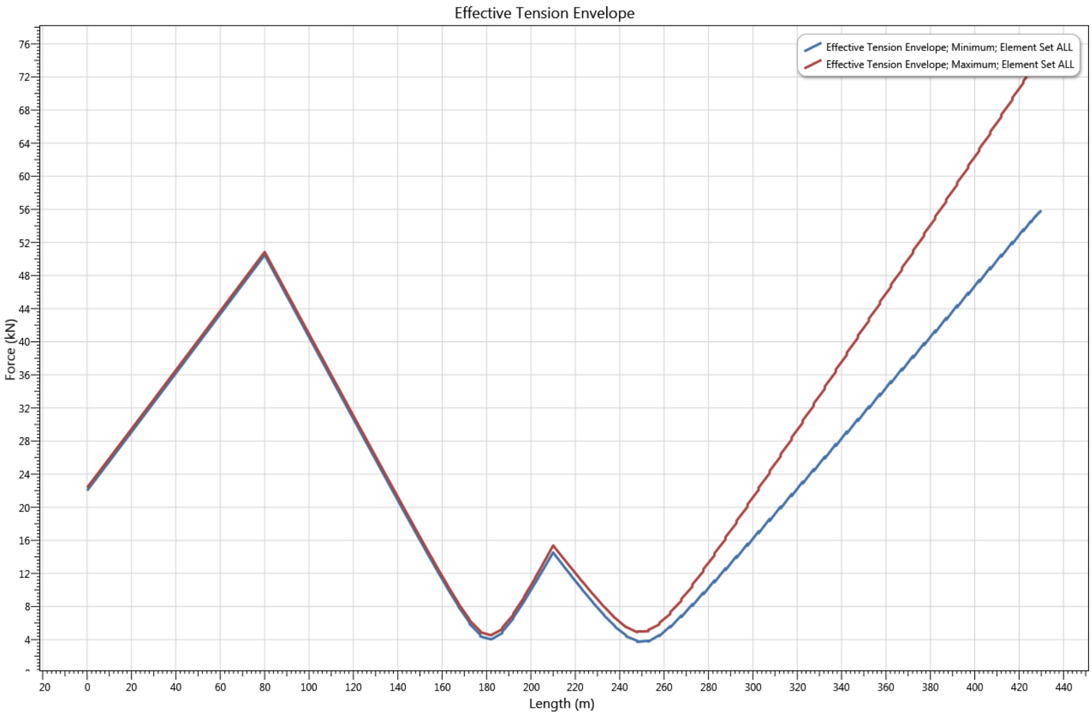
- Element set length (in metres or feet)
- Time (in seconds)
- Frequency (in Hertz)

2D plots are ideally suited to the type of output produced during Database Postprocessing and Summary Postprocessing, the two main sources of output data from Flexcom. Sample outputs include...

- Timetrace: a plot of the variation of a particular parameter as a function of time
- Snapshot: a plot of the spatial distribution of a particular parameter at a given time
- Statistics: a plot of the statistical variation of a particular parameter over the course of an analysis. Options include maximum/minimum envelopes, mean values, standard deviations and extreme values.
- Spectrum: a plot of the spectral distribution of a particular parameter

SAMPLE PLOT

The following is a statistical plot of effective tension (kN) plotted as a function of length along a structure (m).



Sample 2D Plot

The toolbar at the top of the Plotting facility provides useful options for customising plots, including exporting to Excel, manipulating title, axis and legend entries, viewing data tooltips etc. Refer to Plotting Toolbar for a full list of all the relevant commands.

ZOOMING

The quickest and easiest option for zoom control is to use the Mouse Scroll Wheel. Rolling the wheel upwards zooms in, while rolling downwards zooms out.

You can also hold the CTRL key while dragging the mouse, allowing you to select a particular zoom area. If you hold the SHIFT key while dragging the mouse, the zoomed area will maintain the same aspect ratio of the current plot.

An individual axes can be zoomed by hovering over the relevant axis and using the Mouse Scroll Wheel. In this case the other axis remains unchanged. An axis can be also stretched by holding the Right Mouse Button and moving the mouse (up and down for the vertical axis, left and right for the horizontal axis). The mouse cursor also indicates the direction to move the mouse in order to stretch or shrink the displayed range. The axes can also be moved by holding the Left Mouse Button and moving the mouse accordingly.

3D Plots

OVERVIEW

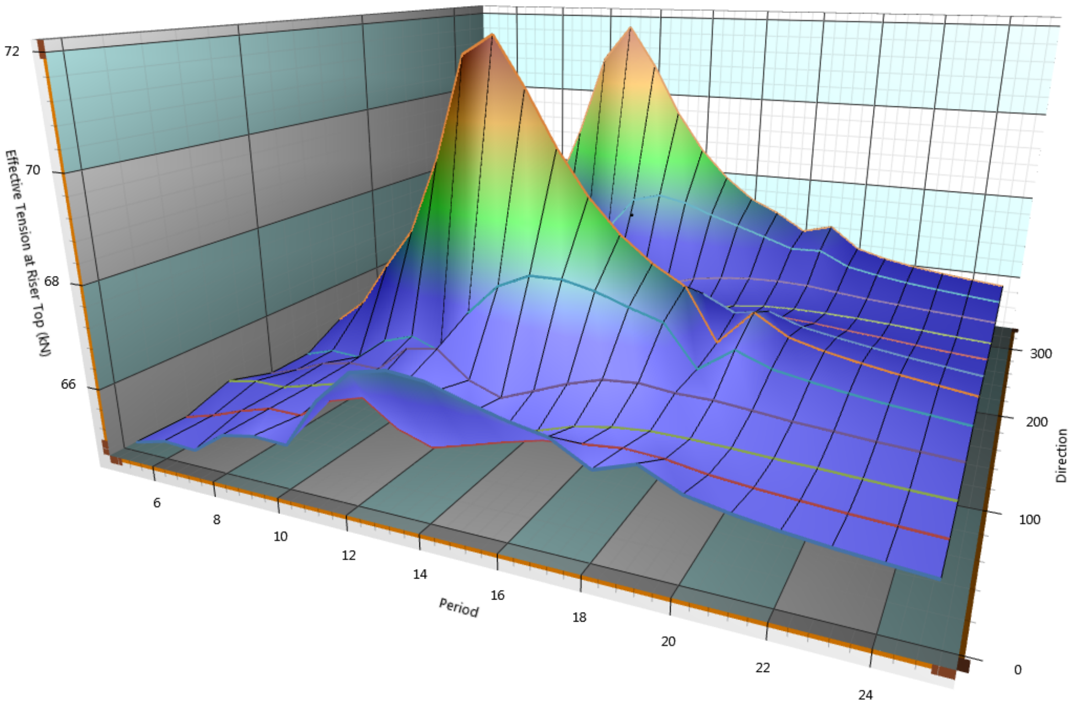
3D plotting is an advanced feature which provides enhanced visualisation of engineering data. One (dependent) variable is plotted against two other (independent) variables in a 3-dimensional space. The plot control is fully interactive, and you can pan, zoom and rotate the viewpoint for ease of inspection. Helpful tooltips allow you to retrieve exact data and any point in the 3D space. It's also possible to switch back to planar/2D views should you prefer.

3D plots are a relatively recent addition to Flexcom. These plots are ideally suited to the examination of vessel RAO data (refer to the Vessel Response Plot feature) but more importantly to summary postprocessing and collation. If you are performing a series of analyses (for example to examine a large number of different load cases), the summary collation facility provides a useful means of assembling all the pertinent summary data across a range of load cases into a single Summary Collation Spreadsheet. Enhanced data visualisation is also provided by the ability to produce 3-dimensional Summary Collation Plots. Here you can plot the variation of any summary postprocessing output against any key driving parameters. For example, you can plot maximum effective tension as a function of both wave period and incident wave heading in a 3-dimensional space.

The toolbar at the top of the Plotting facility provides useful options for customising plots, including exporting to Excel, manipulating title, axis and legend entries, viewing data tooltips etc. Refer to Plotting Toolbar for a full list of all the relevant commands.

SAMPLE PLOT

The following sample plot presents maximum effective tension (kN) as a function of both wave period (s) and incident wave heading (degrees).



Sample Summary Collation Plot

1.7.6.2 Model View

OVERVIEW

The Model View provides both a “live” structure preview facility available during model building, and a way of viewing an animation of the structure response after an analysis has completed. During model building, as you add new components to the model, the structure preview is continuously and automatically updated. As well as facilitating rotating, panning and zooming, it provides other useful features such as displaying node and element numbers, nodal coordinates, seabed topography, water surface profile, and many more. The Model View provides a control system capable of allowing precise inspection of models, allowing you to move freely and examine any region of the model in detail.

TOOLBARS

The figure below shows a typical Model View screen. The various options that are provided to manipulate the display are grouped into three toolbars:

- The View Toolbar at the top of the screen, above the structure display

- The Play Toolbar just below the structure display
- The Draw Toolbar, at the bottom of the screen

The various options each of these provides are described in the following sections.

COMBINING VISUAL ANIMATION WITH ENGINEERING DATA

Note also that if you are viewing a Model View animation and have a timetrace plot from the same analysis concurrently open or on display in the user interface, then a moving line on the plot will automatically show you the value of the variable plotted at each Model View time. Refer to Timetraces and Model View Animation for further information on this feature.

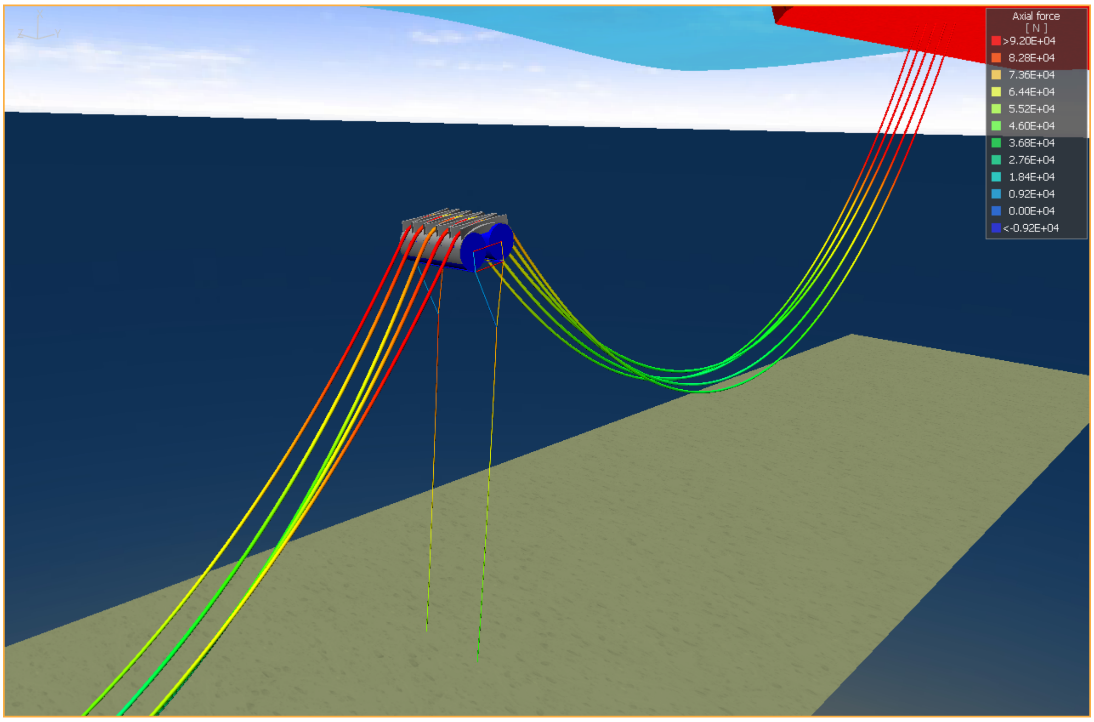
It's also possible to colour elements according to one of the program force or stress outputs, effectively producing a contour plot of the variable nominated. Refer to Control Element Colouring for further information.

ELEMENT INSPECTOR

The Element Inspector is a very useful utility feature which complements the Model View. It allows you to select an element and quickly examine its geometric, structural and hydrodynamic properties, and its internal fluid contents. It also presents derived properties, such as wet and dry weight, total displacement etc. This is very useful feature in terms of Quality Assurance, allowing models to be readily inspected by peer or managerial review.

TROUBLESHOOTING

The Model View Troubleshooting guide should be useful if you experience problems with the Model View (e.g. persistently blank Model View).



Model View

View Toolbar

The View toolbar has buttons for panning, rotating and zooming, as well as providing some standard views. The options, tabulated below, are mostly self-explanatory:

Button	Function	Keys
	Set Number and Layout of Panes	N/A
	Isometric View	Ctrl + I
	Plan View	Alt + ↑
	Elevation View	Alt + ↓

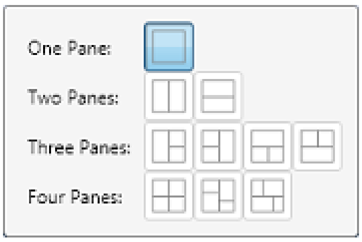
	Left Side View	Alt + ←
	Right Side View	Alt + →
	[Save and Restore Camera Positions](#)	N/A
	Zoom In	Ctrl + ↑
	Zoom Out	Ctrl + ↓
	Zoom All	Ctrl + A
	[Show Zoom Control Box](#)	N/A
	[Show Measuring Tape Control](#)	N/A
	Pan Up	↑
	Pan Down	↓
	Pan Left	←
	Pan Right	→
	Positive Rotation about X	Shift + ←
	Negative Rotation about X	Shift + →

	Positive Rotation about Y	Shift + ↑
	Negative Rotation about Y	Shift + ↓
	Positive Rotation about Z	Ctrl + ←
	Negative Rotation about Z	Ctrl + →
	Copy Screenshot	Ctrl + C

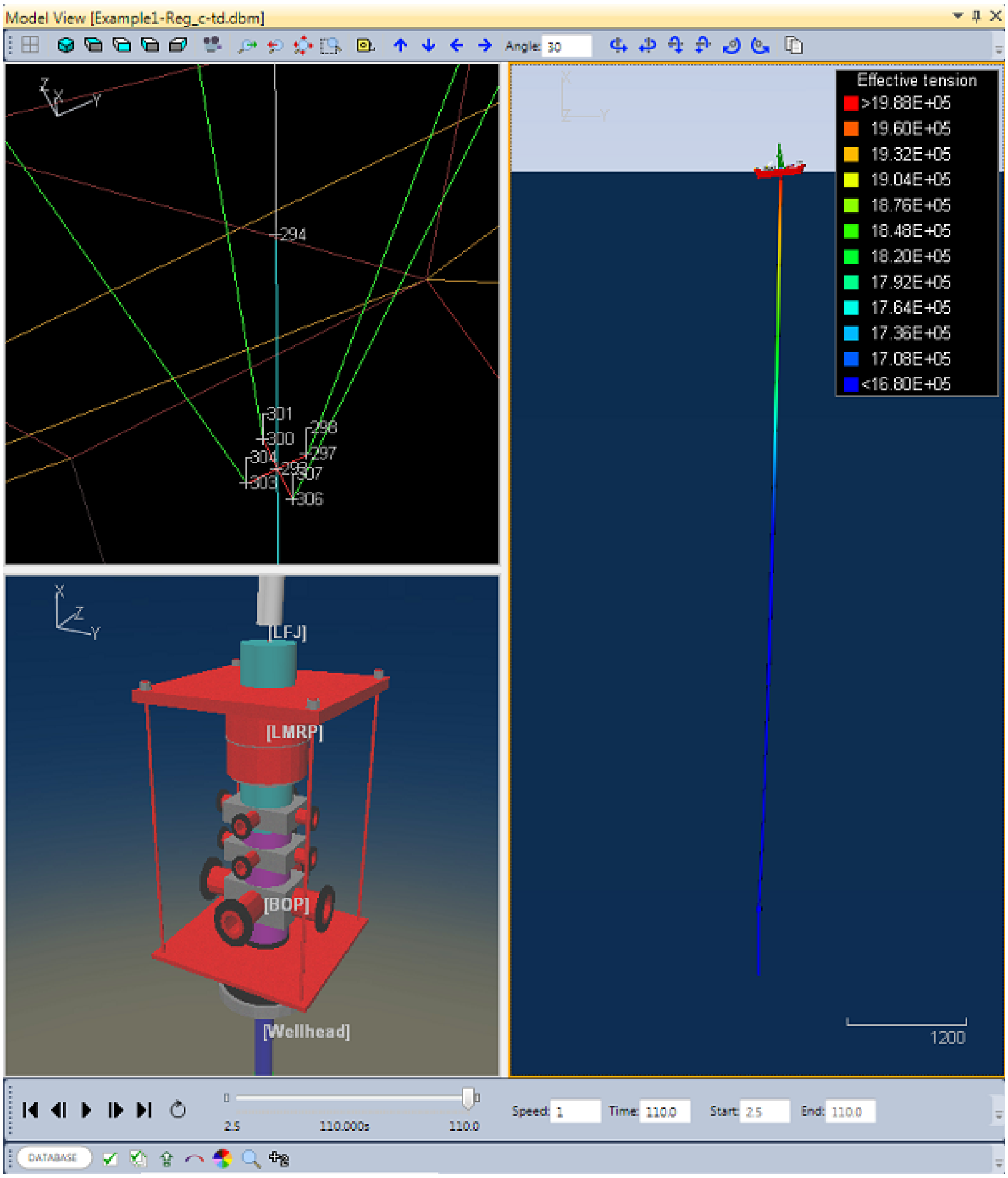
Note that the Rotation buttons relate to the screen X, Y and Z axes (as opposed to the Flexcom global axis system). Screen relative axes use the right hand rule and are defined as follows: Screen X axis is vertical (from bottom to top) along the screen, Screen Y axis is horizontal (from left to right) along the screen, and Screen Z points directly into the screen.

Set Number and Layout of Panes

The Model View window can be split into as many as four panes, each with an independent view. Press the button and select the desired layout from the pop-up:



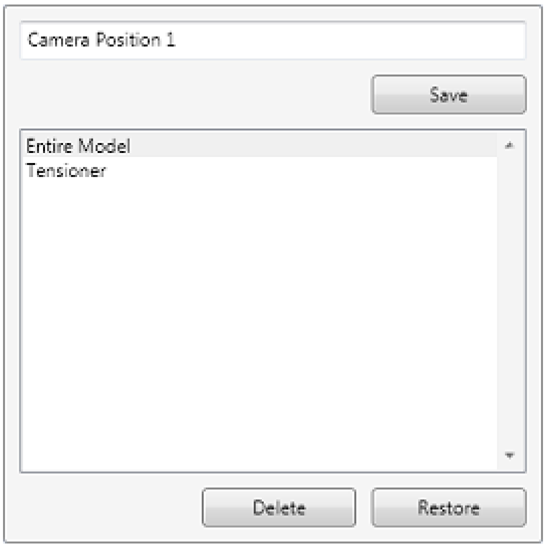
Up to four different views can be requested, in a variety of configurations. Views can be resized by left-clicking on the border between views and then dragging. Each view has an independent camera position and independent draw options.



When multiple views are visible, one is active. The active view is highlighted with an orange border. In the screenshot above for example, the view on the right is the active one. Many actions affect the active view only. You can make a view active by clicking on it.

Save and Restore Camera Positions

Useful camera positions can be saved and quickly restored. Saved camera positions are stored on a file-by-file basis. To save the current camera position, click on the button to get the following pop-up window:



Edit the suggested camera position name (e.g. “Camera Position 1” above) to a more meaningful value if desired and then press the Save button.

The list below the Save button shows camera positions that have already been saved. To restore a previously saved camera position, click on the camera button to open the pop-up. Select an entry from the list by left-clicking on it, and then press the Restore button. The camera is returned to the saved position. Alternatively, double-clicking on an entry in the list will also restore it.

Saved camera positions can also be deleted; left-click on an entry in the list to select it and then press the Delete button to remove it.

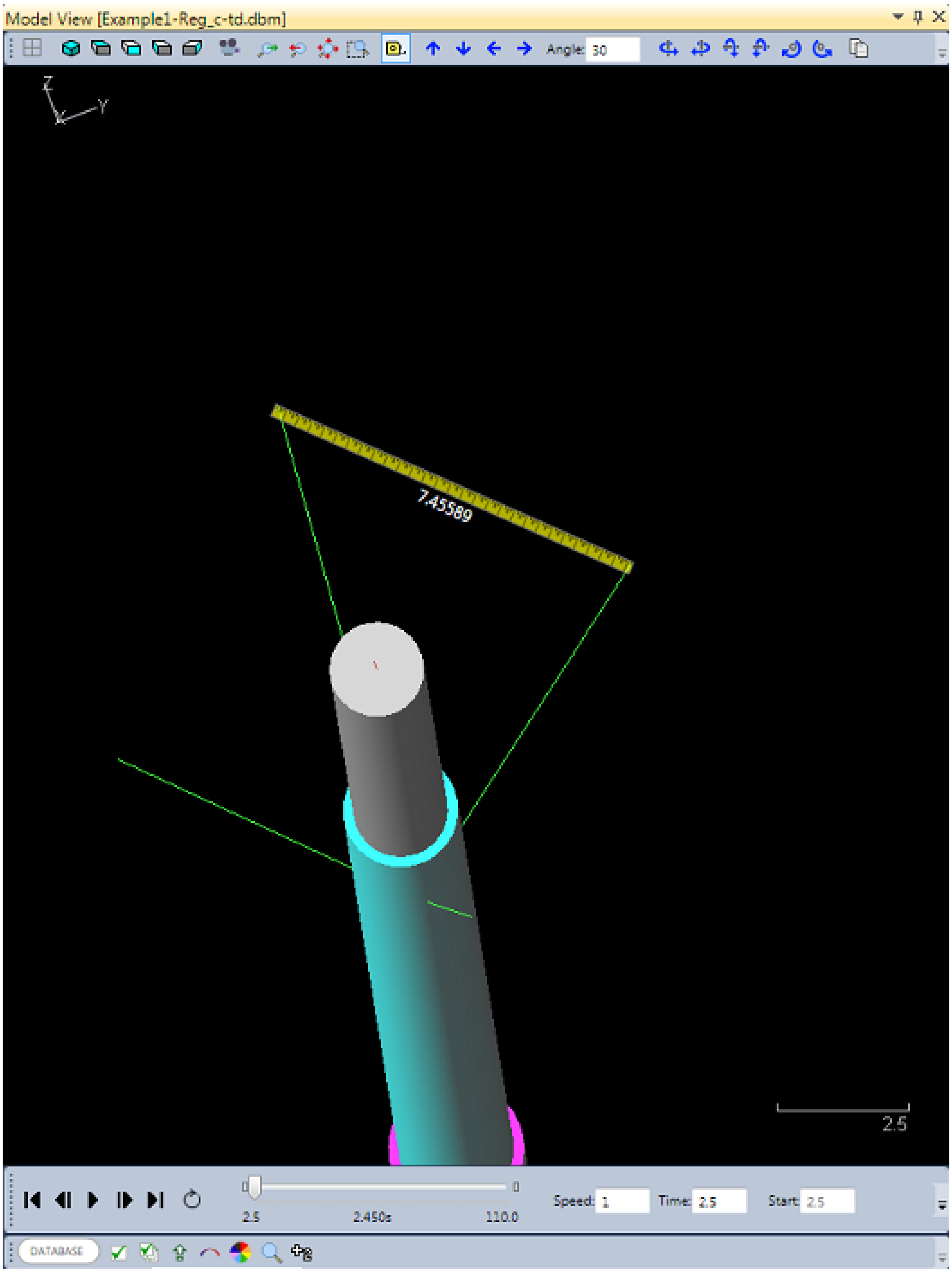
Show Zoom Control Box

The mouse buttons have two different control modes, Standard and Zoom. You use this option to toggle between them. In Standard mode, the left mouse button is used to rotate about the current focus point. This focus point is always located at the centre of the screen, and is marked by a red transparent sphere during panning and zooming operations. The right mouse button pans the model in the current viewing plane. The scroll wheel is used to zoom in and out on the current focus point. In Zoom mode, pressing the left mouse button zooms in at the current cursor position, and pressing the right mouse button zooms out at the current cursor position. Pressing and holding the left mouse button allows you to drag the cursor over an area which you wish to zoom.

Show Measuring Tape Control

Distances can be measured on the screen with the measuring tape control when not in a perspective view. To switch to an orthogonal (non-perspective) view, click one of the standard plan/elevation/right-side/left-side buttons or switch perspective off on the Draw Options pop-up.

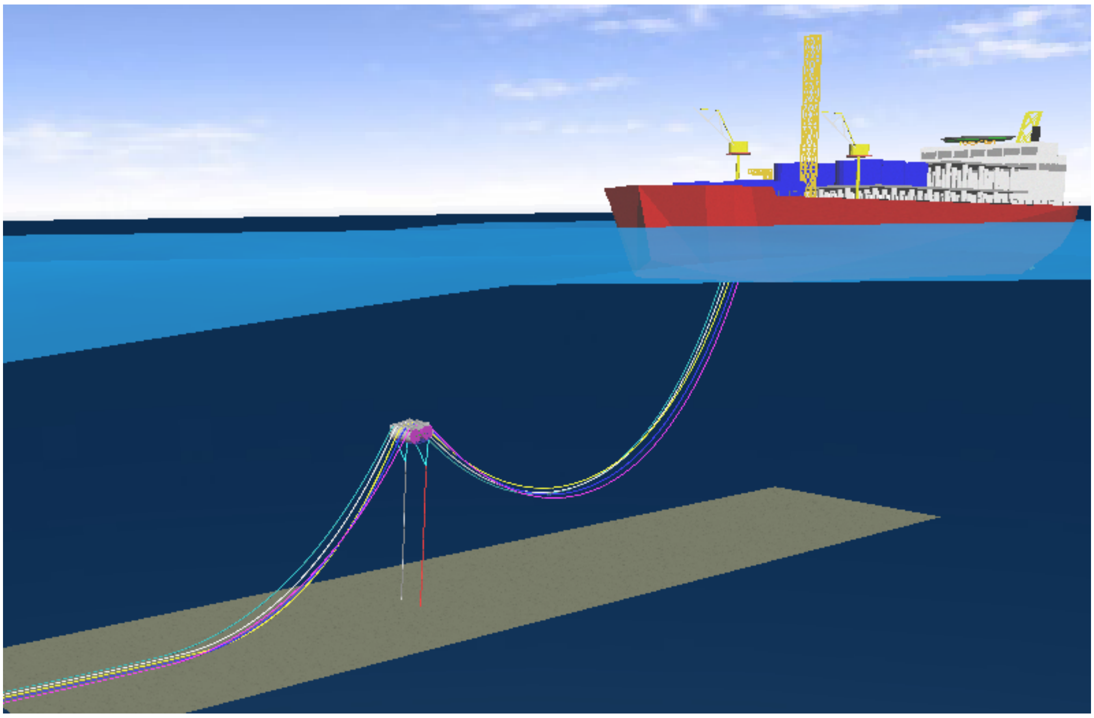
To use the measuring tape control, first click the button on the toolbar. Left-click on the model at the point from which you are measuring and, without releasing the mouse button, drag the mouse pointer to the second point. The measuring tape control shows distance in real-time as you drag it.



The accuracy of the measuring tape control is dependent on screen resolution and the accuracy with which you can control the mouse pointer. It should ideally be considered as an indicative measure of distance rather than an accurate one.

Play Toolbar

The Play toolbar is used to control playback of structural animations. To start an animation running, press the ▶ button. As the animation proceeds, the current animation time is continually updated, and the Progress Bar progresses in synchronisation with the animation. If you want to suspend the animation at any time, press the ⏸ button. The ▶ button will then become enabled, and you can use it to resume the animation. To go back directly to the beginning of an animation, press the ⏮ button. You can then play the animation again, from the start. To jump directly to the end of an animation, press the ⏭ button. Note that you may also jump to any point within the animation by manually dragging the Progress Bar to a desired location. While an animation is suspended, you can resume in frame-by-frame play mode. This is done by pressing the ⏵ or ⏴ buttons. The animation plays forward or backward one step, each time either of these operations is carried out. The ⟳ button can be used to specify that the animation is to start again automatically when the end time is reached (by default the animation terminates once the end is reached).



Sample Structural Animation

You can also vary the animation display Speed. This is a multiple of the actual speed of the simulation time. A display speed of 1 corresponds to real time, which means that 1 second of simulation time is displayed in 1 second of real time. Flexcom automatically relates real time and animation speed using the computer clock, so that the animation speed is machine-independent. If necessary, it skips some animation frames to maintain the correct speed. Consequently, if you set the speed too high for your computer's performance, animations may appear 'jumpy'.

Display Toolbar

The Display toolbar provides a range of useful display options under various headings as follows:

- Source Indicator DATABASE
- Draw Options
    - Tags
    - Vessel
    - Model
    - Environment
    - Drawing
- Scale Nodal Displacements
- Show/Hide Element Sets
- Control Element Colouring
- Find Node, Element or Panel
- Highlight Found Node, Element or Panel

Source Indicator

The source indicator confirms where the data that is being displayed is coming from. The options are Keyword (keyword file), Database (actual finite element solution) and Cable (cable pre-static step). For example, the following indicator shows 'Database' for illustration... 

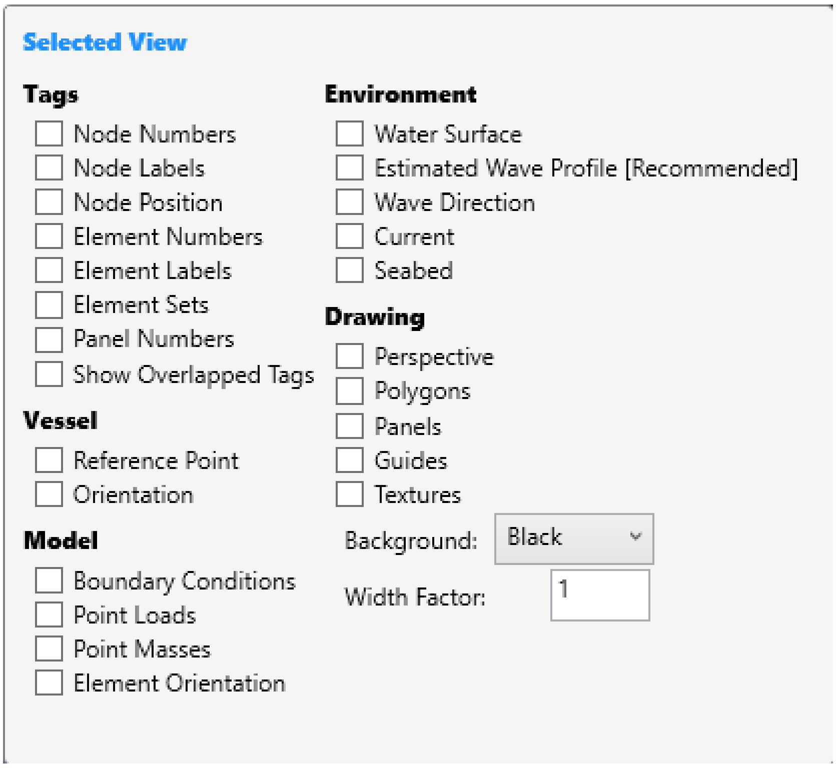
.

In Keyword mode, the data is coming directly from the keyword file you are working on, meaning you effectively have a model preview facility. Database on the other hand means you are looking at a converged solution. Very occasionally Cable mode is displayed, meaning you are looking at the Cable Pre-Static Step solution. If an initial static analysis fails to converge, it may be that the cable solution is giving a poor approximation to the actual static configuration; viewing the cable solution in that case may help identify modelling shortcomings.

Draw Options

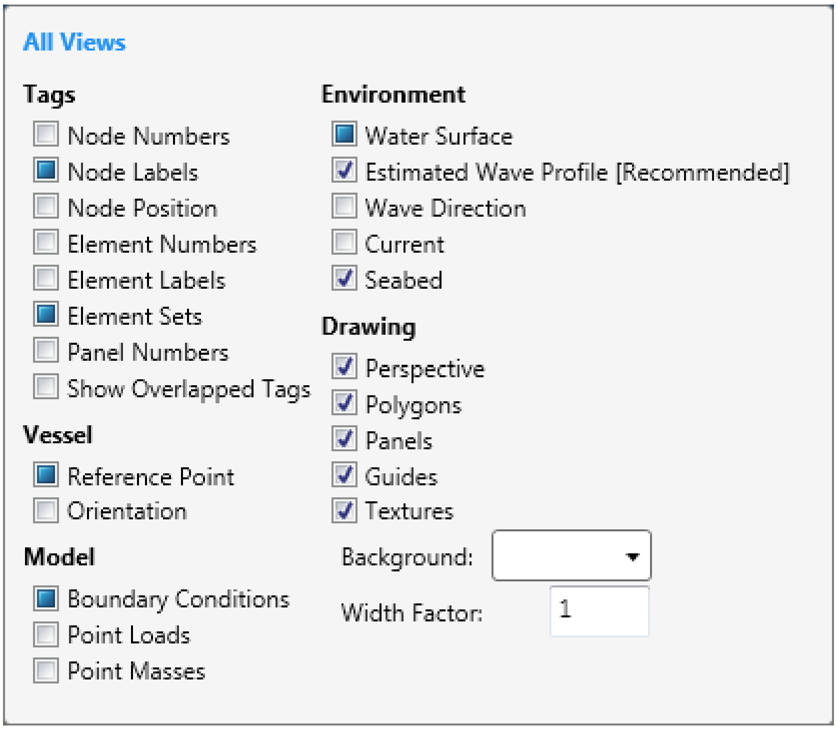
ACTIVE VIEW

Draw Options () groups together a range of options for toggling on or off various aspects of the display via the following pop-up:



ALL VIEWS

The Draw Options - All Views ( ) pop-up is almost identical to the previous one, but allows options to be set for all views and not just the active one.



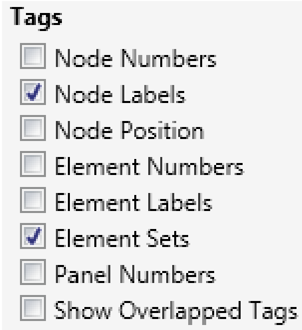
An empty or ticked check-box indicates that the respective option is off or on respectively in all views. If the check-box shows a box instead, it means that, for this option, different views have different settings.

SUB-CATEGORIES

Most of the options are self-explanatory, but are relevant features are outlined in the following sub-sections:

- Tags
- Vessel
- Model
- Environment
- Drawing

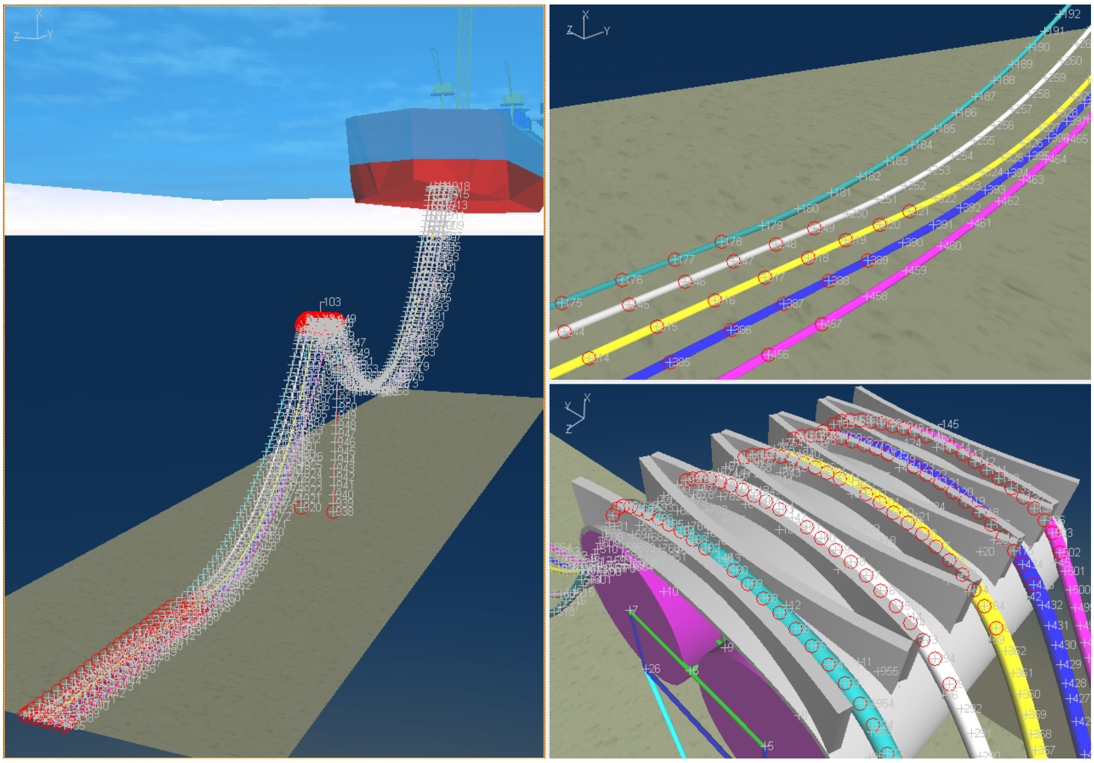
The options under the Tags heading are as follows:



The Tags pop-up allows you to toggle on or off the display of:

- Node Numbers. As well as illustrating node numbers, this option also highlights Contact Nodes.
- Node Labels
- Node Position (shows the instantaneous node X, Y and Z co-ordinates)
- Element Numbers
- Element Labels
- Element Set Names
- Panel Numbers
- Show Overlapped Tags. By default, where the text from two or more tags overlaps on screen, Flexcom suppresses the display of sufficient text to make the display legible. In some cases you may want to stop this behaviour, and instruct the program to show all overlapping text. You use Show Overlapped Tags to do this. An example might be where you want to display node numbers in a pipe-in-pipe model – by default only one nodal tag will be displayed where nodes on the inner and outer sections are coincident or in close proximity.

When you switch on node numbers, all nodes are shown using a + sign with the relevant node number alongside it. Contact nodes are also highlighted by a red circle around the + sign. Contact can occur with the seabed or guide surfaces (both illustrated in image below), or via pipe-in-pipe connections.

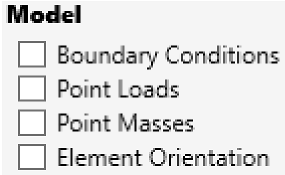


Contact Nodes

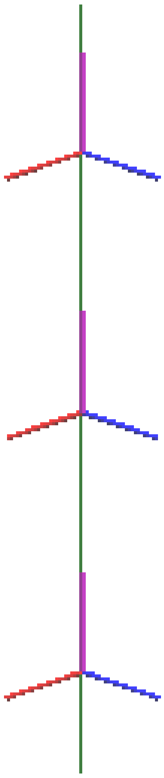
The options under the Vessel heading are as follows:

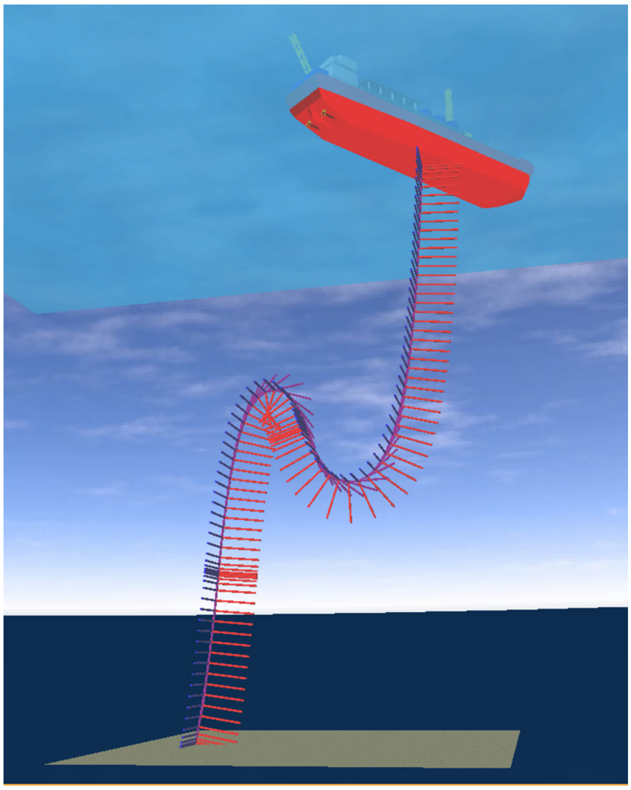
- Reference Point (indicated by a red circle at the instantaneous location of the vessel reference point)
- Orientation (indicated by a red arrow located at the reference point)

The options under the Model heading are as follows:

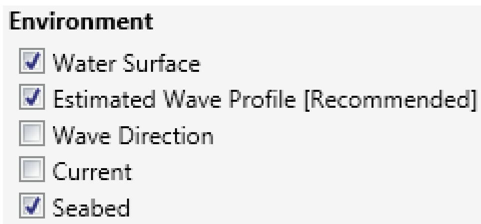


- Boundary Conditions (indicated by a green pyramid centred on the constrained node)
- Point Loads (indicated by a sphere centred on the relevant node, with an arrow showing the load direction)
- Point Masses (indicated by a cube centred on the relevant node)
- Element Orientation
    - This feature displays the local convected axes for each element. Flexcom uses a convected coordinate axes technique for modelling finite rotations in three dimensions. Each element of the finite element discretisation has a convected axis system associated with it, which moves with the element as it displaces in space. Refer to Finite Element Formulation for further details.
    - Key: **Purple** is local x-axis, **Red** is local-y axis, **Blue** is local-z axis.



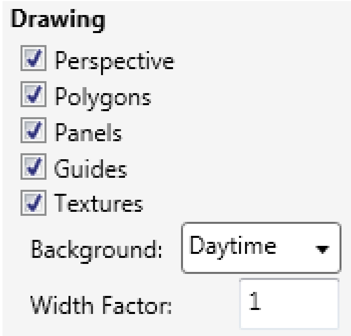


The options under the Environment heading are as follows:



- Water Surface
- Estimated Wave Profile (uses a reduced number of wave harmonics for a more efficient display)
- Wave Direction (indicated by a blue arrow at a corner of the wave surface)
- Current
- Seabed

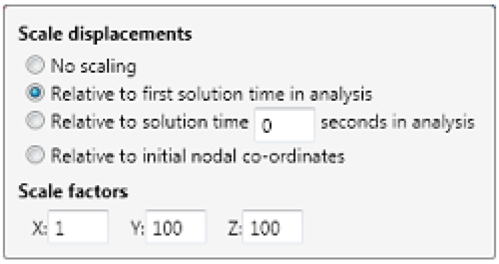
The options under the Drawing heading are as follows:



- Perspective
- Polygons (Polygons are used to display beam elements as cylinders - or strictly speaking octagonal prisms)
- Panels (Auxiliary panels)
- Guides
- Textures (Texture mapping is used to enhance the realism of surfaces, by mapping appropriate textures over surfaces for example the seabed, a wave)
- Background Scheme (The options are Daytime, Sunset, Navy, Black and White)
- Width Factor (Given that risers tend to be relatively long slender structures, this option allows you to scale the element diameters to facilitate visual inspection, particularly when colouring elements based on force outputs such as effective tension and bending moment)

Scale Nodal Displacements

In many of the simulations you perform with Flexcom, the displacements will be many orders of magnitude less than the major model dimensions (this is particularly true in deep water). This option ( ) allows you to scale or magnify displacements in each of the global X, Y and Z directions, to give an animation that is more meaningful or useful (because displacements are more visible). A typical implementation is shown here:

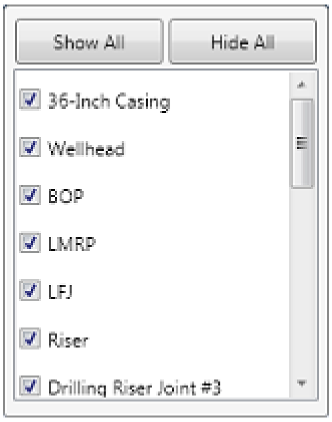


By default, No scaling is applied. You can though specify scale factors that are applied to the displacements, and the various options allow you to specify what exactly you mean by displacement – motion from the first solution time, from a subsequent solution time, or from the initial nodal co-ordinates (which may be nominal for structures comprised of cables or catenary sections). In the example shown above, displacements from the configuration at the first solution time in both horizontal directions have a scale factor of 100 applied.

When you scale displacements in this way, the text Nodal Displacements Scaled is displayed at bottom left of the screen for information.

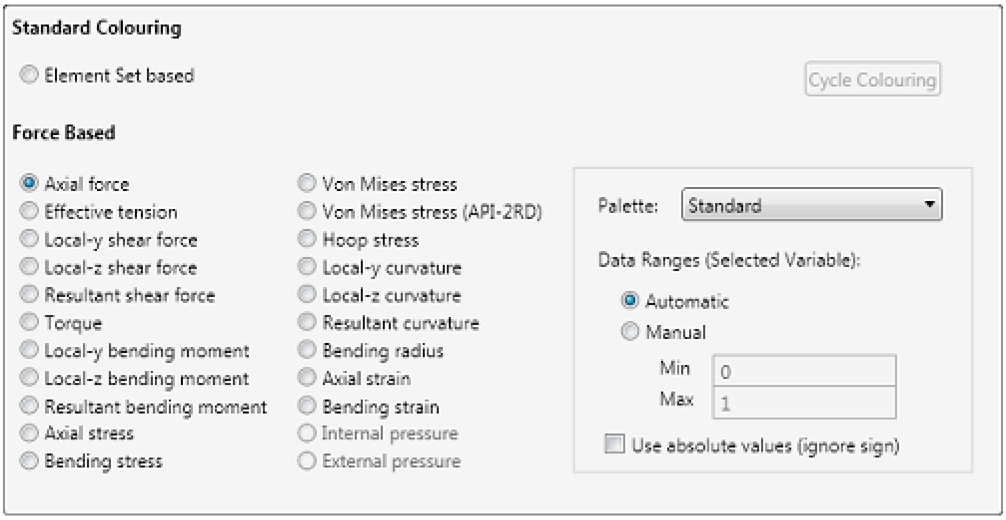
Show/Hide Element Sets

The Show/Hide Element Sets option ( ) provides the capability to view particular element sets rather than a full model. This allows you for example to focus in a complex model on a area of interest which might otherwise be obscured. A typical menu is shown below: the operation is self-explanatory.



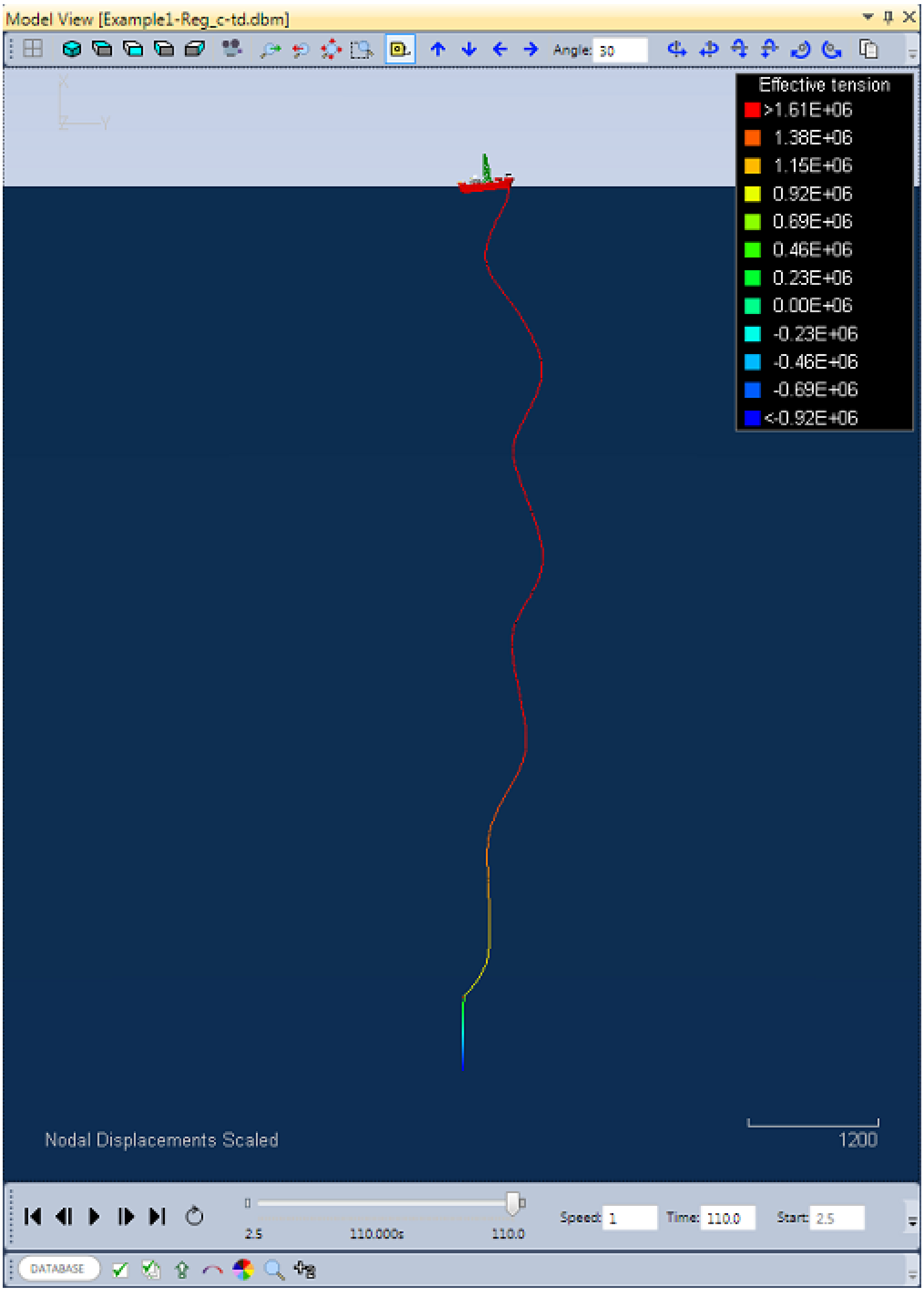
Control Element Colouring

By default, elements in a Model View display are colour-coded by element set, element set in this context meaning the set an element is part for the specification of geometric properties. You can though also colour elements according to one of the program force or stress outputs, effectively producing a contour plot of the variable nominated. The Control Element Colouring option ( ) is used to invoke this feature.



Control Element Colouring Inputs

This screen grab shows a drilling riser colour-coded by effective tension – note that horizontal displacements have also been magnified for visual effect.



Sample Stress Contouring

By default, the data ranges are based on the maximum and minimum values experienced over the course of the finite element simulation. Occasionally, the default ranges may be unsuitable. For example...

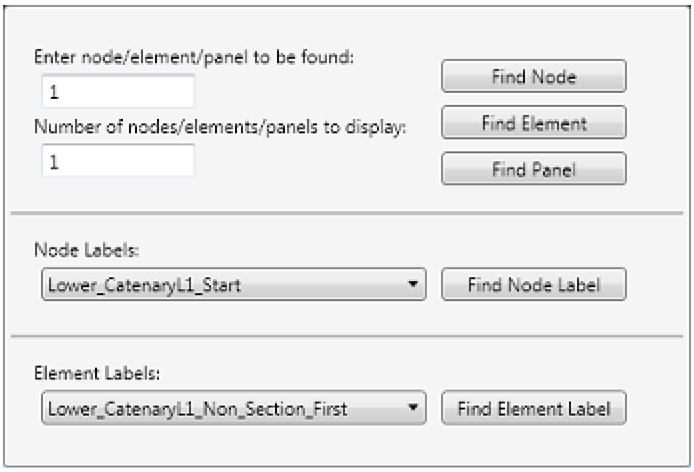
- If the model contains some rigid elements of extreme stiffness, then the bending moment ranges may be dominated by these elements, to the extent that the remainder of the model appears to experience very low levels of bending by comparison (i.e. the entire model may appear blue). In such circumstances, the Show/Hide Element Sets feature may be used to hide unwanted elements from the display, and the data ranges will automatically update to reflect the visible elements only.

- Some users may prefer not to display models which contain a lot of red colouring, in case the engineering results (e.g. effective or bending moment) are perceived as being excessive. If this is the case, the *Data Ranges* option may be switched from *Automatic* to *Manual*, and the *Max* value increased sufficiently high such that the most critical region of the model appears orange rather than red.

Find Node, Element or Panel

The Find Node, Element or Panel option ( ) allows you to locate any node, element or auxiliary panel in the model. If found, the relevant number is displayed on the model. There is also an option to display a specified number of adjacent numbers.

It is also possible to search for node and element labels. The majority of model building in Flexcom is performed using Lines, which provide an automatic mesh creation facility to greatly expedite the model creation process. Node and element labels are automatically created as part of this process - refer to Line Start and End Locations for further details.

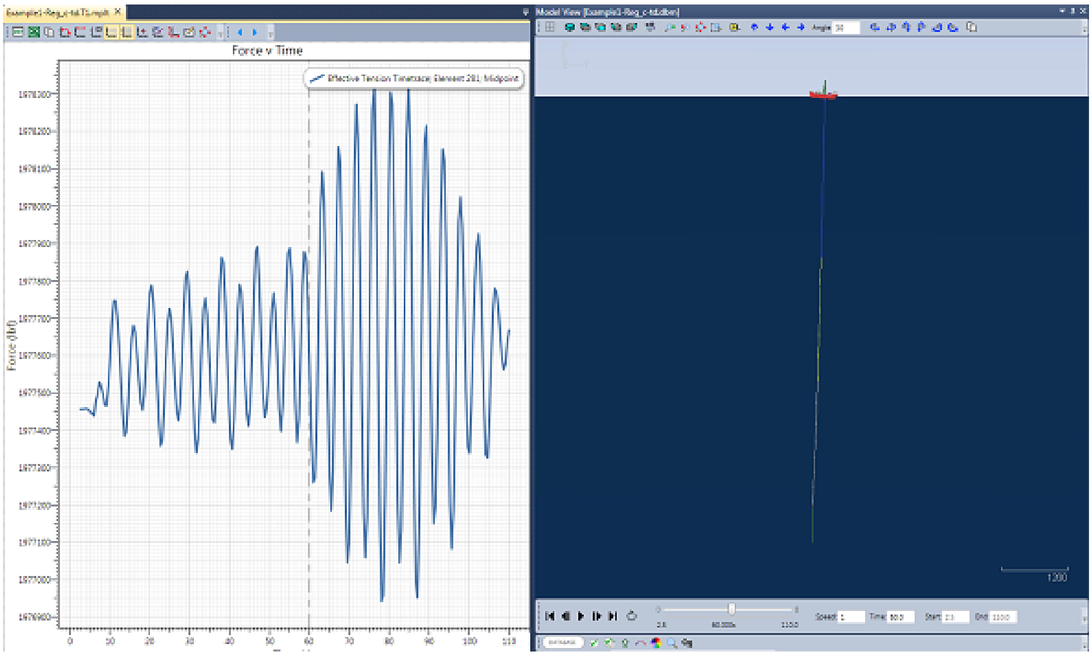


Highlight Found Node, Element or Panel

When the Find Node, Element or Panel option has located a specified object, its number is displayed on the model. The Highlight option () is used to toggle the highlighted number(s) on or off.

Timetraces and Model View Animation

If you are viewing a Model View animation and have a timetrace plot from the same analysis concurrently open or on display in the user interface, then a moving line on the plot will automatically show you the value of the variable plotted at each Model View time. This screen grab illustrates the operation:



The Model View is currently displaying the solution at 108.81s; the vertical line on the timetrace picks out the value of the effective tension at the midpoint of Element 281 at that time. You even have the option to ‘move’ the animation and the line on the plot to a specific time, using the following sequence of steps:

- Pause the animation

- On the plot, move the cursor to the time that you want to select
- Simultaneously press Alt and the mouse left button; the line will move to the time you selected and the structure configuration at that time will be displayed in the Model View

This feature can be very useful for example if you are trying to understand what is actually happening in a structure (or part thereof) when a key parameter is taking on a particular value. You can move the animation to the time of that value, and then use the Model View to examine the structure response at that time in detail.

Model View Troubleshooting

The following troubleshooting guide should be useful if you experience problems with the Model View. The facility works perfectly for the vast majority of users, but some users may experience a blank Model View which is incapable of displaying anything.

If at any stage you attempt a suggested corrective action and it does not help, repeat the guide from Step 1.

1. Does this problem occur with all models, including standard examples? If yes, go to Step 6, otherwise proceed to Step 2.
2. Can this model be opened in the Flexcom Database Viewer? This package may be launched from within Flexcom itself (via the Tools menu) or from the Windows Start Menu (under the Flexcom sub-section). If yes, try running Flexcom with default settings from the Windows Start Menu. If no, proceed to Step 3.
3. Was this analysis run with a later version of Flexcom than the one which you are currently using? There is a possibility that the database format may not be compatible. If yes, try re-running the analysis in this version of Flexcom. If no, proceed to Step 4.
4. Is this a Flexcom 8.2.3 analysis which also contains Point Masses? Flexcom 8.2.3 experiences problems when the number of point masses is an integer multiple of eight. If yes, either add some token point masses (with a mass of zero), or re-run the analysis in a later version of the software. If no, proceed to Step 5.

5. Is this a large model with tens of thousands of nodes/elements? If yes, and you have a 32-bit version of Flexcom installed, try installing the 64-bit version of Flexcom to see if that helps. If no, or you already have a 64-bit version of Flexcom installed, please Contact Wood with a copy of your model file.

6. Are you accessing Flexcom via Remote Desktop or Terminal Services? If yes, Flexcom uses DirectX which is not generally compatible with Remote Desktop or Terminal Services. To view the Model View, you must open your models on a computer to which you are directly logged-in. If no, proceed to Step 7.

7. Can you open any models in the Flexcom Database Viewer? If yes, try running Flexcom with default settings from the Windows Start Menu. If no, proceed to Step 8.

8. Try running the DirectX Diagnostic Tool. Search for the program “DxDiag.exe” on your machine and run it. This program is typically located in your Windows system folder or one of its sub-folders. The easiest way to find it is to use the “Search Programs” option which is located immediately above the Windows Start Button after pressing it. The DirectX Diagnostic Tool checks your DirectX and display driver installation. Once it’s running, click on the tab titled “Display 1”. The entry “Direct3D Acceleration” should read “Enabled”. If it does not, upgrade your display drivers (search your computer manufacturer's website for appropriate downloads for your system) and then reinstall DirectX from the Flexcom installation. If it does, proceed to Step 9.

9. Are you running on a laptop and does the Model View work when on the battery but not when charging, or vice versa? If yes, some laptops have dual graphics adapters (one which is used in a high-performance, high-power, setting and one which is used in a low-performance, low-power, setting) and you may be experiencing a problem with one of these adapters. It is assumed that you have already ensured that you are running the most up-to-date display drivers in Step 8. It may be possible to alter settings that will configure Flexcom to use one adapter exclusively. Consult your computer vendor for instructions. If no, Contact Wood.

1.7.6.3 Element Inspector

OVERVIEW

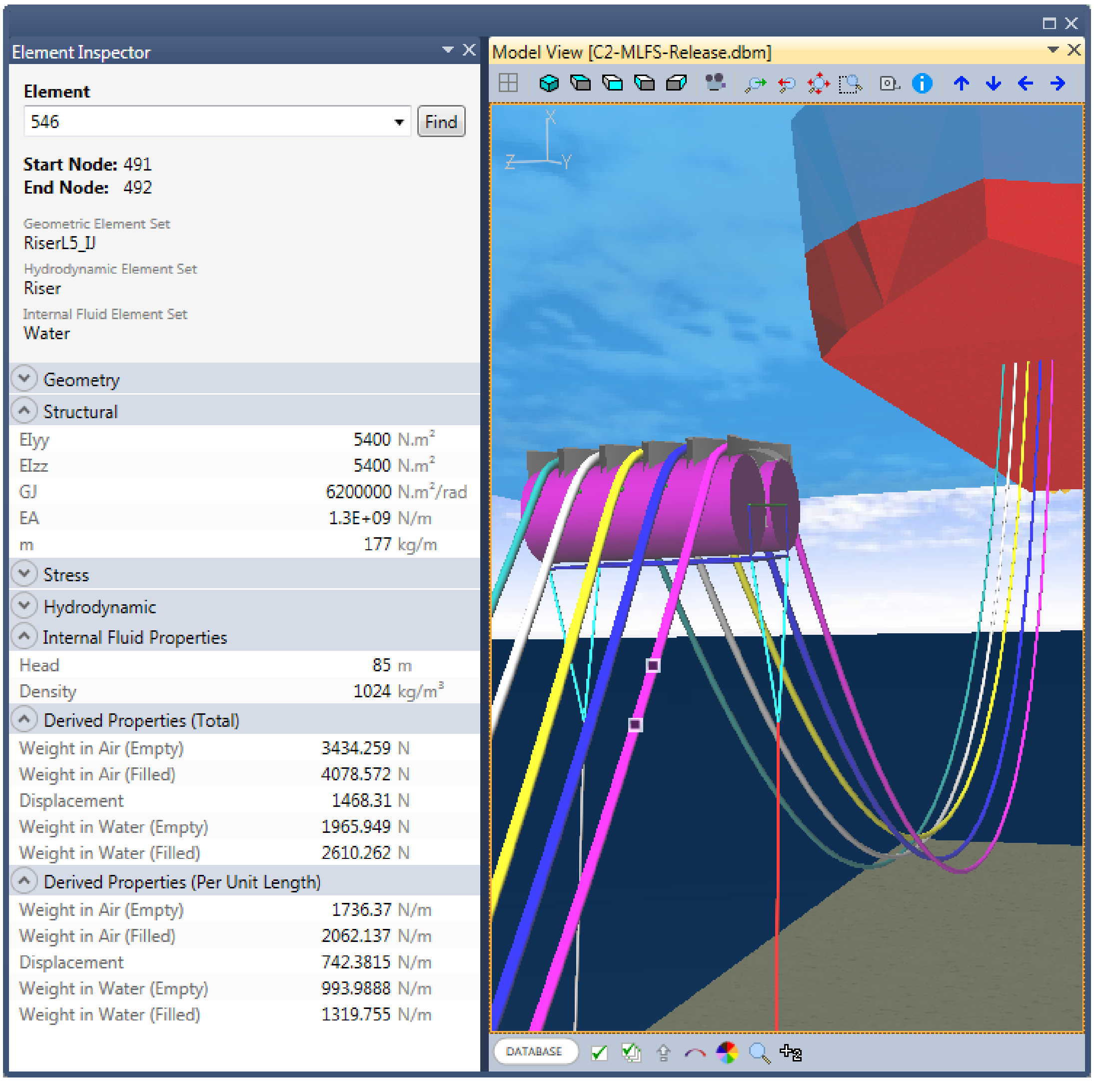
The Element Inspector is a very useful utility feature which complements the Model View. It allows you to select an element and quickly examine its geometric, structural and hydrodynamic properties, and its internal fluid contents. It also presents derived properties, such as wet and dry weight, total displacement etc. This is very useful feature in terms of Quality Assurance, allowing models to be readily inspected by peer or managerial review.

ELEMENT SELECTION

There are a number of ways to select your preferred element:

- **Model View** – Hold down the CTRL key and click on any element in the Model View
- **Element Label** – Begin typing an element label or select from the dropdown list of element labels found in the model.
- **Element Number** – Type any element number in the search box and press “Enter” or click “Find”.

The select element is highlighted on screen by means of two square boxes centred on its end nodes, as shown below.

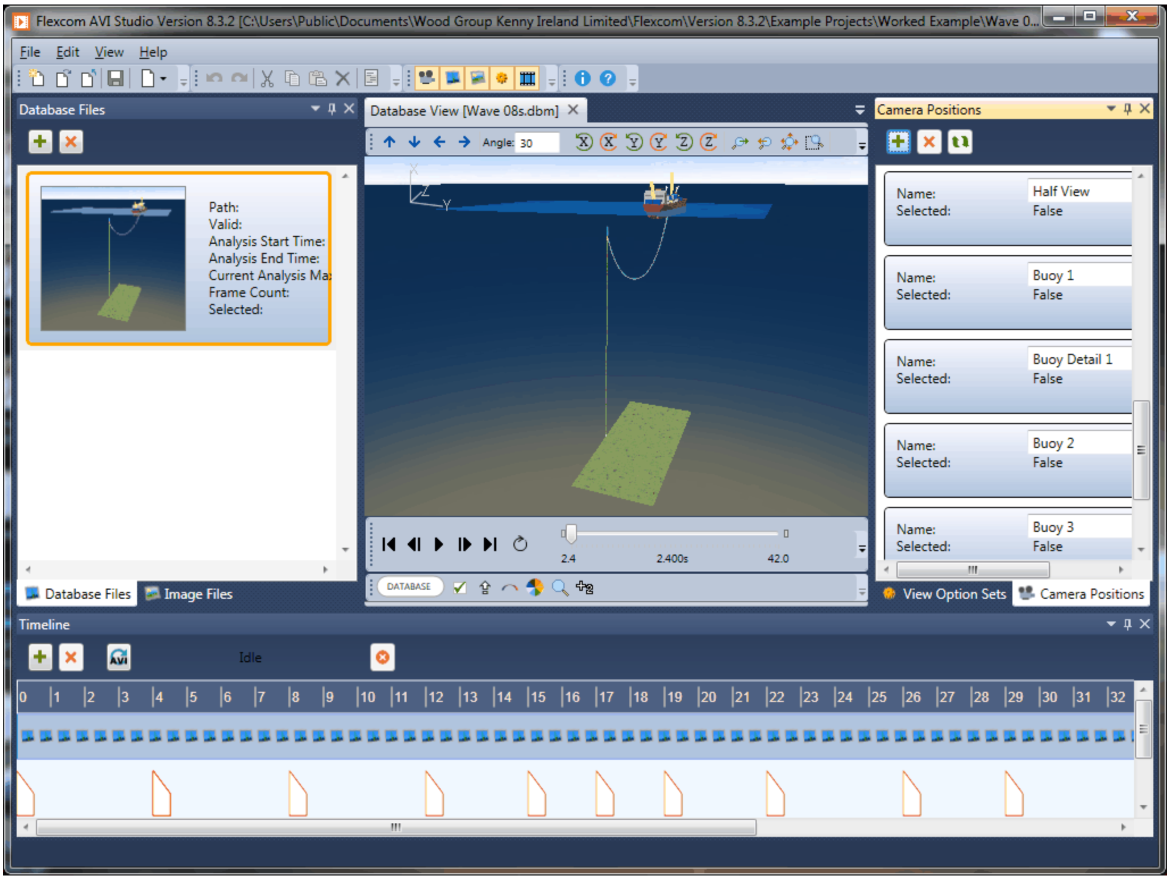


Element Inspector

1.7.6.4 AVI Studio

This section presents a brief overview of the AVI Studio program which accompanies Flexcom. This feature allows you to create videos of Flexcom structural animations, and store them in AVI format (Audio Video Interleave is a multimedia storage format developed by Microsoft). It's possible to combine several different Flexcom animations in a single AVI file, and you may even include external images.

The AVI Studio program is quite intuitive to use, and once you have mastered the basic functions, you should be able to create reasonably complex videos with relative ease. A sample layout for the AVI Studio is shown below. This includes the following windows (listed in order of appearance, from top left towards bottom right): *Database Files*, *Preview*, *Camera Positions* and *Timeline*. Also included but currently hidden are the *Image Files* and *Option Sets* windows.



Note that all the various components are dockable windows. Each individual panel may be minimised, maximised, floating, docked etc. This allows you to customise the entire layout to suit your own individual preferences.

The *Top Menu Bar* contains standard functions for loading and saving files, invoking online help, etc. The Toolbars provide shortcuts to the most commonly invoked menu options.

The first step is to create a new AVI project, using the *New* command on the *File* menu. The project file allows you to store a range of inputs and settings, from which you can generate an AVI file at any stage. Typically you then select a Flexcom database file whose structural response forms the basis of the video, assemble a list of suitable viewpoints and arrange these into a suitable timeline.

Creating a Video

STRUCTURAL RESPONSE

The most basic input is the specification of a Flexcom motion database file (DBM file extension), from which the structural display is sourced. Use the button to add a database file to the *Database Files* view. Naturally the database file must exist, so it is necessary to have successfully performed a Flexcom analysis in advance. A video is typically based on a single database file, but you may add further database files if you wish. You may also remove database files subsequently from the list using the button.

Once you select a database file in the Database Files view, it is loaded in the Preview window where it may be previewed. This view contains all the functionality of the standard Model View in the main Flexcom program. It also allows you to view an animation of the structure response, and also facilitates rotating, panning, and zooming. It also has other useful display features such node and element numbering, nodal coordinates, seabed topography, water surface profile, and many more.

Should you wish to augment the eventual video with some external captions/pictures, it is also possible to load a range of image files into the project. This is handled by the Image Files view, and again the and buttons may be used to add or remove images. Once you select an image file, it is loaded to facilitate a preview.

STORING CAMERA POSITIONS

Typically you will formulate a series of desirable views while examining the structural response over time, and you may store each viewpoint as a dedicated Camera Position. In a similar vein to the *Database Files* view, the and buttons are used to compile a list of *Camera Positions* in the *Camera Positions* view. Once you select an item in the *Camera Positions* view, it is loaded in the *Preview* window. If you are not entirely satisfied with the *Camera Position*, you may alter it by panning or zooming, and then storing the updated viewpoint using the button.

Notes:

a) If successive Camera Positions are different i.e. depict different locations, the AVI editor will move between the viewpoints when it comes to generating the AVI file. If you wish to keep a single viewpoint for a certain period of time, place the same Camera Position at the start and finish time of this interval.

b) A mathematical algorithm is used to move the camera between two Camera Positions. On occasion, due to the nature of this interpolation the images created between successive Camera Positions may appear distorted. This distortion can be prevented by creating and inserting more intermediate Camera Positions into the timeline to smooth the transition.

ADDING DISPLAY OPTIONS

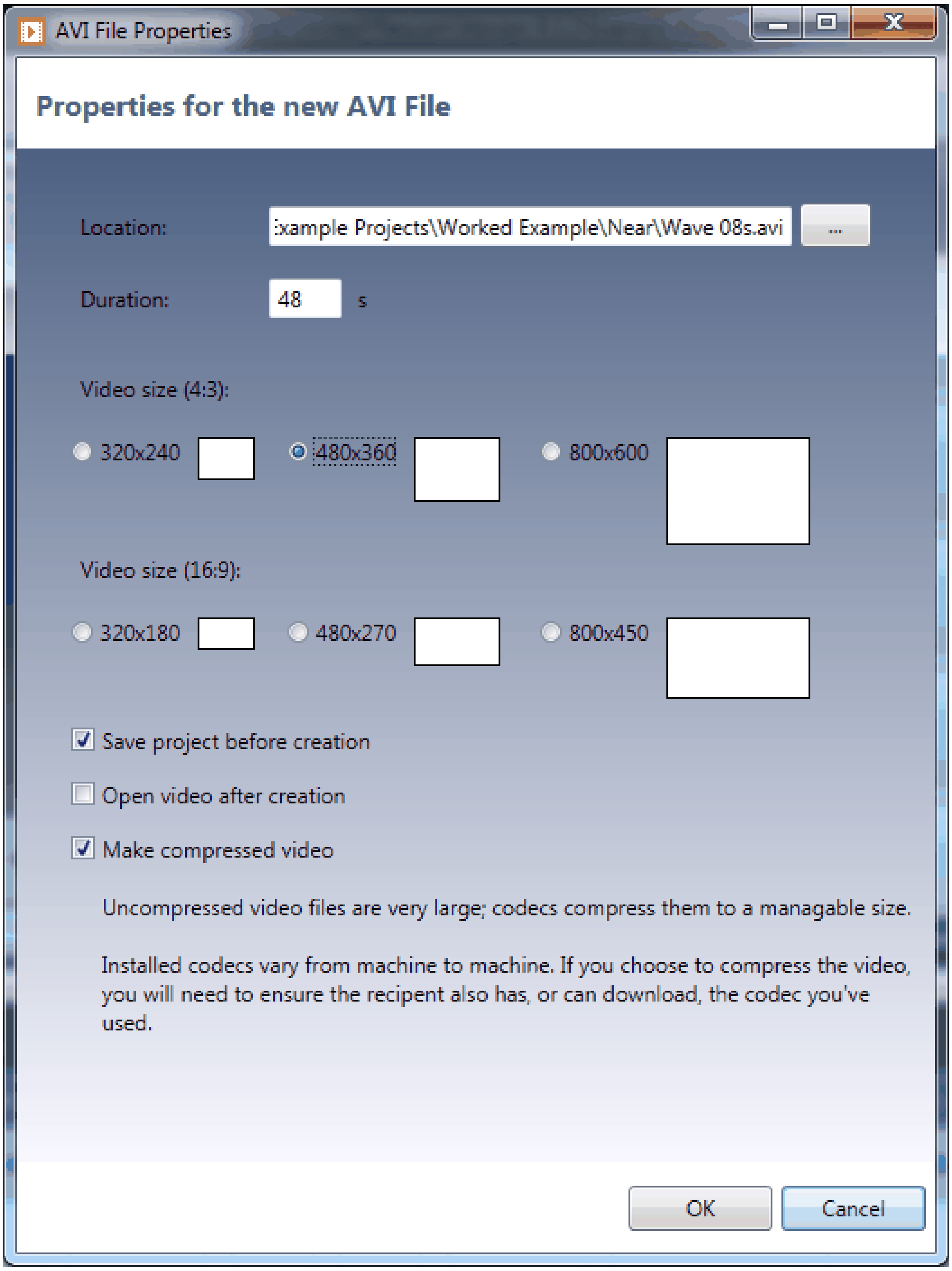
Additional information may be added to the display in the Preview window, such as the inclusion of node numbers, labels or coordinates, selectively viewing certain element sets only, or even changing the background colour. Collectively, each of these additional features may be described as display options, and different configurations may be created and stored in the View Options Sets window. As before, options sets may be added, removed or updated via the , and buttons, respectively.

CREATING THE TIMELINE

Once you have assembled all the required database/image files, camera positions and option sets, you can now compile them into a series of events which form the basis of the final video. Events may be added or removed via the and buttons in the Timeline View. The addition of a visual element (such as a database, image file or blank screen) is a mandatory first step, for which you specify the start time and duration of the image display. In the case of a database file, you must also specify the desired playback speed, and the simulation time at which the display is to commence from. Pre-defined camera positions or display options may also be initiated at various times in the sequence of events.

Generating an AVI File

Once you are happy with the series of events in the timeline, you may generate the video by pressing the button. Depending on the level of complexity incorporated into your video, this can take anything from a few seconds to several minutes, and the generation process may be cancelled at any stage using the button. Before the generation process commences, you are required to select some option for the AVI file, including the file name, duration (which defaults to the *Timeline* length), video size etc. In order to ensure the size of the resulting AVI file is reasonable, video compression is recommended, and you may select from a range of standard codecs.


AVI File Properties
Properties for the new AVI File
Location:
:xample Projects\Worked Example\Near\Wave 08s.avi
...
Duration:
48
s
Video size (4:3):
320x240
480x360
800x600
Video size (16:9):
320x180
480x270
800x450
Save project before creation
Open video after creation
Make compressed video
Uncompressed video files are very large; codecs compress them to a managable size.
Installed codecs vary from machine to machine. If you choose to compress the video, you will need to ensure the recipent also has, or can download, the codec you've used.
OK
Cancel

“Codec” is short for “coder/decoder”. Codecs represent different video formats and compress video files to reduce their size. Usually, higher compression implies lower quality video. Requesting uncompressed video produces an AVI file of the highest quality, the animation looks exactly as it would in Flexcom itself. However these files tend to be very large indeed and unsuitable for applications where size is a concern, such as emailing.

The exact list of codecs available varies from machine to machine and you may like to experiment to find out which codec provides a satisfactory trade off between image quality and file size. Bear in mind that whichever codec you choose, the recipient of your AVI file will also need the same one installed on their machine. Thankfully, this is often not a great concern as many modern video playing packages, will automatically download and install codecs on-demand.

The Xvid codec is recommended for a good balance between the image quality and file size. Note however that if you choose Xvid, you are advised to disable the *Encoding Status* display, as this can cause problems for the AVI Studio program. You’ll find this option in the *Xvid Configuration* settings, as illustrated by the following images. If you do not already have the Xvid codec installed on your machine, it may be downloaded from the Downloads Section of the Xvid website.

1.7.6.5 Unity Plug-In

OVERVIEW

This section presents an overview of the Unity Plug-In application which accompanies Flexcom. Unity is a gaming engine which is quite popular amongst video game developers. Given its advanced graphical technology, you can create more advanced visualisations in Unity than is possible with Flexcom’s standard Model View component. The plug-in allows you to transfer results from a Flexcom database into the Unity environment, which means that the enhanced display is based on the motions derived from the finite element simulation. This feature may be used to create promotional videos which highlight innovative device concepts or service offerings. Such videos are also VR-ready, so computational models can be experienced a fully-immersive virtual reality environment.

Unity is a third party software which is licensed separately from Flexcom. In order to use it, you would need to possess an active Unity license, and also some graphical design skills in order to maximise its full potential. As many engineering firms may not have graphical designers in house, Wood are happy to create videos for individual customers as a service offering. Please contact out technical support team if you are interested in further details.

FURTHER INFORMATION

Further information on the Unity Plug-In are contained in the following sub-sections.

- Visualisation illustrates the powerful graphical capabilities of Unity, by comparing some promotional videos created using Unity with corresponding videos created using Flexcom's video creation package, the AVI Studio.
- Interactive Demonstration allows to explore a Unity-generated model in more detail, and also offers you the opportunity to experience it in 3D virtual reality.
- Tutorial provides some useful information about model building within Unity.

Visualisation

ADVANTAGES

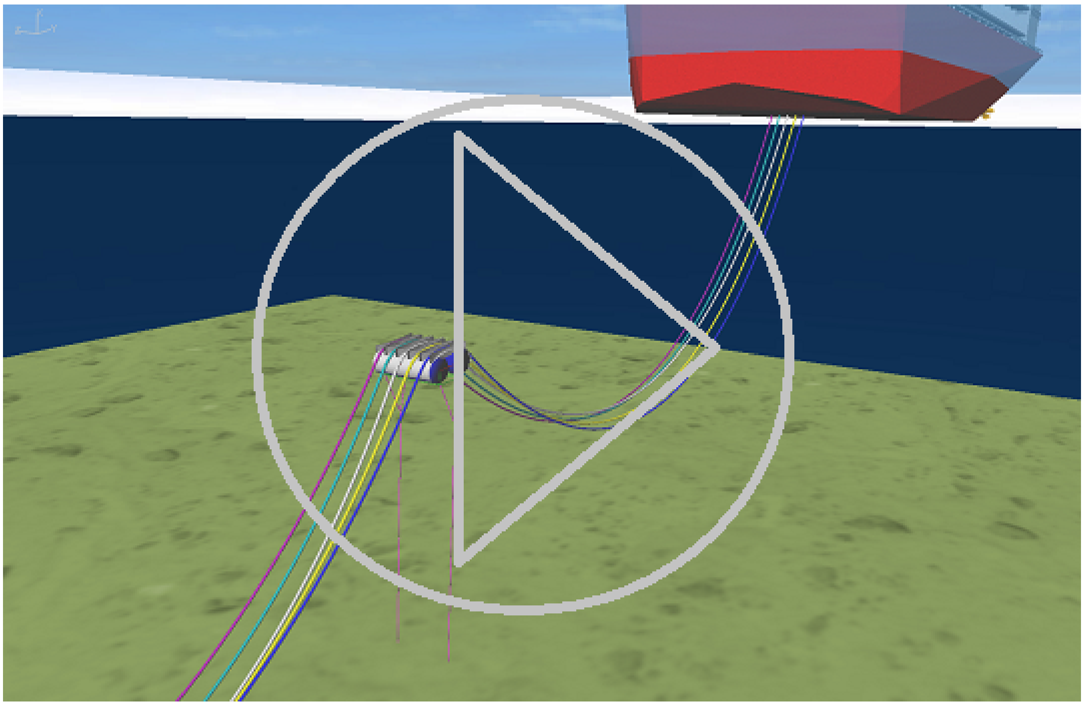
Unity is a powerful gaming engine which is capable of creating more advanced visualisations than Flexcom's standard Model View. The main advantages are as follows:

- Unity is capable of delivering photo-realistic graphics, whereas Flexcom's visualisation is essentially an animation. The comparison is analogous to watching a CGI movie (Computer Generated Imagery) versus an animated/cartoon style film. Naturally, such high definition videos can serve as very useful promotional material for engineering companies.

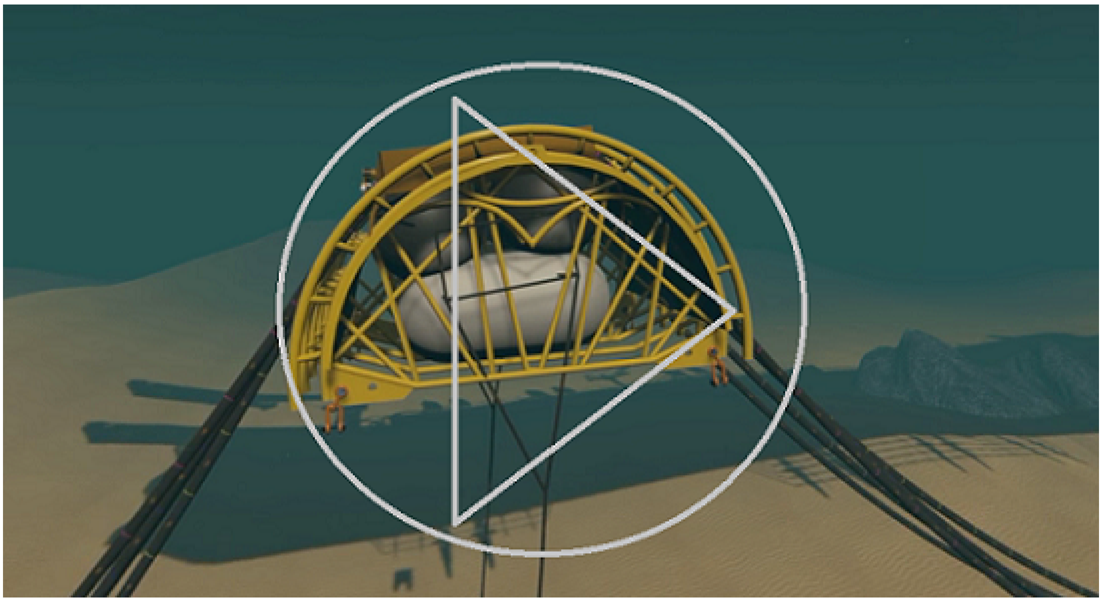
- Unity videos are compatible with virtual reality, offering the viewer a fully immersive 3D viewing experience. Experiencing a subsea model in 3D space provides an excellent sense of scale and perspective which is simply not possible from a flat screen monitor. Feedback from engineering teams suggests that this can be particularly important for subsea layouts where a number of components are operating in close proximity – in this case the spatial awareness afforded by VR can enhance engineering insight.

EXAMPLES

In order to highlight these benefits, let's compare and contrast a sample model viewed in Flexcom and in Unity. The model contains 5 flexible risers, which pass over a tethered mid-water arch in a lazy-S configuration, and is typical of the shallow water production systems deployed in North Sea field developments. You will notice that the Unity version offers much greater image sharpness, on the vessel and the mid-water arch, and the water surface definition looks incredibly realistic.



[Flexible Riser System - Flexcom]



Flexible Riser System - Unity

Interactive Demonstration

VIEWING A UNITY MODEL ON SCREEN

The Unity video presented in the Visualisation section followed a pre-determined fly-by pattern which was created by a graphic designer at Wood. If you would like to explore the Unity environment in further detail yourself, you can view the Unity Web Demo. This will allow you to pan, rotate and zoom within the Unity environment, so you can explore different aspects of the model at your leisure.

Most modern internet browsers will support the Unity Web Demo. We recommend using Chrome or Firefox as Unity WebGL content is not currently supported by Internet Explorer.

The viewing experience with Flexcom illustrations created by Unity is seamless. However, as this is an online demonstration which is viewed via a web browser, the response of the viewing controls may appear a little sluggish on your computer. This is purely a performance limitation associated with the browser and internet connection speeds, and is not a true reflection of the capabilities of Unity or the virtual reality experience.

EXPERIENCE A UNITY MODEL IN 3D VIRTUAL REALITY

Virtual reality environments are normally provided by means of a specialised headset, which is supported by a high performance computer. However you can also experience a similar, albeit slightly lower spec version, using a standard smartphone and a cardboard headset. Open one of the links below on your smartphone using YouTube, set the view mode to 'Cardboard', set the quality to 'High Definition', and then pop on the cardboard headset. Once you're in the VR environment, you can adjust your viewing direction by physically turning around or looking up or down. The main limitation is that you cannot fly to different locations within the 3D space – your position in space is effectively static – which is why we have created two separate locations for you to explore, one above the water surface and one subsea.

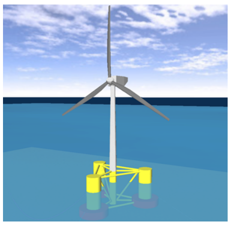
- Unity 360 Video - Vessel
- Unity 360 Video - Subsea

Tutorial

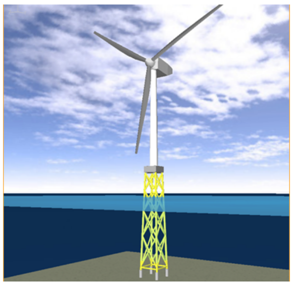
As mentioned in the Overview section, many engineering firms may not have the necessary experience in house to use Unity and so may prefer to ask Wood to create a customised video as a service. However if you prefer to work directly with Unity itself, this tutorial will get you up and running with the Unity Plug-In which accompanies Flexcom, and provide some useful information about model building within Unity.

Rather than providing detailed written instructions, we have instead created a helpful Tutorial Video for you to view online. The tutorial is self-explanatory, but feel free to Contact Us for further technical support if required.

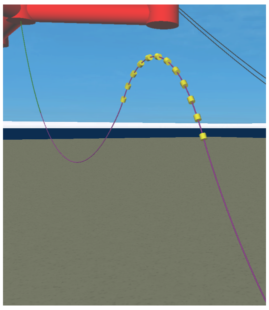
1.7.7 Flexcom Wind



Flexcom Wind is Wood's dedicated simulator package for offshore wind turbines. Guided by an independent technical advisory group, real-world feedback from key players in the offshore wind industry has helped to shape this new software product. Our design philosophy is based on the provision of advanced computational techniques to provide confidence in the engineering design, coupled with a user-friendly interface which facilitates optimum productivity.



State-of-the-art aerodynamic modelling is delivered via a fully-integrated coupling with the industry-recognised OpenFAST software. Structural modelling accuracy is ensured by Flexcom's industry-proven finite element formulation. The software has been fully validated using benchmarking exercises with other industry leading software. Flexcom Wind comes at an affordable cost structure and very flexible licensing options.



BENEFITS OF USING FLEXCOM WIND

- **Confidence**: State-of-the-art aerodynamic and structural models delivered via a fully-integrated coupling between Flexcom and OpenFAST. Technical papers validating the numerical solution have been published at industry conferences including IOWTC, ISOPE & SPE Offshore Europe.

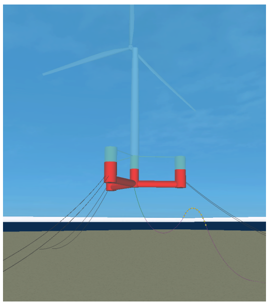
- **One stop solution**: Global assessment of floating platform, tower & turbine, mooring system and dynamic power cable. Suitable for all stages of device design, from conceptual prototypes right through to full scale versions.

- **Quality assurance**: Mathematical equations are fully supported within Flexcom, allowing users to develop and validate template models, streamlining the QA process

- **Advanced visualisation**: Powerful visual aids such as colour contouring of stresses provide the user with an intuitive visual representation of engineering data

- **Trust and reliability**: Backed up by Wood's renowned software support service. World leading consultancy services also available through our wider Wood capability

- **Cost effective solution**: Affordable cost structure and very flexible licensing options. Plus our user interface does not require a licence, maximising the number of licences available for numerical simulation.



USER EXPERIENCE

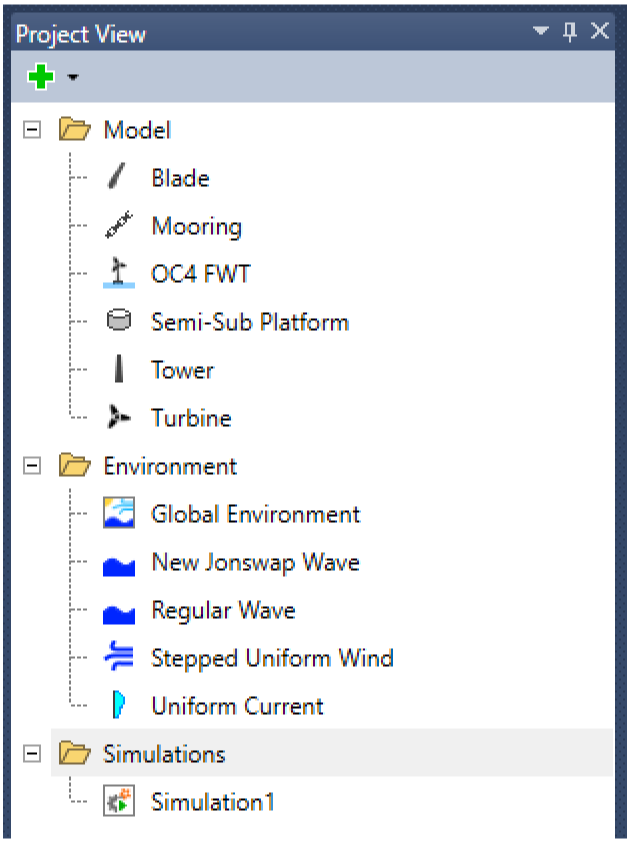
- 'Engineering tool' design philosophy and intuitive user interface facilitates quick and easy software familiarisation
- Detailed models may be created quickly and easily, allowing the user to concentrate on the engineering design
- Productivity-enhancing features including highly automated analyses with multiple load cases and automatic results generation
- Streamlined post-processing presents the designer with key structural outputs such as platform motions, mooring line tensions & anchor loads, nacelle accelerations and tower bending moments, plus all the standard aerodynamic parameters such as generator power, rotor speed and blade pitch.
- Enhanced visualisation of engineering data such as stress contouring and 3D surface plotting
- Direct interface to Microsoft Excel, with optional access to VBA code which enables power users to develop specialised post-processing tools
- Dedicated video creation studio which allows users to create customised videos of models and showcase innovative engineering designs. Creation of fully immersive virtual reality experience available as an optional add-on.

1.7.7.1 Project View

The Project View acts as a container for all of the components in your project, and provides a good overview of your project workspace. It has a tree-like structure and is divided into separate sub-folders for convenience.

- Model folder contains Devices and Device Parts.
  - Devices are the offshore wind turbines in your project workspace. You can assemble a particular device using any of the sub-components which you have created.
  - Device Parts are sub-components of the offshore wind turbine, such as turbines, towers, platforms, mooring lines etc. You can define as many sub-components as you like, even if you do not intend to use all of them immediately. So you can build up a library for both current and future projects.
- Environment folder stores a range of different environmental conditions which the offshore wind turbine may be designed to operate in.
- Simulations folder allows you to perform numerical simulations and examine analysis results.

The Project View acts as a container for all of the components in your project, and provides a good overview of your project workspace. It has a tree-like structure and is divided into separate sub-folders for convenience.



Sample Project View

Model Folder

The Model folder contains Devices and Device Parts.

- Devices are the offshore wind turbines in your project workspace. You can assemble a particular device using any of the sub-components which you have created.
- Device Parts are sub-components of the offshore wind turbine, such as turbines, towers, platforms, mooring lines etc. You can define as many sub-components as you like, even if you do not intend to use all of them immediately. So you can build up a library for both current and future projects.

Further details on each individual component is available by clicking on the relevant hyperlink.

Device Parts

- Aerofoil

- Blade
- Turbine
- Tower
- Semi-Sub Floater
- Mooring Line

System Level

- Floating Offshore Wind Turbine

Aerofoil

The Aerofoil component is still under development. Early versions of Flexcom Wind provide a range of pre-defined aerofoils which you may choose from when defining a Blade component. If you wish to define your own custom aerofoil, you can create a relevant aerofoil description which adheres to the OpenFAST conventions for defining aerofoils. Contact Wood if you require further assistance from the Flexcom Software Support Team.

Blade

STRUCTURAL DETAILS

Input:	Description
Blade Mass:	The mass of a single blade.
Blade Centre of Mass Location:	The centre of mass location, measured with respect to the root along the preconed axis.

BLADE GEOMETRY

Input:	Description

Blade Span:	The distance along the (possibly preconed) blade-pitch axis from the root. This is illustrated by 'BlSpn' in Blade Geometry - Side View. The span entries must be entered in monotonically increasing order, from the most inboard to the most outboard. The first span must be zero and the last span should be located at the blade tip. Each blade definition must have properties defined for at least two values of blade span.
Aerodynamic Centre Out-of-Plane Offset:	The local out-of-plane offset (when the blade-pitch angle is zero) of the aerodynamic center (reference point for the aerofoil lift and drag forces), normal to the blade-pitch axis, as a result of blade curvature. This is illustrated by 'BlCrvAC' in Blade Geometry - Side View. The offset is measured positive downwind; upwind turbines have negative offsets for improved tower clearance.
Aerodynamic Centre In-Plane Offset:	The local in-plane offset (when the blade-pitch angle is zero) of the aerodynamic center (reference point for the aerofoil lift and drag forces), normal to the blade-pitch axis, as a result of blade sweep. This is illustrated by 'BlSwpAC' in Blade Geometry - Front View. A positive offset is opposite the direction of rotation.
Curvature Angle:	The local angle (in degrees) from the blade-pitch axis of a vector normal to the plane of the aerofoil, as a result of blade out-of-plane curvature (when the blade-pitch angle is zero). This is illustrated by 'BlCrvAng' in Blade Geometry - Side View. Curvature angle is measured positive downwind; upwind turbines have negative curvature angles for improved tower clearance.
Twist Angle:	The local aerodynamic twist angle (in degrees) of the aerofoil. It is the orientation of the local chord about the vector normal to the plane of the aerofoil, positive to feather, leading edge upwind. The blade-pitch angle will be added to the local twist.
Chord Length:	The local chord length.

Aerofoil Index:	This entry specifies which aerofoil data input file is to associated with this local blade span position. Valid values are numbers between 1 and the Total Number of Aerofoil Input Files. Multiple blade nodes can use the same aerofoil data.

NOTES

(a) Early versions of Flexcom Wind use a single Point Mass at the hub to represent the weight of all three blades. As noted in Software Modelling Limitations, the effects of blade rotational inertia are not currently modelled. This item will be revisited in a future software development cycle.

Turbine

BLADES

Input:	Description
Blade Precone Angle:	The angle in degrees between a flat rotor disk and the cone swept by the blades, positive downwind. Upwind turbines have a negative precone angle for improved tower clearance. This is illustrated by 'Precone' in Turbine Geometry and subsequent schematics.
Blade Pitch Angle:	The blade pitch angle in degrees, positive to feather, leading edge upwind. The axis of rotation is illustrated by 'Blade-Pitch Axis' in Turbine Geometry, while the angle convention is illustrated by 'm: Pitching' in Blade Local Coordinate System.

HUB

Input:	Description

Hub Radius:	The radius to the blade root from the center-of-rotation along the (possibly preconed) blade-pitch axis. This is illustrated by 'HubRad' in Turbine Geometry and subsequent schematics.
Hub Mass:	The mass of the hub.

ROTOR

Input:	Description
Rotor Speed:	The fixed rotor speed (positive clockwise looking downwind).
Shaft Tilt:	The angle in degrees between the rotor shaft and the horizontal plane. A positive shaft tilt angle means that the downwind end of the shaft is the highest. Upwind turbines have a negative shaft tilt angle for improved tower clearance. This is illustrated by 'ShftTilt' in Turbine Geometry and subsequent schematics.
Rotor Overhang:	The distance along the (possibly tilted) rotor shaft between the tower centerline and hub center, measured positive downwind. Upwind rotors have a negative overhang. This is illustrated by 'Overhang' in Turbine Geometry and subsequent schematics.
Rotor Shaft to Tower Distance:	This is the vertical distance from the top of the tower and yaw bearing to the intersection of the rotor shaft axis and the lateral plane (i.e. plane perpendicular to wind direction). The distance is measured parallel to the vertical (tower) axis. This is illustrated by 'Twr2Shft' in Conventional Upwind Turbine Layout.
Rotor Inertia:	Rotational inertia of the rotors about the low-speed shaft which is $\int r^2 dm$, the integral of the rotor (blades + hub + low-speed shaft) mass by the distance from the axis of rotation.

NACELLE

Input:	Description
Nacelle Mass:	The mass of the nacelle.
Nacelle Yaw Bearing Mass:	The mass of the nacelle yaw bearing.
Nacelle Yaw:	This is fixed nacelle yaw angle. It is positive counterclockwise when looking down on the turbine.
Tower Top to Nacelle Centre of Mass (Downwind):	This is the downwind distance to the nacelle mass center from the top of the tower, measured parallel to the x_n-axis. It is positive downwind. This is illustrated by 'NacCMxn' in Conventional Upwind Turbine Layout. Refer to Nacelle/Yaw Coordinate System for an illustration of the local axes.
Tower Top to Nacelle Centre of Mass (Lateral):	This is the lateral distance to the nacelle mass center from the top of the tower, measured parallel to the y_n-axis. It is positive to the left when looking downwind or positive into the page of Conventional Upwind Turbine Layout. Refer to Nacelle/Yaw Coordinate System for an illustration of the local axes.
Tower Top to Nacelle Centre of Mass (Vertical):	This is the vertical distance to the nacelle mass center from the top of the tower, measured parallel to the z_n-axis. It is positive upward when looking downwind. This is illustrated by 'NacCMzn' in Conventional Upwind Turbine Layout. Refer to Nacelle/Yaw Coordinate System for an illustration of the local axes.

DRIVETRAIN

Input:	Description
Gearbox Ratio:	This is the ratio of the high-speed shaft to the low-speed shaft. This value should be greater than zero and equal to unity for a direct-drive turbine.

CONTROL

Input:	Description
ServoDyn Input:	Path to the ServoDyn input file which must contain the Bladed-style dynamic link library (dll) name for implementing turbine control. See Note (a).

NOTES

(a) The ServoDyn input file must have the variable `PCmode` in the `PITCH CONTROL` section set to 0 (*None*) or 5 (*User-Defined from Bladed-style DLL)*. In addition to this, `VSContrl` in the `GENERATOR AND TORQUE CONTROL` section should be set to 0 (*None*) or 5 (*User-Defined from Bladed-style DLL)*. `YCMode` in the `NACELLE-YAW CONTROL` section should be set to 0 (*None*) or 5 (*User-Defined from Bladed-style DLL)*. The `DLL_Filename` in the `BLADED INTERFACE` section should contain the path to the user defined control dll. Please see the ServoDyn documentation for further details on the input file variables.

Users should be conscious of the bitness of the user dll used in relation to the bitness of the Flexcom installation. Flexcom Wind provides a standard example; L01 - OC4 Semisubmersible, which contains ServoDyn input data and the DISCON OC3 Hywind dll (32 & 64bit) as shipped with the FAST v8 CertTest Test25. Users should point to the 32bit dll for a 32bit Flexcom installation and visa versa for a 64bit installation.

Tower

GEOMETRY

Input:	Description
Tower Height:	The distance from mean sea level to the top of the tower and yaw bearing. This is illustrated by 'TowerHt' in Conventional Upwind Turbine Layout.
Base Diameter:	The diameter of the tower at its base.
Base Thickness:	The wall thickness of the tower at its base.
Top Diameter:	The diameter of the tower at its top.
Top Thickness:	The wall thickness of the tower at its top.

MATERIAL PROPERTIES

Input:	Description
Young's Modulus:	The Young's Modulus (E) of the tower material.
Shear Modulus:	The Shear Modulus (G) of the tower material.
Density:	The mass density of the tower material.

STRUCTURAL PROPERTIES

Input:	Description

Stiffness Damping Coefficient:	The stiffness proportional damping coefficient for the tower material. Refer to Damping Coefficients for further details.

NOTES

(a) Refer to Tower Geometry for an illustration of the tower inputs.

(b) Early versions of Flexcom Wind model a conical tower whose diameter decreases linearly from base to top. It may be possible to model towers of arbitrary profile in a future software version.

(c) The diameter and thickness values are used to compute a mass per unit length for the structural elements which model the tower.

Semi-Sub Platform

The Semi-Sub Platform component is sub-divided into 3 sub-sections...

- Geometry
- Structural Properties
- Dynamic Loads
- Mooring

PLATFORM

Input:	Description
Depth of Platform below SWL:	The depth of the platform's lowest point (typically the base of the outer columns) below the still water line.

Platform Reference Point Location:	The depth of the platform's reference point below the still water line. The reference point is the location at which the various hydrodynamic forces are applied to the floating platform.
Platform Heading:	The initial yaw orientation of the platform, measured anticlockwise from the global Y-axis.
Displaced water:	The total volume of water displaced by the floater in its initial, undisplaced position.

MAIN COLUMN

Input:	Description
Main Column Length:	The length of the central column.
Main Column Diameter:	The diameter of the central column.

OUTER COLUMNS

Input:	Description
Upper Section Length:	The length of the upper section of each outer column.
Upper Section Diameter:	The diameter of the upper section of each outer column.
Base Section Length:	The length of the base section of each outer column.

Base Section Diameter:	The diameter of the base section of each outer column.
Spacing:	The spacing between the outer columns (which are arranged in the form of an equilateral triangle in plan view) measured centre-to-centre (i.e. excluding the column diameters).

PONTOONS & CROSS BRACES

Input:	Description
Diameter of Cross Braces:	The diameter of the cross braces.

MASS

Input:	Description
Platform Mass:	The mass of the floating platform.
Platform Ballast:	The total mass of the water ballast which is contained within the platform's columns.
Centre of Mass Location:	The platform's centre of mass location, measured vertically from the base of the central column. The total mass and rotational inertia of the floating platform is concentrated at this point.
Centre of Buoyancy Location:	The platform's centre of buoyancy location, measured vertically from the base of the central column. Buoyancy forces on the floating platform, which are modelled via linear hydrostatic stiffness terms, are concentrated at this point.

Roll Inertia:	The inertia about the local roll axis.
Pitch Inertia:	The inertia about the local pitch axis.
Yaw Inertia:	The inertia about the local yaw axis

HYDROSTATIC STIFFNESS

Input:	Description
Heave:	The hydrostatic stiffness in the local heave direction.
Roll:	The hydrostatic stiffness in the local roll direction.
Pitch:	The hydrostatic stiffness in the local pitch direction.
Heave-Pitch:	The hydrostatic stiffness coupling term between heave and pitch.

ADDED MASS & RADIATION DAMPING

Input:	Description
Added Mass Coefficients:	The name of the external data file which contains the added mass data. Refer to Added Mass Coefficients for further information on the file format. If you are modelling the floater hydrodynamics via the simplified Morison's Equation approach, you should leave this entry blank.

Radiation Damping Coefficients:	The name of the external data file which contains the radiation damping data. Refer to Radiation Damping Coefficients for further information on the file format. If you are modelling the floater hydrodynamics via the simplified Morison's Equation approach, you should leave this entry blank.
Reference Frequency:	The reference frequency to be used in determining the added mass of the floating platform. Refer to floating body Hydrodynamic Theory for further information on the convolution integral technique employed by Flexcom to model frequency dependent added mass and radiation damping terms in a time domain simulation.
Normal Added Mass Coefficient for All Members:	The added mass coefficient in the direction normal to each structural member. If you are modelling the floater hydrodynamics via the simplified Morison's Equation approach, you should specify a realistic added mass coefficient here. If you are using the more detailed Radiation-Damping approach, you should leave this entry blank.
Tangential Added Mass Coefficient for Outer Columns' Base:	The added mass coefficient in the direction tangential to the base of each outer column. Flexcom does not model hydrodynamic loads on element end faces directly, so the tangential component of Morison's Equation may be used to simulate loads on the relatively large circular area at the base of the outer columns. If you are using the more detailed Radiation-Damping approach, you should leave this entry blank.
Normal Inertia Coefficient for All Members:	The inertia coefficient in the direction normal to each structural member. If you are modelling the floater hydrodynamics via the simplified Morison's Equation approach, you should specify a realistic added mass coefficient here. If you are using the more detailed Radiation-Damping approach, you should leave this entry blank.

RAO & QTF

Input:	Description
Force RAOs:	The name of the external data file which contains the force RAO data. Refer to Force RAOs for further information on the file format. If you are modelling the floater hydrodynamics via the simplified Morison's Equation approach, you should leave this entry blank.
QTF Coefficients:	The name of the external data file which contains the QTF data. Refer to QTF Coefficients for further information on the file format. If you are modelling the floater hydrodynamics via the simplified Morison's Equation approach, you should leave this entry blank.

VISCOUS DRAG

Input:	Description
Normal Drag Coefficient for Main Column:	The drag coefficient in the direction normal to the central column.
Normal Drag Coefficient for Outer Columns' Upper Section:	The drag coefficient in the direction normal to the upper section of the outer columns.

Normal Drag Coefficient for Outer Columns' Base Section:	The drag coefficient in the direction normal to the base section of the outer columns.
Normal Drag Coefficient for Cross Braces:	The drag coefficient in the direction normal to the cross braces.
Tangential Drag Coefficient for Outer Columns' Base Section:	The drag coefficient in the direction tangential to the base section of the outer columns. Flexcom does not model hydrodynamic loads on element end faces directly, so the tangential component of Morison's Equation may be used to simulate loads on the relatively large circular area at the base of the outer columns.
Tangential Drag Coefficient for All Other Members:	The drag coefficient in the direction tangential to each structural member.

NOTES

(a) The Hydrodynamic Data Importer is a very helpful utility program which allows you to automatically import characteristic data relating to a Floating Body from a range of well known hydrodynamic simulation packages into Flexcom.

MOORING ANCHOR CONNECTIONS

Input:	Description
Anchor X:	The global X coordinate of the mooring anchor point.
Anchor Y:	The global Y coordinate of the mooring anchor point.
Anchor Z:	The global Z coordinate of the mooring anchor point.

Notes

(a) In Flexcom, the global X axis is vertical, with the global Y and Z axes lying in a horizontal plane such that the system is right-handed. Refer to Axes and Displacements for further information.

Mooring Line

GEOMETRIC PROPERTIES

Input:	Description
Length:	The length of the mooring line.
Min Mesh Density:	The minimum element length to be used in creating the finite element model of the mooring line.
Max Mesh Density:	The maximum element length to be used in creating the finite element model of the mooring line.
Diameter:	The diameter of the mooring line.

HYDRODYNAMIC PROPERTIES

Input:	Description

Normal Drag Coefficient:	The drag coefficient for the direction normal to the mooring line.
Tangential Drag Coefficient:	The drag coefficient for the direction tangential to the mooring line.
Normal Added Mass Coefficient:	The added mass coefficient for the direction normal to the mooring line.
Tangential Added Mass Coefficient:	The added mass coefficient for the direction tangential to the mooring line.

STRUCTURAL PROPERTIES

Input:	Description
Axial Stiffness:	The axial stiffness for the mooring line.
Mass per Unit Length:	The mass per unit length of the mooring line.
Stiffness Damping Coefficient:	The stiffness proportional damping coefficient for the mooring line material. Refer to Damping Coefficients for further details.

Mass Damping Coefficient:	The mass proportional damping coefficient for the mooring line material. Refer to Damping Coefficients for further details.

NOTES

(a) Refer to Line Mesh Generation for a detailed discussion of the automatic mesh creation facility.

(b) Hydrodynamic loading on the mooring lines is based on Morison's Equation.

Floating Offshore Wind Turbine

COMPONENTS

Input:	Description
Blade:	Select the relevant Blade from the list of components.
Turbine:	Select the relevant Turbine from the list of components.
Tower:	Select the relevant Tower from the list of components.
Platform:	Select the relevant platform from the list of components. This is currently limited to semi-submersibles, spars and TLPs will be introduced in a future version.

MOORING CONNECTIONS

Input:	Description
Line:	Select the relevant Mooring Line from the list of components.
Anchor X:	The global X coordinate of the anchor point.
Anchor Y:	The global Y coordinate of the anchor point.
Anchor Z:	The global Z coordinate of the anchor point.

Fairlead Connection:	Select the relevant fairlead position from the list of available connection points on the Platform component.

Environment Folder

The Environment folder stores a range of different environmental conditions which the offshore wind turbine may be designed to operate in.

Further details on each individual component is available by clicking on the relevant hyperlink.

- Environment
- Wind
- Current
- Regular Wave
- JONSWAP Wave

Environment

GENERAL

Input:	Description
Gravity Constant:	The acceleration due to gravity.

AIR

Input:	Description
Air Density:	The air density; a typical value is around 1.225 kg/m^3.

Air Kinematic Viscosity:	The kinematic viscosity of the air (used in the Reynolds number calculation); a typical value is around 1.460E-5 m^2/s.
Speed of Sound in Air:	The speed of sound in air (used in the Mach number calculation); a typical value is around 340.3 m/s.
Atmospheric Pressure:	The atmospheric pressure above the free surface. This is typically around 101,325 Pa.
Vapour Pressure:	The vapor pressure of the fluid. For seawater this is typically around 2,000 Pa.
Fluid Depth:	The distance from the hub center to the free surface.

Notes

(a) The last three parameters (*Atmospheric Pressure*, *Vapour Pressure* & *Fluid Depth*) are only used when a Cavitation Check is enabled for MHK (Marine and Hydrokinetic Technology) turbines. As Flexcom is currently designed to work with wind turbines only, these options are not relevant.

WATER

Input:	Description
Water Depth:	The water depth.
Water Density:	The mass density (mass per unit volume) of seawater.

SEABED

Input:	Description

Seabed Stiffness:	The elastic stiffness per unit length of the seabed, in units of [Force]/[Distance]/[Distance].
Longitudinal Coefficient of Friction:	The coefficient of friction in the longitudinal direction. For any lines which contact the seabed, the local longitudinal direction is parallel to the line axis, while the local transverse direction is perpendicular to it.
Transverse Coefficient of Friction:	The coefficient of friction in the transverse direction.

Notes

(a) Refer to Seabed Interaction for further information.

Wind

Input:	Description
Wind Speed:	The steady wind speed located at an elevation corresponding to the hub height. The hub height is determined from the vertical elevation of the hub node in the structural model. See Note (a).
Shear Exponent:	The power-law shear exponent which defines the wind speed profile as a function of vertical elevation. See Note (a).
InflowWind Input:	Path to the InflowWind input file. Inclusion of a file path here overrides the data in the *Wind Speed* and *Shear Exponent* entries above. See Note (b).

NOTES

(a) The local undisturbed wind speed, U(x), for any given blade or tower node is determined using the following expression:

$$U(x) = WindSpeed\left(\frac{x}{HubHeight}\right)^{ShearExponent}$$

where x is the instantaneous elevation of the blade or tower node above the mean water line. Refer to the AeroDyn documentation for further information.

(b) Refer to the InflowWind documentation for a full description of the InflowWind input file values and format.

Current

The Current component is still under development. If you wish to apply current loading in your model, you can always Migrate to Flexcom itself. Contact Wood if you require further assistance from the Flexcom Software Support Team.

Regular Wave

REGULAR WAVES

Input:	Description
Amplitude:	The regular Airy wave amplitude (MWL to crest or trough).
Period:	The regular Airy wave period in seconds.
Direction:	The wave direction, measured in degrees anti-clockwise from the global Y-direction. All waves in Flexcom emanate from the origin.
Phase:	The wave phase angle in degrees. This input is appropriate only if more than one regular wave is specified.

NOTES

(a) Refer to Regular Airy Wave for further information on Airy wave theory.

JONSWAP Wave

JONSWAP WAVES

Input:	Description
Wave Height:	The spectrum significant wave height Hs.
Peak Period:	The spectrum peak period Tp in seconds.
Peakedness Parameter:	The spectrum peakedness parameter γ.
Max Frequency Increment:	The maximum frequency increment in Hz. to be used in the spectral discretisation. See Note (b).
Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. See Note (b).
No. of Harmonics:	The number of harmonics to be used in the spectral discretisation. See Note (b).
Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. See Note (c).

Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. See Note (c).

NOTES

(a) Refer to Jonswap Wave for further information on the Jonswap spectral formulation.

(a) The wave spectrum may be discretised into segments based on an equal area approach (which divides the area under the spectrum into segments of equal area) or a geometric progression approach (based on frequency increments that form a geometric progression). Refer to Spectrum Discretisation for a detailed discussion of this discretisation procedure.

(b) A multi-directional random sea is defined in terms of a dominant wave direction and the number of wave directions. Refer to Wave Energy Spreading for further information.

Simulations Folder

The Simulation folder contains a list of the Simulation components which you wish to run. It links up the offshore wind turbine model with the environmental conditions to which it is subjected. Each Simulation component allows you to Run Simulations and examine Simulation Results.

Simulation

The Simulation component is sub-divided into 3 sub-sections...

- Simulation Parameters
- Running Simulations
- Simulation Results

DEVICE & ENVIRONMENT

SOLUTION TIME STEPPING

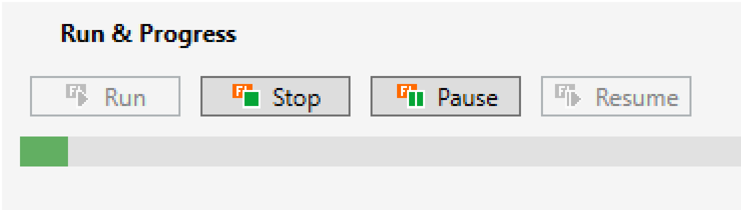
Input:	Description
Duration:	The analysis duration in seconds. Selection of an appropriate simulation length depends on whether you are running a regular wave or random sea. Refer to Simulation Length for further details.
Time Step:	The fixed time step to be used in the analysis. It is important to choose a time-step which picks up necessary detail in excitation and response. Refer to Choice of Time Step for further details. For wind turbine modelling, NREL recommend that the solution time step for aerodynamic calculations be set such that there are at least 200 time steps per rotor revolution.
Ramp Time:	The time over which applied loads and displacements are gradually increased to their full value. Refer to Load Ramping for further details.

OVERVIEW

When you are satisfied with the model and environmental inputs, you can proceed to running a simulation. Select the simulation of interest in the Project View, right-click and select 'Run Simulation'. The software then uses all the data which you have defined in Flexcom Wind and uses it to create a Flexcom model in standard Keyword File format. It then passes this input file to the engine to perform the finite element computations. Assuming the simulation has completed successfully, Simulation Results will be available.

SIMULATION PROGRESS

While the simulation is in progress, you will notice a green progress bar to indicate the simulation progress.



Progress Bar

RUN SEQUENCE

In case you are wondering what is happening in the background; for each simulation, Flexcom Wind actually creates a series of analysis files which run consecutively.

- Firstly an Initial Static Analysis is performed in order to determine the static equilibrium configuration of the system subject to gravity and buoyancy loads only. In order to aid solution convergence, this simulation is normally performed in two stages, with the floater being temporarily restrained using additional boundary conditions, which are subsequently removed in a Restart Static Analysis. Floating systems can be sensitive to minor changes in displacement, so this second stage is typically performed as a Quasi-Static Analysis.

- One further restart analysis may also be performed if static Current Loads on the floater are being modelled.

- Finally the ambient environmental loads due to wind and waves are simulated in a Time Domain Analysis.

Existing Flexcom users will be familiar with the concept of an Analysis Job, which typically contains a series of individual analysis files which run consecutively. The procedure is very similar here, except that Flexcom Wind has automatically set up the analysis job for increased convenience to the user.

After the simulation has completed successfully, results will be available from the Simulation Results tab. It presents a series of 2D Plots which cover the main outputs which are typically of interest from a simulation. Customised post-processing, where users of Flexcom Wind will be able to request specific outputs, is planned for inclusion in a future version. At the moment, the information below is presented as standard. If you require any additional outputs, you could always Migrate to Flexcom itself.

- Platform Heave

- Platform Surge
- Platform Sway
- Platform Yaw
- Platform Roll
- Platform Pitch
- Undisturbed Wind Velocity at Top of Tower
- Rotor Speed
- Rotor Aerodynamic Power
- Rotor Aerodynamic Force Fx (wind direction)
- Rotor Aerodynamic Force Fy (lateral direction)
- Rotor Aerodynamic Force Fz (vertical direction)
- Rotor Aerodynamic Moment Mx (roll direction)
- Rotor Aerodynamic Moment My (pitch direction)
- Rotor Aerodynamic Moment Mz (yaw direction)
- Electrical Generator Torque
- Blade pitch
- Electrical Generator Power
- Mooring Line 1 Fairlead Tension
- Mooring Line 2 Fairlead Tension
- Mooring Line 3 Fairlead Tension

Existing Flexcom users may recognise that the *Simulation Results* component of Flexcom Wind is effectively Flexcom's standard Plotting module running in standalone mode.

1.7.7.2 Migrate to Flexcom

While simulations may be performed directly within the Flexcom Wind module itself, experienced Flexcom users may prefer to use the specialised module for initial model construction only, and then switch to the main Flexcom environment as it provides greater modelling flexibility. Migrating is as easy as flicking a switch, you simply select the 'Open as Flexcom Project' from the *Simulations* menu. The wind turbine module produces a well-structured, heavily parameterised keyword file which can be subsequently customised to meet your own individual requirements.

The structural components of the model, such as the floater, tower and mooring lines, are all constructed in standard fashion using established Flexcom modelling capabilities. Refer to Model Building for further information - this article outlines a typical series of steps to follow when building a model of a floating wind turbine.

The aerodynamic inputs are all neatly organised in a dedicated section of the keyword file called $AERODYN, and the *AERODYN DRIVER keyword provides the key link between the structural and aerodynamic models. So apart from learning a few new keywords, experienced Flexcom users will not have any difficulty in building their own floating wind turbine models. Additionally, a dedicated aerodynamic section has been added to the *TIMETRACE command to facilitate extraction of turbine specific information, such as power, rotor speed, aerodynamic forces and moments etc.

1.7.8 Flexcom Wave

OVERVIEW

Flexcom Wave is Wood's dedicated simulator package for wave energy conversion devices. It represents a user-friendly, cost effective, software solution which is intended to allow developers to gain a deeper understanding of the structural response and energy generation potential of their device. Information derived from realistic simulations will facilitate progressive migration through the various stages, from conceptual, scale model, prototype right though to full scale versions.

Transition from early stage concept development to detailed engineering is seamless, as Flexcom Wave interfaces directly with ExceedenceFinance, a techno-financial modelling tool which can provide financial appraisals throughout each stage of the project life cycle. Through the collaboration, users of Flexcom Wave can readily access wave resource information for locations around the globe, while results from the engineering simulator ensures that all financial appraisals are based on realistic power data for a given wave energy converter and ambient environment.

The numerical modelling capabilities have been validated via several studies, including:

- Simulation of a two-body floating point absorber chosen by the U.S. Department of Energy as a reference model to benchmark marine energy technology performance. Validation involved code-to-code benchmarks with other software, and comparisons with experimental data derived from model-scale tank test facilities (Connolly et al., 2018).

- Simulation of a real-world device named ‘DUO’, a novel concept, which connects two oscillating bodies via angled pre-tensioned cable linkages. Results from the numerical simulations were validated with empirical data derived from model-scale tank tests (Connolly & Brewster, 2017).

FURTHER INFORMATION

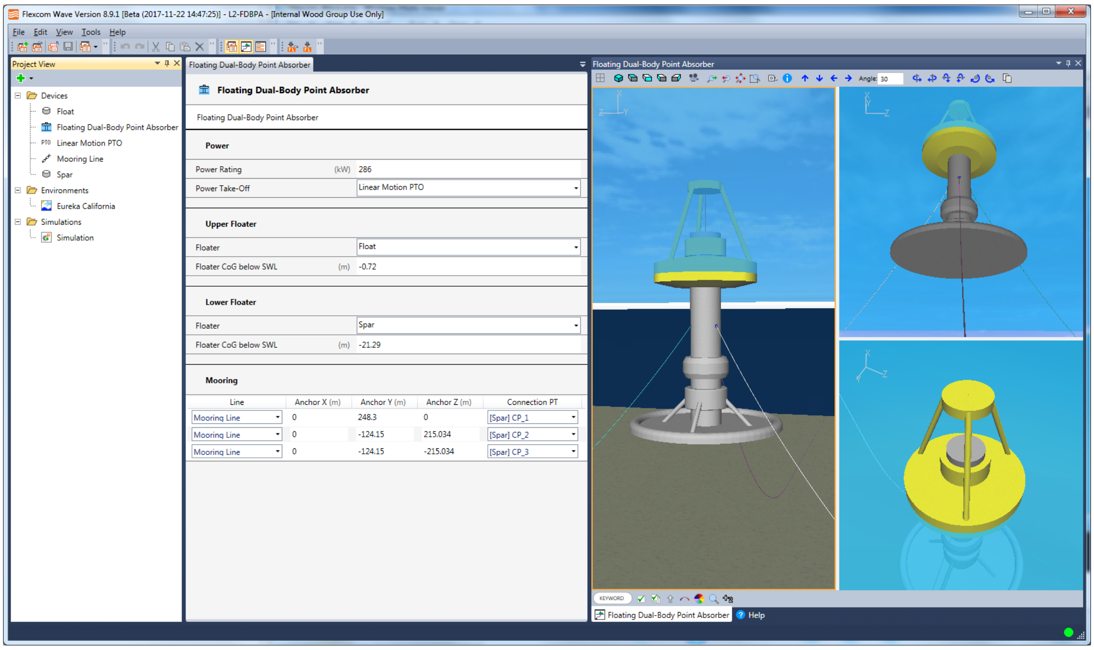
Further information on this topic is contained in the following sections:

- Project View introduces the project workspace, a container for all of the components in your project (e.g. floaters, PTO, mooring lines etc.)

- Model View describes the structure display facility. This feature provides both a structure preview facility for use during model building, and a way of viewing an animation of the structure response after an analysis has completed.

- Migrate to Flexcom allows you to switch to the main Flexcom environment for greater modelling flexibility.

The above sections focus on the user interface aspects of the Flexcom Wave module. For further background information on the mathematical engine and associated theory, refer to Floating Body.



Flexcom Wave

1.7.8.1 Project View

The Project View acts as a container for all of the components in your project, and provides a good overview of your project workspace. It has a tree-like structure and is divided into separate sub-folders for convenience.

- Model contains Devices and Device Parts.
  - Devices are the wave energy converters in your project workspace. You can assemble a particular device using any of the sub-components which you have created.
  - Device Parts are sub-components of the wave energy converter, such as floating bodies, power take-offs, mooring lines etc. You can define as many sub-components as you like, even if you do not intend to use all of them immediately. So you can build up a library for both current and future projects.

- Environments stores a range of different environmental conditions which the wave energy converter may be designed to operate in. Flexcom Wave interfaces directly with ExceedenceFinance, a third-party software package which contains databases of metocean data for various geographical locations around the world. Provided you have a license for this software, you can quickly download wave resource information for your chosen location, and automatically Import from Exceedence into Flexcom Wave.

- Simulations allows you to perform numerical simulations and examine analysis results. In the Simulation Component you nominate a wave energy converter and choose the environmental conditions to which it is subjected. You may then proceed to Running Simulations. Following successful completion of the numerical simulations, you may examine important Simulation Results. Crucially you can also Export to Exceedence, sending the power matrix computed by Flexcom's detailed engineering model back into ExceedenceFinance, and perform financial appraisals based on realistic power data for a given wave energy converter and ambient environment.

Device Parts

Device Parts are sub-components of the wave energy converter, such as floating bodies, power take-offs, mooring lines etc. You can define as many sub-components as you like, even if you do not intend to use all of them immediately. So you can build up a library for both current and future projects.

Further details on each individual component is available by clicking on the relevant hyperlink.

- Floater
- Mooring Line
- Tether
- Power Take-Off

Floater

The Floater component is sub-divided into 3 sub-sections...

- Geometry

- Structural Properties
- Dynamic Loads

DIMENSIONS

Input:	Description
Length:	The overall length of the floating body.
Diameter:	The outer diameter of the floating body.
Center of Mass Location:	The position of the floating body's center of mass, measured upwards from the base of the floating body.

CONNECTION POINTS

Input:	Description
Number of Connection Points:	The number of connection points on the floater. Any of these connection points may be referenced subsequently when creating a device.
Height Above Floater Base:	The vertical height of the connection point above the base of the floater.
Offset from Central Axis:	The horizontal distance between the connection point and the central axis of the floater.

DISPLAY

Input:	Description
Profile File:	The name of the profile file used to provide a visual representation of the floater. See Note (a).

Notes

(a) Structurally the floater is represented by a small number of finite elements and it will look rather skeleton-like in the Model View, not remotely resembling a floating body. So the addition of an auxiliary Component Profile is advisable. While this will have no structural function, it will greatly enhance the visual appeal of the model, and will assist in the understanding of floater motions post-simulation. The profile is defined in XML format, and several sample profiles are available in your local Flexcom installation directory, under the sub-folder ‘StandardProfiles’. Each profile file contains a range of input data, capable of defining standard shapes such as boxes, cylinders etc., as well as arbitrary shapes defined by a series of points about which a mesh is constructed. All locations within the profile file are defined with respect to a local origin (0, 0, 0). The location of this origin in the global axis system is then defined by the *Floater Origin* point in the Device component where it is referenced. Should you require any assistance in creating your own customised profile, feel free to contact sw.support@woodplc.com and our support team will discuss the format of the XML information in more detail.

MASS & INERTIA

Input:	Description
Mass:	M - the mass of the floating body.
Roll Inertia:	I_{44} - the inertia about the local roll axis.
Pitch Inertia:	I_{55} - the inertia about the local pitch axis.
Yaw Inertia:	I_{66} - the inertia about the local yaw axis

HYDROSTATIC STIFFNESS

Input:	Description
Heave:	K_{33} - the hydrostatic stiffness in the local heave direction.
Roll:	K_{44} - the hydrostatic stiffness in the local roll direction.
Pitch:	K_{55} - the hydrostatic stiffness in the local pitch direction.
Heave-Roll:	K_{34} - the hydrostatic stiffness coupling term between heave and roll.
Heave-Pitch:	K_{35} - the hydrostatic stiffness coupling term between heave and pitch.
Roll-Pitch:	K_{45} - the hydrostatic stiffness coupling term between roll and pitch.
Roll-Yaw:	K_{46} - the hydrostatic stiffness coupling term between roll and yaw.
Pitch-Yaw:	K_{56} - the hydrostatic stiffness coupling term between pitch and yaw.

ADDED MASS & RADIATION DAMPING

Input:	Description
Reference Frequency:	The reference frequency to be used in determining the added mass of the floating body. Refer to floating body Hydrodynamic Theory for further information on the convolution integral technique employed by Flexcom to model frequency dependent added mass and radiation damping terms in a time domain simulation.

Added Mass Coefficients:	The name of the external data file which contains the added mass data. Refer to Added Mass Coefficients for further information on the file format.
Radiation Damping Coefficients:	The name of the external data file which contains the radiation damping data. Refer to Radiation Damping Coefficients for further information on the file format.

RAO & QTF

Input:	Description
Force RAOs:	The name of the external data file which contains the force RAO data. Refer to Force RAOs for further information on the file format.
QTF Coefficients:	The name of the external data file which contains the QTF data. Refer to QTF Coefficients for further information on the file format.

NOTES

(a) The Hydrodynamic Data Importer is a very helpful utility program which allows you to automatically import characteristic data relating to a Floating Body from a range of well known hydrodynamic simulation packages into Flexcom.

Mooring Line

GEOMETRIC PROPERTIES

Input:	Description
Unstretched Length:	The length of the mooring line.

Minimum Mesh Density:	The minimum element length to be used in creating the finite element model of the mooring line.
Maximum Mesh Density:	The maximum element length to be used in creating the finite element model of the mooring line.
Diameter:	The diameter of the mooring line.

Notes

(a) Refer to Line Mesh Generation for a detailed discussion of the automatic mesh creation facility.

STRUCTURAL PROPERTIES

Input:	Description
Axial Stiffness:	The axial stiffness for the mooring line.
Mass per unit Length:	The mass per unit length of the mooring line.
Stiffness Damping Coefficient:	The stiffness proportional damping coefficient for the mooring line material. Refer to Damping Coefficients for further details.
Mass Damping Coefficient:	The mass proportional damping coefficient for the mooring line material. Refer to Damping Coefficients for further details.
Bending Stiffness:	The bending stiffness for the mooring line.

Torsional Stiffness:	The torsional stiffness for the mooring line.

HYDRODYNAMIC PROPERTIES

Input:	Description
Normal Drag Coefficient:	The drag coefficient for the direction normal to the mooring line.
Tangential Drag Coefficient:	The drag coefficient for the direction tangential to the mooring line.
Normal Added Mass Coefficient:	The added mass coefficient for the direction normal to the mooring line.
Tangential Added Mass Coefficient:	The added mass coefficient for the direction tangential to the mooring line.

Notes

(a) Hydrodynamic loading on the mooring line is based on Morison's Equation.

Tether

GEOMETRIC PROPERTIES

Input:	Description
Minimum Mesh Density:	The minimum element length to be used in creating the finite element model of the tether.
Maximum Mesh Density:	The maximum element length to be used in creating the finite element model of the tether.
Diameter:	The diameter of the tether.

Notes

(a) Refer to Line Mesh Generation for a detailed discussion of the automatic mesh creation facility.

STRUCTURAL PROPERTIES

Input:	Description
Axial Stiffness:	The axial stiffness for the tether.
Mass per unit Length:	The mass per unit length of the tether.
Stiffness Damping Coefficient:	The stiffness proportional damping coefficient for the tether material. Refer to Damping Coefficients for further details.
Mass Damping Coefficient:	The mass proportional damping coefficient for the tether material. Refer to Damping Coefficients for further details.

HYDRODYNAMIC PROPERTIES

Input:	Description
Normal Drag Coefficient:	The drag coefficient for the direction normal to the tether.
Tangential Drag Coefficient:	The drag coefficient for the direction tangential to the tether.
Normal Added Mass Coefficient:	The added mass coefficient for the direction normal to the tether.
Tangential Added Mass Coefficient:	The added mass coefficient for the direction tangential to the tether.

Notes

(a) Hydrodynamic loading on the tether is based on Morison's Equation.

Linear Motion Power Take-Off

GEOMETRY

Input:	Description
Initial Length:	The overall length of the power-take off device in its initial configuration. In other words, its maximum length is equal to the sum of the initial length plus the upper end stop distance

Upper End Stop Distance:	The maximum extension of the power-take off device before it reaches its end stop, relative to its initial position. This entry is optional and if omitted, no upper end stop is modelled.
Lower End Stop Distance:	The maximum contraction of the power-take off device before it reaches its end stop, relative to its initial position. This entry is optional and if omitted, no lower end stop is modelled.
Diameter:	The physical diameter of the power-take off device, assuming a cylindrical shape. This entry is optional, and is used for visualisation purposes only.

Notes

(a) The power-take off mechanism is represented as a cylindrical piston-and-sleeve type device in Flexcom. In terms of a physical device, this modelling approach is suitable for hydraulic or pneumatic pistons, or linear electric generators.

(b) The end stops govern the range of free movement of the mechanism, as discussed further in Stiffness.

(c) The *Diameter* input relates to the size of the outer sleeve. While it has no structural function, it provides a useful visualisation of the power-take off mechanism in operation. The diameter of the piston is assumed to be equal to half that of the sleeve.

DAMPING

Input:	Description
Constant Damping Force:	The constant damping force applied by the power-take off. This entry is optional and if omitted, no constant damping force is modelled.
Velocity Threshold :	A velocity threshold below which the constant damping term is linearly ramped. This entry is optional and if omitted, the constant damping force is applied at all times. See Note (b).

Linear Damping Coefficient:	The linear damping coefficient for the power-take off. This entry is optional and if omitted, no linear damping term is modelled.
Quadratic Damping Coefficient:	The quadratic damping coefficient for the power-take off. This entry is optional and if omitted, no quadratic damping term is modelled.

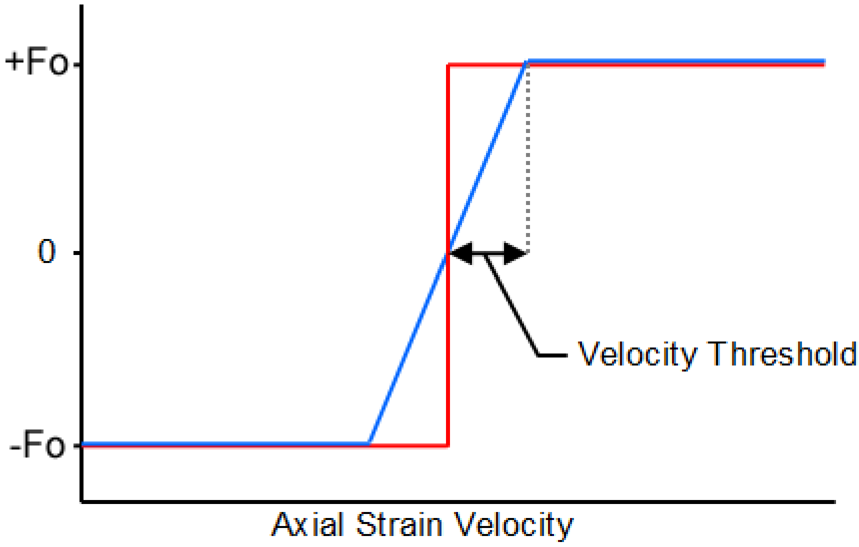
Notes

(a) Flexcom models the damping aspects of the power take-off using a non-linear damper element. The axial force exerted by the damper element, F, is defined as follows:

$$F = -(F_0 + C_1 v + C_2 v^2)$$

where v is the relative velocity between the element end nodes in the axial direction of the element, F_0 is the constant damping force, and C_1 & C_2 are the linear and quadratic damping coefficients respectively.

(b) It is possible to control the application of the constant damping force (F_0 term) by specifying a velocity threshold. By default no ramping is applied, so the damping force is applied in full at all times. The significance of the velocity threshold is illustrated in the figure below. Rather than switching instantaneously between positive and negative force values, the constant force term is gradually ramped up over a finite range of relative velocity. This tends to smooth out the applied damping forces and eliminates the possibility of a dramatic alternation between positive and negative force terms for small extensions and contractions of the damper element.



Velocity Threshold

STIFFNESS

Input:	Description
Constant Force:	The constant spring force applied by the power-take off. This entry is optional and if omitted, no constant force is modelled.
Extension Stiffness:	The standard resistive stiffness of the power-take off in regions of free movement. This entry is optional and if omitted, the extension stiffness term is assumed to be zero.
Upper End Stop Stiffness:	The high resistive stiffness of the power-take off device as the device reaches its upper end stop. This entry is optional and if omitted, no upper end stop is modelled.
Lower End Stop Stiffness:	The high resistive stiffness of the power-take off device as the device reaches its lower end stop. This entry is optional and if omitted, no lower end stop is modelled.
Upper End Stop Ramp Length:	The length of a transition region between the extension stiffness and the upper end stop stiffness. This entry is optional and if omitted, there is an instantaneous change in slope of the stiffness curve at the transition point.

Lower End Stop Ramp Length:	The length of a transition region between the extension stiffness and the lower end stop stiffness. This entry is optional and if omitted, there is an instantaneous change in slope of the stiffness curve at the transition point.

Notes

(a) Flexcom models the stiffness aspects of the power take-off using a non-linear spring element. In regions of free movement, the restoring force provided by the spring element, F, is defined as follows:

$$F = F_0 + K_{FM} x$$

where F_0 is the constant spring force, K_{FM} is the extension stiffness, and x is the extension in the spring element.

(b) As the spring extends or contracts towards its end stop limits, the spring stiffness is increased to the end stop stiffness over the end stop ramp length to provide a high level of resistance to any further motion. For example, at the end of the upper end stop transition region, the restoring force, F, provided by the spring element is defined as follows:

$$F = F_0 + K_{FM}(x_{ES}) + K_{ES}(x - x_{ES})$$

where K_{ES} is the upper end stop stiffness, and x_{ES} is the upper end stop distance.

POWER CONVERSION EFFICIENCY

Input:	Description
Conversi on Efficiency Factor:	A power conversion efficiency factor which accounts for the losses between the generated mechanical power and the electrical power output.

Devices

Devices are the wave energy converters in your project workspace. You can assemble a particular device using any of the sub-components which you have created.

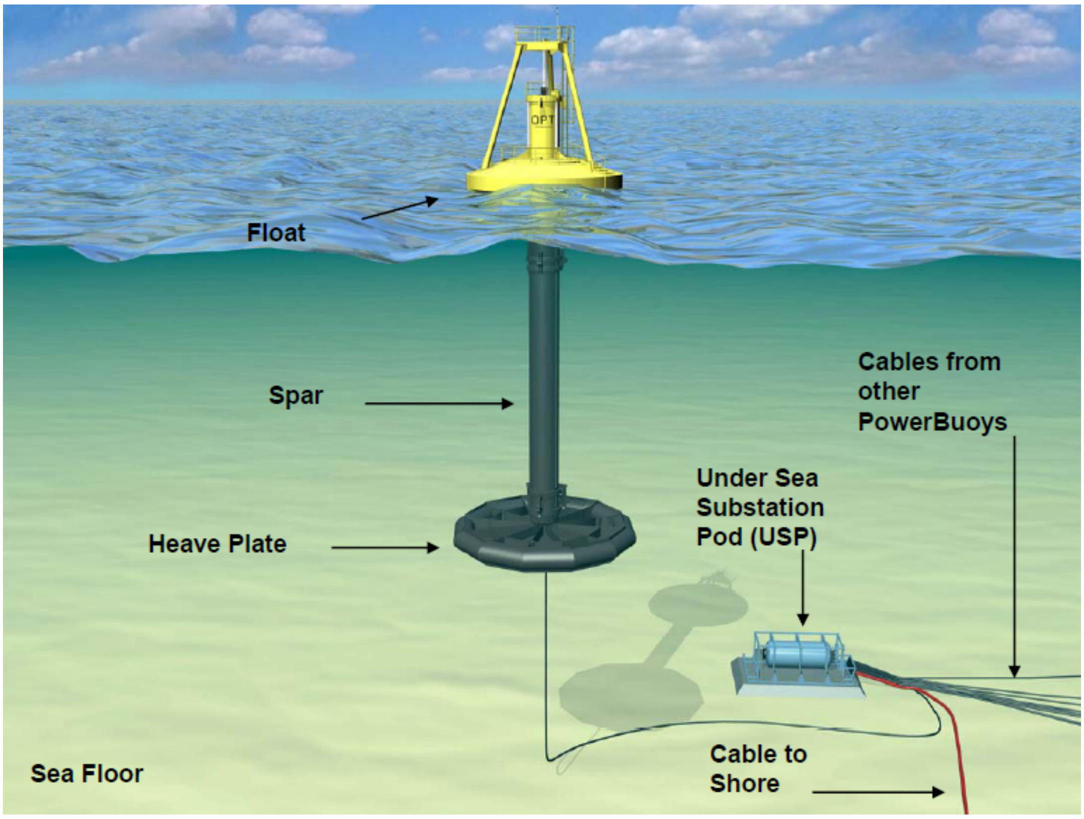
For further information on a particular wave energy converter, refer to the relevant link from the following list.

- Floating Dual-Body Point Absorber
- Submerged Tether-Moored Point Absorber

Floating Dual-Body Point Absorber

OVERVIEW

This device concept was inspired by a device known as *PowerBuoy* (Ocean Power Technologies, 2017), which is a two-body floating point absorber design. The device consists of a surface float which moves in response to wave motion, relative to a vertical column spar buoy which is attached to a large reaction plate submerged at a considerable depth below the mean water line. Generation of electrical power occurs predominately by harnessing oscillations of the surface float in the heave direction (the device is designed to accommodate relative heave motions of up to 4 metres along the shaft). Stability of the device is ensured via a spread mooring configuration, and it is designed to operate in water depths of between 40 and 100 metres.



Floating Dual-Body Point Absorber (Ocean Power Technologies, 2017)

This particular device was chosen as an illustrative concept, but Flexcom Wave is naturally capable of simulating other similar wave energy converters. Another similar device is *Wavebob*, which uses a submerged volume rather than a damping plate for stability.

If you have a specific device concept which is not currently accommodated by Flexcom Wave, you can always Migrate to Flexcom itself.

POWER

Input:	Description
Power Rating	The maximum power rating of this device.
Power-Take Off:	The power take-off component. You can select the relevant Power Take-Off from the list of components which you have already created in the Device Parts folder.

Notes

(a) The electrical power matrix is capped at the maximum power rating of the device, which helps to limit the size and cost of the electrical generator.

(b) The Power Take-Off component is modelled using a combination of spring, damper and rigid elements. In the case of a Floating Dual-Body Point Absorber, the power take-off mechanism is inserted between the respective centres of gravity of upper and lower floaters. You should ensure that the physical separation of these points in space (as governed by the inputs above) is consistent with the Geometry of the power take-off mechanism. Specifically, the separation must not be less than the *Contracted Length* and not greater than the Extended Length (which is equal to the sum of the *Contracted Length* and the *Maximum Stroke-Out*).

FLOATERS - UPPER AND LOWER

Input:	Description
Floater:	You can select the relevant Floater from the list of components which you have already created in the Device Parts folder.
Floater CoG below SWL:	The vertical distance between the Centre of Gravity of the floater and the still water line. A negative value indicates submergence.

MOORING

Input:	Description
Mooring Line:	The mooring line component. You can select the relevant Mooring Line from the list of components which you have already created in the Device Parts folder.
Anchor X:	The global X coordinate of the mooring anchor point.
Anchor Y:	The global Y coordinate of the mooring anchor point.
Anchor Z:	The global Z coordinate of the mooring anchor point.

Fairlead Connection:	The connection point on the Lower Floater. You can select the relevant Connection Point from the list of connections which you have already defined in the Floater component.

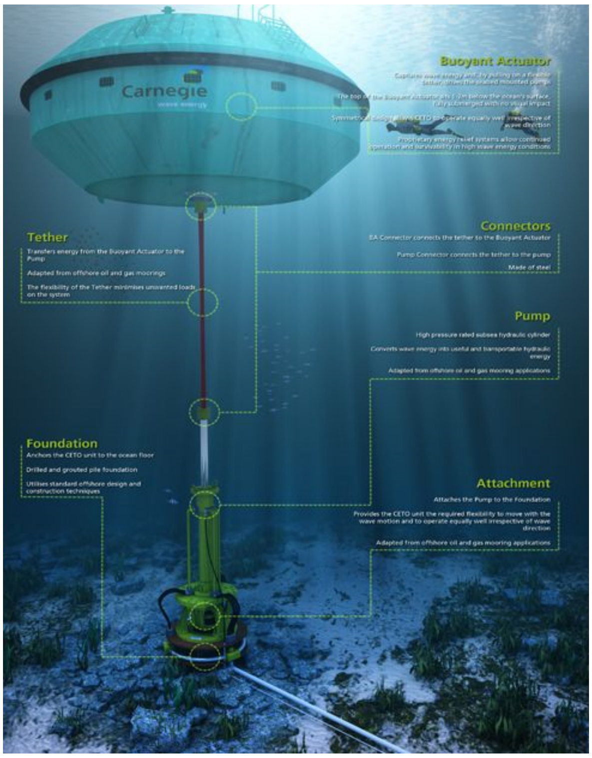
Notes

(a) In Flexcom, the global X axis is vertical, with the global Y and Z axes lying in a horizontal plane such that the system is right-handed. Refer to Axes and Displacements for further information.

Submerged Tether-Moored Point Absorber

OVERVIEW

This device concept was inspired by a device known as *CETO5* (Carnegie Clean Energy, 2017), which is a bottom-referenced point absorber design. It consists of a cylindrical buoy which is submerged a couple of metres below the still water line. The device is tethered to a hydraulic pump-based power take-off mechanism which is located on the seabed.



Submerged Tether-Moored Point Absorber (Carnegie Clean Energy, 2017)

This particular device was chosen as an illustrative concept, but Flexcom Wave is naturally capable of simulating other similar wave energy converters. Other similar devices include *SeaBased* and *CETO6* (where the relative positions of the tether and power take-off mechanism are reversed).

If you have a specific device concept which is not currently accommodated by Flexcom Wave, you can always Migrate to Flexcom itself.

POWER

Input:	Description
Power Rating	The maximum power rating of this device.
Power-Take Off:	The power take-off component. You can select the relevant Power Take-Off from the list of components which you have already created in the Device Parts folder.

Notes

(a) The electrical power matrix is capped at the maximum power rating of the device, which helps to limit the size and cost of the electrical generator.

(b) The Power Take-Off component is modelled using a combination of spring, damper and rigid elements. In the case of a Submerged Tether-Moored Point Absorber, the power take-off mechanism is inserted between the seabed and the lower end of the tether. You should ensure that the physical separation of these points in space (as governed by the *Floater CoG below SWL*, *Tether Length* and *Seabed X* inputs) is consistent with the Geometry of the power take-off mechanism. Specifically, the separation must not be less than the *Contracted Length* and not greater than the Extended Length (which is equal to the sum of the *Contracted Length* and the *Maximum Stroke-Out*).

FLOATER

Input:	Description

Floater:	You can select the relevant Floater from the list of components which you have already created in the Device Parts folder.
Floater CoG below SWL:	The vertical distance between the Centre of Gravity of the floater and the still water line. A positive value indicates submergence.

TETHERS

Input:	Description
Tether:	The supporting tether which connects the floating body to the power take-off component.

Environments

OCEAN

Input:	Description
Water Depth:	The water depth.
Water Density:	The mass per unit volume of seawater.
Gravity:	The acceleration due to gravity.

SEABED

Input:	Description
Seabed Stiffness:	The elastic stiffness per unit length of the seabed, in units of [Force]/[Distance]/[Distance].

Longitudinal Coefficient of Friction:	The coefficient of friction in the longitudinal direction. For any lines which contact the seabed, the local longitudinal direction is parallel to the line axis, while the local transverse direction is perpendicular to it.
Transverse Coefficient of Friction:	The coefficient of friction in the transverse direction.

Notes

(a) Refer to Seabed Interaction for further information.

WAVE SPECTRUM

Input:	Description
Spectrum Type:	The options are Pierson-Moskowitz and Jonswap.
Period Type:	The options are T_z (mean zero up-crossing period, the default), T_p (peak period) and T_e (energy period).
Number of Harmonics:	The number of harmonics to be used in the Spectrum Discretisation.
Model as Regular Wave:	This option allows you to simulate a random seastate using a regular wave approximation. Caution is strongly advised regarding the user application of this feature. As it provides a highly simplistic representation of the seastate, it is intended for initial screening studies only, and should not be used to assess the operational power performance of a wave energy converter.

Notes

(a) Flexcom has traditionally modelled wave spectra in terms of either T_z or T_p. If you specify a scatter diagram in terms of T_e, Flexcom converts T_e into T_z. Refer to Wave Energy Period for further information on the conversion process.

SCATTER DIAGRAM

Input:	Description
Hs:	The seastate significant wave height Hs.
Tz/Tp/Te:	The seastate mean zero up-crossing period Tz, peak period Tp, or energy period Te.
No. of Occurrences:	The number of occurrences in a given period (typically one year) of the particular combination of H_s and $T_z/T_p/T_e$ values which that cell represents.

Notes

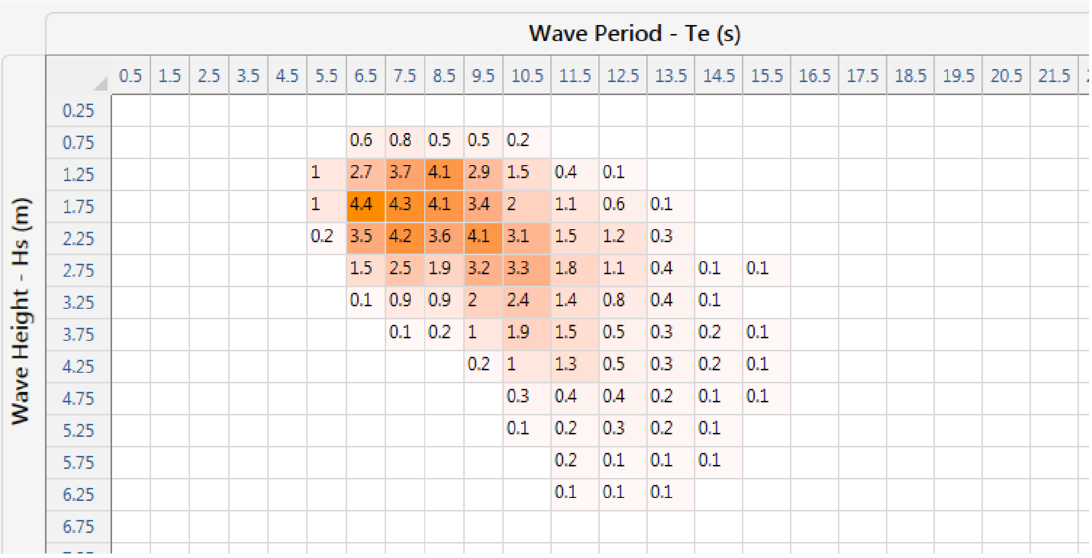
(a) Refer to Metocean Data and Seastate Blocks and Reference Seastates for further information regarding the specification of a scatter diagram and all its associated options.

Metocean Data

Firstly you need to obtain metocean data for your geographical location of interest, and then you may input the relevant scatter diagram information in Flexcom Wave.

- Flexcom Wave interfaces directly with ExceedenceFinance, a third-party software package which contains databases of metocean data for various geographical locations around the world. Provided you have a license for this software, you can quickly download wave resource information for your chosen location, and automatically import this data into Flexcom Wave. Refer to Import from Exceedence for further details. In this case, the Scatter Diagram is automatically populated for you.
- You may have access to existing metocean data from a previous project. If so, simply copy the Environment component from that project into your current project workspace.

- You may have access to an open source or commercial database of metocean data. If this data is available in spreadsheet format, it may be readily copied and pasted directly into the Scatter Diagram. If not, then you will need to manually type in the data.



Wave Period - Te (s)

Wave Height - Hs (m)	0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5
0.25																						
0.75							0.6	0.8	0.5	0.5	0.2											
1.25						1	2.7	3.7	4.1	2.9	1.5	0.4	0.1									
1.75						1	4.4	4.3	4.1	3.4	2	1.1	0.6	0.1								
2.25						0.2	3.5	4.2	3.6	4.1	3.1	1.5	1.2	0.3								
2.75							1.5	2.5	1.9	3.2	3.3	1.8	1.1	0.4	0.1	0.1						
3.25							0.1	0.9	0.9	2	2.4	1.4	0.8	0.4	0.1							
3.75								0.1	0.2	1	1.9	1.5	0.5	0.3	0.2	0.1						
4.25										0.2	1	1.3	0.5	0.3	0.2	0.1						
4.75											0.3	0.4	0.4	0.2	0.1	0.1						
5.25											0.1	0.2	0.3	0.2	0.1							
5.75												0.2	0.1	0.1	0.1							
6.25												0.1	0.1	0.1								
6.75																						

Scatter Diagram

The software also requires some additional information...

- The wave spectrum type used in the definition of the scatter diagram. The options currently supported are Pierson-Moskowitz and Jonswap.

- Whether your wave periods are defined in terms of T_z or T_p or T_e.

Note that the number of occurrences typically represents the number of “three-hour intervals” of that particular combination of H_s and $T_z/T_p/T_e$ during a 10 or 20-year period for example.

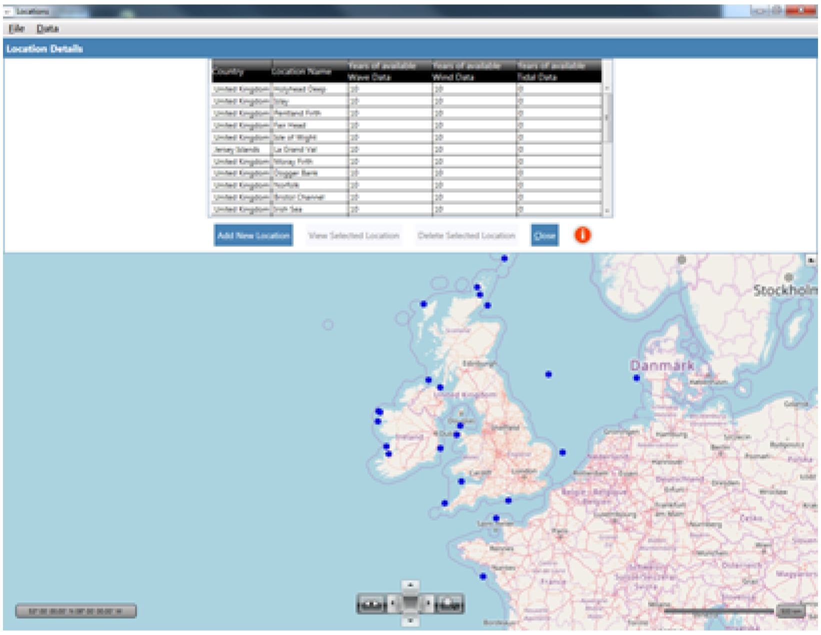
The duration of the period measuring the number of occurrences is in fact immaterial - the specified data is subsequently normalised into percentage occurrence values, so it is only the relative magnitudes which are of importance. If there is no occurrence of a particular combination of H_s and $T_z/T_p/T_e$, you simply leave the corresponding cell blank.

Note that as the latest version of Flexcom predicts an electrical power matrix (e.g. kW), rather than estimating total annual energy production (e.g. MWh), the number of occurrence entries in the scatter diagram are actually immaterial as they are not used by the software.

The simulation of an entire scatter diagram in the time domain can be quite time consuming, so you may wish to explore the possibility of nominating Seastate Blocks and Reference Seastates. This feature allows you to estimate simulation results for seastate combinations which you have not actually simulated, and may be useful during preliminary feasibility studies.

Import from Exceedence

Flexcom Wave interfaces directly with ExceedenceFinance, a third-party software package which contains databases of metocean data for various geographical locations around the world. Provided you have a license for this software, you can quickly download wave resource information for your chosen location, and automatically import this data into Flexcom Wave.



Environmental Location (image courtesy Exceedence)

Two methods of data transfer between the software products are provided.

- **File Transfer**. This involves a software user working in the ExceedenceFinance environment, explicitly exporting data to an external file, then switching over to Flexcom Wave and manually requesting the retrieval of this data using an import command.
- **Cloud Transfer**. This approach allows a Flexcom Wave user to request and obtain the relevant information directly from within the engineering environment, without having to actually open the Exceedence software.

Both options may be invoked from the *File* menu within Flexcom Wave.

Simulations

The Simulations folder allows you to perform numerical simulations and examine analysis results. In the Simulation Parameters tab you nominate a wave energy converter and choose the environmental conditions to which it is subjected. You may then proceed to Running Simulations. Following successful completion of the numerical simulations, you may examine important Simulation Results. Flexcom Wave automatically collates all pertinent information into 3D Plots and a summary spreadsheet for you to examine at your convenience. For example, you can view electric power for your wave energy converter as a function of both Hs and Tz/Tp/Te in a 3-dimensional space. It also possible to gain further insight by examining results from individual seastates such as time histories of floating body motions and mooring fairlead tensions. Crucially you can also Export to Exceedence, sending the power matrix computed by Flexcom's detailed engineering model back into ExceedenceFinance, and perform financial appraisals based on realistic power data for a given wave energy converter and ambient environment.

For further information on a particular area, refer to the relevant link from the following list.

- Simulation Parameters
- Running Simulations
- Simulation Results

Simulation Parameters

DEVICE & ENVIRONMENT

Input:	Description
Device:	The wave energy converter. You can select the relevant device from the list of components which you have already created in the Devices folder.

Environment:	The environmental conditions which the wave energy converter is subjected. You can select the relevant environment from the list of components which you have already created in the Environments folder.

SOLUTION TIME STEPPING

Input:	Description
Duration:	The analysis duration in seconds. Selection of an appropriate simulation duration depends on whether you are running a regular wave or random sea. Refer to Simulation Length for further details.
Time Step:	The fixed time step to be used in the analysis. It is important to choose a time-step which picks up necessary detail in excitation and response. Refer to Choice of Time Step for further details.
Ramp Time:	The time over which applied loads and displacements are gradually increased to their full value. Refer to Load Ramping for further details.

SCATTER DIAGRAM

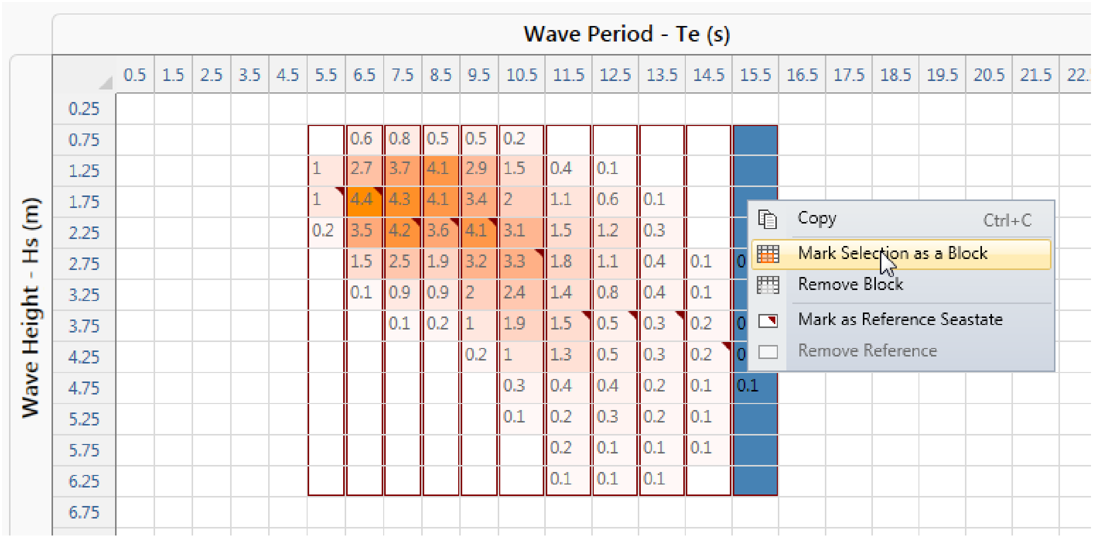
The simulation of an entire scatter diagram in the time domain can be quite time consuming, so you may wish to explore the possibility of nominating Seastate Blocks and Reference Seastates. This feature allows you to estimate simulation results for seastate combinations which you have not actually simulated, and may be useful during preliminary feasibility studies.

Flexcom Wave allows you to estimate simulation results for seastate combinations which you have not actually simulated. Based on some selected 'reference seastates' in your scatter diagram, the software estimates results for adjacent seastates based on an extrapolation technique. The simulation of an entire scatter diagram in the time domain can be quite time consuming, so this is a highly efficient solution technique which can save you considerable computational time. However given the inherent approximation involved, caution is strongly advised regarding the application of this feature.

This 'reference seastate' method can certainly be used during preliminary feasibility studies, and also during initial screening processes which consider various types of wave energy converter in a particular location. For computational efficiency, it may also be useful while you are fine-tuning a chosen device to suit the ambient environment. However once you are reasonably satisfied with a particular design configuration, it is strongly recommended that you explicitly simulate all seastate combinations in the scatter diagram. This will provide the most accurate estimate of a device's operational performance. A related benefit is that it will also allow you to quantify any inaccuracies associated with the 'reference seastate' method.

Within the Scatter Diagram itself...

- The *Mark Seastate Blocks* feature is used to group similar seastates into blocks. This task can be performed at any stage during the input of the scatter diagram data, but would typically be performed after all of the H_s and $T_z/T_p/T_e$ data has been input. You may also delete blocks subsequently, allowing you to change the way in which seastates are grouped in blocks without inputting the scatter diagram again in full.
- The *Mark Reference Seastates* feature is used to nominate reference seastates within each block. This task can only be performed after one or more seastate blocks have been defined. If you click on a seastate in a block where a reference seastate is already nominated, the previous nomination becomes deselected.



Block Selection

Some caution is advised regarding the number and selection of reference seastates which are chosen to approximately represent the full scatter diagram. This will require engineering judgment on your part, but the following general advice may be helpful.

- Generally speaking, any inaccuracies associated with the extrapolation process are likely to be more pronounced across different wave periods rather than wave amplitudes. Hence, it is not advisable to have blocks which span more than two different values of $T_z/T_p/T_e$.

- Structural responses at smaller wave amplitudes tend to be more linear than responses at higher amplitudes, so at smaller wave amplitudes the extrapolation technique is likely to be more accurate.

Running Simulations

OVERVIEW

Once you are satisfied with your selection of Device and Environment, you can proceed to running a simulation. Simply select the simulation of interest in the Project View, right-click and select 'Run Simulation' from the list of options.

- Flexcom Wave uses all the data which you have defined in the Device component and uses it to create a Flexcom model in standard Keyword File format.

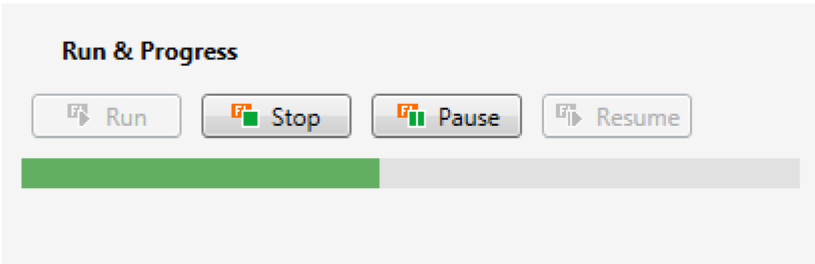
- The ambient environment is characterised by a Scatter Diagram in the Environment component. This provides Flexcom Wave with a large number of different combinations of Hs and Tz/Tp/Te. Each combination requires a separate numerical simulation, and the software automatically sets up the required input files, and neatly organises all the simulations into individual, appropriately named files within your project workspace.

- Depending on your Designation of Seastate Blocks and Nomination of Reference Seastates, Flexcom Wave may run all, or a selection of, seastates from your scatter diagram. If you are running only a selection of seastates for computational efficiency, Flexcom Wave estimates simulation results for adjacent seastates based on an Extrapolation Technique.

- Assuming the simulation has completed successfully, Simulation Results will be available for review and inspection.

SIMULATION PROGRESS

While the simulation is in progress, you will notice a green progress bar to indicate the simulation progress. Progress may appear quite slow, as each combination of Hs and Tz/Tp/Te requires a separate Time Domain simulation, and each simulation may consider 1800s (30 minutes) of simulation time or perhaps even longer.



Simulation Progress

When the simulation has fully completed, you can proceed to examining Simulation Results.

RUN SEQUENCE

In case you are wondering what is happening in the background; for each Simulation Component, Flexcom Wave actually creates a series of analysis files which run consecutively.

- Firstly an Initial Static Analysis is performed in order to determine the static equilibrium configuration of the system subject to gravity and buoyancy loads only.

- Depending on the complexity of the selected Device, Flexcom Wave may perform the initial static simulation in two stages in order to aid solution convergence. In such cases, the floating bodies are temporarily restrained using additional boundary conditions, which are subsequently removed in a Restart Static Analysis. Floating systems can be sensitive to minor changes in displacement, so this second stage is typically performed as a Quasi-Static Analysis.

- Finally the ambient environmental loads are simulated in a series of separate Time Domain Analyses. Each combination of Hs and Tz/Tp/Te requires a separate simulation, so the software neatly organises these into appropriately named sub-folders within your project workspace.

Existing Flexcom users will be familiar with the concept of an Analysis Job, which typically contains a series of individual analysis files which run consecutively. The procedure is very similar here, except that Flexcom Wave has automatically set up the analysis job for increased convenience to the user.

Simulation Results

INTRODUCTION

Following successful completion of the numerical simulations, you may examine important simulation results. Flexcom Wave provides you with various options for examining key parameters which are typically associated with wave energy converters...

- Annual energy production

- Electrical power matrix

- Floating body motions

- Mooring fairlead tensions

In the current version of Flexcom Wave, the above information is presented as standard. If you require any additional outputs, you could always Migrate to Flexcom itself. Customised post-processing, where users will be able to request specific outputs, is planned for inclusion in a future version.

HIGH LEVEL SUMMARY

Flexcom Wave simulates a scatter diagram of a relatively large number of different combinations of Hs and Tz/Tp/Te. For your convenience, the software automatically collates all pertinent information for you to inspect at a high level. The following options are provided.

- **Power Matrix and Annual Energy Production**. The electrical power matrix (in kW) is presented in tabular format, along with the annual energy production (in MWh). The AEP figure is obtained by combining the electrical power matrix with the probability distribution defined by the Wave Scatter Diagram. Specifically...

$$AEP = 24 * 365 * \sum_{i=1}^{n} Pe(Hs_i, Te_i) * p(Hs_i, Te_i) \quad (1)$$

where Pe (Hs_i, Te_i) is the electrical power absorbed by the device for a given seastate defined by significant wave height Hs_i and wave energy period Te_i, and p (Hs_i, Te_i) is the probability of that seastate occurring at the site location. Assuming you are satisfied with the power matrix, you can Export to Exceedence.

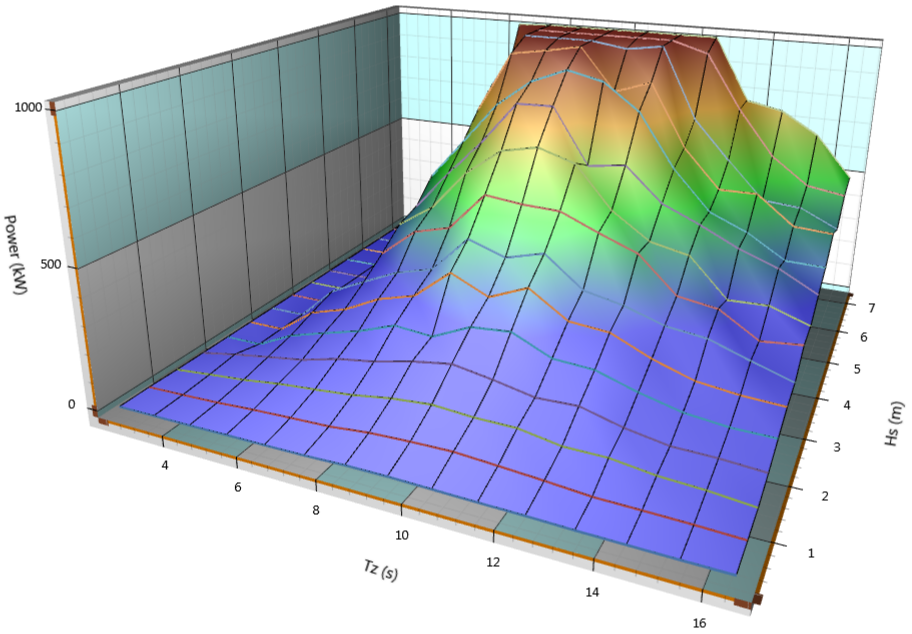
Power Matrix (kW) Annual Energy Production: 667.27 (MWh)

Wave Period - Te (s) / Wave Height - Hs (m)

Hs \ Te	0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	22.5	23.5	24.5
0.25																									
0.75						7	8	8	8	7	6	6	6	5	5	5									
1.25						19	21	22	22	19	17	17	16	15	14	13									
1.75						38	41	44	42	37	34	33	31	29	27	26									
2.25						62	68	73	70	61	56	55	50	47	44	43									
2.75						93	102	109	104	92	84	82	75	71	66	64									
3.25						130	143	152	146	128	117	114	105	99	92	89									
3.75						173	190	202	194	170	156	152	140	131	123	118									
4.25						222	244	260	249	219	200	196	180	169	158	152									
4.75						277	286	286	286	273	250	244	225	211	197	190									
5.25						286	286	286	286	286	286	286	275	258	241	232									
5.75						286	286	286	286	286	286	286	286	286	286	278									
6.25						286	286	286	286	286	286	286	286	286	286	286									
6.75																									

Power Matrix

- **View 3D Plots**. This option allows you to view key parameters such as electric power as a function of both Hs and Tz/Tp/Te in a 3-dimensional space.



3D Surface Plot of Electrical Power

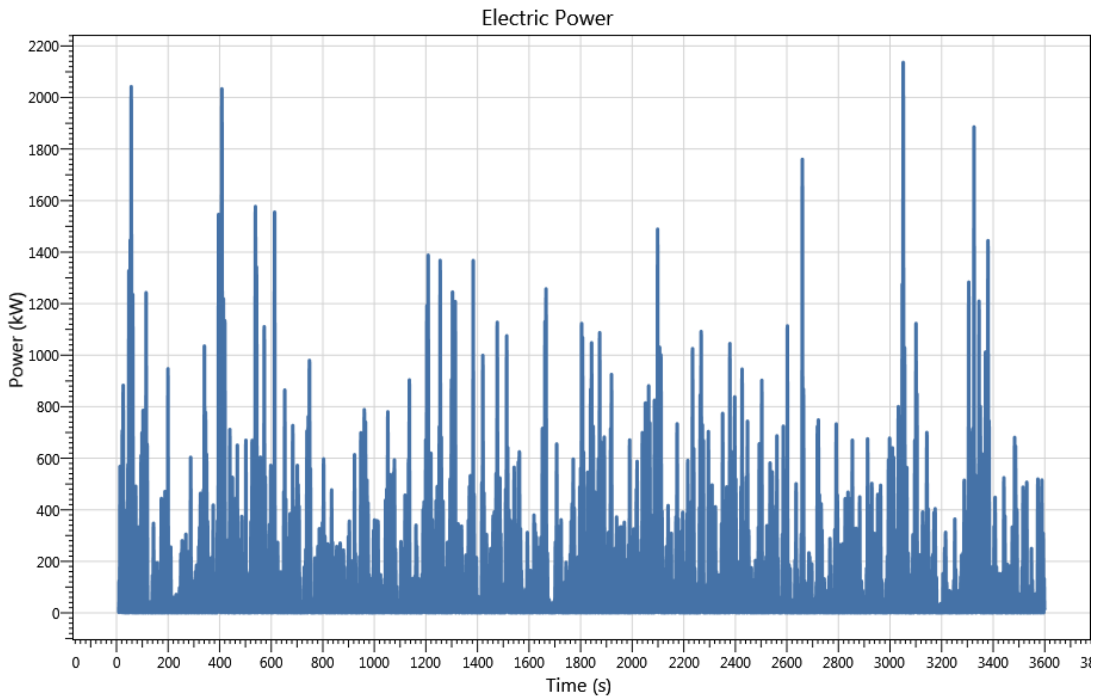
- **View Spreadsheet**. This option allows you to examine a spreadsheet (known as a Summary Collation Spreadsheet in Flexcom terminology), which assembles summary output data from all of the seastates in the scatter diagram. For each of the output parameters listed above (i.e. power, motions and tensions), the spreadsheet presents maximum, minimum, mean, range and standard deviation values for each individual simulation.

INDIVIDUAL SEASTATES

While the High Level Summary outlined above provides an excellent overview, you may wish to seek further insight by examining results from any of the individual seastates. For example, if the maximum mooring line tension appears excessive for a particular combination of Hs and Tz/Tp/Te, you may be interested in viewing a time history of effective tension to check if the predicted maximum represents an isolated peak (which could be a numerical issue rather than a physical phenomenon) or is attained regularly throughout the simulation. Based on additional insights, you may wish to perform some sensitivity studies, to examine the effect on key solution parameters of reducing the analysis time step for example.

Firstly you select the seastate of interest by identifying the relevant combination of Hs and Tz/Tp/Te, and then you can invoke any or all of the following options.

- **View Plots**. This option allows you to view the variation of key parameters such as electric power as a function of time.
- **View Structural Animation**. This option allows you to view an animation of the wave energy converter in operation. The [Model View] provides rotate, pan and zoom options, allowing you to move freely and examine any region of the model in detail.
- **View Output File**. This option allows you to examine a text file, known as the Main Output File in Flexcom terminology. It contains an echo of all the specified input data. It also provides information regarding Convergence Ratios in the case of unsuccessful simulations, which can be useful for debugging solution problems.



Sample Time History of Power Output

You can export the power matrix computed by Flexcom's detailed engineering model into ExceedenceFinance (a third-party techno-financial modelling tool), where you can perform financial appraisals based on realistic power data for a given wave energy converter and ambient environment.

Two methods of data transfer between the software products are provided.

- **File Transfer**. This involves a Flexcom Wave software user exporting data to an external file. The user would then have to manually request the retrieval of this data using an import command within ExceedenceFinance.

- **Cloud Transfer**. This approach allows a Flexcom Wave user to transfer the information directly over the cloud, directly updating the relevant database on the Exceedence side, without having to actually open the Exceedence software.

Both options may be invoked from the *File* menu within Flexcom Wave.

1.7.8.2 Migrate to Flexcom

While simulations may be performed directly within the Flexcom Wave module itself, experienced Flexcom users may prefer to use the specialised module for initial model construction only, and then switch to the main Flexcom environment as it provides greater modelling flexibility. Migrating is as easy as flicking a switch, you simply select the 'Open as Flexcom Project' from the *Simulations* menu. The wave energy simulator module produces a well-structured, heavily parameterised keyword file which can be subsequently customised to meet your own individual requirements.

The structural components of the model, such as the floater, PTO and mooring lines, are all constructed in standard fashion using established Flexcom modelling capabilities. Refer to Model Building for further information - this article outlines a typical series of steps to follow when building a model of a wave energy device.

1.8 Data Inputs

This section contains information on all Flexcom data inputs. See also Parameters, Equations and Variations in the Keyword Parameterisation Section.

- `$AERODYN` corresponds to AeroDyn, which is a time-domain wind turbine aerodynamics module that has been developed by the National Renewable Energy Laboratory of the United States Department of Energy. Refer to Wind Turbine Modelling for further details.

- $CLEAR corresponds to Clear, an ancillary module of Flexcom, which performs clearance/interference postprocessing calculations. Refer to Clearance & Interference Postprocessing for further details.

- $CODE CHECKING corresponds to the code checking facility, which allows you to check analysis results against specific design codes/procedures. Refer to Code Checking for further details.

- $DATABASE POSTPROCESSING corresponds to the database postprocessing facility, which generally represents the most comprehensive postprocessing resource. Refer to Database Postprocessing for further details.

- $HISTOGRAM corresponds to Histogram, which is a frequency domain general cycle counting tool. Refer to Histogram Overview for further details.

- $LIFE FREQUENCY corresponds to LifeFrequency, which is a frequency domain fatigue postprocessor to Flexcom. Refer to Frequency Domain Fatigue Analysis for further details.

- $LIFETIME CYCLE corresponds to LifeTime Mode 2. LifeTime is an ancillary module to Flexcom which performs time domain fatigue analysis (Mode 1) and general cycle counting (Mode 2). Refer to Cycle Counting Analysis (Mode2) for further details.

- $LIFETIME FATIGUE corresponds to LifeTime Mode 2. LifeTime is an ancillary module to Flexcom which performs time domain fatigue analysis (Mode 1) and general cycle counting (Mode 2). Refer to Cycle Counting Analysis (Mode2) for further details.

- $LOAD CASE includes data such as environmental parameters (e.g. current and waves), boundary conditions of various kinds, internal fluid loading, and the analysis type and solution parameters.

- $MODEL includes data such as the finite element discretisation, structural and hydrodynamic properties, plus any other inputs which characterise the initial model configuration (e.g. initial vessel position, seabed properties, ocean depth etc.) – basically any information which cannot logically change from run to run.

- $MODES corresponds to Modes, an ancillary module of Flexcom, which performs modal analysis. Refer to Modal Analysis for further details.

- `$MODES POSTPROCESSING` - This section corresponds to the postprocessing of modal analyses. Refer to Modal Analysis Postprocessing for further details.

- `$PREPROCESSOR` - This section corresponds to the keyword parameterisation facility, which allows you to model a series of load case variations about a base model. Refer to Keyword Parameterisation for further details.

- `$SHEAR7` corresponds to prediction of VIV fatigue damage by interfacing with SHEAR7 program developed by a team at MIT. Refer to SHEAR7 Interface for further details.

- `$SUMMARY COLLATE` corresponds to the summary collation postprocessing module, which allows you to collate the summary postprocessing results across a range of different time domain analyses. Refer to Summary Collate for further details.

- `$SUMMARY POSTPROCESSING` corresponds to the summary postprocessing facility, which allows you to extract pertinent results (e.g. maximum/minimum values) from time domain analyses in succinct tabular format. Refer to Summary Postprocessing for further details.

- `$SUMMARY WAVE SCATTER` corresponds to the Summary Wave Scatter feature, which allows you to generate Summary Database Files for seastate combinations which you have not actually simulated. Based on some selected 'reference seastates' in your scatter diagram, Flexcom estimates simulation results for adjacent seastates based on an extrapolation technique. You may then use the Summary Postprocessing Collation to collate information from all seastates. Refer to Summary Wave Scatter for further details.

- `$TIMETRACE POSTPROCESSING` corresponds to the timetrace postprocessing facility, which is mainly used in the area of time domain fatigue analysis. Refer to Timetrace Postprocessing for further details.

- `$UNIVERSAL` contains universal keywords/directives which can occur in any section of the keyword file (e.g. `$MODEL`, `$LOAD CASE` etc.).

1.8.1 $AERODYN

This section corresponds to the wind turbine modelling feature in Flexcom. Refer to AeroDyn Overview for further information.

This section contains the following keywords:

- *AEROFOIL INFO is used to specify the aerofoil data input file information.
- *BEDDOES-LEISHMANN is used to specify the Beddoes Leishmann unsteady aerodynamic modelling options.
- *BEM THEORY is used to specify the blade element momentum (BEM) theory options.
- *BLADE PROPERTIES is used to specify the rotor/blade properties.
- *DBEM THEORY is used to specify the dynamic blade element momentum (DBEM) theory options.
- *DESCRIPTION is used to specify AeroDyn header information.
- *ENVIRONMENTAL is used to specify environmental conditions.
- *GENERAL is used to specify general analysis options.
- *OLAF THEORY is used to specify the cOnvecting LAgrangian Filaments (OLAF) theory options (also known as free vortex wake model).
- *OUTPUTS is used to specify the AeroDyn output settings.
- *TOWER INFLUENCE is used to specify the tower influence and drag properties.

1.8.1.1 *AEROFOIL INFO

PURPOSE

To specify the aerofoil data input file information.

THEORY

Refer to AeroDyn Overview for further information.

KEYWORD FORMAT

A block of two lines of data, defining the format of the tables of static aerofoil coefficients within each of the aerofoil input files as follows:

```
AFTABMOD=Aerofoil Interpolation, ALPHA=AoA Column, CL=Lift Coeff Column, CD=I
FILES_TO_USE=Number Aerofoil Files
```

This is then followed by one line of data containing the name of the first aerofoil input file. This may be repeated as often as required to define the full list of aerofoil input file names.

```
FILE=Aerofoil File
```

AeroDyn will interpolate on AoA (Angle of Attack) using the data provided via linear interpolation or via cubic splines, depending on the setting of `InterpOrd` input in the aerofoil input file. If *Aerofoil Interpolation* is set to 1, only the first airfoil table in each file will be used. If *Aerofoil Interpolation* is set to 2, AeroDyn will find the airfoil tables that bound the computed Reynolds number, and linearly interpolate between the tables, using the logarithm of the Reynolds numbers. If *Aerofoil Interpolation* is set to 3, it will find the bounding airfoil tables based on the UserProp field and linearly interpolate the tables based on it. Note that OpenFAST currently sets the `UserProp` input value to 0 unless the DLL controller is used and sets the value, so using this feature may require a code change.

AoA Column, Lift Coeff Column, Drag Coeff Column & Pitching Moment Column are column numbers in the tables containing the AoA (Angle of Attack), lift-force coefficient, drag-force coefficient, and pitching-moment coefficient, respectively (normally these are *1, 2, 3,* and *4*, respectively). If aerodynamic pitching-moment terms are neglected (via setting `USEBLCM=0` under `*BLADE PROPERTIES`), *Pitching Moment Column* may be set to *0*.

Min Pressure Column is the column number containing the minimum pressure coefficient for cavitation checks in hydrokinetic rotors. This feature is not yet available in AeroDyn (v15.02.04), so you should specify *0* for this entry currently.

Number Aerofoil Files specifies the number of aerofoil data input files to be used, followed by *Number Aerofoil Files* lines of file names.

TABLE INPUT

Aerofoil Information

Input:	Description

Aerofoil Interpolation:	The interpolation mode for the airfoil tables. AeroDyn will interpolate on AoA (Angle of Attack) using the data provided via linear interpolation or via cubic splines, depending on the setting of InterpOrd input in the aerofoil input file. If *Aerofoil Interpolation* is set to 1, only the first airfoil table in each file will be used. If *Aerofoil Interpolation* is set to 2, AeroDyn will find the airfoil tables that bound the computed Reynolds number, and linearly interpolate between the tables, using the logarithm of the Reynolds numbers. If *Aerofoil Interpolation* is set to 3, it will find the bounding airfoil tables based on the *UserProp* field and linearly interpolate the tables based on it. Note that OpenFAST currently sets the *UserProp* input value to 0 unless the DLL controller is used and sets the value, so using this feature may require a code change.
Angle of Attack Column Index:	The number of the column in the tables of static aerofoil coefficients within each of the aerofoil input files which represents Angle of Attack.
Lift-Force Coefficient Column Index:	The number of the column in the tables of static aerofoil coefficients within each of the aerofoil input files which represents Lift-Force Coefficient.
Drag-Force Coefficient Column Index:	The number of the column in the tables of static aerofoil coefficients within each of the aerofoil input files which represents Drag-Force Coefficient.
Pitching-Moment Coefficient Column Index:	The number of the column in the tables of static aerofoil coefficients within each of the aerofoil input files which represents Pitching-Moment Coefficient. If aerodynamic pitching-moment terms are neglected (via setting USEBLCM=0 under `*BLADE PROPERTIES`), this column index may be set to zero.

Minimum Pressure Coefficient Column Index:	The number of the column in the tables of static aerofoil coefficients within each of the aerofoil input files which represents Minimum Pressure Coefficient (for cavitation checks in hydrokinetic rotors). This feature is not yet available in AeroDyn (v15.02.04), so you should specify zero for this entry currently.
Total Number of Aerofoil Input Files:	The total number of aerofoil input files to be used. The entry should be consistent with the number of aerofoil input file names listed under Aerofoil Files.

Aerofoil Files

Input:	Description
Aerofoil File Name:	The name of the n^{th} aerofoil input file. This entry may be repeated as often as required to define the full list of aerofoil input file names. The total number of files listed should be consistent with the Total Number of Aerofoil Input Files.

1.8.1.2 *BEDDOES-LEISHMAN

PURPOSE

To specify the Beddoes Leishmann unsteady aerodynamic modeling options.

THEORY

Refer to AeroDyn Overview for further information.

KEYWORD FORMAT

Two lines of data, defining Beddoes Leishmann options as follows:

```
UAMOD=Unsteady Model
FLOOKUP=Separation Distance Lookup
```

`Unsteady Model` determines the unsteady aerodynamic model. Setting `Unsteady Model` to *1* enables the original theoretical developments of Beddoes-Leishmann, *2* enables the extensions to Beddoes-Leishmann developed by González, and *3* enables the extensions to Beddoes-Leishmann developed by Minnema/Pierce.

`Separation Distance Lookup` determines how the nondimensional separation distance value, *f'*, will be calculated. When `Separation Distance Lookup` is set to *1*, *f'* is determined via a lookup into the static lift-force coefficient and drag-force coefficient data. Note that `Separation Distance Lookup` must be *1* in this version of AeroDyn (v15.02.04).

TABLE INPUT

Input:	Description
Unsteady Aerofoil Aerodynamics:	This option allows you to enable the original theoretical developments of Beddoes-Leishmann, the extensions to Beddoes-Leishmann developed by González, or the extensions to Beddoes-Leishmann developed by Minnema/Pierce.
Separation Distance Lookup:	This option allows you to specify how the nondimensional separation distance value, f', will be calculated. The only option currently (AeroDyn v15.02.04) is for f' to be determined via a lookup into the static lift-force coefficient and drag-force coefficient data. Using best-fit exponential equations is not yet available.

1.8.1.3 *BEM THEORY

PURPOSE

To specify the blade element momentum (BEM) theory options.

THEORY

Refer to AeroDyn Overview for further information.

KEYWORD FORMAT

Six lines of data, defining blade element momentum theory parameters as follows:

```
SKEWMOD=Skewed-Wake Correction
SKEWMODFACTOR=Skewed-Wake Correction Factor
TIPLOSS=Tip Loss, HUBLOSS=Hub Loss
TANIND=Tangential Induction
AIDRAG=Axial-Induction Drag, TIDRAG=Tangential-Induction Drag
INDTOLER=Convergence Threshold
MAXITER=Maximum Iterations
```

Set `Skewed-Wake Correction` to 1 to use the uncoupled BEM solution technique without an additional skewed-wake correction, or 2 to include the Pitt/Peters correction model.

`Skewed-Wake Correction Factor` is used only when SkewMod = 1. Enter a scaling factor to use in the Pitt/Peters correction model, or enter `DEFAULT` to use the default value of 15 PI /32.

Set `Tip Loss` to *1* to include the Prandtl tip-loss model, or *0* to disable it. Likewise, set `Hub Loss` to *1* to include the Prandtl hub-loss model, or *0* to disable it.

Set `Tangential Induction` to *1* to include tangential induction (from the angular momentum balance) in the BEM solution, or *0* to neglect it. Set `Axial-Induction Drag` to *1* to include drag in the axial-induction calculation, or *0* to neglect it. If `Tangential Induction` = *1*, set `Tangential-Induction Drag` to *1* to include drag in the tangential-induction calculation, or *0* to neglect it.

`Convergence Threshold` sets the convergence threshold for the iterative nonlinear solve of the BEM solution. `Maximum Iterations` determines the maximum number of iterations steps in the BEM solve.

TABLE INPUT

Input:	Description
Skewed-Wake Correction:	The option are *Uncoupled* (uncoupled BEM solution technique without an additional skewed-wake correction), or the *Pitt/Peters* correction model.

Skewed-Wake Correction Factor:	This factor is used only when *Skewed-Wake Correction* is set to *Pitt/Peters*. Enter a scaling factor to use in the Pitt/Peters correction model, or enter DEFAULT to use the default value of 15 PI /32.
Prandtl Tip-Loss:	This option allows you to include the Prandtl tip-loss model, or to disable it.
Prandtl Hub-Loss:	This option allows you to include the Prandtl hub-loss model, or to disable it.
Tangential Induction:	This option allows you to include tangential induction (from the angular momentum balance) in the BEM solution, or to neglect it.
Axial-Induction Drag:	This option allows you to include drag in the axial-induction calculation, or to neglect it.
Tangential-Induction Drag:	This option allows you to include drag in the tangential-induction calculation, assuming tangential induction itself is being included, or to neglect it.
Convergence Threshold:	The convergence threshold for the iterative nonlinear solve of the BEM solution. The nonlinear solve is in terms of the inflow angle, but the convergence threshold represents the tolerance of the nondimensional residual equation, with no physical association possible. 'DEFAULT' may be specified rather than a numerical value, allowing AeroDyn to select an appropriate default value as recommended by NREL.
Maximum Iterations:	The maximum number of iterations steps in the BEM solve. If the residual value of the BEM solve is not less than or equal to *Convergence Threshold* in *Maximum Iterations*, AeroDyn will exit the BEM solver and return an error message.

1.8.1.4 *DBEM THEORY

PURPOSE

To specify the dynamic blade element momentum (DBEM) theory options.

THEORY

Refer to AeroDyn Overview for further information.

KEYWORD FORMAT

Two lines of data, defining dynamic blade element momentum theory parameters as follows:

```
DBEMT_MOD=DBEMT Model
TAU1_CONST=DBEMT time constant
```

Set *DBEMT Model* to 1 for the constant-tau1 model, set *DBEMT Model* to 2 to use a model where tau1 varies with time, or set *DBEMT Model* to 3 to use a continuous-state model with constant tau1.

If *DBEMT Model*=1 (constant-tau1 model) or *DBEMT Model*=3 (continuous-state constant-tau1 model), set *DBEMT time constant* to the time constant to use for DBEMT.

TABLE INPUT

Input:	Description
DBEMT Model:	The options are *Constant Tau1*, *Time Varying Tau1* and *Continuous State*.
DBEMT time constant:	The time constant to be used when DBEMT Model is *Constant Tau1* or *Continuous State*.

1.8.1.5 *DESCRIPTION

PURPOSE

To specify a line of descriptive information about the AeroDyn simulation.

THEORY

Refer to AeroDyn Overview for further information.

KEYWORD FORMAT

A single line containing some descriptive text about the AeroDyn simulation.

```
Description Line
```

TABLE INPUT

Input:	Description
Description:	Some descriptive text about the AeroDyn simulation. This is for your own reference, but is not used by the software.

1.8.1.6 *ENVIRONMENT

PURPOSE

To specify environmental conditions.

THEORY

Refer to AeroDyn Overview for further information.

KEYWORD FORMAT

Three lines of data, defining environmental conditions as follows:

```
AIRDENS=Air Density
KINVISC=Kinematic Viscosity
SPDSOUND=Speed of Sound
PATM=Atmospheric Pressure
PVAP=Vapour Pressure
FLUIDDEPTH=Fluid Depth
```

A typical value of `Air Density` is around 1.225 kg/m^3 for air.

A typical value of `Kinematic Viscosity` is around 1.460E-5 m^2/s for air.

A typical value of `Speed of Sound` in air is around 340.3 m/s.

The last three parameters (`Atmospheric Pressure`, `Vapour Pressure` & `Fluid Depth`) are only used when a *Cavitation Check* is enabled for MHK (Marine and Hydrokinetic Technology) turbines. As Flexcom is currently designed to work with wind turbines only, these options are not relevant.

TABLE INPUT

Input:	Description
Air Density:	The air density. A typical value is around 1.225 kg/m^3 for air.
Kinematic Viscosity:	The kinematic viscosity of the air (used in the Reynolds number calculation). A typical value is around 1.460E-5 m^2/s for air.
Speed of Sound:	The speed of sound in air (used to calculate the Mach number within the unsteady airfoil aerodynamics calculations). A typical value is around 340.3 m/s.
Atmospheric Pressure:	The atmospheric pressure above the free surface. This is typically around 101,325 Pa.
Vapour Pressure:	The vapor pressure of the fluid. For seawater this is typically around 2,000 Pa.
Fluid Depth:	The distance from the hub center to the free surface.

NOTES

(a) The last three parameters (*Atmospheric Pressure*, *Vapour Pressure* & *Fluid Depth*) are only used when a *Cavitation Check* is enabled for MHK (Marine and Hydrokinetic Technology) turbines. As Flexcom is currently designed to work with wind turbines only, these options are not relevant.

1.8.1.7 *GENERAL

PURPOSE

To specify general analysis options.

THEORY

Refer to AeroDyn Overview for further information.

KEYWORD FORMAT

Seven lines of data, defining general analysis options as follows:

```
ECHO=Input Echo
DTAERO=Aerodynamic Time Step
WAKEMOD=Rotor Wake/Induction Effects
AFAEROMOD=Aerofoil Aerodynamics
TWRPOTENT=Tower Potential-Flow Influence
TWRSHADOW=Tower Shadow Influence
TWRAERO=Tower Drag Loads
FROZENWAKE=Frozen Wake
CAVITCHECK=Cavitation Check
COMPAA=Aero-acoustics
AA FILE=AA File Name
INPUT FILE=AeroDyn Primary Input File Name
```

Set `Input Echo` to *1* to instruct AeroDyn to echo the contents of the primary, aerofoil, and blade input files (useful for debugging errors in the input files). Set `Input Echo` to *0* if no input echo is required.

`Aerodynamic Time Step` is not currently an input, as it is automatically determined by Flexcom's structural analysis timestep, so it assumes a value of `DEFAULT` always. In standard offshore applications, appropriate time variables are typically dependent on the ambient seastate. For wind turbine modelling, NREL recommend that the solution time step for aerodynamic calculations be set such that there are at least 200 time steps per rotor revolution.

Set `Rotor Wake/Induction Effects` to *0* if you want to disable rotor wake/induction effects, or *1* to include these effects using the (quasi-steady) blade element momentum theory model. You can also set `Rotor Wake/Induction Effects` to 2, to use a dynamic blade element momentum theory model (DBEMT), also referred to as dynamic inflow or dynamic wake model. Or set it to 3 to use the free vortex wake model, also referred to as OLAF.

Set `Aerofoil Aerodynamics` to *1* to include steady blade aerofoil aerodynamics, or *2* to enable unsteady blade aerofoil aerodynamics.

Set `Tower Potential-Flow Influence` to *0* to disable the potential-flow influence of the tower on the wind local to the blade, *1* to enable the standard potential-flow model, or *2* to include the Bak correction in the potential-flow model.

Set `Tower Shadow Influence` to 0 to disable to the tower shadow model, 1 to enable the Powles tower shadow model, or 2 to use the Eames tower shadow model. These models calculate the influence of the tower on the flow local to the blade based on the downstream tower shadow model. If the tower influence from potential flow and tower shadow are both enabled, the two influences will be superimposed.

Set `Tower Drag Loads` to *1* to calculate wind drag loads on the tower, or *0* to disable these effects.

During linearisation analyses with AeroDyn coupled OpenFAST and BEM enabled (`Rotor Wake/Induction Effects` = 1), set the `Frozen Wake` flag to 1 to employ frozen-wake assumptions during linearisation (i.e. to fix the axial and tangential induces velocities, and, at their operating-point values during linearisation) or 0 to recalculate the induction during linearisation using BEM theory.

Set `Cavitation Check` to 1 to perform a cavitation check for MHK (Marine and Hydrokinetic Technology) turbines or 0 to disable this calculation. If `Cavitation Check` is 1, `Aerofoil Aerodynamics` must be set to 1 because the cavitation check does not function with unsteady airfoil aerodynamics. As Flexcom is currently designed to work with wind turbines only, `Cavitation Check` should always be set to 0.

Set `Aero-acoustics` to 1 to run aero-acoustic calculations. This option is only available for `Rotor Wake/Induction Effects` = 1 or 2. Refer to NREL AeroDyn documentation for information on how to use this feature.

The `AA File Name` input is used to specify the input file for the aeroacoustics sub-module. Refer to NREL AeroDyn documentation for information on how to use this feature.

The `AeroDyn Primary Input File Name` input is used to specify the name of the AeroDyn primary input file that is generated. The file name should have a .dat file extension. This entry is optional and, if omitted, defaults to the name of the keyword file with the .dat extension.

TABLE INPUT

Input:	Description
Input Echo:	This option allows you to instruct AeroDyn to echo the contents of the primary, aerofoil, and blade input files (useful for debugging errors in the input files). The echo file has the naming convention of OutRootFile.AD.ech.
Aerodynamic Time Step:	The time step for the aerodynamic calculations. This is not currently an input, as it is automatically determined by Flexcom's structural analysis timestep. In standard offshore applications, appropriate time variables are typically dependent on the ambient seastate. For wind turbine modelling, NREL recommend that the solution time step for aerodynamic calculations be set such that there are at least 200 time steps per rotor revolution.

Rotor Wake/Induction Effects:	The options are *None* (rotor wake/induction effects disabled), *BEMT* (quasi-steady, blade element momentum theory), *DBEMT* (dynamic blade element momentum theory model), and *OLAF* (free vortex wake model).
Aerofoil Aerodynamics:	This option allows you to select steady blade aerofoil aerodynamics or unsteady aerofoil aerodynamics.
Tower Potential-Flow Influence:	This option allows you to disable the potential-flow influence of the tower on the wind local to the blade, to enable the standard potential-flow model, or to include the Bak correction in the potential-flow model.
Tower Shadow Influence:	This option allows you to include the influence of the tower on the wind local to the blade based on the downstream tower shadow model. The options are *None*, *Powles model* or *Eames model*. If the tower influence from potential flow and tower shadow are both enabled, the two influences will be superimposed.
Tower Drag Loads:	This option allows you to calculate wind drag loads on the tower, or disable these effects.
Frozen Wake:	This option allows you to employ frozen-wake assumptions during linearisation (i.e. to fix the axial and tangential induces velocities, and, at their operating-point values during linearisation) or to recalculate the induction during linearisation using BEM theory.
Cavitation Check:	This option allows you to perform a cavitation check for MHK (Marine and Hydrokinetic Technology) turbines or to disable this calculation. If the cavitation check is invoked, *Aerofoil Aerodynamics* must be set to *Steady* because the cavitation check does not function with unsteady airfoil aerodynamics. As Flexcom is currently designed to work with wind turbines only, *Cavitation Check* should always be set to No.

Aero-acoustics:	This option allows you to run aero-acoustic calculations. This option is only available for *Rotor Wake/Induction Effects* of BEMT or DBEMT. Refer to NREL AeroDyn documentation for information on how to use this feature.
Aero-acoustic File Name:	This input allows you to specify the input file for the aeroacoustics sub-module. Refer to NREL AeroDyn documentation for information on how to use this feature.
AeroDyn Input File:	The name of the AeroDyn primary input file that is generated. The file name should have a .dat file extension. This entry is optional and, if omitted, defaults to the name of the keyword file with the .dat extension.

1.8.1.8 *OLAF THEORY

PURPOSE

To specify the cOnvecting LAgrangian Filaments (OLAF) theory options (also known as free vortex wake model).

THEORY

Refer to AeroDyn Overview for further information.

KEYWORD FORMAT

One line of data, defining the name of the OLAF input file. This file contains settings for the free vortex wake model, and is only used when *`Rotor Wake/Induction Effects`* = 3.

```
OLAF FILE=OLAF Input File Name
```

TABLE INPUT

Input:	Description
OLAF Input File Name:	The name of the OLAF input file. This file contains settings for the free vortex wake model, and is only used when *Rotor Wake/Induction Effects* are set to OLAF.

1.8.1.9 *OUTPUTS

PURPOSE

To specify the AeroDyn output settings.

THEORY

Refer to AeroDyn Overview for further information.

KEYWORD FORMAT

A block of 3 lines of data, defining the required output nodes as follows:

```
SUMPRINT=Summary Print
NBLOUTS=Number of Blade Output Nodes, BLOUTND=Blade Out Node 1, ..., Blade Ou
NTWOUTS=Number of Tower Output Nodes, TWOUTND=Tower Out Node 1, ..., Tower Ou
```

This is then followed by a block of 2 lines of data requesting the relevant outputs, which may be repeated as often as required:

```
COMMENT=Comment Line
OUTLIST=Output List
```

Set `Summary Print` to *1* to generate a summary file with name *OutFileRoot.AD.sum. OutFileRoot* is specified in the I/O SETTINGS section of the driver input file.

`Number of Blade Output Nodes` specifies the number of blade nodes for which output is requested. Valid inputs range between 0 and 9. `Blade Out Node` is a list (which is `Number of Blade Output Nodes` long) of node numbers between 1 and `Number Blade Nodes` (corresponding to a row number in the blade analysis node table in the blade data input files), separated by any combination of commas, semicolons, spaces, and/or tabs. All blades have the same output node numbers.

`Number of Tower Output Nodes` specifies the number of tower nodes for which output is requested. Valid inputs range between 0 and 9. `Tower Out Node` is a list (which is `Number of Tower Output Nodes` long) of node numbers between 1 and `Number Tower Nodes` (corresponding to a row number in the tower analysis node table), separated by any combination of commas, semicolons, spaces, and/or tabs.

The outputs specified in the `Output List` section determine which quantities are actually output at these nodes. Enter one or more lines containing comma separated output parameter names. The parameters are written in the order they are listed in the input file. You can use multiple lines so that you can break your list into meaningful groups and so the lines can be shorter. You may enter comments to be appended to the `Output List` line using the `COMMENT=Comment Line` specifier. Blade and tower node-related quantities are generated for the requested nodes identified through the `Blade Out Node` and `Tower Out Node` lists above. If you wish to post-process AeroDyn output parameters using the Flexcom *TIMETRACE, TYPE=AERODYN definition then the output parameters must be requested from the AeroDyn module so they are present in the AeroDyn output file prior to Flexcom post-processing request.

TABLE INPUT

Output Information

Input:	Description
Summary Print:	This option allows you to instruct AeroDyn to generate a summary file with name OutFileRoot.AD.sum.
Number of Blade Output Nodes:	The number of blade nodes for which output is requested. Valid inputs range between 0 and 9.

List of Blade Output Node Numbers:	A list (which is *Number of Blade Output Nodes* long) of node numbers between 1 and *Number of Blade Nodes* (corresponding to a row number in the blade analysis node table in the blade data input files, which are created by `*BLADE GEOMETRY`), separated by any combination of commas, semicolons, spaces, and/or tabs. All blades have the same output node numbers.
Number of Tower Output Nodes:	The number of tower nodes for which output is requested. Valid inputs range between 0 and 9.
List of Tower Output Node Numbers:	A list (which is *Number of Tower Output Nodes* long) of node numbers between 1 and *Number of Tower Nodes* (corresponding to a row number in the tower analysis node table), separated by any combination of commas, semicolons, spaces, and/or tabs.

Output Channels

Input:	Description

List of Output Parameters:	A list of output quantities to be generated by AeroDyn. Enter one or more rows containing quoted strings that in turn contain one or more output parameter names. Separate output parameter names by commas. If you prefix a parameter name with a minus sign, "-", underscore, "_", or the characters "m" or "M", AeroDyn will multiply the value for that channel by -1 before writing the data. The parameters are written in the order in which they are listed here. Use multiple rows so that you can break your list into meaningful groups and so the rows can be shorter. Blade and tower node-related quantities are generated for the requested nodes identified through the *List of Blade Output Node Numbers* and the *List of Tower Output Node Numbers*. If the software encounters an unknown/invalid channel name, it will issue a warning and remove the suspect channel from the output file. Please refer to List of Output Channels for a complete list of possible output parameters.
Comments:	You may enter comments relating to each output or group of similar output. These comments are included in the AeroDyn input file for information purposes only.

LIST OF OUTPUT CHANNELS

A list of all possible output parameters for the AeroDyn module is presented below. The names are grouped by meaning, but can be requested in any order you see fit. BαNβ, refers to output node β of blade α, where α is a number in the range [1,3] and β is a number in the range [1,9], corresponding to entry β in the *List of Blade Output Node Numbers*. TwNβ refers to output node β of the tower and is in the range [1,9], corresponding to entry β in the *List of Tower Output Node Numbers*.

Refer to the schematics in the Theory section for a graphical illustration of the Tower Geometry, Blade Geometry - Side View, Blade Geometry - Front View and the Blade Local Coordinate System.

Channel Name(s)	Units	Description
Tower		

TwNβVUndx, TwNβVUndy, TwNβVUndz	(m/s), (m/s), (m/s)	Undisturbed wind velocity at TwNβ in the local tower coordinate system
TwNβSTVx, TwNβSTVy, TwNβSTVz	(m/s), (m/s), (m/s)	Structural translational velocity at TwNβ in the local tower coordinate system
TwNβVrel	(m/s)	Relative wind speed at TwNβ
TwNβDynP	(Pa)	Dynamic pressure at TwNβ
TwNβRe	(-)	Reynolds number (in millions) at TwNβ
TwNβM	(-)	Mach number at TwNβ
TwNβFdx, TwNβFdy	(N/m), (N/m)	Drag force per unit length at TwNβ in the local tower coordinate system
Blade		
BαAzimuth	(deg)	Azimuth angle of Bα
BαPitch	(deg)	Pitch angle of Bα
BαNβClrnc*	(m)	Tower clearance at BαNβ*
BαNβVUndx, BαNβVUndy, BαNβVUndz	(m/s), (m/s), (m/s)	Undisturbed wind velocity at BαNβ in the local blade coordinate system
BαNβVDisx, BαNβVDisy, BαNβVDisz	(m/s), (m/s), (m/s)	Disturbed wind velocity at BαNβ in the local blade coordinate system
BαNβSTVx, BαNβSTVy, BαNβSTVz	(m/s), (m/s), (m/s)	Structural translational velocity at BαNβ in the local blade coordinate system
BαNβVrel	(m/s)	Relative wind speed at BαNβ
BαNβDynP	(Pa)	Dynamic pressure at BαNβ
BαNβRe	(-)	Reynolds number (in millions) at BαNβ
BαNβM	(-)	Mach number at BαNβ
BαNβVIndx, BαNβVIndy	(m/s), (m/s)	Axial and tangential induced wind velocity at BαNβ
BαNβAxInd, BαNβTnInd	(-), (-)	Axial and tangential induction factors at BαNβ
BαNβAlpha, BαNβTheta, BαNβPhi, BαNβCurve	(deg), (deg), (deg), (deg)	AoA, pitch+twist angle, inflow angle, and curvature angle at BαNβ
BαNβCl, BαNβCd, BαNβCm,	(-), (-), (-), (-), (-), (-), (-)	Lift force, drag force, pitching moment, normal force (to plane), tangential

BαNβCx, BαNβCy†, BαNβCn, BαNβCt		force (to plane)†, normal force (to chord), and tangential force (to chord) coefficients at BαNβ
BαNβFl, BαNβFd, BαNβMm, BαNβFx, BαNβFy†, BαNβFn, BαNβFt	(N/m), (N/m), (N·m/m), (N/m), (N/m), (N/m), (N/m)	Lift force, drag force, pitching moment, normal force (to plane), tangential force (to plane)†, normal force (to chord), and tangential force (to chord) per unit length at BαNβ
Rotor		
RtSpeed	(rpm)	Rotor speed
RtTSR	(-)	Rotor tip-speed ratio
RtVAvgxh, RtVAvgyh, RtVAvgzh	(m/s), (m/s), (m/s)	Rotor-disk-averaged relative wind velocity in the hub coordinate system (not including induction)
RtSkew	(deg)	Rotor inflow-skew angle
RtAeroFxh, RtAeroFyh, RtAeroFzh, RtAeroMxh, RtAeroMyh, RtAeroMzh	(N), (N), (N) (N·m), (N·m), (N·m)	Total rotor aerodynamic load in the hub coordinate system
RtAeroPwr	(W)	Rotor aerodynamic power
RtArea	(m2)	Rotor swept area
RtAeroCp, RtAeroCq, RtAeroCt	(-), (-), (-)	Rotor aerodynamic power, torque, and thrust coefficients

List of all possible output parameters from AeroDyn

1.8.1.10 *TOWER INFLUENCE

PURPOSE

To specify the tower influence and drag properties, and the tower mesh settings for the aerodynamic model.

THEORY

Refer to AeroDyn Overview for further information.

KEYWORD FORMAT

Multiple lines of data, defining the tower properties from the base to the top as follows:

```
Tower Elevation, Tower Diameter, Tower Drag Force Coefficient, Tower Turbuler
Tower Elevation, Tower Diameter, Tower Drag Force Coefficient, Tower Turbuler
      ::        ,      ::       ,      ::                      ,      ::
Tower Elevation, Tower Diameter, Tower Drag Force Coefficient, Tower Turbuler
```

`Tower Elevation` is measured from the mean water line to the tower node. The *Tower Elevation* must be entered in monotonically increasing order - from the lowest (tower-base) to the highest (tower-top) elevation. Refer to the Tower Geometry schematic for an illustration.

`Tower Turbulence Intensity` is only used with the Eames tower shadow model (`Tower Shadow Influence`=2). It is specified as a fraction (rather than a percentage) of the wind fluctuation.

TABLE INPUT

Input:	Description
Tower Elevation:	The local elevation of the tower node above the mean water line. The *Tower Elevation* must be entered in monotonically increasing order - from the lowest (tower-base) to the highest (tower-top) elevation. Refer to the Tower Geometry schematic for an illustration.
Tower Diameter:	The local tower diameter at the current elevation.
Tower Drag Force Coefficient:	The local tower drag force coefficient at the current elevation.
Tower Turbulence Intensity:	The turbulence intensity used in the Eames tower shadow model (Tower Shadow Influence set to *Eames model*) as a fraction (rather than a percentage) of the wind fluctuation.

NOTES

(a) As the tower is normally tapered from a wide base to a more slender top, it is normally constructed in Flexcom using several Line Sections of different diameter. The mesh density for the structural model is governed by the Line Mesh Generation settings for the line and its sub-sections, while the mesh density for the aerodynamic model is controlled via this `*TOWER INFLUENCE` keyword. It is not necessary to use the same mesh density for both models, but as a minimum, structural nodes should be placed at elevations which correspond exactly to equivalent nodes in the aerodynamic model. In practice, the structural tower is typically modelled using a certain number of sub-sections of equal length, and the intersection points between these sections serve as the aerodynamic nodes also.

(b) The number of tower nodes in the aerodynamic model, which is governed by the number of tower elevations specified above, must be greater than one. The higher the number, the finer the resolution and longer the computational time. NREL recommend that between 10 and 20 tower nodes be used, to provide a balance between accuracy and computational expense.

1.8.2 $CLEAR

This section corresponds to Clear, an ancillary module of Flexcom, which performs clearance/interference postprocessing calculations. Refer to Clearance & Interference Postprocessing for further details.

This section contains the following keywords:

- `*ANALYSIS TYPE` is used to specify the Clear analysis type and related parameters.
- `*DATABASE` is used to specify the names of the Flexcom database files on which the Clear analysis is to be based.
- `*ELEMENT SETS` is used to specify data relating to the element sets on both structures between which Clear is to perform clearance calculations.
- `*NAME` is used to specify a title for a Clear analysis run.

1.8.2.1 *ANALYSIS TYPE

Purpose

To specify the Clear analysis type and related parameters.

Theory

Refer to Analysis Type for further information on this feature.

Keyword Format

Two lines of data, defining the analysis type, and data specific to that type of analysis.

Line defining the analysis type:

```
TYPE=Analysis Type
Analysis Type can either be SNAPSHOT, TIMETRACE, or STATISTICS.
```

The format of the next line depends on the type specified, as follows.

Line for specifying SNAPSHOT data:

```
Time, No. of Points
```

Line for specifying TIMETRACE data:

```
Start Time, Minimum Distance, No. of Points
```

Line for specifying STATISTICS data:

```
Start Time, No. of Points
```

SNAPSHOT DATA

Purpose

To specify data relating to a Clear snapshot analysis

Table Input

Input:	Description
Time:	The time at which the clearance snapshot is required. This defaults to the first database output time.
No. Of Points:	The number of points or locations on each element to be used in calculating clearances. This defaults to a value of 2. See Note (a).

Notes

(a) The *No. of Points* entry must be an integer greater than or equal to 2. The default value of 2 means interference calculations are performed for the ends of the elements only. If more than 2 points are specified, interior points are distributed equally over each element.

TIMETRACE DATA

Purpose

To specify data relating to a Clear timetrace analysis

TABLE INPUT

Input:	Description
Start Time:	The time at which to begin calculating the time history of clearance. See Notes (a) and (b).
Minimum Distance	A separation value below which Clear is to consider that interference has taken place between the two structures. This defaults to zero, in which case two structures interfere only when they are actually predicted by Clear as being in contact. See Note (c).
No. Of Points:	The number of points or locations on each element to be used in calculating clearances. This defaults to a value of 2. See Note (d).

NOTES

(a) Use the *Start Time* entry to exclude initial transients from the clearance timetrace.

(b) The default *Start Time* used by Clear depends on whether the program is reading from one or two database files. If one database is being read, the default Start Time is the first database output time. If two databases are being read, Start Time defaults to the larger of the two database output start times. Of course if the two files have the same first database output time, this becomes the default.

(c) For a discussion of the significance of the Minimum Distance entry, refer to Minimum Distance.

(d) The *No. of Points* entry must be an integer greater than or equal to 2. The default value of 2 means interference calculations are performed for the ends of the elements only. If more than 2 points are specified, interior points are distributed equally over each element.

STATISTICS DATA

PURPOSE

To specify data relating to a Clear statistics analysis

TABLE INPUT

Input:	Description
Start Time:	The time at which to begin calculating the time history of clearance. See Notes (a) and (b).
No. Of Points:	The number of points or locations on each element to be used in calculating clearances. This defaults to a value of 2. See Note (c).

Notes

(a) Use the *Start Time* entry to exclude initial transients from the clearance timetrace.

(b) The default *Start Time* used by Clear depends on whether the program is reading from one or two database files. If one database is being read, the default Start Time is the first database output time. If two databases are being read, *Start Time* defaults to the larger of the two database output start times. Of course if the two files have the same first database output time, this becomes the default.

(c) The *No. of Points* entry must be an integer greater than or equal to 2. The default value of 2 means interference calculations are performed for the ends of the elements only. If more than 2 points are specified, interior points are distributed equally over each element.

1.8.2.2 *DATABASE

PURPOSE

To specify the names of the Flexcom database files on which the Clear analysis is to be based.

THEORY

Refer to Database Name for further information on this feature.

KEYWORD FORMAT

Single line defining the database names:

```
[DATABASE_1=Database Name 1] [, DATABASE_2=Database Name 2]
```

If neither database name is provided, both default to the name of the Clear analysis. If the first database name is provided, but the second one is omitted, the latter defaults to the former.

TABLE INPUT (ELEMENTS – DATA)

Input:	Description
Start Element:	The first element of the element set.
End Element:	The last element of the element set. See Note (a).
External Diameter:	The effective external diameter of the elements comprising the element set, for the purpose only of calculating clearances in Clear. This defaults to the value for the effective external diameter of each element stored in the Flexcom database file. See Note (b).
Database Name:	The name of the Flexcom database output file containing the positions and/or motions of the elements of the set. The file extension (dbm) should be omitted. See Note (c).

NOTES

(a) Element numbers are assumed to be continuous from the specified start to the end numbers.

(b) For a discussion of the significance of the *External Diameter* entry, refer to [External Diameter].

(c) The Flexcom results to be accessed by Clear can be in one or two database files, depending on whether the structures under consideration were analysed separately or together. Refer to [Database Name] for further details.

1.8.2.3 *ELEMENT SETS

PURPOSE

To specify data relating to the element sets on both structures between which Clear is to perform clearance calculations.

THEORY

Refer to [Clearance & Interference Postprocessing] for further information on this feature.

KEYWORD FORMAT

Two lines of data, defining composition of element sets for clearance calculations:

```
Set 1 First Element (Number or Label), Set 1 Last Element (Number or Label)
Set 2 First Element (Number or Label), Set 2 Last Element (Number or Label)
```

If you specify an element label rather than an element number, it must be enclosed in {} brackets. The user specified diameters are optional. If they are not specified (or are zero), the external diameters in the databases are used.

TABLE INPUT

Input:	Description
Start Element:	The first element (number or label) of the element set. If you specify an element label rather than an element number, it must be enclosed in {} brackets

End Element:	The last element (number or label) of the element set. See Note (a). If you specify an element label rather than an element number, it must be enclosed in {} brackets
External Diameter:	The effective external diameter of the elements comprising the element set, for the purpose only of calculating clearances in Clear. This defaults to the value for the effective external diameter of each element stored in the Flexcom database file. See Note (b).
Database Name:	The name of the Flexcom database output file containing the positions and/or motions of the elements of the set. The file extension (dbm) should be omitted. See Note (c).

NOTES

(a) Elements (numbers or labels) are assumed to be continuous from the specified start to the end.

(b) For a discussion of the significance of the *External Diameter* entry, refer to External Diameter.

(c) The Flexcom results to be accessed by Clear can be in one or two database files, depending on whether the structures under consideration were analysed separately or together. Refer to Database Name for further details.

1.8.2.4 *NAME

PURPOSE

To specify a title for a Clear analysis run.

THEORY

Refer to Clearance & Interference Postprocessing for further information on this feature.

KEYWORD FORMAT

A single line containing the analysis name.

```
Name
```

TABLE INPUT

Input:	Description
Title:	A descriptive title to be associated with the Clear analysis. This entry is optional.

1.8.3 $CODE CHECKING

This section corresponds to the code checking facility, which allows you to check analysis results against specific design codes. Refer to Code Checking for further details.

This section contains the following keywords:

- `*CODE` is used to specify the relevant code/procedure to be used in code checking.
- `*ELEMENT SETS` is used to group individual elements into element sets.
- `*ENVIRONMENTAL` is used to specify a list of Flexcom analyses to be included in the code checking calculations.
- `*GENERAL` is used to specify a general usage factor and an incidental to design pressure ratio to be used in code checking.
- `*LOAD FACTOR` is used to specify load effect factors to be used in code checking.
- `*MATERIAL` is used to specify materials properties to be used in DNV and API code checking.
- `*MATERIAL-ISO` is used to specify materials properties to be used in ISO code checking.
- `*REFERENCE PRESSURE` is used to specify the local internal pressure at a reference point for ISO code checking.
- `*RESISTANCE FACTOR` is used to specify resistance factors to be used in code checking.

- *SECTION PROPERTIES is used to specify section properties to be used in code checking.
- *TIMETRACE is used to request the creation of timetrace plots.

1.8.3.1 *CODE

PURPOSE

To specify the relevant code/procedure to be used in code checking.

THEORY

Refer to Code Checking for further information on this feature.

KEYWORD FORMAT

One line specifying the relevant code:

```
CODE=Code Name
Code Name can be DNV-OS-F101 (Submarine Pipeline Systems), DNV-OS-F201
(Dynamic Risers), API-STD-2RD (Dynamic Risers for Floating Production Systems
or ISO-13628-7(Completion/workover riser systems).
```

TABLE INPUT

Input:	Description
Code Name:	The name of the relevant code/procedure to be used in code checking.

NOTES

(a) Refer to Code Checking for a detailed discussion of the code checking facility.

1.8.3.2 *ELEMENT SETS

PURPOSE

To group individual elements into element sets.

THEORY

Refer to Code Checking for further information on this feature.

KEYWORD FORMAT

Blocks of lines that define a set repeated as often as necessary. Each block begins with a line defining the set name. This is followed by various types of lines that define the elements in the set. These lines can be mixed and repeated as often as necessary until every element in the set is defined.

Line to define set name:

```
SET=Set Name
```

Line containing a list of elements. This line can contain up to 20 elements (numbers or labels). Any further elements must be defined on a new line.

```
List of Elements (Numbers or Labels)
```

Line defining a sequence of elements:

```
GEN=Start Element (Number or Label), End Element (Number or Label) [, Element
```

Line refere another set of elements:

```
SUBSET=Subset Set Name
```

The set ALL is predefined and cannot be redefined. Every element assigned to a set must be defined using `*ELEMENT`. Set names are up to 256 characters long, can include spaces and are case insensitive. If you specify an element label rather than an element number, it must be enclosed in {} brackets. *`Element Increment`* defaults to one.

TABLE INPUT

Input:	Description
Set Name:	A unique label for the element set. Set names are not case sensitive, so the set name 'RISER' is equivalent to 'Riser', which is in turn equivalent to 'riser'.
Elements:	The elements comprising the set. These can be input in three ways, namely: (i) A list of elements (numbers or labels), such as for example "1, 5, 7".

	(ii) A group of consecutive elements (numbers or labels), input using the format: “11 – 15”. This definition specifies Elements 11 to 15 inclusive. The specification “11 - 15 – 2” can be used to specify the Elements 11, 13 and 15 - that is, from Element 11 to Element 15 in steps of 2. (iii) Another set name. For example you might define three sets named SET_1, SET_2 and SET_3, and then combine them in a further set, say ALLSETS, by inputting “SET_1, SET_2, SET_3”. If you specify an element label rather than an element number, it must be enclosed in {} brackets. All three specifications can be combined, as for example in “1, 7, 9, 12-15, 17, 20-50-10, RISER”. This set combines Elements 1, 7, and 9; elements 12 to 15 inclusive; Element 17; Elements 20, 30, 40 and 50; and the elements comprising the set RISER.
SubSets:	An additional element set or sets whose elements are to be added to the current set definition. If more than one set is referenced, use commas to separate out the set names.

NOTES

(a) Use as many lines as you need to completely define the elements comprising a particular set. Simply leave the first column blank for second and subsequent lines.

(b) If a set name is included in the specification of another set, then obviously the elements comprising that set must be separately defined.

(c) There is one predefined element set in Flexcom, which is named All. Not surprisingly this comprises all of the elements of the finite element discretisation and is the default element set. Note that Flexcom will resist any attempt to redefine the make-up of the set *All*.

(d) The names and composition of element sets you define in a preceding section carry through to all dependent (restart) sections. For example, if you define an element set in the `$MODEL` section, it will automatically be available in a subsequent `$DATABASE POSTPROCESSING` section. So there is no need to repeat the specifications again – you can just use the set names directly. If you redefine the composition of a set previously defined in a preceding section, Flexcom will output a warning, but will continue with the most recent set definition.

1.8.3.3 *ENVIRONMENTAL

PURPOSE

To specify a list of Flexcom analyses to be included in the code checking calculations.

THEORY

Refer to Code Checking for further information on this feature.

KEYWORD FORMAT

A block of lines specifying a list of Flexcom analyses to be included in the code checking calculations. The block begins with a line specifying the environmental (and optionally the functional) analysis file name. This is followed by an optional line specifying start and end times for the code checking calculations. This may be followed by a list of element sets which are to be considered for checking. The element set line may be repeated as often as required. The entire block of data may then be repeated for the second and successive Flexcom analysis.

Mandatory line specifying the file name:

```
ENVIRONMENTAL=Environmental File Name [,FUNCTIONAL=Functional File Name]
[,FACTOR=Design Factor] [, COLLATE= Collate Option]
```

Optional line specifying the start and end times:

```
[START=Start Time] [, END=End Time]
```

Optional (repeatable) line specifying an element set name:

```
[SET=Set Name]
```

If the `Functional File Name` is not specified, it defaults to the analysis immediately preceding the environmental analysis in the restart chain. `Start Time` and `End Time` default to the analysis start and end times respectively. `Set Name` defaults to all elements (set `ALL`).

TABLE INPUT

Input:	Description
Environmental File Name:	The name of the Flexcom analysis which represents the environmental load case. A file extension is not required.
Functional File Name:	The name of the Flexcom analysis which represents the functional load case. If not specified, it defaults to the analysis immediately preceding the environmental analysis in the restart chain.
Design Factor:	This design factor for use with the API-STD-2RD and ISO-13628-7 code check.
Collation Option:	Invoking this option means a summary of maximum utilizations of that environment will be included in a summary file (with the file extension SUM)
Start Time:	The start time for the code checking calculations. This entry is optional, and defaults to the analysis start time if omitted.
End Time:	The end time for the code checking calculations. This entry is optional, and defaults to the analysis end time if omitted.
Set Name:	The name of the element set to be considered for code checking. This defaults to all elements.

NOTES

(a) Use as many rows as you need to define a list of element sets in a particular Flexcom analysis to be considered for code checking. Simply leave Columns 1, 2 and 3 blank for the second and subsequent element sets. For subsequent Flexcom analyses, put the file name in Column 1 and specify the element set data in the same way.

1.8.3.4 *GENERAL

PURPOSE

To specify a general usage factor and an incidental to design pressure ratio to be used in code checking.

THEORY

Refer to Code Checking for further information on this feature.

KEYWORD FORMAT

A block of lines to specify the relevant factors.

```
[USAGE=Usage Factor]
[INCIDENTAL=Incidental to design pressure ratio]
```

Usage Factor defaults to 0.75.

Incidental to design pressure ratio defaults to 1.1.

TABLE INPUT

Input:	Description
Usage Factor:	The Working Stress Design usage (WSD) as per the relevant DNV standard. This value defaults to 0.75 if not specified.
Incidental to Design Pressure Ratio:	This is the incidental to design pressure ration as per the relevant DNV standard. This value defaults to 1.1 if not specified.

1.8.3.5 *LOAD FACTOR

PURPOSE

To specify load effect factors to be used in code checking.

THEORY

Refer to Code Checking for further information on this feature.

KEYWORD FORMAT

A block of three lines to specify the relevant load effect factors:

```
[FUNCTIONAL=Gamma F]
[ENVIRONMENTAL=Gamma E]
[OPTIMISE=Optimise Option]
```

Gamma F and *Gamma E* default to 1.1 and 1.3 respectively. Optimise Option can be YES (the default) or NO.

TABLE INPUT

Input:	Description
Functional (Gamma F):	The functional factor (γf) as per the relevant DNV standard. This value defaults to 1.1 if not specified.
Environmental (Gamma E):	The environmental factor (γe) as per the relevant DNV standard. This value defaults to 1.3 if not specified.
Optimise Factors:	Option to optimise load factors to maximise load effects. The default value of *Yes* means that the environmental factor is taken as 1/γe if the environmental loading reduces the combined load effects.

1.8.3.6 *MATERIAL

PURPOSE

To specify standard material properties to be used in DNV and API code checking.

THEORY

Refer to Code Checking for further information on this feature.

KEYWORD FORMAT

A block of two lines defining the material name and its associated properties. The block may be repeated as often as necessary to define all the required materials.

```
NAME=Material Name
SMYS, SMTS, Young's Modulus, Fy temp, Fu temp [, Alpha U] [, Poisson's Ratio]
```

`Alpha U` and `Poisson's Ratio` default to 0.96 and 0.3 respectively.

MATERIAL TABLE INPUT

Input:	Description
Material Name:	The name of the material.
SMYS:	The Specified Minimum Yield Strength for the material.
SMTS:	The Specified Minimum Tensile Strength for the material.
Young's Modulus:	Young's modulus for the material.
Fy Temp:	The temperature derating factor for the yield stress ($F_{y,temp}$) as per the relevant DNV standard.
Fu Temp:	The temperature derating factor for the tensile stress ($F_{u,temp}$) as per the relevant DNV standard.
Alpha U:	The material strength factor (α_u) as per the relevant DNV standard. This value defaults to 0.96 if not specified.
Poisson's Ratio:	Poisson's ratio for the material. This value defaults to 0.3 if not specified.

1.8.3.7 *MATERIAL-ISO

PURPOSE

To specify material properties to be used in ISO code checking.

THEORY

Refer to Code Checking for further information on this feature.

KEYWORD FORMAT

A block of two lines defining the material name and its associated properties. The block may be repeated as often as necessary to define all the required materials.

```
NAME=Material Name
Rt05, Rm, Temperature, [A5], Young's Modulus, Tc, Mc, Pec[, Poisson's Ratio]
```

A5 and *Poisson's Ratio* default to 14% and 0.3 respectively.

ISO MATERIAL TABLE INPUT

Input:	Description
Material Name:	The name of the material.
Rt05:	Specified minimum yield strength for 0.5% total elongation at room temperature.
Rm:	Specified minimum ultimate tensile strength at room temperature.
Temperature:	Elevated temperature.
A5:	Percentage minimum elongation after fracture. This value defaults to 14% if not specified.
Young's Modulus:	Young's modulus for the material.
Tc	Single load ultimate tension capacity.
Mc	Single load ultimate bending capacity.
Pec	Single load ultimate pressure capacity.
Poisson's Ratio:	Poisson's ratio for the material. This value defaults to 0.3 if not specified.

1.8.3.8 *REFERENCE PRESSURE

PURPOSE

To specify the local internal pressure at a reference point for ISO code checking.

THEORY

Refer to Code Checking for further information on this feature.

KEYWORD FORMAT

A block of two lines to specify the reference point element number and the local internal pressure at the midpoint of this element:

```
ELEMENT=Element
PRESSURE=Pressure
```

TABLE INPUT

Input:	Description
Element:	The element at which the local internal pressure is to be defined. The reference point is the midpoint of this element.
Pressure:	The local internal pressure from which the internal pressure at the reference point is calculated.

1.8.3.9 *RESISTANCE FACTOR

PURPOSE

To specify resistance factors to be used in code checking.

THEORY

Refer to Code Checking for further information on this feature.

KEYWORD FORMAT

A block of two lines to specify the relevant resistance factors:

```
[MATERIAL=Gamma m]
[SAFETY=Gamma SC]
```

Gamma m and *Gamma SC* default to 1.15 and 1.26 respectively.

TABLE INPUT

Input:	Description
Material (Gamma m):	The material resistance factor (γ_m) as per the relevant DNV standard. This value defaults to 1.15 if not specified.
Safety (Gamma SC):	The safety class resistance factor (γ_{sc}) as per the relevant DNV standard. This value defaults to 1.26 if not specified.

1.8.3.10 *SECTION PROPERTIES

PURPOSE

To specify section properties to be used in code checking.

THEORY

Refer to Code Checking for further information on this feature.

KEYWORD FORMAT

A block of two lines defining the section and its associated properties. The block may be repeated as often as necessary to define all the required section properties.

```
SET=Set Name, MATERIAL=Material Name
[Alpha C], [Diameter], [Thickness], [T corr], [Alpha fab], [f0], [Moment fact
```

Alpha C is calculated as per the relevant DNV standard if unspecified. *Diameter* and *Thickness* are obtained from the Flexcom analysis if unspecified. *T corr*, *Alpha fab* and *Ovality* default to 0.0, 0.85 and 0.005, respectively. *Moment Factor* and *Tension Factor* default to 1.0.

TABLE INPUT

Input:	Description
Set Name:	The name of the element set.

Material Name:	The name of the associated material.
Alpha C:	Flow stress parameter accounting for strain hardening (α_c) as per the relevant DNV standard.
Diameter:	The section diameter. This value defaults to the corresponding element outer diameter used in the Flexcom analysis.
Thickness:	The section thickness. This value defaults to the corresponding element wall thickness used in the Flexcom analysis.
T corr:	Corrosion allowance (T_{corr}) as per the relevant DNV standard. This value defaults to 0.0 if not specified.
Alpha Fab:	Manufacturing process reduction factor (α_{fab}) as per the relevant DNV standard. This value defaults to 0.85 if not specified.
Ovality:	The section ovality (f_o) as per the relevant DNV standard. This value defaults to 0.005 if not specified.
Moment Factor:	Moment scale factor. This value defaults to 1.0 if not specified.
Tension Factor:	Tension scale factor. This value defaults to 1.0 if not specified.

1.8.3.11 *TIMETRACE

PURPOSE

To request the creation of timetrace plots.

THEORY

Refer to Code Checking for further information on this feature.

KEYWORD FORMAT

A block of two or more lines defining the location (i.e. element, local node) at which the outputs are required, and the associated element set definition. The element line may be repeated to request further plots within the same element set. The block may be repeated to request outputs for different element sets.

Block requesting force timetraces. Lines 2-4 can be repeated and mixed as often as necessary.

```
SET=Set Name
Element (Number or Label), Local Node, OUTPUT=Output Type
```

Output Type can be `ALL, LFRD, WSD, VMS, METHOD1, METHOD2, METHOD3, METHOD4A,` or `METHOD4B`. If you specify an element label rather than an element number, it must be enclosed in {} brackets.

TABLE INPUT

Input:	Description
Set Name:	The name of the element set.
Element:	The element (number or label) for which the timetrace is required. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Local Node:	This option allows you to choose between three locations on the specified element. First Node refers to the first specified node of the element, Last Node is the second specified node, and Midpoint is half-way between the two.
Output:	This list allows you to select a relevant output parameter. The entries are largely self-explanatory. See Note (a).

NOTES

(a) Refer to Output for a discussion of the various output parameters. The abbreviations used as are follows:

- LRFD Load and Resistance Factor Design

- WSD Working Stress Design
- VMS Von Mises Stress
- METHOD1 API-STD-2RD: Utilization of combined membrane load
- METHOD2 API-STD-2RD: Utilization of axial load based on yield tension
- METHOD3 DNV-OS-F201: LRFD Utilization
- METHOD4A API-STD-2RD: Utilization of combined pressure and axial load
- METHOD4B API-STD-2RD: Utilization of bending strain

1.8.4 $DATABASE POSTPROCESSING

This section corresponds to the database postprocessing facility, which generally represents the most comprehensive postprocessing resource. Refer to Database Postprocessing for further details.

This section contains the following keywords:

- `*AXIS/VECTOR` is used to define axis systems and vectors for use in postprocessing.
- `*ELEMENT SETS` is used to group individual elements into element sets.
- `*EXTREMA` is used to specify the procedure and input parameters to be used in the computation of extreme values.
- `*OUTPUT FILES` is used to specify the types of output file required from database postprocessing.
- `*PROPERTIES` is used to assign effective structural properties to element sets for use in calculating stresses.
- `*RAO` is used to request the creation of RAO plots.
- `*RESTART` is used to indicate that a postprocessing run is to be restarted from an analysis file of different stub name.
- `*SNAPSHOT` is used to request the creation of snapshot plots.
- `*SPECTRUM` is used to request the creation of spectrum plots.

- *STANDARD OUTPUT is used to quickly request a selection of commonly used outputs.
- *STATISTICS is used to request the creation of statistics plots.
- *TIMETRACE is used to request the creation of timetrace plots.

1.8.4.1 *AXIS/VECTOR

PURPOSE

To define axis systems and vectors for use in postprocessing.

THEORY

Refer to Angles Output for further information on this feature.

KEYWORD FORMAT

The keyword starts with a line defining the name of the axis system or vector. This is then followed by a line with either seven numbers or four numbers, depending on whether an axis system or vector is being defined. Any further axis systems or vectors can be defined by repeating the two lines as often as necessary.

Lines defining an axis system:

```
AXIS=Axis Name
Origin Node (Number or Label), X1, Y1, Z1, X2, Y2, Z2
```

Lines defining a vector:

```
VECTOR=Vector Name
Origin Node (Number or Label), X, Y, Z
```

If you specify a node label rather than a node number, it must be enclosed in {} brackets. The two vectors defining an axis system must have non-zero length and must be orthogonal. Any vectors defined must have non-zero length.

AXIS/VECTOR - AXIS SYSTEMS

TABLE INPUT

Input:	Description

Name:	A unique name for the axis system being defined.
Origin Node:	The node (number or label) at which the axis system is being positioned. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
X1:	The component in the global X-direction of the local x-axis of the axis system being defined.
Y1:	The component in the global Y-direction of the local x-axis of the axis system being defined.
Z1:	The component in the global Z-direction of the local x-axis of the axis system being defined.
X2:	The component in the global X-direction of the local y-axis of the axis system being defined.
Y2:	The component in the global Y-direction of the local y-axis of the axis system being defined.
Z2:	The component in the global Z-direction of the local y-axis of the axis system being defined.

NOTES

(a) If you do not specifically input a value in any of Columns 3-8, then the corresponding variable defaults to 0. Note however that all three of *X1, Y1* and *Z1* or all three of *X2, Y2* and *Z2* cannot equal 0 – the null vector cannot be either the local x-axis or the local y-axis.

(b) Note that the local x-axis and local y-axis that you define here must be *orthogonal*, that is, the true angle between the two vectors defining the local x- and y-axes must be 90°. Flexcom will generate an error message if this is not the case, as vectors that are not orthogonal cannot be used to form a valid axis system.

AXIS/VECTOR - VECTOR

TABLE INPUT

Input:	Description
Name:	A unique name for the vector being defined.
Origin Node:	The node (number or label) at which the vector is being positioned. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
X:	The component in the global X-direction of the vector being defined.
Y:	The component in the global Y-direction of the vector being defined.
Z:	The component in the global Z-direction of the vector being defined.

NOTES

(a) If you do not specifically input a value in any of Columns 3, 4 or 5, then the corresponding variable defaults to 0. Note however that all three of *X, Y* and *Z* cannot equal 0 – the angle between an element and the null vector is not a valid output.

1.8.4.2 *ELEMENT SETS

PURPOSE

To group individual elements into element sets.

THEORY

Refer to Database Postprocessing for further information on this feature.

KEYWORD FORMAT

Blocks of lines that define a set repeated as often as necessary. Each block begins with a line defining the set name. This is followed by various types of lines that define the elements in the set. These lines can be mixed and repeated as often as necessary until every element in the set is defined.

Line to define set name:

```
SET=Set Name
```

Line containing a list of elements. This line can contain up to 20 elements (numbers or labels). Any further elements must be defined on a new line.

```
List of Elements (Numbers or Labels)
```

Line defining a sequence of elements:

```
GEN=Start Element (Number or Label), End Element (Number or Label) [, Element
```

Line referencing another set of elements:

```
SUBSET=Subset Set Name
```

The set `ALL` is predefined and cannot be redefined. Every element assigned to a set must be defined using `*ELEMENT`. Set names are up to 256 characters long, can include spaces and are case insensitive. If you specify an element label rather than an element number, it must be enclosed in {} brackets. *Element Increment* defaults to one.

TABLE INPUT

Input:	Description
Set Name:	A unique label for the element set. Set names are not case sensitive, so the set name ‘RISER’ is equivalent to ‘Riser’, which is in turn equivalent to ‘riser’.
Elements:	The elements comprising the set. These can be input in three ways, namely: (i) A list of elements (numbers or labels), such as for example “1, 5, 7”.

	(ii) A group of consecutive elements (numbers or labels), input using the format: "11 – 15". This definition specifies Elements 11 to 15 inclusive. The specification "11 - 15 – 2" can be used to specify the Elements 11, 13 and 15 - that is, from Element 11 to Element 15 in steps of 2. (iii) Another set name. For example you might define three sets named SET_1, SET_2 and SET_3, and then combine them in a further set, say ALLSETS, by inputting "SET_1, SET_2, SET_3". If you specify an element label rather than an element number, it must be enclosed in {} brackets. All three specifications can be combined, as for example in "1, 7, 9, 12-15, 17, 20-50-10, RISER". This set combines Elements 1, 7, and 9; elements 12 to 15 inclusive; Element 17; Elements 20, 30, 40 and 50; and the elements comprising the set RISER.
SubSets:	An additional element set or sets whose elements are to be added to the current set definition. If more than one set is referenced, use commas to separate out the set names.

NOTES

(a) Use as many lines as you need to completely define the elements comprising a particular set. Simply leave the first column blank for second and subsequent lines.

(b) If a set name is included in the specification of another set, then obviously the elements comprising that set must be separately defined.

(c) There is one predefined element set in Flexcom, which is named *All*. Not surprisingly this comprises all of the elements of the finite element discretisation and is the default element set. Note that Flexcom will resist any attempt to redefine the make-up of the set *All*.

(d) The names and composition of element sets you define in a preceding section carry through to all dependent (restart) sections. For example, if you define an element set in the `$MODEL` section, it will automatically be available in a subsequent `$DATABASE POSTPROCESSING` section. So there is no need to repeat the specifications again – you can just use the set names directly. If you redefine the composition of a set previously defined in a preceding section, Flexcom will output a warning, but will continue with the most recent set definition.

1.8.4.3 *EXTREMA

PURPOSE

To specify the procedure and input parameters to be used in the computation of extreme values.

THEORY

Refer to Extreme Values for further information on this feature.

KEYWORD FORMAT

A block of two lines of data. The first line defines the procedure to be used in the computation of extreme values. The second line defines the necessary input parameters.

Line defining the extreme value calculation procedure:

```
TYPE=Calculation Procedure
```

The next line defines the parameters required for extreme value postprocessing. The format depends on the *Calculation Procedure* specified – the options are RAYLEIGH (the default) and WEIBULL. For RAYLEIGH extrema postprocessing, the line takes the following format:

```
DATA=[Storm Duration] [, Probability] [, Calculation Start Time]
```

For WEIBULL extrema postprocessing, it takes the following format:

```
DATA=[Storm Duration] [, Probability] [, Threshold] [, Calculation Start Time
```

Calculation Procedure can be RAYLEIGH or WEIBULL. *Storm Duration* defaults to 3 hours and Probability to 0.01 (1%). *Threshold* defaults to 1. *Calculation Start Time* defaults to the analysis start time.

TABLE INPUT

Extreme Values - Rayleigh

Input:	Description
Storm Duration:	The storm duration in hours to be used when calculating extreme values. The default is 3 hours.
Probability:	The exceedance probability to be used when calculating extreme values. The default is 0.01 (= 1%).
Start Time:	The time in the analysis to start the extreme value calculation. Use this entry to exclude initial transients from the calculation. The default is the database output start time. See Note (b).

Extreme Values - Weibull

Input:	Description
Storm Duration:	The storm duration in hours to be used when calculating extreme values. The default is 3 hours.
Probability:	The exceedance probability to be used when calculating extreme values. The default is 0.01 (= 1%).
Threshold:	The proportion of largest maximums /smallest minimums to be used when calculating extreme values. The default is 1. See Note (c).
Start Time:	The time in the analysis to start the extreme value calculation. Use this entry to exclude initial transients from the calculation. The default is the database output start time. See Note (b).

NOTES

(a) Refer to Extreme Values for a detailed discussion of all aspects of specifying data relating to extreme value postprocessing.

(b) The duration of the timetrace (end time minus start time) can be less than the storm duration.

(c) A threshold equal to 1 denotes all the maximums/minimums in the timetrace will be used to calculate the extreme values. A threshold equal to 1/*n* (e.g. 1/3) uses the upper 1/n of largest maximums and lowest 1/*n* of smallest minimums to calculate the extreme value.

(d) Note that the *Extreme Values – Rayleigh* and *Extreme Values – Weibull* tables are mutually exclusive. Specification of data in either table indicates the extreme value postprocessing approach which you wish to use.

1.8.4.4 *OUTPUT FILES

PURPOSE

To specify the types of output file required from database postprocessing.

THEORY

Refer to Graphical Output, Tabular Output and Spreadsheet Output for further information on this feature.

KEYWORD FORMAT

A single line listing the types of output file required.

```
[PLOT=Plot Option] [,TEXT=Text Option] [,EXCEL=Excel Option] [,TIMETRACE=Time
```

If the keyword is not present, then the first three types of output file are produced by default while timetrace and CSV output are disabled. `Plot Option` can be `YES` (the default) or `NO`. `Text Option` can be `YES` (the default) or `NO`. `Excel Option` can be `YES` (the default) or `NO`. `Timetrace Option` can be `YES` or `NO` (the default). `CSV Option` can be `YES` or `NO` (the default).

TABLE INPUT

Input:	Description
Plot:	Invoking this option means that graphical output will be produced in the form of plot files (with the file extension MPLT). These files may be examined using the *Plotting* facility.

Text:	Invoking this option means that text based output will be produced in the form of tabular output files (with the file extension TAB).
Excel:	Invoking this option means that spreadsheet based output will be produced in the form of Excel files (with the file extension XLSX).
Timetrace:	Invoking this option means that full timetrace output will be present in TAB and Excel files, rather than just statistics.
CSV	Invoking this option means that a CSV (comma separated values) file is created to accompany each plot file. The CSV file contains similar information to the plot file but the advantage is that it can be read automatically by Excel, facilitating further processing of the data. The Plot option must also be invoked for CSV output to be produced.

1.8.4.5 *PROPERTIES

PURPOSE

To assign effective structural properties to element sets for use in calculating stresses.

THEORY

Refer to Stress Properties for further information on this feature.

KEYWORD FORMAT

A block of two lines repeated as often as necessary. Each property specified on the second line can be omitted, in which case a default value will be used for stress calculations:

```
SET=Set Name
Do, Di, A, Iyy, Izz, Tmin
```

The properties specified here are used in calculating stresses during database postprocessing. This keyword is optional and may be ignored if you wish to use the actual properties (e.g. via the `*GEOMETRIC SETS` keyword in `$MODEL`), or properties explicitly specified for stress computations (i.e. via the corresponding `*PROPERTIES` keyword in `$MODEL`), as the values specified in the model definition are carried through to the database postprocessor.

If any of the parameters `A, Iyy, Izz` or `Tmin` are omitted, default values are computed based on the specified (or default) values of `Do` and `Di`. The minimum wall thickness (`Tmin`) is only used in the alternative computation of von Mises stress, according to API-2RD.

TABLE INPUT

Input:	Description
Set Name:	The name of the element set to which the stress properties are to be assigned. This defaults to all elements.
Do:	The effective outside diameter for the elements of the set. The default depends on the format used to specify geometric data. If you used the *Flexible Riser* format, then the default is the drag diameter for the elements of the set. If you used the *Rigid Riser* or *Mooring Line* formats, then Do here defaults to Do input in the properties data.
Di:	The effective internal diameter for the elements of the set. The default is the internal diameter specified in the geometric data.
A:	The effective cross-sectional area for the elements of the set. The default is given by: $$A = \frac{\pi\left(D_o^2 - D_i^2\right)}{4}$$ where Do and Di are the inputs described above, default or otherwise.
Iyy:	The second moment of area about the local y-axis for the elements of the set. The default is given by: $$I_{yy} = \frac{\pi\left(D_o^4 - D_i^4\right)}{64}$$ where Do and Di are the inputs described above, default or otherwise.

Izz:	The second moment of area about the local z axis for the elements of the set. The default is Iyy above.
Min. Wall Thicknes s:	The minimum wall thickness for the elements of the set. The default is given by: $T_{\min} = \frac{(D_o - D_i)}{2}$ where Do and Di are the inputs described above, default or otherwise.

NOTES

(a) This keyword is identical to the `*PROPERTIES` keyword of the main `$MODEL` section. If you use the keyword in `$MODEL` to specify stress properties, then these automatically carry through to database postprocessing and you do not need to repeat the specification here. Refer to Stress Properties for a detailed discussion of the various parameters above.

(b) The minimum wall thickness is only used in the computation of von Mises stress according to API-2RD. Refer to Von Mises Stress (API-2RD Method) for a discussion of the API-2RD computation procedure.

1.8.4.6 *RAO

PURPOSE

To request the creation of RAO plots.

THEORY

Refer to Computation of RAOs for further information on this feature.

KEYWORD FORMAT

The keyword begins with a line defining miscellaneous parameters for all the RAO plots. This is followed by an optional line to specify the independent variable to be used. This is then followed by blocks of lines repeated as often as necessary to request the actual RAO plots.

Line defining miscellaneous parameters:

```
[PLOT=YES/NO] [STEP=Time Step], ENSEMBLES=No. of Ensembles [, START=Start Tim
```

Optional line defining independent variable:

```
[OPTION=Independent Variable]
```

Block requesting motion RAOs. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=MOTION
Node (Number or Label), DOF [, Scale] [, PHASE=YES/NO] [, UNITS=Unit]
[TITLE=Figure Title]
```

Block requesting force RAOs. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=FORCE
Element (Number or Label), Local Node, Variable [, Scale] [, PHASE=YES/NO] [,
[TITLE=Figure Title]
```

Refer to Force Variable Input for further information on acceptable *Variable* values. Refer to Location Parameter Input for further information on the significance of the *Location* input. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis.

Block requesting reaction RAOs. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=REACTION
Node (Number or Label), DOF [, Scale] [, PHASE=YES/NO] [, UNITS=Unit]
[TITLE=Figure Title]
```

Block requesting floating body motion RAOs. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=MOTION
FLOATING BODY=Body Name, DOF [, Scale] [, PHASE=YES/NO] [, UNITS=Unit]
[TITLE=Figure Title]
```

`A Time Step` value is required in a random sea dynamic analysis with a variable time-step; it is ignored in the case of an analysis with a fixed step. `No. of Ensembles` is a required input which must be greater than 1. `Start Time` defaults to the analysis start time. If no `Cut-off Frequency` is specified, Flexcom calculates a suitable default based on the analysis time variables. `Independent Variable` can be S (seconds, the default), H (Hertz), or R (radians/second). If you specify a node/element label rather than a node/element number, it must be enclosed in {} brackets. Scale defaults to 1. The Unit entry defaults to base units. `Figure Title` defaults to blank.

RAO PLOTS - GENERAL PARAMETERS

Purpose

To define a number of parameters relating to the calculation of RAOs.

Table Input

Input:	Description
Start Time:	The start time for the calculation of RAOs. Flexcom excludes any values before this time. Use this entry to exclude initial transients from RAO calculations.
Ensembles:	The number of ensembles to be used in calculating RAOs. See Note (a).
Time Step:	The time step to be used when calculating RAOs from the results of a random sea dynamic analysis with a variable time step. See Note (b). If the analysis used a fixed time step then this entry is not required, and any value you might specify is ignored.
Cut-off Frequency:	The RAO cut off frequency in Hz. Flexcom truncates all output at this frequency. This entry is optional; by default, Flexcom calculates an appropriate value from the simulation length and the number of ensembles you specify.

Independent Variable:	This option allows you to select the independent variable (that is, the variable on the plot X axis) for RAO plots. The options are *Period (seconds)* which is the default, *Frequency (Hz)* and *Circular Frequency (radians/s).*
Plot Files:	This option allows you to request the creation of RAO plot files. The options are *Yes* (the default) and *No*.

Notes

(a) The procedure used by Flexcom in calculating spectra is as follows. Firstly, the output timetrace is divided equally into a number of smaller timetraces or ensembles. A spectrum for each ensemble is then calculated using the Fast Fourier Transform (FFT) algorithm. Finally, the actual spectrum to be output is found as an average of the spectra calculated for each ensemble. This standard procedure minimises the statistical error associated with the FFT process. You specify the number of ensembles to be used in this process using the *Ensembles* entry above. This value should always be greater than 1.

(b) The FFT algorithm requires a record with a fixed time step. When you perform a variable step analysis, obviously such a record is not available, and so Flexcom must synthesise one by interpolating from the variable step record. The *Time Step* entry tells Flexcom what time step to use in the synthesised record.

(c) The procedure followed by the software for the computation of RAOs depends on whether the relevant analysis involved regular or irregular wave loading. Refer to Computation of RAOs for further details.

RAOS - MOTION

Purpose

To request plots of RAOs of motion.

Table Input

Input:	Description

Node:	The node (number or label) for which the motion RAO is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global degree of freedom (DOF) for which the plot is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, or 3 for the global Z-direction; 4, 5 or 6 for the components of the rotation vector in the global X, Y and Z axes respectively.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (a).
Phase:	This option allows you to specify if you want a plot of phase angle as well as RAO. The default option is *No*, which specifies that no plot of phase angle is produced. Whatever option you specify here is of course immaterial if you specify that no output plots are to be created.
Fig. Title:	A descriptive title to be associated with the output. This entry is optional.

Notes

(a) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

RAOS - FORCE

Purpose

To request plots of RAOs of element restoring forces.

Table Input

Input:	Description
Element:	The element (number or label) for which the force RAO is required. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Local Node:	This option allows you to choose between three locations on the specified element. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis.
Variable:	This list allows you to select a relevant output parameter. The entries are largely self-explanatory. See Note (a).
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (b).
Phase:	This option allows you to specify if you want a plot of phase angle as well as RAO. The default option is No, which specifies that no plot of phase angle is produced. Whatever option you specify here is of course immaterial if you specify that no output plots are to be created.
Location:	This parameter is appropriate when you request bending stress, bending strain, von Mises stress, or pressure. See Note (c).
Fig. Title:	A descriptive title to be associated with the output. This entry is optional.

Notes

(a) Refer to Force Variable Input for further information on acceptable *Variable* values.

(b) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

(c) Refer to Location Parameter Input for further information on the significance of the *Location* input.

RAOS - REACTION

Purpose

To request plots of RAOs of reaction at specified restrained nodes.

Table Input

Input:	Description
Node:	The node (number or label) for which the reaction RAO is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global degree of freedom (DOF) for which the plot is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, or 3 for the global Z-direction; 4, 5 or 6 for moments about the global X, Y and Z axes respectively.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (a).

Phase:	This option allows you to specify if you want a plot of phase angle as well as RAO. The default option is *No*, which specifies that no plot of phase angle is produced. Whatever option you specify here is of course immaterial if you specify that no output plots are to be created.
Fig. Title:	A descriptive title to be associated with the output. This entry is optional.

Notes

(a) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

RAOS - FLOATING BODY - MOTION

Purpose

To request RAO motion plots for floating bodies, in the floating body local axes.

Table Input

Input:	Description
Floating Body:	The name of the floating body for which the motion RAO is requested.
DOF:	The degree of freedom in the local floating body axis system for which the motion RAO is requested. The options are *Heave, Surge, Sway, Yaw, Roll* and *Pitch.*
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.

Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (a).
Phase:	This option allows you to specify if you want a plot of phase angle as well as RAO. The default option is *No*, which specifies that no plot of phase angle is produced. Whatever option you specify here is of course immaterial if you specify that no output plots are to be created.
Fig. Title:	A descriptive title to be associated with the output. This entry is optional.

Notes

(a) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

1.8.4.7 *RESTART

PURPOSE

To indicate that a postprocessing run is to be restarted from an analysis file of different stub name.

THEORY

Refer to Restart Analyses for further information on this feature.

KEYWORD FORMAT

A single line defining the file name.

```
LAST=File Name
```

The keyword is optional, and is not required if the $DATABASE POSTPROCESSING section is contained within the same file as the actual analysis data.

TABLE INPUT

Input:	Description
Restart File:	The name of the analysis from which the postprocessing run is to be restarted. This may be input in terms of the analysis name (e.g. *Example1*) or the full path of the analysis (e.g. *C:\Flexcom\Examples\Example1-Static*).

1.8.4.8 *SNAPSHOT

PURPOSE

To request the creation of snapshot plots.

THEORY

Refer to Database Postprocessing for further information on this feature.

KEYWORD FORMAT

Blocks of lines repeated as often as necessary. Each block begins with a line defining the type of snapshot. This is followed by lines requesting the actual snapshot plots.

Block requesting motion snapshots. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=MOTION [, SET=Set Name]
DOF, Time [, Scale] [, UNITS=Unit]
[TITLE=Figure Title]
```

Block requesting force snapshots. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=FORCE [, SET=Set Name]
Variable, Time [, Scale] [, LOCATION=Location] [, UNITS=Unit]
[TITLE=Figure Title]
```

Refer to Force Variable Input for further information on acceptable *Variable* values. Refer to Location Parameter Input for further information on the significance of the *Location* input.

Block requesting structure snapshots. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=STRUCTURE [, SET=Set Name]
Time [, Scale] [, Viewpoint X] [, Viewpoint Y] [, Viewpoint Z] [, UNITS=Unit]
[TITLE=Figure Title]
```

Block requesting snapshot of seabed clearance. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=SEABED [, SET=Set Name]
Time [, Scale] [, UNITS=Unit]
[TITLE=Figure Title]
```

Set Name defaults to all elements (set ALL). *Scale* defaults to 1. The Unit entry defaults to base units. Figure Title defaults to blank.

SNAPSHOT - MOTION

Purpose

To request a plot of the distribution of motion in one of the global DOFs at a particular solution time.

Table Input

Input:	Description
DOF:	The global degree of freedom (DOF) for which the snapshot is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, or 3 for the global Z-direction; 4, 5 or 6 for the components of the rotation vector in the global X, Y and Z axes respectively; or 7 for the magnitude of 4, 5, and 6.
Time:	The analysis time at which the snapshot is required. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.

Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (b).
Element Set:	The element set for which the snapshot is to be produced. All plots are produced by default for all elements of the structure which corresponds to the element set *All*.

Notes

(a) When you specify a time value in the second column, Flexcom searches the analysis database for the solution time nearest to the value you input. For a snapshot at the analysis finish time, specify a time which is greater than the finish time you specified.

(b) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

SNAPSHOT - FORCE

Purpose

To request a plot of the distribution of a restoring force at a particular solution time.

Table Input

Input:	Description
Variable:	This list allows you to select a relevant output parameter. The entries are largely self-explanatory. See Note (a).
Time:	The analysis time at which the snapshot is required. See Note (b).
Title:	A descriptive title to be associated with the output. This entry is optional.

Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Element Set:	The element set for which the plot is to be produced. All plots are produced by default for all elements of the structure, which corresponds to the element set *All*.
Location:	This parameter is appropriate when you request bending stress, bending strain, von Mises stress, or pressure. See Note (c).
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (d).

Notes

(a) Refer to Force Variable Input for further information on acceptable *Variable* values.

(b) When you specify a time value, Flexcom searches the analysis database for the solution time nearest to the value you input. For a snapshot at the analysis finish time, specify a time which is greater than the finish time you specified.

(c) Refer to Location Parameter Input for further information on the significance of the *Location* input.

(d) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

SNAPSHOT - STRUCTURE

Purpose

To request a structure geometry plot at a particular solution time.

Table Input

Input:	Description
Time:	The analysis time at which the snapshot is required. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
Viewpoint X	The X component of the viewpoint vector. This entry is optional, and defaults to a value of 0.
Viewpoint Y	The Y component of the viewpoint vector. This entry is optional, and defaults to a value of 0.
Viewpoint Z	The Z component of the viewpoint vector. This entry is optional, and defaults to a value of 1.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Element Set:	The element set for which the plot is to be produced. All plots are produced by default for all elements of the structure, which corresponds to the element set *All*.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (b).

Notes

(a) When you specify a time value, Flexcom searches the analysis database for the solution time nearest to the value you input. For a snapshot at the analysis finish time, specify a time which is greater than the finish time you specified.

(b) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

SNAPSHOT - SEABED CLEARANCE

Purpose

To request a plot of the seabed clearance at a particular solution time.

Table Input

Input:	Description
Time:	The solution time at which the snapshot is required. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Element Set:	The element set for which the snapshot is to be produced. All plots are produced by default for all elements of the structure, which corresponds to the element set *All*.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (b).

Notes

(a) When you specify a time value, Flexcom searches the analysis database for the solution time nearest to the value you input. For a snapshot at the analysis finish time, specify a time which is greater than the finish time you specified.

(b) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

1.8.4.9 *SPECTRUM

PURPOSE

To request the creation of spectrum plots.

THEORY

Refer to Database Postprocessing for further information on this feature.

KEYWORD FORMAT

The keyword begins with an optional line defining miscellaneous parameters for all the spectrum plots. This is then followed by blocks of lines repeated as often as necessary to request the actual spectrum plots.

Line defining miscellaneous parameters:

```
[STEP=Time Step] [, ENSEMBLES=No. of Ensembles] [, START=Start Time] [, CUTOF
```

Block requesting kinematic spectra. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=KINEMATIC
Node (Number or Label), DOF [, Scale] [, PARA=Parameter] [, UNITS=Unit]
[TITLE=Figure Title]
```

Block requesting force spectra. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=FORCE
Element (Number or Label), Local Node, Variable [, Scale] [, LOCATION=Locatio
[TITLE=Figure Title]
```

Refer to Force Variable Input for further information on acceptable *Variable* values. Refer to Location Parameter Input for further information on the significance of the *Location* input. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis.

Block requesting reaction spectra. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=REACTION
Node (Number or Label), DOF [, Scale] [, UNITS=Unit]
[TITLE=Figure Title]
```

Block requesting a spectrum of wave elevation. Line 3 is optional.

```
TYPE=ELEVATION
Scale [, Target Spectrum Flag] [, UNITS=Unit]
[TITLE=Figure Title]
```

A `Time Step` value is required in a random sea dynamic analysis with a variable time-step; it is ignored in the case of an analysis with a fixed step. `No. of Ensembles` defaults to 4. Start Time defaults to the analysis start time. If no `Cut-off Frequency` is specified, Flexcom calculates a suitable default based on the analysis time variables. If you specify a node/element label rather than a node/element number, it must be enclosed in {} brackets. `Scale` defaults to 1 where optional. `Parameter` can be M (for motion, the default), V (for velocity) or A (for acceleration). The latter two are appropriate only if the motion database contains these parameters. `Target Spectrum Flag` should be set to 1 if the target spectrum is to be created in addition to the realised spectrum. The Unit entry defaults to base units. `Figure Title` defaults to blank.

SPECTRUM - GENERAL PARAMETERS

Purpose

To define a number of parameters relating to the calculation of spectra.

Table Input

Input:	Description
Start Time:	The start time for the calculation of spectra. Flexcom excludes any values before this time. Use this entry to exclude initial transients from spectrum calculations.

Ensembles:	The number of ensembles to be used in calculating spectra. See Note (a).
Time Step:	The time step to be used when calculating spectra from the results of a random sea dynamic analysis with a variable time step. See Note (b). If the analysis used a fixed time step then this entry is not required, and any value you might specify is ignored.
Cut-off Frequency:	The spectrum cut off frequency in Hz. Flexcom truncates all output at this frequency. This entry is optional; by default, Flexcom calculates an appropriate value from the simulation length and the number of ensembles you specify.

Notes

(a) The procedure used by Flexcom in calculating spectra is as follows. Firstly, the output timetrace is divided equally into a number of smaller timetraces or ensembles. A spectrum for each ensemble is then calculated using the Fast Fourier Transform (FFT) algorithm. Finally, the actual spectrum to be output is found as an average of the spectra calculated for each ensemble. The number of ensembles should be sufficiently large to ensure that the effects of the phase associated with each frequency component in the FFT process are minimised. You specify the number of ensembles to be used in this process using the *Ensembles* entry above. This value should always be greater than 1.

(b) The FFT algorithm requires a record with a fixed time step. When you perform a variable step analysis, obviously such a record is not available, and so Flexcom must synthesise one by interpolating from the variable step record. The *Time Step* entry tells Flexcom what time step to use in the synthesised record.

SPECTRUM - KINEMATIC

Purpose

To request a spectrum of the motion of a node in one of the global DOFs.

Table Input

Input:	Description
Node:	The node (number or label) for which the motion spectrum is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global degree of freedom (DOF) at this node for which the spectrum is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, or 3 for the global Z-direction; 4, 5 or 6 for the components of the rotation vector in the global X, Y and Z axes respectively; or 7 for the magnitude of 4, 5, and 6; or 8 for the magnitude of 1, 2, and 3.
Parameter:	This entry allows you to select the response parameter to be plotted against time. The options are *Motion* (the default), *Velocity* and *Acceleration*. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (b).

Notes

(a) In order to produce spectra of velocity and acceleration you must have requested extended output to the analysis database. Refer to the `*DATABASE CONTENT` keyword for more details.

(b) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

SPECTRUM - FORCE

Purpose

To request a spectrum of restoring force in a particular element.

Table Input

Input:	Description
Element:	The element (number or label) for which the force spectrum is required. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Local Node:	This option allows you to choose between three locations on the specified element. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis.
Variable:	This list allows you to select a relevant output parameter. The entries are largely self-explanatory. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Location:	This parameter is appropriate when you request bending stress, bending strain, von Mises stress, or pressure. See Note (b).

Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (c).

Notes

(a) Refer to Force Variable Input for further information on acceptable *Variable* values.

(b) Refer to Location Parameter Input for further information on the significance of the *Location* input.

(c) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

SPECTRUM - REACTION

Purpose

To request a spectrum of a reaction at a restrained node.

Table Input

Input:	Description
Node:	The node (number or label) for which the reaction spectrum is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global degree of freedom (DOF) for which the plot is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, or 3 for the global Z-direction; 4, 5 or 6 for moments about the global X, Y and Z axes respectively, or 7 for the magnitude of 4, 5, and 6; or 8 for the magnitude of 1, 2, and 3.

Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (a).

Notes

(a) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

SPECTRUM - ELEVATION

Purpose

To request a spectrum of water surface elevation (at Y = Z = 0).

Table Input

Input:	Description
Scale:	A scale factor to apply to the data.
Title:	A descriptive title to be associated with the output. This entry is optional.
Plot Target Wave Spectrum:	This option allows you to plot the target wave spectrum, in addition to the realised spectrum, which is useful for comparison purposes.

Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (a).

Notes

(a) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

1.8.4.10 *STANDARD OUTPUT

PURPOSE

To quickly request a selection of commonly used outputs, namely effective tension, resultant bending moment and von Mises stress.

THEORY

Refer to Standard Output for further information on this feature.

KEYWORD FORMAT

A single line defining the element set for which the standard output is required, repeated as often as necessary. If no element set is listed, it is assumed that output is required for all elements (i.e. SET=ALL).

```
[SET=Set Name]
```

TABLE INPUT

Input:	Description

Element Set:	The element set for which the standard output is required. If no element sets are listed, it is assumed that output is required for all elements (i.e. element set `All`).

1.8.4.11 *STATISTICS

PURPOSE

To request the creation of statistics plots.

THEORY

Refer to Database Postprocessing for further information on this feature.

KEYWORD FORMAT

Blocks of lines repeated as often as necessary. Each block begins with a line defining the type of statistics plot. This is followed by lines requesting the actual plots.

Block requesting kinematic statistics. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=KINEMATIC [, SET=Set Name]
Statistic, DOF [, PARA=Parameter] [, Start Time] [, Scale] [, UNITS=Unit]
[TITLE=Figure Title]
```

Block requesting force statistics. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=FORCE [, SET=Set Name]
Statistic, Variable [, Start Time] [, Scale] [, LOCATION=Location] [, UNITS=U
[TITLE=Figure Title]
```

Block requesting statistics of seabed reaction forces. Lines 2 and 3 can be repeated as often as necessary. Line 3 is optional.

```
TYPE=SEABED_REACTION [, SET=Set Name]
Statistic, DOF, [, Start Time] [, Scale] [, UNITS=Unit]
[TITLE=Figure Title]
```

Refer to Force Variable Input for further information on acceptable *Variable* values. Refer to Location Parameter Input for further information on the significance of the *Location* input.

If an element set is specified then its composition must be defined either here or in the Analysis data. If a plot of extreme value is requested, the `*EXTREMA` keyword must be invoked. `Set Name` defaults to all elements (set ALL). `Statistic` can be ENVELOPE, MEAN, DEVIATION or EXTREME. `Parameter` can be M (for motion, the default), V (for velocity) or A (for acceleration). The latter two are appropriate only if the motion database contains these parameters. Start Time defaults to the analysis start time. Scale defaults to 1. The Unit entry defaults to base units. `Figure Title` defaults to blank.

STATISTICS - KINEMATIC

Purpose

To request a plot of motion statistics.

Table Input

Input:	Description
Statistic:	This entry allows you to specify the statistical parameter required. *Envelope* (the default) in this context means max/min envelopes, *Mean* is mean value, *Deviation* is standard deviation, and *Extreme* is extreme value calculated using either the Rayleigh or Weibull distribution. See Note (a).
DOF:	The global degree of freedom (DOF) for which the statistics plot is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, or 3 for the global Z-direction; 4, 5 or 6 for the components of the rotation vector about the global X, Y and Z axes respectively, or 7 for the magnitude of 4, 5, and 6; or 8 for the magnitude of 1, 2, and 3.
Parameter:	This entry allows you to select the response parameter to be plotted against time. The options are *Motion* (the default), *Velocity* and *Acceleration*. See Note (b).
Start Time:	The start time for the calculation of statistics. Flexcom excludes any values before this time. Use this entry to exclude initial transients from statistical calculations.

Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Element Set:	The element set for which the statistics plot is to be produced. The plot is produced by default for all elements of the structure, which corresponds to the element set *All*.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (c).

Notes

(a) If you request an extreme value plot, you must use the relevant *Extreme Values* table to nominate the distribution (Rayleigh or Weibull) to use in calculating these extrema, and to input parameter values appropriate to the distribution you nominate.

(b) In order to produce spectra of velocity and acceleration you must have requested extended output to the analysis database. Refer to the `*DATABASE CONTENT` keyword for more details.

(c) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

STATISTICS - FORCE

Purpose

To request a plot of force statistics.

Table Input

Input:	Description
Parameter:	This option allows you to specify the statistical parameter required. *Envelope* (the default) in this context means max/min envelopes, *Mean* is mean value, *Deviation* is standard deviation, and *Extreme* is extreme value calculated using either the Rayleigh or Weibull distribution. See Note (a).
Variable:	This list allows you to select a relevant output parameter. The entries are largely self-explanatory. See Note (b).
Start Time:	The start time for the calculation of statistics. Flexcom excludes any values before this time. Use this entry to exclude initial transients from statistical calculations.
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Element Set:	The element set for which the plot is to be produced. The plot is produced by default for all elements of the structure, which corresponds to the element set *All*.
Location:	This parameter is appropriate when you request bending stress, bending strain, von Mises stress, or pressure. See Note (c).
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (d).

Notes

(a) If you request an extreme value plot, you must use the the *EXTREMA keyword to nominate the distribution (Rayleigh or Weibull) to use in calculating these extrema, and to input parameter values appropriate to the distribution you nominate.

(b) Refer to Force and Stress Outputs for a detailed discussion of the various output parameters.

(c) Refer to Location Parameter Input for further information on the significance of the *Location* input.

(d) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

STATISTICS - SEABED REACTION

Purpose

To request a plot of seabed reaction force statistics.

Table Input

Input:	Description
Statistic:	This entry allows you to specify the statistical parameter required. *Envelope* (the default) in this context means max/min envelopes, *Mean* is mean value, *Deviation* is standard deviation, and *Extreme* is extreme value calculated using either the Rayleigh or Weibull distribution. See Note (a).
DOF:	The global degree of freedom (DOF) for which the statistics plot is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, 3 for the global Z-direction, or 4 for the magnitude of 1, 2, and 3.
Start Time:	The start time for the calculation of statistics. Flexcom excludes any values before this time. Use this entry to exclude initial transients from statistical calculations.

Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Element Set:	The element set for which the statistics plot is to be produced. The plot is produced by default for all elements of the structure, which corresponds to the element set *All*.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (b).

Notes

(a) If you request an extreme value plot, you must use the relevant *Extreme Values* table to nominate the distribution (Rayleigh or Weibull) to use in calculating these extrema, and to input parameter values appropriate to the distribution you nominate.

(b) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

1.8.4.12 *TIMETRACE

PURPOSE

To request the creation of timetrace plots.

THEORY

Refer to Database Postprocessing for further information on this feature.

KEYWORD OVERVIEW

Blocks of lines repeated as often as necessary. Each block begins with a line defining the type of timetrace plot. This is followed by lines requesting the actual timetraces.

TIMETRACE – KINEMATIC

Purpose

To request plots of the time history of the motion of nodes throughout the analysis.

Keyword Format

Block requesting kinematic timetraces. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=KINEMATIC
[START=Start Time] [, END=End Time]
Node (Number or Label), DOF, [Scale] [, PARA=Parameter][, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

Start Time and *End Time* default to the analysis start and end times, respectively. If you specify a node label rather than a node number, it must be enclosed in {} brackets. *Scale* defaults to 1. *Parameter* can be M (for motion), D (for displacement), V (for velocity) or A (for acceleration). The latter two are appropriate only if the motion database contains these parameters. The Unit entry is described in Note (c). *Figure Title* defaults to blank. *Unique Plot ID* can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Node:	The node (number or label) for which the motion timetrace is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global degree of freedom (DOF) at this node for which the timetrace is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, or 3 for the global Z-direction; 4, 5 or 6 for the components of the rotation vector in the global X, Y and Z axes respectively.

Parameter:	This entry allows you to select the response parameter to be plotted against time. The options are *Motion* (the default), *Displacement, Velocity* and *Acceleration*. See Note (a).
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time. See Note (b).
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time. See Note (b).
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (c).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) In order to produce timetraces of velocity and acceleration you must have requested extended output to the analysis database. Refer to the `*DATABASE CONTENT` keyword for more details.

(b) Use the Start facility to exclude initial transients if required. The *`Start`* and *`End`* facilities can be used together to break up a longer timetrace in several smaller ones, to enable the finer detail of the response to be examined.

(c) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – FORCE

Purpose

To request time histories of the variation of restoring forces in particular elements throughout the analysis.

Keyword Format

Block requesting force timetraces. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=FORCE
[START=Start Time] [, END=End Time]
Element (Number or Label), Local Node, Variable, [Scale] [, LOCATION=Locatior
[TITLE=Figure Title], [ID=Unique Plot ID]
```

Refer to Force Variable Input for further information on acceptable *Variable* values. Refer to Location Parameter Input for further information on the significance of the *Location* input. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis.

Start Time and *End Time* default to the analysis start and end times, respectively. Scale defaults to 1. If you specify an element label rather than an element number, it must be enclosed in {} brackets.

Table Input

Input:	Description
Element:	The element (number or label) for which the force timetrace is required. If you specify an element label rather than an element number, it must be enclosed in {} brackets.

Local Node:	This option allows you to choose between three locations on the specified element. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis.
Variable:	This list allows you to select a relevant output parameter. The entries are largely self-explanatory. See Note (a).
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time. See Note (b).
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time. See Note (b).
Location:	This parameter is appropriate when you request bending stress, bending strain, von Mises stress, or pressure. See Note (c).
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (d).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

NOTES

(a) Refer to Force Variable Input for further information on acceptable *Variable* values.

(b) Use the Start facility to exclude initial transients if required. The *Start* and *End* facilities can be used together to break up a longer timetrace into several smaller ones, to enable the finer details of the response to be examined.

(c) Refer to Location Parameter Input for further information on the significance of the *Location* input.

(d) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – REACTION

Purpose

To request time history plots of reaction forces at nodes restrained by boundary conditions.

Keyword Format

Block requesting reaction timetraces. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=REACTION
[START=Start Time] [, END=End Time]
Node (Number or Label), DOF, [Scale] [, AXIS=Axis Name] [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

`Start Time` and `End Time` default to the analysis start and end times, respectively. If you specify a node label rather than a node number, it must be enclosed in {} brackets. Scale defaults to 1. Any `Axis Name` you reference must be defined using `*AXIS/VECTOR`. The `Unit` entry is described in Note (b). `Figure Title` defaults to blank. `Unique Plot ID` can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description

Node:	The node (number or label) for which the reaction timetrace is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global degree of freedom (DOF) for which the plot is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, or 3 for the global Z-direction; 4, 5 or 6 for moments about the global X, Y and Z axes respectively, or 7 for the magnitude of 4, 5, and 6; or 8 for the magnitude of 1, 2, and 3.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Axis:	The name of the axis system for which the angle timetrace is required. This entry defaults to the global XYZ axes.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time. See Note (b).
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time. See Note (b).
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (b).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) Use the `Start` facility to exclude initial transients if required. The `Start` and `End` facilities can be used together to break up a longer timetrace into several smaller ones, to enable the fine detail of the response to be examined.

(b) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – SEABED REACTION

Purpose

To request time history plots of reaction forces at nodes which contact the seabed.

Keyword Format

Block requesting seabed reaction timetraces. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=SEABED_REACTION
[START=Start Time] [, END=End Time]
Node (Number or Label), DOF, [Scale] [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

Start Time and *End Time* default to the analysis start and end times, respectively. If you specify a node label rather than a node number, it must be enclosed in {} brackets. Scale defaults to 1. The *Unit* entry is described in Note (b). *Figure Title* defaults to blank. *Unique Plot ID* can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Node:	The node (number or label) for which the reaction timetrace is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.

DOF:	The global degree of freedom (DOF) for which the plot is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, 3 for the global Z-direction, or 4 magnitude of 1, 2, and 3.
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time. See Note (b).
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time. See Note (b).
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (b).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) Use the *`Start`* facility to exclude initial transients if required. The *`Start`* and *`End`* facilities can be used together to break up a longer timetrace into several smaller ones, to enable the fine detail of the response to be examined.

(b) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – WAVE ELEVATION

Purpose

To request a plot of the time history of wave elevation (at Y = Z = 0).

Keyword Format

Block requesting a timetrace of water surface elevation. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=ELEVATION
[START=Start Time] [, END=End Time]
Scale [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

`Start Time` and `End Time` default to the analysis start and end times, respectively. The Unit entry is described in Note (a). `Figure Title` defaults to blank. `Unique Plot ID` can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Scale:	A scale factor to apply to the data.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – TIME STEP

Purpose

To request a plot of the time history of time step size in a variable step analysis.

Keyword Format

Block requesting a timetrace of analysis time-step. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=TIME-STEP
[START=Start Time] [, END=End Time]
Scale [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

Start Time and *End Time* default to the analysis start and end times, respectively. The *Unit* entry is described in Note (a). *Figure Title* defaults to blank. *Unique Plot ID* can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.

End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – ELEMENT-ELEMENT ANGLES

Purpose

To request a plot of the time history of the angle between two elements.

Keyword Format

Block requesting timetraces of the angle between two elements. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=ELEMENT_ELEMENT_ANGLE
[START=Start Time] [, END=End Time]
Element 1 (Number or Label), Element 2 (Number or Label), [Scale] [, UNITS=Ur
[TITLE=Figure Title], [ID=Unique Plot ID]
```

`Start Time` and `End Time` default to the analysis start and end times, respectively. Scale defaults to 1. The `Unit` entry is described in Note (c). Figure Title defaults to blank. If you specify an element label rather than an element number, it must be enclosed in {} brackets. `Figure Title` defaults to blank. `Unique Plot ID` can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Element 1:	The first element (number or label). If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Element 2:	The second element (number or label). If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (c).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) Refer to Angle between Two Elements for a detailed discussion of this facility and of the significance of the inputs.

(b) One (or indeed both) of the elements you specify can be an auxiliary element.

(c) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – ELEMENT-VECTOR ANGLES

Purpose

To request a plot of the time history of the angle between an element and a vector in their respective instantaneous locations.

Keyword Format

Block requesting timetraces of the angle between an element and a vector. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=ELEMENT_VECTOR_ANGLE
[START=Start Time] [, END=End Time]
Element (Number or Label), [Scale], VECTOR=Vector Name [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

`Start Time` and `End Time` default to the analysis start and end times, respectively. Any `Vector Name` you reference must be defined using `*AXIS/VECTOR`. *Scale* defaults to 1. The `Unit` entry is described in Note (d). `Figure Title` defaults to blank. If you specify an element label rather than an element number, it must be enclosed in {} brackets. `Figure Title` defaults to blank. `Unique Plot ID` can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description

Element:	The element (number or label) for which the angle timetrace is required. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Vector:	The name of the vector for which the angle timetrace is required.
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (d).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) Refer to Angle between an Element and a Vector for a detailed discussion of this facility and of the significance of the inputs.

(b) You must specify the location and initial orientation of each vector referenced using the Vectors table.

(c) The element you specify can be an auxiliary element.

(d) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – ELEMENT-AXIS ANGLES

Purpose

To request a plot of the time history of the angle between an element and an axis system in their respective instantaneous locations.

Keyword Format

Block requesting timetraces of the angle between an element and an axis system. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=ELEMENT_AXIS_ANGLE
[START=Start Time] [, END=End Time]
Element (Number or Label), [Scale] [, AXIS=Axis Name] [, ANG=Angle Type] [, U
[TITLE=Figure Title], [ID=Unique Plot ID]
```

Start Time and *End Time* default to the analysis start and end times, respectively. If you specify an element label rather than an element number, it must be enclosed in {} brackets. Any *Axis Name* you reference must be defined using *AXIS/VECTOR. *Angle Type* may be AC (actual), XY (projected-XY), YZ (projected-YZ) or XZ (projected-XZ). Scale defaults to 1. The Unit entry is described in Note (c). *Figure Title* defaults to blank. *Unique Plot ID* can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Element:	The element (number or label) for which the angle timetrace is required. If you specify an element label rather than an element number, it must be enclosed in {} brackets.

Axis:	The name of the axis system for which the angle timetrace is required. See Notes (a) and (b). This entry defaults to the global XYZ axes. To specify a local axis system, simply type in the name or label associated with the local axes. If you change from the default, the axes you specify become the default for subsequent postprocessing requests, until you explicitly change them again.
Angle:	The type of angle output required. The options are *Actual* (the default), Projected-xy, *Projected-yz,* and *Projected-xz*. See Note (a). If you change from this default, the value you specify becomes the default for subsequent postprocessing requests, until you explicitly change it again.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (c).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) Refer to Angle between an Element and an Axis System for a detailed discussion of this facility and of the significance of the inputs.

(b) If you request a timetrace of angle relative to a local axis system, then you must specify the location and initial orientation of the local axes using the *Axis Systems* table.

(c) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – VECTOR-VECTOR ANGLES

Purpose

To request a plot of the time history of the angle between two vectors in their respective instantaneous locations.

Keyword Format

Block requesting timetraces of the angle between two vectors. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=VECTOR_VECTOR_ANGLE
[START=Start Time] [, END=End Time]
VECTOR1=Vector Name 1, VECTOR2=Vector Name 2, [Scale] [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

`Start Time` and `End Time` default to the analysis start and end times, respectively. Any `Vector Name` you reference must be defined using `*AXIS/VECTOR`. *Scale* defaults to 1. The `Unit` entry is described in Note (c). `Figure Title` defaults to blank. `Unique Plot ID` can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
First Vector:	The number of the first vector.

Second Vector:	The number of the second vector.
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (c).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) Refer to Angle between a Vector and a Vector for a detailed discussion of this facility and of the significance of the inputs.

(b) You must specify the location and initial orientation of each vector referenced using the *Vectors* table.

(c) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – GUIDE SURFACE REACTION

Purpose

To request plots of the time history of the contact reaction at a flat guide surface throughout the analysis.

Keyword Format

Block requesting timetraces of the reaction at a flat guide surface. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=CONTACT
[START=Start Time] [, END=End Time]
Flat Guide Name, Contact DOF, [Scale] [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

`Start Time` and `End Time` default to the analysis start and end times, respectively. *Scale* defaults to 1. `Figure Title` defaults to blank. `Contact DOF` can be 1, 2 or 3, for reactions in the global X, Y and Z directions, or 4, for the reaction magnitude. The Unit entry is described in Note (a). `Figure Title` defaults to blank. `Unique Plot ID` can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Guide Name:	The name of the flat guide surface for which the reaction timetrace is required.

DOF:	The global degree of freedom (DOF) for which the reaction timetrace is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, 3 for the global Z-direction, or 4 for the magnitude of reaction.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – ZERO-GAP GUIDE REACTION

Purpose

To request plots of the time history of the contact reaction at a zero-gap guide throughout the analysis.

Keyword Format

Block requesting timetraces of the reaction at a zero-gap guide. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=GUIDE
[START=Start Time] [, END=End Time]
Zero-Gap Guide No., Contact DOF, [Scale] [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

`Start Time` and `End Time` default to the analysis start and end times, respectively. Scale defaults to 1. Contact DOF can be 1, 2 or 3, for reactions in the global X, Y and Z directions, or 4, for the reaction magnitude. The Unit entry is described in Note (a). `Figure Title` defaults to blank. `Unique Plot ID` can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Zero-Gap Guide No.:	The number of the zero-gap guide for which the reaction timetrace is required. This corresponds to the order in which it appeared in the definition of zero-gap guides in the model definition.
DOF:	The global degree of freedom (DOF) for which the reaction timetrace is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, 3 for the global Z-direction, or 4 for the magnitude of reaction.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.

Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – PIPE-IN-PIPE REACTION

Purpose

To request plots of the time history of the contact reaction at a pipe-in-pipe connection throughout the analysis.

Keyword Format

Block requesting timetraces of the reaction at a pipe-in-pipe connection. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=PIP
[START=Start Time] [, END=End Time]
Pipe-In-Pipe Connection No., PIP DOF, [Scale] [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

`Start Time` and `End Time` default to the analysis start and end times, respectively. Scale defaults to 1. `Figure Title` defaults to blank. PIP DOF can be 2 or 3, for reactions in the normal or transverse directions, or 4, for the reaction magnitude. The Unit entry is described in Note (c). `Figure Title` defaults to blank. `Unique Plot ID` can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Pipe-in-Pipe Connection No.:	The number of the pipe-in-pipe connection for which the reaction timetrace is required. This corresponds to the order in which it appeared in the definition of connections in the model definition. See Note (a).
DOF:	The local degree of freedom (DOF) for which the reaction timetrace is required. Specify a value of 1 for the axial direction, 2 for the normal direction, 3 for the transverse direction, or 4 for the magnitude of reaction. See Note (b).
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (c).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) Pipe-in-pipe connections may be defined in several ways. Connections may be created directly between individual nodes, or generated between groups of nodes, and there are both “standard” and “sliding” connection types. In order to identify the pipe-in-pipe connection for which the reaction timetrace is required, you can check the connection number by examining the ‘Pipe-in-Pipe Connections Data’ section of the output (.out) file.

(b) As described in Standard Connections, a pipe-in-pipe connection can be considered to act like a spring between the connected nodes, with the direction of the spring acting normal to the outer pipe. In actual fact, an equivalent “spring” is also introduced at 90° to the normal direction, mainly to aid solution robustness. Another good reason for the inclusion of a second spring is that for concentric pipe sections, it is difficult to distinguish between the normal and transverse directions initially, until some relative motion occurs. A DOF input of 2 or 3 corresponds to these normal and transverse directions, respectively.

(c) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – KINETIC ENERGY

Purpose

To request a plot of the time history of total kinetic energy.

Keyword Format

Block requesting a timetrace of kinetic energy. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=ENERGY
[START=Start Time] [, END=End Time]
Scale [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

`Start Time` and `End Time` default to the analysis start and end times, respectively. The `Unit` entry is described in Note (a). `Figure Title` defaults to blank. `Unique Plot ID` can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – CLASHING REACTION

Purpose

To request plots of the time history of various parameters relating to clashing analyses.

Keyword Format

Block requesting timetraces of parameters related to clashing analyses. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=CLASH
[START=Start Time] [, END=End Time]
Region No., Clashing Parameter, [Scale] [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

Start Time and *End Time* default to the analysis start and end times, respectively. *Scale* defaults to 1. Figure Title defaults to blank. *Clashing Parameter* can be 1, 2 or 3, for minimum clearance, reaction force or clashing impulse, respectively. The *Unit* entry is described in Note (b). *Figure Title* defaults to blank. *Unique Plot ID* can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Region No.:	The number of the region for which the timetrace is required. This corresponds to the order in which it appeared in the definition of clashing region in the model definition.
Parameter:	The parameter for which the timetrace is required. The options are *Minimum Clearance* (the default), *Reaction Force* or *Clashing Impulse*. See Note (a).
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.

End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (b).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) There are three options for specifying the parameter for which the clearance is required as follows.

- *Minimum Clearance* – the minimum clearance between element sets for a clashing region. A clearance value of zero or less indicates that contact has occurred. A value of less than zero indicates that some degree of relative penetration has occurred between the lines.
- *Reaction Force* – the total reaction force for a clashing region. If there is more than one clashing location, the various reaction magnitudes are simply summed together. The reaction at any particular location is based on the contact stiffness times the relative penetration of the lines, plus any damping contribution (proportional to the relative velocity of approaching lines).
- *Clashing Impulse* – the integral of the reaction force with respect to time.

(b) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – LENGTH

Purpose

To request a plot of the time history of total set length. This is particularly useful for monitoring the length of winch sets.

Keyword Format

Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=LENGTH [, SET=Set Name]
[START=Start Time] [, END=End Time]
Scale [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

Set Name defaults to all elements (set ALL). *Start Time* and *End Time* default to the analysis start and end times, respectively. The *Unit* entry is described in Note (a). *Figure Title* defaults to blank. *Unique Plot ID* can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Element Set:	The element set for which a timetrace of length is to be created. The default is, ALL, the set containing all elements in the model.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.

Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – DISTANCE

Purpose

To request a plot of the time history of distance between two nodes. This is particularly useful for monitoring clearance between two points.

Keyword Format

Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=DISTANCE
[START=Start Time] [, END=End Time]
First Node (Number or Label), Second Node (Number or Label), Scale [, UNITS=
[TITLE=Figure Title], [ID=Unique Plot ID]
```

`Start Time` and `End Time` default to the analysis start and end times, respectively. `First Node` and `Second Node` are the node numbers (or labels) of the two nodes between which a timetrace of distance is required. The `Unit` entry is described in Note (a). `Figure Title` defaults to blank. `Unique Plot ID` can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time.
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time.
First Node:	The first node (number or label) of two, between which a timetrace of distance is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Second Node:	The second node (number or label) of two, between which a timetrace of distance is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – AERODYN

Purpose

To request plots of AeroDyn output where a Flexcom wind turbine analysis has been run in conjunction with the AeroDyn module.

Keyword Format

Block requesting timetraces of AeroDyn output channels. Lines 2-3 can be repeated and mixed as often as necessary.

```
TYPE=AERODYN
OUTPUT=AeroDyn Output Channel, [Scale] [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

The `AeroDyn Output Channel` must already be defined in the AeroDyn input file via the `*OUTPUTS` keyword. *Scale* defaults to 1. The *`Unit`* entry is described in Note (b). *`Figure Title`* defaults to blank. *`Unique Plot ID`* can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Output:	The name of the AeroDyn output channel. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.

Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (b).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

a) The output channel must already be defined in the AeroDyn input file. A full list of the AeroDyn output channels are available in the `*OUTPUTS` description.

b) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – SERVODYN

Purpose

To request plots of ServoDyn output where a Flexcom wind turbine analysis has been run using the ServoDyn module to control the turbine.

Keyword Format

Block requesting timetraces of ServoDyn output channels. Lines 2-3 can be repeated and mixed as often as necessary.

```
TYPE=SERVODYN
OUTPUT=ServoDyn Output Channel, [Scale] [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

The `ServoDyn Output Channel` must already be defined in the ServoDyn data input file which is referred to via the **SERVODYN, INPUT FILE=* specification. *Scale* defaults to 1. The *Unit* entry is described in Note (b). *Figure Title* defaults to blank. *Unique Plot ID* can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Output:	The name of the ServoDyn output channel. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (b).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

a) The output channel must already be defined in the ServoDyn input file. A full list of the ServoDyn output channels are as follows:

Category:	Name:	Description	Convention	Units
Pitch Control	BlPitchC1	Blade 1 pitch angle command	Positive towards feather about the minus zc1- and minus zb1-axes	deg
	BlPitchC2	Blade 2 pitch angle command	Positive towards feather about the minus zc2- and minus zb2-axes	deg
	BlPitchC3	Blade 3 pitch angle command	Positive towards feather about the minus zc3- and minus zb3-axes	deg
Generator and Torque Control	GenTq	Electrical generator torque	Positive reflects power extracted and negative represents a motoring-up situation (power input)	kN.m
	GenPwr	Electrical generator power	Same sign as GenTq	kW
High Speed Shaft Brake	HSSBrTqC	High-speed shaft brake torque command	Always positive (indicating dissipation of power)	kN.m

Category:	Name:	Description	Convention	Units
Nacelle Yaw Control	YawMomCom or YawMom	Nacelle yaw moment command	About the zn- and zp-axes	kN.m
Nacelle Tuned Mass Damper (TMD)	NTMD_XQ	Nacelle X TMD position (displacement)	Relative to rest position	m
	NTMD_XQD	Nacelle X TMD velocity	Relative to nacelle	m/s
	NTMD_YQ	Nacelle Y TMD position	Relative to rest position	m
	NTMD_YQD	Nacelle Y TMD velocity	Relative to nacelle	m/s
Tower Tuned Mass Damper (TMD)	TTMD_XQ	Tower X TMD position (displacement)	Relative to rest position	m

Category:	Name:	Description	Convention	Units
	TTMD_XQD	Tower X TMD velocity	Relative to tower	m/s
	TTMD_YQ	Tower Y TMD position	Relative to rest position	m
	TTMD_YQD	Tower Y TMD velocity	Relative to tower	m/s

b) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – INFLOWWIND

Purpose

To request plots of InflowWind output where a Flexcom wind turbine analysis has been run using the InflowWind module to provide the wind loading on the turbine

Keyword Format

Block requesting timetraces of InflowWind output channels. Lines 2-3 can be repeated and mixed as often as necessary.

```
TYPE=INFLOWWIND
OUTPUT=InflowWind Output Channel, [Scale] [, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

The `InflowWind Output Channel` must already be defined in the *InflowWind* data input file which is referred to via the *`*INFLOWWIND, INPUT FILE=`* specification. *Scale* defaults to 1. The *`Unit`* entry is described in Note (b). *`Figure Title`* defaults to blank. *`Unique Plot ID`* can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Output:	The name of the *InflowWind* output channel. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (b).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

b) The output channel must already be defined in the InflowWind input file. A full list of the InflowWind output channels are as follows:

Channel Name:	Description	Units
Wind1VelX ... Wind9VelX	Wind velocity in the horizontal Xi-axis (FAST Inertial Reference Frame) at up to 9 fixed points.	m/s
Wind1VelY ... Wind9VelY	Wind velocity in the horizontal Yi-axis (FAST Inertial Reference Frame) at up to 9 fixed points.	m/s
Wind1VelZ ... Wind9VelZ	Wind velocity in the horizontal Yi-axis (FAST Inertial Reference Frame) at up to 9 fixed points.	m/s

c) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – VESSEL

Purpose

To request plots of the time history of vessel motions throughout the analysis.

Keyword Format

Block requesting vessel timetraces. Lines 2-4 can be repeated and mixed as often as necessary.

```
TYPE=VESSEL, VESSEL=Vessel Name [, AXIS=Axis Type] [, LOCATION=X,Y,Z]
[START=Start Time] [, END=End Time]
DOF, [Scale] [, PARA=Parameter][, UNITS=Unit]
[TITLE=Figure Title], [ID=Unique Plot ID]
```

The `Vessel Name` specified here must be defined using [`*VESSEL INTEGRATED`](). `Axis Type` is optional and can be LOCAL (the default) or GLOBAL. An optional location may be defined in the [Local Vessel Axis System]() and relative to the vessel reference point. The location `X,Y` and `Z` all default to zero if unspecified. Optional `Start Time` and `End Time` default to the analysis start and end times, respectively - see Note (a). `DOF` is a number between 1 and 6 - see Note (b). `Scale` defaults to 1. `Parameter` can be M (for motion, the default), V (for velocity) or A (for acceleration). The latter two are appropriate only if the motion database contains these parameters. The `Unit` entry is described in Note (d). `Figure Title` defaults to blank. `Unique Plot ID` can be numbers or text to uniquely identify a plot for a later variable-variable timetrace, if necessary.

Table Input

Input:	Description
Vessel Name:	The name of the vessel whose motions are to be output.
DOF:	The degree of freedom of the output. See Note (b).
Parameter:	This entry allows you to select the response parameter to be plotted against time. The options are *Motion* (the default), *Velocity* and *Acceleration*. See Note (c).
Axis Type:	The axis system that forms the basis of the output. The options are Local and Global. This entry is optional and defaults to Local. See Note (b).
X:	The distance along the local vessel X axis between the vessel reference point and the location point (the point on the vessel for which the motion is calculated). This entry is optional and defaults to 0.

Y:	The distance along the local vessel Y axis between the vessel reference point and the location point (the point on the vessel for which the motion is calculated). This entry is optional and defaults to 0.
Z:	The distance along the local vessel Z axis between the vessel reference point and the location point (the point on the vessel for which the motion is calculated). This entry is optional and defaults to 0.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Start:	The timetrace start time. This entry is optional, and defaults to the analysis start time. See Note (b).
End:	The timetrace end time. This entry is optional, and defaults to the analysis finish time. See Note (b).
Unit:	The units to be used for the vertical axis of the plot. This entry is optional, and defaults to the base units for this output type. See Note (d).
Title:	A descriptive title to be associated with the output. This entry is optional.
ID:	Numbers or text to uniquely identify this plot. This entry is optional and only required if this plot is to be referenced in a variable-variable timetrace plot.

Notes

(a) Use the Start facility to exclude initial transients if required. The `Start` and `End` facilities can be used together to break up a longer timetrace in several smaller ones, to enable the finer detail of the response to be examined.

(b) If Axis Type is Local then a degree of freedom value of 1 represents the vessel heave; 2 the surge, 3 the sway, 4 the yaw, 5 the roll and 6 the pitch relative to the Local Vessel Axis System. The convention of successive yaw, roll and pitch is defined so that the yaw is first applied about the local vessel X-axis, next the roll is applied about the yawed y-axis, and finally the pitch is applied about the yawed and rolled z-axis. If Axis Type is Global then a value of 1 represents the global X-direction; 2 the global Y-direction; 3 the global Z-direction; and values 4, 5 and 6 represent the Euler angles that orient the vessel relative to the Global axis system applied in the same order as for the Local Axis Type.

(c) In order to produce timetraces of vessel velocity and acceleration you must have requested extended output to the analysis database. Refer to the `*DATABASE CONTENT` keyword for more details.

(d) The units entry explicitly specifies what units are to be assigned to the vertical axis of the plot. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid. For a list of the units accepted by Flexcom, refer to Units Reference Guide.

TIMETRACE – VARIABLE-VARIABLE

Purpose

To request plots of the variation of one variable against another. The independent variable on a timetrace plot is time. The dependent variable from two previously defined timetrace plots are plotted on the horizontal and vertical axes of a new plot. The source timetrace plots must be requested elsewhere under `*TIMETRACE` and have unique IDs.

Keyword Format

Block requesting a variable-variable timetrace. Lines 2-4 can be repeated as often as necessary.

```
TYPE=VARIABLE-VARIABLE
HORIZONTAL=Plot ID
VERTICAL=Plot ID
[TITLE=Figure Title]
```

`Plot ID` must be the ID of another plot requested under `*TIMETRACE`, see Note (a).

`Figure Title` defaults to blank.

Table Input

Input:	Description
Horizontal Plot ID:	The ID of the timetrace plot to use on the horizontal axis. See Note (a).
Vertical Plot ID:	The ID of the timetrace plot to use on the vertical axis. See Note (a).
Title:	A descriptive title to be associated with the output. This entry is optional.

Notes

(a) A unique ID can be optionally specified for each timetrace plot requested under `*TIMETRACE` and these are required here to identify which variables should be on the horizontal and vertical axes of the new variable-variable plot. Each of the referenced timetrace plots must have matching start and end times or a composite plot cannot be created.

1.8.5 $HISTOGRAM

This section corresponds to Histogram, which is a frequency domain general cycle counting tool. Refer to Histogram Overview for further details.

This section contains the following keywords:

- `*BINS` is used to specify bins or divisions for histogram output.
- `*PARA` is used to request response histograms of restoring force, stress, reaction, or relative rotation.
- `*PDF` is used to specify the probability density function to be used in the calculation of histograms.

- *SEASTATE FILES is used to specify all of the Flexcom random sea analyses on which the Histogram calculations are to be based, and to tabulate the percentage occurrences of the corresponding seastates.

1.8.5.1 *BINS

PURPOSE

To specify bins or divisions for histogram output.

THEORY

Refer to Histogram Overview for further information on this feature.

KEYWORD FORMAT

The format depends on whether the Max/Min or Direct specification option is being used.

Max/Min:

Two lines of data as follows:

```
TYPE=MAX/MIN
Minimum, Maximum, No. of Bins
```

Direct:

A single line repeated as often as necessary to define the number of bins or divisions of the histogram. A histogram is assumed to start at 0, so the first entry on the first line is assumed to be the upper end of the first bin; the second entry is the upper end of the second bin and so on. Up to 20 values may be specified on a single line.

```
Value, Value, Value, Value, Value, etc.
```

TABLE INPUT

Max-Min Specification

Input:	Description

Minimum:	The lower or smaller value of the two values bracketing the first or lowest bin in the histogram. The default is 0.
Maximum:	The upper or larger value of the two values bracketing the last or highest bin in the histogram.
No. of Bins:	The number of histogram bins or divisions.

Direct Specification

Input:	Description
Divisions for Histogram Output:	This entry allows you to enter values to specify how values are grouped in histogram output.

NOTES

(a) If you use *Max/Min* specification option to define, for example, a *Minimum* value of 0 (the default), a *Maximum* of 20, and set *No. of Bins* to 20, then the Histogram bins are 0-1, 1-2, 2-3 and so on up to 19-20 for the 20^{th} bin.

(b) If you use *Direct* specification option, you can specify up to 20 values per line, and you may use as many lines as you need to completely define your histogram. A histogram is assumed to start at 0, so the first value you specify is the upper limit of the first bin of the histogram.

(c) The bin distribution which you define applies to all histograms generated during a particular Histogram analysis. This means that if you want to generate histograms of parameters with very different magnitudes, then it may be better to do this in a series of separate analyses rather than in one Histogram run, when it might be difficult to define a bin distribution suitable for all parameters

1.8.5.2 *PARA

PURPOSE

To request response histograms of restoring force, stress, reaction, or relative rotation.

THEORY

Refer to Histogram Overview for further information on this feature.

KEYWORD FORMAT

A block of 2-3 lines for each histogram request. The format of the inputs varies with parameter type, as follows.

Lines for a response histogram of restoring force:

```
TYPE=1
Element (Number or Label), Local Node, Variable
[Scale]
```

Lines for a response histogram of stress:

```
TYPE=1
Element (Number or Label), Local Node, Variable, Stress Point
[Scale][,Do][,Di][,A][,Iyy]
```

Lines for a response histogram of reaction at a restrained node:

```
TYPE=2
Node (Number or Label), DOF
[Scale]
```

Lines for a response histogram of relative rotation between the two nodes of an element:

```
TYPE=3
Element (Number or Label)
```

If you specify a node/element label rather than a number, it must be enclosed in {} brackets. Scale defaults to 1. *Do*, *Di*, *A* and *Iyy* default to the corresponding values used in the first Flexcom random sea analysis specified in the seastate file.

The following Variable values are valid when requesting a force or stress histogram.

1. Axial Force
2. Local-y Shear
3. Local-z Shear
4. Torque
5. Local-y Moment
6. Local-z Moment

7. Effective Tension

8. von Mises Stress

9. Bending Stress

10. Axial Stress

11. Combined Stress

For `Variable` values of 8, 9 or 11 (von Mises, bending or combined stress), a `Stress Point` input is appropriate, where `Stress Point` is an integer value specifying a point on the cross-section. If omitted, `Stress Point` defaults to the point of maximum stress on the cross-section.

TABLE INPUT

Parameters - Force

Input:	Description
Element:	The element (number or label) for which the restoring force histogram is being requested. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Local Node:	This option allows you to choose between three locations on the specified element. *First Node* refers to the first specified node of the element, *Last Node* is the second specified node, and *Midpoint* is half-way between the two.
Variable:	This list allows you to select a relevant output parameter. The entries are largely self-explanatory. See Note (a).
Scale Factor:	This value is used to transform data in the Flexcom units to appropriate units for the Histogram output. This input is optional, and defaults to 1.

Parameters - Stress

Input:	Description

Element:	The element (number or label) for which the restoring force histogram is being requested. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Local Node:	This option allows you to choose between three locations on the specified element. *First Node* refers to the first specified node of the element, *Last Node* is the second specified node, and *Midpoint* is half-way between the two.
Variable:	This list allows you to select a relevant output parameter. The entries are largely self-explanatory. See Note (a).
De:	The external diameter of the element for which stresses are required. See Note (b).
Di:	The internal diameter of the element for which stresses are required. See Note (b).
A:	The effective cross-sectional area of the element for which stresses are required. See Note (b).
Iyy:	The second moment of area about the local y-axis of the element for which stresses are required. See Note (b).
Stress Point:	This gives a choice of location on the cross-section for which stresses are to be generated. Valid values are between 1 and 8 inclusive. This corresponds to an angle measured in degrees anti-clockwise from the local element cross-section y-axis where 1 = 0°, 2 = 45°, 3 = 90°, etc.
Scale Factor:	This value is used to transform data in the Flexcom units to appropriate units for the Histogram output. This input is optional, and defaults to 1.

Parameters - Reaction

Input:	Description

Node:	The restrained node (number or label) for which the reaction histogram is being requested. If you specify a node label rather than a node number, it must be enclosed in {} brackets. See Note (c).
DOF:	The global degree of freedom (DOF) of the reaction force. Specify a value of 1 for the global X-direction, 2, for global Y-direction, and 3 for the global Z-direction. DOFs 4, 5, and 6 refer to the components of the rotation vector at the node.
Scale Factor:	This value is used to transform data in the Flexcom units to appropriate units for the Histogram output. This input is optional, and defaults to 1.

Parameters - Rotation

Input:	Description
Element:	The element (number or label) for which the rotation histogram is being requested. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
DOF:	The degree of freedom (DOF) on which the rotation is based. Only values of 4, 5, or 6 are valid.

NOTES

(a) Refer to Force and Stress Outputs for a detailed discussion of the various output parameters.

(b) The `De`, `Di`, `A` and `Iyy` entries are optional. Where a particular value is not specified, then Histogram retrieves an appropriate value from the first Flexcom analysis in the list that you input using Flexcom Files.

(c) Histogram will generate an error if you request a reaction histogram for a node and DOF which are unrestrained.

1.8.5.3 *PDF

PURPOSE

To specify the probability density function to be used in the calculation of histograms.

THEORY

Refer to Histogram Overview for further information on this feature.

KEYWORD FORMAT

Two lines of data specifying the probability density function as follows.

```
PDF=PDF Type
[Threshold Bandwidth]
```

PDF Type can be RAYLEIGH DIRLIK or AUTO. The Threshold Bandwidth input is only relevant if PDF Type is set to AUTO. Threshold Bandwidth must be between 0 and 1.

TABLE INPUT

Input:	Description
Probability Density Function:	This option allows you to choose the probability density function (PDF) to be used in the calculation of histograms. The options are Rayleigh (the default), Dirlik and Automatic.
Threshold Bandwidth:	The threshold bandwidth below which Histogram is to automatically use the Rayleigh PDF, and above which Dirlik's PDF is to be used. The bandwidth must be between 0 and 1.

NOTES

(a) The *Rayleigh* PDF is narrow banded, while the Dirlik PDF is more appropriate when the stress spectra are broad banded.

(b) The check on whether to use the Rayleigh or Dirlik's PDF is applied on a spectrum by spectrum basis. The bandwidth of each response spectrum (that is, of the spectrum from each analysis you specify) is individually computed, and then the check is applied. The chosen PDF is then used in calculating the contribution to the response histogram from that spectrum. Naturally if you nominate either *Rayleigh* or *Dirlik*, then that PDF is used with all response spectra.

(c) The value of the bandwidth for each individual response spectrum is echoed by Histogram to the output file, where the program also indicates which pdf is used based on the threshold you specify. This allows you to monitor or verify the program operation.

1.8.5.4 *SEASTATE FILES

PURPOSE

To specify all of the Flexcom random sea analyses on which the Histogram calculations are to be based, and to tabulate the percentage occurrences of the corresponding seastates.

THEORY

Refer to Histogram Overview for further information on this feature.

KEYWORD FORMAT

Blocks of two lines, defining the file name and the percentage occurrence.

```
FILE=File Name
PERCENT=Percentage Occurrence
```

`File Name` should include the entire path of the included file without extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks. `Percentage Occurrence` should naturally be greater than 0 and less than or equal to 100.

TABLE INPUT

Input:	Description
File Name:	The generic name of the Flexcom analysis for a particular fatigue seastate. A file extension is not required.
Percentage Occurrence:	The percentage occurrence of this seastate per annum.

NOTES

(a) The total percentage occurrence should sum to 100%.

1.8.6 $LIFEFREQUENCY

This section corresponds to LifeFrequency, which is a frequency domain fatigue postprocessor to Flexcom. Refer to Frequency Domain Fatigue Analysis for further details.

This section contains the following keywords:

- *BLOCK is used to input the fatigue analysis wave scatter diagram, including the definition of blocks and/or reference seastates as appropriate.
- *DIRECTION is used to specify long-term directionality data.
- *FATIGUE DATA is used to assign fatigue data to hot spot sets.
- *HOT SPOT SETS is used to define the fatigue analysis hot spots - these are the locations on the structure for which fatigue life estimates are required.
- *NAME is used to specify a title for a LifeFrequency analysis run.
- *PDF is used to specify the probability density function to be used in calculating fatigue life estimates from stress spectra.
- *PROPERTIES is used to assign effective structural properties to hot spot sets for use in calculating stresses.
- *SEASTATE FILES is used to specify all of the Flexcom random sea analyses on which the LifeFrequency calculations are to be based, and to tabulate the percentage occurrences of the corresponding seastates.
- *S-N CURVE is used to define fatigue analysis S-N curves.
- *SPECTRUM is used to specify the wave spectrum type to use for fatigue life calculations.

1.8.6.1 *BLOCK

PURPOSE

To input the fatigue analysis wave scatter diagram, including the definition of blocks and/or reference seastates as appropriate.

THEORY

Refer to Long Term Environmental Conditions for further information on this feature.

KEYWORD FORMAT

The way in which this keyword is included in the seastate file depends on how you want to specify the seastate information, as follows:

For one seastate, the format of the keyword in the file is as follows, with line 3 repeated for as many seastate values as you want to specify:

```
*BLOCK
REFERENCE=Hs, Tz/Tp, Number of Occurrences, DIRECTION=Direction, FILE=File Na
Hs, Tz/Tp, Number of Occurrences
```

For selected seastates, the REFERENCE= line is repeated for each seastate block that you want to define, with line 3 repeated for as many seastate values as you want to include within that seastate block:

```
*BLOCK
REFERENCE=Hs, Tz/Tp, Number of Occurrences, DIRECTION=Direction, FILE=File Na
Hs, Tz/Tp, Number of Occurrences
```

For all seastates, the REFERENCE= line is repeated for every seastate value:

```
*BLOCK
REFERENCE=Hs, Tz/Tp, Number of Occurrences, DIRECTION=Direction, FILE=File Na
```

This keyword is required for the *Postprocessor* with RAOs mode, but is not relevant to *Postprocessor with Stress Spectra* mode. `Direction` can be `N, NW, W, SW, S, SE, E` or `NE`.

SEASTATE SCATTER DIAGRAM

Purpose

To input the fatigue analysis wave scatter diagram, including the definition of blocks and/or reference seastates as appropriate.

Table Input

Input:	Description
Hs:	The seastate significant wave height Hs.
Tz/Tp:	The seastate mean zero up-crossing period Tz or peak period Tp.
No. of Occurrences:	The number of occurrences in a given period (typically one year) of the particular combination of H_s and T_z/T_p values which that cell represents.

Notes

(a) The scatter diagram is defined in terms of significant wave height (H_s) and mean zero up-crossing period (T_z), or wave spectrum peak period (T_p). If there is no occurrence of a particular combination of H_s and T_z/T_p, you simply leave the corresponding cell blank.

(b) The duration of the period measuring the number of occurrences is in fact immaterial - LifeFrequency transforms the data you specify into percentage occurrence values, so it is the relative magnitudes only of the entries that are of importance.

(c) The *Mark Seastate Blocks* option is used to group seastates into blocks. This task can be performed at any stage during the input of the scatter diagram data, but would typically be performed after all of the H_s/T_z or H_s/T_p data has been input. You may also delete blocks subsequently, allowing you to change the way in which seastates are grouped in blocks without inputting the scatter diagram again in full.

(d) The *Mark Reference Seastates* option is used to nominate reference seastates within each block. This task can only be performed after one or more seastate blocks have been defined. If you click on a seastate in a block where a reference seastate is already nominated, the previous nomination becomes deselected.

PRE-RUN ANALYSES

Purpose

To specify a list of Pre-Run Analyses, consisting of the Flexcom analyses corresponding to the scatter diagram reference seastates.

Table Input

Input:	Description
Hs:	The reference seastate significant wave height Hs.
Tz/Tp:	The reference seastate mean zero up-crossing period T_z or peak period T_p. Depending on the option selected in the *Seastate Scatter Diagram* table, the time values specified here can relate to either T_z or T_p.
Direction:	One of eight compass directions.
Flexcom Analysis:	The name of the file containing axial force and bending moment RAOs for this particular combination of environmental conditions.

Notes

(a) You must specify an RAO file name in the *Pre-Run Analyses* table for each reference seastate for each direction with non-zero percentage occurrence.

(b) You do not need to specify a file type in the *Flexcom Analysis* input, just the analysis root or generic name.

1.8.6.2 *DIRECTION

PURPOSE

To specify long-term directionality data.

THEORY

Refer to Long Term Environmental Conditions for further information on this feature.

KEYWORD FORMAT

One line specifying the directionality data.

`Direction=Percentage Annual Occurrence`

This keyword is required for the *Postprocessor* with RAOs mode, but is not relevant to *Postprocessor with Stress Spectra* mode. `Direction` can be `N, NW, W, SW, S, SE, E` or `NE`.

TABLE INPUT

Input:	Description
Direction:	One of eight compass directions.
Percentage Occurrence:	The percentage of 1 year during which storms occur from this direction.

NOTES

(a) “North” is defined as being the direction of the positive sense of the global Y axis. ”West” is then in the direction of the positive sense of the global Z axis, ”South” in the negative sense of global Y, and ”East” in the negative sense of global Z. The other directions are naturally intermediate to these.

(b) Waves in the ”North” direction are defined as travelling in the positive global Y direction, that is towards rather than from the compass point, and likewise for the other directions.

(c) The total of all of the percentage occurrence values must equal 100%.

1.8.6.3 *FATIGUE DATA

PURPOSE

To assign fatigue data to hot spot sets.

THEORY

Refer to Fatigue Data for further information on this feature.

KEYWORD FORMAT

A block of two lines to specify the fatigue data for each hot spot set. The first line defines the name of the hot spot set to which the fatigue data will be assigned. The second line specifies the fatigue data for the hot spots in the set in the following format.

```
SET=Hot Spot Set Name
CURVE=S-N Curve Name [, SCF=SCF Value] [, STRESS=Stress
Type] [, TB=Threshold Thickness] [, EXPONENT=Exponent Value]
[, ESTIMATE=Estimate Option]
```

`Hot Spot Set Name` must be defined using `*HOT SPOT SETS`. `S-N Curve Name` must be defined using `*S-N CURVE`. `SCF Value` defaults to 1.0. `Stress Type` can be COMBINED (the default), BENDING or AXIAL. If a `Threshold Thickness` is not specified, thickness effects are not included in the fatigue-life calculations. The specification of an `Exponent Value` is meaningful only if thickness effects are included. `Exponent Value` defaults to 4. `Estimate Option` can be `ALL`. If omitted, one fatigue life estimate (the minimum) is included in the output file per hot spot.

TABLE INPUT

Input:	Description
Set Name:	The hot spot set to which the fatigue properties are to be assigned.
S-N Curve:	The name of a defined S-N curve.
SCF:	The stress concentration factor (SCF) to be used in fatigue calculations.
Stress Type:	This option allows you to nominate the type of stress to be used in the fatigue calculations. You can choose between bending stresses only, axial stresses only or combined bending and axial stresses, which is the default.
Threshold Thickness:	The threshold thickness for the inclusion of thickness effects. See Notes (a) and (b).
Exponent:	The exponent value *n* used in the calculation of the stress multiplication factor when threshold thickness effects are included. This defaults to 4. See Notes (a) and (b).

Output:	An option to nominate how many fatigue life estimates are to included in the LifeFrequency output file per hot spot. LifeFrequency calculates fatigue life estimates at eight points around the outer circumference (these points are known as “stress points”). This option allows you to choose how many of these are echoed to the program output file. The default is *Minimum*, which means one value only, the minimum value, is output. The alternative is *All*, which means all eight values are output.

NOTES

(a) The specification of a threshold thickness allows you to take account of the fact that the fatigue strength of some structural members can be dependent on material thickness, fatigue strength decreasing with increasing thickness. If you specify a threshold thickness, the stresses calculated by LifeTime are further multiplied by a factor f given by:

$$f=\left(\frac{t}{t_b}\right)^{\frac{1}{n}}$$

where t_b is the specified threshold thickness; and t is the greater of t_b and the actual thickness of the particular location under consideration (this ensures that f is always greater than or equal to 1). Note that although a single value of t_b is input, f is naturally computed individually for each hot spot, since the structure thickness may vary from location to location. The exponent value n can also be specified and defaults to 4.

(b) The specification of threshold thickness is optional. By default, thickness effects are ignored unless you explicitly specify a threshold thickness.

1.8.6.4 *HOT SPOT SETS

PURPOSE

To define the fatigue analysis hot spots - these are the locations on the structure for which fatigue life estimates are required.

THEORY

Refer to Fatigue Data for further information on this feature.

KEYWORD FORMAT

Blocks of lines that define a set repeated as often as necessary. Each block begins with a line defining the hot spot set name. Next comes a line defining a local node number, followed by either of two types of lines defining elements (numbers or labels). A combination of node and element lines completely describe the hot spots in the set. The element lines can be mixed and repeated as often as necessary.

Line to define set name:

`SET=`*`Hot Spot Set Name`*

Line directly defining a local node number

`NODE=`*`Local Node`*

Line defining hot spot(s) using a single element:

`Element (Number or Label)`

Line defining hot spots using a list of elements:

`GEN=`*`Start Element (Number or Label)`*`,` *`End Element (Number or Label)`* `[,` *`Increme`*

`Local Node` may be *1* (First Node), *2* (Midpoint), *3* (Last Node) or *4* (All).

TABLE INPUT

Input:	Description
Set Name:	A unique name for the hot spot set.

Elements:	The numbers or labels of the elements comprising the set. These are input in the standard format used by Flexcom. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Local Node:	This option allows you to choose between three locations on the specified element. *First Node* refers to the first specified node of the element, *Last Node* is the second specified node, and *Midpoint* is half-way between the two. All corresponds to all three locations.

1.8.6.5 *NAME

PURPOSE

To specify a title for a LifeFrequency analysis run.

THEORY

Refer to Frequency Domain Fatigue Analysis for further information on this feature.

KEYWORD FORMAT

A single line containing the analysis name.

```
Name
```

TABLE INPUT

Input:	Description
Title:	A descriptive title to be associated with the LifeFrequency analysis. This entry is optional.

1.8.6.6 *PDF

PURPOSE

To specify the probability density function to be used in calculating fatigue life estimates from stress spectra.

THEORY

Refer to Frequency Domain Fatigue Analysis for further information on this feature.

KEYWORD FORMAT

One line of data specifying the probability density function as follows.

```
PDF=PDF Type
```

PDF Type can be RAYLEIGH or DIRLIK. This keyword is optional. If omitted, *PDF Type* defaults to RAYLEIGH.

TABLE INPUT

Input:	Description
Probability Density Function:	This option allows you to choose the probability density function (pdf) to be used in calculating fatigue life estimates from stress spectra. The standard *Rayleigh* PDF (the default) is narrow banded, while the Dirlik PDF is more appropriate when the stress spectra are broad banded.

1.8.6.7 *PROPERTIES

PURPOSE

To assign effective structural properties to hot spot sets for use in calculating stresses.

THEORY

Refer to Stress Properties for further information on this feature.

KEYWORD FORMAT

A block of two lines repeated as often as necessary. Each property specified on the second line can be omitted, in which case a default value will be used for stress calculations:

```
SET=Set Name
Do, Di, A, Iyy, Izz
```

The properties specified here are used in calculating stresses during fatigue analysis. This keyword is optional and may be ignored if you wish to use the actual properties (e.g. via the `*GEOMETRIC SETS` keyword in `$MODEL`), or properties explicitly specified for stress computations (i.e. via the corresponding `*PROPERTIES` keyword in `$MODEL`), as the values specified in the model definition are carried through to LifeFrequency.

If any of the parameters `A, Iyy` or `Izz` are omitted, default values are computed based on the specified (or default) values of `Do` and `Di`.

TABLE INPUT

Input:	Description
Set Name:	The name of the element set to which the stress properties are to be assigned. This defaults to all elements.
Do:	The effective outside diameter for the elements of the set. The default depends on the format used to specify geometric data. If you used the Flexible Riser format, then the default is the drag diameter for the elements of the set. If you used the *Rigid Riser* or *Mooring Line* formats, then Do here defaults to Do input in the properties data.
Di:	The effective internal diameter for the elements of the set. The default is the internal diameter specified in the geometric data.
A:	The effective cross-sectional area for the elements of the set. The default is given by: $$A = \frac{\pi\left(D_o^2 - D_i^2\right)}{4}$$ where Do and Di are the inputs described above, default or otherwise.

Iyy:	The second moment of area about the local y-axis for the elements of the set. The default is given by: $$I_{yy} = \frac{\pi\left(D_o^4 - D_i^4\right)}{64}$$ where Do and Di are the inputs described above, default or otherwise.
Izz:	The second moment of area about the local z axis for the elements of the set. The default is Iyy above.

NOTES

(a) This table is identical to the Properties – Stress table of the main `$MODEL` section. If you use the table in `$MODEL` to specify stress properties, then these automatically carry through to LifeFrequency and you do not need to repeat the specification here. Refer to Stress Properties for a detailed discussion of the various parameters above.

(b) LifeFrequency reads stress properties from the output files produced by the first Flexcom analysis in your *Pre-Run Analyses* list. So if you specify stress properties in the Flexcom data for that analysis, once again you do not need to repeat them here.

1.8.6.8 *SEASTATE FILES

PURPOSE

To specify all of the Flexcom random sea analyses on which the LifeFrequency calculations are to be based, and to tabulate the percentage occurrences of the corresponding seastates.

THEORY

Refer to Pre-run Analyses for further information on this feature.

KEYWORD FORMAT

Blocks of two lines, defining the file name and the percentage occurrence.

```
FILE=File Name
PERCENT=Percentage Occurrence
```

This keyword is required for the *Postprocessor with Stress Spectra* mode, but is not relevant to *Postprocessor with RAOs* mode. `File Name` should include the entire path of the included file without extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks. `Percentage Occurrence` should naturally be greater than 0 and less than or equal to 100.

TABLE INPUT

Input:	Description
Flexcom Analysis:	The generic name of the Flexcom analysis for a particular fatigue seastate. A file extension is not required.
Percentage Occurrence:	The percentage occurrence of this seastate per annum.

NOTES

(a) The total percentage occurrence should sum to 100%.

1.8.6.9 *S-N CURVE

PURPOSE

To define fatigue analysis S-N curves.

THEORY

Refer to Fatigue Data for further information on this feature.

KEYWORD FORMAT

Blocks of lines that define an S-N curve repeated as often as necessary. Each block begins with a line defining the S-N curve name and type. This is then followed by either of three types of lines to define the S-N curve.

Line to define S-N curve name and type:

```
CURVE=S-N Curve Name, TYPE=S-N Curve Type
```

The format of subsequent lines depends on the type of S-N curve being defined.

Line to define a log-linear (single sloped) curve, with or without an endurance limit:

```
m, K [, Endurance Limit]
```

Lines to define a piecewise log-linear curve (the line is repeated as often as necessary):

```
m, K, N1, N2
```

Lines to define a curve directly in terms of (S, N) data pairs (the line is repeated as often as necessary):

```
S, N
```

S-N Curve Type can be LOG-LINEAR or DATA PAIRS. LOG-LINEAR in this context refers also to piecewise log-linear. `Endurance Limit` defaults to 1, that is, no endurance limit.

S-N CURVE - LINEAR

Purpose

To specify linear S-N curves.

Table Input

Input:	Description
Curve Name:	A unique name for the S-N curve.
m:	The first parameter defining the S-N curve. See Note (a).
K:	The second parameter defining the S-N curve. See Note (a).
Endurance Limit:	A stress range value below which no fatigue damage occurs, regardless of the number of cycles. See Note (b).

Notes

(a) S-N curves are generally defined in the form $NS^m = K$ where S denotes stress range, N the number of cycles to failure at this range, and *m* and *K* are constants. Taking logarithms of both sides and rearranging gives:

$$\log S = \frac{1}{m}\log N + \frac{1}{m}\log K$$

which is the equation of a straight line when log S is plotted against log N. In this case m is the inverse slope and K is a function of the line intercept. These are the parameters input above.

(b) The specification of an *Endurance Limit* is optional, and by default there is no endurance limit.

S-N CURVE - PIECEWISE LINEAR

Purpose

To specify piecewise linear S-N curves.

Table Input

Input:	Description
Curve Name:	A unique name for the S-N curve.
m:	The first parameter defining the S-N curve. See Note (a).
K:	The second parameter defining the S-N curve. See Note (a).
N1:	The number of cycles value defining the lower end of the line segment where these m and K values apply.
N2:	The number of cycles value defining the upper end of the line segment where these m and K values apply.

Notes

(a) A piecewise-linear S-N curve is one which plots as a series of straight line segments when log S is plotted against log N. In this case *m* and *K* vary between line segments; the N1 and N2 values above specify the segment of the S-N curve X (number of cycles to failure) axis where each *m* and K combination apply.

(b) Use as many lines as you need to completely define a particular S-N curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent curves, put the curve name in Column 1 and specify the S-N data in the same way.

(c) For stress ranges which are within the bounds of the specified S-N data, log-linear interpolation to find the number of cycles to failure.

(d) The smallest specified stress range is assumed to correspond to the endurance limit of the material, hence no fatigue is deemed to occur for stress ranges which are lower than the smallest specified stress range, regardless of the number of cycles experienced.

(e) For stress ranges which are higher than the largest specified stress range, the number of cycles to failure is assumed equal to the N value corresponding to the largest specified stress range.

S-N CURVE - DATA PAIRS

Purpose

To specify S-N curves directly in terms of S and N data pairs.

Table Input

Input:	Description
Curve Name:	A unique name for the S-N curve.
S:	A stress range value.
N:	The number of cycles to cause failure at this stress range.

Notes

(a) Use as many lines as you need to completely define a particular S-N curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent curves, put the curve name in Column 1 and specify the S-N data in the same way.

(b) For stress ranges which are within the bounds of the specified S-N data, log-linear interpolation to find the number of cycles to failure.

(c) The smallest specified stress range is assumed to correspond to the endurance limit of the material, hence no fatigue is deemed to occur for stress ranges which are lower than the smallest specified stress range, regardless of the number of cycles experienced.

(d) For stress ranges which are higher than the largest specified stress range, the number of cycles to failure is assumed equal to the N value corresponding to the largest specified stress range.

1.8.6.10 *SPECTRUM

PURPOSE

To specify the wave spectrum type to use for fatigue life calculations.

THEORY

Refer to Wave Loading for further information on this feature.

KEYWORD FORMAT

One line specifying the spectrum type.

```
TYPE=Spectrum Type [, OPTION=Option]
```

This keyword is required for the *Postprocessor with RAOs* mode, but is not relevant to *Postprocessor with Stress Spectra* mode. *Spectrum Type* can be PIERSON-MOSKOWITZ or JONSWAP. *Option* can be `TZ` or `TP`, and defaults to `TZ` if omitted.

TABLE INPUT

Input:	Description
Spectrum Type:	The options are *Pierson-Moskowitz* and *Jonswap.*
Period Type:	The options are T_z (the default) and T_p.

NOTES

(a) The spectrum type is a necessary input when LifeFrequency is using stress RAOs to generate stress spectra from wave spectra for the various individual scatter diagram cells.

(b) When you specify Jonswap spectra, LifeFrequency uses the algorithm described in Jonswap Wave to calculate the Jonswap parameters f_p, α and γ from the H_s and T_z/T_p values in the scatter diagram.

1.8.7 $LIFETIME CYCLE

This section corresponds to LifeTime Mode 2. LifeTime is an ancillary module to Flexcom which performs time domain fatigue analysis (Mode 1) and general cycle counting (Mode 2). Refer to Cycle Counting Analysis (Mode2) for further details.

This section contains the following keywords:

- `*BINS` is used to specify bins or divisions for histogram output.
- `*CHANNELS` is used to specify channels for cycle counting.
- `*HISTOGRAM OUTPUT` is used to specify data relating to histogram output.
- `*NAME` is used to specify a title for a LifeTime analysis run.
- `*SEASTATE FILES` is used to specify all of the Flexcom random sea analyses on which the LifeTime calculations are to be based, and to tabulate the percentage occurrences of the corresponding seastates.

1.8.7.1 *BINS

PURPOSE

To specify bins or divisions for histogram output.

THEORY

Refer to Cycle Counting Analysis (Mode 2) for further information on this feature.

KEYWORD FORMAT

A block of lines beginning with the bin set name. This is then followed by a single line repeated as often as necessary to define the number of bins or divisions of the histogram. The entire block may be then repeated to define subsequent bins.

Line specifying the bin set name:

```
BIN_SET=Set Name
```

Line defining the divisions of the histogram:

```
Value, Value, Value, Value, Value, etc.
```

A histogram is assumed to start at 0, so the first entry on the first line is assumed to be the upper end of the first bin; the second entry is the upper end of the second bin and so on. Up to 20 values may be specified on a single line.

TABLE INPUT

Input:	Description
Bin Set:	A set name for the bin definition. This entry is optional, and defaults to *Default* if omitted.
Divisions for Stress Histogram Output:	This entry allows you to enter values to specify how values are grouped in histogram output.

NOTES

(a) Definition of bins is required for a Mode 2 analysis (but optional for a Mode 1 analysis). Several bin sets are typically required in a Mode 2 analysis, as you may be cycle counting various parameters. The *Bin Set* entry is not relevant to Mode 1 analyses, as the histograms produced are always stress histograms, and would typically all tend to have a similar order of magnitude.

(b) You can specify up to 20 values per line, and you may use as many lines as you need to completely define your histogram. A histogram is assumed to start at 0, so the first value you specify is the upper stress of the first bin of the histogram.

(c) Stress Histograms discusses stress histograms in more detail, and provides insights into the significance and use of some of the other input parameters relating to this type of LifeTime output.

1.8.7.2 *CHANNELS

PURPOSE

To specify channels for cycle counting.

THEORY

Refer to Cycle Counting Analysis (Mode 2) for further information on this feature.

KEYWORD FORMAT

A block of lines beginning with the bin set name. This is then followed by a single line repeated as often as necessary to specify the required channels. Up to 20 values may be specified on a single line. The entire block may be then repeated to define subsequent channels.

Line specifying the bin set name:

```
[SET=Set Name]
```

Line specifying the required channels:

```
Channel Number [, Channel Number] [, Channel Number] etc.
```

TABLE INPUT

Input:	Description
Bin Set:	The bin set definition to be used for the current channel. This entry is optional, and defaults to *Default* if omitted.
Channel Number:	The number of the channel for cycle counting. This must be non-zero, and must be less than or equal to the total number of channels. To count more than one channel, input one channel number per line.

1.8.7.3 *HISTOGRAM OUTPUT

PURPOSE

To specify data relating to histogram output.

THEORY

Refer to Cycle Counting Analysis (Mode 2) for further information on this feature.

KEYWORD FORMAT

One line with three values, two of which are optional:

```
No. of Channels in Timetrace File, [Start Time] [, Scale Factor]
```

`Start Time` defaults to the timetrace output start time. `Scale Factor` defaults to 1.

TABLE INPUT

Input:	Description
Number of Channels:	The number of individual data values output by Flexcom to the timetrace output files at each solution time. See Note (a).
Start Time:	The time (in seconds) at which to start using data from the seastate files. It is used to exclude initial transients from the cycle counting calculations. Any data from solution times prior to this start time is ignored.
Scale Factor:	This value is used to transform data in the Flexcom units to appropriate units for the histogram output. The value specified here is applied to all output channels. This input is optional, and defaults to 1.

NOTES

(a) *The Number of Channels* must be the same for all of the files listed in the Seastate File – Specify table. Likewise, for meaningful cycle counting, the individual channels must be the same in each of the files. Note however that there is no requirement that the simulation lengths are the same in all of the analyses. Nor is it necessary that the individual solution times be the same in all of the Flexcom analyses.

1.8.7.4 *NAME

PURPOSE

To specify a title for a LifeTime analysis run.

THEORY

Refer to Cycle Counting Analysis (Mode 2) for further information on this feature.

KEYWORD FORMAT

A single line containing the analysis name.

```
Name
```

TABLE INPUT

Input:	Description
Title:	A descriptive title to be associated with the LifeTime analysis. This entry is optional.

1.8.7.5 *SEASTATE FILES

PURPOSE

To specify all of the Flexcom random sea analyses on which the LifeTime calculations are to be based, and to tabulate the percentage occurrences of the corresponding seastates.

THEORY

Refer to Cycle Counting Analysis (Mode 2) for further information on this feature.

KEYWORD FORMAT

Blocks of two lines, defining the file name and the percentage occurrence.

```
FILE=File Name
PERCENT=Percentage Occurrence
```

`File Name` should include the entire path of the included file without extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks. `Percentage Occurrence` should naturally be greater than 0 and less than or equal to 100.

TABLE INPUT

Input:	Description
File Name:	The generic name of the Flexcom analysis for a particular fatigue seastate. A file extension is not required.
Percentage Occurrence:	The percentage occurrence of this seastate per annum.

NOTES

(a) The total percentage occurrence should sum to 100%.

1.8.8 $LIFETIME FATIGUE

This section corresponds to LifeTime Mode 1. LifeTime is an ancillary module to Flexcom which performs time domain fatigue analysis (Mode 1) and general cycle counting (Mode 2). Refer to Fatigue Analysis (Mode 1) for further details.

This section contains the following keywords:

- `*BINS` is used to specify bins or divisions for histogram output.
- `*FATIGUE DATA` is used to assign fatigue data to hot spot sets.
- `*HISTOGRAM DATA` is used to specify histogram options for hot spots.
- `*HOT SPOT SETS` is used to define the fatigue analysis hot spots - these are the locations on the structure for which fatigue life estimates are required
- `*NAME` is used to specify a title for a LifeTime analysis run.

- `*PDF` is used to specify the probability density function to be used in calculating fatigue life estimates from stress spectra.
- `*PROPERTIES` is used to assign effective structural properties to hot spot sets for use in calculating stresses.
- `*SEASTATE FILES` is used to specify the names, and corresponding percentage occurrences, of the Flexcom simulations on which the LifeTime calculations are to be based.
- `*S-N CURVE` is used to define fatigue analysis S-N curves.
- `*SOURCE TYPE` is used to indicate the type of data storage file you wish to use as input to the fatigue analysis.
- `*TD OPTIONS` is used to specify a number of miscellaneous parameters, many of which relate to the calculation of stress spectra.

1.8.8.1 *BINS

PURPOSE

To specify bins or divisions for histogram output.

THEORY

Refer to Fatigue Analysis (Mode 1) for further information on this feature.

KEYWORD FORMAT

A single line repeated as often as necessary to define the number of bins or divisions of the histogram. A histogram is assumed to start at 0, so the first entry on the first line is assumed to be the upper end of the first bin; the second entry is the upper end of the second bin and so on. Up to 20 values may be specified on a single line.

```
Value, Value, Value, Value, Value, etc.
```

This input is optional. If it is omitted the program uses a default bins specification depending on the unit system.

TABLE INPUT

Input:	Description
Divisions for Stress Histogram Output:	This entry allows you to enter values to specify how values are grouped in histogram output.

NOTES

(a) Definition of bins is optional for a Mode 1 analysis (but required for a Mode 2 analysis). Several bin sets are typically required in a Mode 2 analysis, as you may be cycle counting various parameters. The *Bin Set* entry is not relevant to Mode 1 analyses, as the histograms produced are always stress histograms, and would typically all tend to have a similar order of magnitude.

(b) You can specify up to 20 values per line, and you may use as many lines as you need to completely define your histogram. A histogram is assumed to start at 0, so the first value you specify is the upper stress of the first bin of the histogram.

(c) Stress Histograms discusses stress histograms in more detail, and provides insights into the significance and use of some of the other input parameters relating to this type of LifeTime output.

1.8.8.2 *FATIGUE DATA

PURPOSE

To assign fatigue data to hot spot sets.

THEORY

Refer to Fatigue Analysis (Mode 1) for further information on this feature.

KEYWORD FORMAT

A block of two lines to specify the fatigue data for each hot spot set. The first line defines the name of the hot spot set to which the fatigue data will be assigned. The second line specifies the fatigue data for the hot spots in the set in the following format.

```
SET=Hot Spot Set Name
CURVE=S-N Curve Name [, SCF=SCF Value] [, STRESS=Stress
Type] [, TB=Threshold Thickness] [, EXPONENT=Exponent Value]
```

`Hot Spot Set Name` must be defined using `*HOT SPOT SETS`. `S-N Curve Name` must be defined using `*S-N CURVE`. `SCF Value` defaults to 1.0. `Stress Type` can be COMBINED (the default), BENDING or AXIAL. If a `Threshold Thickness` is not specified, thickness effects are not included in the fatigue-life calculations. The specification of an Exponent Value is meaningful only if thickness effects are included. `Exponent Value` defaults to 4.

TABLE INPUT

Input:	Description
Set Name:	The hot spot set to which the fatigue properties are to be assigned.
S-N Curve:	The name of a defined S-N curve.
SCF:	The stress concentration factor (SCF) to be used in fatigue calculations.
Stress Type:	This option allows you to nominate the type of stress to be used in the fatigue calculations. You can choose between bending stresses only, axial stresses only or combined bending and axial stresses, which is the default.
Threshold Thickness:	The threshold thickness for the inclusion of thickness effects. See Notes (a) and (b).
Exponent:	The exponent value *n* used in the calculation of the stress multiplication factor when threshold thickness effects are included. This defaults to 4. See Notes (a) and (b).

NOTES

(a) The specification of a threshold thickness allows you to take account of the fact that the fatigue strength of some structural members can be dependent on material thickness, fatigue strength decreasing with increasing thickness. If you specify a threshold thickness, the stresses calculated by LifeTime are further multiplied by a factor f given by:

$$f=\left(\frac{t}{t_b}\right)^{\frac{1}{n}}$$

where t_b is the specified threshold thickness; and t is the greater of t_b and the actual thickness of the particular location under consideration (this ensures that is always greater than or equal to 1). Note that although a single value of t_b is input, f is naturally computed individually for each hot spot, since the structure thickness may vary from location to location. The exponent value n can also be specified and defaults to 4.

(b) The specification of threshold thickness is optional. By default, thickness effects are ignored unless you explicitly specify a threshold thickness.

1.8.8.3 *HISTOGRAM DATA

PURPOSE

To specify histogram options for hot spots.

THEORY

Refer to Fatigue Analysis (Mode 1) for further information on this feature.

KEYWORD FORMAT

A single line specifying the equivalent range index, followed by one or more lines requesting the creation of stress histograms.

Optional line specifying the index:

```
[RANGE=Equivalent Range Index]
```

Line requesting the creation of a stress histogram:

`SET=Hot Spot Set Same [, LOCATION=Stress Point]`

This keyword is optional. `Equivalent Range Index` defaults to 1 if not specified. If omitted, a stress histogram is produced by default for the stress point at which the lowest fatigue life occurs.

TABLE INPUT

Input:	Description
Set Name:	The hot spot set for which you are defining histogram properties.
Stress Point:	This is an optional input to specify the point on the circumference (“stress point”) for which a stress histogram is to be produced. This input is invalid unless you selected *Create Stress Histogram* in the previous input. If no value is input here, a histogram is produced for the stress point at which the predicted fatigue life is a minimum. Input a value between 1 and 8 to nominate a stress point. This corresponds to an angle measured in degrees anti-clockwise from the local element cross-section y-axis where 1 = 0°, 2 = 45°, 3 = 90°, etc.

1.8.8.4 *HOT SPOT SETS

PURPOSE

To define the fatigue analysis hot spots - these are the locations on the structure for which fatigue life estimates are required.

THEORY

Refer to Fatigue Analysis (Mode 1) for further information on this feature.

KEYWORD FORMAT

Blocks of lines that define a set repeated as often as necessary. Each block begins with a line defining the hot spot set name. Next comes a line defining a local node number, followed by either of two types of lines defining elements (numbers or labels). A combination of node and element lines completely describe the hot spots in the set. The element lines can be mixed and repeated as often as necessary.

Line to define set name:

```
SET=Hot Spot Set Name
```

Line directly defining a local node number

```
NODE=Local Node
```

Line defining hot spot(s) using a single element:

```
Element (Number or Label)
```

Line defining hot spots using a list of elements:

```
GEN=Start Element (Number or Label), End Element (Number or Label) [, Increme
Local Node may be 1 (First Node), 2 (Midpoint), 3 (Last Node) or 4 (All).
```

TABLE INPUT

Input:	Description
Set Name:	A unique name for the hot spot set.
Elements:	The numbers or labels of the elements comprising the set. These are input in the standard format used by Flexcom. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Local Node:	This option allows you to choose between three locations on the specified element. *First Node* refers to the first specified node of the element, *Last Node* is the second specified node, and *Midpoint* is half-way between the two. All corresponds to all three locations.

1.8.8.5 *NAME

PURPOSE

To specify a title for a LifeTime analysis run.

THEORY

Refer to Fatigue Analysis (Mode 1) for further information on this feature.

KEYWORD FORMAT

A single line containing the analysis name.

```
Name
```

TABLE INPUT

Input:	Description
Title:	A descriptive title to be associated with the LifeTime analysis. This entry is optional.

1.8.8.6 *PDF

PURPOSE

To specify the probability density function to be used in calculating fatigue life estimates from stress spectra.

THEORY

Refer to Fatigue Analysis (Mode 1) for further information on this feature.

KEYWORD FORMAT

One line of data specifying the probability density function as follows.

```
PDF=PDF Type
```

PDF Type can be RAYLEIGH or DIRLIK. This keyword is optional. If omitted, PDF Type defaults to RAYLEIGH.

TABLE INPUT

Input:	Description

Probability Density Function:	This option allows you to choose the probability density function (pdf) to be used in calculating fatigue life estimates from stress spectra. The standard *Rayleigh* PDF (the default) is narrow banded, while the *Dirlik* PDF is more appropriate when the stress spectra are broad banded.

1.8.8.7 *PROPERTIES

PURPOSE

To assign effective structural properties to hot spot sets for use in calculating stresses.

THEORY

Refer to Stress Properties for further information on this feature.

KEYWORD FORMAT

A block of two lines repeated as often as necessary. Each property specified on the second line can be omitted, in which case a default value will be used for stress calculations:

```
SET=Set Name
Do, Di, A, Iyy, Izz
```

The properties specified here are used in calculating stresses during fatigue analysis. This keyword is optional and may be ignored if you wish to use the actual properties (e.g. via the `*GEOMETRIC SETS` keyword in the `$MODEL` section), or properties explicitly specified for stress computations (i.e. via the corresponding `*PROPERTIES` keyword in the `$MODEL` section), as the values specified in the model definition are carried through to LifeTime (assuming you have requested database output storage).

If any of the parameters `A, Iyy` or `Izz` are omitted, default values are computed based on the specified (or default) values of `Do` and `Di`.

TABLE INPUT

Input:	Description

Set Name:	The name of the element set to which the stress properties are to be assigned. This defaults to all elements.
Do:	The effective outside diameter for the elements of the set. The default depends on the format used to specify geometric data. If you used the *Flexible Riser* format, then the default is the drag diameter for the elements of the set. If you used the *Rigid Riser* or *Mooring Line* formats, then Do here defaults to Do input in the properties data.
Di:	The effective internal diameter for the elements of the set. The default is the internal diameter specified in the geometric data.
A:	The effective cross-sectional area for the elements of the set. The default is given by: $$A = \frac{\pi\left(D_o^2 - D_i^2\right)}{4}$$ where Do and Di are the inputs described above, default or otherwise.
Iyy:	The second moment of area about the local y-axis for the elements of the set. The default is given by: $$I_{yy} = \frac{\pi\left(D_o^4 - D_i^4\right)}{64}$$ where Do and Di are the inputs described above, default or otherwise.
Izz:	The second moment of area about the local z axis for the elements of the set. The default is Iyy above.

NOTES

(a) This table is identical to the Properties – Stress table of the main `$MODEL` section. If you use the table in `$MODEL` to specify stress properties, then these automatically carry through to LifeTime (assuming you have requested database output storage) and you do not need to repeat the specification here. Refer to Stress Properties for a detailed discussion of the various parameters above.

(b) It may seem curious that in general you need to specify diameter, cross-section area and moment of inertia values for hot spots although these values have already been input to Flexcom. The reason is because Flexcom does not echo this data to timetrace output files, and by default only Flexcom timetrace output is required as input to LifeTime. Flexcom does however output the external and internal diameter for each element to database output files. So if Flexcom database output is available, then LifeTime can read the required structural data from that source, thus eliminating the need for a repeat specification of this data. LifeTime reads the name of each Flexcom analysis from the *Seastate File – Specify* data and opens the appropriate timetrace output files for that analysis. The program also checks for the existence of a database output file from the analysis. If a database exists, the program retrieves the external and internal diameter values for each hot spot from that file, and then uses these values to calculate cross-section areas and moments of inertia.

(c) There are a number of important points to note with regard to this facility:

- Only one database needs to exist, and it can correspond to any analysis in your list.
- LifeTime reads database output once only, regardless of how many database files exist. Once it has retrieved the required structural properties from a database file, the program does not check for the existence of database output from any subsequent analysis in the seastate file.
- The database file need not contain very much actual results output. The inputs required by LifeTime are written by default to a header block at the start of the database file at the time of the first output to the database. So in fact database output at only one time is sufficient, and the actual amount of output can be minimised accordingly.
- If database output exists but you specify structural properties for some or all hot spot sets, the values you input in LifeTime take precedence over the values in the database.

1.8.8.8 *SEASTATE FILES

PURPOSE

To specify the names, and corresponding percentage occurrences, of the Flexcom simulations on which the LifeTime calculations are to be based.

THEORY

Refer to Fatigue Analysis (Mode 1) for further information on this feature.

KEYWORD FORMAT

Blocks of two lines, defining the file name and the percentage occurrence.

```
FILE=File Name
PERCENT=Percentage Occurrence, [FREQUENCY=Modal frequency]
```

File Name should include the entire path of the included file without extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks. *Percentage Occurrence* should naturally be greater than 0 and less than or equal to 100. *Modal frequency* pertains specifically to VIV Induced Fatigue of Pipe-in-Pipe Systems.

TABLE INPUT

Input:	Description
File Name:	The name of the Flexcom analysis for a particular loading condition. A file extension is not required.
Percentage Occurrence:	The corresponding percentage occurrence, relative to other loading conditions, per annum.
Modal Frequency:	This input pertains specifically to VIV Induced Fatigue of Pipe-in-Pipe Systems. See Note (b).

NOTES

(a) The total percentage occurrence should sum to 100%.

(b) For the highly specialised case of VIV induced fatigue of inner pipes in pipe-in-pipe systems, the computational procedure involves the construction of regular/periodic time histories of bending moment, derived from the results of a static analysis of a riser system deformed into a specific mode shape, which are then post-processed by LifeTime. In this case, you should select PIP VIV. Refer to VIV Induced Fatigue of Pipe-in-Pipe Systems for further details

1.8.8.9 *S-N CURVE

PURPOSE

To define fatigue analysis S-N curves.

THEORY

Refer to Fatigue Analysis (Mode 1) for further information on this feature.

KEYWORD FORMAT

Blocks of lines that define an S-N curve repeated as often as necessary. Each block begins with a line defining the S-N curve name and type. This is then followed by either of three types of lines to define the S-N curve.

Line to define S-N curve name and type:

```
CURVE=S-N Curve Name, TYPE=S-N Curve Type
```

The format of subsequent lines depends on the type of S-N curve being defined.

Line to define a log-linear (single sloped) curve, with or without an endurance limit:

```
m, K [, Endurance Limit]
```

Lines to define a piecewise log-linear curve (the line is repeated as often as necessary):

```
m, K, N1, N2
```

Lines to define a curve directly in terms of (S, N) data pairs (the line is repeated as often as necessary):

```
S, N
```

S-N Curve Type can be LOG-LINEAR or DATA PAIRS. LOG-LINEAR in this context refers also to piecewise log-linear. *Endurance Limit* defaults to 1, that is, no endurance limit.

S-N CURVE - LINEAR

Purpose

To specify linear S-N curves.

Table Input

Input:	Description
Curve Name:	A unique name for the S-N curve.
m:	The first parameter defining the S-N curve. See Note (a).
K:	The second parameter defining the S-N curve. See Note (a).
Endurance Limit:	A stress range value below which no fatigue damage occurs, regardless of the number of cycles. See Note (b).

Notes

(a) S-N curves are generally defined in the form $NS^m = K$ where S denotes stress range, N the number of cycles to failure at this range, and *m* and K are constants. Taking logarithms of both sides and rearranging gives:

$$\log S = -\frac{1}{m}\log N + \frac{1}{m}\log K$$

which is the equation of a straight line when log S is plotted against log N. In this case m is the inverse slope and K is a function of the line intercept. These are the parameters input above.

(b) The specification of an Endurance Limit is optional, and by default there is no endurance limit.

S-N CURVE - PIECEWISE LINEAR

Purpose

To specify piecewise linear S-N curves.

Table Input

Input:	Description
Curve Name:	A unique name for the S-N curve.
m:	The first parameter defining the S-N curve. See Note (a).
K:	The second parameter defining the S-N curve. See Note (a).
N1:	The number of cycles value defining the lower end of the line segment where these m and K values apply.
N2:	The number of cycles value defining the upper end of the line segment where these m and K values apply.

Notes

(a) A piecewise-linear S-N curve is one which plots as a series of straight line segments when log S is plotted against log N. In this case m and K vary between line segments; the N1 and N2 values above specify the segment of the S-N curve X (number of cycles to failure) axis where each m and K combination apply.

(b) Use as many lines as you need to completely define a particular S-N curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent curves, put the curve name in Column 1 and specify the S-N data in the same way.

(c) For stress ranges which are within the bounds of the specified S-N data, log-linear interpolation to find the number of cycles to failure.

(d) The smallest specified stress range is assumed to correspond to the endurance limit of the material, hence no fatigue is deemed to occur for stress ranges which are lower than the smallest specified stress range, regardless of the number of cycles experienced.

(e) For stress ranges which are higher than the largest specified stress range, the number of cycles to failure is assumed equal to the N value corresponding to the largest specified stress range.

S-N CURVE - DATA PAIRS

Purpose

To specify S-N curves directly in terms of S and N data pairs.

Table Input

Input:	Description
Curve Name:	A unique name for the S-N curve.
S:	A stress range value.
N:	The number of cycles to cause failure at this stress range.

Notes

(a) Use as many lines as you need to completely define a particular S-N curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent curves, put the curve name in Column 1 and specify the S-N data in the same way.

(b) For stress ranges which are within the bounds of the specified S-N data, log-linear interpolation to find the number of cycles to failure.

(c) The smallest specified stress range is assumed to correspond to the endurance limit of the material, hence no fatigue is deemed to occur for stress ranges which are lower than the smallest specified stress range, regardless of the number of cycles experienced.

(d) For stress ranges which are higher than the largest specified stress range, the number of cycles to failure is assumed equal to the N value corresponding to the largest specified stress range.

1.8.8.10 *SOURCE TYPE

PURPOSE

To indicate the type of data storage file you wish to use as input to the fatigue analysis.

THEORY

Refer to Fatigue Analysis (Mode 1) for further information on this feature.

KEYWORD FORMAT

One line of data specifying the source type.

```
SOURCE=Source Type
```

`Source Type` can be TIMETRACE or DATABASE. This keyword is optional. If omitted, `Source Type` defaults to TIMETRACE.

TABLE INPUT

Input:	Description
Source Type:	The options are *Timetrace* (the default) or *Database.* See notes for further details on each of these options.

NOTES

(a) Fatigue analysis in Flexcom is typically based on a series of random sea dynamic analyses, representing the loading experienced by an offshore structure over the course of its lifetime. In this case, Flexcom reads time histories of axial force and bending moment from Timetrace or Database output files. Timetrace output remains the default storage medium as earlier versions of Flexcom specifically required this format. Nowadays you can store this data using the more widely used database output.

(b) For the highly specialised case of VIV induced fatigue of inner pipes in pipe-in-pipe systems, the computational procedure involves the construction of regular/periodic time histories of bending moment, derived from the results of a static analysis of a riser system deformed into a specific mode shape, which are then post-processed by LifeTime. In this case, you should select *Timetrace*. Refer to VIV Induced Fatigue of Pipe-in-Pipe Systems for further details.

(c) Regardless of which source type you are using, the names of the actual data files which are to be used as input to the fatigue simulation are defined via the `*SEASTATE FILES` keyword.

1.8.8.11 *TD OPTIONS

PURPOSE

To specify a number of miscellaneous parameters, many of which relate to the calculation of stress spectra.

THEORY

Refer to Fatigue Analysis (Mode 1) for further information on this feature.

KEYWORD FORMAT

One line with three values, one of which is optional:

```
Start Time, ENSEMBLES=No. of Ensembles [, SCALE=Scale Factor]
```

No. of Ensembles must be greater than 1. *Scale Factor* defaults to 1.

TABLE INPUT

Input:	Description
Scale Factor for Stress:	This is used to transform stresses in the Flexcom units to units consistent with the S-N curve data. For Imperial units, a value of 6.9444E-06 would be typical to transform lb/ft2 to ksi. For SI units, a value of 1.E-06 would be typical to transform N/m2 to MPa. In some cases LifeTime is in a position to decide the appropriate value to use, in which case it is not necessary to explicitly specify a scale factor. See Note (a).
Start Time:	This input is used to exclude initial transients from the fatigue calculations. Any data from solution times prior to this start time is ignored in the LifeTime calculations.
Number of Ensembles:	The number of ensembles to be used in calculating stress spectra. This number must be greater than 1, and in fact defaults to 4. See Note (b).

NOTES

(a) Flexcom outputs the value you specify for g, the gravitational constant, to every analysis database. If any of the Flexcom analyses you list in your *Seastate File – Specify* data generated a database file, then LifeTime retrieves this value and uses it to determine if you used Imperial or SI units in your data. Specifically, if 9 <= g <= 10, then LifeTime decides you employed SI units; if 32 <= g <= 33, then LifeTime decides you used Imperial units. In either case, LifeTime automatically determines the scale factor required to transform stresses to MPa or ksi as appropriate, so you do not need to explicitly specify a *Scale Factor for Stress* - just let LifeTime determine the appropriate scale. But if none of your Flexcom analyses generated a database, you should explicitly specify a scale factor – otherwise no factor will be applied (or in other words, a factor of 1.0 will be applied) – and this may not be consistent with your S-N curve specification.

(b) The procedure used by Flexcom in calculating spectra is as follows. Firstly, the output timetrace is divided equally into a number of smaller timetraces or ensembles. A spectrum for each ensemble is then calculated using the Fast Fourier Transform (FFT) algorithm. Finally, the actual spectrum to be output is found as an average of the spectra calculated for each ensemble. This standard procedure minimises the statistical error associated with the FFT process. You specify the number of ensembles to be used in this process using the *Number of Ensembles* entry above. This value should always be greater than 1.

1.8.9 $LOAD CASE

This section includes data such as environmental parameters (e.g. current and waves), boundary conditions of various kinds, internal fluid loading, and the analysis type and solution parameters.

This section contains the following keywords:

- `*AERODYN DRIVER` is used to provide a link between the structural (Flexcom) and aerodynamic (AeroDyn) models in a wind turbine simulation
- `*ANALYSIS TYPE` is used to specify the analysis type.
- `*BOUNDARY` is used to define boundary conditions.
- `*CALM LOAD` is used to specify data defining the force terms to be applied to a CALM buoy.

- *CLASHING SOLUTION is used to specify solution parameters associated with clashing.
- *CONTACT MODELLING is used to specify specialised parameters relating to guide surface contact modelling.
- *CRITERIA is used to specify certain criteria that need to be satisfied in a static analysis, and to define how the model is to be adjusted to satisfy the desired criteria.
- *CURRENT is used to specify current loading.
- *CURRENT COEFF is used to specify current coefficients used to determine the current loading on a floating body or moored vessel.
- *DAMPING is used to incorporate damping into a dynamic analysis.
- *DAMPING FORMULATION is used to specify the damping formulation to be used in a time domain dynamic analysis.
- *DAMPING RATIO is used to specify stiffness damping coefficients as a function of a damping ratio and a damping period.
- *DATABASE is used to specify the frequency of database output.
- *DATABASE CONTENT is used to customise the contents of the database output files.
- *DRIFT is used to define vessel drift motions.
- *FD ANIMATION is used to define playback parameters for a representative (time domain) structural animation which is fabricated from a frequency domain solution.
- *FORCE RAO is used to specify force RAOs for a floating body.
- *FRICTION is used to specify seabed friction stiffnesses.
- *INFLOWWIND is used to provide a link between Flexcom and FAST's wind-inflow data processing module InflowWind in a wind turbine simulation.
- *INTEGRATION is used to specify the number of integration points to be used in the Gaussian quadrature of element mass and stiffness matrices and load vectors.
- *INTERNAL FLUID is used to define the properties of an internal fluid.

- *LOAD is used to define arbitrary loading.
- *MOMENTS is used to specify the value of Molin's yaw coefficient for a moored vessel, and also to specify the fractions of Molin's Moment and Munk's Moment that are applied to the moored vessel.
- *NAME is used to specify a title for the Flexcom analysis run.
- *NO FINAL STATIC is used to suppress the final static analysis step that is typically performed automatically by Flexcom following a frequency domain dynamic analysis.
- *NO FRICTION is used to suppress the effects of friction.
- *NO HYSTERESIS is used to suppress bending hysteresis effects.
- *NO PIP SLIDING is used to disable the interchangeable nature of sliding pipe-in-pipe connections.
- *NONLINEAR MODEL is used to specify a modelling approach for non-linear materials.
- *NONLINEAR STATIC is used to specify solution parameters used in the final static analysis performed automatically by Flexcom following a frequency domain dynamic analysis.
- *OFFSET is used to specify an offset of an attached vessel from its initial position.
- *PRINT is used to request additional printed output to the main output file.
- *QTF is used to specify Quadratic Transfer Functions (QTFs) that allow the slow drift loads on a floating body or moored vessel to be determined.
- *QTF CALIBRATION FB is used to specify calibration coefficients used to scale the QTF coefficients for a floating body.
- *RAMP is used to specify a linear or non-linear variation in the ramp value applied to loads and displacements.
- *RAO is used to specify Response Amplitude Operators for a vessel.
- *RAO,LOAD is used to specify RAO load data for a frequency domain analysis.

- *REGULAR WAVE EQUIVALENT is used to specify that Flexcom is to replace a random wave spectrum or spectra by an equivalent regular wave or waves, and to calculate equivalent sinusoidal boundary conditions for attached nodes.
- *RESTART is used to indicate that an analysis is to be restarted from a previous run.
- *SERVODYN is used to provide a link between Flexcom and FAST's wind turbine control module ServoDyn in a wind turbine simulation.
- *SLUGS is used to specify parameters relating to slug flow.
- *TEMPERATURE is used to apply thermal loading.
- *THRUSTER is used to specify thruster loads on a moored vessel.
- *TIME is used to define time parameters for an analysis.
- *TIME STEPPING is used to select the time stepping algorithm and to define associated numerical damping coefficients.
- *TIMETRACE is used to request the storage of results for timetrace postprocessing (this is mainly used in the area of time domain fatigue analysis).
- *TOLERANCE is used to define the analysis convergence tolerance measure and related data.
- *USER DEFINED ELEMENT provides advanced Flexcom users with the ability to define custom code for altering element properties.
- *USER SOLVER VARIABLES provides advanced Flexcom users with the ability to define custom code for increased modelling flexibility.
- *UPSTREAM STRUCTURE is used to specify the name of the Flexcom analysis of the upstream structure subjected to the free or undisturbed stream current velocity field.
- *VESSEL TIMETRACE is used to specify that the combined high and low frequency motions of a vessel are to be read from an ASCII timetrace data file.
- *VESSEL VELOCITY is used to specify a vessel constant velocity horizontal velocity.
- *VIV DRAG is used to instruct Flexcom to read vortex-induced vibration (VIV) drag coefficient amplification factors from the results of a Shear7 analysis.

- *VIV EFFECTS is used to instruct Flexcom to continuously run Modes/Shear7 analyses of the downstream structure during the wake interference analysis.
- *WAKE DOWNSTREAM is used to specify the composition of the downstream structure in terms of element sets for use in wake interference calculations. The keyword also facilitates the (optional) definition of lift coefficients in the case of a *User-Defined* wake interference model.
- *WAKE INTERFERENCE is used to specify that the present analysis should include wake interference effects, and to specify associated parameters.
- *WAKE UPSTREAM is used to specify the composition of the upstream structure in terms of element sets, for use in wake interference calculations. The keyword also selects the wake interference model to be used, and defines associated data to characterise the wake field.
- *WAVE-DEANS is used to specify Dean's Stream regular wave loading.
- *WAVE-GENERAL is used to specify miscellaneous parameters to wave loading.
- *WAVE-JONSWAP is used to specify a JONSWAP random sea wave spectrum or spectra.
- *WAVE-OCHI-HUBBLE is used to specify an Ochi-Hubble random sea wave spectrum or spectra.
- *WAVE-PIERSON-MOSKOWITZ is used to specify a Pierson-Moskowitz random sea wave spectrum or spectra.
- *WAVE-REGULAR is used to specify regular Airy wave loading.
- *WAVE-STOKES is used to specify Stokes V regular wave loading.
- *WAVE-TIME-HISTORY is used to specify a random seastate in terms of a time history of water surface elevation.
- *WAVE-TORSETHAUGEN is used to specify a Torsethaugen random sea wave spectrum or spectra.
- *WAVE-USER-DEFINED is used to specify a User-Defined random sea wave spectrum or spectra.

- `*WINCH` is used to define winch elements.
- `*WIND` is used to specify wind loading.
- `*WIND COEFF` is used to specify wind coefficients used to determine the wind loading on a floating body or moored vessel.

1.8.9.1 *AERODYN DRIVER

PURPOSE

To provide a link between the structural (Flexcom) and aerodynamic (AeroDyn) models in a wind turbine simulation.

THEORY

Refer to Coupling between Flexcom and AeroDyn for further information on this feature.

KEYWORD FORMAT

A block of lines which provides the key pieces of information to couple AeroDyn to Flexcom:

```
TOWER=Tower Element Set Name
WIND SPEED=Wind Speed
SHEAR EXPONENT=Shear Exponent
ROTOR SPEED=Fixed Rotor Speed
INPUT FILE=Name of AeroDyn Primary Input File
DRIVER FILE=Name of AeroDyn Driver Input File
```

TABLE INPUT

Input:	Description
Tower Element Set Name:	The portion (set of elements) of the finite element model which represents the tower.
Wind Speed:	The steady wind speed located at an elevation corresponding to the hub height. The hub height is determined from the vertical elevation of the *Hub Node* in the structural model. See Note (a).

Shear Exponent:	The power-law shear exponent which defines the wind speed profile as a function of vertical elevation. See Note (a).
Rotor Speed:	The fixed rotor speed (positive clockwise looking downwind).
AeroDyn Primary Input File Name:	The name of the AeroDyn primary input file. The file name must have a .dat file extension. This file may be generated using the $AERODYN section.
AeroDyn Driver Input File Name:	The name of the AeroDyn driver input file. The file name must have a .dvr file extension. This file will be generated by Flexcom when the analysis is run.

NOTES

(a) The local undisturbed wind speed, U(x), for any given blade or tower node is determined using the following expression:

$$U(x) = WindSpeed\left(\frac{x}{HubHeight}\right)^{ShearExponent}$$

where x is the instantaneous elevation of the blade or tower node above the mean water line. Refer to the AeroDyn documentation for further information.

(b) The file name entries allows you to control the names of the input files created by Flexcom for use by AeroDyn. The actual names chosen are not particularly important, but should be unique for every different load case simulation performed in order to avoid existing files being overwritten. A natural choice for both file names would be the name of the corresponding Flexcom dynamic simulation input file (.keyxm file extension). Note that the AeroDyn primary input file must have a .dat file extension, while the the AeroDyn driver input file must have a .dvr file extension. However all three files can, and ideally should, share the same file stub.

1.8.9.2 *ANALYSIS TYPE

PURPOSE

To specify the analysis type.

THEORY

Refer to Static Analysis, Time Domain Analysis, Frequency Domain Analysis and Quasi-Static Analysis for further information.

KEYWORD FORMAT

A single line that defines the analysis type.

```
TYPE=Analysis Type [, SOLUTION=Solution Type] [, MOORING=Mooring Type]
```

`Analysis Type` can be STATIC, DYNAMIC, FREQUENCY DYNAMIC, QUASI-STATIC or MOORING. `Solution Type` can be LINEAR or NONLINEAR (the default). The use of MOORING= is appropriate only if TYPE=MOORING, in which case `Mooring Type` can be FIXED, STATIC or DYNAMIC.

ANALYSIS TYPE

Purpose

To specify the analysis type.

Table Input

Input:	Description
Analysis Type:	The options are *Static* (the default), *Dynamic, Frequency Domain Dynamic, Quasi-Static,* and *Mooring.*

Notes

(a) If you select *Static, Dynamic, Frequency Domain Dynamic* or *Quasi-Static*, there is no further data specification required. If you select Mooring, then you must use the Mooring Analysis Type list to make a further choice of mooring analysis type.

(b) Depending on the analysis type you select, a number of options may become disabled, if they are not relevant to the selected analysis type. For example, if you select *Frequency Domain Dynamic*, any inputs which pertain specifically to the time domain (e.g. *Time, Time Stepping* etc.) are automatically disabled.

SOLUTION TYPE

Purpose

To specify a solution procedure.

Table Input

Input:	Description
Solution Type:	The options are *Nonlinear* (the default) and *Linear*.

MOORING ANALYSIS TYPE

Purpose

To choose between mooring analysis types.

Table Input

Input:	Description
Mooring Analysis Type:	The options are *Static Fixed* (the default), *Static Mooring* and *Dynamic Mooring*.

Notes

(a) You use this table to specify the required mooring analysis type if you selected *Mooring* in the *Analysis Type* list.

(b) Refer to Analysis Procedure for a discussion of the Flexcom mooring analysis types.

1.8.9.3 *BOUNDARY

PURPOSE

To define boundary conditions.

THEORY

Refer to Boundary Conditions for further information on this feature.

KEYWORD OVERVIEW

Boundary condition data is specified in blocks, with each block beginning with a TYPE= line defining the boundary condition type. This is then followed by as many lines as necessary to specify the boundary conditions of that type, and this in turn is then followed as necessary by data for other types. Data for the different boundary condition types can be specified in any order, and indeed types can be repeated in the same `*BOUNDARY` specification.

CONSTANT BOUNDARY CONDITIONS

Purpose

To define constant or time invariant boundary conditions.

Keyword Format

Block defining constant boundary conditions:

```
TYPE=CONSTANT
```

Line defining a single boundary condition:

```
Node (Number or Label), DOF [, Displacement] [, FIXATION=RELATIVE]
```

Line generating similar boundary conditions at multiple nodes:

```
GEN=Start Node (Number or Label), End Node (Number or Label) [, Node Incremen
```

If you specify a node label rather than a node number, it must be enclosed in {} brackets. `Node Increment` defaults to 1. `Displacement` defaults to 0. If the relative fixation option is invoked, the node is fixed with respect to the position at the end of the preceding analysis, rather than the initial position.

Table Input (Direct Specification)

Input:	Description
Node:	The node (number or label) at which the constant boundary condition is to be applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global DOF in which the boundary condition is applied at this node. For translations, specify a value of 1 for the global X -direction, 2 for the global Y-direction or 3 for the global Z-direction. DOFs 4, 5 and 6 refer to the components of the rotation vector at the node. Refer to Notes (b) and (c).
Displacement:	The magnitude of the applied boundary condition at this node. This defaults to a value of 0. Rotational boundary conditions are specified in degrees.
Fixation:	This option allows you to select whether the boundary condition is *Absolute* or *Relative*. If you select Absolute (the default), the node is fixed with respect to the initial position. If you select Relative, the node is fixed with respect to the position at the end of the preceding analysis, rather than the initial position.

Table Input (Generate Specification)

Input:	Description
Start Node:	The node (number or label) at which the first constant boundary condition is to be applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
End Node:	The node (number or label) at which the last constant boundary condition is to be applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Increment:	The increment to be used in assigning boundary conditions to the range of nodes. This input defaults to a value of 1, which will apply in the majority of cases. To specify a decrement value, simply input a negative number.

DOF:	The global DOF in which the boundary condition is applied at this node. For translations, specify a value of 1 for the global X -direction, 2 for the global Y-direction or 3 for the global Z-direction. DOFs 4, 5 and 6 refer to the components of the rotation vector at the node. Refer to Note (b).
Displacement:	The magnitude of the applied boundary condition at this node. This defaults to a value of 0. Rotational boundary conditions are specified in degrees.
Fixation:	This option allows you to select whether the boundary condition is *Absolute* or *Relative*. If you select Absolute (the default), the node is fixed with respect to the initial position. If you select Relative, the node is fixed with respect to the position at the end of the preceding analysis, rather than the initial position.

Notes

(a) The specification of boundary conditions is optional, and by default no constraints are applied. This allows free-falling objects, for example, to be analysed. However, most problems will require some boundary conditions to be specified.

(b) It is important if specifying displacements in DOFs 4-6 to understand the significance of these degrees of freedom. Refer to Preliminary Note on Rotational Constraints for further information.

VESSEL BOUNDARY CONDITIONS

Purpose

To identify those nodes on the structure whose motions are defined from the motions of an attached vessel.

Keyword Format

Line defining a user subroutine for vessel motions:

```
USER_VESSEL=File Name
```

Block defining vessel boundary conditions:

```
TYPE=VESSEL, VESSEL=Vessel Name
```

Line defining a single vessel boundary condition:

Node (Number or Label), DOF [, *Displacement*]

If any drift motions are defined using a user subroutine, then the USER_VESSEL=option is mandatory. If you specify a node label rather than a node number, it must be enclosed in {} brackets. Node Increment defaults to 1. *Displacement* defaults to 0. Data for any *Vessel Name* specified here must be input under *VESSEL.

Table Input (Vessel Motion – Drift – Subroutine File)

Input:	Description
Subroutine File:	The name of the DLL file containing the drift user-subroutine.

Table Input (Direct Specification)

Input:	Description
Vessel:	The name of the attached vessel. See Note (b).
Node:	The node (number or label) at which the vessel boundary condition is to be applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global DOF in which the boundary condition is applied at this node. For translations, specify a value of 1 for the global X - direction, 2 for the global Y-direction or 3 for the global Z-direction. DOFs 4, 5 and 6 refer to the components of the rotation vector at the node. Refer to Note (c).
Displacement:	The magnitude of the offset applied to connect the riser to the vessel in its reference position. The default value is 0. Refer to Note (d).

Notes

(a) The specification of boundary conditions is optional, and by default no constraints are applied. This allows free-falling objects, for example, to be analysed. However, most problems will require some boundary conditions to be specified.

(b) More than one attached vessel can be included in a Flexcom analysis. You distinguish between connected vessels by giving each a unique name. The name you choose is then used in this *Boundary - Vessel* table to identify the vessel whose motions define the response at a particular node. The actual name of course has no significance other than as a label to distinguish between vessels. You use the name again in the other tables, such as those used to specify the vessel initial position, offset, RAO file name, drift motions etc.

(c) It is important if specifying displacements in DOFs 4-6 to understand the significance of these degrees of freedom. Refer to Preliminary Note on Rotational Constraints for further information.

(d) A full discussion of the calculation of vessel motions and corresponding structural displacements is given in Vessels and Vessel Motions.

(e) Refer to Arbitrary Boundary Conditions for a detailed discussion of the drift user-subroutine facility.

SINUSOIDAL BOUNDARY CONDITIONS

Purpose

To identify those nodes on the structure whose boundary conditions vary sinusoidally with time.

Keyword Format

Block defining sinusoidal boundary conditions:

```
TYPE=SINUSOIDAL
Node (Number or Label), DOF, Amplitude, Phase [, Period]
```

If you specify a node label rather than a node number, it must be enclosed in {} brackets. The period of a sinusoidal boundary condition defaults to the wave period if a single regular wave is specified, otherwise a period must be specified. Sinusoidal boundary conditions are not relevant for frequency domain analysis.

Table Input

Input:	Description

Node:	The node (number or label) at which the sinusoidal boundary condition is to be applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global DOF in which the boundary condition is applied at this node. For translations, specify a value of 1 for the global X -direction, 2 for the global Y-direction or 3 for the global Z-direction. DOFs 4, 5 and 6 refer to the components of the rotation vector at the node. Refer to Note (c).
Amplitude:	The amplitude of the sinusoidal function defining the motion.
Phase:	The phase of the sinusoidal function relative to the wave datum.
Period:	The period of the sinusoidal function defining the motion. If a (single) regular wave is used in the analysis, this input defaults to the period of the wave. There is no default value for multiple regular waves or irregular waves.

Notes

(a) The specification of boundary conditions is optional, and by default no constraints are applied. This allows free-falling objects, for example, to be analysed. However, most problems will require some boundary conditions to be specified.

(b) More than one sinusoidal boundary condition may be imposed at a particular Node-DOF combination.

(c) It is important if specifying displacements in DOFs 4-6 to understand the significance of these degrees of freedom. Refer to Preliminary Note on Rotational Constraints for further information.

(d) Where structure displacements are specified using the Boundary - Sinusoidal option, any other boundary condition may also be applied at the same degree of freedom and at the same node. This can be used, for example, where vessel drift is specified using the Boundary - Sinusoidal option, and high frequency vessel motion is specified using the Boundary - Vessel option.

(e) This sinusoidal boundary condition option was provided originally for modelling the effect of vessel drift. However more comprehensive and accurate drift modelling capabilities are now provided, and you may prefer to use those now for that purpose. Sinusoidal boundary conditions are retained for compatibility with earlier versions, and you are of course perfectly entitled to use them as required.

TIMETRACE BOUNDARY CONDITIONS

Purpose

To identify the nodes on the structure whose time-varying displacements are to be read from an ASCII data file, and to input the name of that file.

Keyword Format

Block defining timetrace file boundary conditions:

```
TYPE=FILE, USER=File Name
Node (Number or Label), DOF
```

File Name should include the entire path of the boundary condition file with its extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks. If you specify a node label rather than a node number, it must be enclosed in {} brackets.

Table Input

Input:	Description
Displacement File:	The name of the ASCII data file containing the timetrace of displacement to be applied. This should ideally include the entire path of the file, including extension. If the file name or any part of its path contains spaces, the full name should be enclosed in double quotation marks (" ").
Node:	The node (number or label) at which the boundary condition is to be applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.

DOF:	The global DOF in which the boundary condition is applied at this node. For translations, specify a value of 1 for the global X-direction, 2 for the global Y-direction or 3 for the global Z-direction. DOFs 4, 5 and 6 refer to the components of the rotation vector at the node. Refer to Note (b).

Notes

(a) The specification of boundary conditions is optional, and by default no constraints are applied. This allows free-falling objects, for example, to be analysed. However, most problems will require some boundary conditions to be specified.

(b) It is important if specifying displacements in DOFs 4-6 to understand the significance of these degrees of freedom. Refer to Preliminary Note on Rotational Constraints for further information.

(c) Refer to Displacement Boundary Conditions for further information on this feature.

REFERENCE POINT BOUNDARY CONDITIONS

Purpose

To identify the nodes on the structure whose time-varying motions are to be calculated from the motion of a so-called reference point, and to specify the name of the file containing the motions of this reference point.

Keyword Format

Block defining reference point timetrace file boundary conditions:

```
TYPE=REFERENCE, USER=File Name
Node (Number or Label), DOF [, Displacement]
```

`File Name` should include the entire path of the boundary condition file with its extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks. If you specify a node label rather than a node number, it must be enclosed in {} brackets.

Table Input

Input:	Description

Reference Point File:	The name of the ASCII data file containing the timetrace of reference point motion. This should ideally include the entire path of the file, including extension. If the file name or any part of its path contains spaces, the full name should be enclosed in double quotation marks (" ").
Node:	The node (number or label) at which the boundary condition is to be applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global DOF in which the boundary condition is applied at this node. For translations, specify a value of 1 for the global X-direction, 2 for the global Y-direction or 3 for the global Z-direction. DOFs 4, 5 and 6 refer to the components of the rotation vector at the node. Refer to Note (b).
Displacement:	The magnitude of the static offset applied to the node before connection to the nominal reference point.

Notes

(a) The specification of boundary conditions is optional, and by default no constraints are applied. This allows free-falling objects, for example, to be analysed. However, most problems will require some boundary conditions to be specified.

(b) It is important if specifying displacements in DOFs 4-6 to understand the significance of these degrees of freedom. Refer to Preliminary Note on Rotational Constraints for further information.

(c) Refer to Reference Point Boundary Conditions for further information on this feature.

USER-SUBROUTINE BOUNDARY CONDITIONS

Purpose

To identify nodes on the structure whose boundary conditions are defined via a user-subroutine.

Keyword Format

Line defining a user subroutine for boundary conditions:

```
USER_BOUNDARY=File Name
```

Block defining user subroutine boundary conditions:

```
TYPE=SUBROUTINE
Node (Number or Label), DOF
```

The USER_BOUNDARY entry is only relevant if (at least one) boundary condition is defined in a user subroutine. `File Name` should include the entire path of the boundary condition file with its extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks. If you specify a node label rather than a node number, it must be enclosed in {} brackets.

Table Input (Boundary – Subroutine)

Input:	Description
Node:	The node (number or label) at which the boundary condition is to be applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global DOF in which the boundary condition is applied at this node. For translations, specify a value of 1 for the global X - direction, 2 for the global Y-direction or 3 for the global Z-direction. DOFs 4, 5 and 6 refer to the components of the rotation vector at the node. Refer to Note (b).

Table Input (Boundary – Subroutine File)

Input:	Description
Subroutine File:	The name of the DLL file containing the boundary condition user-subroutine.

Notes

(a) The specification of boundary conditions is optional, and by default no constraints are applied. This allows free-falling objects, for example, to be analysed. However, most problems will require some boundary conditions to be specified.

(b) It is important if specifying displacements in DOFs 4-6 to understand the significance of these degrees of freedom. Refer to Preliminary Note on Rotational Constraints for further information.

(c) Further details regarding the use of the boundary condition user-subroutine facility are contained in Arbitrary Boundary Conditions. You must specify a boundary condition subroutine file name using the *Boundary - Subroutine* Files table if you invoke the *Boundary - Subroutine* option.

HARMONIC BOUNDARY CONDITIONS

Purpose

To identify nodes on the structure whose motions vary harmonically in a frequency domain regular wave analysis, and to specify the parameters relating to this harmonic variation.

Keyword Format

Block defining harmonic boundary conditions:

```
TYPE=HARMONIC
Node (Number or Label), DOF [, Static Displacement] [, Harmonic Displacement]
```

If you specify a node label rather than a node number, it must be enclosed in {} brackets.

Static Displacement, *Phase Lag* and *Harmonic Displacement* default to 0.

Harmonic boundary conditions are not relevant for time domain analysis.

Table Input

Input:	Description
Node:	The node (number or label) at which the boundary condition is to be applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global DOF in which the boundary condition is applied at this node. For translations, specify a value of 1 for the global X-direction, 2 for the global Y-direction or 3 for the global Z-direction. DOFs 4, 5 and 6 refer to the components of the rotation vector at the node. See Note (b).

Static Displacement:	The mean static displacement (defaulting to zero) about which the harmonic motion takes place.
Harmonic Displacement:	The amplitude of the harmonic displacement (defaulting to zero).
Phase Lag:	The phase lag relative to the regular wave input in degrees (defaulting to zero). See Note (c).

Notes

(a) The specification of boundary conditions is optional, and by default no constraints are applied. This allows free-falling objects, for example, to be analysed. However, most problems will require some boundary conditions to be specified.

(b) It is important if specifying displacements in DOFs 4-6 to understand the significance of these degrees of freedom. Refer to Application of Rotational Constraints for further information.

(c) The phase angle is relative to the wave at the static offset position. By convention, a positive phase angle implies a phase lag, whereas a negative phase angle implies a phase lead.

1.8.9.4 *CALM LOAD

PURPOSE

To specify data defining the force terms to be applied to a CALM buoy.

THEORY

Refer to CALM Buoy for further information on this feature.

KEYWORD FORMAT

Three lines defining the calm buoy loads:

```
Heave Force Amplitude, Heave Force Phase
Surge Force Amplitude, Surge Force Phase
Pitch Force Amplitude, Pitch Force Phase
```

TABLE INPUT

Input:	Description
DOF:	The buoy degree of freedom in which a sinusoidally varying point load will be applied directly in the local axes at the COG node. This column has three pre-defined values that you cannot change. These are 'Heave', 'Surge' and 'Pitch', the significance of which is obvious. For each DOF you specify force RAO and phase values.
Force RAO:	Force RAO for this degree of freedom. See Note (b).
Phase:	Force phase angle relative to the wave at the COG node for this degree of freedom. See Note (b).

NOTES

(a) This keyword specifies the force terms to be applied to a calm buoy you defined in the *CALM MODEL keyword. If your Flexcom model includes a CALM buoy, both of these keywords must contain data for the buoy.

(b) The sinusoidally varying point load applied directly in the local axes at the COG node is calculated from the data you input here using the following equation:

$$F_i = A_w R_i \cos(kS - \omega t + \phi_i)$$

where:

- F_i is the force in degree of freedom i
- A_w is the wave amplitude
- R_i is the force RAO for degree of freedom i
- k is the regular wave number (calculated by Flexcom)
- S is the distance of the CoG node from the vertical axis (Y=Z=0) at the start of the dynamic analysis (calculated by Flexcom)

- ω is the regular wave circular period
- t is the current analysis or simulation time
- φ_i is the force phase angle

1.8.9.5 *CLASHING SOLUTION

PURPOSE

To specify solution parameters associated with clashing.

THEORY

Refer to Timestep Size, Contact Ramp and Axis System for further information on this feature.

KEYWORD FORMAT

Two lines of data defining solution parameters as follows.

```
[AXIS=Axis Type], [OUTPUT=YES]
[Maximum Time Step] [, Threshold Clearance] [, Contact Ramp] [, Successive Sc
```

All parameters are optional. Axis type can be `Instantaneous` or `Initial`, defaulting to `Instantaneous` if omitted. `Maximum Time Step` defaults to the analysis maximum time step if omitted. `Threshold Clearance` defaults to 10% (of the average contact diameter) if omitted. `Contact Ramp` defaults to 0% (of the average contact diameter) if omitted. `Successive Solutions` defaults to 0 if omitted.

TABLE INPUT

Input:	Description
Maximum Time Step:	The maximum time step to be used during clashing. This input is optional. This input defaults to the overall maximum analysis time step, assuming that you are using a variable time step. See Note (a).

Threshold Clearance:	The minimum clearance below which the maximum time step specified above must be adhered to. This value is input as a percentage of the average contact diameter of the elements which come into contact. This input is optional, and defaults to 10%. See Note (a).
Contact Ramp:	The distance over which the contact stiffness is ramped up. This value is specified as a percentage of the average contact diameter of the elements. This input is optional, and defaults to 0%, which means that no ramp is modelled. See Note (b).
Successive Solutions:	The number of successive solutions after which contact is deemed to be firmly established (i.e. the connected nodes remain the same). This option is only relevant if the interaction between two lines is likely to be constant rather than intermittent. See Note (c).
Axis System	This option allows you to specify whether the line of action of contact force application is to be based on the *Instantaneous* (the default) or the Initial positions of the contact elements. See Note (d).
Output File:	This option allows you to specify that active contact elements are to be echoed to the Output File. The options are No (the default) and Yes. See Note (e).

NOTES

(a) The Maximum Time Step limits the time step used during clashing. Refer to Timestep Size for further details.

(b) When two lines come into close proximity, the interaction between the two lines is effectively modelled by the insertion of a non-linear spring between the points at which minimum separation occurs. Refer to Contact Ramp for further details.

(c) In certain circumstances, it may not be desirable to persist with a very small time step while clashing is taking place, particularly if the interaction between two lines is likely to be constant rather than intermittent. Refer to Timestep Size for further details.

(d) By default, the Axis System for application of contact forces between a pair of contact elements is based on the instantaneous positions of the elements at any given time. Refer to Axis System for further details.

(e) You can optionally specify that active contact elements are be echoed to the Output File. If you invoke this option, the output file shows you exactly what elements contact each other at every solution time. This feature is particularly useful as it allows you to refine the contact set element definitions and consequently reduce run times.

1.8.9.6 *CONTACT MODELLING

PURPOSE

To specify specialised parameters relating to guide surface contact modelling.

THEORY

Refer to Contact Modelling for further information on this feature.

KEYWORD FORMAT

A block of two lines defining the how the contact stiffness is ramped up.

```
RAMP=Contact Ramp [, POWER=Exponent]
```

`Contact Ramp` defaults to 0.0. `Exponent` defaults to 1.5, but this value is immaterial unless a non-zero value is assigned to `Contact Ramp`.

TABLE INPUT

Input:	Description
Contact Ramp:	The distance over which the contact stiffness is ramped up to its full value. By default, the contact ramp is assumed to be zero, in which case a linear contact model is used.
Exponent:	The exponent to be used in the power-law approximation. This entry is optional and defaults to a value of 1.5 if omitted.

1.8.9.7 *CRITERIA

PURPOSE

To specify certain criteria that need to be satisfied in a static analysis, and to define how the model is to be adjusted to satisfy the desired criteria.

THEORY

Refer to Solution Criteria Automation for further information on this feature.

KEYWORD FORMAT

The keyword begins with a mandatory line defining the adjustment criterion, which may be repeated as often as necessary for multiple criteria:

`CRITERION=`*Criterion Type*`, MONITOR=`*`Criterion Monitor`*`, TARGET=`*`Target Value`*` [, TOLE`

`Criterion Type` may be HANGOFF ANGLE, TENSION, AXIAL STRESS, AXIAL STRAIN, BENDING STRESS, BENDING STRAIN, BENDING MOMENT, SUPPORT REACTION, SEABED CONTACT or VON MISES STRESS.

`Criterion Monitor` refers to an element number, an element label, an element set or a guide name definition. *`Target Value`* is a numerical value of the *`Criterion Type`* you want to achieve. *`Target Tolerance`* (optional) is a percentage that is permitted as an acceptable deviation from the target value and defaults to 1% if not specified.

The second line of the keyword is mandatory line specifying the adjustment variable. The adjustment variable options are NODE, VESSEL or LENGTH. The format of the line depends on the selected option.

Line specifying that a nodal location is to be varied (the direction of motion is controlled by the specified vector):

`VARIABLE=NODE, ADJUST=`*Node (Number or Label)*`, VECTOR=`*Vector Name*

Or:

Line specifying that a vessel location is to be varied (the direction of motion is controlled by the specified vector):

`VARIABLE=VESSEL, ADJUST=`*`Vessel Name`*`, VECTOR=`*`Vector Name`*

Or:

Line specifying that a section length is to be varied:

```
OPTION=LENGTH, ADJUST=Element (Number or Label or Set Name)
```

Vector Name refers to a vector defined using *VECTOR.

Finally the last line of the keyword is optional, and contains some solution parameters:

```
[MAX ITERATIONS=Max Iterations][, MIN ADJUSTMENT=Min Adjustment][, MAX ADJUST
```

Max Iterations is the maximum number of iterations permitted to satisfy the criteria. This entry is optional and defaults to 100 if omitted. Min Increment is the lower bound that Flexcom may adjust the variable by in order to satisfy a criterion. This entry is optional and defaults to 0.1m (or 0.1 degrees for angular adjustments) if omitted. *Max Increment* is the upper bound that Flexcom may adjust the variable within a single increment. This entry is optional and defaults to 50m (or 10 degrees for angular adjustments) if omitted.

TABLE INPUT

Criterion Type

Input:	Description
Criterion Type:	The criterion type which will be monitored for a predefined value or range.
Criterion Monitor:	The element number/label/set name (or guide name) where the criterion is to be monitored.
Target Value:	The target value in units of the criterion.
Tolerance:	The tolerance value in percent (0 – 100%) that is permitted as an acceptable deviation from the target value. This entry is optional and defaults to 1%.

Criteria Variable – Adjust Node

Input:	Description
Adjust:	The node (number or label) whose position is to be adjusted. If you specify a node label, it must be enclosed in {} brackets.
Vector:	The name of the vector along which the node is adjusted.

Criteria Variable – Adjust Length

Input:	Description
Adjust :	The element (number or label or set) whose length is to be adjusted. If you specify an element label, it must be enclosed in {} brackets.

Criteria Variable – Adjust Vessel

Input:	Description
Adjust:	The name of the vessel whose position is to be adjusted.
Vector:	The name of the vector along which the vessel is adjusted.

Criteria Solution Parameters

Input:	Description
Max Iterations:	The maximum number of iterations permitted to satisfy the criteria. This entry is optional and defaults to 100 if omitted.
Min Adjustment:	The lower bound that Flexcom may adjust the variable within a single iteration. This entry is optional and defaults to 0.1m (or 0.1 degrees for angular adjustments) if omitted.
Max Adjustment:	The upper bound that Flexcom may adjust the variable within a single iteration. This entry is optional and defaults to 50m (or 10 degrees for angular adjustments) if omitted.

Notes

(a) Note also that *PRINT is used to request additional printed output to the main output file. Specifically, the OUTPUT=CRITERIA option may be used to request additional information regarding solution convergence towards specified criteria.

1.8.9.8 *CURRENT

PURPOSE

To specify current loading.

THEORY

Refer to Current Loading for further information on this feature.

KEYWORD FORMAT

A block of lines whose format depends on the current type.

Block of lines defining a current that is constant with depth:

```
TYPE=UNIFORM
Velocity[, Direction ]
```

Block of lines defining a piecewise linear current. The second line is repeated as often as necessary to define the current profile:

```
TYPE=PIECEWISE LINEAR [, OPTION=Specification Option] [, LEVEL=Level Option]
Elevation, Velocity[, Direction]
```

One line specifying a user defined subroutine:

```
TYPE=SUBROUTINE, USER=File Name
```

At least two points must be defined for a piecewise linear current. The user subroutine option is not relevant for frequency domain dynamic analysis. *Direction defaults* to 0. *Specification Option* can be either ASCEND (the default) or DESCEND, defining the current profile in terms of distance above the datum X=0 or distance below the MWL, respectively. *Level Option* can be either MWL or AMBIENT (the default) to specify current stretching, see Note (f) below. *File Name* should include the entire path of the user subroutine file with its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

UNIFORM CURRENT

Purpose

To specify a horizontal current velocity that is uniform with depth.

Table Input

Input:	Description
Velocity:	The horizontal current velocity.
Direction:	The current direction, measured in degrees anti-clockwise from the global Y-direction. The default is 0°.

Notes

(a) The specification of current is optional. By default, no current is included in the analysis.

(b) If a seabed is not included in the model then the current is applied only above the datum X=0.

PIECEWISE LINEAR CURRENT

Purpose

To define a horizontal current velocity distribution that is varying in magnitude and direction with depth.

Table Input (Elevation Specification)

Input:	Description
Elevation Above Datum:	The vertical distance from the datum X=0 to the point at which the current velocity and direction is being specified.
Velocity:	The horizontal current velocity at this point.
Direction:	The current direction at this point, measured in degrees anti-clockwise from the global Y-direction. The default value is 0°.

Table Input (Depth Specification)

Input:	Description
Distance Below Mean Waterline:	The vertical distance from the MWL to the point at which the current velocity and direction is being specified.
Velocity:	The horizontal current velocity at this point.
Direction:	The current direction at this point, measured in degrees anti-clockwise from the global Y-direction. The default value is 0°.

Table Input (Water Surface)

Input:	Description
Current Stretching:	The options are *Ambient* (the default) and *MWL*. See Note (f).

Notes

(a) A minimum of two points is required to define a piecewise linear current distribution.

(b) The current can be specified at arbitrary depths below the datum in the case of a sloping or arbitrary seabed. If a seabed is not included in the model then the current is applied only above the datum.

(c) Current velocity and direction at elevations intermediate to those specified here are found by linear interpolation.

(d) Below the lowest elevation specified, the current velocity and direction are assumed constant, with values equal to those at the lowest elevation.

(e) Above the highest elevation specified, the current velocity and direction are assumed constant with values equal to those at the highest elevation.

(f) Flexcom by default applies current based on the depth relative to the AMBIENT water surface. In this case, the current velocity distribution is “stretched” (or “compressed”) so that the velocity at the ambient water surface (rather than at the *MWL*) is constant throughout an analysis. With the alternative *MWL* option, the current velocity distribution remains unaltered throughout an analysis, regardless of the ambient water surface elevation.

(g) The specification of current is optional. By default, no current is included in the analysis.

USER-SUBROUTINE CURRENT

Purpose

To specify the name of the DLL file containing the current user-subroutine.

Table Input

Input:	Description
Current Subroutine File:	The name of the DLL file containing the current user-subroutine.

Notes

(a) This table is not relevant for frequency domain dynamic analysis.

1.8.9.9 *CURRENT COEFF

PURPOSE

To specify current coefficients used to determine the current loading on a floating body or moored vessel.

THEORY

Refer to Current Loads and Current Force Coefficients for further information on this feature.

KEYWORD FORMAT

Data may defined explicitly within the keyword file itself, or defined separately in an external file which is simply referenced using a FILE= entry. The latter approach is highly recommended as it reduces unnecessary clutter in the keyword file, and also helps to improve user interface performance and enhanced user experience.

Floating Body

A block of data consisting of floating body name followed by a single line defining the name of an external file which contains current coefficient data.

```
FLOATING BODY=Floating Body Name
FILE=File Name
```

File Name should include the entire path of the file including its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

A block of data consisting of floating body name followed by a single line repeated as often as necessary to define the current coefficients over a sufficiently large range of headings. The block itself may be repeated for different floating bodies.

```
FLOATING BODY=Floating Body Name
Floating Body/Current Heading, Current Coefficient in Surge,
Current Coefficient in Sway, Current Coefficient in Yaw
```

Moored Vessel

A block of data consisting of a moored vessel tag line followed by a single line defining the name of an external file which contains current coefficient data.

```
MOORED VESSEL
FILE=File Name
```

File Name should include the entire path of the file including its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

A block of data consisting of a moored vessel tag line followed by a single line repeated as often as necessary to define the current coefficients over a sufficiently large range of headings. Only one block is required, as a mooring analysis only contains one moored vessel.

```
MOORED VESSEL
Moored Vessel/Current Heading, Current Coefficient in Surge,
Current Coefficient in Sway, Current Coefficient in Yaw
```

TABLE INPUT

Floating Body (External File)

Input:	Description
Floating Body:	The name of the floating body.
File Name:	The name of the external data file.

Floating Body

Input:	Description
Floating Body:	The name of the floating body.
Heading:	The current heading in degrees.
Cc_y:	The current coefficient in surge, corresponding to the specified heading.
Cc_z:	The current coefficient in sway, corresponding to the specified heading.
Ccθ:	The current coefficient in yaw, corresponding to the specified heading.

Moored Vessel (External File)

Input:	Description
File Name:	The name of the external data file.

Moored Vessel

Input:	Description
Heading:	The current heading in degrees.
Cc_y:	The current coefficient in surge, corresponding to the specified heading.
Cc_z:	The current coefficient in sway, corresponding to the specified heading.
Ccθ:	The current coefficient in yaw, corresponding to the specified heading.

NOTES

(a) If the equivalent current incidence does not exactly match one of the relative current headings specified, Flexcom linearly interpolates between the nearest user-specified headings to find the relevant values of the current coefficients.

(b) If the equivalent current incidence is outside the range of user-specified headings, the current coefficients are assumed to be zero. For this reason, it is recommended that you specify current coefficients over a sufficiently large range of headings to ensure that current coefficients are defined for any heading likely to arise in an analysis.

(c) The headings for the current coefficients need not necessarily be specified in any particular order (ascending order is not necessary).

1.8.9.10 *DAMPING

PURPOSE

To incorporate damping into a dynamic analysis.

THEORY

Refer to Damping for further information on this feature.

KEYWORD FORMAT

A block of lines which define the damping parameters as a function of element set. The block of lines can be repeated as often as necessary.

Block defining damping coefficients for standard type.

```
TYPE=STANDARD
SET=Set Name
Stiffness Damping, Mass Damping
```

Block defining individual stiffness damping coefficients for each mode of deformation, namely axial, bending and torsion.

```
TYPE=DEFORMATION
SET=Set Name
Axial Stiffness Damping Coefficient, Bending Stiffness Damping Coefficient,
Torsional Stiffness Damping Coefficient, Mass Damping
```

The specification of damping for an element set using the *DAMPING keyword precludes the specification of damping for the same element set using the *DAMPING RATIO keyword, and vice versa. Deformation mode damping (TYPE=DEFORMATION) pertains specifically to time domain analysis, and should be omitted for frequency domain analysis.

DAMPING COEFFICIENTS

Purpose

To specify damping coefficients.

Table Input

Input:	Description
Element Set Name:	The name of the element set to which the damping coefficients are assigned. This defaults to all elements.
Stiffness Damping Coefficient:	The value of the stiffness damping coefficient λ. See Note (a). This defaults to a value of 0.
Mass Damping Coefficient:	The value of the mass damping coefficient μ. See Note (a). This defaults to a value of 0.

Notes

(a) Structural damping is generally specified directly in terms of stiffness and mass proportional damping coefficients. Refer to Damping Coefficients for further details.

(b) If damping coefficients are assigned to an element set in this table, the same element set may not be referenced in the Damping – Deformation or Damping Ratio tables.

DEFORMATION MODE DAMPING

Purpose

To specify deformation mode damping in a time domain dynamic analysis.

Table Input

Input:	Description
Element Set:	The name of the element set to which the damping data is assigned. This defaults to all elements.
Axial Coefficient:	The value of the axial stiffness damping coefficient λ_{Axial}. See Note (a). This defaults to a value of 0.
Bending Coefficient:	The value of the bending stiffness damping coefficient $\lambda_{Bending}$. See Note (a). This defaults to a value of 0.
Torsion Coefficient	The value of the torsion stiffness damping coefficient $\lambda_{Torsion}$. See Note (a). This defaults to a value of 0.
Mass Coefficient	The value of the mass damping coefficient m. See Note (a). This defaults to a value of 0.

NOTES

(a) Specification of deformation mode damping allows you to define individual stiffness damping coefficients for each of the structure deformation modes, namely, axial, bending and torsion. Refer to Deformation Mode Damping for further details.

(b) If damping is assigned to an element set in this table, the same element set may not be referenced in the Damping or Damping Ratio tables.

(c) This table is not relevant for frequency domain dynamic analysis.

1.8.9.11 *DAMPING FORMULATION

PURPOSE

To specify the damping formulation to be used in a time domain dynamic analysis.

THEORY

Refer to Damping Formulation for further information on this feature.

KEYWORD FORMAT

A single line to specify the damping formulation.

```
OPTION=Damping Formulation
```

Damping Formulation may be either UPDATED (the default) or CONSTANT. This option pertains specifically to time domain analysis, and should be omitted for frequency domain analysis.

TABLE INPUT

Input:	Description
Damping Formulation:	This option allows you to select the damping formulation. There are two types available, *Updated* (the default) and *Constant*. See Note (a).

NOTES

(a) The structure damping matrix is by default computed at each solution step based on the instantaneous mass and stiffness matrices, and this is the case when the *Updated* damping formulation is selected. Alternatively, a *Constant* damping matrix may be used throughout an analysis, based on the mass and stiffness matrices at the static equilibrium position (that is, the position at the start of the dynamic analysis).

(b) This option is not relevant for frequency domain dynamic analysis.

1.8.9.12 *DAMPING RATIO

PURPOSE

To specify stiffness damping coefficients as a function of a damping ratio and a damping period.

THEORY

Refer to Damping Ratio for further information on this feature.

KEYWORD FORMAT

A block of lines which define the damping parameters as a function of element set. The block of lines can be repeated as often as necessary.

Block defining damping ratio data:

```
SET=Set Name
Damping Ratio [, Damping Period]
```

The specification of damping for an element set using `*DAMPING` precludes the specification of damping for the same element set using `*DAMPING RATIO`, and vice versa. The *Damping Ratio* must be between 0 and 1. The *Damping Period* is optional in the case of a single regular wave analysis and defaults to the regular wave period. Otherwise it is a mandatory input.

TABLE INPUT

Input:	Description
Element Set:	The name of the element set to which the damping ratio data is assigned. This defaults to all elements.
Damping Ratio:	The value of the damping ratio ξ.
Damping Period:	The value of the damping period T.

NOTES

(a) Rather than explicitly defining a coefficient (or coefficients), stiffness proportional damping may alternatively be defined as a function of a damping ratio and a damping period in a time domain dynamic analysis. Refer to Damping Ratio for further details.

(b) The damping ratio ξ must be between 0 and 1. If the excitation in your dynamic analysis is a single regular wave, then the specification of damping period T is optional. If omitted, T defaults to the regular wave period. If the excitation is of any other type, then the period input is required.

(c) If damping ratios and periods are assigned to an element set in this table, the same element set may not be referenced in the *Damping or Damping - Deformation* tables.

1.8.9.13 *DATABASE

PURPOSE

To specify the frequency of database output.

THEORY

Refer to Database Files for further information on this feature.

KEYWORD FORMAT

Block of one or two lines defining storage requirements:

`TIME=`*`Time Option`*

Line that follows if TIME=SELECTED is specified:

`Start Time, Recording Interval`

`Time Option` can be ALL, SELECTED, END or NONE. This keyword may occur once only in the `$LOAD CASE` section. It is not relevant for frequency domain dynamic analysis.

TABLE INPUT

Input:	Description

Record At:	This option allows you to specify the frequency of database output. The options include *All time steps* (the default), which produces database output at all solution times, *Selected time steps*, which produces output at a recording interval you specify, and *End of Analysis*, which produces output at the final solution time only. *None* suppresses the creation of database output.
Start Time:	The time (in seconds) at which to start output to the database file when the *Selected time steps* output option is invoked. By default output begins at the simulation start time.
Recording Interval:	The time interval in seconds between database outputs when the *Selected time steps* output option is invoked.

NOTES

(a) Database output is desirable in the vast majority of cases. However, the keyword/table inputs detailed above are optional and so may be inadvertently overlooked. In such circumstances, Flexcom creates a default database based on the analysis type...

- Static and quasi-static analyses – database output produced at the final solution time only (equivalent to the *End of Analysis* option)
- Dynamic analyses – database output produced at certain time steps only (equivalent to the *Selected time steps* option). The *Recording Interval* is equal to the solution time step in a fixed time step analysis, and the maximum time step in a variable time step analysis. The *Start Time* is equal to the solution start time.

1.8.9.14 *DATABASE CONTENT

PURPOSE

To customise the contents of the database output files.

THEORY

Refer to Database Files for further information on this feature.

KEYWORD FORMAT

Optional line specifying the set of elements which is be included in the database contents (defaults to SET=ALL if omitted).

`[SET=`*`Set Name`*`]`

Optional line specifying that statistics are to be included in the database:

`[OPTION=STATISTICS] [, START=`*`Statistics Start Time`*`]`

Optional line specifying parameters to be included in the database contents. Each of the parameters is denoted by a number as outlined below:

`[INCLUDE=`*`List of Numbers`*`]`

Optional line specifying parameters to be excluded from the database contents. Each of the parameters is denoted by a number as outlined below:

`[EXCLUDE=`*List of Numbers*`]`

Note that INCLUDE= and EXCLUDE= are mutually exclusive – any given parameter should not appear under both headings.

Optional line specifying whether output relates to nodes or local integration points:

`[OUTPUT=`*`Output Location`*`][, NUMBER=`*`Data Storage Points`*`]`

`Output Location` can be `NODES` (default) or `INTEGRATION-POINTS`. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis. *`Data Storage Points`* can have a value of 1 (element mid-points only), 2 (element start and end-points) or 3 (element start, middle and end-points - the default), and is only relevant when *`Output Location`* is set to `NODES`.

This keyword may occur once only in the `$LOAD CASE` section. It is not relevant for frequency domain dynamic analysis.

The parameters which correspond to the *List of Numbers* are as follows:

1. Motion
2. Velocity
3. Acceleration
4. Reactions
5. Axial Force

6. Local-y Shear Force
7. Local-z Shear Force
8. Torque
9. Local-y Bending Moment
10. Local-z Bending Moment
11. Effective Tension
12. Local-y Curvature
13. Local-z Curvature
14. Axial Strain
15. Temperature
16. Pressure
17. Damper Power
18. Plastic Strains
19. Equivalent Plastic Strains
20. Vessel Velocities and Accelerations
21. Convected Coordinate Axes

TABLE INPUT

Input:	Description
Set Name:	The name of the element set to be included in the database. This defaults to the set 'All'.
Statistics:	This option allows you to specify whether runtime generated statistics of nodal motions and element restoring forces are included in the database files. Runtime statistics are excluded by default. See Note (b).

Statistics Start Time:	The time (in seconds) from which Flexcom begins calculating runtime statistics. By default, calculation of runtime statistics begins at the end of the analysis ramp time, or at the start of the analysis if no ramp is specified. This entry is ignored unless statistics output to the database is requested. See Note (b).
Motion (1):	This option allows you to specify whether or not nodal motions are included in the motion database file. Nodal motions are included by default.
Velocity (2):	This option allows you to specify whether or not nodal velocities are included in the motion database file. Nodal velocities are excluded by default.
Acceleration (3):	This option allows you to specify whether or not nodal accelerations are included in the motion database file. Nodal accelerations are excluded by default.
Reactions (4):	This option allows you to specify whether reactions at restrained nodes are included in the force database file. Reactions are included by default.
Axial Force (5):	This option allows you to specify whether axial forces are included in the force database file. Axial forces are included by default.
Local-y Shear Force (6):	This option allows you to specify whether shear forces in the local-y axis are included in the force database file. Local-y shear forces are included by default.
Local-z Shear Force (7):	This option allows you to specify whether shear forces in the local-z axis are included in the force database file. Local-z shear forces are included by default
Torque (8):	This option allows you to specify whether torque forces are included in the force database file. Torque forces are included by default.

Local-y Bending Moment (9):	This option allows you to specify whether bending moments in the local-y axis are included in the force database file. Local-y bending moments are included by default.
Local-z Bending Moment (10):	This option allows you to specify whether bending moments in the local-z axis are included in the force database file. Local-z bending moments are included by default.
Effective Tension (11):	This option allows you to specify whether effective tensions are included in the force database file. Effective tensions are included by default.
Local-y Curvature (12):	This option allows you to specify whether curvatures in the local-y axis are included in the force database file. Local-y curvatures are included by default.
Local-z Curvature (13):	This option allows you to specify whether curvatures in the local-z axis are included in the force database file. Local-z curvatures are included by default.
Axial Strain (14):	This option allows you to specify whether axial strains are included in the force database file. Axial strains are included by default.
Temperature (15):	This option allows you to specify whether temperatures are included in the force database file. Temperatures are excluded by default.
Pressure (16):	This option allows you to specify whether internal and external pressures are included in the force database file. Internal and external pressures are included by default.
Damper Power (17):	This option allows you to specify whether power generated by damper elements are included in the force database file. Damper powers are excluded by default.

Plastic Strains (18):	This option allows you to specify whether if plastic axial strain, plastic local-y curvature and plastic local-z curvature are included in the force database file. These parameters are excluded by default. See Note (d).
Equivalent Plastic Strains (19):	This option allows you to specify whether if equivalent plastic axial strain and equivalent plastic curvature are included in the force database file. These parameters are excluded by default. See Note (d).
Vessel Velocity and Acceleration (20):	This option allows you to specify whether or not vessel velocities and accelerations are included in the motion database file. Vessel velocities and accelerations are excluded by default.
Convected Coordinate Axes (21):	This option allows you to specify whether or not element convected axes are included in the motion database file (they are excluded by default). These can be very useful for visualisation purposes in the Model View post-simulation. See Note (g).
Output Location:	This option allows you to specify whether element based outputs are provided on a node or integration point basis. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis (e).
Data Storage Points:	The number of data storage points per element. This can have a value of 1 (element mid-point only), 2 (element start and end-points) or 3 (element start, middle and end-points). See Note (f).

NOTES

(a) This table allows you to customise the contents of the motion and force database files that are produced by a Flexcom analysis. By reducing the number of parameters that are output to the database files, and the number of elements for which those parameters are output, it is possible to substantially reduce the size of the database files produced. This can be particularly useful for large models and/or long simulation runtimes, especially if only a particular section of the model or certain parameters are of interest.

(b) If requested, Flexcom automatically calculates statistical parameters (minimum, maximum, mean and standard deviation) during dynamic analyses for nodal positions and certain element restoring forces (effective tensions, local-y and local-z-shear forces, torque moments, and local-y and local-z bending moments) while the analysis is running. These results are shown in the main analysis output file (jobname.out) at the end of the analysis.

Additionally, these statistical parameters are written to the database files. This may be useful for analyses with long simulation runtimes where subsequently generating statistics using the database postprocessor can take a long time. If runtime-generated statistics have been included in the database files, and you subsequently request a plot of statistics using the database postprocessor, the postprocessor checks if the relevant data is included in the runtime statistics. If it is, the relevant data is simply read from the database files; if it is not, the database postprocessor must scan the database files to calculate the relevant statistical parameters. Including runtime-generated statistics in the database files can save a considerable amount of time during subsequent postprocessing, although this is at the expense of increased database file size.

If output of statistics is selected, statistical parameters are output for nodal motions (provided additionally that nodal motions are selected for output) and each of the element restoring forces that have been selected for output to the database files. Furthermore, statistics are output only for nodes and elements for which output to the database is specified (see Note (c)). The *Statistics Start Time* option allows you to specify at what point during the analysis calculation of statistics should commence. Flexcom excludes any values before this time. This allows you to exclude initial transients from statistical calculations.

(c) Data should only be entered into this table when database output is requested. Any data entered here only takes effect if the *Database Request* table is also present.

(d) This type of output is automatically suppressed if no plastic hardening models are associated with any element sets.

(e) This option is relevant to results that are calculated at the integration points, such as generalised strains and stresses. Motions, velocities, accelerations and reactions of any kind are still calculated and outputted at the relevant nodes. If this option is set to *Integration points*, then the parameters are stored in the force database file directly as calculated at integration points, without any smoothing. This type output can be used for testing purposes and is relevant only for elements with three integration points. If this option is set to *Nodes*, then the parameters are exported from the integration points to the nodes and middle of the element. They are ultimately smoothed based on the element connectivity in the mesh and stored in the force database file. This is the default option and is independent of the number of integration points specified for the element.

(f) By default, 3 data storage points are used for every element, ensuring full data storage. Using a reduced storage scheme can significantly reduce disk space however, potentially reducing the size of the force database file by over 60% if data is stored at element mid-points only. Note that this option is only relevant when results are stored on a nodal basis rather than integration points.

(g) Flexcom uses a convected coordinate axes technique for modelling finite rotations in three dimensions. Each element of the finite element discretisation has a convected axis system associated with it, which moves with the element as it displaces in space. Refer to Finite Element Formulation for further details. The Convected Coordinate Axes option here stores the local convected axes for each element as a function of time in the motion database file.

1.8.9.15 *DRIFT

PURPOSE

To define vessel drift motions.

THEORY

Refer to Low Frequency Drift Motions for further information on this feature.

KEYWORD FORMAT

A block of two lines repeated as many times as necessary to define all the vessel drifts. The first line specifies the vessel name and drift type.

```
VESSEL=Vessel Name, TYPE=Drift Type [, OPTION=Axis System]
```

The format of the second line depends on the drift type.

For drift motions read from a timetrace file:

```
FILE=File Name
```

For sinusoidal drift:

```
Drift DOF, Amplitude, Phase, Period
```

A vessel may not be assigned both timetrace drift and sinusoidal drift. Timetrace drift is not relevant for frequency domain dynamic analysis. `Drift Type` can be SINUSOIDAL or FILE. *Axis System* can be GLOBAL (the default) or LOCAL. `File Name` should include the entire path of the vessel drift motion file including its extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

TIMETRACE DRIFT

Purpose

To specify that vessel drift or low frequency motions are to be read from an ASCII timetrace file, and to specify the file name.

Table Input

Input:	Description
Vessel:	The name of the vessel for which drift data is being input.
Timetrace File Name:	The name of the ASCII data file containing the timetrace of low frequency motion of the vessel reference point. *Timetrace File Name* should ideally include the entire path of the vessel drift motion file, including extension. If the name or any part of its path contains spaces, the full *File Name* should be enclosed in double quotation marks (" ").
Axis Type:	The options are *Global* (the default) and *Local*.

Notes

(a) The format of the drift timetrace file is as follows. The file contains seven columns of data. The first column contains time data, and the remaining six correspond to displacements of the reference point in six degrees of freedom. Comment lines, denoted by a capital 'C' in the first column, are permitted, while lines that are completely blank are ignored. An example data file extract is shown below.

```
C
C Time    DOF1     DOF2    DOF3    DOF4    DOF5     DOF6
C
  0.0      0.0      0.0     0.0     0.0     0.0     0.0
  0.2      0.2      0.5     0.0     0.0     0.0     3.419
  0.4      0.3      0.9     0.0     0.0     0.0     6.794
  0.6      0.4      1.2     0.0     0.0     0.0    10.081
  0.8      0.6      1.4     0.0     0.0     0.0    13.236
  1.0      0.8      1.5     0.0     0.0     0.0    16.219
  1.2      0.9      1.6     0.0     0.0     0.0    18.990
  1.4      1.1      1.8     0.0     0.0     0.0    21.513
  1.6      0.8      2.0     0.0     0.0     0.0    23.757
  1.8      0.6      2.1     0.0     0.0     0.0    25.690
```

The first column of data contains time values. Columns 2 – 4 contain the displacements (not coordinates) of the reference point from its initial position at the start of the dynamic analysis. This means that the data in the timetrace file should not include the value of any static offset you apply to the vessel. If it does, what will happen is that the offset will be applied twice. These displacements are in either the global XYZ axes or else in a local axis system defined by the initial orientation of the vessel – you specify which using the *Vessel Motions – Axis Type* option. Columns 5 – 7 contain rotations in degrees. Column 5 contains the yaw rotation of the vessel about the vertical or global X axis. Columns 6 and 7 are roll and pitch respectively, relative to either the yawed or initial vessel axes, depending on whether large or small angle theory is specified. Note that each line of the file (other than comment lines or blank lines) must contain 7 numerical values.

Flexcom uses cubic spline interpolation to find the displacements and rotations of the reference point at times intermediate to those specified in the data file (for this reason the analysis solution times do not need to match those in the data file).

(b) The choice of axis system refers to data you specify for vessel drift motions. Motions in this context means translations only – rotations always refer to vessel axes. Naturally Global stipulates that translations represent drift or combined motions in any or all of the global X, Y or Z axes, and/or that the angle you specify in defining a constant velocity is relative to global Y. Conversely Local indicates that translations represent any or all of heave, surge or sway, defined with reference to the initial orientation of the vessel axes, or that the angle you specify in defining a constant vessel velocity is relative to the vessel surge axis.

SINUSOIDAL DRIFT

Purpose

To specify sinusoidal vessel drift or low frequency motions.

Table Input

Input:	Description
Vessel:	The name of the vessel for which drift data is being input.
DOF:	The degree of freedom (DOF) of the vessel drift motion. Specify a value of 1, 2 or 3 for a translation in the global or local axes, depending on the Axis System you nominate. A value of 4 represents a rotation about the global X axis, while 5 and 6 are low frequency roll and pitch respectively relative to either the yawed or initial vessel axes, depending on whether large or small angle theory is specified.
Amplitude:	The amplitude of the sinusoidal drift. A non-zero positive value is required.
Phase:	The phase in degrees of the sinusoidal drift, which defaults to 0. See Note (b).
Period:	The period in seconds of the sinusoidal drift. A non-zero positive value is required. See Note (c).
Axis Type:	The options are *Global* (the default) and *Local*.

Notes

(a) This sinusoidal drift facility is similar to the boundary conditions option offered via the *Boundary – Sinusoidal* (time domain) and *Boundary – Harmonic* (frequency domain) options. However, the data specified here defines the drift motion of a vessel reference point whereas the boundary condition options directly apply a sinusoidal motion to a node of the finite element discretisation.

(b) In the time domain, this input simply relates or ties the drift time variation to the time datum. A sinusoidal drift is actually applied as a cosine wave, so if the default phase value of 0^o is specified, the first maximum value of drift response occurs at time t=0s.

(c) In the frequency domain, the period of drift motions is very often outside of the range of harmonics in a wave spectrum. However there is no requirement on you to invoke the Wave – Frequencies option to force Flexcom to include a harmonic in a random sea discretisation at the drift frequency (or frequencies) you input here. The program does this automatically, without user intervention.

(d) The choice of axis system refers to data you specify for vessel drift motions. Motions in this context means translations only – rotations always refer to vessel axes. Naturally *Global* stipulates that translations represent drift or combined motions in any or all of the global X, Y or Z axes, and/or that the angle you specify in defining a constant velocity is relative to global Y. Conversely *Local* indicates that translations represent any or all of heave, surge or sway, defined with reference to the initial orientation of the vessel axes, or that the angle you specify in defining a constant vessel velocity is relative to the vessel surge axis.

1.8.9.16 *FD ANIMATION

PURPOSE

To define playback parameters for a representative (time domain) structural animation which is fabricated from a frequency domain solution.

THEORY

Refer to Model View for further information on this feature.

KEYWORD FORMAT

A single block containing two lines, both of which are optional:

```
[SUPPRESS=Suppress Option]
[Start Time] [, End Time] [, Time Step]
```

Suppress Option can be NO (default) or YES. *Start Time* defaults to the finish time of the preceding static analysis. *End Time* defaults to 4 wave periods for a regular wave analysis, or 0.5 hours (1800s) for a random sea analysis. *Time Step* defaults to1/12th of a wave period for a regular wave analysis, or 0.5s for a random sea analysis.

TABLE INPUT

Input:	Description
Start Time:	The simulation start time. This entry is optional and defaults to the finish time of the preceding static analysis.
End Time:	The simulation end time. This entry is optional and defaults to 4 wave periods for a regular wave analysis, or 0.5 hours (1800s) for a random sea analysis.
Time Step:	The database storage time step used in the creation of the animation. This entry is optional and defaults to 1/12th of a wave period for a regular wave analysis, or 0.5s for a random sea analysis.
Suppress Animation:	Option to suppress the generation of the structural animation. The default value is *No*, meaning that an animation is generated automatically unless explicitly suppressed.

1.8.9.17 *FORCE RAO

PURPOSE

To specify force RAOs for a floating body.

THEORY

Refer to First-Order Wave Loads and Force RAOs for further information on this feature.

KEYWORD OVERVIEW

Data may defined explicitly within the keyword file itself, or defined separately in an external file which is simply referenced using a FILE= entry. The latter approach is highly recommended as it reduces unnecessary clutter in the keyword file, and also helps to improve user interface performance and enhanced user experience.

DATA SPECIFIED IN EXTERNAL FILE

Keyword Format

Line defining floating body name:

```
FLOATING BODY=Floating Body Name
```

Line defining name of external file which contains force RAO data.

```
FILE=File Name
```

File Name should include the entire path of the file including its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

Table Input

Input:	Description
Floating Body:	The name of the floating body.
File Name:	The name of the external data file. See Note (a).

Notes

(a) Refer to the following sections for further information regarding the required format of data within the external file.

DATA SPECIFIED EXPLICITLY WITHIN KEYWORD FILE

Keyword Format

Block of data consisting of floating body name followed by force RAO data. An optional line indicates the position of the floating body reference point to be used in calculating the wave forces. If a rolling mean is to be used, this line must be followed by a line specifying the number of time steps. Following this is an optional line defining the layout of the RAO data. This is then followed by blocks of lines defining the actual RAO data. The format in which the RAO data is specified depends on the layout being used. The entire block of data can then be repeated to specify force RAO data for second and subsequent floating bodies.

Line to define floating body name:

```
FLOATING BODY=Floating Body Name
```

Line defining the layout of the RAO data.

```
[LAYOUT=Layout Name]
```

Optional line defining position of the reference point to be used in calculating wave forces:

```
[POSITION=ORIGINAL/INSTANTANEOUS/MEAN]
```

Optional line defining number of time steps, relevant to POSITION=MEAN only:

```
[TIMESTEPS=Number]
```

Optional line specifying whether or not a phase shift is to be included due to the position of the RAO reference point away from the global origin:

```
[WAVELENGTH PHASE SHIFT=INCLUDE/EXCLUDE]
```

Blocks of lines defining force RAO data. Flexcom supports two different schemes for laying out the data in the RAO file, namely the *MCS* layout and the *Line* layout.

The *MCS* layout begins with a single line defining the incident wave heading at which the RAOs are being defined. This is followed by a block of three lines specifying the incident wave frequency and the relevant RAO amplitudes and phases for this heading and wave frequency, as shown below. Note that entries shown below in italics should be replaced with their actual numeric values in the RAO file.

```
HEADING=Wave Heading
Wave Frequency
Heave RAO, Surge RAO, Sway RAO, Yaw RAO, Roll RAO, Pitch RAO
Heave Phase, Surge Phase, Sway Phase, Yaw Phase, Roll Phase, Pitch Phase
```

The block of three lines specifying the incident wave frequency, RAOs and phases are repeated as often as necessary until the RAOs are defined over the required range of wave frequencies. To define RAOs at more incident wave headings, you simply repeat the HEADING= line for the new wave heading, and the RAO data for this wave heading is specified as before. If RAOs are being defined for more than one incident wave heading, then they must also be defined for more than one frequency at each wave heading, and the number of frequencies at which the RAOs are defined is the same for each heading. If the RAOs are independent of the incident wave heading, then the HEADING= line should be omitted.

The *Line* layout is very similar to the standard *MCS* layout, with the exception that the RAO and phase data for a particular heading and frequency appear on a single line, the format of which is shown below. Note that, for clarity, the data shown below is split over a number of lines, but is in reality specified on a single line of the RAO file.

```
Wave Heading, Wave Frequency, Heave RAO, Heave Phase, Surge RAO, Surge Phase,
Yaw RAO, Yaw Phase, Roll RAO, Roll Phase, Pitch RAO, Pitch Phase
```

This line is repeated for every wave frequency and every wave heading for which RAOs are being defined. As with the *MCS* layout, if RAOs are being defined at more than one incident wave heading, then they must also be defined at more than one frequency at each wave heading, and the number of frequencies at which the RAOs are defined is the same for each heading. So, for example, if RAOs were to be defined at three wave headings and ten frequencies per heading, then 30 lines of data would be required to specify the RAOs using the Line layout. If the same wave heading is specified on all lines, then Flexcom assumes that the RAOs are independent of wave heading. Note also that the order in which the lines of RAO data appear is not important. Flexcom automatically sorts the RAO data by heading and frequency.

Data for any *Floating Body* Name defined here must be input under *FLOATING BODY. *Layout Name* can be either *MCS* or *LINE*, defaulting to *MCS* if omitted.

Table Input - MCS Layout

Input:	Description
Floating Body:	The name of the floating body.

Heading:	The wave heading in degrees.
Frequency:	The wave frequency in Hertz.
Heave RAO:	The RAO in heave, corresponding to the specified heading and frequency.
Surge RAO:	The RAO in surge, corresponding to the specified heading and frequency.
Sway RAO:	The RAO in sway, corresponding to the specified heading and frequency.
Yaw RAO:	The RAO in yaw, corresponding to the specified heading and frequency.
Roll RAO:	The RAO in roll, corresponding to the specified heading and frequency.
Pitch RAO:	The RAO in pitch, corresponding to the specified heading and frequency.
Heave Phase:	The phase angle corresponding to the heave RAO.
Surge Phase:	The phase angle corresponding to the surge RAO.
Sway Phase:	The phase angle corresponding to the sway RAO.
Yaw Phase:	The phase angle corresponding to the yaw RAO.
Roll Phase:	The phase angle corresponding to the roll RAO.

Pitch Phase:	The phase angle corresponding to the pitch RAO.
Position:	The position of the floating body reference point to be used in calculating the wave forces. The options are *Original* (the default), *Instantaneous* or *Mean*. If you specify *Mean*, a rolling mean will be used, based on the average position of the floating body reference point over the previous *No. of Timesteps*. See Note (d).
No. of Timesteps:	The number of time steps to be used in computing the mean positions of the floating body reference point. This input is only relevant if *Position* is set to *Mean*.
Wavelength Phase Shift:	The options are *Include* (the default) and *Exclude*. See Note (e).

Table Input - Line Layout

Input:	Description
Floating Body:	The name of the floating body.
Heading:	The wave heading in degrees.
Frequency:	The wave frequency in Hertz.
Heave RAO:	The RAO in heave, corresponding to the specified heading and frequency.
Surge RAO:	The RAO in surge, corresponding to the specified heading and frequency.
Sway RAO:	The RAO in sway, corresponding to the specified heading and frequency.

Yaw RAO:	The RAO in yaw, corresponding to the specified heading and frequency.
Roll RAO:	The RAO in roll, corresponding to the specified heading and frequency.
Pitch RAO:	The RAO in pitch, corresponding to the specified heading and frequency.
Heave Phase:	The phase angle corresponding to the heave RAO.
Surge Phase:	The phase angle corresponding to the surge RAO.
Sway Phase:	The phase angle corresponding to the sway RAO.
Yaw Phase:	The phase angle corresponding to the yaw RAO.
Roll Phase:	The phase angle corresponding to the roll RAO.
Pitch Phase:	The phase angle corresponding to the pitch RAO.
Position:	The position of the floating body reference point to be used in calculating the wave forces. The options are *Original* (the default), *Instantaneous* or *Mean*. If you specify *Mean*, a rolling mean will be used, based on the average position of the floating body reference point over the previous *No. of Timesteps*. See Note (d).
No. of Timesteps:	The number of time steps to be used in computing the mean positions of the floating body reference point. This input is only relevant if *Position* is set to *Mean*.

Wavelength Phase Shift:	The options are *Include* (the default) and *Exclude*. See Note (e).

NOTES

(a) Flexcom supports two different schemes for laying out the data in the RAO file, namely the *MCS* layout and the *Line* layout. This former is the standard layout traditionally used by Flexcom. The *Line* layout is a more recent addition to the software, and it represents a more general layout that, for example, simplifies copying and pasting RAO data from spreadsheet programs.

(b) Flexcom uses linear interpolation to calculate RAOs and phase angles at wave headings and frequencies intermediate to those in the RAO file. Outside of the range of user-specified headings and frequencies, RAOs and phase angles are assumed to be zero, so it is important to ensure you cover the full range of conditions likely to be encountered in an analysis when inputting the RAO data.

(c) Wave heading is defined as the angle between the direction of approach of a wave harmonic incident on the floating body and the local surge axis (refer to Definition of Wave Heading used with RAO Data).

(d) The reference point position should be set to *Original* in the current version of Flexcom.

- Using the instantaneous position can introduce higher order effects which are undesirable and inconsistent with the radiation-diffraction potential flow theory. Experience suggests that it can lead to excessive mean surge motions. So using the original position is preferable.

- One complication to be aware of, however, is that in random sea analyses, the platform may exhibit both high and low frequency movements (assuming it is subject to low frequency second-order forces). Strictly speaking, the first order forces should be based on the instantaneous position based on the second order response only (i.e., excluding first order effects), however it is not trivial to distinguish the instantaneous first and second order response in a time domain analysis (we plan to introduce a low-pass filtering system in a future version). For this reason you are advised to use the original position for the computation of first order forces, except when second order response is likely to be significant in which case a sensitivity to using the instantaneous position should be performed.

(e) When computing the phase angle of the applied forces, Flexcom automatically accounts for the physical separation between the floating body and the global origin. So in addition to the RAO phase angles, an addition 'ks' phase shift term is added, where k is the wave number of the regular wave (or wave harmonic in random seas), and s is the horizontal distance from the global X axis to the RAO reference point in the direction of wave propagation. Refer to First Order Wave Loads for further details. If you would like to suppress this additional phase shift term, for example if you believe it has already been accounted for during the derivation of the RAO phase angles, then you can use the *Exclude* option here.

1.8.9.18 *FRICTION

PURPOSE

To specify seabed friction stiffnesses.

THEORY

Refer to Motion in the Plane of the Seabed for further information on this feature.

KEYWORD FORMAT

One line containing the friction stiffness values:

```
Longitudinal Stiffness, Transverse Stiffness
```

This keyword is only relevant for frequency domain dynamic analyses.

TABLE INPUT

Input:	Description
Longitudinal Friction Stiffness:	The value of the longitudinal friction stiffness. This defaults to a value of 0.
Transverse Friction Stiffness:	The value of the transverse friction stiffness. This defaults to a value of 0.

NOTES

(a) This table is only relevant for a frequency domain dynamic analysis, where the initial static analysis has at least one non-zero seabed friction coefficient.

(b) You can model friction in a frequency domain dynamic analysis using either a total restraint, or a partial restraint modelled using a spring stiffness. For more information, please refer to Motion in the Plane of the Seabed.

(c) By default, seabed friction continues to be modelled in the same way as in earlier program versions (the so-called *Fully Restrained Model*). This corresponds to the situation where both stiffness entries in this table are zero.

(d) To apply a partial restraint in either the longitudinal or transverse direction, you input a non-zero stiffness in this table in the appropriate direction.

(e) Note however that a non-zero value is immaterial and unused if the corresponding friction coefficient was zero in your initial static analysis. In that case, free motion in the appropriate direction is permitted.

1.8.9.19 *INFLOWWIND

PURPOSE

To provide a link between Flexcom and FAST's wind-inflow data processing module InflowWind in a wind turbine simulation.

THEORY

Refer to InflowWind Overview for further information on this feature.

KEYWORD FORMAT

A line which provides the ability to couple InflowWind to Flexcom:

```
INPUT FILE=Path to InflowWind input data file (.dat)
```

TABLE INPUT

Input:	Description
Input File:	The path to the InflowWind control data file. See the Wind Component for additional details.

1.8.9.20 *INTEGRATION

PURPOSE

To specify the number of integration points to be used in the Gaussian quadrature of element mass and stiffness matrices and load vectors.

THEORY

Refer to Gaussian Quadrature for further information on this feature.

KEYWORD FORMAT

One line containing the number of integration points:

```
Number of Integration Points
```

The possible values that can be used are 2, 3, 5 and 10. The default value is 3.

TABLE INPUT

Input:	Description
Number of Integration Points:	The number of integration points for Gaussian quadrature. The default value is 3. Other valid values are 2, 5, and 10.

NOTES

(a) The default number of integration points is sufficient for the vast majority of analyses, and this option should be very rarely invoked.

1.8.9.21 *INTERNAL FLUID

PURPOSE

To define the properties of an internal fluid.

THEORY

Refer to Internal Fluid for further information on this feature.

KEYWORD FORMAT

A block of two lines repeated as many times as necessary.

```
SET=Set Name
Level Above Mudline, Mass Density, [Internal Pressure],
[Velocity],
[CORIOLIS=INCLUDE/EXCLUDE],
[CENTRIFUGAL=INCLUDE/EXCLUDE],
[AXIAL_INERTIA=INCLUDE/EXCLUDE],
[LATERAL_INERTIA=INCLUDE/EXCLUDE],
[DYNAMIC_PRESSURE=INCLUDE/EXCLUDE]
```

Internal Pressure and *Velocity* default to 0. The remaining terms – CORIOLIS, CENTRIFUGAL, AXIAL_INERTIA, LATERAL_INERTIA and DYNAMIC_PRESSURE – all default to INCLUDE.

TABLE INPUT

Input:	Description
Set Name:	The element set name. The default is all elements.
Level Above Mudline:	The level above the mudline to which the internal fluid extends. This affects element weights (in terms of whether an element is fluid filled or empty), and element internal pressures (via the internal fluid hydrostatic pressure contribution). See note (c).
Mass Density:	The mass density (mass per unit volume) of this internal fluid.
Internal Pressure:	The internal fluid constant pressure above hydrostatic. This defaults to a value of 0. See Note (e).
Velocity:	The internal fluid velocity. This defaults to a value of 0. See Notes (f) and (g).
Coriolis Force:	This is related to the internal fluid *Velocity*. Specifically, *Coriolis Force* is used to specify whether a Coriolis force term due to the velocity of the internal fluid is to be included or excluded from the equations of motion. The default is *Include*, indicating that the Coriolis force term is to be included. See Note (g).
Centrifugal Force:	This is related to the internal fluid *Velocity*. Specifically, *Centrifugal Force* is used to specify whether a centrifugal force term due to the velocity of the internal fluid is to be included or excluded from the equations of motion. The default is *Include*, indicating that the centrifugal force term is to be included. See Note (g).
Axial Inertia:	This option allows you to specify whether internal fluid contributes to the structure axial inertia in a dynamic analysis. The default is *Include*, indicating that the internal fluid contributes to axial inertia. See Note (h).

Lateral Inertia:	This option allows you to specify whether internal fluid contributes to the structure lateral inertia in a dynamic analysis. The default is *Include*, indicating that the internal fluid contributes to lateral inertia. See Note (i).
Dynamic Pressure:	This is related to the internal fluid *Velocity*. Specifically, *Dynamic Pressure* is used to specify whether a dynamic pressure term due to the velocity of the internal fluid is to be included or excluded from the equations of motion. The default is *Include*, indicating that the dynamic pressure term is to be included. See Notes (g).

NOTES

(a) The specification of internal fluid is optional. By default the structure is assumed empty, that is, filled with air.

(b) The element set name is associated with the elements comprising the set using the *Element Sets* table. Obviously different internal fluids can be defined for different element sets.

(c) The term *Level Above Mudline* is most meaningful for riser analysis. For example, a production riser is typically filled with oil, whereas a drilling riser contains mud. In such cases, the elevation input intuitively corresponds to the global X coordinate value at the uppermost end of the riser. Caution must be exercised in other situations however, such as pipeline analysis, where the structure is predominately flat. Theoretically, the elevation input should correspond exactly to the centreline of the pipeline, or marginally above it. In practice, it is strongly recommended that the elevation term is augmented to include a margin of safety. This is necessary to ensure that the pipeline remains flooded at all times throughout a simulation, irrespective of any local fluctuations in vertical displacement. Without an augmented elevation specification, certain portions of the pipeline could inadvertently be modelled as being empty, particularly in restart analyses (due to local deformations in the preceding analysis stage). If you have any concerns about pipeline flooding, you are advised to examine the input data echo section of the detailed output file.

(d) The internal diameter input with the geometric properties is used with the inputs here, to calculate the effect of the internal fluid.

(e) Refer to Hydrostatic Pressure for a discussion on how the hydrostatic pressure head due to the presence of internal fluid and slugs is modelled in Flexcom.

(f) The Flexcom internal fluid model caters for (i) uniform steady state internal flow, and (ii) multi-phase slug flow (the slug flow capability has a separate associated table *Slugs*). The uniform steady state internal fluid flow option is invoked by inputting a non-zero *Velocity* here.

(g) Steady state internal fluid flow induces both a centrifugal force (related to pipe curvature) and a Coriolis force (related to pipe rotation). Refer to Centrifugal Force and Coriolis Force for further details. An additional dynamic pressure term which affects pipe wall tension is also modelled. Refer to Dynamic Pressure for further details. Options are provided, as noted above, to suppress any of these terms in a particular analysis, allowing the relative significance of each term to be assessed.

(h) This modelling capability is specifically intended for the analysis of drilling risers in emergency disconnect mode (riser hangoff analysis). Refer to Inertial Effects for further details.

(i) The *Lateral Inertia* option operates in a similar manner to the *Axial Inertia* option, as it allows you to exclude internal fluid from the riser lateral inertia. Refer to Inertial Effects for further details.

1.8.9.22 *LOAD

PURPOSE

To define arbitrary loading.

THEORY

Refer to Point and Distributed Loads and User-Subroutine Loads for further information on this feature.

KEYWORD FORMAT

Load data is specified in blocks, with each block beginning with a TYPE= line defining the load type. This is then followed by as many lines as necessary to specify the loads of that type, and this in turn is then followed as necessary by data for other types.

Block defining point loads. The format of the second and subsequent lines depends on whether the loading is based on a single numerical value or a file containing a range of time-dependent values.

```
TYPE=CONSTANT
Node (Number or Label), DOF, Load Value
```

or

```
Node (Number or Label), DOF, FILE=File Name, Column Number
```

Block defining distributed loads. The format of the second and subsequent lines depends on whether the loading is based on a single numerical value or a file containing a range of time-dependent values.

```
TYPE=DISTRIBUTED
Start Element (Number or Label), End Element (Number or Label), Co-ordinate I
```

or

```
Start Element (Number or Label), End Element (Number or Label),
Co-ordinate Direction, FILE=File Name, Column Number
```

Line defining a user subroutine:

```
USER=File Name
```

Block defining point loads specified in a user subroutine:

```
TYPE=SUBROUTINE, LOAD=POINT
Node (Number or Label), DOF
```

Block defining distributed loads specified in a user subroutine:

```
TYPE=SUBROUTINE, LOAD=DISTRIBUTED
Start Element (Number or Label), End Element (Number or Label), Co-ordinate I
```

Block defining vessel RAO loads:

```
TYPE=VESSEL
Node (Number or Label), DOF
```

Block defining harmonic loads:

```
TYPE=HARMONIC
Node (Number or Label), DOF, Amplitude [, Phase]
```

If any user subroutine loads are defined, then the USER=option is mandatory. File Name should include the entire path of the file including its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks. If you specify a node/element label rather than a number, it must be enclosed in {} brackets. The user subroutine option is not relevant for frequency domain dynamic analysis. Vessel and harmonic loads are only relevant for frequency domain dynamic analysis.

CONSTANT LOADS

Purpose

To define the location, direction and magnitude of constant (time invariant) loads.

Table Input (Point Loads – Constant)

Input:	Description
Node:	The node (number or label) at which the point load is applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global DOF in which this load acts. Specify a value of 1 for the global X-direction, 2 for the global Y-direction or 3 for the global Z-direction. Degrees of freedom 4, 5 or 6 refer to constant point moments at the node.
Load Value:	The magnitude of the point load or moment. A negative value indicates a load in the negative axis direction.

Table Input (Distributed Loads – Constant)

Input:	Description
Start Element:	The element (number or label) at the start of the section over which the distributed load is applied. If you specify an element label rather than an element number, it must be enclosed in {} brackets.

End Element:	The element (number or label) at the end of the section over which the distributed load is applied. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Coordinate Direction:	The direction in the global coordinate system in which this load acts. Specify a value of 1 for the global X-direction, 2 for the global Y-direction or 3 for the global Z-direction.
Load Value:	The magnitude of the distributed load in terms of force per unit length. A negative value indicates a load in the negative axis direction.

Notes

(a) Loads are input as components in the global coordinate directions. A load oblique to the coordinate axes is specified as three load components.

(b) Uniformly distributed loads are specified as acting on an element or group of consecutive elements, defined in terms of a start element and an end element (start and end elements are the same for a load acting on a single element).

TIME VARYING LOADS

Purpose

To define the location, direction and magnitude of time-dependent loads.

Table Input (Point Loads – Time Varying)

Input:	Description
Node:	The node (number or label) at which the point load is applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global DOF in which this load acts. Specify a value of 1 for the global X-direction, 2 for the global Y-direction or 3 for the global Z-direction. Degrees of freedom 4, 5 or 6 refer to constant point moments at the node.

File Name:	The name of the ASCII file containing the time-dependent load data.
Column Number:	The column of data in the external file which contains the load data. Force columns are labelled upwards from one (i.e. the time column is considered 'column zero').

Table Input (Distributed Loads – Time Varying)

Input:	Description
Start Element:	The element (number or label) at the start of the section over which the distributed load is applied. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
End Element:	The element (number or label) at the end of the section over which the distributed load is applied. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Coordinate Direction:	The direction in the global coordinate system in which this load acts. Specify a value of 1 for the global X-direction, 2 for the global Y-direction or 3 for the global Z-direction.
File Name:	The name of the ASCII file containing the time-dependent load data.
Column Number:	The column of data in the external file which contains the load data. Force columns are labelled upwards from one (i.e. the time column is considered 'column zero').

Notes

(a) Loads are input as components in the global coordinate directions. A load oblique to the coordinate axes is specified as three load components.

(b) Uniformly distributed loads are specified as acting on an element or group of consecutive elements, defined in terms of a start element and an end element (start and end elements are the same for a load acting on a single element).

(c) The data file is ASCII based, and its format depends on the number of loads which are specified in this manner. In general the file contains one column of data for the time and an additional column for each load definition. The layout of data within the text file must be consistent with the definition of loads in the keyword file, the latter being governed by the *Column Number* parameter. Force columns are labelled upwards from one (i.e. the time column is considered 'column zero').

Columns of data in the ASCII data file should be separated by blank spaces or tabs. Comment lines, denoted by a capital 'C' in the first column, are permitted, while lines that are completely blank are ignored. You can specify several time-varying loads in a single data file.

(d) When a time varying load is specified, Flexcom reads the relevant force data from the specified data file at the relevant time. The program uses linear interpolation to obtain force values at solution times intermediate to those specified in the data file. For solution times before the earliest time specified in the load file, Flexcom uses the earliest force values available. Similarly, for solution times after the latest time in the load file, Flexcom continues to use the values corresponding to the latest available time.

SUBROUTINE LOADS

Purpose

To define the point/area of application and direction of loads, the magnitude of which is defined, typically as a function of time, in a user-subroutine.

Table Input (Point Loads)

Input:	Description
Node:	The node (number or label) at which the point load is applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global DOF in which this load acts. Specify a value of 1 for the global X-direction, 2 for the global Y-direction or 3 for the global Z-direction. Degrees of freedom 4, 5 or 6 refer to constant point moments at the node.

Table Input (Distributed Loads)

Input:	Description
Start Element:	The element (number or label) at the start of the section over which the distributed load is applied. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
End Element:	The element (number or label) at the end of the section over which the distributed load is applied. If you specify en element label rather than an element number, it must be enclosed in {} brackets.
Coordinate Direction:	The direction in the global coordinate system in which this load acts. Specify a value of 1 for the global X-direction, 2 for the global Y-direction or 3 for the global Z-direction.

Table Input (Subroutine File)

Input:	Description
Name of Subroutine File:	The name of the DLL file containing the load user-subroutine.

Notes

(a) Refer to User-Subroutine Loads for a detailed discussion of the load user-subroutine facility.

(b) Loads are input as components in the global coordinate directions. This means that a force oblique to the coordinate axes is specified as three load components.

(c) The name of the DLL file containing the subroutine that defines the magnitude of the load specified in the *Subroutine Loads – Point* or *Subroutine Loads – Distributed* table must be specified using the *Subroutine File* table.

(d) Uniformly distributed loads are specified as acting on an element or group of consecutive elements, defined in terms of a start element and an end element (start and end elements are the same for a load acting on a single element).

(e) Subroutine loads are not relevant for frequency domain dynamic analysis.

VESSEL LOADS

Purpose

To specify the point of application and direction of point loads computed from vessel RAOs.

Table Input

Input:	Description
Node:	The node (number or label) at which the point load is applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global DOF in which this load acts. Specify a value of 1 for the global X-direction, 2 for the global Y-direction and 3 for the global Z-direction. Degrees of freedom 4, 5, and 6 refer to point moments at the node

Notes

(a) Vessel loads are used to specify frequency-dependent time-varying loads, for example, the variation in top tension of a riser connected to a TLP.

(b) The analysis type must be frequency domain dynamic if vessel loads are to be specified. The name of the file containing the load RAO must be specified in the *Loads - RAO File* table.

(c) Vessel loads can be used with constant point loads. In this case, the constant load (applied at all frequencies) is added to the vessel load at each particular frequency.

HARMONIC LOADS

Purpose

To specify the parameters relating to harmonically varying point loads in a frequency domain regular wave analysis.

Table Input

Input:	Description
Node:	The node (number or label) at which the point load is applied. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global DOF in which this load acts. Specify a value of 1 for the global X-direction, 2 for the global Y-direction and 3 for the global Z-direction. Degrees of freedom 4, 5, and 6 refer to harmonic point moments at the node.
Amplitude:	The amplitude of the point load.
Phase:	The phase difference relative to the regular wave, input in degrees. The phase lag or lead is relative to the wave at the mean offset position of the node in question. A phase lag is indicated by a positive phase angle and a phase lead by a negative phase angle. The default is 0°.

Notes

(a) Harmonic point loads are input as components in the global coordinate directions. This means that a force oblique to the coordinate axes is specified as three load components.

(b) The analysis type must be frequency domain dynamic if harmonic loads are to be specified. If you specify such loads, then the analysis must have one or many regular waves invoked. The harmonic load has the same period as the analysis regular wave(s).

1.8.9.23 *MOMENTS

PURPOSE

To specify the value of Molin’s yaw coefficient for a moored vessel, and also to specify the fractions of Molin’s Moment and Munk’s Moment that are applied to the moored vessel.

THEORY

Refer to Current Loads for further information on this feature.

KEYWORD FORMAT

One line defining the moment coefficients:

```
Molin’s Coefficient, Molin’s Yaw Moment Coefficient, Munk’s Moment Coefficier
```

The keyword is only relevant to mooring analyses.

TABLE INPUT

Input:	Description
Molin’s Coefficient:	Molin’s yaw coefficient.
Molin’s Yaw Moment Factor:	A number between 0 and 1 specifying the fraction of the Molin’s yaw Moment which is added to the current loads on the vessel. The default is 0.
Munk’s Moment Factor:	A number between 0 and 1 specifying the fraction of the Munk’s Moment which is subtracted from the current loads on the vessel. The default is 0.

1.8.9.24 *NAME

PURPOSE

To specify a title for the Flexcom analysis run.

THEORY

Refer to Static Analysis, Time Domain Analysis, Frequency Domain Analysis and Quasi-Static Analysis for further information on this feature.

KEYWORD FORMAT

A single line containing the analysis name.

```
Name
```

TABLE INPUT

Input:	Description
Title:	A descriptive title to be associated with the Flexcom analysis. This entry is optional.

1.8.9.25 *NO FINAL STATIC

PURPOSE

To suppress the final static analysis step that is typically performed automatically by Flexcom following a frequency domain dynamic analysis.

THEORY

Refer to Static Solution for further information on this feature.

KEYWORD FORMAT

This keyword does not contain any further data.

TABLE INPUT

Input:	Description
Suppress Final Static Analysis:	The options are *No* (the default) and *Yes*.

NOTES

(a) The default is to *No*. This is the recommended setting for the majority of analyses.

(b) To suppress the final static analysis step, select *Yes*. In certain rare circumstances where it proves difficult to achieve a converged solution (notwithstanding the effect of varying the solution parameters), then you may wish to suppress the final static analysis step. This means that the current forces may not be entirely accurate, but invoking this option may represent a reasonable compromise. Further details are provided in Static Solution, but it suffices to say that the linearised drag forces due to current loading cannot be evaluated with complete accuracy until after the dynamic response has been found.

(c) This option is only relevant for frequency domain dynamic analysis.

1.8.9.26 *NO FRICTION

PURPOSE

To suppress the effects of friction.

THEORY

Refer to Seabed Friction and Guide Surface Friction for further information on this feature.

KEYWORD FORMAT

This keyword does not contain any further data.

TABLE INPUT

Input:	Description
Suppress Friction:	The options are No (the default) and Yes.

NOTES

(a) The default is to *No*. This is the recommended setting for the majority of analyses.

(b) To suppress the effects of friction, select *Yes*. Very occasionally, you may want to exclude seabed friction, for example, during the initiation of pipeline/seabed contact in a pipeline analysis.

(c) This option is only relevant where frictional effects are being modelled. For example, if a seabed or guide surface is present, and at least one non-zero friction coefficient has been specified.

(d) This option is not relevant for a frequency domain dynamic analysis.

1.8.9.27 *NO HYSTERESIS

PURPOSE

To suppress bending hysteresis effects.

THEORY

Refer to Hysteretic Bending for further information on this feature.

KEYWORD FORMAT

This keyword does not contain any further data.

TABLE INPUT

Input:	Description
Suppress Hysteresis:	The options are *No* (the default) and *Yes*.

NOTES

(a) Hysteresis effects are included by default in all analysis where hysteresis is defined. To suppress the effect of bending hysteresis in an analysis select the Yes option.

(b) The option is provided on the basis that a flexible riser needs to be pressurised before the hysteresis bending can occur, so the riser is typically compliant until fully pressurised (e.g. during installation).

1.8.9.28 *NO PIP SLIDING

PURPOSE

To disable the interchangeable nature of sliding pipe-in-pipe connections.

THEORY

Refer to Sliding Connections for further information on this feature.

KEYWORD FORMAT

This keyword does not contain any further data.

TABLE INPUT

Input:	Description
Disable Pipe-in-Pipe Sliding:	The options are *No* (the default) and *Yes*.

NOTES

(a) The default is to *No*. This is the recommended setting for the majority of analyses.

(b) To disable the interchangeable nature of sliding pipe-in-pipe connections, select *Yes*. In certain circumstances, it may be desirable to allow the software to initially determine appropriate connections between an inner and outer pipe, and to subsequently treat these connections as Standard Connections. For example, when setting up a work-over riser model, the initial set-down of the inner tubing can make it very difficult to manually identify the required nodal connections in advance of the initial static analysis. Designating the connections as *Sliding* allows the program to automatically determine the optimal set of pipe-in-pipe connections, minimising effort on the part of the user. While there may be significant relative axial motion between the initial model definition and the converged static solution, there is comparatively little axial motion during the actual simulation itself (e.g. when the model is subjected to wave loading). Invoking the `*NO PIP SLIDING` keyword in subsequent restart analyses thereby ensures computational efficiency (any overhead associated with the monitoring of nodal locations is eliminated), and can also provide enhanced numerical stability (connectivity of the finite element model remains consistent throughout the simulation).

1.8.9.29 *NONLINEAR MODEL

PURPOSE

To specify a modelling approach for non-linear materials.

THEORY

Refer to Tangent and Secant Stiffness and Curvature Slippage for further information on this feature.

KEYWORD FORMAT

A single line specifying the modelling option.

```
METHOD=Method
```

Method can be either TANGENT or SECANT. Whichever option is invoked carries through to any or all subsequent restarts, until it is changed again. If the keyword is never invoked, the tangent method is used by default.

An optional line to adjust material properties for an element set.

```
EI=EI Type [, NONLINEAR=Nonlinear type] [, SET=Set Name]
```

This line may be repeated as often as required to redefine element properties.

EI Type can be LINEAR or NONLINEAR (the default). *Nonlinear type* can be ORIGINAL (the default) or CURVATURE_SLIPPAGE. *Set Name* defaults to ALL if omitted.

Whichever options are invoked, they carry through to any or all subsequent restarts, until subsequently altered.

TABLE INPUT

Non-linear Model – Solution Method

Input:	Description
Method:	The options are *Tangent* (the default) and *Secant*. See Note (a).

Non-linear Model – Bending Stiffness

Input:	Description

Bending Stiffness:	The type of bending stiffness to be used. The options are *Linear* (the default) and *Nonlinear*. See Note (b).
Non-linear Data:	The type of non-linear moment-curvature relationship to be used. The options are *Original* (the default) and *Curvature Slippage*. See Notes (c) and (d).
Set Name:	The element set name. The default is all elements.

NOTES

(a) This option relates to non-linear structural properties, whether defined in the *Flexible Riser* or the *Rigid Riser* format, and non-linear elements. Under normal circumstances, both methods should produce exactly the same results, but the secant method may provide additional robustness occasionally. For example, the tangent method generally requires that the slope should either monotonically increase or monotonically decrease with increasing displacement. Otherwise, there may be more than one location along the nonlinear relationship which results in the same restoring force, and this may contribute to solution instability. The tangent method is still retained as the default option, as the secant method should, in theory at least, typically require a slightly larger number of iterations for solution convergence. Refer to Tangent and Secant Stiffness for further information.

(b) When the *Linear* option is invoked, Flexcom will use the linear bending stiffness specified under the `*GEOMETRIC SETS` keyword. When the *Nonlinear* option is invoked, Flexcom will use the non-linear moment-curvature relationship defined under the `*MOMENT-CURVATURE` keyword, and referenced under the `*GEOMETRIC SETS` keyword. The non-linear relationship may be consistent with the original data (see Note (c)), or modified automatically by the program (see Note (d)).

(c) If the *Original* option is specified, Flexcom will simply apply the (unmodified) user-defined moment-curvature relationship, as defined under the `*MOMENT-CURVATURE` keyword, to the specified element set.

(d) If the *Curvature Slippage* option is specified, Flexcom will automatically modify the user-defined moment-curvature relationship, as defined under the `*MOMENT-CURVATURE` keyword, and apply the modified data to the specified element set. Refer to Curvature Slippage for further information on this feature. The modification procedure may be summarised as follows.

- The program ascertains the bending moment and curvature at the element mid-point, at the end of the preceding analysis stage. These governing values, typically derived from a preceding static analysis, are termed M_s and K_s, respectively.

- For the purposes of determining the moment from a given curvature, the curvature is first adjusted, by subtracting the governing K_s value. The moment is then obtained from the non-linear moment-curvature relationship, and subsequently translated by the slip moment M_s.

- Note also that the non-linear relationship is mirrored about the $\{K_s, M_s\}$ slip point, and that all points on the original non-linear moment-curvature relationship should be zero or positive (i.e. the original curve should be defined in the first quadrant).

1.8.9.30 *NONLINEAR STATIC

PURPOSE

To specify solution parameters used in the final static analysis performed automatically by Flexcom following a frequency domain dynamic analysis.

THEORY

Refer to Static Solution for further information on this feature.

KEYWORD FORMAT

One line with the appropriate solution parameters, all of which are optional:

```
[Number of Steps] [, Required Tolerance Measure] [, Maximum Number of Iterati
```

This keyword is only relevant for frequency domain dynamic analysis. *Number of Steps* defaults to 1. *Required Tolerance Measure* defaults to 0.001. *Maximum Number of Iterations* defaults to 20.

TABLE INPUT

Input:	Description

Number of Steps:	The number of analysis steps used in the final static analysis phase of the overall solution. This defaults to a value of 1.
Required Tolerance Measure:	The tolerance used in determining if convergence has been achieved between successive iterations. The default value is 0.001 (0.1%).
Maximum Number of Iterations:	The maximum number of iterations that may be performed. The default number of iterations is 20.

NOTES

(a) The default values are adequate in the vast majority of cases and this option is rarely invoked.

(b) If the specified number of steps is 1, then the static forces comprising the load vector are applied initially at their full magnitudes, and the solution is completed in one step. If the *Number of Steps* is specified as greater than 1, then the static loads are increased up to their maximum values over the specified number of steps.

(c) This table is only relevant for frequency domain dynamic analysis

1.8.9.31 *OFFSET

PURPOSE

To specify an offset of an attached vessel from its initial position.

THEORY

Refer to Vessel Offsets for further information on this feature.

KEYWORD FORMAT

A block of two lines defining the offset, repeated as many times as necessary:

```
VESSEL=Vessel Name
OFFSET=X Offset [, Y Offset] [, Z Offset] [, Yaw Offset] [, Roll Offset] [, P
```

Data for any `Vessel Name` specified here must be input under `*VESSEL`.

TABLE INPUT

Input:	Description
Vessel:	The name of the attached vessel.
X:	The magnitude of the vessel offset in the global X-direction.
Y:	The magnitude of the vessel offset in the global Y-direction. This entry is optional and defaults to zero if omitted.
Z:	The magnitude of the vessel offset in the global Z-direction. This entry is optional and defaults to zero if omitted.
Yaw	The vessel rotational offset about the vertical or global X direction. This entry is optional and defaults to zero if omitted.
Roll	The rotational offset, in degrees, about a vessel local surge axis. This entry is optional and defaults to zero if omitted.
Pitch	The rotational offset, in degrees, about a vessel local sway axis. This entry is optional and defaults to zero if omitted.

1.8.9.32 *PRINT

PURPOSE

To request additional printed output to the main output file.

THEORY

Refer to Notes below for further information on this feature.

KEYWORD FORMAT

A block of lines to request additional printed output. All lines are optional, and may occur in any order. The first line shown below contains the start time, the time increment and end time at which subsequent data requests are to be output. The remaining lines allow you to request the output of various parameters.

```
[START=Start Time, INCREMENT=Time Increment, END=End Time]
[OUTPUT=AERODYNAMIC FORCES] [, BLADE=Blade Number] [, DOF=DOF Number]
[OUTPUT=AUXILIARY DATA]
[OUTPUT=CENTRIFUGAL FORCES] [, SET=Set Name]
[OUTPUT=CONVECTED ELEMENT AXES] [, SET=Set Name]
[OUTPUT=CRITERIA]
[OUTPUT=DETAILED ELEMENT OUTPUT][, SET=Set Name]
[OUTPUT=FREQUENCY DOMAIN SOLUTION]
[OUTPUT=PIP CONNECTIONS]
[OUTPUT=REYNOLDS COF OUTPUT]
[OUTPUT=SLUG FLOW][, SET=Set Name]
[OUTPUT=WATER PARTICLE HYDRODYNAMICS][, SET=Set Name]
[OUTPUT=WAVE DISCRETISATION]
[OUTPUT=PLASTIC HARDENING INTEGRATION]
```

Naturally the FREQUENCY DOMAIN SOLUTION option is only relevant for frequency domain dynamic analysis. The other output types are only relevant for time domain analysis.

Even in the case of time domain analyses, the start and end time inputs are not always relevant (e.g. in the case of wave discretisation data). *Start Time* defaults to the analysis start time. *Time Increment* defaults to the analysis time step (or suggested time step, in the case of Variable Time Stepping). *End Time* defaults to the analysis end time.

TABLE INPUT

Print Times

Input:	Description
Time of First Recording:	The time of first output. This defaults to the simulation start time.
Recording Interval:	The time interval in seconds between outputs. This defaults to the analysis time step (or suggested time step, in the case of Variable Time Stepping)
Time of Last Recording:	The time of last output. This defaults to the simulation end time.

Print Options

Input:	Description
Parameter:	The parameter for which the additional output is required.
Element Set:	The element set (if applicable) for which the printed output is required. This defaults to *All*, to indicate all elements.
Blade No:	Blade number can be 1, 2 or 3, referring to one of the blades noted in the `*TURBINE ROTOR` keyword.
DOF:	The global degree of freedom in which aerodynamic forces are requested.

NOTES

(a) **Aerodynamic Forces**. This option presents the aerodynamic forces acting on a particular blade as a function of time. Forces are provided at each blade node in the requested degree of freedom. Refer to Computational Methodology for further information on the wind loads coming from OpenFAST.

(b) **Auxiliary Data**. This option presents information relating to auxiliary nodes, elements and panels. Refer to Auxiliary Bodies for further details.

(c) **Centrifugal Forces**. This option presents the following information at each timestep. Refer to Centrifugal Force for further details.

- Node number
- Centrifugal force component in the x-direction
- Centrifugal force component in the y-direction
- Centrifugal force component in the z-direction
- Total centrifugal force

(d) **Convected Element Axes**. Flexcom uses a convected coordinate axes technique for modelling finite rotations in three dimensions. Each element of the finite element discretisation has a convected axis system associated with it, which moves with the element as it displaces in space. Refer to Finite Element Formulation for further details. The *Convected Element Axes* option presents the local convected axes for each element as a function of time.

(e) **Solution Criteria**. It is possible to specify certain solution criteria that need to be satisfied in a static analysis, and to define how the model is to be adjusted to satisfy the desired criteria. Refer to Solution Criteria Automation for further details. In such circumstances, it can be useful to request information about the model adjustments performed by the software, and the effect on the solution criteria of interest. The *Criteria* option presents the following information at each solution step.

- Solution iteration number
- Value of relevant criteria, at the current solution iteration
- Incremental model adjustment, at the current solution iteration
- Absolute model adjustment, at the current solution iteration
- Final value of relevant criteria, compared with target values

(f) **Detailed Element Output**. This option provides additional output beyond that provided by the standard post-processing methods. Specifically, it provides information regarding both bending moment and curvature at every integration point within each element of interest (standard post-processing provides information at element start, mid and end points only).

(g) **Frequency Domain Solution**. For frequency domain analyses, this option allows you to request a text based summary of the frequency domain solution. Refer to Frequency Domain Analysis for further details. For every element, the following parameters are presented in terms of mean, amplitude and phase of the dynamic response.

- Motions in DOFs 1-6
- Axial Force and Effective Tension
- Local Y and Local Z Shear Force
- Local Y, Local Z and Total Bending Moment

- Torque

(h) **PIP Connections**. For models which contain Pipe-in-Pipe Connections, it is sometimes useful to monitor these connections over time. For example, sliding connections are interchangeable, and are useful for modelling scenarios where there is significant relative axial motion between inner and outer pipes. The *Pipe-in-Pipe Connections* option facilitates a detailed inspection of the connected nodes at any point during the simulation, and provides greater transparency regarding the internal workings of the software. The following information is presented at each timestep.

- Connection number
- Connection type (standard or sliding)
- Primary node
- Secondary node
- Status (active or inactive - refers to sliding connections only)
- Primary element
- Secondary element
- Connection vector

(i) **Reynolds COF Output**. For analyses where you specify that hydrodynamic coefficients are to be computed as a function of instantaneous Re, this option allows you to create .COF files. This file contains a table showing the drag and inertia coefficients actually used by the program in calculating hydrodynamic forces at each integration point on each element. Refer to Reynolds Number Dependent Coefficients for further details.

(j) **Slug Flow**. This option presents the following information at each timestep. Refer to Slug Flow for further details.

- Element number
- Slot number (centred around integration points)
- Percentage filled with slug
- Slug velocity

- Slug density
- Slot length

(k) **Water Particle Hydrodynamics**. This option provides detailed output regarding the spatial and temporal distributions of water particle velocity and acceleration. Specifically, the following information is presented at all solution times of interest…

- Element number
- Slot number (centred around integration points)
- Global XYZ position
- Water particle velocity and acceleration in global X direction
- Water particle velocity and acceleration in global Y direction
- Water particle velocity and acceleration in global Z direction

Refer to Water Particle Velocities and Accelerations for further information on how these terms are defined for any point in the wave field.

(l) **Wave Discretisation**. This option presents information such as amplitude, period, direction and phase for each of the component harmonics in wave spectrum discretisation. Refer to Spectrum Discretisation for further details.

(m) **Plastic Hardening Integration**. This option is relevant for element sets which are associated a nonlinear plastic hardening model and only for *Static*, *Quasi-static* and *Dynamic* analyses. For plastic materials, the bending moment capacity may be updated by re-integrating the stress-strain curve over the elemental cross-section following significant changes in the axial forces. Refer to Linear Elastic with Plastic Hardening for further details. The Plastic Hardening Integration option here presents the updated axial force-strain and moment-curvature curves.

1.8.9.33 *QTF

PURPOSE

To specify Quadratic Transfer Functions (QTFs) that allow the slow drift loads on a floating body or moored vessel to be determined.

THEORY

Refer to Wave Drift Loads, QTF Coefficients and QTF Calibration Coefficients for further information on this feature.

KEYWORD FORMAT

Data may defined explicitly within the keyword file itself, or defined separately in an external file which is simply referenced using a FILE= entry. The latter approach is highly recommended as it reduces unnecessary clutter in the keyword file, and also helps to improve user interface performance and enhanced user experience.

Floating Body

A block of data consisting of floating body name followed by a single line defining the name of an external file which contains QTF data.

```
FLOATING BODY=Floating Body Name
FILE=File Name
```

File Name should include the entire path of the file including its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

A block of data consisting of floating body name, followed by a frequency value, followed by the QTFs for that frequency, with the QTF line repeated as necessary. The block is then itself repeated as necessary for further frequencies. The whole block may then be repeated to specify QTFs for different floating bodies.

```
FLOATING BODY=Floating Body Name
FREQ=Frequency
Floating Body/QTF Heading, Surge QTF Value, Sway QTF Value, Yaw QTF Value
```

Moored Vessel

A block of data consisting of a moored vessel tag line followed by a single line defining the name of an external file which contains current coefficient data.

```
MOORED VESSEL
FILE=File Name
```

File Name should include the entire path of the file including its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

A block of data consisting of a moored vessel tag line, followed by a frequency value, followed by the QTFs for that frequency, with the QTF line repeated as necessary. The block is then itself repeated as necessary for further frequencies.

```
MOORED VESSEL
FREQ=Frequency
Floating Body/QTF Heading, Surge QTF Value, Sway QTF Value, Yaw QTF Value
```

TABLE INPUT

Floating Body (External File)

Input:	Description
Floating Body:	The name of the floating body.
File Name:	The name of the external data file.

Floating Body

Input:	Description
Floating Body:	The name of the floating body.
Frequency:	The wave frequency in Hertz.
Heading:	The wave heading in degrees.
QTF_y:	The QTF coefficient in surge, corresponding to the specified frequency and heading.
QTF_z:	The QTF coefficient in sway, corresponding to the specified frequency and heading.
QTF_θ:	The QTF coefficient in yaw, corresponding to the specified frequency and heading.

Moored Vessel (External File)

Input:	Description
File Name:	The name of the external data file.

Moored Vessel

Input:	Description
Frequency:	The wave frequency in Hertz.
Heading:	The wave heading in degrees.
QTF_y:	The QTF coefficient in surge, corresponding to the specified frequency and heading.
QTF_z:	The QTF coefficient in sway, corresponding to the specified frequency and heading.
QTF_θ:	The QTF coefficient in yaw, corresponding to the specified frequency and heading.

NOTES

(a) To determine the relevant values of the QTFs for a given wave in a seastate, Flexcom first determines the relative heading for that wave. If the frequency of the wave matches any of the user-specified frequencies, the program then linearly interpolates between the closest user-specified headings for that frequency to find the relevant QTF data.

(b) If the frequency of the wave does not match any of the user-specified frequencies, Flexcom linearly interpolates between the closest user-specified headings and the closest user-specified frequencies to find QTF data for the particular wave heading and frequency.

(c) If either the frequency or heading of the particular wave falls outside the range of user-specified frequencies and headings, the value of the QTFs is assumed to be zero. Again, it is recommended that a sufficiently large range of frequencies and headings is specified to avoid this occurrence. The sign of the QTF values for a particular wave heading should reflect the direction of the force that a wave in that direction would exert on the body.

1.8.9.34 *QTF CALIBRATION FB

PURPOSE

To specify calibration coefficients used to scale the QTF coefficients for a floating body.

THEORY

Refer to Wave Drift Loads, QTF Coefficients and QTF Calibration Coefficients for further information on this feature.

KEYWORD FORMAT

Data may defined explicitly within the keyword file itself, or defined separately in an external file which is simply referenced using a FILE= entry. The latter approach is highly recommended as it reduces unnecessary clutter in the keyword file, and also helps to improve user interface performance and enhanced user experience.

A block of data consisting of floating body name followed by a single line defining the name of an external file which contains QTF calibration coefficients.

```
FLOATING BODY=Floating Body Name
FILE=File Name
```

File Name should include the entire path of the file including its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

A block of data consisting of floating body name followed by a single line repeated as often as necessary to define the QTF calibration coefficients over a sufficiently large range of headings. The block may be repeated for different floating bodies.

```
FLOATING BODY=Floating Body Name
Floating Body/QTF Heading, Surge QTF Calibration Value, Sway
QTF Calibration Value, Yaw QTF Calibration Value
```

TABLE INPUT

QTF Calibration Coefficients (External File)

Input:	Description
Floating Body:	The name of the floating body.
File Name:	The name of the external data file.

QTF Calibration Coefficients

Input:	Description
Floating Body:	The name of the floating body.
Heading:	The wave heading in degrees.
$QTFCAL_y$:	The QTF calibration coefficient in surge, corresponding to the specified heading.
$QTFCAL_z$:	The QTF calibration coefficient in sway, corresponding to the specified heading.
$QTFCAL_\theta$:	The QTF calibration coefficient in yaw, corresponding to the specified heading.

NOTES

(a) All QTF calibration coefficients default to 1.0 if unspecified.

(b) If QTF calibration coefficients are specified for a particular floating body at a particular heading, then QTF coefficients must also be defined for that same floating body at that same heading in the *Characteristics – QTF Coefficients* table.

(c) The headings for the QTF calibration coefficients need not necessarily be specified in any particular order (ascending order is not necessary).

1.8.9.35 *RAMP

PURPOSE

To specify a linear or non-linear variation in the ramp value applied to loads and displacements.

THEORY

Refer to Load Ramping for further information on this feature.

KEYWORD FORMAT

A single line which defines ramp type:

```
TYPE=LINEAR/NONLINEAR
```

TABLE INPUT

Input:	Description
Ramp Type:	This entry allows you to select the type of ramp to be used. The options are *Linear* and *Nonlinear* (the default). This entry is ignored in a static analysis.

NOTES

(a) The *Ramp Time* and *Ramp Type* entries allow control over the build-up of dynamic loads and displacements in an analysis with waves and/or vessel motions. For example, wave loads in a regular wave analysis are typically ramped on over 1 wave period, and the solution then proceeds for a further number of wave periods to achieve a steady state solution. When a Nonlinear ramp is specified (and this is the default), a half cosine ramp function is used (as opposed to a linear function).

1.8.9.36 *RAO

PURPOSE

To specify Response Amplitude Operators for a vessel.

THEORY

Refer to High Frequency RAO Motions for further information on this feature.

KEYWORD FORMAT

Block of data consisting of a vessel name followed by RAO data. The data may be explicitly defined, or imported from a file which is generated by an external program.

Explicitly Defined RAO Data

```
VESSEL=Vessel Name
[FORMAT=Format Name] [, LAYOUT=Layout Name]
Block of RAO Data
```

or

```
VESSEL=Vessel Name
[FORMAT=Format Name] [, LAYOUT=Layout Name]
FILE=File Name
```

The `Vessel Name` specified must be defined under the `*VESSEL` keyword. An optional line can be added on the second line, referencing an RAO conversion, and/or defining the layout of the RAO data. Data for any conversion format referenced here must be input under `*RAO FORMAT`. `Format Name` cannot be MCS, AQWA, WAMIT, MOSES or ORCAFLEX – these are predefined formats. `Layout Name` can be either MCS or LINE, defaulting to MCS if omitted. The next line can either specify a `File Name` (file path of a file containining the RAO data), or is the first of many lines defining RAO data in the keyword file directly.

Flexcom supports two different schemes for laying out RAO data, namely the MCS layout and the *Line* layout. The *MCS* layout begins with a single line defining the incident wave heading at which the RAOs are being defined. This is followed by a block of three lines specifying the incident wave frequency and the relevant RAO amplitudes and phases for this heading and wave frequency, as shown below. Note that entries shown below in italics should be replaced with their actual numeric values in the RAO file.

```
HEADING=Wave Heading
Wave Frequency
Heave RAO, Surge RAO, Sway RAO, Yaw RAO, Roll RAO, Pitch RAO
Heave Phase, Surge Phase, Sway Phase, Yaw Phase, Roll Phase, Pitch Phase
```

The block of three lines specifying the incident wave frequency, RAOs and phases are repeated as often as necessary until the RAOs are defined over the required range of wave frequencies. To define RAOs at more incident wave headings, you simply repeat the HEADING= line for the new wave heading, and the RAO data for this wave heading is specified as before. If RAOs are being defined for more than one incident wave heading, then they must also be defined for more than one frequency at each wave heading, and the number of frequencies at which the RAOs are defined is the same for each heading. If the RAOs are independent of the incident wave heading, then the HEADING= line should be omitted.

The `Line` layout is very similar to the standard MCS layout, with the exception that the RAO and phase data for a particular heading and frequency appear on a single line, the format of which is shown below. Note that, for clarity, the data shown below is split over a number of lines, but is in reality specified on a single line of the RAO file.

```
Wave Heading, Wave Frequency, Heave RAO, Heave Phase, Surge RAO, Surge Phase,
Sway RAO, Sway Phase, Yaw RAO, Yaw Phase, Roll RAO, Roll Phase, Pitch RAO,
Pitch Phase
```

This line is repeated for every wave frequency and every wave heading for which RAOs are being defined. As with the *MCS* layout, if RAOs are being defined at more than one incident wave heading, then they must also be defined at more than one frequency at each wave heading, and the number of frequencies at which the RAOs are defined is the same for each heading. So, for example, if RAOs were to be defined at three wave headings and ten frequencies per heading, then 30 lines of data would be required to specify the RAOs using the `Line` layout. If the same wave heading is specified on all lines, then Flexcom assumes that the RAOs are independent of wave heading. Note also that the order in which the lines of RAO data appear is not important. Flexcom automatically sorts the RAO data by heading and frequency.

This concludes the specification for explicitly specified RAO data. The entire block of data can then be repeated to specify force RAO data for subsequent vessels if required.

Imported RAO Data

```
VESSEL=Vessel Name
PROGRAM=Program Name
FILE=File Name
[UNITS=Unit Type]
```

The `Vessel Name` specified here must be defined under the `*VESSEL` keyword. The next two lines specify the name of the external program from which the RAO data originates and the path of the file containing the RAO data. `Program Name` may be AQWA, WAMIT, MOSES or ORCAFLEX. This is followed by an optional line which indicates the units used in this RAO file. `Unit Type` may be METRIC or IMPERIAL.

This concludes the specification for externally imported RAO data. The entire block of data can then be repeated to specify force RAO data for subsequent vessels if required.

TABLE INPUT

RAO Data – MCS Layout

Input:	Description
Vessel:	The name of the vessel.
Format:	The format of the RAO data, if an RAO conversion is being applied.
Heading:	The wave heading in degrees.
Frequency:	The wave frequency in Hertz.
Heave RAO:	The RAO in heave, corresponding to the specified heading and frequency.
Surge RAO:	The RAO in surge, corresponding to the specified heading and frequency.
Sway RAO:	The RAO in sway, corresponding to the specified heading and frequency.
Yaw RAO:	The RAO in yaw, corresponding to the specified heading and frequency.
Roll RAO:	The RAO in roll, corresponding to the specified heading and frequency.

Pitch RAO:	The RAO in pitch, corresponding to the specified heading and frequency.
Heave Phase:	The phase angle corresponding to the heave RAO.
Surge Phase:	The phase angle corresponding to the surge RAO.
Sway Phase:	The phase angle corresponding to the sway RAO.
Yaw Phase:	The phase angle corresponding to the yaw RAO.
Roll Phase:	The phase angle corresponding to the roll RAO.
Pitch Phase:	The phase angle corresponding to the pitch RAO.

RAO Data – Line Layout

Input:	Description
Vessel:	The name of the vessel.
Format:	The format of the RAO data, if an RAO conversion is being applied.
Heading:	The wave heading in degrees.
Frequency:	The wave frequency in Hertz.
Heave RAO:	The RAO in heave, corresponding to the specified heading and frequency.
Heave Phase:	The phase angle corresponding to the heave RAO.

Surge RAO:	The RAO in surge, corresponding to the specified heading and frequency.
Surge Phase:	The phase angle corresponding to the surge RAO.
Sway RAO:	The RAO in sway, corresponding to the specified heading and frequency.
Sway Phase:	The phase angle corresponding to the sway RAO.
Yaw RAO:	The RAO in yaw, corresponding to the specified heading and frequency.
Yaw Phase:	The phase angle corresponding to the yaw RAO.
Roll RAO:	The RAO in roll, corresponding to the specified heading and frequency.
Roll Phase:	The phase angle corresponding to the roll RAO.
Pitch RAO:	The RAO in pitch, corresponding to the specified heading and frequency.
Pitch Phase:	The phase angle corresponding to the pitch RAO.

RAO Data – External Import

Input:	Description
Vessel:	The name of the vessel.
Program:	The name of the external program.

File Name:	The file containing the RAO data.
Units:	The units used in the external RAO file. See Note (d).

NOTES

(a) Flexcom uses linear interpolation to calculate RAOs and phase angles at wave headings and frequencies intermediate to those in the RAO file. Outside of the range of user-specified headings and frequencies, RAOs and phase angles are assumed to be zero, so it is important to ensure you cover the full range of conditions likely to be encountered in an analysis when inputting the RAO data.

(b) Wave heading is defined as the angle between the direction of approach of a wave harmonic incident on the vessel and the local surge axis. However, please note that in strict mathematical terms the incident wave heading is the angle between the wave direction drawn at the vessel reference point and the negative direction of the local surge axis.

(c) Phase angles are specified in degrees, and represent a phase lag or lead relative to the wave at the vessel reference point. A positive phase angle denotes a phase lag relative to the incident wave harmonic.

(d) Flexcom will attempt to ascertain the unit system from the external file, but as this is not always possible, you are advised to explicitly state the unit system in order to avoid any possible ambiguity.

1.8.9.37 *RAO,LOAD

PURPOSE

To specify RAO load data for a frequency domain analysis.

THEORY

Refer to Frequency Domain Analysis for further information on this feature.

KEYWORD FORMAT

Blocks of lines defining the RAO load data. Each block contains three lines specifying the incident wave frequency and the relevant RAO amplitudes and phases for this wave frequency, as shown below. Note that entries shown below in italics should be replaced with their actual numeric values in the RAO file.

```
Wave Frequency
Heave RAO, Surge RAO, Sway RAO, Yaw RAO, Roll RAO, Pitch RAO
Heave Phase, Surge Phase, Sway Phase, Yaw Phase, Roll Phase, Pitch Phase
```

The block of three lines specifying the incident wave frequency, RAOs and phases are repeated as often as necessary until the RAOs are defined over the required range of wave frequencies.

This keyword is only relevant for frequency domain dynamic analysis.

TABLE INPUT

Input:	Description
Frequency:	The wave frequency in Hertz.
Heave RAO:	The RAO in heave, corresponding to the specified frequency.
Surge RAO:	The RAO in surge, corresponding to the specified frequency.
Sway RAO:	The RAO in sway, corresponding to the specified frequency.
Yaw RAO:	The RAO in yaw, corresponding to the specified frequency.
Roll RAO:	The RAO in roll, corresponding to the specified frequency.
Pitch RAO:	The RAO in pitch, corresponding to the specified frequency.
Heave Phase:	The phase angle corresponding to the heave RAO.

Surge Phase:	The phase angle corresponding to the surge RAO.
Sway Phase:	The phase angle corresponding to the sway RAO.
Yaw Phase:	The phase angle corresponding to the yaw RAO.
Roll Phase:	The phase angle corresponding to the roll RAO.
Pitch Phase:	The phase angle corresponding to the pitch RAO.

NOTES

(a) You must specify an RAO file name using this table if you identify vessel loads using the *Loads – Vessel* table.

(b) This table is only relevant for frequency domain analysis.

1.8.9.38 *REGULAR WAVE EQUIVALENT

PURPOSE

To specify that Flexcom is to replace a random wave spectrum or spectra by an equivalent regular wave or waves, and to calculate equivalent sinusoidal boundary conditions for attached nodes.

THEORY

Refer to Equivalent Regular Waves for further information on this feature.

KEYWORD FORMAT

One line of data specifying the wave period and height coefficients.

```
[Period Coefficient] [, Height Coefficient]
```

The keyword is only valid in time domain random sea analyses containing at least one vessel and one attached node. No sinusoidal boundary conditions should be specified.

Period Coefficient defaults to 0.95. *Height Coefficient* defaults to 1.86.

TABLE INPUT

Input:	Description
Factor for Wave Period:	The factor to use in calculating an equivalent regular wave period from the wave spectrum peak period. This defaults to a value of 0.95.
Factor for Wave Height	The factor to use in calculating an equivalent regular wave height from the wave spectrum significant wave height Hs. This defaults to a value of 1.86.

NOTES

(a) The random sea analysis must contain at least one vessel and one attached node.

(b) No sinusoidal boundary conditions should be specified when using this option.

1.8.9.39 *RESTART

PURPOSE

To indicate that an analysis is to be restarted from a previous run.

THEORY

Refer to Restart Analyses for further information on this feature.

KEYWORD FORMAT

A single line defining the restart type, file name and database option.

```
LAST=File Name [, TYPE=Restart Type] [, OPTION=APPEND]
```

Restart Type can be DYNAMIC/STATIC or CONTINUATION. OPTION=APPEND is appropriate only for a CONTINUATION analysis. *Restart Type* is set to DYNAMIC/STATIC by default. The only entry relevant for frequency domain dynamic analysis is *File Name*.

TABLE INPUT

Input:	Description
Type:	This option allows you to specify the restart type. There are three available types, New Loads or BCs (the default), *Continuation - New Database*, and *Continuation - Append*. See Note (b).
Restart File:	The name of the analysis from which the present run is to be restarted. This may be input in terms of the analysis name (e.g. *Example1-Static*) or the full path of the analysis (e.g. *C:\Flexcom\Example 1\Example1-Static*). See Note (d).

NOTES

(a) Most restarts involve the addition of new loads or BCs such as current or waves, and for this you use the *New Loads* or BCs restart type. Occasionally, a restart is performed which is simply a continuation of the previous run, and for this you can use either the Con*tinuation - New Database* or *Continuation - Append*. *Continuation - New Database* simply means that a new database is created, while *Continuation - Append* means that the results from the previous analysis are contained with the restart analysis results in the restart analysis database. An example here might be where a regular wave analysis is found not to have reached steady state, and must be continued for further wave periods.

(b) The only input which is relevant for frequency domain dynamic analysis is *Restart File*. The *Type* option pertains specifically to the time domain.

(c) If a file has spaces included in the file name i.e. (sample 1.keyx) then in order for the full filename i.e. “sample 1” to be consider for the restart it must be enclosed by inverted commas.

1.8.9.40 *SERVODYN

PURPOSE

To provide a link between Flexcom and FAST's wind turbine control module ServoDyn in a wind turbine simulation.

THEORY

Refer to Coupling between Flexcom and ServoDyn for further information on this feature.

KEYWORD FORMAT

A block of lines which provides the key pieces of information to couple ServoDyn to Flexcom:

```
TURBINE SET=Turbine Element Set Name
ROTOR INERTIA=Rotor Inertia
GEARBOX RATIO=Gearbox Ratio
INPUT FILE=Path to ServoDyn input data file (.dat)
```

TABLE INPUT

Input:	Description
Turbine Set:	The portion (set of elements) of the finite element model which represents the turbine assembly which will be yawed by the yaw controller. See Note (a)
Rotor Inertia:	Rotational inertia of the rotors about the low-speed shaft which is $\int r^2 dm$, the integral of the rotor (blades + hub + low-speed shaft) mass by the distance from the axis of rotation.
Gearbox Ratio:	The ratio of the high-speed to low-speed shaft speed. This value should be greater than zero and equal to unity for a direct-drive turbine.
Input File:	The path to the ServoDyn control data file. See the Turbine Component for additional details.

NOTES

(a) The turbine set should contain all elements contained in the nacelle assembly that are to be yawed about the tower vertical axis.

1.8.9.41 *SLUGS

PURPOSE

To specify parameters relating to slug flow.

THEORY

Refer to Slug Flow for further information on this feature.

KEYWORD FORMAT

Block of lines defining the slug data. The format used depends on whether the slugs have constant properties over the course of the analysis, or whether the slug properties vary as a function of time.

An optional line may be included at the beginning of the keyword to indicate if the visual display of slug elements in the structural animation is required or not.

```
[DISPLAY=Display]
```

For slugs with constant properties, the following block of three lines is used:

```
PROPERTIES=CONSTANT
SET=Set Name
Length, Density, [Velocity], Start Time, [No. of Slugs], [Time Lag]
[, COLOUR=Colour]
```

For slugs with time-varying properties, the following block of four lines is used:

```
PROPERTIES=TIME VARIANT
FILE=File Name
SET=Set Name
Length, Start Time, [No. of Slugs], [Time Lag], Column Number,
[VELOCITY=Velocity Profile] [, COLOUR=Colour]
```

The last line of the block may be repeated if more than one slug is present in a particular element set. The last two lines of the block may be repeated to define slugs in different element sets. When specifying time-varying slugs, several slugs may share the same ASCII file – otherwise a new file name should precede the element set specification. The entire block of lines may be repeated to combine both constant and time-varying slugs in the same analysis.

Display may be either YES (the default) or NO. The *No. of Slugs* defaults to 1. A value for *Time Lag* is only valid, and must be greater than 0, if *No. of Slugs* is greater than 1. *Velocity* defaults to the internal fluid velocity for the element set. If the *File Name* input contains spaces then it should be enclosed in double quotation marks. For time variant slugs, *Velocity Profile* may be either INTERPOLATED (the default) or CONSTANT.

SLUG ASCII FILE FORMAT

The format of the ASCII file which characterises time-varying slugs is as follows:

```
Time t1, Slug A head velocity at t1, Slug A tail velocity at t1, Slug A head
Time t2, Slug A head velocity at t2, Slug A tail velocity at t2, Slug A head
Time t3, Slug A head velocity at t3, Slug A tail velocity at t3, Slug A head
…
…
Time tn, Slug A head velocity at tn, Slug A tail velocity at tn, Slug A head
```

Further rows are added to define slug properties at additional times, from time t3 up to time tn.

Further columns may be added to define properties for additional slugs. Alternatively you may prefer to create a new ASCII file for each slug definition. If you have only one type of slug in the simulation, then you will only define properties for Slug A (i.e. there is no Slug B, or Slug C).

Refer to the Notes section for further information.

TABLE INPUT

Slug Display

Input:	Description
Display:	The options are *Yes* (the default) and *No*. See Note (f).

Slugs – Constant Properties

Use this input format to define slugs whose properties remain constant over time (i.e. their length, density and velocity do not change during a simulation)

Input:	Description
Set Name:	The element set name. The default is all elements.
Slug Length:	The length of the slug(s) being defined.
Slug Density:	The density of the slug(s) being defined.

Slug Velocity:	The velocity of the slug(s) being defined.
Start Time:	The time at which the slug (or the first of a series of similar slugs) enters the first element in the set.
No. of Slugs:	The number of slugs in a series with the present slug properties. This entry defaults to 1, indicating that a single slug rather than a series of slugs is being defined.
Time Lag:	This is an optional entry, only used when a series of slugs is being defined. Time Lag is the delay between the times that each slug in the series enters the first element of the set.
Colour:	This is an optional entry to specify the colour of the slug in the structural animation. The entry must be the name of a standard colour or a user defined colour. A list of the standard colours can be found in the `*COLOUR DEFINE` keyword.

Slugs – Time-Varying Properties

Input:	Description
File Name:	The name of the ASCII file containing the time-varying data. See Note (e).
Set Name:	The element set name. The default is all elements.
Slug Length:	The initial length of the slug(s) being defined.
Start Time:	The time at which the slug (or the first of a series of similar slugs) enters the first element in the set.
No. of Slugs:	The number of slugs in a series with the present slug properties. This entry defaults to 1, indicating that a single slug rather than a series of slugs is being defined.

Time Lag:	This is an optional entry, only used when a series of slugs is being defined. *Time Lag* is the delay between the times that each slug in the series enters the first element of the set.
Column Number:	The column of data in the external file which contains the slug head velocity as a function of time. Data columns are labelled upwards from one (i.e. the time column is considered 'column zero'). The remaining data for a particular slug is assumed to directly follow the velocity column within the file. The order of inputs is (i) velocity at slug head, (ii) velocity at slug tail, (iii) density at slug head, and (iv) density at slug tail, respectively.
Velocity Profile:	The options are Interpolated (default) and Constant. The Interpolated option means that the velocity at a given point on the slug is computed by linear interpolation along the slug using head and tail velocities. Constant means that the distance travelled by head and tail are calculated using head and tail velocities, but the velocity of the slug at all points is equal to the head velocity. This approach means that a slug can have varying lengths but a constant velocity.
Colour:	This is an optional entry to specify the colour of the slug in the structural animation. The entry must be the name of a standard colour or a user defined colour. A list of the standard colours can be found in the keyword *COLOUR DEFINE.

NOTES

(a) The element set name is associated with the elements comprising the set using the *Element Sets* table. Obviously, different slug flow specifications can be defined for different element sets.

(b) Use as many lines as you need to completely define the slug data for a particular set. Simply leave the *Set Name* column blank for second and subsequent lines if specifying further slugs or series of slugs. For subsequent sets, put the set name in the *Set Name* column and specify the slug data in the same way.

(c) For slugs of constant properties, *Slug Velocity* is an optional input. By default, the velocity of a slug within a set is equal to the internal fluid velocity for that set, which is specified using the *Internal Fluid* table.

(d) When specifying time-varying slugs, several slugs may share the same ASCII file. Simply leave the *File Name* column blank for second and subsequent lines if specifying further slugs which pertain to the same input file. If you wish to use a different file name, insert the file name in the *File Name* column and specify the slug data in the same way.

(e) The data file is ASCII based, and its format depends on the number of time-varying slugs which are associated with the data file. In general the file contains one column of data for the time and four additional columns for each slug definition – relating to instantaneous (i) velocity at slug head, (ii) velocity at slug tail, (iii) density at slug head, and (iv) density at slug tail, respectively. The layout of data within the text file must be consistent with the definition of time-varying slugs in the keyword file – this is governed by the *Column Number* parameter. Data columns are labelled upwards from one (i.e. the time column is considered 'column zero'). So for example, the head velocity for the first slug would typically be located in Column No.1, the head velocity for the second slug in Column No. 5, and so forth. See Slug ASCII File Format for an illustration.

Columns of data in the ASCII data file should be separated by blank spaces or tabs. Comment lines, denoted by a capital 'C' in the first column, are permitted, while lines that are completely blank are ignored. You can specify several time-varying slugs in a single data file.

(f) Slugs are displayed in Flexcom using a series of auxiliary elements, which provide a very helpful means of visualising the slug flow post-simulation. Although visualisation is an integral part of understanding slug induced loads, the use of auxiliary elements does result in increased database file sizes, which take up more disk space, and may also have some impact on simulation and post-processing times. For maximum efficiency, some users prefer to set the Slug Display option to *No*, and switch it on occasionally, for example to visually inspect slug paths during critical load cases only.

1.8.9.42 *TEMPERATURE

PURPOSE

To apply thermal loading.

THEORY

Refer to Temperature Loading for further information on this feature.

KEYWORD FORMAT

Blocks of two lines defining the temperature data repeated as often as necessary:

```
SET=Set Name
Coefficient of Expansion, Thermal Variation
```

TABLE INPUT

Input:	Description
Element Set:	The element set name.
Coefficient of Expansion:	The coefficient of thermal expansion for this set.
Temperature Variation:	The temperature change, positive or negative, in degrees.

NOTES

(a) The element set name is associated with the elements comprising the set using the *Element Sets* table. Obviously, different temperature data may be specified for different sets.

1.8.9.43 *THRUSTER

PURPOSE

To specify thruster loads on a moored vessel.

THEORY

Refer to Thruster Loads for further information on this feature.

KEYWORD FORMAT

Blocks of three lines defining the thruster data repeated as often as necessary:

```
AXIS=Axis Option
TIME=Start Time, End Time
Y-Coordinate of Thruster Location, Z-Coordinate of Thruster
Location, Thruster Load, Thruster Load Direction
```

The keyword is only relevant to mooring analyses. `Axis Option` can be GLOBAL or LOCAL.

TABLE INPUT

Input:	Description
Axis:	The axis system in which the thruster data is defined. The default of *Global* naturally specifies the global coordinate axes; the alternative is *Local*, which specifies the local vessel axis system centred at the vessel COG. See Note (a).
Start Time:	The analysis time at which the thrusters start operating.
End Time:	The analysis time at which the thrusters stop operating.
Y-Coordinate Location:	The y-coordinate of the thruster location in the local vessel axis system.
Z-Coordinate Location:	The z-coordinate of the thruster location in the local vessel axis system.
Thruster Load:	The thruster load.
Angle of Thrust :	The angle of operation of the thruster load measured in degrees anticlockwise from the global Y axis or the local vessel y axis, depending on which system you nominated using *Axis* above

NOTES

(a) If you define thruster data in the global axes, then the direction of the thruster load is constant over the duration for which the thruster is operating. If on the other hand you nominate local axes, then the direction of the thruster load is constant relative to the vessel axis system.

1.8.9.44 *TIME

PURPOSE

To define time parameters for an analysis.

THEORY

Refer to Time Variables for further information on this feature.

KEYWORD FORMAT

One of the two blocks of lines below.

Block of two lines for a fixed time-step analysis:

```
STEP=FIXED
Start Time, End Time [, Time Step] [, Ramp Time]
```

Block of three lines for a variable time-step analysis:

```
STEP=VARIABLE
Start Time, End Time [, Ramp Time]
Suggested Time Step, Min. Time Step, Max. Time Step [, Step Length]
```

Step Length defaults to 0.055.

FIXED TIME STEP

Table Input

Input:	Description
Start Time:	The simulation start time.

Finish Time:	The simulation end time.
Time Step:	The fixed time step to be used in the analysis. This entry is optional for a static analysis. See Note (a).
Ramp Time:	The time over which applied loads and displacements are gradually increased to their full value. This entry is optional, and is ignored in a static analysis. See Note (b).

Notes

(a) In a static analysis, the time variables are largely notional, and are typically from 0 to 1 second, or possibly from 1 to 2 seconds in a restarted static run. Normally, static loads and displacements can be applied in a single step, and if this is the case, no input is required for *Time Step* - by default it is the difference between start and end times. If a time step is specified, then the static loads and displacements are built up to their full values at the end of the analysis over the specified number of steps. So for example in a static analysis from 1 to 2 seconds with a time step of 0.1 seconds, the loads increase linearly to their full value over 10 steps.

(b) The *Ramp Time* entry is a time duration - it begins at the *Start Time* and ends *Ramp Time* seconds later. It allows control over the build-up of dynamic loads and displacements in an analysis with waves and/or vessel motions. For example, wave loads in a regular wave analysis are typically ramped on over 1 wave period, and the solution then proceeds for a further number of wave periods to achieve a steady state solution. If a *Nonlinear* ramp is used (and this is the default), a half cosine ramp function is used (as opposed to a linear function).

VARIABLE TIME STEP

Table Input

Input:	Description
Start Time:	The simulation start time.

Finish Time:	The simulation end time.
Ramp Time:	The time over which applied loads and displacements are gradually increased to their full value. This entry is optional, and is ignored in a static analysis. See Note (d).
Suggested Time Step:	A suggested time step which the program will use at the start of the analysis.
Minimum Time Step:	The minimum time step value. If a time step below this value is required by the time stepping algorithm, the Flexcom analysis terminates unsuccessfully.
Maximum Time Step:	A maximum time step value. The program will not allow the time step to exceed this value, regardless of the recommendation of the time stepping algorithm.
Step Length:	The factor to be used by the program in calculating the analysis time step from the instantaneous current period. See Note (c). This defaults to a value of 0.055.

Notes

(a) The variable time step option is usually confined to dynamic analyses, although it may be necessary in some extreme static cases.

(b) The program chooses an optimum time step based on two main criteria, namely i) the number of iterations required for the last three convergent solutions, and ii) the instantaneous current period, which is a measure of the dominant period in the response at any particular instant. The exact details are immaterial here, except to note that the process is efficient and largely transparent to the user.

(c) The program calculates the analysis time step, when this is based on the current period, by multiplying the instantaneous current period value by the factor specified under the heading of *Step Length*. So for example the default *Step Length* value of 0.055 means the time step is approximately 1/18th of the dominant period in the dynamic response at any time. The default value is adequate in almost all cases, but may occasionally be increased (say to 0.1 or 1/10th) by experienced users who feel the resulting time step is too short.

(d) The *Ramp Time* entry allows control over the build up of dynamic loads and displacements in an analysis with waves and/or vessel motions. For example, wave loads in a regular wave analysis are typically ramped on over 1 wave period, and the solution then proceeds for a further number of wave periods to achieve a steady state solution. If a *Nonlinear* ramp is used (and this is the default), a half cosine ramp function is used (as opposed to a linear function).

1.8.9.45 *TIME STEPPING

PURPOSE

To select the time stepping algorithm and to define associated numerical damping coefficients.

THEORY

Refer to Time Integration Algorithms for further information on this feature.

KEYWORD FORMAT

A block of two lines to select the time stepping algorithm and to specify the numerical damping coefficients. You can specify either Generalised-Alpha (the default) or Hilber-Hughes. There are actually two possible formats for Generalised-Alpha, depending on how you specify the coefficients:

Generalised-Alpha:

```
TYPE=GENERALISED-ALPHA
P-INFINITY=[P-Infinity]
```

Or:

```
TYPE=GENERALISED-ALPHA
[Alpha-f][, Alpha-m]
```

If you do not specify any coefficients, the program assumes a *P-Infinity* value of 0.4, and the *Alpha-f* (0.286) and *Alpha-m* (-0.143) coefficients are derived from this.

Hilber-Hughes:

```
TYPE=HILBER-HUGHES
[Numerical Damping Coefficient]
```

The *Numerical Damping Coefficient* defaults to -0.25.

TABLE INPUT

Generalised Alpha

Input:	Description
ρ-Infinity:	The spectral radius at infinity, ρ_∞. A value of 0.4 is assumed by default unless you specify otherwise. Experience with Flexcom models have suggested that this is an optimal value which helps to ensure a robust temporal integration model without adversely affecting solution accuracy. Unless specified explicitly, the α_f and α_m coefficients are derived from ρ_∞.
Alpha-f:	The numerical damping coefficient α_f.
Alpha-m:	The numerical damping coefficient α_m.

Hilber-Hughes-Taylor

Input:	Description
Hilber Hughes Damping Coefficient:	The numerical damping coefficient. This defaults to a value of -0.25. Values must be between 0 and –1/3.

Notes

(a) Varying the default values may improve convergence in some highly sensitive analyses, however you should use this keyword very rarely or not at all.

(b) Default values for α_f (0.286) and α_m (-0.143) for Generalised-Alpha time stepping are derived from ρ_∞ as follows:

$$\alpha_f = \left(\frac{\rho_\infty}{\rho_\infty + 1}\right); \alpha_m = \left(\frac{2\rho_\infty - 1}{\rho_\infty + 1}\right)$$

(c) If you specify α_f=0.25 and α_m=0 for the Generalised-Alpha time stepping algorithm, this effectively replicates the Hilber-Hughes-Taylor operator.

(d) Earlier versions of Flexcom (up to and including Flexcom 8.10) used the Hilber-Hughes-Taylor operator as the default. Generalised-Alpha has since been shown to provide more effective numerical damping, particularly for sensitive models, so it is now the default method. If you are re-running some old simulations which previously used Hilber-Hughes, it is possible that you may notice some very slight differences in results.

1.8.9.46 *TIMETRACE

PURPOSE

To request the storage of results for timetrace postprocessing (this is mainly used in the area of time domain fatigue analysis).

THEORY

Refer to Timetrace Postprocessing for further information on this feature.

KEYWORD FORMAT

The data begins with up to three lines, most of which are optional, which define general parameters relating to subsequent postprocessing of timetrace output. The individual timetrace requests then follow, in blocks of lines according to output type.

The first line, which is optional, defines miscellaneous variables relating to the duration and format of output:

```
[START=Output Start Time] [, END=Output End Time] [, INTERVAL=Interval] [, ST
```

The second line specifies data relating to the calculation of statistics and spectra by the *Timetrace Postprocessor*. It is required in the case of a random sea analysis, but optional otherwise.

```
STATISTICS=No. of Ensembles [, Calculation Start Time] [, Time Step]
```

The third line, which is again optional, specifies the procedure to be used in the computation of extreme values by the *Timetrace Postprocessor*.

```
[EXTREME=Calculation Procedure]
```

The fourth line is required if the line with EXTREME= is present. The format of the fourth line depends on the *Calculation Procedure* specified – the options are RAYLEIGH (the default) and WEIBULL. For RAYLEIGH extrema postprocessing, the fourth line takes the following format:

```
DATA=[Storm Duration] [, Probability]
```

For WEIBULL extrema postprocessing, it takes the following format:

```
DATA=[Storm Duration] [, Probability] [, Threshold]
```

Block of lines requesting motion timetrace output. The second line can be repeated as often as necessary.

```
TYPE=MOTION
Node (Number or Label), DOF
```

Block of lines requesting force timetrace output. The requests can be made on the basis of element sets or individual elements. The requests may be mixed and/or repeated as often as necessary.

Force timetrace output request for an element set:

```
TYPE=FORCE
SET=Set Name
Local Node, Variable [, LOCATION=Location]
```

Refer to Force Variable Input for further information on acceptable *Variable* values. Refer to Location Parameter Input for further information on the significance of the *Location* input. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis.

Force timetrace output request for an individual element:

```
TYPE=FORCE
ELEMENT=Element (Number or Label)
Local Node, Variable [, LOCATION=Location]
```

Block of lines requesting reaction timetrace output. The second line can be repeated as often as necessary.

```
TYPE=REACTION
Node (Number or Label), DOF
```

A `Time Step` value is required in a random sea dynamic analysis with a variable time-step; it is ignored in the case of an analysis with a fixed step. The DATA= line is required if EXTREME= is invoked. This keyword is not relevant for frequency domain dynamic analysis.

`Output Start Time` and `Calculation Start Time` both default to the analysis start time. End Time defaults to the analysis end time. If no `Interval` value is specified, data is output at all solution times. `Format` can be ASCII, BINARY (the default) or IMPORT. `Calculation Procedure` can be RAYLEIGH or WEIBULL. `Storm Duration` defaults to 3 hours and Probability to 0.01 (1%). `Threshold` defaults to 1. If you specify a node/element label rather than a number, it must be enclosed in {} brackets. `Scale` defaults to 1 (see Note a).

TIMETRACE - STORAGE

Purpose

To specify data relating to the frequency and format of timetrace output.

Table Input

Input:	Description
Start Time:	The start time for timetrace output. This entry is optional and defaults to the analysis start time.
End Time:	The end time for timetrace output. This entry is optional and defaults to the analysis end time.
Recording Interval:	The time interval in seconds between timetrace outputs. This entry is optional and defaults to all solution times.

Storage Format:	This option allows you to specify the format in which the timetrace output is to be stored. The options are ASCII, *Binary* (the default) and *Import*. A detailed description of these options is given in Note (a).

Notes

(a) In earlier versions of Flexcom, timetrace output was stored in ASCII format by default. In this case, values at each solution time are output on a series of lines – the actual solution time (only) is on the first line of the series, and the requested outputs are then on subsequent lines, typically four values to a line. This means the data can be readily examined and interpreted, but has the disadvantage that for a long simulation with many outputs requested, the file size can be large. The recommended storage method is now *Binary* – the main advantages being that storage/retrieval of data tends to be quicker, and file sizes are smaller, but means that you cannot edit the file manually. If the *Import* format is requested, the file format is similar to ASCII, but all values at a particular solution time, including the actual solution time itself, are output on a single line. This allows the data to be readily imported into, say, Excel.

TIMETRACE - STATISTICS

Purpose

To specify the start time for the calculation of response statistics in any subsequent *Timetrace Postprocessing* run.

Table Input

Input:	Description
Start Time:	The start time for the calculation of statistics. Any values before this time are excluded. Use this entry to exclude initial transients from statistics calculations.

Notes

(a) The use of this table is optional. By default, the start time for the calculation of statistics is the first timetrace output time.

(b) The entry you specify here is also used as the start time for the calculation of response spectra in a random sea analysis.

TIMETRACE - SPECTRA

Purpose

To define parameters relating to the calculation of response spectra in any subsequent *Timetrace Postprocessing* run.

Table Input

Input:	Description
No. of Ensembles:	The number of ensembles to be used in calculating spectra. See Note (a).
Time Step:	The time step to be used when calculating spectra from the results of a random sea dynamic analysis with a variable time step. See Note (b). If the analysis used a fixed time step then this entry is not required, and any value you specify is ignored.

Notes

(a) The procedure used by Flexcom in calculating spectra is as follows. Firstly, the output timetrace is divided equally into a number of smaller timetraces or ensembles. A spectrum for each ensemble is then calculated using the Fast Fourier Transform (FFT) algorithm. Finally, the actual spectrum to be output is found as an average of the spectra calculated for each ensemble. This standard procedure minimises the statistical error associated with the FFT process. You specify the number of ensembles to be used in this process using the *No. of Ensembles* entry above. This value should always be greater than 1.

(b) The FFT algorithm requires a record with a fixed time step. When you perform a variable step analysis, obviously such a record is not available, and so Flexcom must synthesise one by interpolating from the variable step record. The *Time Step* entry tells Flexcom what time step to use in the synthesised record.

EXTREME VALUES - RAYLEIGH

Purpose

To specify parameter values to be used in calculating extreme values with the Rayleigh distribution in any subsequent *Timetrace Postprocessing* run.

Table Input

Input:	Description
Storm Duration:	The storm duration in hours to be used when calculating extreme values. The default is 3 hours.
Probability:	The exceedance probability to be used when calculating extreme values. The default is 0.01 (= 1%).

Notes

(a) Refer to Extreme Values for a detailed discussion of all aspects of specifying data relating to extreme value postprocessing.

(b) Note that the *Extreme Values – Rayleigh* and *Extreme Values – Weibull* tables are mutually exclusive. Specification of data in either table indicates the extreme value postprocessing approach which you wish to use.

EXTREME VALUES - WEIBULL

Purpose

To specify parameter values to be used in calculating extreme values with the Weibull distribution in any subsequent *Timetrace Postprocessing* run.

Table Input

Input:	Description
Duration:	The duration in hours to be used when calculating extreme values. The default is 3 hours
Probability:	The exceedance probability to be used when calculating extreme values. The default is 0.01 (= 1%).

Thresh old:	The proportion of largest maxima /smallest minima to be used when calculating extreme values. The default is 1. See Note (b).

Notes

(a) Refer to Extreme Values for a detailed discussion of all aspects of specifying data relating to extreme value postprocessing.

(b) A threshold equal to 1 denotes all the maxima/minima in the timetrace will be used to calculate the extreme values. A threshold equal to 1/n (e.g. 1/3) uses the upper 1/n of largest maxima and lowest 1/n of smallest minima to calculate the extreme value.

(c) Note that the *Extreme Values – Rayleigh* and *Extreme Values – Weibull* tables are mutually exclusive. Specification of data in either table indicates the extreme value postprocessing approach which you wish to use.

TIMETRACE - MOTION

Purpose

To request timetraces of the motions of selected nodes.

Table Input

Input:	Description
Node:	The node (number or label) at which the motion timetrace is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global degree of freedom (DOF) at this node for which the timetrace is required. Specify a value of 1 for translation in the global X-direction, 2 for the global Y-direction, or 3 for the global Z-direction; 4, 5 or 6 for the components of the rotation vector in the global X, Y and Z directions respectively; or 7 for the magnitude of rotation.

Notes

(a) If any timetrace is requested, then by default a timetrace of wave elevation at Y = Z = 0 (that is, at the vertical axis) is included in the output.

TIMETRACE - FORCE (BY SET)

Purpose

To request timetraces of restoring forces in specified element sets.

Table Input

Input:	Description
Element Set:	The element set for which timetrace output is required.
Local Node:	This option allows you to choose between three locations on the specified elements. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis.
Variable:	This list allows you to select a relevant output parameter. The entries are largely self-explanatory. See Note (a).
Location:	This parameter is appropriate when you request bending stress, bending strain, von Mises stress, or pressure. See Note (c).

Notes

(a) Refer to Force Variable Input for further information on acceptable *Variable* values.

(b) If any timetrace is requested, then by default a timetrace of wave elevation at Y = Z = 0 (that is, at the vertical axis) is included in the output.

(c) Refer to Location Parameter Input for further information on the significance of the *Location* input.

TIMETRACE - FORCE (BY ELEMENT)

Purpose

To request timetraces of restoring forces in specified elements.

Table Input

Input:	Description
Element:	The element (number or label) for which timetrace output is required. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Local Node:	This option allows you to choose between three locations on the specified element. *First Node* refers to the first specified node of the element, *Last Node* is the second specified node, and *Midpoint* is half-way between the two.
Variable:	This list allows you to select a relevant output parameter. The entries are largely self-explanatory. See Note (a).
Location:	This parameter is appropriate when you request bending stress, bending strain, von Mises stress, or pressure. See Note (c).

Notes

(a) Refer to Force Variable Input for further information on acceptable *Variable* values.

(b) If any timetrace is requested, then by default a timetrace of wave elevation at Y = Z = 0 (that is, at the vertical axis) is included in the output.

(c) Refer to Location Parameter Input for further information on the significance of the *Location* input.

TIMETRACE - REACTION

Purpose

To request timetraces of reactions at restrained nodes.

Table Input

Input:	Description

Node:	The node (number or label) for which the reaction timetrace is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global degree of freedom (DOF) at the node for which the timetrace is required. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, 3 for the global Z-direction, or 4, 5 or 6 for moments about the global X, Y and Z axes respectively.

Notes

(a) If any timetrace is requested, then by default a timetrace of wave elevation at Y = Z = 0 (that is, at the vertical axis) is included in the output.

1.8.9.47 *TOLERANCE

PURPOSE

To define the analysis convergence tolerance measure and related data.

THEORY

Refer to Solution Convergence for further information on this feature.

KEYWORD FORMAT

Block of two lines with the appropriate tolerance data, all of which are optional.

The first line defines the analysis type.

```
ANALYSIS=Analysis Type
```

The format of the second line depends on the analysis type.

For all static and time domain dynamic analysis the format is:

```
[Displacement Tolerance] [, Max. Iterations] [, Small Torque Value], [Rigid S
Rigid Seabed Negative Reaction Value] [, Energy Tolerance]
```

For frequency domain dynamic analysis the format is:

```
[Tolerance] [, Max. Iterations] [, Relaxation Parameter]
```

`Analysis Type` can be STATIC/TIME or FREQUENCY. `Displacement Tolerance` defaults to 0.001 for static analyses and 0.025 for dynamic. `Max. Iterations` defaults to 20. `Small Torque Value` defaults to 10. If no value is specified for `Rigid Seabed Threshold Penetration`, `Rigid Seabed Negative Reaction Value` or `Energy Tolerance`, the corresponding convergence check is not applied. Where a value is specified for any of these parameters, it should be a positive value. `Tolerance` defaults to 0.025 for frequency domain analyses. `Relaxation Parameter` defaults to 0.8.

TIME DOMAIN ANALYSIS - TOLERANCE

Table Input

Input:	Description
Tolerance Measure:	The tolerance used in determining if convergence has been achieved between successive iterations at a particular solution time. This entry is optional. If you do not specify a value, then Flexcom chooses a default based on the analysis type. Specifically, the default for static analyses is 0.001 (0.1%), while for dynamics it is 0.025 (2.5%). See Note (a).
Maximum Number of Iterations:	The maximum number of iterations Flexcom may perform at a particular time. The default number of iterations is 20.
Small Torque Value:	A value of torque below which Flexcom does not enforce convergence on torque for each element. The default value is 10. See Note (b).
Rigid Surface Threshold Penetration:	A threshold value to use in checking seabed penetration as part of the convergence calculations. This entry is left blank by default. See Note (c).
Negative Reaction Threshold:	A threshold value to use in checking reactions at seabed nodes as part of the convergence calculations. This entry is left blank by default. See Note (c).

Energy Residual Tolerance:	An alternative convergence criteria based on energy residuals. This entry is left blank by default. See Note (d).

Notes

(a) Refer to Solution Convergence for further information on convergence measures.

(b) Refer to Solution Convergence for further information on the significance of the *Small Torque Value*.

(c) Refer to Seabed Penetration and Negative Contact Reactions for further information on the significance of the *Rigid Surface Threshold Penetration* and *Negative Reaction Threshold* inputs, respectively.

(d) The energy residual criterion works alongside the standard convergence criterion. If you do not specify an energy residual tolerance level, the criterion is not checked. A converged solution is achieved when both criteria are satisfied for a particular iteration. Refer to Energy Residual Convergence for a detailed discussion of the energy residual criterion.

(e) The parameters of this *Tolerance* menu should be only rarely varied. The default values are adequate for the majority of analyses.

FREQUENCY DOMAIN ANALYSIS - TOLERANCE

Table Input

Input:	Description
Tolerance Measure:	The tolerance used in determining if convergence has been achieved between successive iterations. The default value is 0.025 (2.5%).
Maximum Number of Iterations:	The maximum number of iterations that Flexcom may perform. The default number of iterations is 20.

Relaxation Parameter:	The relaxation parameter used by Flexcom to improve the rate of convergence of dynamic analyses. See Note (a).

Notes

(a) Refer to Frequency Domain Convergence for further information on the significance of the *Relaxation Parameter*.

1.8.9.48 *UPSTREAM STRUCTURE

PURPOSE

To specify the name of the Flexcom analysis of the upstream structure subjected to the free or undisturbed stream current velocity field.

THEORY

Refer to Wake Interference for further information on this feature.

Note also that the old `*WAKE INTERFERENCE` and `*UPSTREAM STRUCTURE` keywords have effectively been superseded by the new wake interference keywords which are more generic and provide additional functionality. Refer to `*WAKE UPSTREAM` and `*WAKE DOWNSTREAM` for further information.

KEYWORD FORMAT

Blocks of two lines.

```
[FILE=Full Path of Flexcom Current Analysis File]
[SET=Element Set Name]
```

The use of `*UPSTREAM STRUCTURE` is compulsory (in the same analysis) if `*WAKE INTERFERENCE` is invoked. One or other of the two lines must be present. The file name should be specified without extension. If it contains spaces, then it should be enclosed in double quotation marks. *Element Set Name* is an existing element set, which must be defined in the initial static analysis of the upstream structure.

The FILE= input is the name of the file for the static or quasi-static analysis with current of the upstream structure. It is required if the two structures under consideration are being analysed in separate models. It is omitted where the upstream and downstream structures are included in the same model. In the former case, SET= may be optionally invoked to specify which elements of the model in the file are to be included in the wake calculations; the default is all elements. In the latter case, SET= is required to identify the elements of the combined model which comprise the upstream structure; no default is possible in this case.

TABLE INPUT

Input:	Description
Flexcom File:	The name of the Flexcom analysis file. You do not need to include a file extension. See Note (b).
Set Name:	The name of the set that contains the elements on the upstream structure that you want to include in the wake interference analysis. The default is *All*.

NOTES

(a) If the two structures are in separate models, then the *Flexcom File* input is mandatory, and the specification of a *Set Name* is optional; by default all of the elements in the *Flexcom File* model constitute the upstream structure. If the two structures are in the same model, then the *Flexcom File* input should be omitted. In this case the specification of a *Set Name* other than *All* is mandatory; since the two structures are in the same model, then the upstream structure cannot consist of all of the elements in that model.

1.8.9.49 *USER SOLVER VARIABLES

PURPOSE

To provide advanced Flexcom users with the ability to define custom code for increased modelling flexibility. This feature is intended for advanced applications, facilitating customised solutions to cater for situations where the standard modelling options do not completely cover user requirements. This option allows you, for example, to directly augment the global force vector to simulate an arbitrary time-varying load.

THEORY

Refer to User Solver Variables for further information on this feature. You may also be interested in reading about a related feature, User Defined Element.

KEYWORD FORMAT

A single line of data containing the name of the DLL file.

```
FILE=DLL File Name
```

DLL File Name should include the entire path of the user subroutine file with its extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

TABLE INPUT

Input:	Description
DLL File Name:	The name of the DLL file which contains the user subroutine.

1.8.9.50 *USER DEFINED ELEMENT

PURPOSE

To provide advanced Flexcom users with the ability to define custom code for altering element properties. This feature is intended for advanced applications, facilitating customised solutions to cater for situations where the standard modelling options do not completely cover user requirements. This option allows you, for example, to vary an element's structural properties while a simulation is in progress.

THEORY

Refer to User Defined Element for further information on this feature. You may also be interested in reading about a related feature, User Solver Variables.

KEYWORD FORMAT

A single line of data containing the name of the DLL file.

```
FILE=DLL File Name
```

DLL File Name should include the entire path of the user subroutine file with its extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

TABLE INPUT

Input:	Description
DLL File Name:	The name of the DLL file which contains the user subroutine.

1.8.9.51 *VESSEL TIMETRACE

PURPOSE

To specify that the combined high and low frequency motions of a vessel are to be read from an ASCII timetrace data file.

THEORY

Refer to Combined High and Low Frequency Motion Timetraces for further information on this feature.

KEYWORD FORMAT

A block of two lines repeated as many times as necessary to define motions for all appropriate vessels.

```
VESSEL=Vessel Name [, OPTION=Axis System]
FILE=File Name
```

Data for any `Vessel Name` specified here must be input under `*VESSEL`. As this keyword is used to specify that both high and low frequency vessel motions are to be read from a text file, it may not be used in combination with `*RAO` or `*DRIFT` for any particular vessel. `Axis System` can be GLOBAL (the default) or LOCAL. `File Name` should include the entire path of the vessel timetrace motion file including its extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

TABLE INPUT

Input:	Description
Vessel:	The name of the vessel for which combined motion data is being input.
Timetrace File Name:	The name of the ASCII data file containing the timetrace of the motion of the vessel reference point. *Timetrace File Name* should ideally include the entire path of the vessel drift motion file, including extension. If the name or any part of its path contains spaces, the full *Timetrace File Name* should be enclosed in double quotation marks (" "). The required format of the data file is described in Note (a).
Axis Type:	The options are *Global* (the default) and *Local*.

NOTES

(a) The format of the ASCII vessel motion timetrace file is as follows. The file contains seven columns of data. The first column contains time data, and the remaining six correspond to displacements of the reference point in six degrees of freedom. Comment lines, denoted by a capital ‘C’ in the first column, are permitted, while lines that are completely blank are ignored. An example data file extract is shown below.

```
C
C Time    DOF1     DOF2    DOF3    DOF4    DOF5     DOF6
C
  0.0      0.0      0.0     0.0     0.0     0.0     0.0
  0.2      0.2      0.5     0.0     0.0     0.0     3.419
  0.4      0.3      0.9     0.0     0.0     0.0     6.794
  0.6      0.4      1.2     0.0     0.0     0.0    10.081
  0.8      0.6      1.4     0.0     0.0     0.0    13.236
  1.0      0.8      1.5     0.0     0.0     0.0    16.219
  1.2      0.9      1.6     0.0     0.0     0.0    18.990
  1.4      1.1      1.8     0.0     0.0     0.0    21.513
  1.6      0.8      2.0     0.0     0.0     0.0    23.757
  1.8      0.6      2.1     0.0     0.0     0.0    25.690
```

The first column of data contains time values. Columns 2 – 4 contain the displacements (not coordinates) of the reference point from its initial position at the start of the dynamic analysis. This means that the data in the timetrace file should not include the value of any static offset you apply to the vessel. If it does, what will happen is that the offset will be applied twice. These displacements are in either the global XYZ axes or else in a local axis system defined by the initial orientation of the vessel – you specify which using the *Vessel Motions – Axis Type* option. Columns 5 – 7 contain rotations in degrees. Column 5 contains the yaw rotation of the vessel about the vertical or global X axis. Columns 6 and 7 are roll and pitch respectively, relative to either the yawed or initial vessel axes, depending on whether large or small angle theory is specified. Note that each line of the file (other than comment lines or blank lines) must contain 7 numerical values.

Flexcom uses cubic spline interpolation to find the displacements and rotations of the reference point at times intermediate to those specified in the data file (for this reason the analysis solution times do not need to match those in the data file).

(b) For analysis times before the earliest time specified in the data file, Flexcom uses the values at that earliest time until the analysis time exceeds this value. Similarly, for analysis times after the latest time in the data file, Flexcom uses the values at that latest time.

(c) The choice of axis system refers to data you specify for vessel timetrace motions. Motions in this context means translations only – rotations always refer to vessel axes. Naturally *Global* stipulates that translations represent drift or combined motions in any or all of the global X, Y or Z axes, and/or that the angle you specify in defining a constant velocity is relative to global Y. Conversely *Local* indicates that translations represent any or all of heave, surge or sway, defined with reference to the initial orientation of the vessel axes, or that the angle you specify in defining a constant vessel velocity is relative to the vessel surge axis.

(d) Vessel motions are normally due to wave excitation. If you apply a time history of vessel motion, then you must ensure that you also specify the corresponding wave excitation which generated those vessel motions. This is typically specified via a Time History of Water Surface Elevation.

1.8.9.52 *VESSEL VELOCITY

PURPOSE

To specify a vessel constant velocity horizontal velocity.

THEORY

Refer to Vessels and Vessel Motions for further information on this feature.

KEYWORD FORMAT

A block of two lines to define constant vessel velocities which is repeated as necessary.

```
VESSEL=Vessel Name, [AXIS=Axis System]
Speed, Angle
```

Data for any `Vessel Name` specified here must be input under `*VESSEL INTEGRATED`. As this keyword is used to specify that a constant vessel velocity is to be combined with high frequency vessel motions, it may not be used in combination with `*VESSEL TIMETRACE` for any particular vessel. This keyword is not relevant for frequency domain dynamic analysis.

`Axis System` can be either GLOBAL (the default) or LOCAL.

TABLE INPUT

Input:	Description

Vessel Name:	The name of the attached vessel.
Speed:	The constant vessel speed.
Angle:	The motion direction, measured in degrees anti-clockwise from either the global Y-direction or the local surge axis. The default is 0°.
Axis Type:	The options are *Global* (the default) and *Local*.

NOTES

(a) The velocity you specify here is integrated to define a time history of horizontal motion in the direction you specify, and this is simply added to the vessel displacements from other sources.

(b) The choice of axis system refers to data you specify for vessel constant velocity motions. Motions in this context means translations only – rotations always refer to vessel axes. Naturally *Global* stipulates that translations represent drift or combined motions in any or all of the global X, Y or Z axes, and/or that the angle you specify in defining a constant velocity is relative to global Y. Conversely *Local* indicates that translations represent any or all of heave, surge or sway, defined with reference to the initial orientation of the vessel axes, or that the angle you specify in defining a constant vessel velocity is relative to the vessel surge axis.

1.8.9.53 *VIV DRAG

PURPOSE

To instruct Flexcom to read vortex-induced vibration (VIV) drag coefficient amplification factors from the results of a Shear7 analysis.

THEORY

Refer to VIV Drag for further information on this feature.

KEYWORD FORMAT

Blocks of two lines.

```
FILE=Full Path of Shear7 Output File
[SET=Element Set Name]
```

The *Shear7* file name is required, and should be specified without extension – Flexcom assumes the file type is .plt. If the full path contains spaces, then it should be enclosed in double quotation marks. The default *Element Set Name* is all elements in the model. `*VIV DRAG` can be specified at any stage in a restart cascade; the drag amplification factors will be used in the analysis in which `*VIV DRAG` is invoked (provided it is a static or quasi-static analysis) and in any subsequent static or quasi-static analysis in the cascade; the data is not used in a dynamic analysis. For the *Shear7* data to be meaningfully applied, then the elements in *Element Set Name* (whether default or user-defined) must form a continuous line of elements.

TABLE INPUT

Input:	Description
Shear7 File:	The name of the Shear7 output file. You do not need to include the file extension - Flexcom always searches for a file with the correct extension (.plt). This input is mandatory.
Set Name:	The name of the set that contains the elements whose drag coefficients are to be multiplied by the factors that are read from the Shear7 file. This input is optional. The default is All.

1.8.9.54 *VIV EFFECTS

PURPOSE

To instruct Flexcom to continuously run Modes/Shear7 analyses of the downstream structure during the wake interference analysis.

THEORY

Refer to VIV Effects for further information on this feature.

KEYWORD FORMAT

A block of three lines as follows:

```
MODES FILE=Full Path of File for Modes Analysis of Downstream Structure
[SHEAR7 PATH=Full Path of Shear7 Executable]
[SHEAR7 VERSION=Version Number]
```

The Modes keyword file name is required if `*VIV EFFECTS` is invoked. It should be specified without extension. If the Modes file name or the *Shear7* path contains spaces, then they should be enclosed in double quotation marks.

Notes: SHEAR7 PATH defaults to "C:\Shear7\Bin\Shear7.exe". Version Number is specified in the form of X.Y where X and Y are integer values, and defaults to 4.6 if not explicitly specified.

TABLE INPUT

Input:	Description
Modes File:	The name of the Modes analysis file. The Modes file must be supplied as a keyword (.key) file. This input is mandatory.
Shear7 Installation Directory:	The path to the Shear7 software program. The default path is C:\Shear7\Bin\Shear7.exe.
Shear7 Version:	The version of Shear7. This defaults to 4.6 if not explicitly specified.

NOTES

(a) The *VIV Effects* option is not available with the *User-Defined* wake model.

(b) Different versions of Shear7 typically require a different 'common.cl' file. If you are upgrading between different versions, but maintaining the same installation directory, you should manually delete the 'common.cl' file. Shear7 will create a version compatible 'common.cl' file when it is next run.

1.8.9.55 *WAKE DOWNSTREAM

PURPOSE

To specify the composition of the downstream structure in terms of element sets for use in wake interference calculations. The keyword also facilitates the (optional) definition of lift coefficients in the case of a *User-Defined* wake interference model.

THEORY

Refer to Wake Interference for further information on this feature.

KEYWORD FORMAT

Blocks of lines defining the set(s) of elements which comprise the downstream structure, optionally an associated lift coefficient with each set, and optionally linking the downstream element set with a specific portion of the upstream structure.

A single line as follows. The line may be repeated as many times as necessary to completely define the downstream structure.

```
SET=Downstream Element Set Name [, LIFT=Lift Coefficient] [, UPSTREAM=Upstrea
```

Restrictions: The use of `*WAKE DOWNSTREAM` is compulsory if wake interference effects are to be included in an analysis. *Downstream Element Set Name* must be an existing element set, defined in the initial static analysis of the downstream structure. *Lift Coefficient* is a mandatory input in the case of a *User-Defined* wake interference model – otherwise it is irrelevant. *Upstream Element Set Name* must be an existing element set, defined in the initial static analysis of the upstream structure.

TABLE INPUT

Input:	**Downstream Element Set Name**	**Lift Coefficient**	**Upstream Element Set Name**

Description	The name of an existing set that contains the elements on the downstream structure that you want to include in the wake interference analysis. Can be specified as many times as required.	Mandatory lift coefficient in USER-Defined model. See note (a).	Optional name of an existing upstream element set. See note (b).

Notes

(a) Lift Coefficient is mandatory in a User-Defined model otherwise it is irrelevant.

(b) While the specification of an upstream element set is optional, if utilised correctly, it will improve the efficiency of the internal computations. Specifically, in determining the appropriate wake field for a downstream element, the program will first check a specific portion of the upstream structure (if specified) for a corresponding upstream element, before proceeding to checking the entire upstream structure if a suitable element is not found within the designated element set

1.8.9.56 *WAKE INTERFERENCE

PURPOSE

To specify that the present analysis should include wake interference effects, and to specify associated parameters.

THEORY

Refer to Wake Interference for further information on this feature.

Note also that the old `*WAKE INTERFERENCE` and `*UPSTREAM STRUCTURE` keywords have effectively been superseded by the new wake interference keywords which are more generic and provide additional functionality. Refer to `*WAKE UPSTREAM` and `*WAKE DOWNSTREAM` for further information.

KEYWORD FORMAT

Two lines, the second of which is optional, defining general parameters, followed by one or more lines whose format(s) depends on the first of the initial two lines.

```
MODEL=Wake Effect Model [, OUTPUT=YES] [, OPTION=Coefficient Type]
[SET=Element Set Name]
```

Wake Effect Model can be HUSE, BLEVINS or USER. The format of the third and subsequent lines depends on which of these is nominated. The use of OPTION= is appropriate only for the USER model.

Huse's Formulation:

One line with three values:

```
[k1, k2, k3]
```

Blevins' Formulation:

One line with three values:

```
[a1, a2, a3]
```

User-Defined Formulation:

Data in this case consists of drag and lift coefficient data, and is specified in blocks. A first line with either TYPE=DRAG or TYPE=LIFT indicates which data is being input (they can be specified in either order). The coefficients are specified in terms of coefficient factors or absolute coefficient values, depending on the *Coefficient Type* selected. The first line following TYPE= lists the transverse separation values at which the coefficients are defined, normalised with respect to the drag diameter of the upstream structure. Each line after that starts with a longitudinal separation value, again normalised with respect to the drag diameter of the upstream structure, followed by coefficients corresponding to the separation values. If L represents the centre-to-centre distance between upstream and downstream structures in the longitudinal direction (direction of undisturbed flow), and T is the centre-to-centre spacing in the transverse direction (normal to undisturbed flow); then the i^{th} normalised separations are denoted li=Li/Dd and ti =Ti/Dd . The following then summarises the above:

```
TYPE=DRAG
    t1,        t2,        t3,        ......  tn
l1, Cd(l1,t1), Cd(l1,t2), Cd(l1,t3) ....... Cd(l1,tn)
l2, Cd(l2,t1), Cd(l2,t2), Cd(l2,t3) ....... Cd(l2,tn)
...
...
lm, Cd(lm,t1), Cd(lm,t2), Cd(lm,t3) ....... Cd(lm,tn)
TYPE=LIFT
    t1,        t2,        t3,        ....... tn
```

```
l1, Cl(l1,t1), Cl(l1,t2), Cl(l1,t3) ....... Cl(l1,tn)
l2, Cl(l2,t1), Cl(l2,t2), Cl(l2,t3) ....... Cl(l2,tn)
...
...
lm, Cl(lm,t1), Cl(lm,t2), Cl(lm,t3) ....... Cl(lm,tn)
```

If lift coefficients are defined in terms of coefficient factors, it is necessary to define the actual lift coefficients themselves, and this is specified via a data block beginning with a TYPE=LIFT COEFFICIENT line. This is then followed by blocks of two lines defining lift coefficients on an element set by element set basis.

```
TYPE=LIFT COEFFICIENT
[SET=Element Set Name]
Wake Lift Coefficient
```

If the SET= line is omitted, the specified lift coefficient applies to all elements, in which case previous or subsequent blocks of lines assigning coefficients to other element sets are invalid.

`*WAKE INTERFERENCE` may only be invoked in a static or quasi-static analysis. Both occurrences of *`Element Set Name`* must refer to an existing element set, which must be defined in the initial static analysis of the downstream structure. The specification of drag and lift data is required if *`Wake Effect Model`* is set to USER. Drag and lift data must be specified at the same l and t values. The l and t values must be specified in ascending order. The specification of lift coefficients is required for the downstream structure if the lift data is specified in terms of coefficient factors.

The second line following `*WAKE INTERFERENCE` (with SET=) is required where the upstream and downstream structures are included in the same model, in which case it is used to identify the elements of the combined model which comprise the downstream structure. If the two structures are being analysed in separate models, it is optional; the default is all elements in the present (downstream) structure model. Likewise the line following TYPE=LIFT COEFFICIENT (with SET=) is optional if the two structures are being analysed in separate models; if omitted the element set again defaults to all elements (of the downstream structure).

The specification of the *k1, k2* and *k3* values for Huse's formulation is optional; these default to 0.25, 1 and 0.693 (not 0.639) respectively. Likewise, the specification of the *`a1, a2`* and *`a3`* values for Blevins' formulation is optional; these default to 1, 4.5 and -10.6 respectively. *`Coefficient Type`* may be either FACTORS (the default) or ABSOLUTE, and is only relevant to the User-Defined wake formulation.

The pair of lines following TYPE=LIFT COEFFICIENT are repeated as often as necessary to define wake lift coefficients for all elements of the downstream structure model.

WAKE INTERFERENCE MODEL

Purpose

To specify the formulation to be used in calculating wake interference effects.

Table Input

Input:	Description
Formulation:	The options are *Huse's* (the default), *Blevins'* and *User-Defined*.

OUTPUT

Purpose

To request additional output from wake modelling calculations.

Table Input

Input:	Description
Output:	The options are *No* (the default) and *Yes*.

Notes

(a) If you invoke this option, Flexcom presents detailed information regarding the wake modelling calculations. Specifically, for every integration point on every element of the downstream structure, the following is provided:

- Longitudinal (L) and transverse (T) centre-to-centre distance to upstream structure
- Reduced current velocity
- Drag and lift coefficients

- Lift Force

This information may be useful in some circumstances in understanding wake interference analysis results, but in general it is not required, so this option would be rarely invoked.

HUSE'S FORMULATION DATA

Purpose

To specify parameters relating to Huse’s wake interference formulation.

Table Input

Input:	Description
Set Name:	The name of the set that contains the elements on the downstream structure that you want to include in the wake interference analysis. The default is *All*. See Note (a).
k1:	The k1 coefficient of Huse’s formulation equation. The default is 0.25.
k2:	The k2 coefficient of Huse’s formulation equation. The default is 1.
k3:	The k3 coefficient of Huse’s formulation equation. The default is 0.693.

Notes

(a) If the two structures are in separate models, then the specification of a Set Name is optional. By default, all of the elements in the current model constitute the downstream structure. If the two structures are in the same model, then the specification of a Set Name other than All is mandatory. In this case, the downstream structure cannot consist of all of the elements in that model.

BLEVIN'S FORMULATION DATA

Purpose

To specify parameters relating to Blevins’ wake interference formulation.

Table Input

Input:	Description
Set Name :	The name of the set that contains the elements on the downstream structure that you want to include in the wake interference analysis. The default is *All*. See Note (a).
a1:	The a1 coefficient of Blevins' formulation equation. The default is 1.
a2:	The a2 coefficient of Blevins' formulation equation. The default is 4.5.
a3:	The a3 coefficient of Blevins' formulation equation. The default is -10.6.

Notes

(a) If the two structures are in separate models, then the specification of a Set Name is optional. By default, all of the elements in the current model constitute the downstream structure. If the two structures are in the same model, then the specification of a *Set Name* other than *All* is mandatory. In this case, the downstream structure cannot consist of all of the elements in that model.

USER-DEFINED DATA

Purpose

To specify the downstream structure to be used with the User-Defined Wake Model.

Table Input

Input:	Description
Set Name:	The name of the set that contains the elements on the downstream structure that you want to include in the wake interference analysis. The default is *All*. See Note (a).

Notes

(a) If the two structures are in separate models, then the specification of a *Set Name* is optional. By default, all of the elements in the current model constitute the downstream structure. If the two structures are in the same model, then the specification of a *Set Name* other than *All* is mandatory. In this case, the downstream structure cannot consist of all of the elements in that model.

DRAG COEFFICIENT DATA

Purpose

To specify drag coefficient data for the *User-Defined* wake model.

Table Input

You use this table to enter the drag coefficient data as functions of longitudinal (L) or transverse (T) centre-to-centre structure distances. The separations are normalised with respect to the drag diameter of the upstream structure. All values default to zero unless you specify otherwise. You may also customise the ranges to meet your own specific requirements.

Notes

(a) The coefficients may be defined in terms of absolute coefficient values (as was the case in earlier versions) or as coefficient factors (this is now the default option). These factors multiply or scale the (nominal) hydrodynamic coefficients to give the coefficients to be used in the calculation of the wake-affected hydrodynamic forces.

WAKE COEFFICIENTS TYPE

Purpose

To specify the type of wake coefficients to be used with the *User-Defined* wake interference model.

Table Input

Input:	Description
Coefficients Type:	The options are *Factors* (the default) and *Absolute*.

Notes

(a) The coefficients may be defined in terms of absolute coefficient values (as was the case in earlier versions) or as coefficient factors (this is now the default option). These factors multiply or scale the (nominal) hydrodynamic coefficients to give the coefficients to be used in the calculation of the wake-affected hydrodynamic forces.

LIFT COEFFICIENT DATA

Purpose

To specify lift coefficient data for the *User-Defined* wake model.

Table Input

You use this table to enter the lift coefficient data as functions of longitudinal (L) or transverse (T) centre-to-centre structure distances. The separations are normalised with respect to the drag diameter of the upstream structure. All values default to 0 unless you specify otherwise. You may also customise the ranges to meet your own specific requirements.

Notes

(a) The coefficients may be defined in terms of absolute coefficient values (as was the case in earlier versions) or as coefficient factors (this is now the default option). These factors multiply or scale the (nominal) hydrodynamic coefficients to give the coefficients to be used in the calculation of the wake-affected hydrodynamic forces.

WAKE LIFT COEFFICIENTS

Purpose

To define lift coefficients to be used with the *User-Defined* wake interference model.

Table Input

Input:	Description
Set Name:	The name of the element set.
Wake Lift Coefficient:	The corresponding wake lift coefficient.

Notes

(a) One important point to note regarding the lift coefficients is that they do not necessarily represent physically meaningful quantities. Unlike drag forces, lift forces outside of the wake zone are non-existent, so while drag coefficients have a physical significance in reality, lift coefficients are somewhat notional. For this reason, the specification of lift coefficients is grouped with the specification of wake interference data, in both the user interface and in the keyword file; whereas the specification of drag coefficients is (as always) part of the specification of the standard hydrodynamic properties.

1.8.9.57 *WAKE UPSTREAM

PURPOSE

To specify the composition of the upstream structure in terms of element sets, for use in wake interference calculations. The keyword also selects the wake interference model to be used, and defines associated data to characterise the wake field.

THEORY

Refer to Wake Interference for further information on this feature.

KEYWORD FORMAT

The keyword begins with a mandatory line selecting the wake interference model which can be HUSE, BLEVINS or USER and optional additional OUTPUT= which can be Yes or No. This is followed by an optional line to define the path to the Flexcom analysis file containing the upstream structure. This is followed by a mandatory line defining the set(s) of elements which comprise the upstream structure. In the case of a *User-Defined* wake interference model, the relevant set names and corresponding Drag and Lift coefficients are input through an external data file. As with the *Huse* and *Blevin's* models, the data entry in terms of set name, drag & lift coefficients may be repeated as many times as necessary to represent the upstream structure

Mandatory line selecting the wake interference model:

```
MODEL=Wake Effect Model [, OUTPUT=YES]
```

Optional line defining upstream structure analysis file:

```
[UPSTREAM_FILE=Full Path of Upstream Structure Analysis File]
```

Mandatory line defining upstream structure element set(s):

```
SET=Element Set Name
```

Huse’s formulation:

The element set name is followed by a single line with three coefficient values:

```
[k1, k2, k3]
```

Blevins’ formulation:

The element set name is followed by a single line with three coefficient values:

```
[a1, a2, a3]
```

The entire block of data (i.e. element set name and associated wake field data) may be repeated as many times as necessary to completely define the upstream structure.

User Defined formulation:

The USER defined Drag & Lift coefficients are input through a separate data file by a mandatory line; WAKE_USER_FILE= *Full path of USER Drag & Lift data*. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks. The format of the data file will be a line containing the element set name which is followed by blocks of lines which define the wake field in terms of drag and lift coefficient data. A first line with either TYPE=DRAG or TYPE=LIFT indicates which data is being input (they can be specified in either order). Note that the coefficient data represents coefficient factors rather than absolute coefficient values. The first line following TYPE= lists the transverse separation values at which the coefficient factors are defined, normalised with respect to the drag diameter of the upstream structure. Each line after that starts with a longitudinal separation value, again normalised with respect to the drag diameter of the upstream structure, followed by coefficient factors corresponding to the separation values. If L represents the centre-to-centre distance between upstream and downstream structures in the longitudinal direction (direction of undisturbed flow), and T is the centre-to-centre spacing in the transverse direction (normal to undisturbed flow); then the ith normalised separations are denoted li=Li/Dd and ti =Ti/Dd . The following then summarises the above:

```
TYPE=DRAG
    t1,        t2,        t3,        ......  tn
```

```
l1, Cdf(l1,t1), Cdf(l1,t2), Cdf(l1,t3) .......
Cdf(l1,tn)

l2, Cdf(l2,t1), Cdf(l2,t2), Cdf(l2,t3) .......
Cdf(l2,tn)

...

...

lm, Cdf(lm,t1), Cdf(lm,t2), Cdf(lm,t3) .......
Cdf(lm,tn)

TYPE=LIFT

    t1,        t2,        t3,        ....... tn

l1, Clf(l1,t1), Clf(l1,t2), Clf(l1,t3) .......
Clf(l1,tn)

l2, Clf(l2,t1), Clf(l2,t2), Clf(l2,t3) .......
Clf(l2,tn)

...

...

lm, Clf(lm,t1), Clf(lm,t2), Clf(lm,t3) .......
Clf(lm,tn)
```

A sample USER-Defined data file is as follows:

```
SET=Riser-UP-set1

 TYPE=DRAG

 -12.0, -8.0, -4.0, -2.0,  0.0,  2.0,  4.0,  8.0,
12.0

  0.2,  1.0,  1.0,  1.0,  0.1,  0.1,  0.1,  1.0,
1.0,  1.0

  1.0,  1.0,  0.6,  0.4,  0.2,  0.1,  0.2,  0.4,
0.6,  1.0

  2.0,  1.0,  0.6,  0.6,  0.4,  0.2,  0.4,  0.6,
0.6,  1.0
```

```
  4.0,  1.0,  0.8,  0.6,  0.6,  0.2,  0.6,  0.6,
0.8,  1.0
  8.0,  1.0,  1.0,  0.8,  0.8,  0.4,  0.8,  0.8,
1.0,  1.0
 12.0,  1.0,  1.0,  1.0,  0.8,  0.6,  0.8,  1.0,
1.0,  1.0
 TYPE=LIFT
 -12.0, -8.0, -4.0,  0.0,  4.0,  8.0,  12.0
 1.0,  0.0,  1.6,  0.8,  0.0,  0.8,  1.6,  0.0
 4.0,  0.0,  1.2,  0.6,  0.0,  0.6,  1.2,  0.0
 12.0,  0.0,  1.0,  0.5,  0.0,   0.5,  1.0,  0.0
 SET=Riser-UP-set2
TYPE=DRAG
 -12.0, -8.0, -4.0, -2.0,  0.0,  2.0,  4.0,  8.0,
12.0
  0.2,  1.0,  1.0,  1.0,  0.1,  0.1,  0.1,  1.0,
1.0,  1.0
  1.0,  1.0,  0.6,  0.4,  0.2,  0.1,  0.2,  0.4,
0.6,  1.0
  2.0,  1.0,  0.6,  0.6,  0.4,  0.2,  0.4,  0.6,
0.6,  1.0
  4.0,  1.0,  0.8,  0.6,  0.6,  0.2,  0.6,  0.6,
0.8,  1.0
  8.0,  1.0,  1.0,  0.8,  0.8,  0.4,  0.8,  0.8,
1.0,  1.0
 12.0,  1.0,  1.0,  1.0,  0.8,  0.6,  0.8,  1.0,
1.0,  1.0
 TYPE=LIFT
 -12.0, -8.0, -4.0,  0.0,  4.0,  8.0,  12.0
 1.0,  0.0,  1.6,  0.8,  0.0,  0.8,  1.6,  0.0
 4.0,  0.0,  1.2,  0.6,  0.0,  0.6,  1.2,  0.0
 12.0,  0.0,  1.0,  0.5,  0.0,   0.5,  1.0,  0.0
```

The use of `*WAKE UPSTREAM` is compulsory if wake interference effects are to be included in an analysis and may only be invoked in a static or quasi-static analysis. If the Upstream Structure Analysis File is included, it should be specified without extension. If it contains spaces, then it should be enclosed in double quotation marks. Element Set Name must be an existing element set, defined in the initial static analysis of the upstream structure. The specification of drag and lift data is required if Wake Effect Model is set to USER. The l and t values must be specified in ascending order.

The Upstream Structure Analysis File is the name of the file for the static or quasi-static analysis with current of the upstream structure. It is required if the two structures under consideration are being analysed in separate models. It is omitted where the upstream and downstream structures are included in the same model. The specification of the k1, k2 and k3 values for Huse's formulation is optional – these default to 0.25, 1 and 0.693 respectively. Likewise, the specification of the a1, a2 and a3 values for Blevins' formulation is optional – these default to 1, 4.5 and -10.6 respectively.

WAKE INTERFERENCE MODEL OPTION

Purpose

To specify the formulation to be used in calculating wake interference effects and request additional output from wake modelling calculations.

Table Input

Input:	Description
Formulation:	The options are *Huse's* (the default), *Blevins'* and *User-Defined*.
Output:	The options are *No* (the default) and *Yes*. See Note (b).

UPSTREAM STRUCTURE DATA

Purpose

To specify the path to the analysis file containing the upstream structure file if the two structures are modelled separately.

Table Input

Input:	Description
Upstream File:	Path of Upstream Structure Analysis File. See Note (c).

HUSE'S FORMULATION DATA

Purpose

To specify parameters relating to Huse's wake interference formulation.

Table Input

Input:	Set Name:	k1:	k2:	k3:
Description	The name of the set that contains the elements on the upstream structure that you want to include in the wake interference analysis with the corresponding coefficients. The default is *All*. Can be specified as many times as required.	The k1 coefficient of Huse's formulation equation. The default is 0.25.	The k2 coefficient of Huse's formulation equation. The default is 1.	The k3 coefficient of Huse's formulation equation. The default is 0.693.

BLEVIN'S FORMULATION DATA

Purpose

To specify parameters relating to Blevins' wake interference formulation.

Table Input

Input:	Set Name:	a1:	a2:	a3:
Description	The name of the set that contains the elements on the upstream structure that you want to include in the wake interference analysis with the corresponding coefficients. The default is All. Can be specified as many times as required.	The a1 coefficient of Blevins' formulation equation. The default is 1.	The a2 coefficient of Blevins' formulation equation. The default is 4.5	The a3 coefficient of Blevins' formulation equation. The default is -10.6.

USER DEFINED DATA

Purpose

To specify the USER defined Drag & Lift data file to be used with the User-Defined Wake Model.

Table Input

Input:	Description
User Wake File:	The file path of the USER defined data file containing Drag & Lift coefficients for upstream structure sets. See Note (d).

Notes

(a) Refer to Wake Interference for a detailed discussion of the wake interference modelling capabilities in Flexcom and the significance of all associated inputs.

(b) If you invoke this option, Flexcom presents detailed information regarding the wake modelling calculations. Specifically, for every integration point on every element of the downstream structure, the following is provided:

- Longitudinal (L) and transverse (T) centre-to-centre distance to upstream structure
- Reduced current velocity
- Drag and lift coefficients
- Lift Force

This information may be useful in some circumstances in understanding wake interference analysis results, but in general it is not required, so this option would be rarely invoked.

(c) This entry is optional, and is only relevant if the upstream and downstream structures are analysed in separate models.

(d) In the case of a *User-Defined* wake interference model, the relevant set names and corresponding Drag and Lift coefficients are input through an external data file. As with the *Huse* and *Blevin's* models, the data entry in terms of set name, drag & lift coefficients may be repeated as many times as necessary to represent the upstream structure.

1.8.9.58 *WAVE-DEANS

PURPOSE

To specify Dean's Stream regular wave loading.

THEORY

Refer to Deans Stream Wave for further information on this feature.

KEYWORD FORMAT

Blocks of lines which define the wave loading parameters.

Block defining a Dean's Stream Function wave:

```
[OPTION=SUPERPOSE]
Wave Height, Wave Period [, Direction] [, Eulerian Current Velocity] [, Order
Number of Ramp Steps] [, Equivalent Airy Wave Amplitude] [, Equivalent Airy W
```

A non-zero `Equivalent Airy Wave Period` with no `Equivalent Airy Wave Amplitude` (or an amplitude of 0) is invalid. `Order of Expansion`, `Max Iterations` and `Number of Ramp Steps` default to 15, 100 and 5, respectively.

TABLE INPUT

Dean's Stream - Multiple Order

Input:	Description
Wave Height:	The Dean's Stream wave height (crest to trough).
Wave Period:	The Dean's Stream wave period.
Direction:	The wave direction, measured anti-clockwise from the global-Y direction. All waves in Flexcom emanate from the origin. The default direction is 0°.

Eulerian Current Velocity:	The velocity of the underlying current. This value defaults to 0. See Note (b).
Order of Expansion:	The order of expansion of the Dean’s Stream Function equations which defines the wave. This value defaults to 15. See Note (c).
Max Iterations:	The maximum number of iterations to be used in solving the Dean’s Stream function equations. This value defaults to 100.
Number of Ramp Steps:	The number of ramp steps over which the wave height is applied. This value defaults to 5. See Note (d).

Dean's Stream - Equivalent Airy

Input:	Description
Wave Height:	The Dean’s Stream wave height (crest to trough).
Wave Period:	The Dean’s Stream wave period.
Direction:	The wave direction, measured anti-clockwise from the global-Y direction. All waves in Flexcom emanate from the origin. The default direction is 0°.
Eulerian Current Velocity:	The velocity of the underlying current. This value defaults to 0. See Note (b).
Order of Expansion:	The order of expansion of the Dean’s Stream Function equations which defines the wave. This value defaults to 15. See Note (c).

Max Iterations:	The maximum number of iterations to be used in solving the Dean's Stream function equations. This value defaults to 100.
Number of Ramp Steps:	The number of ramp steps over which the wave height is applied. This value defaults to 5. See Note (d).
Equivalent Airy Wave Amplitude:	The amplitude of an Airy wave used to determine vessel response to the Dean's Stream wave. The default value is 0. See Note (e).
Equivalent Airy Wave Period:	The period of the Airy wave used to determine vessel response to the Dean's Stream wave, if this feature is used. The default is the Dean's Stream wave period. See Note (e).

Dean's Stream - Superposition of Harmonics

Input:	Description
Wave Height:	The Dean's Stream wave height (crest to trough).
Wave Period:	The Dean's Stream wave period.
Direction:	The wave direction, measured anti-clockwise from the global-Y direction. All waves in Flexcom emanate from the origin. The default direction is 0°.
Eulerian Current Velocity:	The velocity of the underlying current. This value defaults to 0. See Note (b).

Order of Expansion:	The order of expansion of the Dean’s Stream Function equations which defines the wave. This value defaults to 15. See Note (c).
Max Iterations:	The maximum number of iterations to be used in solving the Dean’s Stream function equations. This value defaults to 100.
Number of Ramp Steps:	The number of ramp steps over which the wave height is applied. This value defaults to 5. See Note (d).

Notes

(a) If your analysis includes vessel response to a Dean’s Stream wave:

- Using the *Multiple Order* option means that the order of the response will be the same as the order of expansion of the Dean’s Stream Function.
- Using the *Equivalent Airy* option means that the response will be first order (sinusoidal).
- Using the *Superposition of Harmonics* option means that that response will be calculated as a superposition of N individual components, where N is the order of expansion of the Dean’s Stream function.

(b) The Eulerian current velocity corresponds to the current recorded by a stationary meter. The principal consequence of an underlying current is a Doppler shift of the apparent wave period.

(c) The order of expansion of the Dean’s stream function is essentially a measure of how non-linear a wave is. In deep water the order can be relatively low (between 3 and 5), while in shallow water the order can be as high as 30.

(d) The Fourier theory formulations used by Flexcom to solve the Dean’s Stream function equation can occasionally lead to multicrested solutions, particularly in shallow water. In order to eliminate this, the wave height is gradually increased to the required height in the number of ramp steps specified.

(e) You can specify an equivalent Airy wave amplitude without inputting a corresponding period, in which case the Dean's Stream wave period is used by default. Specification of an Airy wave period without a corresponding wave amplitude is invalid. Of course, if an analysis does not include vessel response to waves, equivalent Airy wave parameters are immaterial and unused.

1.8.9.59 *WAVE-GENERAL

PURPOSE

To specify miscellaneous parameters to wave loading.

THEORY

Refer to Spectrum Discretisation, Random Seed, Wave Kinematics and Selected Frequencies for further information on this feature.

KEYWORD FORMAT

Format: A number of optional lines to specify a seed value for the random number generator, the Airy wave kinematics algorithm, the extent of the wave zone, solution options relating to floating body analysis, and selected frequencies that Flexcom must include in the range of solution harmonics.

Line setting the discretisation random seed:

```
[SEED=Random Seed]
```

Line containing several optional entries relating to wave kinematics, allowing the user to select the kinematic stretching option, assign scale factors to water particle velocities and accelerations, and specify whether drag loading is to be based on absolute or relative velocities:

```
[OPTION=Kinematics Option], [WATER_PARTICLE_VEL_SCALE=Velocity Scale Factor],
```

Kinematics Option can be *MWL* or *EXTENDED*. If omitted the default of superposition stretching is used. *Velocity Scale Factor* and *Acceleration Scale Factor* both default to 1.0. *Drag Velocity* can be *RELATIVE* (the default) or *ABSOLUTE*.

A single line defining wave zone option:

```
[WAVEZONE=Wavelength Factor]
```

Wavelength Factor defaults to 0.5 if omitted.

A single line requesting the activation or suppression of frequency-dependent radiation damping forces on floating body via convolution integral:

```
[FLOAT_CONVOLUTION=Convolution Option]
```

A single line requesting the activation or suppression of first order wave loads from force RAOs on floating body:

```
[FLOAT_RAO_FORCES=RAO Option]
```

A single line requesting the activation or suppression of second order wave loads from force RAOs on floating body:

```
[FLOAT_QTF_FORCES=QTF Option]
```

Convolution Option may be *NO* (the default) or *YES*. *RAO Option* may be *NO* (the default) or *YES*. *QTF Option* may be *NO* (the default) or *YES*.

A block of two lines defining selected wave frequencies. The second line can be repeated as often as necessary. Defining selected wave frequencies is only valid in the frequency domain.

```
[FREQUENCIES=SELECTED]
Frequency
```

A single line defining an offset of the wave origin from Y=0.0, Z=0.0. Default is 0.0, 0.0 if unspecified.

```
[ORIGIN=Y,Z]
```

RANDOM NUMBER SEED

Purpose

To specify a seed value for the random number generator that assigns random phases to wave components when discretising a wave spectrum or spectra.

Table Input

Input:	Description
Random Number Seed:	A value to be used as the random number generator seed.

Notes

(a) Refer to Random Seed for further information on this feature.

WAVE KINEMATICS

Purpose

To select the kinematic stretching option, assign scale factors to water particle velocities and accelerations, and specify whether drag loading is to be based on absolute or relative velocities:

Table Input

Input:	Description
Kinematic stretching:	The options are *Superposition Stretching* (the default), *MWL Values Above MWL*, or *Extend MWL to Wave Surface*.
Velocity Scale Factor:	A scale factor to be applied to the water particle velocity terms (defaults to 1.0).
Acceleration Scale Factor:	A scale factor to be applied to the water particle acceleration terms (defaults to 1.0).
Drag Velocity:	An option to specify whether Morison's drag loading is to be based on the relative fluid-structure velocity (the default), or simply the absolute fluid velocity.

Notes

(a) Refer to Wave Kinematics and Water Particle Velocities and Accelerations for further information on these features.

WAVE ZONE

Purpose

To specify the extent of the wave zone, used in the computation of water particle velocities and accelerations.

Table Input

Input:	Description
Wavelength Factor:	The wave zone is considered to extend from the mean water line downwards by a distance of one wavelength times the specified factor, (which defaults to 0.5). See Note (a).

Notes

(a) Water particle velocities and accelerations due to the wave field are computed at each element integration point. As these velocities and accelerations decay exponentially with depth, for computational efficiency only the regular waves (or component harmonics in the case of random sea analysis) which are in the wave zone are considered to contribute towards the generation of water particle velocities and accelerations. The wave zone is considered to extend from the mean water line downwards by a distance of one wavelength times the specified *Wavelength Factor*. A value of 0.5 is assumed by default, so any wave whose half-wavelength is less than the distance from the mean water line to the integration point in question is omitted from water particle velocity and acceleration calculations.

FLOATING BODY - SOLUTION

Purpose

To specify solution options relating to floating body analysis.

Table Input

Input:	Description
Activate Convolution Integral:	This option allows you to specifically request that frequency-dependent radiation damping forces are computed via a convolution integral approach, even if the wave loading consists of a single regular wave only. The options are *No* (the default) and *Yes*. See Note (a).
Suppress First Order Wave Loads:	This option allows you to suppress the first order wave loading. The options are *No* (the default) and *Yes*.

Suppress Second Order Wave Loads:	This option allows you to suppress the second order wave loading. The options are No (the default) and Yes.

Notes

(a) Frequency-dependent radiation damping forces are computed via a convolution integral of velocity time history and retardation functions (this is standard modelling procedure for random sea analyses in the time domain). However, there are certain circumstances where a different approach is adopted. For example, Flexcom does not generally compute the damping forces in this manner if the wave loading consists of a single regular wave only, even if frequency-dependent radiation damping data has been specified. Typically the program finds a (constant) radiation damping matrix (by interpolation if necessary) corresponding to the single regular wave frequency, and adds this matrix at an appropriate location on the left-hand side of the equations of motion. However, you have the option to override this default behaviour, and compel Flexcom to perform the convolution calculation, in which case the radiation damping becomes a force on the right-hand side of the equations of motion as before. Refer to Wave Radiation Loads for further information on the theory underlying the coupled analysis technique in the time domain.

SELECTED FREQUENCIES

Purpose

To specify selected frequencies that Flexcom must include in the range of solution harmonics.

Table Input

Input:	Description
Frequency:	A frequency value, which Flexcom must include in the range of solution harmonics.

Notes

(a) This option is normally used to ensure Flexcom includes a solution at any natural frequency occurring in the range of the analysis harmonics. Refer to Selected Frequencies for further information.

(b) This table is only relevant for frequency domain dynamic analysis.

WAVE ORIGIN

Purpose

To specify an offset to be applied to the origin of the wave from Y=0.0, Z=0.0

Table Input

Input:	Description
Y Offset:	The Y coordinate of the wave origin point. Defaults to Y=0 if unspecified.
Z Offset:	The Z coordinate of the wave origin point. Defaults to Z=0 if unspecified.

1.8.9.60 *WAVE-JONSWAP

PURPOSE

To specify a JONSWAP random sea wave spectrum or spectra.

THEORY

Refer to Jonswap Wave for further information on this feature.

KEYWORD FORMAT

Blocks of lines which define the wave loading parameters.

Block defining Jonswap spectra in the normal format with equal area discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=AREA [, SPEC=NORMAL]
Peak Frequency, Philips Constant, Peakedness, Max. Freq.
```

```
Increment, Cut-off Freq., No. of Harmonics, Wave Directions
[, Dominant Direction] [, Wave Spreading Exponent]
```

Block defining Jonswap spectra in the normal format with geometric progression discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=GP [, SPEC=NORMAL]
Peak Frequency, Philips Constant, Peakedness, Cut-off Freq.,
Geometric Progression Factor, Wave Directions [, Dominant
Direction] [, Wave Spreading Exponent]
```

Block defining Jonswap spectra in the H_s/T_z format with equal area discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=AREA, SPEC=HSTZ
Hs, Tz, Max. Freq. Increment, Cut-off Freq., No. of Harmonics, Wave Directio
Exponent]
```

Block defining Jonswap spectra in the H_s/T_z format with geometric progression discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=GP, SPEC=HSTZ
Hs, Tz, Cut-off Freq., Geometric Progression Factor, Wave Directions [, Domi
[, Wave Spreading Exponent]
```

Block defining Jonswap spectra in the H_s/T_p format with equal area discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=AREA, SPEC=HSTP
Hs, Tp, Max. Freq. Increment, Cut-off Freq., No. of Harmonics, Wave Directio
[, Wave Spreading Exponent]
```

Block defining Jonswap spectra in the H_s/T_p format with geometric progression discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=GP, SPEC=HSTP
Hs, Tp, Cut-off Freq., Geometric Progression Factor, Wave Directions [, Domi
 Spreading Exponent]
```

Block defining Jonswap spectra in the H_s/T_p/Gamma format with equal area discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=AREA, SPEC=HSTPGAMMA
Hs, Tp, Peakedness, Max. Freq. Increment, Cut-off Freq., No. of Harmonics, Wa
[, Dominant Direction] [, Wave Spreading Exponent]
```

Block defining Jonswap spectra in the H_s/T_p/Gamma format with geometric progression discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=GP, SPEC=HSTPGAMMA
Hs, Tp, Peakedness, Cut-off Freq., Geometric Progression Factor, Wave Direct
Wave Spreading Exponent]
```

The `Wave Spreading Exponent` for random sea analyses must be an even integer and defaults to 2. The `Wave Spreading Exponent` is omitted for frequency domain analyses and should not be specified for this type of analysis.

WAVE - JONSWAP - STANDARD - EQUAL AREA

Purpose

To define a Jonswap random sea wave spectrum using the normal or standard format, to be discretised using equal areas.

Table Input

Input:	Description
Peak Frequency:	The spectrum peak frequency f_p in Hz.
Phillips Constant:	Phillips constant α, typically 0.0081 (the default).
Peakedness Parameter:	The spectrum peakedness parameter γ.
Max Frequency Increment:	The maximum frequency increment in Hz. to be used in the spectral discretisation. See Note (b). This parameter defaults to a value of 0.05 Hz.
Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
No. of Harmonics:	The number of harmonics to be used in the spectral discretisation.

Wave Directions :	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

WAVE - JONSWAP - STANDARD - GEOMETRIC PROGRESSION

Purpose

To define a Jonswap random sea wave spectrum using the normal or standard format, to be discretised using a geometric progression of frequencies.

Table Input

Input:	Description
Peak Frequency:	The spectrum peak frequency f_p in Hz.
Phillips Constant:	Phillips constant α, typically 0.0081 (the default).
Peakedness Parameter:	The spectrum peakedness parameter γ.
Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
Geometric Progression Factor:	The geometric progression factor for the spectral discretisation. See Note (b). This parameter defaults to a value of 0.02.

No. of Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

WAVE - JONSWAP - HS/TZ - EQUAL AREA

Purpose

To define a Jonswap random sea wave spectrum or spectra in terms of H_s and T_z, to be discretised using equal area increments.

Table Input

Input:	Description
Wave Height:	The spectrum significant wave height Hs.
Up-Crossing Period:	The mean zero up-crossing period T_z in seconds.
Max Frequency Increment:	The maximum frequency increment in Hz. to be used in the spectral discretisation. See Note (b). This parameter defaults to a value of 0.05 Hz.

Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
No. of Harmonics:	The number of harmonics to be used in the spectral discretisation.
Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

WAVE - JONSWAP - HS/TZ - GEOMETRIC PROGRESSION

Purpose

To define a Jonswap random sea wave spectrum or spectra in terms of Hs and Tz, to be discretised using a geometric progression of frequencies.

Table Input

Input:	Description
Wave Height:	The spectrum significant wave height Hs.

Up-Crossing Period:	The mean zero up-crossing period T_z in seconds.
Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
Geometric Progression Factor:	The geometric progression factor for the spectral discretisation. See Note (b). This parameter defaults to a value of 0.02.
No. of Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

WAVE - JONSWAP - HS/TP - EQUAL AREA

Purpose

To define a Jonswap random sea wave spectrum or spectra in terms of H_s and T_p, to be discretised using equal area increments.

Table Input

Input:	Description
Wave Height:	The spectrum significant wave height Hs.

Peak Period:	The spectrum peak period Tp in seconds.
Max Frequency Increment:	The maximum frequency increment in Hz. to be used in the spectral discretisation. See Note (b). This parameter defaults to a value of 0.05 Hz.
Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
No. of Harmonics:	The number of harmonics to be used in the spectral discretisation.
Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

HS/TP - GEOMETRIC PROGRESSION

Purpose

To define a Jonswap random sea wave spectrum or spectra in terms of H_s and T_p, to be discretised using a geometric progression of frequencies.

Table Input

Input:	Description
Wave Height:	The spectrum significant wave height Hs.
Peak Period:	The spectrum peak period Tp in seconds.
Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
Geometric Progression Factor:	The geometric progression factor for the spectral discretisation. See Note (b). This parameter defaults to a value of 0.02.
No. of Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

HS/TP/GAMMA - EQUAL AREA

Purpose

To define a Jonswap random sea wave spectrum or spectra in terms of H_s, T_p and γ, to be discretised using equal area increments.

Table Input

Input:	Description
Wave Height:	The spectrum significant wave height Hs.
Peak Period:	The spectrum peak period Tp in seconds.
Peakedness Parameter:	The spectrum peakedness parameter γ.
Max Frequency Increment:	The maximum frequency increment in Hz. to be used in the spectral discretisation. See Note (b). This parameter defaults to a value of 0.05 Hz.
Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
No. of Harmonics:	The number of harmonics to be used in the spectral discretisation.
Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

HS/TP/GAMMA - GEOMETRIC PROGRESSION

Purpose

To define a Jonswap random sea wave spectrum or spectra in terms of H_s, T_p and γ, to be discretised using a geometric progression of frequencies.

Table Input

Input:	Description
Wave Height:	The spectrum significant wave height Hs.
Peak Period:	The spectrum peak period Tp in seconds.
Peakedness Parameter:	The spectrum peakedness parameter γ.
Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
Geometric Progression Factor:	The geometric progression factor for the spectral discretisation. See Note (b). This parameter defaults to a value of 0.02.
No. of Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.

Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

Notes

(a) A random sea analysis can consider combinations of wave spectra and/or regular waves. Refer to Wave Loading for a description of the available wave specification combinations.

(b) The wave spectrum may be discretised into segments based on an equal area approach (which divides the area under the spectrum into segments of equal area) or a geometric progression approach (based on frequency increments that form a geometric progression). Refer to Spectrum Discretisation for a detailed discussion of this discretisation procedure.

(c) A multi-directional random sea is defined in terms of a dominant wave direction and the number of wave directions. Refer to Wave Energy Spreading for further information on this feature.

1.8.9.61 *WAVE-OCHI-HUBBLE

PURPOSE

To specify an Ochi-Hubble random sea wave spectrum or spectra.

THEORY

Refer to Ochi Hubble Wave for further information on this feature.

KEYWORD FORMAT

Blocks of lines which define the wave loading parameters.

Block defining Ochi-Hubble spectra with equal area discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=AREA
Hs1, Tp1, λ1, Hs2, Tp2, l2, Max. Freq. Increment, Cut-off
Freq., No. of Harmonics, Wave Directions [, Dominant
```

```
Direction] [, Wave Spreading Exponent]
```

Block defining Ochi-Hubble spectra with geometric progression discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=GP
Hs1, Tp1, l1, Hs2, Tp2, λ2, Cut-off Freq., Geometric
Progression Factor, Wave Directions [, Dominant Direction]
[, Wave Spreading Exponent]
```

The *Wave Spreading Exponent* for random sea analyses must be an even integer and defaults to 2. The *Wave Spreading Exponent* is omitted for frequency domain analyses and should not be specified for this type of analysis.

EQUAL AREA

Purpose

To define an Ochi-Hubble random sea wave spectrum or spectra, to be discretised using equal area increments.

Table Input

Input:	Description
Hs_1:	The significant wave height for the lower frequency components.
Tp_1:	The peak period for the lower frequency components.
λ_1:	The shape factor (λ_1) for the lower frequency components.
Hs_2:	The significant wave height for the higher frequency components.
Tp_2:	The peak period for the higher frequency components.
λ**2:**	The shape factor (λ_2) for the higher frequency components.
Max Frequency Increment:	The maximum frequency increment in Hz. to be used in the spectral discretisation. See Note (b). This parameter defaults to a value of 0.05 Hz.

Cut – off Frequency :	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
No. of Harmonics :	The number of harmonics to be used in the spectral discretisation.
Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

OCHI-HUBBLE - GEOMETRIC PROGRESSION

Purpose

To define an Ochi-Hubble random sea wave spectrum or spectra, to be discretised using a geometric progression of frequencies.

Table Input

Input:	Description
Hs_1:	The significant wave height for the lower frequency components.
Tp_1:	The peak period for the lower frequency components.
λ_1:	The shape factor (λ_1) for the lower frequency components.
Hs_2:	The significant wave height for the higher frequency components.

Tp_2:	The peak period for the higher frequency components.
λ_2:	The shape factor (λ_2) for the higher frequency components.
Cut – off Frequency :	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
Geometric Progression Factor:	The geometric progression factor for the spectral discretisation. See Note (b). This parameter defaults to a value of 0.02.
No. of Wave Directions :	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

NOTES

(a) A random sea analysis can consider combinations of wave spectra and/or regular waves. Refer to Wave Loading for a description of the available wave specification combinations.

(b) The wave spectrum may be discretised into segments based an equal area approach (which divides the area under the spectrum into segments of equal area) or a geometric progression approach (based on frequency increments that form a geometric progression). Refer to Spectrum Discretisation for a detailed discussion of this discretisation procedure.

(c) A multi-directional random sea is defined in terms of a dominant wave direction and the number of wave directions. Refer to Wave Energy Spreading for further information on this feature.

1.8.9.62 *WAVE-PIERSON-MOSKOWITZ

PURPOSE

To specify a Pierson-Moskowitz random sea wave spectrum or spectra.

THEORY

Refer to Pierson-Moskowitz Wave for further information on this feature.

KEYWORD FORMAT

Format: Blocks of lines which define the wave loading parameters.

Block defining Pierson-Moskowitz spectra in the H_s/T_z format with equal area discretisation.

The second line can be repeated as often as necessary:

```
FREQUENCY=AREA [, SPEC=HSTZ]
Hs, Tz, Max. Freq. Increment, Cut-off Freq., No. of Harmonics, Wave Directio
[, Dominant Direction] [, Wave Spreading Exponent]
```

Block defining Pierson-Moskowitz spectra in the H_s/T_z format with geometric progression

discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=GP [, SPEC=HSTZ]
Hs, Tz, Cut-off Freq., Geometric Progression Factor, Wave Directions [,
 Dominant Direction] [, Wave Spreading Exponent]
```

Block defining Pierson-Moskowitz spectra in the H_s/T_p format with equal area discretisation.

The second line can be repeated as often as necessary:

```
FREQUENCY=AREA, SPEC=HSTP
Hs, Tp, Max. Freq. Increment, Cut-off Freq., No. of Harmonics, Wave Directio
[, Dominant Direction] [, Wave Spreading Exponent]
```

Block defining Pierson-Moskowitz spectra in the H_s/T_p format with geometric progression

discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=GP, SPEC=HSTP
Hs, Tp, Cut-off Freq., Geometric Progression Factor, Wave Directions [,
Dominant Direction] [, Wave Spreading Exponent]
```

The `Wave Spreading Exponent` for random sea analyses must be an even integer and defaults to 2. The `Wave Spreading Exponent` is omitted for frequency domain analyses and should not be specified for this type of analysis.

WAVE - PIERSON - MOSKOWITZ - HS/TZ - EQUAL AREA

Purpose

To define a Pierson-Moskowitz random sea wave spectrum or spectra in terms of H_s and T_z, to be discretised using equal area increments.

Table Input

Input:	Description
Wave Height:	The spectrum significant wave height Hs.
Up-Crossing Period:	The mean zero up-crossing period T_z in seconds.
Max Frequency Increment :	The maximum frequency increment in Hz. to be used in the spectral discretisation. See Note (b). This parameter defaults to a value of 0.05 Hz.
Cut – off Frequency :	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
No. of Harmonics:	The number of harmonics to be used in the spectral discretisation.
Wave Directions :	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).

Input:	Description
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

WAVE - PIERSON - MOSKOWITZ - HS/TZ- GEOMETRIC PROGRESSION

Purpose

To define a Pierson-Moskowitz random sea wave spectrum or spectra in terms of H_s and T_z, to be discretised using a geometric progression of frequencies.

Table Input

Input:	Description
Wave Height:	The spectrum significant wave height Hs.
Up-Crossing Period:	The mean zero up-crossing period T_z in seconds.
Cut – off Frequency :	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
Geometric Progression Factor:	The geometric progression factor for the spectral discretisation. See Note (b). This parameter defaults to a value of 0.02.

No. of Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

WAVE - PIERSON - MOSKOWITZ - HS/TP - EQUAL AREA

Purpose

To define a Pierson-Moskowitz random sea wave spectrum or spectra in terms of H_s and T_p, to be discretised using equal area increments.

Table Input

Input:	Description
Wave Height:	The spectrum significant wave height Hs.
Peak Period:	The spectrum peak period Tp in seconds.
Max Frequency Increment:	The maximum frequency increment in Hz. to be used in the spectral discretisation. See Note (b). This parameter defaults to a value of 0.05 Hz.
Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
No. of Harmonics:	The number of harmonics to be used in the spectral discretisation.

Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

WAVE - PIERSON - MOSKOWITZ - HS/TP - GEOMETRIC PROGRESSION

Purpose

To define a Pierson-Moskowitz random sea wave spectrum or spectra in terms of H_s and T_p, to be discretised using a geometric progression of frequencies.

Table Input

Input:	Description
Wave Height:	The spectrum significant wave height Hs.
Peak Period:	The spectrum peak period Tp in seconds.
Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.

Geometric Progression Factor:	The geometric progression factor for the spectral discretisation. See Note (b). This parameter defaults to a value of 0.02.
No. of Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).

Notes

(a) A random sea analysis can consider combinations of wave spectra and/or regular waves. Refer to Wave Loading for a description of the available wave specification combinations.

(b) The wave spectrum may be discretised into segments based an equal area approach (which divides the area under the spectrum into segments of equal area) or a geometric progression approach (based on frequency increments that form a geometric progression). Refer to Spectrum Discretisation for a detailed discussion of this discretisation procedure.

(c) A multi-directional random sea is defined in terms of a dominant wave direction and the number of wave directions. Refer to Wave Energy Spreading for further information on this feature.

1.8.9.63 *WAVE-REGULAR

PURPOSE

To specify regular Airy wave loading.

THEORY

Refer to Regular Airy Wave for further information on this feature.

KEYWORD FORMAT

Blocks of lines which define the wave loading parameters.

Block defining regular Airy waves for a time domain dynamic analysis. The line can be repeated as often as necessary:

```
Amplitude, Wave Period [, Direction] [, Phase]
```

Block defining a single regular Airy wave for a frequency domain dynamic analysis:

```
OPTION=ONE
Amplitude, Wave Period [, Direction]
```

Block defining multiple regular Airy waves for a frequency domain dynamic analysis:

```
OPTION=MANY
Amplitude[, Direction], Lower Period, Upper Period, No of Periods
```

WAVE - REGULAR AIRY

Purpose

To specify the amplitude, period, direction and phase of regular Airy waves.

Table Input

Input:	Description
Amplitude:	The regular Airy wave amplitude (MWL to crest or trough).
Wave Period:	The regular Airy wave period in seconds.

Direction:	The wave direction, measured in degrees anti-clockwise from the global Y-direction. All waves in Flexcom emanate from the origin. The default direction is 0°.
Phase:	The wave phase angle in degrees. This input is appropriate only if more than one regular wave is specified. The default phase value is 0°.

WAVE - MULTIPLE REGULAR AIRY

Purpose

To specify the amplitude, period and direction of regular Airy waves for frequency domain analysis.

Table Input

Input:	Description
Amplitude:	The regular Airy wave amplitude (MWL to crest or trough). This wave amplitude is used with all solution harmonic components.
Direction:	The wave direction, measured in degrees anti-clockwise from the global Y-direction. All waves in Flexcom emanate from the origin. The default direction is 0°.
Lower Period:	The lower limit of the period range of interest, in seconds.
Upper Period:	The upper limit of the period range of interest, in seconds.
No. of Periods:	The number of periods at which solutions are found including the upper and lower periods.

Notes

(a) You can specify more than one regular Airy wave in a time domain dynamic analysis. You simply use one line of the *Wave - Regular Airy* table for each wave harmonic. You can also combine one or more Airy waves with a wave spectrum in the time domain.

(b) If you wish to specify more than one Airy wave in a frequency domain dynamic analysis, you must use the *Frequency Domain – Wave – Regular Airy* table. It is possible to specify multiple wave harmonics using the *Wave - Regular Airy* table, but only the first will be applied, and all subsequent ones will be ignored.

(c) As an example of the operation of the *Frequency Domain – Wave – Regular Airy* facility consider the following specification: unit amplitude; a direction of 0º; lower and upper periods of 5s and 25s respectively; and 21 periods specified. Then solutions will be calculated at the following periods: 5s, 6s, 7s ... 24s and 25s. However the solutions at individual harmonics are independent, unlike the situation in a random sea analysis where the solutions at all the harmonics are linked.

1.8.9.64 *WAVE-STOKES

PURPOSE

To specify Stokes V regular wave loading.

THEORY

Refer to Stokes V Wave for further information on this feature.

KEYWORD FORMAT

Blocks of lines which define the wave loading parameters.

Block defining a regular Stokes wave:

```
[OPTION=SUPERPOSE]
Height, Wave Period [, Direction] [, Equivalent Airy Wave Amplitude] [, Equiv
```

A non-zero `Equivalent Airy Wave Period` with no `Equivalent Airy Wave Amplitude` (or an amplitude of 0) is invalid.

TABLE INPUT

Stokes – 5th Order

Input:	Description
Wave Height:	The Stokes V wave height (crest to trough).
Wave Period:	The Stokes V wave period.
Direction:	The wave direction, measured anti-clockwise from the global-Y direction. All waves in Flexcom emanate from the origin. The default direction is 0°.

Stokes – Equivalent Airy

Input:	Description
Wave Height:	The Stokes V wave height (crest to trough).
Wave Period:	The Stokes V wave period.
Direction:	The wave direction, measured anti-clockwise from the global-Y direction. All waves in Flexcom emanate from the origin. The default direction is 0°.
Equivalent Airy Wave Amplitude:	The amplitude of the Airy wave used to determine vessel response to the Stokes V wave.
Equivalent Airy Wave Period:	The period of the Airy wave used to determine vessel response to the Stokes V wave. The default is the Stokes V wave period.

Stokes – Superposition of Harmonics

Input:	Description

Wave Height:	The Stokes V wave height (crest to trough).
Wave Period:	The Stokes V wave period.
Direction:	The wave direction, measured anti-clockwise from the global-Y direction. All waves in Flexcom emanate from the origin. The default direction is 0°.

NOTES

(a) If your analysis includes vessel response to a Stokes V wave:

- Using the Fifth Order option means that the order of the response will be 5^{th} order.
- Using the Equivalent Airy Wave option means that the response will be first order (sinusoidal).
- Using the Superposition of Harmonics option means that that response will be calculated as a superposition of 5 individual components.

(b) You can specify an equivalent Airy wave amplitude without inputting a corresponding period, in which case the Stokes V wave period is used by default. Specification of an Airy wave period without a corresponding wave amplitude is invalid. Of course if an analysis does not include vessel response to waves, equivalent Airy wave parameters are immaterial and unused.

1.8.9.65 *WAVE-TIME-HISTORY

PURPOSE

To specify a random seastate in terms of a time history of water surface elevation.

THEORY

Refer to Time History of Water Surface Elevation for further information on this feature.

KEYWORD FORMAT

Block of lines defining a wave input as a water surface elevation time history:

```
FILE=File Name
[Direction] [, Y0] [, Z0] [, Time Step], ENSEMBLES=No. of Ensembles
```

Defining a water surface history is only valid in the time domain. `File Name` should include the entire path of the water surface elevation file with its extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks. *Y0* and *Z0* default to 0. `Time Step` is mandatory if a water surface elevation time history is at uneven intervals. `No. of Ensembles` is a required input which must be greater than 0.

TABLE INPUT

Input:	Description
File Name:	The name of the ASCII data file containing the wave elevation time history. File Name should ideally include the entire path of the wave time history file, including extension. If the name or any part of its path contains spaces, the full File Name should be enclosed in double quotation marks (" "). The required format of the data file is described in Note (a).
Wave Direction	The wave direction, measured in degrees anti-clockwise from the global Y-direction. The default value is 0°.
Y-Offset	The offset of the wave source in the global Y-direction. The default value is 0. See Note (b).
Z-Offset	The offset of the wave source in the global Z-direction. The default value is 0. See Note (b).
Time Step	The time step to be used in interpolating a time history with a fixed time step when the data in the file has a variable time step.
Ensembles:	The number of ensembles to be used in the processing of the wave elevation time history. See Note (c).

NOTES

(a) The format of the wave elevation timetrace file is as follows.

- The file contains one line of data per time value, with two values per line. The first value is the actual time. The second value is the water surface elevation at that time.
- Comment lines, denoted by a capital 'C' in the first column, are permitted.
- Lines that are completely blank are ignored.

(b) By default, the time history is presumed to give the water surface elevation at the vertical axis Y=Z=0, but you can use the *Y-Offset* and *Z-Offset* values above to change from the default.

(c) Flexcom uses a Discrete Fourier Transform algorithm to process the wave time history data. Refer to Time History of Water Surface Elevation for further information on this feature.

1.8.9.66 *WAVE-TORSETHAUGEN

PURPOSE

To specify a Torsethaugen random sea wave spectrum or spectra.

THEORY

Refer to Torsethaugen Wave for further information on this feature.

KEYWORD FORMAT

Blocks of lines which define the wave loading parameters.

Block defining Torsethaugen spectra with equal area discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=AREA
Hs, Tp, Max. Freq. Increment, Cut-off Freq., No. of
Harmonics, Wave Directions [, Dominant Direction] [, Wave
Spreading Exponent] [, Fetch Factor]
```

Block defining Torsethaugen spectra with geometric progression discretisation. The second line can be repeated as often as necessary:

```
FREQUENCY=GP
```

Hs, Tp, Cut-off Freq., Geometric Progression Factor, Wave Directions [, *Dominant Direction*] [, *Wave Spreading Exponent*] [, *Fetch Factor*]

The *Wave Spreading Exponent* for random sea analyses must be an even integer and defaults to 2. The *Wave Spreading Exponent* is omitted for frequency domain analyses and should not be specified for this type of analysis.

WAVE-TORSETHAUGEN - EQUAL AREA

Purpose

To define a Torsethaugen random sea wave spectrum or spectra, to be discretised using equal area increments.

Table Input

Input:	Description
Wave Height:	The spectrum significant wave height Hs.
Peak Period:	The spectrum peak period Tp in seconds.
Max Frequency Increment:	The maximum frequency increment in Hz. to be used in the spectral discretisation. See Note (b). This parameter defaults to a value of 0.05 Hz.
Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
No. of Harmonics:	The number of harmonics to be used in the spectral discretisation.
Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).

Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).
Fetch Factor	Factor dependant on fetch. This entry defaults to 6.6. See Note (d).

WAVE-TORSETHAUGEN - GEOMETRIC PROGRESSION

Purpose

To define a Torsethaugen random sea wave spectrum or spectra, to be discretised using a geometric progression of frequencies.

Table Input

Input:	Description
Wave Height:	The spectrum significant wave height Hs.
Peak Period:	The spectrum peak period Tp in seconds.
Cut – off Frequency:	The cut-off or Nyquist frequency in Hz. This parameter defaults to a value of 0.5 Hz.
Geometric Progression Factor:	The geometric progression factor for the spectral discretisation. See Note (b). This parameter defaults to a value of 0.02.
No. of Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (c).

Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (c).
Fetch Factor	Factor dependant on fetch. This entry defaults to 6.6. See Note (d).

NOTES

(a) A random sea analysis can consider combinations of wave spectra and/or regular waves. Refer to Wave Loading for a description of the available wave specification combinations.

(b) The wave spectrum may be discretised into segments based an equal area approach (which divides the area under the spectrum into segments of equal area) or a geometric progression approach (based on frequency increments that form a geometric progression). Refer to Spectrum Discretisation for a detailed discussion of this discretisation procedure.

(c) A multi-directional random sea is defined in terms of a dominant wave direction and the number of wave directions. Refer to Wave Energy Spreading for further information on this feature.

(d) The suggested value of 6.6 for fetch factor relates to the metric unit system, in which a factor of 6.6 $m^{-1/3}s$ is typically used. If you are using the imperial unit system, you should adjust the fetch factor accordingly.

1.8.9.67 *WAVE-USER-DEFINED

PURPOSE

To specify a User-Defined random sea wave spectrum or spectra.

THEORY

Refer to User-Defined Wave Spectrum for further information on this feature.

KEYWORD FORMAT

Blocks of lines which define the wave loading parameters.

Block defining a user-specified spectrum with equal area discretisation. The third line can be repeated as often as necessary:

```
FREQUENCY=AREA
Max Freq. Increment, No. of Harmonics, Wave Directions [, Dominant Direction]
Frequency, Spectral Value
```

Block defining a user-specified spectrum with geometric progression discretisation. The third line can be repeated as often as necessary:

```
FREQUENCY=GP
Geometric Progression Factor, Wave Directions [, Dominant Direction] [, Wave
Frequency, Spectral Value
```

The `Wave Spreading Exponent` for random sea analyses must be an even integer and defaults to 2. The `Wave Spreading Exponent` is omitted for frequency domain analyses and should not be specified for this type of analysis.

WAVE - USER SPECTRUM

Purpose

To define general wave spectra as a series of frequency/spectral ordinate pairs.

Table Input

Input:	Description
Spectrum Name:	The name of the user-defined spectrum. See Note (b).
Frequency:	A frequency value in Hz.
Spectral Value:	The corresponding value of the wave spectrum.

WAVE - USER SPECIFIED - EQUAL AREA

Purpose

To define a number of parameters relating to the specification of a general wave spectrum to be discretised using equal areas.

Table Input

Input:	Description
Spectrum Name:	The name of the user-defined spectrum. See Note (b).
Max Frequency Increment:	The maximum frequency increment in Hz. to be used in the spectral discretisation. See Note (c). This parameter defaults to a value of 0.05 Hz.
No. of Harmonics:	The number of harmonics to be used in the spectral discretisation.
Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (d).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (d).

WAVE - USER SPECIFIED - GEOMETRIC PROGRESSION

Purpose

To define a number of parameters relating to the specification of a general wave spectrum to be discretised using a geometric progression of frequencies.

Table Input

Input:	Description
Spectrum Name:	The name of the user-defined spectrum. See Note (b).
Geometric Progression Factor:	The geometric progression factor for the spectral discretisation. See Note (c). This parameter defaults to a value of 0.02.
No. of Wave Directions:	The number of wave directions. The default of 1 gives a uni-directional random sea, greater than 1 gives a multi-directional sea. See Note (d).
Dominant Direction:	The wave direction in a uni-directional sea, or the dominant wave direction in a multi-directional random sea, measured in degrees anticlockwise from the global Y direction. The default is 0°.
Wave Spreading Exponent:	The exponent used in distributing wave energy between directions in a multi-directional random sea. This entry defaults to 2. See Note (d).

NOTES

(a) A random sea analysis can consider combinations of wave spectra and/or regular waves. Refer to Wave Loading for a description of the available wave specification combinations.

(b) The specification of data when you want to input a wave spectrum directly, that is as a series of (frequency, spectral ordinate) data pairs, is in two parts. Naturally the most important part is the actual series of values. You input this using the User Spectrum table. The second part of the data for a user-specified spectrum comprises general parameters relating to the spectrum discretisation. This data is input using either the *User Spectrum – Equal Area* or *User Spectrum – Geometric Progression* table. Note that the *Spectrum Name* inputs are used to link each spectrum with its associated discretisation parameters.

(c) The wave spectrum may be discretised into segments based an equal area approach (which divides the area under the spectrum into segments of equal area) or a geometric progression approach (based on frequency increments that form a geometric progression). Refer to Spectrum Discretisation for a detailed discussion of this discretisation procedure.

(d) A multi-directional random sea is defined in terms of a dominant wave direction and the number of wave directions. Refer to Wave Energy Spreading for further information on this feature.

1.8.9.68 *WINCH

PURPOSE

To define winch elements.

THEORY

Refer to Winch Elements for further information on this feature.

KEYWORD FORMAT

Data is specified in blocks of two lines repeated as many times as necessary to define the properties of each winch element set. The format used depends on the type of analysis being carried out.

In the case of a static analysis the following block is used:

```
SET=Element Set
LENGTH=Added Length
```

In the case of a time domain dynamic or quasi-static analysis, either of the following two blocks can be used:

For a standard velocity profile:

```
SET=Element Set
Winching Velocity, Ramp-up Start, Ramp-up End, Ramp-down Start, Ramp-down Enc
```

For winch payout based on a timetrace file:

```
SET=Element Set
FILE=File Name
```

The block of data used must be in accordance with the type of analysis being carried out. So the first block can only be used for static analyses. Likewise the second or third blocks can only be used for time domain dynamic or quasi-static analyses. The winching time sequence must be specified in ascending order. This keyword is not relevant for frequency domain dynamic analysis.

The `Added Length` is the length by which by the winch element set is to expand or contract (the value may be positive or negative) during the course of the static analysis.

`File Name` should include the entire path of the winch time history file including its extension. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

TABLE INPUT

Winch Elements – Static Analysis

Input:	Description
Set Name:	The element set to which the winch element properties are to be assigned.
Added Length:	The length by which by the winch element set is to expand or contract (the value may be positive or negative) during the course of an analysis.

Winch Elements – Dynamic Analysis – Velocity

Input:	Description
Set Name:	The element set to which the winch element properties are to be assigned.
Velocity:	The maximum pay-out (or reel-in) velocity of the winch elements.
Ramp-up Start:	The start time of ramp-up in the winching time sequence.

Ramp-up End:	The end time of ramp-up in the winching time sequence.
Ramp-down Start:	The start time of ramp-down in the winching time sequence.
Ramp-down End:	The end time of ramp-down in the winching time sequence.

Winch Elements – Dynamic Analysis – Time History

Input:	Description
Set Name:	The element set to which the winch element properties are to be assigned.
File Name:	The name of the file which contains the time history of winch element payout. See note (b).

NOTES

(a) Winch elements are typically normal beam elements, and are defined in the same manner as all beam elements, but are unique in that their lengths may be dynamically modified during an analysis. Refer to Winch Elements for further information on this feature.

(b) The winch timetrace file consists of rows of time and winch length data. The first value on each line is the solution time and the second is the associated change in length of the winch set from the original length of the set. The two values on each line should be separated by a space(s) and/or a comma. Blank lines and comment lines (lines where "C" is the first character on the line) will be ignored. Rows must be specified in ascending time order and cubic spline interpolation is used to calculate winch lengths for solution times between those specified in the winch timetrace file. An example of the winch timetrace file format is shown below. Say, for a winch set with original length of 10m, the following file describes a scenario where the winch set length remains at 10m until it ramps up to 15m (original + change) from 20s.

```
C Time   Length Change
  1,        0.0
  10,       0.0
  20,       5.0
  30,       5.0
```

```
40,       5.0
```

1.8.9.69 *WIND

PURPOSE

To specify wind loading. This is a legacy keyword which relates specifically to a moored vessel or a floating body. To apply wind loading to to an offshore wind turbine, please use the `*AERODYN DRIVER` keyword for steady wind or the `*INFLOWWIND` keyword for a more advanced wind field definition.

THEORY

Refer to Wind Loads and Wind Force Coefficients for further information on this feature.

KEYWORD FORMAT

One line defining the wind loading:

```
Density of Air, Wind Velocity, Wind Direction
```

The keyword is only relevant to analyses which included a moored vessel or a floating body.

TABLE INPUT

Input:	Description
Density of Air:	The density of the air.
Wind Velocity:	The velocity of the wind.
Wind Direction:	The wind direction measured in degrees anticlockwise from the global Y-axis. The default value is 0°.

1.8.9.70 *WIND COEFF

PURPOSE

To specify wind coefficients used to determine the wind loading on a floating body or moored vessel.

THEORY

Refer to Wind Loads and Wind Force Coefficients for further information on this feature.

KEYWORD FORMAT

Floating Body

A block of data consisting of floating body name followed by a single line defining the name of an external file which contains wind coefficient data.

```
FLOATING BODY=Floating Body Name
FILE=File Name
```

File Name should include the entire path of the file including its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

A block of data consisting of floating body name followed by a single line repeated as often as necessary to define the wind coefficients over a sufficiently large range of headings. The block itself may be repeated for different floating bodies.

```
FLOATING BODY=Floating Body Name
Floating Body/Wind Heading, Wind Coefficient in Surge,
Wind Coefficient in Sway, Wind Coefficient in Yaw
```

Moored Vessel

A block of data consisting of a moored vessel tag line followed by a single line defining the name of an external file which contains wind coefficient data.

```
MOORED VESSEL
FILE=File Name
```

File Name should include the entire path of the file including its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

A block of data consisting of a moored vessel tag line followed by a single line repeated as often as necessary to define the wind coefficients over a sufficiently large range of headings. Only one block is required, as a mooring analysis only contains one moored vessel.

```
MOORED VESSEL
Moored Vessel/Wind Heading, Wind Coefficient in Surge,
Wind Coefficient in Sway, Wind Coefficient in Yaw
```

TABLE INPUT

Floating Body (External File)

Input:	Description
Floating Body:	The name of the floating body.
File Name:	The name of the external data file.

Floating Body

Input:	Description
Floating Body:	The name of the floating body.
Heading:	The wind heading in degrees.
Cw_y:	The wind coefficient in surge, corresponding to the specified heading.
Cw_z:	The wind coefficient in sway, corresponding to the specified heading.
Cw_θ:	The wind coefficient in yaw, corresponding to the specified heading.

Moored Vessel (External File)

Input:	Description

File Name:	The name of the external data file.

Moored Vessel

Input:	Description
Heading:	The wind heading in degrees.
Cw_y:	The wind coefficient in surge, corresponding to the specified heading.
Cw_z:	The wind coefficient in sway, corresponding to the specified heading.
Cw_θ:	The wind coefficient in yaw, corresponding to the specified heading.

NOTES

(a) If the equivalent wind incidence does not exactly match one of the relative wind headings specified, Flexcom linearly interpolates between the nearest user-specified headings to find the relevant values of the wind coefficients.

(b) If the equivalent wind incidence is outside the range of user-specified headings, the wind coefficients are assumed to be zero. For this reason, it is recommended that you specify wind coefficients over a sufficiently large range of headings to ensure that wind coefficients are defined for any heading likely to arise in an analysis.

(c) The headings for the wind coefficients need not necessarily be specified in any particular order (ascending order is not necessary).

1.8.10 $MODEL

This section includes data such as the finite element discretisation, structural and hydrodynamic properties, plus any other inputs which characterise the initial model configuration (e.g. initial vessel position, seabed properties, ocean depth etc.) – basically any information which cannot logically change from run to run.

This section contains the following keywords:

- *ADDED MASS is used to define added mass for a floating body.
- *AUXILIARY is used to specify the auxiliary elements which comprise an auxiliary body.
- *BENDING HYSTERESIS is used to define hysteresis moment-curvature backbone curves for non-linear materials.
- *BLADE GEOMETRY is used to specify the geometry of wind turbine blades.
- *BLADE STRUCTURE is used to specify the structural properties of wind turbine blades.
- *BODY,INTEGRATED is used to add subsea components to the structural animation for enhanced visual appeal.
- *CABLE is used to specify that part of the structure forms a cable between specified end points.
- *CABLE BUNDLE is used to specify that two (or more) cables are to be "bundled" together.
- *CALM MODEL is used to specify the properties of calm buoys.
- *CLASHING is used to specify regions where clashing may occur, and suitable contact stiffness and damping values.
- *COATINGS is used to apply coatings on the basis of element set.
- *COLOUR DEFINE is used to define a specific colour using RGB colour coding.
- *DAMPER is used to define damper elements in the structural discretisation.
- *DAMPER DATA is used to define time-varying coefficients for use with damper elements.
- *DRAG CHAIN is used to include drag chains in the structural model.
- *DRAG LIFT is used to define non-linear drag lift relationships.
- *ELASTIC SURFACE is used to specify the properties of an elastic seabed.
- *ELEMENT is used to specify the finite element connectivity of the structural model.

- *ELEMENT,AUXILIARY is used to specify the connectivity of auxiliary elements in the model.
- *ELEMENT SETS is used to group individual elements into element sets.
- *EMBEDMENT is used to define force-embedment curves for an elastic seabed.
- *EQUIVALENT is used to specify that two individual nodes are a single equivalent node.
- *FLEX JOINT is used to define flex joint elements in the structural discretisation.
- *FLOATING BODY is used to define a floating body and its associated properties.
- *FORCE-STRAIN is used to define force-strain curves for non-linear materials.
- *GEOMETRIC SETS is used to assign geometric properties to element sets.
- *GUIDE is used to define guide (contact) surfaces.
- *HINGE is used to specify hinge elements in the structural discretisation.
- *HYDRODYNAMIC COUPLING is used to define hydrodynamic coupling between adjacent floating bodies.
- *HYDRODYNAMIC SETS is used to assign hydrodynamic coefficients to element sets.
- *LABEL is used to associate a descriptive name with a node or element of the finite element model.
- *LINE LOCATIONS is used to define named locations along a line, allowing you to request that nodes be positioned exactly at specific lengths along a line.
- *LINE SECTION GROUPS is used to define groups of line sections which may be reused and repeated along line sections.
- *LINES is used to define a line (and sections within that line), by specifying relevant set names, lengths, start and end locations, and mesh generation settings.
- *LINES PIP is used to specify that two lines are connected in a pipe-in-pipe (or pipe-on-pipe) configuration.
- *LOCAL AXIS SYSTEM is used to define local axis systems.

- *MASS is used to specify a point mass or a point rotary inertia.
- *MOMENT CURVATURE is used to define moment-curvature curves for non-linear materials.
- *MOONPOOL is used to define the location and extent of the vessel moonpool at solution initiation, for the purpose of applying hydrodynamic loading on elements of the structure contained within the vessel moonpool.
- *MOORED VESSEL is used to define a moored vessel and its associated properties.
- *NODE is used to define nodal coordinates in the global Cartesian coordinate system.
- *NODE,AUXILIARY is used to define auxiliary nodal coordinates in the global Cartesian coordinate system.
- *NODE,CURVILINEAR is used to specify nodes at particular locations along a cable, or to specify a number of equally spaced nodes along a cable segment between two nodes.
- *NODE SPRING is used to specify the stiffness and direction of node springs.
- *NONLINEAR STIFFENER is used to define a stress-strain curve for a non-linear bend stiffener.
- *NO OPTIMISE is used to specify whether or not bandwidth optimisation is to be performed.
- *OCEAN is used to specify general parameters defining the ocean environment.
- *PANEL,AUXILIARY is used to specify nodes which are connected by an auxiliary panel.
- *PANEL SECTIONS,AUXILIARY is used to specify the auxiliary panels that make up an auxiliary body.
- *PIP CONNECTION is used to define pipe-in-pipe connections between nodes of the finite element model.
- *PIP SECTION is used to define internal and external pipe sections when part of a pipe model is contained within another.

- *PIP STIFFNESS is used to define force-deflection curves for non-linear pipe-in-pipe connection stiffnesses.
- *PLASTIC HARDENING is used to define hardening models for plastic materials.
- *POINT BUOY is used to define point buoys and their associated hydrodynamic properties.
- *POISSON is used to include Poisson's ratio effects in an analysis.
- *PROPERTIES is used to assign effective structural properties to element sets for use in calculating stresses.
- *P-Y is used to define P-y curves for modelling soil structure interaction.
- *RADIATION DAMPING is used to define radiation damping for a floating body.
- *RAO FORMAT is used to define custom RAO conversion settings.
- *RIGID SURFACE is used to specify the properties of a rigid seabed.
- *SEABED PROFILE is used to specify the seabed profile/bathymetry.
- *SEABED PROPERTIES is used to specify properties such as seabed type (i.e. rigid or elastic) and friction parameters.
- *SEABED STIFFNESS is used to define nonlinear seabed contact stiffness.
- *SET COLOURS is used to assign a specific colour to an element set or sets.
- *SPRING ELEMENT is used to define the stiffness characteristics of spring elements.
- *STIFFENER is used to define the properties of a conical bend stiffener positioned on a flexible riser or pipe.
- *STRESS/STRAIN is used to define generalised stress-strain curves for non-linear materials.
- *STRESS/STRAIN DIRECT is used to define direct stress-strain curves for non-linear materials.
- *TAPER is used to assign properties to a set of elements that collectively comprises a tapered stress joint, typically in a model of a rigid riser.

- *TORQUE-TWIST is used to define torque-twist curves for non-linear materials.
- *TURBINE GEOMETRY is to specify information relating to a wind turbine, including the hub, shaft, bearings and nacelle.
- *TURBINE ROTOR is used to assemble a wind turbine rotor by selecting blades and specifying related information.
- *T-Z is used to define T-z curves for modelling soil structure interaction.
- *VECTOR is used to define vectors for use with the solution criteria feature.
- *VESSEL is used to specify the initial position and undisplaced orientation of a vessel.
- *VESSEL,INTEGRATED is used to specify all information pertaining to a vessel or vessels.
- *VISCOUS DRAG is used to define viscous drag for a floating body.
- *WAMIT is used to specify that Flexcom is to read floating body data from WAMIT output.

1.8.10.1 *ADDED MASS

PURPOSE

To define added mass for a floating body.

THEORY

Refer to Wave Radiation Loads and Added Mass Coefficients for further information on this feature.

KEYWORD OVERVIEW

Data may defined explicitly within the keyword file itself, or defined separately in an external file which is simply referenced using a FILE= entry. The latter approach is highly recommended as it reduces unnecessary clutter in the keyword file, and also helps to improve user interface performance and enhanced user experience.

Block of data consisting of the floating body name followed by the added mass definition. For frequency independent added mass, a 6x6 matrix is defined by a single block preceded by a TYPE=CONSTANT line. For frequency dependent added mass, multiple 6x6 matrices are defined by several blocks of data, each one preceded by a FREQ= line for the different frequencies, with a TYPE=FREQUENCY line at the beginning. This block is repeated as often as necessary to define the added mass over a range of frequencies. Finally, an optional final block of data defining the added mass at the cut-off frequency, preceded by TYPE=CUTOFF, may be included. The entire block of data can then be repeated to specify added mass for second and subsequent floating bodies.

DATA SPECIFIED IN EXTERNAL FILE

Keyword Format

Line defining floating body name:

```
FLOATING BODY=Floating Body Name
```

Line defining name of external file which contains added mass data.

```
FILE=File Name
```

File Name should include the entire path of the file including its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

Table Input

Input:	Description
Floating Body:	The name of the floating body.
File Name:	The name of the external data file. See Note (a).

Notes

(a) Refer to the following sections for further information regarding the required format of data within the external file.

CONSTANT ADDED MASS

Purpose

To define constant (frequency-independent) added mass for a floating body.

Keyword Format

Line to define floating body name:

```
FLOATING BODY=Floating Body Name
```

Block of lines defining frequency independent added mass:

```
TYPE=CONSTANT
```
A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16}
A_{21}, A_{22}, A_{23}, A_{24}, A_{25}, A_{26}
A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36}
A_{41}, A_{42}, A_{43}, A_{44}, A_{45}, A_{46}
A_{51}, A_{52}, A_{53}, A_{54}, A_{55}, A_{56}
A_{61}, A_{62}, A_{63}, A_{64}, A_{65}, A_{66}

Table Input

Input:	Description
Floating Body:	The name of the floating body.
Matrix (6x6):	A 6x6 matrix of added mass data. See Note (a).

Notes

(b) Refer to Added Mass Coefficients for further information on the layout of the added mass terms.

(c) The type of added mass specified for a particular floating body, either *Constant or Frequency Dependent*, should be consistent with the type of radiation damping specified for the same floating body.

REFERENCE MATRICES

Purpose

To reference frequency dependent and cut-off added mass matrices for use with a floating body.

Keyword Format

Line to define floating body name:

```
FLOATING BODY=Floating Body Name
```

Table Input

Input:	Description
Floating Body:	The name of the floating body.
Frequency Dependent Added Mass:	The name of the frequency-dependent added mass matrix definition.
Cut-Off Added Mass:	The name of the cut-off added mass matrix definition.

Notes

(a) Any properties which you refer to for a particular floating body must be defined in other tables – i.e. *Body - Frequency Dependent Added Mass* and *Body – Cut-Off Added Mass*.

FREQUENCY DEPENDENT ADDED MASS

Purpose

To define frequency-dependent added mass for a floating body.

Keyword Format

Block of lines defining frequency dependent added mass, preceded by a single TYPE=FREQUENCY line:

```
TYPE=FREQUENCY
FREQ=Frequency
```

A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16}
A_{21}, A_{22}, A_{23}, A_{24}, A_{25}, A_{26}
A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36}
A_{41}, A_{42}, A_{43}, A_{44}, A_{45}, A_{46}
A_{51}, A_{52}, A_{53}, A_{54}, A_{55}, A_{56}
A_{61}, A_{62}, A_{63}, A_{64}, A_{65}, A_{66}

Table Input

Input:	Description
Name:	The name of the added mass matrix.
Frequency:	The frequency in Hertz to which the 6x6 added mass matrix relates.
Matrix (6x6):	A 6x6 matrix of added mass data. See Note (a).

Notes

(a) Refer to Added Mass Coefficients for further information on the layout of the added mass terms

(b) This table is used in conjunction with the *Body – Cut-Off Added Mass* table, in which added mass at the cut-off frequency is specified. Added mass at the cut-off frequency is completely defined by a single 6x6 matrix. The cut-off frequency itself is specified in the *Body – Frequency* table.

(c) For frequencies below the cut-off frequency, the added mass corresponds to the added mass at the cut-off frequency. If no cut-off frequency is specified, the added mass is assumed to be zero below the range of user-specified frequencies.

(d) For frequencies above the range of user-specified frequencies, the added mass is assumed to be zero.

(e) For frequencies that do not exactly match one of the frequencies specified, Flexcom linearly interpolates between the nearest frequencies to find the relevant values of added mass.

(f) You do not need to specify the frequencies in any particular order; ascending order is not necessary.

(g) The type of added mass specified for a particular floating body, either *Constant or Frequency Dependent*, should be consistent with the type of radiation damping specified for the same floating body.

CUT-OFF ADDED MASS

Purpose

To define cut-off added mass for a floating body.

Keyword Format

Block of lines defining added mass at the cut-off frequency:

TYPE=CUTOFF

A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16}
A_{21}, A_{22}, A_{23}, A_{24}, A_{25}, A_{26}
A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36}
A_{41}, A_{42}, A_{43}, A_{44}, A_{45}, A_{46}
A_{51}, A_{52}, A_{53}, A_{54}, A_{55}, A_{56}
A_{61}, A_{62}, A_{63}, A_{64}, A_{65}, A_{66}

Table Input

Input:	Description
Name:	The name of the added mass matrix.
Matrix (6x6):	A 6x6 matrix of added mass data. See Note (a).

Notes

(a) Refer to Added Mass Coefficients for further information on the layout of the added mass terms.

(b) This table is used in conjunction with the *Body - Frequency Dependent Added Mass* table, in which frequency-dependent added mass is specified. The cut-off frequency itself is specified in the *Body – Frequency* table.

(c) For frequencies below the cut-off frequency, the added mass corresponds to the added mass at the cut-off frequency. If no cut-off frequency is specified, the added mass is assumed to be zero below the range of user-specified frequencies.

1.8.10.2 *AUXILIARY

PURPOSE

To specify the auxiliary elements which comprise an auxiliary body.

THEORY

Refer to Auxiliary Bodies for further information on this feature.

Note also that the old auxiliary body keywords (`*NODE,AUXILIARY`, `*ELEMENT,AUXILIARY`, `*AUXILIARY`, `*PANEL,AUXILIARY` and `*PANEL SECTIONS,AUXILIARY`) keywords have effectively been superseded by the integrated vessel/body features which provide a range of standard vessel and subsea component profiles. Refer to `*VESSEL,INTEGRATED` and `*BODY,INTEGRATED` for further information.

KEYWORD FORMAT

A block of lines that define an auxiliary body repeated as often as necessary. The block of lines starts with a line defining the body name. It is followed by either of two types of line defining the auxiliary elements assigned to the body, which can be mixed and repeated as often as necessary.

Line defining the body name:

```
BODY=Vessel Name/Node Number
```

Two types of lines for defining auxiliary elements:

```
List of Elements
```

or

```
GEN=Start Element, End Element [, Element Increment]
```

The list of elements can contain up to 20 element numbers. Element Increment defaults to 1. Vessel Name/Node Number can be NONE if no motions are to be applied to the auxiliary body.

TABLE INPUT

Input:	Description
Body:	This entry allows you to associate the auxiliary body with a vessel (in which case you specify the vessel name) or a structural node (in which case you specify the node number).
Elements:	The auxiliary elements that comprise the auxiliary body. You enter these elements in the same manner in which you enter elements in the Element *Sets – Define* table, with the exception that you cannot define auxiliary bodies in terms of other auxiliary bodies. You can only enter element numbers.

NOTES

(a) This table allows you to group together auxiliary elements into named auxiliary bodies. You may also associate the *auxiliary body* with a vessel or a node of the finite element discretisation data. If this option is invoked, the auxiliary elements translate and rotate with the vessel/node during the course of an analysis or series of analyses (static and dynamic). When viewed in the dynamic display, the motions of the auxiliary body will be clearly visible.

(b) You do not need to explicitly assign all auxiliary elements to auxiliary bodies. An auxiliary element that is not assigned to a body remains stationary throughout all subsequent analysis phases. For example, you might want to visually check for interference between a riser or hose and a nominally fixed structure.

1.8.10.3 *BENDING HYSTERESIS

PURPOSE

To define hysteresis moment-curvature backbone curves for non-linear materials.

THEORY

Refer to Hysteretic Bending for further information on this feature.

KEYWORD FORMAT

A block of lines which define a moment-curvature relationship. The block begins with a line defining its name, followed by as many lines as necessary to define each point on the moment-curvature curve. The block can be repeated as often as necessary.

Line defining curve name:

```
CURVE=Curve Name
```

Line defining a point on the curve:

```
Moment, Curvature
```

Each curve must have at least two points defined. The hysteresis curve may not be associated with non-linear beam elements which are defined using the rigid riser format for geometric properties specification.

TABLE INPUT

Input:	Description
Curve Name:	The generic name of the moment-curvature curve.
Moment:	A bending moment value for positive loading increasing from zero.
Curvature:	The corresponding curvature value.

NOTES

(a) Use as many lines as you need to completely define a particular moment-curvature curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent moment-curvature curves, put the curve name in Column 1 and specify the moment-curvature data in the same way.

(b) The points defining the non-linear moment-curvature curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of curvature.

(c) Two to ten segments normally capture the essential characteristics of the hysteresis response. Two segments give a bi-linear curve, while a larger number of segments can be used to represent a smoother hysteresis curve. The hysteresis curve is required to be such that the (positive) bending stiffness decreases as the loading increases, that is, the curve softens and does not harden.

(d) Flexcom converts the hysteresis curve into a multiple component system of elasto-plastic stiffnesses and a linear elastic stiffness. The linear elastic bending stiffness equals the slope on the last segment of the hysteresis curve. The last slope must be positive and usually equals the bending stiffness when hysteresis is not included in the model, for example, a depressurised flexible pipe.

(e) When hysteresis is included in the model, the bending stiffness is assumed symmetric about the local-y and local-z axes.

1.8.10.4 *BLADE GEOMETRY

PURPOSE

To specify the geometry of wind turbine blades.

THEORY

Refer to Turbine Geometry for further information.

KEYWORD FORMAT

A block of lines starting with the blade name, followed by as many lines as necessary to completely define the blade geometry as a function of length (span) along each blade. The entire block of data may then be repeated for other blades. Note that all blades must have the same number of sections.

Line defining the blade name:

```
BLADE NAME=Blade Name
```

Line defining the blade properties at a particular value of blade span (this line may be repeated as often as required):

```
Blade Span, Aerodynamic Centre Out-of-Plane Offset, Aerodynamic Centre In-Pla
```

Refer to the Blade Geometry schematics for an illustration of the various inputs. Each blade definition must have properties defined for at least two values of blade span. The `Blade Span` entries must be entered in monotonically increasing order, from the most inboard to the most outboard. The first span must be zero and the last span should be located at the blade tip. Valid values of `Aerofoil Index` are numbers between 1 and the Total Number of Aerofoil Input Files.

The rotor is assembled according to the blade selections and related information defined in the `*TURBINE ROTOR` keyword. Flexcom creates a node in the finite element discretisation corresponding to each blade span position defined in the `*BLADE GEOMETRY` keyword. These nodes are connected sequentially using finite elements whose structural properties are governed by the inputs in the `*BLADE STRUCTURE` keyword. The structural properties are assumed constant along each element, and determined using linear between the element centrepoint and the nearest available sectional definitions.

TABLE INPUT

Input:	Description
Blade Name:	The name of the blade. Note that the same blade names should be used in the `*BLADE GEOMETRY`, `*BLADE STRUCTURE` and `*TURBINE ROTOR` keywords.
Blade Span:	The distance along the (possibly preconed) blade-pitch axis from the root. This is illustrated by 'BlSpn' in Blade Geometry - Side View. The span entries must be entered in monotonically increasing order, from the most inboard to the most outboard. The first span must be zero and the last span should be located at the blade tip. Each blade definition must have properties defined for at least two values of blade span.
Aerodynamic Centre Out-of-Plane Offset:	The local out-of-plane offset (when the blade-pitch angle is zero) of the aerodynamic center (reference point for the aerofoil lift and drag forces), normal to the blade-pitch axis, as a result of blade curvature. This is illustrated by 'BlCrvAC' in Blade Geometry - Side View. The offset is measured positive downwind; upwind turbines have negative offsets for improved tower clearance.

Aerodynamic Centre In-Plane Offset:	The local in-plane offset (when the blade-pitch angle is zero) of the aerodynamic center (reference point for the aerofoil lift and drag forces), normal to the blade-pitch axis, as a result of blade sweep. This is illustrated by 'BlSwpAC' in Blade Geometry - Front View. A positive offset is opposite the direction of rotation.
Curvature Angle:	The local angle (in degrees) from the blade-pitch axis of a vector normal to the plane of the aerofoil, as a result of blade out-of-plane curvature (when the blade-pitch angle is zero). This is illustrated by 'BlCrvAng' in Blade Geometry - Side View. Curvature angle is measured positive downwind; upwind turbines have negative curvature angles for improved tower clearance.
Twist Angle:	The local aerodynamic twist angle (in degrees) of the aerofoil. It is the orientation of the local chord about the vector normal to the plane of the aerofoil, positive to feather, leading edge upwind. The blade-pitch angle will be added to the local twist.
Chord Length:	The local chord length.
Aerofoil Index:	This entry specifies which aerofoil data input file is to associated with this local blade span position. Valid values are numbers between 1 and the Total Number of Aerofoil Input Files. Multiple blade nodes can use the same aerofoil data.

NOTES

(a) You can define as many blades as you wish in the `*BLADE GEOMETRY` and `*BLADE STRUCTURE` keywords. Only the blades which you reference in the `*TURBINE ROTOR` keyword will be used in the model.

(b) Flexcom creates a node in the finite element discretisation corresponding to each blade span position defined in the `*BLADE GEOMETRY` keyword. These nodes are connected sequentially using finite elements whose structural properties are governed by the inputs in the `*BLADE STRUCTURE` keyword. The structural properties are assumed constant along each element, and determined using linear interpolation between the element centrepoint and the nearest available sectional definitions.

1.8.10.5 *BLADE STRUCTURE

PURPOSE

To specify the structural properties of wind turbine blades.

Please note that Flexcom 2022.1 uses a simplified rotor nacelle assembly (RNA) model considers that blade deformations under applied loading are negligible, hence the blade geometries are approximated as rigid profiles. This means that the `*BLADE STRUCTURE` keyword inputs are currently irrelevant. Work is presently under way to develop a more detailed rotor nacelle assemble (RNA) model that explicitly models the blades with finite elements, thereby allowing blade deformations and rotational inertia to be accurately captured. We hope to make this feature available in the next program release.

THEORY

Refer to Turbine Geometry for further information.

KEYWORD FORMAT

A block of lines starting with the blade name, followed by as many lines as necessary to completely define the blade structural properties as a function of fractional distance along each blade. The entire block of data may then be repeated for other blades.

Line defining the blade name:

```
BLADE NAME=Blade Name
```

Line defining the blade structural properties at a particular value of fractional distance along the blade (this line may be repeated as often as required):

```
Blade Fractional Distance, Pitch Axis, Structural Twist angle, Section Mass,
```

Refer to the Blade Geometry schematics for an illustration of the various inputs. Each blade definition must have properties defined for at least two values of blade fractional distance. The *Blade Fractional Distance* entries must be entered in monotonically increasing order, from the most inboard to the most outboard. The first entry must be 0.0 and the last entry must be 1.0.

The rotor is assembled according to the blade selections and related information defined in the *TURBINE ROTOR keyword. Flexcom creates a node in the finite element discretisation corresponding to each blade span position defined in the *BLADE GEOMETRY keyword. These nodes are connected sequentially using finite elements whose structural properties are governed by the inputs in the *BLADE STRUCTURE keyword. The structural properties are assumed constant along each element, and determined using linear interpolation between the element centrepoint and the nearest available sectional definitions.

TABLE INPUT

Input:	Description
Blade Name:	The name of the blade. Note that the same blade names should be used in the *BLADE GEOMETRY, *BLADE STRUCTURE and *TURBINE ROTOR keywords.
Blade Fractional Distance:	The fractional distance of the blade along the blade pitch axis. Values must vary from 0 to 1. The first distance corresponds to the blade root and must have a value of 0. The last distance, which corresponds to the blade tip, must have a value of 1.
Pitch Axis:	This input is used to locate the aerodynamic center of the corresponding airfoil section. Sometimes referred to as "AeroCent" in OpenFAST notation, it represents the fractional distance along the chordline from the leading to the trailing edge, where it is assumed that pitch axis passes through the airfoil section at 25% chord so that the leading edge is 25% ahead of the pitch axis along the chordline and the trailing edge is 75% aft of the pitch axis along the chordline. The input is limited to values between 0.0 and 1.0. A value of 0.0 corresponds to the leading edge, a value of 0.25 corresponds to the blade pitch axis, and a value of 1.0 corresponds to the trailing edge. Refer to OpenFAST documentation for further details.
Structural Twist Angle:	This input indicates the orientation of the principal axis. It must be greater than -180 and less or equal than 180 degrees.

Section Mass:	This input indicates the blade section mass per unit length.
Flapwise Stiffness:	This input indicates the blade section flapwise stiffness.
Edgewise Stiffness:	This input indicates the blade section edgewise stiffness.
Torsional Stiffness:	This input indicates the blade section torsional stiffness.
Extensional Stiffness:	This input indicates the blade section extensional stiffness.

NOTES

(a) You can define as many blades as you wish in the `*BLADE GEOMETRY` and `*BLADE STRUCTURE` keywords. Only the blades which you reference in the `*TURBINE ROTOR` keyword will be used in the model.

(b) Flexcom creates a node in the finite element discretisation corresponding to each blade span position defined in the `*BLADE GEOMETRY` keyword. These nodes are connected sequentially using finite elements whose structural properties are governed by the inputs in the `*BLADE STRUCTURE` keyword. The structural properties are assumed constant along each element, and determined using linear interpolation between the element centrepoint and the nearest available sectional definitions.

1.8.10.6 *BODY,INTEGRATED

PURPOSE

To add subsea components to the structural animation for enhanced visual appeal.

THEORY

Refer to Auxiliary Bodies for further information on this feature.

KEYWORD FORMAT

A block of lines defining a structural body. The block begins with a mandatory pair of lines defining the body name and its physical location. This is followed by an optional line associating the body with a node of the finite element discretisation. This is then followed by additional optional lines associating a predefined profile with the body. The entire block may then be repeated for subsequent bodies.

Mandatory pair of lines defining the body name and its physical location:

```
BODY=Body Name
LOCATION=X, Y, Z
```

Optional line associating the body with a node of the finite element discretisation:

```
[NODE=Node (Number or Label)]
```

Mandatory line associating a predefined profile with the body:

```
PROFILE=Profile Type
```

If the profile type is user-defined, then another mandatory line immediately follows:

```
FILE=File Name
```

Optional line defining overall dimensions for the body:

```
[DIMENSIONS=Height, Length, Width]
```

If you specify a node label rather than a node number, it must be enclosed in { } brackets. *Profile Type* may be BOP, CALMBUOY, PLET, ROV or USER. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks. If dimensions are not explicitly defined, some generic dimensions are applied automatically to the body.

TABLE INPUT

Input:	Description
Body Name:	A descriptive name for the body.
X, Y, Z:	The global X, Y, & Z coordinates of the centre of the body at solution initiation.

Node:	The node (number or label) whose motions govern the movement of the body. If you specify a node label rather than a node number, it must be enclosed in { } brackets.
Profile:	The profile type used to represent the body. You may choose from a range of standard profiles, or opt to use a customised profile.
User Profile:	If you are using a customised profile, this input defines the location of the profile file. See Note (a).
Height, Length, Width:	The overall dimensions of the profile in the global X, Y and Z directions. If dimensions are not explicitly defined, some generic dimensions are applied automatically to the body.

NOTES

(a) The profile is defined in XML format, and several sample profiles are available in your local Flexcom installation directory, under the sub-folder 'StandardProfiles'. Each profile file contains a range of input data, capable of defining standard shapes such as boxes, cylinders etc., as well as arbitrary shapes defined by a series of points about which a mesh is constructed. All locations within the profile file are defined with respect to a local origin (0, 0, 0). The location of this origin in the global axis system is then defined by the X, Y, & Z coordinates of the body centre. Should you require any assistance in creating your own customised profile, feel free to contact sw.support@woodplc.com and our support team will discuss the format of the XML information in more detail.

1.8.10.7 *CABLE

PURPOSE

To specify that part of the structure forms a cable between specified end points.

THEORY

Refer to Cables for further information on this feature.

Note also that the old *NODE, *ELEMENT and *CABLE keywords have been largely superseded by the new *LINES keyword. Lines provide an automatic mesh creation facility to greatly expedite the model creation process. Using lines is a fundamentally different approach to working directly with nodes, elements and cables, although the information is ultimately handled in the same fashion internally. Since lines provide automatic mesh generation, you do not to concern yourself with explicit node and element numbering. Indeed the availability of lines makes nodes, elements and cables redundant to some degree, but they are retained for complete generality, and also to maintain downward compatibility with previous program versions. Refer to Lines for further information on this feature.

KEYWORD FORMAT

A single line defining a cable. This line may be repeated as often as necessary.

```
    Cable Number, Start Node (Number or Label), End Node (Number or Label), Cable
```

A *Cable Number* cannot be redefined. If you specify a node label rather than a node number, it must be enclosed in {} brackets. The *Cable Length* must be greater than zero.

TABLE INPUT

Input:	Description
Cable Number:	The number of the cable being defined.
Start Node:	The node (number or label) defining the start of the cable. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
End Node:	The node (number or label) defining the end of the cable. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Cable Length:	The length of the cable from the start to the end node

1.8.10.8 *CABLE BUNDLE

PURPOSE

To specify that two (or more) cables are to be "bundled" together for the purposes of the Cable Pre-Static Step.

THEORY

Refer to Cable Bundles for further information on this feature.

KEYWORD FORMAT

A single line defining a cable bundle. This line may be repeated as often as necessary.

```
COMBINE=First Cable Number, ..., Nth Cable Number
```

A maximum of 20 cables may be included together in a single bundle. Any cable may not appear in more than one bundle.

TABLE INPUT

Input:	Description
List of Cables:	A list of the cable numbers within a bundle.

NOTES

(a) If you invoke this option, you can specify that two (or more, up to a maximum of 20) cables are to be grouped together within a single bundle for the purposes of the Cable Pre-Static Step only. This means that internally, all cables within a bundle share common properties for the purposes of the cable computations, ensuring that each cable adopts a similar profile before the full finite element solution initiates.

1.8.10.9 *CALM MODEL

PURPOSE

To specify the properties of calm buoys.

THEORY

Refer to CALM Buoy for further information on this feature.

KEYWORD FORMAT

A block of five lines defining the calm buoy properties.

Node (Number or Label)
M_{11}, I_{11}, I_{22}, I_{33}
A_{11}, A_{33}, A_{55}, A_{15}
K_{33}, K_{44}
D_{11}, D_{33}, D_{55}, D_{15}

where:

Node = the node of the finite element mesh located at the buoy Centre of Gravity. If you specify a node label rather than a node number, it must be enclosed in {} brackets.

M_{11} = M_{22} = M_{33} = mass of buoy

I_{11} = roll inertia of the buoy

I_{22} = pitch inertia of the buoy

I_{33} = yaw inertia of the buoy

A_{11} = surge added mass

A_{33} = heave added mass

A_{55} = pitch added mass

A_{15} = coupled surge pitch added mass

K_{33} = heave stiffness

K_{44} = K_{55} = roll and pitch stiffness terms

D_{11} = surge damping

D_{33} = heave damping

D_{55} = pitch damping

D_{15} = coupled surge pitch damping

TABLE INPUT

Input:	Description
Node:	The node (number or label) of the finite element mesh located at the buoy Centre of Gravity. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Mass:	The mass of the buoy.
Roll Inertia:	The roll inertia of the buoy.
Pitch Inertia:	The pitch inertia of the buoy.
Yaw Inertia:	The yaw inertia of the buoy.
Surge Added Mass:	The added mass in surge.
Heave Added Mass:	The added mass in heave.
Pitch Added Mass:	The added mass in pitch.
Coupled Surge-Pitch Added Mass	The coupled surge-pitch added mass.
Heave Stiffness:	The stiffness (due to buoyancy) in heave.
Roll and Pitch Stiffness:	The stiffness (due to buoyancy) in roll and pitch.

Surge Added Damping:	The added damping in surge.
Heave Added Damping:	The added damping in heave.
Pitch Added Damping:	The added damping in pitch.
Coupled Surge-Pitch Added Damping	The coupled surge-pitch added damping.

NOTES

(a) Refer to Model Set-up for further information on how to incorporate a CALM buoy into the finite element model using an assemblage of standard beam-column elements.

(b) Refer to Buoy Properties for further information on the specification of the mass, buoyancy stiffness, added mass and radiation damping matrices.

1.8.10.10 *CLASHING

PURPOSE

To specify regions where clashing may occur, and suitable contact stiffness and damping values.

THEORY

Refer to Line Clashing for further information on this feature.

KEYWORD FORMAT

A single line of data which may be repeated as often as necessary.

```
FIRST=First Set Name, SECOND=Second Set Name, STIFFNESS=Stiffness [, DAMPING=
```

Damping defaults to zero if omitted.

TABLE INPUT

Input:	Description
First Element Set:	The first element set to be monitored for clashing. Clashing may occur between this set and the corresponding second element set. See Note (a).
Second Element Set:	The second element set to be monitored for clashing. Clashing may occur between this set and the corresponding first element set. See Note (a).
Contact Stiffness:	The contact stiffness used in the clashing model. This input is mandatory. There is no default value. See Note (b).
Contact Damping:	The contact damping. This input is optional. By default, no damping effect is modelled.

NOTES

(a) Monitoring of clashing introduces a significant computational overhead, as all elements in the first set are checked for contact with all elements in the second set, for all the clashing regions that you have defined. When defining contact regions, you should be careful that you specify reasonable data to prevent excessive runtimes, by ensuring that the program is not checking for contact between points that will never approach. It is generally recommended that similar spatial discretisations be used for any element sets which may come into contact with each other. The choice of which element set is designated as first or second should be irrelevant – it is merely a convenience for the internal operation of the contact modelling algorithm.

(b) It may be difficult to quantify what value of contact stiffness represents the physical reality. Generally speaking, the higher the stiffness value chosen, the greater the impact will be, and the smaller the time step that will be required for a robust solution. So you should avoid using excessively high stiffness values, which may lead to convergence difficulties. Conversely, the stiffness value effectively determines the maximum clashing force that may be modelled – the maximum reaction force is equal to the contact stiffness times the average contact radius of the elements which interact with each other. In other words, the maximum relative penetration of the contact elements should not be allowed to exceed the average contact radius. If this occurs, the lines will simply pass though each other. Although in practice this situation would be quite rare, to prevent the situation from occurring, you should ensure that the contact stiffness is sufficiently high. While it is difficult to provide meaningful guidance as to what value might represent a “reasonable” level for all cases, a value of approximately 100 kN/m (or equivalent in Imperial units) is tentatively suggested as a starting point if you are unsure.

1.8.10.11 *COATINGS

PURPOSE

To apply coatings on the basis of element set.

THEORY

Refer to Coatings for further information on this feature.

KEYWORD FORMAT

A block of lines which defines a coating, repeated as often as necessary.

```
SET=Element Set
TYPE=Type
NAME=Name, THICKNESS=Thickness, DENSITY=Mass Density
```

Notes: Type may be either INTERNAL or EXTERNAL. The third line may be repeated to assign multiple coatings in succession. Coatings are assumed to occur in order from the base section – i.e. inwards in the case of internal coatings, and outwards for external coatings. The second line may be repeated to switch between internal and external coatings for a particular element set.

TABLE INPUT

Input:	Description
Set Name:	The element set to which the coatings are assigned. This defaults to *All*, to indicate all elements.
Type:	This option allows you to select whether the coatings are *Internal* or *External* (the default).
Name:	The name of the coating, for example, “Concrete”.
Thickness:	The thickness of the coating.
Mass Density:	The mass density of the coating

NOTES

(a) The option to specify coatings is independent of the geometric format option used to define the base element set. Flexcom assumes that the coatings occur in order from the base section, which means inwards in the case of internal coatings, and outwards for external coatings.

(b) The application of coatings causes an increase in mass per unit length, while the stiffness properties remain unchanged.

(c) The various diameters (drag, buoyancy, external, contact and internal) are adjusted as appropriate. Specifically, the internal diameter is reduced where one or more internal coatings are present, and the external diameter is increased if one or more external coatings are present. The remaining diameters (drag, buoyancy and contact) are only adjusted where the relevant user-specified value is exceeded by the external diameter plus any external coatings. It is important to note that the diameters used for stress computations are unaffected by the presence of coatings, and the values specified for the base section take priority.

1.8.10.12 *COLOUR DEFINE

PURPOSE

To define a specific colour using RGB colour coding.

THEORY

Refer to Model View for further information on this feature.

KEYWORD FORMAT

A pair of lines, repeated as often as necessary.

```
NAME=Colour Name
Red Value, Green Value, Blue Value, Alpha Value
```

Note that `Colour Name` must not be the same as a predefined colour name. Refer to the `*SET COLOURS` keyword for a full list of predefined colour names.

The `Red, Green` & `Blue` values are integers ranging from 0 to 255 which represent the amounts of the primary colours red, green or blue in the colour being defined.

`Alpha Value` defines a transparency value in the range 0 to 255, 0 being fully transparent and 255 being fully opaque.

TABLE INPUT

Input:	Description
Colour Name	The name of the colour to be defined. See Note (a).
Red Value	Must be an integer between 0 and 255. See Note (b).
Green Value	Must be an integer between 0 and 255. See Note (b).
Blue Value	Must be an integer between 0 and 255. See Note (b).
Alpha Value	Must be an integer between 0 and 255. See Note (c).

NOTES

a) *Colour Name* cannot be the same as a predefined colour name. Refer to the `*SET COLOURS` keyword for a full list of predefined colour names.

b) *Red, Green & Blue* values are integers ranging from 0 to 255.

c) *Alpha Value* is an integer from 0 to 255 representing transparency, 0 being fully transparent and 255 fully opaque.

1.8.10.13 *DAMPER

PURPOSE

To define damper elements in the structural discretisation.

THEORY

Refer to Damper Elements for further information on this feature.

KEYWORD FORMAT

At least one line (in the case of translational damper) repeated as often as necessary.

```
[TYPE=TRANSLATIONAL]
Element (Number or Label) [, C1 or C1=Name] [, C0 or C0=Name] [, C2 or C2=Name
[, C0_THRESHOLD=C0_Threshold]

[TYPE=ROTATIONAL]
Element (Number or Label) [, C1]
```

All coefficients default to zero. If `TYPE=` not specified it defaults to `TRANSLATIONAL`. For translational dampers, each coefficient may be specified in terms of a single value, or as a series of values which depend on time or velocity. For rotational dampers, the damping coefficient must be a single numerical value. You must use the `*DAMPER DATA` keyword to define any damping coefficient relationship you reference here.

TABLE INPUT

Translational Damper Elements

Input:	Description
Element:	The element (number or label) of the damper element. If you specify an element label rather than an element number, it must be enclosed in {} brackets.

Linear Coefficient (C1):	The linear damping coefficient for the damper element.
Constant Force (F0):	The constant damping force for the damper element.
Quadratic Coefficient (C2):	The quadratic damping coefficient for the damper element
Velocity Threshold:	A velocity threshold below which the constant damping force is linearly ramped. By default no ramping is applied, so the constant damping force is applied in full at all times.

Rotational Damper Elements

Input:	Description
Element:	The element (number or label) of the damper element. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Linear Coefficient (C1):	The linear damping coefficient for the damper element.

NOTES

(a) For translational damper elements, the axial force exerted by the element, F, is defined as follows:

$$F = -(F_0 + C_1 v + C_2 v^2)$$

where v is the relative velocity between the element end nodes in the axial direction of the element, F_0 is the constant damping force, and C_1 & C_2 are the linear and quadratic damping coefficients respectively. Each coefficient may be specified in terms of a single value, or as the name of a damping coefficient relationship. Refer to Damper Elements for further information.

(b) For rotational damper elements, the moment exerted by the element, M, is defined as follows:

$$M = -(C_1 v)$$

where v is the relative rotational velocity between the element end nodes and C_1 is the linear damping coefficient.

(c) If the name of a damping coefficient relationship is specified, then the relationship must be defined in the `*DAMPER DATA` keyword.

1.8.10.14 *DAMPER DATA

PURPOSE

To define damping coefficients for use with damper elements.

THEORY

Refer to Damper Elements for further information on this feature.

TIME-DEPENDENT DAMPER ELEMENTS

Keyword Format

A block of lines that define a time-coefficient relationship repeated as often as necessary. The block begins with a line defining the relationship name. It is followed by as many lines as necessary to define each pair of time-coefficient values.

Line defining the relationship name:

```
NAME=Name [, VARIABLE=TIME]
```

Line defining time-coefficient values:

```
Time, Coefficient
```

Each time-coefficient relationship must have at least two points defined.

Table Input

Input:	Description
Name:	The name of the time-coefficient relationship.
Time:	A time value.
Coefficient:	A corresponding damping coefficient value.

VELOCITY-DEPENDENT DAMPER ELEMENTS

Keyword Format

A block of lines that define a velocity-coefficient relationship repeated as often as necessary. The block begins with a line defining the relationship name. It is followed by as many lines as necessary to define each pair of velocity-coefficient values.

Line defining the relationship name:

```
NAME=Name, VARIABLE=VELOCITY
```

Line defining velocity-coefficient values:

```
Velocity, Coefficient
```

Each velocity-coefficient relationship must have at least two points defined.

Table Input

Input:	Description
Name:	The name of the velocity-coefficient relationship.
Velocity:	A velocity value.
Coefficient:	A corresponding damping coefficient value.

NOTES

(a) Use as many lines as you need to completely define a particular relationship. You can leave the *Name* column blank for second and subsequent lines. For subsequent relationships, put the name in the *Name* column and specify the coefficient data in the same way.

(b) The points defining the relationship can be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of time or velocity.

(c) If the instantaneous solution variable (be it time or velocity) lies between the specified data points, Flexcom uses linear interpolation to determine the relevant damping coefficient.

(d) If the instantaneous solution variable (be it time or velocity) lies outside the specified data range, then Flexcom simply extrapolates from the first or last section of data as appropriate.

1.8.10.15*DRAG CHAIN

PURPOSE

To include drag chains in the structural model.

THEORY

Refer to Drag Chains for further information on this feature.

KEYWORD FORMAT

Two types of line repeated as often as necessary.

Line defining a single drag chain:

```
Node (Number or Label), Total Chain Length, Mass/Unit Length [, Lift Coeffic
```

Line generating a number of drag chains:

```
GEN=Master Node (Number or Label), No. of Chains [, Node Increment]
```

When generating a number of drag chains a drag chain must have been previously defined at the master node. If you specify a node label rather than a node number, it must be enclosed in {} brackets. `Node Increment` defaults to 1.

TABLE INPUT

Drag Chains - Define Directly

Input:	Description
Node Number:	The node (number or label) at which the drag chain is being attached. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Total Chain Length:	The total length of the drag chain.
Mass/Unit Length:	The mass per unit length of the drag chain.
Lift Coefficient:	The drag chain lift coefficient CL, used to calculate the lift force per unit length of the drag chain. See Note (b).

Drag Chains - Generate

Input:	Description
Master Node:	The node (number or label) at which the drag chain properties for the master chain are defined. The properties of this chain are assigned to all of the chains that are generated here. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Number of Chains:	The number of chains to be generated, where this includes the master chain.
Node Increment:	The node number increment between nodes at which drag chains are to be positioned. This defaults to a value of 1.

NOTES

(a) Gravity and lift forces for the drag chain are computed and applied as point loads at the specified node of the structure. The gravity load is calculated from the length of drag chain not lying on the seabed.

(b) The lift force on each chain is calculated from the relative fluid/structure velocity at the node at which the drag chain is attached, and the lift coefficient that you specify here. The lift force is computed using the following equation

$$\mathrm{F}_L = \tfrac{1}{2} C_L \rho_w u^2 l^2$$

where F_L is the total lift force, C_L is the specified drag chain lift coefficient, P_W is the mass density of seawater, U is the magnitude of the horizontal component of relative fluid/structure velocity at the drag chain node, and l is the length of drag chain not lying on the seabed.

1.8.10.16 *DRAG LIFT

PURPOSE

To define non-linear drag lift relationships.

THEORY

Refer to Drag Lift for further information on this feature.

KEYWORD FORMAT

Two lines, with the second line repeated as often as necessary to define drag lift coefficients at different ratios.

```
CURVE=Curve Name
Ratio, Drag Lift
```

TABLE INPUT

Input:	Description
Curve Name:	A name for the drag lift curve.

Ratio:	The ratio of the gap between the riser and the seabed and the drag diameter of the riser, for a point on the curve.
Drag Lift Coefficient:	The corresponding drag lift coefficient value

NOTES

(a) This table is used to define non-linear drag lift curves for a particular set of elements. Non-linear drag lift curves may be assigned to element sets using the `*HYDRODYNAMIC SETS` keyword.

(b) Use as many lines as you need to completely define a particular drag lift curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent drag lift curves, put the curve name in Column 1 and specify the drag lift data in the same way.

(c) The points defining the non-linear drag lift curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of gap:diameter ratio.

(d) If the gap:diameter ratio for an element lies between the data points you specify, Flexcom uses linear interpolation to determine the relevant drag lift coefficient for the element.

(e) If the gap:diameter ratio for the element lies outside the specified range of the drag lift curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

1.8.10.17 *ELASTIC SURFACE

PURPOSE

To specify the properties of an elastic seabed.

THEORY

Refer to Seabed Interaction for further information on this feature.

Note also that the old `*ELASTIC SURFACE` keyword has effectively been superseded by the new seabed definition keywords which facilitate the modelling of arbitrary seabed profiles. Refer to `*SEABED PROPERTIES`, `*SEABED PROFILE` and `*SEABED STIFFNESS` for further information.

KEYWORD FORMAT

Two lines, the second of which is optional, with the following format:

[*Seabed Stiffness*], *Longitudinal Coefficient*, *Transverse Coefficient* [, *Lateral Seabed Stiffness*] [, *Suction Stiffness*] [, *Suction Zone Extent*] [, *Maximum Longitudinal Characteristic Length*] [, *Maximum Transverse Characteristic Length*] [, *Constant Friction Force Coefficient*] [, *Longitudinal Mobilisation Length*] [, *Transverse Mobilisation Length*]

Line defining uniform seabed slope:

[*Seabed Slope*]

Line defining a point of the arbitrary seabed:

Y Co-ordinate, *Seabed Elevation*

A model may not contain both rigid and elastic surfaces. Omitting the Seabed Stiffness input means a non-linear seabed is being input, in which case the `*EMBEDMENT` keyword is required. The *Seabed Slope* is measured in degrees. A sloping seabed is assumed to pass through the global origin. A flat seabed is assumed if no slope is specified.

Both *Maximum Characteristic Length* inputs default to 10ft or 3.048m. Where a *Mobilisation Length* is explicitly specified, it takes precedence over the corresponding *Maximum Characteristic Length* (in this case the latter value is immaterial as it is unused). If a *Mobilisation Length* is omitted, then the mobilisation length is governed by the *Maximum Characteristic Length* and the finite element mesh discretisation. Refer to Seabed Friction for further details.

TABLE INPUT

Elastic Surface

Input:	Description

Seabed Stiffness:	The elastic stiffness per unit length of the seabed.
Longitudinal Coefficient of Friction:	The coefficient of friction in the longitudinal direction. This defaults to a value of 0.
Transverse Coefficient of Friction:	The coefficient of friction in the transverse direction. This defaults to a value of 0.
Slope:	The slope of a uniformly sloping seabed, in degrees. This defaults to a value of 0o, which gives a horizontal seabed. See Note (f).
Lateral Seabed Stiffness:	This input allows you to specify a lateral resistance to horizontal motion for the elastic seabed. This defaults to a value of 0. See Note (g).
Suction Stiffness:	This input allows you to input a seabed stiffness for elements which have moved off an elastic seabed but which are still in a so called "suction zone" between the mudline and the elevation at which suction forces disappear. Suction Stiffness defaults to a value of 0. See Note (h).
Suction Zone Extent:	This is related to the previous input, Suction Stiffness. Suction Zone Extent represents the elevation or height above the mudline at which suction forces become zero. This input is meaningless unless the soil Suction Stiffness is nonzero, but is required if the Suction Stiffness is non-zero.
Maximum Longitudinal Characteristic Length:	The maximum characteristic length used to simulate frictional restraint in the longitudinal direction on the seabed. This entry is optional. See Note (i).

Maximum Transverse Characteristic Length:	The maximum characteristic length used to simulate frictional restraint in the transverse direction on the seabed. This entry is optional. See Note (i).
Constant Friction Force Coefficient:	The constant friction force coefficient in the transverse direction to be used in addition to the classical limiting friction. This entry is optional and defaults to 0. See Note (j).
Longitudinal Mobilisation Length:	The mobilisation length used to simulate frictional restraint in the longitudinal direction on the seabed. This entry is optional. See Note (i).
Transverse Mobilisation Length	The mobilisation length used to simulate frictional restraint in the transverse direction on the seabed. This entry is optional. See Note (i).

Bathymetry

Input:	Description
Y Co-ordinate	Y Co-ordinate along seabed
Seabed Elevation	Elevation point along seabed

NOTES

(a) The longitudinal friction coefficient refers to a local direction parallel to the pipe axis, and the transverse coefficient to a direction normal to the pipe axis.

(b) By default, no seabed is included in your analysis.

(c) The default friction coefficients give a smooth seabed, with no frictional resistance.

(d) The seabed stiffness is input as a stiffness per unit length of riser or pipeline, in units of [Force]/[Distance]/[Distance] or [Force]/[Distance]^2.

(e) This keyword and the *RIGID SURFACE keyword are mutually exclusive. If you have included a *Rigid Surface* in your analysis, then you may not include an *Elastic Surface* also.

(f) The seabed slope is input in degrees; a positive slope defines a seabed sloping upwards in the positive global Y direction, while a negative slope gives a seabed sloping in the opposite direction. A uniformly sloping seabed is assumed to pass through the global origin of co-ordinates. If a non-zero seabed slope is specified, then the definition of an arbitrary seabed bathymetry is not permitted.

(g) The *Lateral Seabed Stiffness* entry specifies a lateral resistance to horizontal motion on the elastic seabed. Refer to Lateral Resistance for further information on this feature.

(h) The suction or restraining force experienced by a riser element in a “suction zone” just above the mudline is modelled with a linear spring resistance similar to that provided against downward vertical motion by the elastic seabed itself. Refer to Suction Zone for further information on this feature.

(i) The *Maximum Characteristic Length* and *Mobilisation Length* inputs relate to the operation of the Flexcom seabed friction model, and allow user control over the characteristics of non-linear springs used to model seabed frictional restraint. All these entries are optional. The *Maximum Characteristic Length* inputs default to 3.048m or 10ft. Where a *Mobilisation Length* is explicitly specified, it takes precedence over the corresponding *Maximum Characteristic Length* (in this case the latter value is immaterial as it is unused). If a *Mobilisation Length* is omitted, then the mobilisation length is governed by the *Maximum Characteristic Length* and the finite element mesh discretisation. Refer to Seabed Friction for further details.

(j) With the *Constant Friction Force Coefficient* input, the total limiting friction, in the transverse direction, is given by:

$$F_{total} = \mu N + CFF$$

where μN is the Coulombian limiting friction force, and *CFF* is the constant friction force. This equation is applicable only to transverse friction. The constant friction force is evaluated for each seabed contact element, and is given by the element length multiplied by the constant friction force coefficient.

1.8.10.18 *ELEMENT

PURPOSE

To specify the finite element connectivity of the structural model.

THEORY

Refer to Elements for further information on this feature.

Note also that the old `*NODE`, `*ELEMENT` and `*CABLE` keywords have been largely superseded by the new `*LINES` keyword. Lines provide an automatic mesh creation facility to greatly expedite the model creation process. Using lines is a fundamentally different approach to working directly with nodes, elements and cables, although the information is ultimately handled in the same fashion internally. Since lines provide automatic mesh generation, you do not to concern yourself with explicit node and element numbering. Indeed the availability of lines makes nodes, elements and cables redundant to some degree, but they are retained for complete generality, and also to maintain downward compatibility with previous program versions. Refer to Lines for further information on this feature.

KEYWORD FORMAT

Two types of line that may be mixed and/or repeated as often as necessary.

Line defining a single element:

```
Element, Start Node (Number or Label), End Node (Number or Label) [, v1, v2,
```

Line generating a number of elements from an existing element:

```
GEN=Master Element, No. of Elements [, Element Increment]
```

If the generation option is used then the master element must be defined previously. If you specify a node label rather than a node number, it must be enclosed in {} brackets. *Element Increment* defaults to 1. The default values of v1, v2, v3, w1, w2, w3 depend on whether the element is defined on a cable or not. Refer to Undeformed Versus Initial Positions for a detailed discussion on this topic.

TABLE INPUT

Elements - Define Directly

Input:	Description

Element:	The number of the element being defined.
First Node:	The first node (number or label) of this element. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Last Node:	The second node (number or label) of this element. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
V1, V2, V3:	The components in the global coordinate axes of a vector V, (one of two) defining the undeformed orientation of the element. These entries are optional - see Notes (c) and (d).
W1, W2, W3:	The components in the global coordinate axes of a vector W, (one of two) defining the undeformed orientation of the element. These entries are optional - see Notes (c) and (d).

Elements - Generate

Input:	Description
Master Element:	The number of the element defining the node numbering pattern to be copied in the generated elements.
Number of Elements:	The number of elements to be generated, where this number includes the master element.
Element Increment:	The element number increment to be used in assigning numbers to the generated elements. The default value is 1, which will apply in the majority of cases.

NOTES

(a) Element numbers do not need to be continuous in a Flexcom model - as with node numbers you can use any arbitrary scheme for assigning element numbers.

(b) Articulations and spring elements are fully-fledged elements of the finite element discretisation, and you must define their connectivity, using the *Elements – Define Directly* or the *Elements - Generate* table.

(c) The specification of the components of the vectors V and W is related to the Flexcom facility for analysing a structure which is initially deformed. Basically, you specify V and W explicitly when the configuration defined by the nodal coordinates you have input does not represent a stress-free structure orientation. However, in the majority of analyses, this is not the case, and the specified nodal coordinates do represent the undeformed as well as the initial position. In this situation, the specification of V and W is optional, and should in general be omitted (with two exceptions to be discussed shortly). If you want to enter V and W values then all of the values V1, V2, V3, W1, W2, and W3 must be entered. When you do not define V and W explicitly, Flexcom calculates nominal values for these vectors using a default algorithm, based on the specified nodal coordinates. Refer to Undeformed Versus Initial Positions for a detailed discussion on this topic.

(d) The use of cables as discussed in Structures with Cables has one important ramification with regard to the orientation of the elements that comprise a cable. Where you allow the program to calculate the element orientation, as you normally will, a potential problem arises. Specifically, since the coordinates of the nodes along a cable are not known, the default algorithm of Note (c) above cannot be used. The program must, therefore, invoke an assumption or convention for this situation, and the convention used is that for all of those elements on a cable whose orientation is not explicitly defined by the user, the local axis system is defined as follows. The local x-axis is coincident with the global X-axis. The plane formed by the x and y local axes is coincident with the plane of the structure. This means that for systems defined initially in the global XY plane, the local axes are coincident with the global axes. This is of little significance in most cases, except in two situations. The first is where you want to apply a rotational boundary condition at a node that is on a cable. In this case you must ensure that the displacement term associated with the rotational constraint causes the element to be aligned with the desired orientation. The second case is where a model combines both cable elements and elements which are not on any cable. In this case you have to be careful that the undeformed configuration does not contain a bend or 'kink' where a cable and rigid element meet at a node, due to the fact that these elements have different orientations. This situation is considerably less common than the first. These issues are not discussed further here - interested readers are instead referred to Mixing Cable and Rigid Elements and Rotational Boundary Conditions and Cables.

1.8.10.19 *ELEMENT,AUXILIARY

PURPOSE

To specify the connectivity of auxiliary elements in the model.

THEORY

Refer to Auxiliary Bodies for further information on this feature.

Note also that the old auxiliary body keywords (`*NODE,AUXILIARY`, `*ELEMENT,AUXILIARY`, `*AUXILIARY`, `*PANEL,AUXILIARY` and `*PANEL SECTIONS,AUXILIARY`) keywords have effectively been superseded by the integrated vessel/body features which provide a range of standard vessel and subsea component profiles. Refer to `*VESSEL,INTEGRATED` and `*BODY,INTEGRATED` for further information.

KEYWORD FORMAT

Two types of line that may be mixed and/or repeated as often as necessary.

Line defining a single element:

```
Element, Start Node, End Node [, Display Diameter]
```

Line generating a number of elements from an existing element:

```
GEN=Master Element, No. of Elements [, Element Increment] [, Display Diameter
```

If the generation option is used then the master element must be defined previously.

`Element Increment` defaults to 1. `Display Diameter` defaults to 0.1.

TABLE INPUT

Auxiliary Elements - Define Directly

Input:	Description
Element:	The number of the auxiliary element being defined.
First Node:	The number of the first auxiliary node of this element.
Last Node:	The number of the second auxiliary node of this element
Diameter:	A display diameter for the auxiliary element. This defaults to 0.1 if unspecified.

Auxiliary Elements - Generate

Input:	Description
Master Element:	The number of the auxiliary element defining the node numbering pattern to be copied in the generated elements.
Number of Elements:	The number of auxiliary elements to be generated, where this number includes the master element.

Element Increment:	The element number increment to be used in assigning numbers to the generated elements. The default value is 1, which will apply in the majority of cases.
Diameter:	A display diameter for the auxiliary elements. This defaults to 0.1 if unspecified.

NOTES

(a) Auxiliary nodes, elements and bodies allow you to include objects in your model that are not part of the finite element discretisation, but which are included for illustrative purposes only. Outline vessel models are the most common example of such objects.

(b) Auxiliary elements and structural elements (elements that are included in the finite element discretisation) cannot use the same element number. Each element in the model must be assigned a unique element number, irrespective of whether it is a structural element or an auxiliary element.

1.8.10.20 *ELEMENT SETS

PURPOSE

To group individual elements into element sets.

THEORY

Refer to Geometric Properties for further information on this feature.

KEYWORD FORMAT

Blocks of lines that define a set repeated as often as necessary. Each block begins with a line defining the set name. This is followed by various types of lines that define the elements in the set. These lines can be mixed and repeated as often as necessary until every element in the set is defined.

Line to define set name:

```
SET=Set Name
```

Line containing a list of elements. This line can contain up to 20 elements (numbers or labels). Any further elements must be defined on a new line.

```
List of Elements (Numbers or Labels)
```

Line defining a sequence of elements:

`GEN=`*`Start Element (Number or Label), End Element (Number or Label)`* `[,` *`Element`*

Line referencing another set of elements:

`SUBSET=`*`Subset Set Name`*

The set `ALL` is predefined and cannot be redefined. Every element assigned to a set must be defined using `*ELEMENT`. Set names are up to 256 characters long, can include spaces and are case insensitive. If you specify an element label rather than an element number, it must be enclosed in {} brackets. *`Element Increment`* defaults to one.

TABLE INPUT

Input:	Description
Set Name:	A unique label for the element set. Set names are not case sensitive, so the set name 'RISER' is equivalent to 'Riser', which is in turn equivalent to 'riser'.
Elements:	The elements comprising the set. These can be input in three ways, namely: (i) A list of elements (numbers or labels), such as for example "1, 5, 7". (ii) A group of consecutive elements (numbers or labels), input using the format: "11 – 15". This definition specifies Elements 11 to 15 inclusive. The specification "11 - 15 – 2" can be used to specify the Elements 11, 13 and 15 - that is, from Element 11 to Element 15 in steps of 2. (iii) Another set name. For example you might define three sets named SET_1, SET_2 and SET_3, and then combine them in a further set, say ALLSETS, by inputting "SET_1, SET_2, SET_3". If you specify an element label rather than an element number, it must be enclosed in {} brackets.

	All three specifications can be combined, as for example in “1, 7, 9, 12-15, 17, 20-50-10, RISER”. This set combines Elements 1, 7, and 9; elements 12 to 15 inclusive; Element 17; Elements 20, 30, 40 and 50; and the elements comprising the set RISER.
SubSets:	An additional element set or sets whose elements are to be added to the current set definition. If more than one set is referenced, use commas to separate out the set names.

NOTES

(a) Use as many lines as you need to completely define the elements comprising a particular set. Simply leave the first column blank for second and subsequent lines.

(b) If a set name is included in the specification of another set, then obviously the elements comprising that set must be separately defined.

(c) There is one predefined element set in Flexcom, which is named All. Not surprisingly this comprises all of the elements of the finite element discretisation and is the default element set. Note that Flexcom will resist any attempt to redefine the make-up of the set All.

(d) The names and composition of element sets you define in a preceding section carry through to all dependent (restart) sections. For example, if you define an element set in the `$MODEL` section, it will automatically be available in a subsequent `$DATABASE POSTPROCESSING` section. So there is no need to repeat the specifications again – you can just use the set names directly. If you redefine the composition of a set previously defined in a preceding section, Flexcom will output a warning, but will continue with the most recent set definition.

1.8.10.21 *EMBEDMENT

PURPOSE

To define force-embedment curves for an elastic seabed.

THEORY

Refer to Embedment for further information on this feature.

Note also that the old *EMBEDMENT keyword has effectively been superseded by the new seabed definition keywords which facilitate the modelling of arbitrary seabed profiles. Refer to *SEABED STIFFNESS for further information.

KEYWORD FORMAT

A block of lines that define a force-embedment curve repeated as often as necessary. The block begins with a line defining the element set to which the curve applies. It is followed by as many lines as necessary to define each point on the curve.

Line defining the element set:

```
SET=Element Set
```

Line defining a point on a curve:

```
Embedment Ratio, Seabed Force
```

Each curve must have at least two points defined. A force-embedment curve may only be specified for an elastic seabed. *EMBEDMENT is required if no elastic seabed stiffness is input. You can combine linear and non-linear seabeds in the same model. For elements in the set for the non-linear seabed, the non-linear curve will be used. The linear stiffness will apply for any element not in the non-linear seabed set. Several non-linear seabeds may also be specified (for different sets) in the same model.

TABLE INPUT

Input:	Description
Set Name:	The name of the element set associated with this embedment curve. The default is all elements in contact with the seabed.
Embedment Ratio:	An embedment ratio value on the curve.
Seabed Force:	The force exerted by the seabed for this embedment ratio.

NOTES

(a) Flexcom provides an option to specify that seabed stiffness varies with the degree of embedment of a riser or pipeline. Furthermore, the facility is available to specify different stiffness characteristics for different element sets.

(b) The embedment ratio of an element is defined as the average distance the element centreline lies below the seabed, divided by element external diameter. So embedment ratio is dimensionless. “Average distance” in this context means the average of the distances below the seabed of the two nodes on the element.

(c) How this facility operates is as follows. Flexcom computes the average embedment of an element lying on the elastic seabed at each iteration at each solution time. Then using the embedment curve for the element that you specify here, the tangent stiffness of the curve at that embedment is calculated. This value is used as the vertical seabed stiffness for that element.

(d) You should use as many lines as you need to completely define a force/embedment ratio curve for a particular element set. Simply leave Column 1 blank for second and subsequent lines. For subsequent element sets, put the set name in Column 1 and specify the force/embedment ratio data in the same way.

(e) The facility to specify different embedment curves for different element sets is provided for complete generality. However in the majority of applications the same curve will apply everywhere, which is why the default element set is all elements. Of course, *All* in this context means only all elements in contact with the elastic seabed.

(f) The points defining the force/embedment curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of embedment ratio.

(g) If the computed embedment ratio of an element lies between the specified force/embedment data points that you specify, Flexcom uses linear interpolation to determine the seabed force on the element.

(h) If the computed embedment ratio of an element lies outside the specified range of the force/embedment curve, then Flexcom assumes:

- For embedment ratios greater than the range of the curve, the tangent stiffness of the element is equal to the tangent stiffness between the final two points defined on the curve.

- For embedment ratios less than the range of the curve, the tangent stiffness of the element is equal to the tangent stiffness between the first two points defined on the curve.

1.8.10.22 *EQUIVALENT

PURPOSE

To specify that two individual nodes are a single equivalent node.

THEORY

Refer to Equivalent Nodes for further information on this feature.

KEYWORD FORMAT

One type of line repeated as often as necessary.

```
First Node (Number or Label), Second Node (Number or Label)
```

If you specify a node label rather than a node number, it must be enclosed in {} brackets.

TABLE INPUT

Input:	Description
First Node:	The first node (number or label) of two nodes that are to be treated as a single equivalent node. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Second Node:	The second node (number or label) of two nodes node that are to be treated as a single equivalent node. If you specify a node label rather than a node number, it must be enclosed in {} brackets.

NOTES

(a) When two nodes are defined as equivalent, they must both have the same nodal coordinates.

(b) Internally Flexcom treats equivalent nodes as a single node. So nodal data (such as nodal motions etc.) are output by the program for the first-specified of the two nodes only.

1.8.10.23 *FLEX JOINT

PURPOSE

To define flex joint elements in the structural discretisation.

THEORY

Refer to Hinge and Flex Joints for further information on this feature.

KEYWORD FORMAT

A block of lines, the first three of which define (i) the flex joint element or element set, (ii) the weights in air and water of the flex joint, and (iii) the flex joint type. The format of subsequent lines depends on whether the flex joint is linear or non-linear.

Lines defining the flex joint element or element set, the weights in air and water of the flex joint and the flex joint type:

```
[ELEMENT=Element (Number or Label)] or [SET=Element Set]
Weight in Water, Weight in Air
TYPE=Flex Joint Type
```

For linear flex joints, the next line simply specifies the rotational stiffness of the flex joint:

```
Stiffness
```

For non-linear flex joints, the following lines are used to define a moment-angle curve for the flex joint, with each line defining a single moment-angle data pair:

```
Moment, Angle
```

This line may be repeated as often as necessary to fully define the moment-angle curve.

You must specify either one of ELEMENT=*Element (Number or Label)* or SET=*Element Set*. The weight in air of the flex joint must be greater than or equal to the weight in water of the flex joint. At least two moment-angle data points must be specified for non-linear flex joints. *Flex Joint Type* may be either LINEAR or NONLINEAR. If you specify an element label rather than an element number, it must be enclosed in {} brackets.

FLEX JOINTS

Table Input

Input:	Description

Element Set/Element:	The element set or the element (number or label) to which the flex joint properties are being assigned. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Weight in Water:	The total weight in water of the flex joint. This entry is optional, and defaults to 0.
Weight in Air:	The total weight in air of the flex joint. This entry is optional, and defaults to 0.
Rotational Stiffness:	The rotational stiffness of the flex element or the name of a moment-angle curve which defines the rotational stiffness.

Notes

(a) Flex joints are classified as being linear or non-linear. Linear flex joints are characterised by a (non-zero) rotational stiffness, while the behaviour of non-linear flex joints is defined in terms of a moment-angle curve.

(b) You can assign flex joint properties to a single element or a set of elements defined in the usual way. In the former case you input a number in Column 1, in the latter case you specify the name of the element set.

(c) If a non-linear material curve name is specified for the rotational stiffness, then the non-linear moment-angle curve must be defined in the *Moment Angle Curve* table.

(d) The units for rotational stiffness are moment/degree.

MOMENT-ANGLE CURVES

Table Input

Input:	Description
Curve Name:	The generic name of the moment-angle curve.
Moment :	A moment value for a point on the curve.

Angle:	The corresponding angle value (in degrees).

Notes

(a) This table is used to define moment-angle curves that define the behaviour of non-linear flex joints for a particular flex joint element or set of flex joint elements. Moment-angle curves are assigned to flex joint elements using the *Define Flex Joints* table.

(b) Use as many lines as you need to completely define a particular moment-angle curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent moment-angle curves, put the curve name in Column 1 and specify the moment-angle data in the same way.

(c) The points defining the non-linear moment-angle curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of strain.

(d) If the rotation of the non-linear flex joint element lies between the specified moment-angle data points, Flexcom uses linear interpolation to determine the relevant rotational stiffness of the element.

(e) If the rotation of the non-linear flex joint element lies outside the specified range of the moment-angle curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

(f) If none of the specified angle terms have a negative value, the curve is assumed to be symmetrical about the origin (i.e. the behaviour of the flex joint is the same for both positive and negative strains).

1.8.10.24 *FLOATING BODY

PURPOSE

To define a floating body and its associated properties.

THEORY

Refer to Floating Body for further information on this feature.

KEYWORD FORMAT

Block of lines consisting of floating body name followed by data specified in blocks, with each block beginning with a TYPE=line defining the properties of the floating body. The whole block is then repeated for different floating bodies.

Line to define floating body name:

```
FLOATING BODY=Floating Body Name
```

Block of data defining the floating body inertia:

```
TYPE=MASS
M11, I44, I55, I66
```

Block of data defining the floating body hydrostatic stiffness:

```
TYPE=STIFFNESS
K33, K44, K55, [K34], [K35], [K45], [K46], [K56]
```

Block of data defining the floating body geometric properties:

```
TYPE=GEOM
AL, AT, Lfore, Laft, T, B
```

Block of data defining nodes at critical locations on the floating body, along with its initial undisplaced orientation:

```
TYPE=NODE
CoG Node (Number or Label), θG, CoB Node (Number or Label), RAO Node (Number
```

Block of data defining the floating body reference and cut-off frequencies:

```
TYPE=FREQUENCY
Reference, Cut-off
```

Block of data defining parameters which control floating body retardation functions:

```
TYPE=RETARDATION
Time Length, Time Step, Frequency Increments, Scale Disp., Scale Rot.
```

where:

- M_{11} = M_{22} = M_{33} = mass
- I_{44} = roll inertia
- I_{55} = pitch inertia
- I_{66} = yaw inertia
- K_{33} = heave stiffness

- K_{44} = roll stiffness
- K_{55} = pitch stiffness
- K_{34} = heave-roll stiffness coupling
- K_{35} = heave-pitch stiffness coupling
- K_{45} = roll-pitch stiffness coupling
- K_{46} = yaw-roll stiffness coupling
- K_{56} = yaw-pitch stiffness coupling
- A_L = the longitudinal area exposed to wind action
- A_T = the transverse area exposed to wind action
- L_{fore} = the length from the centre of gravity to the forward perpendicular
- L_{aft} = the length from the centre of gravity to the aft perpendicular
- T = draft
- B = beam
- CoG Node = the Centre of Gravity node (number or label)
- θ_G = the heading of the undisplaced floating body (in degrees), measured anticlockwise from the global Y-axis
- CoB Node = the Centre of Buoyancy node (number or label)
- RAO Node = the force RAO application node (number or label)
- CoD Node = the Centre of Drag node (number or label)

- Scale Disp. = the scale factor for output of displacements.
- Scale Rot. = the scale factor for output of rotations.

The order in which the TYPE= statement appears is not significant, as long as the relevant data is given under each statement. The initial draft of the floating body should be at the level where the weight and buoyancy are in equilibrium. If you specify a node label rather than a node number, it must be enclosed in {} brackets.

DEFINE

Table Input

Input:	Description
Floating Body:	The name of the floating body.
Inertia:	The name of the inertia definition.
Stiffness:	The name of the hydrostatic stiffness definition.
Geometry:	The name of the geometry definition.
Location:	The name of the location definition.
Frequency:	The name of the frequency definition.
Retardation:	The name of the retardation function parameter definition

Notes

(a) Any properties which you refer to for a particular floating body must be defined in other tables, such as *Body - Hydrostatic Stiffness, Body - Geometry*, etc.

BODY - FREQUENCY-INDEPENDENT INERTIA

Table Input

Input:	Description

Name:	The name of the inertia definition.
Mass:	The (translational) mass.
Roll Inertia:	The inertia about the local roll axis.
Pitch Inertia:	The inertia about the local pitch axis.
Yaw Inertia:	The inertia about the local yaw axis

BODY - HYDROSTATIC STIFFNESS

Table Input

Input:	Description
Name:	The name of the hydrostatic stiffness definition.
Heave:	The hydrostatic stiffness in the local heave direction.
Roll:	The hydrostatic stiffness in the local roll direction.
Pitch:	The hydrostatic stiffness in the local pitch direction.
Heave-Roll:	The hydrostatic stiffness coupling term between heave and roll. See Note (a).
Heave-Pitch:	The hydrostatic stiffness coupling term between heave and pitch. See Note (a).
Roll-Pitch:	The hydrostatic stiffness coupling term between roll and pitch. See Note (a).
Yaw-Roll:	The hydrostatic stiffness term that couples roll from yaw. See Note (b).

Yaw-Pitch:	The hydrostatic stiffness term that couples roll from pitch. See Note (b).

Notes

(a) The coupling stiffness terms Heave-Roll, Heave-Pitch and Roll-Pitch are mutually symmetric, i.e. $K_{34} = K_{43}$, $K_{35} = K_{53}$, $K_{45} = K_{54}$.

(b) The coupling stiffness terms Yaw-Roll (K_{46}) and Yaw-Pitch (K_{56}) are not mutually symmetric, i.e. K_{64} and K_{65} are always zero.

BODY - GEOMETRY

Table Input

Input:	Description
Name:	The name of the geometry definition.
Longitudinal Body Area:	The longitudinal body area. See Note (a).
Transverse Body Area:	The transverse body area. See Note (a).
Forward CoG Length:	The length from the Centre of Gravity to the forward perpendicular. See Notes (a) and (b).
Aft CoG Length:	The length from the Centre of Gravity to the aft perpendicular. See Note (a) and (b).
Draft:	The floating body draft. See Notes (b) and (c).
Beam:	The floating body beam. See Note (b).

Notes

(a) The two area inputs and the two length inputs are used together to compute the wind loading on the floating body.

(b) The two length inputs together with the *Draft* and *Beam* values are used to compute the current loading on the floating body.

(c) The initial draft of the floating body should be at the level where the weight and buoyancy of the floating body are in equilibrium (the body is floating). This is implicitly assumed by Flexcom, so that changes in the buoyancy loads, due to relative floating body/surface elevation changes, are accounted for by the buoyancy terms in the hydrostatic stiffness matrix.

BODY - NODAL LOCATIONS

Table Input

Input:	Description
Name:	The name of the location definition.
CoG Node:	The node (number or label) located at the Centre of Gravity.
Orientation:	The initial undisplaced orientation of the floating body, specified in degrees anticlockwise from the global Y-Direction. The default is 0°.
CoB Node:	The node (number or label) located at the Centre of Buoyancy.
RAO Reference Node:	The node (number or label) located at the force RAO application point.
CoD Node:	The node (number or label) located at the Centre of Drag.

BODY - FREQUENCY

Table Input

Input:	Description
Name:	The name of the frequency definition.

Reference Frequency:	The reference frequency in Hz. See Note (a).
Cut-off Frequency:	The cut-off frequency in Hz. See Notes (b) and (c).

Notes

(a) The specified *Reference Frequency* is used in determining the added mass of the floating body. Refer to Wave Radiation Loads for further information on the convolution integral technique employed by Flexcom to model frequency dependent added mass and radiation damping terms in a time domain simulation. This input is relevant to time domain analyses only.

(b) The specified *Cut-off Frequency* is used to distinguish low and wave frequency regimes. Frequency-dependent added mass and damping apply in the high frequency regime, while low frequency added mass and damping, taken at the cut-off frequency, apply below the cut-off frequency value.

(c) If a *Cut-off Frequency* value is specified for a particular floating body, then added mass and damping must also be defined for that same floating body at the cut-off frequency in the *Added Mass Cut-off* and *Radiation Damping Cut-off* tables, respectively.

BODY - RETARDATION

Table Input

Input:	Description
Name:	The name of the retardation function parameter definition.
Time Length:	The length of time over which the retardation functions are to be computed. See Note (a).
Time Increment:	The time increment to be used in the retardation function computations. See Note (a).

No. of Frequency Increments:	The number of frequency increments to be used in the retardation function computations. See Note (a).
Displacement Scale Factor:	A scale factor for output of displacements. See Note (b).
Rotation Scale Factor:	A scale factor for output of rotations. See Note (b).

Notes

(a) The retardation functions for a floating body are evaluated over a specified *Time Length* at regular intervals of *Time Increment*. If a time length is not specified, it defaults to 100s (or the total simulation time, if this is less than 100s). The time increment defaults to the analysis time step in the case of fixed time step analyses, and the minimum time step for variable time step analyses. At each time step, the value of each retardation function is computed by integrating the appropriate frequency dependent radiation damping coefficients with respect to frequency using the specified *No. Of Frequency* Increments. If a number of frequency increments is not specified, a default value of 200 is used by the program. The background and theory behind the time domain coupled analysis procedure is described in detail in Wave Radiation Loads.

(b) Flexcom provides a facility whereby the computed retardation functions may be examined and verified by the user. Specifically, the computed functions are echoed to an ACSII file, entitled 'Ret_Fn_I.dat', where I is an integer value indicating the number of the relevant floating body. The output displacement terms are scaled by the specified *Displacement Factor*, the rotation terms by the Rotation Factor, and the coupled displacement/rotation terms by an average of the specified factors. Both the displacement and rotation factors default to 1 if unspecified. The layout of the output file is as follows:

Time, R_{11}, …, R_{22}, …, R_{33}, …, R_{44}, …, R_{55}, …, R_{66}

where:

- Time is the time value in seconds

- R_{11} is the heave retardation function value
- R_{22} is the surge retardation function value
- R_{33} is the sway retardation function value
- R_{44} is the yaw retardation function value
- R_{55} is the roll retardation function value
- R_{66} is the pitch retardation function value

and the intermediate terms (e.g. R_{12}, R_{13}, R_{23} etc.) represent the coupled retardation function terms between various degrees of freedom at the relevant time.

1.8.10.25 *FORCE-STRAIN

PURPOSE

To define force-strain curves for non-linear materials.

THEORY

Refer to Non-Linear Elastic materials for further information on this feature.

Note also that the old `*STRESS/STRAIN` keyword has effectively been superseded by the new non-linear material definition keywords which explicitly distinguish between bending, axial and torsional stiffness. Refer to related keywords `*MOMENT-CURVATURE` and `*TORQUE-TWIST`.

KEYWORD FORMAT

A block of lines that defines a force-strain curve, repeated as often as necessary. The block begins with a line defining the curve name. It is followed by as many lines as necessary to define each point on the curve.

Line defining the curve name:

```
CURVE=Curve Name
```

Line defining a point on a curve:

```
Force, Strain
```

Each curve must have at least two points defined. This type of force-strain curve may not be associated with non-linear beam elements which are defined using the rigid riser format for geometric properties specification.

TABLE INPUT

Input:	Description
Curve Name:	The generic name of the force-strain curve.
Force:	A force value for a point on the curve.
Strain:	The corresponding strain value.

NOTES

(a) This keyword is used to define force-strain curves that define EA for a particular set of elements. Force-strain curves may be assigned to element sets using the `*GEOMETRIC SETS` keyword.

(b) Use as many lines as you need to completely define a particular force-strain curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent force-strain curves, put the curve name in Column 1 and specify the force-strain data in the same way.

(c) The points defining the non-linear force-strain curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of strain.

(d) If the strain in an element lies between the force-strain data points you specify, Flexcom uses linear interpolation to determine the relevant stiffness for the element.

(e) If the strain in the element lies outside the specified range of the force-strain curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

(f) If none of the specified strain terms have a negative value, the curve is assumed to be symmetrical about the origin (i.e. the behaviour of the element is the same for both positive and negative strains).

1.8.10.26 *GEOMETRIC SETS

PURPOSE

To assign geometric properties to element sets.

THEORY

Refer to Geometric Properties for further information on this feature.

KEYWORD FORMAT

A block of three lines, the first one of which is an optional line to specify the format in which the properties are input, which can be FLEXIBLE, RIGID or MOORING. This is followed by two lines defining the element properties. The block may be repeated to allow combinations of flexible riser, rigid riser and mooring line formats in one model.

A block of lines that make up the block when the format is for flexible risers:

```
[OPTION=FLEXIBLE]
SET=Set Name [, BUOYANCY=DISTRIBUTED] [, EIYY=Curve Name] [, EIZZ=Curve Name]
[EIyy], [EIzz], [GJ], [EA], m, p, Di, Dd, Db [, Do, Dc]
[COMPRESSION CHECK=Check Setting]
```

A block of lines that make up the block when the format is for rigid risers with only elastic properties:

```
OPTION=RIGID
SET=Set Name [, BUOYANCY=DISTRIBUTED] [, E=Curve Name]
[E], G, Do, Di, rho [, A, I, J, Dd, Db, Dc]
[COMPRESSION CHECK=Check Setting]
```

A block of lines that make up the block when the format is for rigid risers with elastic-plastic properties:

```
OPTION=RIGID
SET=Set Name [, BUOYANCY=DISTRIBUTED], PLASTIC HARDENING=Plastic Hardening Mo
E, G, Do, Di, rho [, A, I, J, Dd, Db, Dc]
[COMPRESSION CHECK=Check Setting]
```

A block of lines that make up the block when the format is for mooring lines:

```
[OPTION=MOORING]
SET=Set Name
EA, m, Do [, Dd, Db]
```

A block of lines that make up the block when the format is for truss elements:

```
[OPTION=TRUSS]
SET=Set Name [, EA=Curve Name]
[EA], m, Do [, Di, Dd, Db, Dc]
```

If the format option is not specified at all then it defaults to FLEXIBLE. If it is invoked for one set and then not specified explicitly for subsequent sets, then the specified format applies to the subsequent sets.

Where a non-linear material curve is specified on the SET= line, the corresponding numerical value on the next should be omitted. If a non-linear material curve name is specified for E, then the non-linear material curve must be defined in the `*STRESS/STRAIN DIRECT` keyword. If a non-linear material curve name is specified for any of the inputs EIyy, EIzz, GJ or EA, then the non-linear material curve must be defined in the `*MOMENT-CURVATURE` (EIyy & EIzz), `*FORCE-STRAIN` (EA) or `*TORQUE-TWIST` (GJ) keywords. If bending hysteresis is included in the model, you must use the `*BENDING HYSTERESIS` keyword. If a plastic hardening model name is specified, then this must be defined in the `*PLASTIC HARDENING` keyword.

For *Flexible Riser* format, the value for Do is only used when distributed buoyancy is selected. Also, the use of the EI=option precludes the use of both the EIyy=and EIzz=inputs.

`Check Setting` may be either AUTOMATIC (the default), NONE or a numeric value.

FLEXIBLE RISER GEOMETRIC PROPERTIES

Table Input

Input:	Description
Set Name:	The element set to which the geometric properties are to be assigned. This defaults to all elements.
Bending Stiffness:	This option allows you to specify how the bending stiffness of the elements of the set is calculated, and is relevant to element sets whose bending stiffness is defined in terms of a non-linear material M-k curve or a non-linear hysteresis curve. The options are *Linear* (the default), *Symmetric Non-Linear, Asmmetric Non-Linear* and *Hysteresis*. See Note (b).
EIyy:	The bending stiffness about the local y-axis for the elements of the set or the name of a non-linear material stress/strain curve that defines this bending stiffness. See Note (c).

EIzz:	The bending stiffness about the local z-axis for the elements of the set or the name of a non-linear material stress/strain curve that defines this bending stiffness. See Note (c).
GJ:	The torsional stiffness for the elements of the set or the name of a non-linear material stress/strain curve that defines the torsional stiffness. See Notes (c) and (d).
EA:	The axial stiffness for the elements of the set or the name of a non-linear material stress/strain curve that defines the axial stiffness. See Note (c).
m:	The mass per unit length for the elements of the set.
p:	The polar inertia of cross section per unit length for the elements of the set. The units are mass by length. See Note (e).
Di:	The internal diameter of the elements of the set. This is used for computing the buoyancy contribution of the internal fluid, if there is any. The default is an internal diameter of 0.
Dd:	The drag diameter. This is the effective outer diameter for hydrodynamic force evaluation using Morison's Equation.
Db:	The buoyancy diameter. This is the effective outer diameter for buoyancy force calculations for the elements comprising the set.
Do:	The actual outer diameter of the elements of the set, not including additional external buoyancy material. This data is used only if the *Distributed* buoyancy option is invoked, or if any external coatings are applied.
Dc:	The effective diameter for contact calculations. This is relevant only when your Flexcom analysis includes guide surfaces or line clashing. See Note (g).
Buoyancy:	This option allows you to specify how the buoyancy forces experienced by the elements of the set are calculated. The options are *Default* and *Distributed*. See Note (f).

Compression Check:	Flexcom will issue a warning if the compressive load experienced in any element exceeds the critical Euler load. Refer to Compression and Buckling for further information. The options are *Automatic* (the default), *Manual* and *None*. If you opt for *Manual*, you must explicitly set a *Compression Limit* also.
Compression Limit:	A manually defined compression limit to be used in the Euler load check.

Notes

(a) For a discussion of the different formats available to you to define geometric properties, and the program options for specifying non-linear material properties, refer to Geometric Properties in Flexible Riser Format.

(b) Generally speaking, you normally input the same non-linear relationship for EIyy and EIzz. In the program terminology, the *Symmetric* non-linear bending model involves the computation of a single bending stiffness EI at any solution time based on the total curvature k, with both EIyy and EIzz being set equal to EI. This means that the element stiffness will be same in any situation where k is the same, regardless of the values of the individual curvature components. An option to model Asymmetric non-linear bending behaviour is also provided. In this case, EIyy at any solution time is found from the curve you input for that stiffness term based on the instantaneous value of the corresponding local curvature term ky, and likewise EIzz is found based on instantaneous kz. This is true even if you input the same curve name for EIyy and EIzz. This can mean that bending response can differ for the same loading depending on element orientation; an orientation where ky and kz are non-zero can give a different response to an orientation with either ky or kz equal to zero.

(c) If a non-linear material curve name is specified for any of the inputs EIyy, EIzz, GJ or EA, then the non-linear material curve must be defined in the `*MOMENT-CURVATURE` (EIyy & EIzz), `*FORCE-STRAIN` (EA) or `*TORQUE-TWIST` (GJ) keywords. Note also that you may specify a combination of linear and non-linear material properties; for example, you might specify non-linear bending characteristics (EIyy and EIzz), and linear EA and GJ (that is, a single value for each of these).

(d) The units of torsional stiffness in Flexcom are [Force][Length**2]/[Radian].

(e) A discussion of the significance and calculation of the polar inertia per unit length term can be found in Mass and Polar Inertia per Unit Length.

(f) Flexcom provides two options for specifying how buoyancy forces generated by elements are determined; *Default* and *Distributed*. For the majority of analyses, the default buoyancy formulation provides the most realistic and accurate approach to modelling the buoyancy forces on the elements. Please refer to Buoyancy Formulations for a detailed description of both approaches.

(g) Contact diameter is relevant only when your Flexcom analysis includes guide surfaces or line clashing. Flexcom uses this input to determine when contact occurs. Note that this value is used only in contact calculations in the main Analysis module - it is not used by the clearance/interference postprocessing module Clear. Specification of a contact diameter is optional, and Dc defaults to the maximum of Dd, Db and Do if omitted.

(h) There are a number of different element diameters used by Flexcom. In the main *Analysis* module you can specify internal diameter, drag diameter, buoyancy diameter, outer diameter and contact diameter. In addition to these fundamental model inputs, you can specify separate internal and outer diameters for use in stress computations during postprocessing, and you can do this in either the main *Analysis* or *Database Postprocessing* modules. So it is conceivable that you could specify the outer diameter for a given element in three different places, and specify three different values if you so wish. Naturally, such a scenario can appear confusing, particularly for new users of the software. Refer to Diameter Inputs for a detailed discussion on the significance of each diameter input, in order to eliminate any possible ambiguity.

RIGID RISER GEOMETRIC PROPERTIES - ELASTIC

Table Input

Input:	Description
Set Name:	The element set to which the geometric properties are to be assigned. This defaults to all elements.
E:	The Young's Modulus for the elements of the set or the name of a non-linear material stress/strain curve (see Note (e)).

G:	The Shear Modulus for the elements of the set.
Do:	The outer diameter of the elements of the set. This is used for computing the area, moment of inertia and polar moment of inertia of the elements, if these are not specified directly. It is also used as the default drag and buoyancy diameters.
Di:	The internal diameter of the elements of the set. This is also used for computing the area, moment of inertia and polar moment of inertia of the elements, if these are not specified directly. It is also used for computing the buoyancy contribution of the internal fluid, if there is any. The default internal diameter is zero, although this would normally be a positive value.
rho:	The mass density (mass per unit volume) of the material for the elements of the set.
A:	The cross-sectional area of the elements of the set. This entry is optional. See Note (c).
I:	The moment of inertia (second moment of area) of the elements of the set. This entry is optional. See Note (c).
J:	The polar moment of inertia of the elements of the set. This entry is optional. See Note (c).
Dd:	The drag diameter. This is the effective outer diameter for hydrodynamic force evaluation using Morison's Equation. This entry is optional. The drag diameter defaults to the outer diameter Do.
Db:	The buoyancy diameter. This is the effective outer diameter for buoyancy force calculations for the elements comprising the set. This entry is optional. The buoyancy diameter defaults to the outer diameter Do.

Dc:	The effective diameter for contact calculations. This is relevant only when your Flexcom analysis includes guide surfaces or line clashing. See Note (f).
Buoyancy:	This option allows you to specify how the buoyancy forces experienced by the elements of the set are calculated. The options are *Default* and *Distributed*. See Note (d).
Compression Check:	Flexcom will issue a warning if the compressive load experienced in any element exceeds the critical Euler load. Refer to Compression and Buckling for further information. The options are *Automatic* (the default), *Manual* and *None*. If you opt for *Manual*, you must explicitly set a *Compression Limit* also.
Compression Limit:	A manually defined compression limit to be used in the Euler load check.

Notes

(a) For a discussion of the different formats available to you to define geometric properties, and the program options for specifying non-linear material properties, refer to Geometric Properties in Rigid Riser Format.

(b) You must specify non-zero values for E, G and Do, and non-zero values for rho and Di would also be normal. E may be specified as a numerical value or defined in terms of a non-linear material stress/strain curve.

(c) If you do not specify values for A, I and J, Flexcom calculates them from Do and Di using the standard relations. You can specify values for one or two only of these three inputs and leave the others for Flexcom to calculate (for example, you might specify A only and let the program base I and J on Do and Di).

(d) Flexcom provides two options for specifying how buoyancy forces generated by elements are determined; *Default* and *Distributed*. For the majority of analyses, the default buoyancy formulation provides the most realistic and accurate approach to modelling the buoyancy forces on the elements. Please refer to Buoyancy Formulations for a detailed description of both approaches.

(e) If a non-linear material curve name is specified for E, then the non-linear material curve must be defined in the `*STRESS/STRAIN DIRECT` keyword.

(f) Contact diameter is relevant only when your Flexcom analysis includes guide surfaces or line clashing. Flexcom uses this input to determine when contact occurs. Note that this value is used only in contact calculations in the main Analysis module - it is not used by the clearance/interference postprocessing module Clear. Specification of a contact diameter is optional, and Dc defaults to the maximum of Dd, Db and Do if omitted.

(g) There are a number of different element diameters used by Flexcom. In the main *Analysis* module you can specify internal diameter, drag diameter, buoyancy diameter, outer diameter and contact diameter. In addition to these fundamental model inputs, you can specify separate internal and outer diameters for use in stress computations during postprocessing, and you can do this in either the main *Analysis* or *Database Postprocessing* modules. So it is conceivable that you could specify the outer diameter for a given element in three different places, and specify three different values if you so wish. Naturally, such a scenario can appear confusing, particularly for new users of the software. Refer to Diameter Inputs for a detailed discussion on the significance of each diameter input, in order to eliminate any possible ambiguity.

RIGID RISER GEOMETRIC PROPERTIES - PLASTIC HARDENING

Table Input

Input:	Description
Set Name:	The element set to which the geometric properties are to be assigned. This defaults to all elements.
Plastic Hardening:	The name of the plastic hardening model associated with this element set. The plastic hardening model must be defined in the `*PLASTIC HARDENING` keyword.
E:	The Young's Modulus for the elements of the set.
G:	The Shear Modulus for the elements of the set. See Note (b).

Do:	The outer diameter of the elements of the set. This is used for computing the area, moment of inertia and polar moment of inertia of the elements, if these are not specified directly. It is also used as the default drag and buoyancy diameters.
Di:	The internal diameter of the elements of the set. This is also used for computing the area, moment of inertia and polar moment of inertia of the elements, if these are not specified directly. It is also used for computing the buoyancy contribution of the internal fluid, if there is any. The default internal diameter is zero, although this would normally be a positive value.
rho:	The mass density (mass per unit volume) of the material for the elements of the set.
A:	The cross-sectional area of the elements of the set. This entry is optional. See Note (c).
I:	The moment of inertia (second moment of area) of the elements of the set. This entry is optional. See Note (c).
J:	The polar moment of inertia of the elements of the set. This entry is optional. See Note (c).
Dd:	The drag diameter. This is the effective outer diameter for hydrodynamic force evaluation using Morison's Equation. This entry is optional. The drag diameter defaults to the outer diameter Do.
Db:	The buoyancy diameter. This is the effective outer diameter for buoyancy force calculations for the elements comprising the set. This entry is optional. The buoyancy diameter defaults to the outer diameter Do.
Dc:	The effective diameter for contact calculations. This is relevant only when your Flexcom analysis includes guide surfaces or line clashing. See Note (e).

Buoyancy:	This option allows you to specify how the buoyancy forces experienced by the elements of the set are calculated. The options are *Default* and *Distributed*. See Note (d).
Compression Check:	Flexcom will issue a warning if the compressive load experienced in any element exceeds the critical Euler load. Refer to Compression and Buckling for further information. The options are *Automatic* (the default), *Manual* and *None*. If you opt for *Manual*, you must explicitly set a *Compression Limit* also.
Compression Limit:	A manually defined compression limit to be used in the Euler load check.

Notes

(a) For a discussion of the different formats available to you to define geometric properties, and the program options for specifying non-linear material properties, refer to Geometric Properties in Rigid Riser Format.

(b) You must specify non-zero values for E, G and Do, and non-zero values for rho and Di would also be normal. E may only be specified as a numerical value. The value of G will be overwritten to account for the Poisson's ratio.

(c) If you do not specify values for A, I and J, Flexcom calculates them from Do and Di using the standard relations. You can specify values for one or two only of these three inputs and leave the others for Flexcom to calculate (for example, you might specify A only and let the program base I and J on Do and Di).

(d) Flexcom provides two options for specifying how buoyancy forces generated by elements are determined; *Default* and *Distributed*. For the majority of analyses, the default buoyancy formulation provides the most realistic and accurate approach to modelling the buoyancy forces on the elements. Please refer to Buoyancy Formulations for a detailed description of both approaches.

(e) Contact diameter is relevant only when your Flexcom analysis includes guide surfaces or line clashing. Flexcom uses this input to determine when contact occurs. Note that this value is used only in contact calculations in the main Analysis module - it is not used by the clearance/interference postprocessing module Clear. Specification of a contact diameter is optional, and Dc defaults to the maximum of Dd, Db and Do if omitted.

(f) There are a number of different element diameters used by Flexcom. In the main *Analysis* module you can specify internal diameter, drag diameter, buoyancy diameter, outer diameter and contact diameter. In addition to these fundamental model inputs, you can specify separate internal and outer diameters for use in stress computations during postprocessing, and you can do this in either the main *Analysis* or *Database Postprocessing* modules. So it is conceivable that you could specify the outer diameter for a given element in three different places, and specify three different values if you so wish. Naturally, such a scenario can appear confusing, particularly for new users of the software. Refer to Diameter Inputs for a detailed discussion on the significance of each diameter input, in order to eliminate any possible ambiguity.

MOORING LINE GEOMETRIC PROPERTIES

Table Input

Input:	Description
Set Name:	The element set to which the geometric properties are to be assigned. This defaults to all elements.
EA:	The axial stiffness for the elements of the set.
m:	The mass per unit length for the elements of the set.
Do:	The outer diameter of the elements of the set.
Dd:	The drag diameter. This is the effective outer diameter for hydrodynamic force evaluation using Morison's Equation. The drag diameter defaults to the outer diameter Do.

Db:	The buoyancy diameter. This is the effective outer diameter for buoyancy force calculations for the elements comprising the set. The buoyancy diameter defaults to the outer diameter Do.

Notes

(a) For a discussion of different formats available to you to define geometric properties, and the program options for specifying non-linear material properties, refer to Geometric Properties in Mooring Line Format.

(b) There are a number of different element diameters used by Flexcom. In the main Analysis module you can specify outer diameter, drag diameter and buoyancy diameter. In addition to these fundamental model inputs, you can specify separate diameters for use in stress computations during postprocessing, and you can do this in either the main *Analysis* or *Database Postprocessing* modules. So it is conceivable that you could specify the outer diameter for a given element in three different places, and specify three different values if you so wish. Naturally, such a scenario can appear confusing, particularly for new users of the software. Refer to Diameter Inputs for a detailed discussion on the significance of each diameter input, in order to eliminate any possible ambiguity.

TRUSS ELEMENT GEOMETRIC PROPERTIES

Table Input

Input:	Description
Set Name:	The element set to which the geometric properties are to be assigned. This defaults to all elements.
EA:	The axial stiffness for the elements of the set or the name of a non-linear material stress/strain curve that defines the axial stiffness. See Note (b).
m:	The mass per unit length for the elements of the set.
Do:	The outer diameter of the elements of the set.

Di:	The internal diameter of the elements of the set. This is used for computing the buoyancy contribution of the internal fluid, if there is any. The default is an internal diameter of 0.
Dd:	The drag diameter. This is the effective outer diameter for hydrodynamic force evaluation using Morison's Equation. The drag diameter defaults to the outer diameter Do.
Db:	The buoyancy diameter. This is the effective outer diameter for buoyancy force calculations for the elements comprising the set. The buoyancy diameter defaults to the outer diameter Do.
Dc:	The effective diameter for contact calculations. This is relevant only when your Flexcom analysis includes guide surfaces or line clashing.

Notes

(a) For a discussion of different formats available to you to define geometric properties, and the program options for specifying non-linear material properties, refer to Geometric Properties for Truss Elements.

(b) Flexcom regards each set of truss elements defined under the *GEOMETRIC SETS keyword as a coherent structure. So it is important that you include a separate entry in the keyword for each discrete part of the model. For example, if you have a semi-submersible platform which is moored using a 3-line catenary system, then you should have 3 separate definitions of truss element properties under the *GEOMETRIC SETS keyword, even if all the material properties for each mooring line are identical. This helps to reduce the possibility of unnatural compression build up in truss elements, which can be an issue for severe environments. Refer to Compression in Truss Elements for further details.

(c) If a non-linear material curve name is specified for EA, then the non-linear material curve must be defined in the `*FORCE-STRAIN` keyword.

(d) Contact diameter is relevant only when your Flexcom analysis includes guide surfaces or line clashing. Flexcom uses this input to determine when contact occurs. Note that this value is used only in contact calculations in the main Analysis module - it is not used by the clearance/interference postprocessing module Clear. Specification of a contact diameter is optional, and Dc defaults to the maximum of Dd, Db and Do if omitted.

(e) There are a number of different element diameters used by Flexcom. In the main Analysis module you can specify outer diameter, drag diameter and buoyancy diameter. In addition to these fundamental model inputs, you can specify separate diameters for use in stress computations during postprocessing, and you can do this in either the main *Analysis* or *Database Postprocessing* modules. So it is conceivable that you could specify the outer diameter for a given element in three different places, and specify three different values if you so wish. Naturally, such a scenario can appear confusing, particularly for new users of the software. Refer to Geometric Properties for Truss Elements for a detailed discussion on the significance of each diameter input, in order to eliminate any possible ambiguity.

1.8.10.27 *GUIDE

PURPOSE

To define guide (contact) surfaces.

THEORY

Refer to Contact Surfaces for further information on this feature.

KEYWORD FORMAT

There are three possible types of guide surface: flat guides, cylindrical guides and zero-gap guides. The keyword formats are slightly different in each case, so each one is described separately.

Flat Guides

A block of lines that defines a flat guide surface, repeated as often as necessary.

The first line defines the guide type:

```
TYPE=FLAT
```

It is possible to associate the guide surface with a vessel or a structural node, or for the guide to remain stationary. The format of the second line is…

NAME=*Guide Name*, VESSEL=*Vessel Name*, SET=*Set Name*

or...

NAME=*Guide Name*, NODE=*Node (Number or Label)*, SET=*Set Name*

or...

NAME=*Guide Name*, SET=*Set Name*

The format of the remaining lines is as follows:

X0, Y0, Z0
X1, Y1, Z1, X2, Y2, Z2
h, w [, *Characteristic Length*] [, *Thickness*]
[μ_{long}] [, μ_{tran}], k

Refer to the relevant table entries for a detailed discussion of each of these parameters. If you specify a node label rather than a number, it must be enclosed in {} brackets. *X0, Y0* and *Z0* are the global X, Y, & Z coordinates of the origin of the contact surface. *X1, Y1* and *Z1* are the global X, Y, & Z components of a vector that describes the local x-axis of the surface. *X2, Y2* and *Z2* are the global X, Y, & Z components of a vector that describes the local y-axis of the surface. h and w are the height and width of the surface, respectively. μ_{long} and μ_{tran} are the longitudinal and transverse friction coefficients, respectively. k is the elastic contact stiffness of the surface.

Cylindrical Guides

A block of lines that defines a cylindrical guide surface, repeated as often as necessary.

The first line defines the guide type and surface contact type. By default, contact is with the external cylindrical surface:

```
TYPE=CYLINDRICAL [, SURFACE=EXTERNAL]
```

For contact with the internal cylindrical curved surface the format is:

```
TYPE=CYLINDRICAL [, SURFACE=INTERNAL]
```

It is possible to associate the guide surface with a vessel or a structural node, or for the guide to remain stationary. The format of the second line is…

NAME=*Guide Name*, VESSEL=*Vessel Name*, SET=*Set Name* or...

```
NAME=Guide Name, NODE=Node (Number or Label), SET=Set Name
```

or...

```
NAME=Guide Name, SET=Set Name
```

The format of the remaining lines is as follows:

```
[AXIS=Axis System]
Xo, Yo, Zo
Xa, Ya, Za
Xr, Yr, Zr
L, R, [Θ], ϕ [, Thickness]
K
[μaxial] [, μcirc] [, Characteristic Length]
```

Refer to the relevant table entries for a detailed discussion of each of these parameters. If you specify a node label rather than a number, it must be enclosed in {} brackets. You must use the *LOCAL AXIS SYSTEM keyword to define any local axis system you reference here. Xo, Yo and Zo are the X, Y, & Z coordinates of the origin of the contact surface. Xa, Ya and Za are the X, Y, & Z components of a vector that describes the local (longitudinal) axis of the cylinder. Xr, Yr and Zr are the X, Y, & Z components of a vector that describes the local radius of the cylinder. L and R are the length and radius of the cylinder, respectively. θ and ϕ are the starting angle and subtended angle of the cylindrical surface, respectively – these entries allow partial cylinders to be modelled. K is the elastic contact stiffness of the surface. μ_{axial} and μ_{circ} are the friction coefficients in the axial and circumferential directions, respectively.

Zero-gap Guides

A block of lines that defines a zero-gap guide surface, repeated as often as necessary.

The first line defines the guide type:

```
TYPE=ZEROGAP
```

For zero-gap guides, it is possible to associate the guide surface with a vessel or for the guide to remain stationary. The format of the second line is…

```
VESSEL=Vessel Name, SET=Set Name
```

or...

```
SET=Set Name
```

The format of the remaining lines is as follows:

```
X0, Y0, Z0
```

```
X1, Y1, Z1
h
[μlong] [, P]
```

Refer to the relevant table entries for a detailed discussion of each of these parameters. *X0, Y0* and *Z0* are the global X, Y, & Z coordinates of the origin of the zero-gap guide. *X1, Y1* and *Z1* are the global X, Y, & Z components of a vector that describes the orientation of the zero-gap guide. h is the height of the zero-gap guide. μ_{long} is the longitudinal friction coefficient. P is the static preload to be applied to the zero-gap guide.

FLAT GUIDES

Vessel Driven Guides

Input:	Description
Guide Name:	The name of the flat guide that will be referenced throughout the analysis.
Set Name:	The name of an element set that will be monitored during the analysis for contact with this guide surface.
Vessel Name:	The name of the vessel whose motions govern the movement of the guide.
X0, Y0, Z0:	Global X, Y, & Z coordinates of the origin of the contact surface at solution initiation. See Note (a).
X1, Y1, Z1:	The global X, Y, & Z components of a vector that describes the local x-axis of the surface at solution initiation. See Note (a).
X2, Y2, Z2:	The global X, Y, & Z components of a vector that describes the local y-axis of the surface at solution initiation. See Note (a).
Height:	The height (measured along the local x-axis) of the contact surface.
Width:	The width (measured along the local y-axis) of the contact surface.

Thickness:	The thickness (measured along the local z-axis) of the contact surface. This entry is used for display purposes only and does not affect the overall operation of the contact modelling algorithm. It is an optional entry and defaults to 0.1m or 0.328ft.
muLong:	The longitudinal friction coefficient associated with this guide surface.
muTran:	The transverse friction coefficient associated with this guide surface.
Characteristic Length:	The length to be used in determining the non-linear stiffness used to simulate frictional restraint. This entry is optional and defaults to 10% of the guide surface *Height* if omitted.
Stiffness:	The elastic contact stiffness of the surface. See Note (e).

Node Driven Guides

Input:	Description
Guide Name:	The name of the flat guide that will be referenced throughout the analysis.
Set Name:	The name of an element set that will be monitored during the analysis for contact with this guide surface.
Node:	The node (number or label) whose motions govern the movement of the guide. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
X0, Y0, Z0:	Global X, Y, & Z coordinates of the origin of the contact surface at solution initiation. See Note (a).
X1, Y1, Z1:	The global X, Y, & Z components of a vector that describes the local x-axis of the surface at solution initiation. See Note (a).

X2, Y2, Z2:	The global X, Y, & Z components of a vector that describes the local y-axis of the surface at solution initiation. See Note (a).
Height:	The height (measured along the local x-axis) of the contact surface.
Width:	The width (measured along the local y-axis) of the contact surface.
muLong:	The longitudinal friction coefficient associated with this guide surface.
muTran:	The transverse friction coefficient associated with this guide surface.
Characteristic Length:	The length to be used in determining the non-linear stiffness used to simulate frictional restraint. This entry is optional and defaults to 10% of the guide surface *Height* if omitted.
Stiffness:	The elastic contact stiffness of the surface. See Note (e).

Stationary Guides

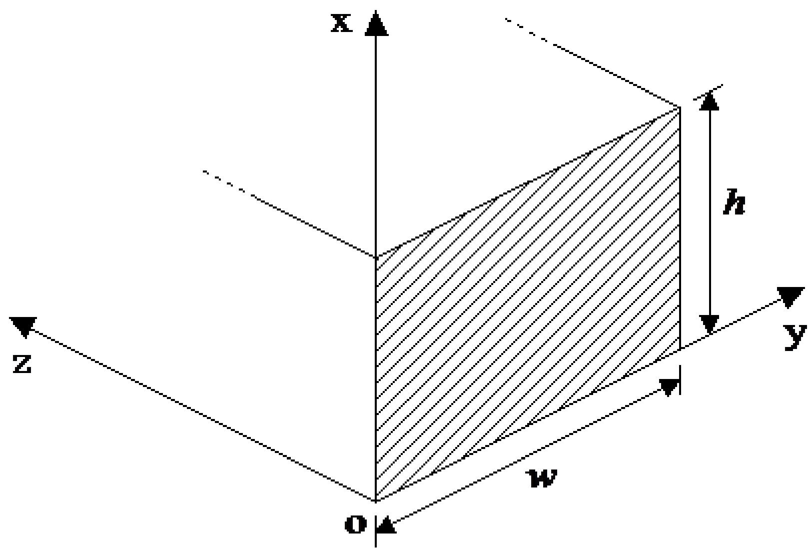
Input:	Description
Guide Name:	The name of the flat guide that will be referenced throughout the analysis.
Set Name:	The name of an element set that will be monitored during the analysis for contact with this guide surface.
X0, Y0, Z0:	Global X, Y, & Z coordinates of the origin of the contact surface at solution initiation. See Note (a).
X1, Y1, Z1:	The global X, Y, & Z components of a vector that describes the local x-axis of the surface at solution initiation. See Note (a).
X2, Y2, Z2:	The global X, Y, & Z components of a vector that describes the local y-axis of the surface at solution initiation. See Note (a).
Height:	The height (measured along the local x-axis) of the contact surface.

Width:	The width (measured along the local y-axis) of the contact surface.
Thickness:	The thickness (measured along the local z-axis) of the contact surface. This entry is used for display purposes only and does not affect the overall operation of the contact modelling algorithm. It is an optional entry and defaults to 0.1m or 0.328ft.
muLong:	The longitudinal friction coefficient associated with this guide surface.
muTran:	The transverse friction coefficient associated with this guide surface.
Characteristic Length:	The length to be used in determining the non-linear stiffness used to simulate frictional restraint. This entry is optional and defaults to 10% of the guide surface *Height* if omitted.
Stiffness:	The elastic contact stiffness of the surface. See Note (e).

Notes

(a) Refer to Flat Guide Surfaces for a detailed discussion of contact modelling with flat guide surfaces.

(b) These tables are used to define the initial location and orientation of rectangular contact surfaces. Each contact surface has a local axis system associated with it, as shown below. The initial location and orientation of the contact surface is defined by specifying the initial coordinates of the origin of this axis system along with the local x- and y-axes.



The origin of the contact surface is defined as the lower left-hand corner of the surface when the surface is viewed from the side that can come into contact with the structure. To define the local x- and y-axes, you must specify the global X, Y & Z components of two vectors, the first of which is aligned with the local x-axis and the second of which is aligned with the local y-axis. The length of these vectors is not significant – it is their orientation that is important. Naturally, there must be a 90° angle between the local x- and y-axes or the program will generate an error. Once the local x- and y-axes are specified, the program automatically finds the local-z axis using the right-hand rule. Note that the orientation of the local-z axis is significant – it must point away from the side of the surface that can come into contact with the structure

(c) If the contact surface has been associated with a vessel or node, then the initial location and orientation of the contact surface correspond to the initial position of that vessel or node. Any subsequent motions from the initial position will cause a corresponding movement of the contact surface.

(d) The longitudinal and transverse friction coefficients for the surface apply to contact with all elements in the contact element set associated with the surface. The *Characteristic Length* input relates to the operation of the guide surface friction model, and allows user control over the characteristics of non-linear springs used to model frictional restraint. For more information, refer to Guide Surface Friction Modelling.

(e) The flat guide is assumed to be a linear elastic surface. The contact stiffness should be specified in units of [Force] / [Distance] (for example N/m in metric or lb/ft in Imperial).

CYLINDRICAL GUIDES

Vessel Driven Guides

Input:	Description
Surface Contact:	The cylindrical surface with which contact is to be made. This is either external (the default) or internal.
Guide Name:	The name of the cylindrical guide that will be referenced throughout the analysis.
Set Name:	The name of an element set that will be monitored during the analysis for contact with this guide surface.
Vessel Name:	The name of the vessel whose motions govern the movement of the guide surface.
Axis:	The name of a local axis system in which the contact surface geometry is defined. This entry is optional and the global axis system is used by default. See Note (b).
Xo, Yo, Zo:	The X, Y, & Z coordinates of the origin of the contact surface at solution initiation. The origin is located at the centre of curvature, at the first end of the cylinder. See Note (c).
Xa, Ya, Za:	The X, Y, & Z components of a vector that describes the local (longitudinal) axis of the cylinder at solution initiation. See Note (c).
Xr, Yr, Zr:	The X, Y, & Z components of a vector that describes the local radius of the cylinder at solution initiation. This vector must be orthogonal to the axial vector. See Note (c).
Length:	The length of the cylinder, measured along the axial vector. See Note (c).
Radius:	The radius of the cylinder, measured along the radial vector. See Note (c).

Start Angle:	The starting angle of the cylindrical surface, measured in a clockwise direction with respect to the radius vector, in a plane which is perpendicular to the axial vector. The *Start Angle* and *Subtended Angle* entries allow partial cylinders to be modelled. *Start Angle* is an optional entry, and defaults to half the *Subtended Angle* times minus one. See Notes (c) and (d).
Subtended Angle:	The subtended angle of the cylindrical surface, measured in a clockwise direction from the *Start Angle*, in a plane which is perpendicular to the axial vector. See Notes (c) and (d).
Thickness:	The thickness of the contact surface. This entry is used for display purposes only and does not affect the overall operation of the contact modelling algorithm. It is an optional entry and defaults to 0.1m or 0.328ft. For external contact it is measured from the external surface in. For internal contact it is measured from the internal surface out.
Stiffness:	The elastic contact stiffness of the surface. See Note (f).
muAxial:	The friction coefficient in the axial direction (i.e. parallel to the axial vector).
muCirc:	The friction coefficient in the circumferential direction (i.e. parallel to the surface tangent vector).
Characteristic Length:	The length to be used in determining the non-linear stiffness used to simulate frictional restraint. This entry is optional and defaults to 10% of the cylinder *Length* if omitted. See Note (g).

Node Driven Guides

Input:	Description
Surface Contact:	The cylindrical surface with which contact is to be made. This is either external (the default) or internal.

Guide Name:	The name of the cylindrical guide that will be referenced throughout the analysis.
Set Name:	The name of an element set that will be monitored during the analysis for contact with this guide surface.
Node:	The node (number or label) whose motions govern the movement of the guide. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Axis:	The name of a local axis system in which the contact surface geometry is defined. This entry is optional and the global axis system is used by default. See Note (b).
Xo, Yo, Zo:	The X, Y, & Z coordinates of the origin of the contact surface at solution initiation. The origin is located at the centre of curvature, at the first end of the cylinder. See Note (c).
Xa, Ya, Za:	The X, Y, & Z components of a vector that describes the local (longitudinal) axis of the cylinder at solution initiation. See Note (c).
Xr, Yr, Zr:	The X, Y, & Z components of a vector that describes the local radius of the cylinder at solution initiation. This vector must be orthogonal to the axial vector. See Note (c).
Length:	The length of the cylinder, measured along the axial vector. See Note (c).
Radius:	The radius of the cylinder, measured along the radial vector. See Note (c).
Start Angle:	The starting angle of the cylindrical surface, measured in a clockwise direction with respect to the radius vector, in a plane which is perpendicular to the axial vector. The *Start Angle* and *Subtended Angle* entries allow partial cylinders to be modelled. *Start Angle* is an optional entry, and defaults to half the *Subtended Angle* times minus one. See Notes (c) and (d).

Subtended Angle:	The subtended angle of the cylindrical surface, measured in a clockwise direction from the *Start Angle*, in a plane which is perpendicular to the axial vector. See Notes (c) and (d).
Thickness:	The thickness of the contact surface. This entry is used for display purposes only and does not affect the overall operation of the contact modelling algorithm. It is an optional entry and defaults to 0.1m or 0.328ft. For external contact it is measured from the external surface in. For internal contact it is measured from the internal surface out.
Stiffness:	The elastic contact stiffness of the surface. See Note (f).
muAxial:	The friction coefficient in the axial direction (i.e. parallel to the axial vector).
muCirc:	The friction coefficient in the circumferential direction (i.e. parallel to the surface tangent vector).
Characteristic Length:	The length to be used in determining the non-linear stiffness used to simulate frictional restraint. This entry is optional and defaults to 10% of the cylinder *Length* if omitted. See Note (g).

Stationary Guides

Input:	Description
Surface Contact:	The cylindrical surface with which contact is to be made. This is either external (the default) or internal.
Guide Name:	The name of the cylindrical guide that will be referenced throughout the analysis.
Set Name:	The name of an element set that will be monitored during the analysis for contact with this guide surface.

Axis:	The name of a local axis system in which the contact surface geometry is defined. This entry is optional and the global axis system is used by default. See Note (b).
Xo, Yo, Zo:	The X, Y, & Z coordinates of the origin of the contact surface at solution initiation. The origin is located at the centre of curvature, at the first end of the cylinder. See Note (c).
Xa, Ya, Za:	The X, Y, & Z components of a vector that describes the local (longitudinal) axis of the cylinder at solution initiation. See Note (c).
Xr, Yr, Zr:	The X, Y, & Z components of a vector that describes the local radius of the cylinder at solution initiation. This vector must be orthogonal to the axial vector. See Note (c).
Length:	The length of the cylinder, measured along the axial vector. See Note (c).
Radius:	The radius of the cylinder, measured along the radial vector. See Note (c).
Start Angle:	The starting angle of the cylindrical surface, measured in a clockwise direction with respect to the radius vector, in a plane which is perpendicular to the axial vector. The *Start Angle* and *Subtended Angle* entries allow partial cylinders to be modelled. *Start Angle* is an optional entry, and defaults to half the *Subtended Angle* times minus one. See Notes (c) and (d).
Subtended Angle:	The subtended angle of the cylindrical surface, measured in a clockwise direction from the *Start Angle*, in a plane which is perpendicular to the axial vector. See Notes (c) and (d).

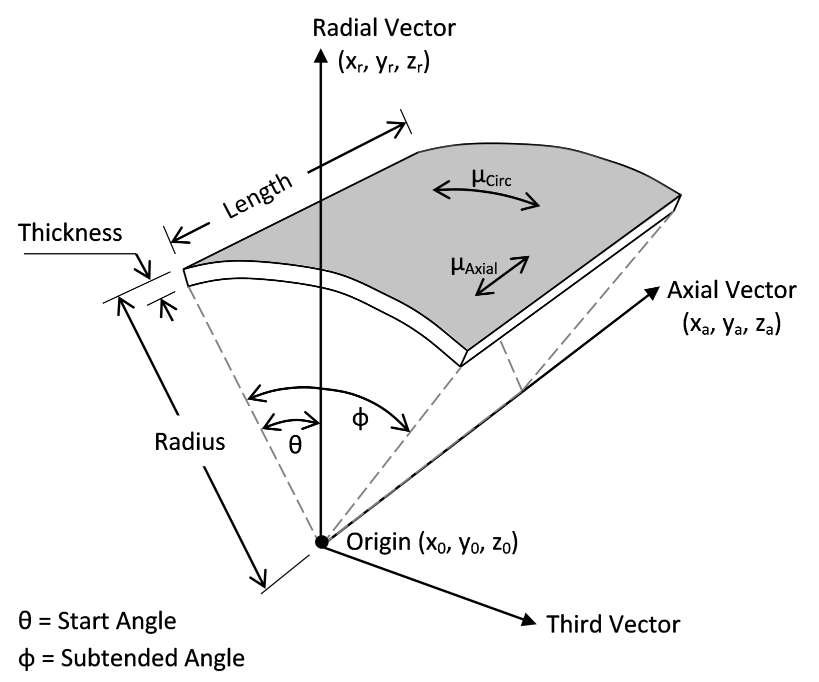
Thickness:	The thickness of the contact surface. This entry is used for display purposes only and does not affect the overall operation of the contact modelling algorithm. It is an optional entry and defaults to 0.1m or 0.328ft. For external contact it is measured from the external surface in. For internal contact it is measured from the internal surface out.
Stiffness:	The elastic contact stiffness of the surface. See Note (f).
muAxial:	The friction coefficient in the axial direction (i.e. parallel to the axial vector).
muCirc:	The friction coefficient in the circumferential direction (i.e. parallel to the surface tangent vector).
Characteristic Length:	The length to be used in determining the non-linear stiffness used to simulate frictional restraint. This entry is optional and defaults to 10% of the cylinder *Length* if omitted. See Note (g).

Notes

(a) Refer to Cylindrical Guide Surfaces for a detailed discussion of contact modelling with cylindrical guide surfaces.

(b) If a local axis system referenced, it must be defined using the `*LOCAL AXIS SYSTEM` keyword.

(c) The overall layout of the cylindrical guide surface is illustrated by the following schematic.



Each cylindrical surface has a local axis system associated with it, as shown above. The axial and radial vectors form two components of a right handed system. The length of these vectors is not significant – it is their orientation that is important. Naturally these two vectors should be orthogonal. A third vector is assembled internally using the right-hand rule.

(d) The sign convention for both the *Start Angle* and the *Subtended Angle* is consistent with the right handed system. Specifically, positive angles represent rotations of the radial vector toward the third vector. If a *Start Angle* is not specified, its magnitude is half that of the *Subtended Angle*, and it differs in sign. In this case, the radial vector bisects the *Subtended Angle*, and represents an axis of symmetry for the cylindrical surface.

(e) If the contact surface has been associated with a vessel or node, then the initial location and orientation of the contact surface correspond to the initial position of that vessel or node. Any subsequent motions from the initial position will cause a corresponding movement of the contact surface.

(f) The cylindrical guide is assumed to be a linear elastic surface. The contact stiffness should be specified in units of [Force] / [Distance] (for example N/m in metric or lb/ft in Imperial).

(g) The axial and circumferential friction coefficients for the surface apply to contact with all elements in the contact element set associated with the surface. The *Characteristic Length* input relates to the operation of the guide surface friction model, and allows user control over the characteristics of non-linear springs used to model frictional restraint. For more information, refer to Guide Surface Friction Modelling.

ZERO-GAP GUIDES

Vessel Driven Guides

Input:	Description
Set Name:	The name of an element set that will be monitored during the analysis for contact with this zero-gap guide.
Vessel Name:	The name of the vessel with which the zero-gap guide is associated. The guide translates and rotates with the vessel during the analysis.
X0, Y0, Z0:	Global X, Y, & Z coordinates of the origin of the zero gap guide at solution initiation. See Note (b).
X1, Y1, Z1:	The global X, Y, & Z components of a vector that defines the orientation of the zero-gap guide at solution initiation. See Note (b).
Height:	The height of the zero-gap guide.
muLong:	The longitudinal friction coefficient associated with this guide surface.
Preload:	The static preload to be applied to the zero-gap guide. See Note (e).

Stationary Guides

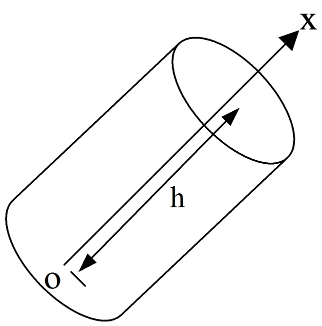
Input:	Description
Set Name:	The name of an element set that will be monitored during the analysis for contact with this zero-gap guide.

X0, Y0, Z0:	Global X, Y, & Z coordinates of the origin of the zero gap guide at solution initiation. See Note (b).
X1, Y1, Z1:	The global X, Y, & Z components of a vector that defines the orientation of the zero-gap guide at solution initiation. See Note (b).
Height:	The height of the zero-gap guide.
muLong:	The longitudinal friction coefficient associated with this guide surface.
Preload:	The static preload to be applied to the zero-gap guide. See Note (e).

Notes

(a) Refer to Zero-Gap Guide Surfaces for a detailed discussion of contact modelling with zero-gap guide surfaces.

(b) The initial location and orientation of the guide is defined by specifying the initial coordinates of the origin of the guide along with an orientation vector, as shown below.



The origin of the zero-gap guide is defined as the centre point of one end (usually bottom) of the guide. To define the orientation vector, you must specify its global (X, Y & Z) components. The length of this vector is not significant – it's the orientation which is important

(c) If the zero-gap guide has been associated with a vessel, then the initial location and orientation of the zero-gap guide correspond to the initial position of that vessel. Any movement of the vessel from its initial position (including movement due to offset, drift, or vessel RAOs) will cause a corresponding movement of the zero-gap guide.

(d) The longitudinal friction coefficient for the zero-gap guide apply to contact with all elements in the contact element set associated with the zero-gap guide.

(e) A static preload may be optionally applied to a zero-gap guide. If a preload is specified, the limiting frictional force, F_{lim}, in the longitudinal direction at a given node in contact with the zero-gap guide is defined as:

$$F_{\lim} = \mu(R_n + P_n)$$

where μ is the longitudinal friction coefficient, R_n is the magnitude of the reaction force at the constrained node, and P_n is the preload associated with the contact node. The user specified preload, P, for the guide is distributed evenly between all nodes which are in contact with the guide at any given time, such that:

$$P_n = \frac{P}{N}$$

where N is the number of nodes in contact with the guide. If no static preload is specified, the limiting frictional force is computed in the usual way as:

$$F_{\lim} = \mu R_n$$

1.8.10.28 *HINGE

PURPOSE

To specify hinge elements in the structural discretisation.

THEORY

Refer to Hinge and Flex Joints for further information on this feature.

KEYWORD FORMAT

One type of line which can be repeated as often as necessary.

Line defining a hinge:

```
Hinge Element (Number or Label), [STIFFNESS=Rotational Stiffness]
```

If you specify an element label rather than an element number, it must be enclosed in {} brackets. `Rotational Stiffness` defaults to zero.

TABLE INPUT

Input:	Description
Element:	The hinge element (number or label). If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Stiffness:	The rotational stiffness of the hinge. This is an optional input which defaults to zero.

NOTES

(a) It is strongly recommended that the two nodes of the hinge element be coincident for unambiguous results.

1.8.10.29 *HYDRODYNAMIC COUPLING

PURPOSE

To define hydrodynamic coupling between adjacent floating bodies.

THEORY

Refer to Hydrodynamic Coupling Coefficients for further information on this feature.

KEYWORD OVERVIEW

Data may defined explicitly within the keyword file itself, or defined separately in an external file which is simply referenced using a FILE= entry. The latter approach is highly recommended as it reduces unnecessary clutter in the keyword file, and also helps to improve user interface performance and enhanced user experience.

The first line specifies the pair of floating bodies which are to be coupled. This is followed by a COUPLING= line defining the type of data to follow, either added mass or radiation damping. This is followed by a TYPE=line defining the format of the coupling, either frequency independent or frequency dependent. For frequency independent coupling, a pair of 6x6 matrices are defined by a single block preceded by a TYPE=CONSTANT line. For frequency dependent coupling, multiple pairs of 6x6 matrices are defined by several blocks of data, each one preceded by a FREQ= line for the different frequencies, with a TYPE=FREQUENCY line at the beginning. This block is repeated as often as necessary to define the coupling over a range of frequencies. Additionally, an optional final block of data defining the coupling at the cut-off frequency, preceded by TYPE=CUTOFF, may be included. The entire block of data can then be repeated to specify hydrodynamic coupling for second and subsequent floating body couples.

DATA SPECIFIED IN EXTERNAL FILE

Keyword Format

Line defining the coupled pair of floating bodies:

```
FLOATING BODY 1=Floating Body Name, FLOATING BODY 2=Floating Body Name
```

Line defining name of external file which contains hydrodynamic coupling data.

```
FILE=File Name
```

File Name should include the entire path of the file including its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

Table Input

Input:	Description
Floating Body 1:	The name of the first floating body of the coupled pair.
Floating Body 2:	The name of the second floating body of the coupled pair.

File Name:	The name of the external data file. See Note (a).

Notes

(a) Refer to the following sections for further information regarding the required format of data within the external file.

CONSTANT HYDRODYNAMIC COUPLING

Purpose

To define constant (frequency-independent) hydrodynamic coupling for a floating body couple. The data may relate to *Added Mass* or *Radiation Damping*.

Keyword Format

Line defining the coupled pair of floating bodies:

FLOATING BODY 1=*Floating Body Name,* FLOATING BODY 2=*Floating Body Name*

Line defining the type of coupling:

COUPLING=ADDED MASS/RADIATION DAMPING

Block of lines defining frequency independent coupling:

TYPE=CONSTANT

$$\begin{bmatrix} A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16} \\ A_{21}, A_{22}, A_{23}, A_{24}, A_{25}, A_{26} \\ A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36} \\ A_{41}, A_{42}, A_{43}, A_{44}, A_{45}, A_{46} \\ A_{51}, A_{52}, A_{53}, A_{54}, A_{55}, A_{56} \\ A_{61}, A_{62}, A_{63}, A_{64}, A_{65}, A_{66} \end{bmatrix}_{1\to 2}$$

.

$$\begin{bmatrix} A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16} \\ A_{21}, A_{22}, A_{23}, A_{24}, A_{25}, A_{26} \\ A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36} \\ A_{41}, A_{42}, A_{43}, A_{44}, A_{45}, A_{46} \\ A_{51}, A_{52}, A_{53}, A_{54}, A_{55}, A_{56} \\ A_{61}, A_{62}, A_{63}, A_{64}, A_{65}, A_{66} \end{bmatrix}_{2\to1}$$

.

Table Input

Input:	Description
Floating Body 1:	The name of the first floating body of the coupled pair.
Floating Body 2:	The name of the second floating body of the coupled pair.
Matrices (6x6):	Two 6x6 matrices of hydrodynamic coupling data. See Note (a).

Notes

(a) Refer to Hydrodynamic Coupling Coefficients for further information on the layout of the hydrodynamic coupling terms.

(b) The type of added mass coupling specified for a particular floating body couple, either *Constant* or *Frequency Dependent*, should be consistent with the type of radiation damping coupling specified for the same floating body couple.

HYDRODYNAMIC COUPLING - REFERENCE MATRICES

Purpose

To reference frequency dependent and cut-off hydrodynamic coupling matrices for use with adjacent floating bodies.

Keyword Format

Line defining the coupled pair of floating bodies:

```
FLOATING BODY 1=Floating Body Name, FLOATING BODY 2=Floating Body Name
```

Line defining the type of coupling:

```
COUPLING=ADDED MASS/RADIATION DAMPING
```

Table Input

Input:	Description
Floating Body 1:	The name of the first floating body of the coupled pair.
Floating Body 2:	The name of the second floating body of the coupled pair.
Type:	This option allows you to specify that the hydrodynamic coupling data relates to *Added Mass* (the default) or *Radiation Damping*.
Frequency Dependent Hydrodynamic Coupling:	The name of the constant or frequency-dependent hydrodynamic coupling matrix definition.
Cut-Off Hydrodynamic Coupling:	The name of the cut-off hydrodynamic coupling matrix definition.

Notes

(a) Any properties which you refer to for a particular floating body couple must be defined in other tables – i.e. *Body - Frequency Dependent Hydrodynamic Coupling* and *Body – Cut-Off Hydrodynamic Coupling*.

FREQUENCY DEPENDENT HYDRODYNAMIC COUPLING

Purpose

To define frequency-dependent hydrodynamic coupling for with a floating body couple. The data may relate to *Added Mass* or *Radiation Damping*.

Keyword Format

Block of lines defining frequency dependent coupling, preceded by a single TYPE=FREQUENCY line:

```
TYPE=FREQUENCY
FREQ=Frequency
```

$$\begin{bmatrix} A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16} \\ A_{21}, A_{22}, A_{23}, A_{24}, A_{25}, A_{26} \\ A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36} \\ A_{41}, A_{42}, A_{43}, A_{44}, A_{45}, A_{46} \\ A_{51}, A_{52}, A_{53}, A_{54}, A_{55}, A_{56} \\ A_{61}, A_{62}, A_{63}, A_{64}, A_{65}, A_{66} \end{bmatrix}_{1\to 2}$$

$$\begin{bmatrix} A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16} \\ A_{21}, A_{22}, A_{23}, A_{24}, A_{25}, A_{26} \\ A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36} \\ A_{41}, A_{42}, A_{43}, A_{44}, A_{45}, A_{46} \\ A_{51}, A_{52}, A_{53}, A_{54}, A_{55}, A_{56} \\ A_{61}, A_{62}, A_{63}, A_{64}, A_{65}, A_{66} \end{bmatrix}_{2\to 1}$$

Table Input

Input:	Description
Name:	The name of the hydrodynamic coupling matrices.
Frequency:	The frequency in Hertz to which the 6x6 hydrodynamic coupling matrices relate.
Matrices (6x6):	Two 6x6 matrices of hydrodynamic coupling data. See Note (a).

Notes

(a) Refer to Hydrodynamic Coupling Coefficients for further information on the layout of the hydrodynamic coupling terms.

(b) This table is used in conjunction with the *Body - Cut-Off Hydrodynamic Coupling* table, in which hydrodynamic coupling at the cut-off frequency is specified. Hydrodynamic coupling at the cut-off frequency is completely defined by two 6x6 matrices. The cut-off frequency itself is specified in the *Body – Frequency* table. You should ensure that both floating bodies in a couple are assigned the same cut-off frequency.

(c) For frequencies below the cut-off frequency, the hydrodynamic coupling corresponds to the hydrodynamic coupling at the cut-off frequency. If no cut-off frequency is specified, the hydrodynamic coupling is assumed to be zero below the range of user-specified frequencies.

(d) For frequencies above the range of user-specified frequencies, the hydrodynamic coupling is assumed to be zero.

(e) For frequencies that do not exactly match one of the frequencies specified, Flexcom linearly interpolates between the nearest frequencies to find the relevant values of hydrodynamic coupling.

(f) You do not need to specify the frequencies in any particular order; ascending order is not necessary.

(g) The type of added mass coupling specified for a particular floating body couple, either *Constant* or *Frequency Dependent*, should be consistent with the type of radiation damping coupling specified for the same floating body couple.

CUT-OFF HYDRODYNAMIC COUPLING

Purpose

To define cut-off hydrodynamic coupling for a floating body couple. The data may relate to *Added Mass* or *Radiation Damping*.

Keyword Format

Block of lines defining coupling at the cut-off frequency:

```
TYPE=CUTOFF
```

$$\begin{bmatrix} A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16} \\ A_{21}, A_{22}, A_{23}, A_{24}, A_{25}, A_{26} \\ A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36} \\ A_{41}, A_{42}, A_{43}, A_{44}, A_{45}, A_{46} \\ A_{51}, A_{52}, A_{53}, A_{54}, A_{55}, A_{56} \\ A_{61}, A_{62}, A_{63}, A_{64}, A_{65}, A_{66} \end{bmatrix}_{1\to2}$$

$$\begin{bmatrix} A_{11}, A_{12}, A_{13}, A_{14}, A_{15}, A_{16} \\ A_{21}, A_{22}, A_{23}, A_{24}, A_{25}, A_{26} \\ A_{31}, A_{32}, A_{33}, A_{34}, A_{35}, A_{36} \\ A_{41}, A_{42}, A_{43}, A_{44}, A_{45}, A_{46} \\ A_{51}, A_{52}, A_{53}, A_{54}, A_{55}, A_{56} \\ A_{61}, A_{62}, A_{63}, A_{64}, A_{65}, A_{66} \end{bmatrix}_{2\to1}$$

Table Input

Input:	Description
Name:	The name of the hydrodynamic coupling matrices.
Matrices (6x6):	Two 6x6 matrices of hydrodynamic coupling data. See Note (a).

Notes

(a) Refer to Hydrodynamic Coupling Coefficients for further information on the layout of the hydrodynamic coupling terms.

(b) This table is used in conjunction with the *Body - Frequency Dependent Hydrodynamic Coupling* table, in which frequency-dependent hydrodynamic coupling is specified. The cut-off frequency itself is specified in the *Body – Frequency* table. You should ensure that both floating bodies in a couple are assigned the same cut-off frequency.

(c) For frequencies below the cut-off frequency, the hydrodynamic coupling corresponds to the hydrodynamic coupling at the cut-off frequency. If no cut-off frequency is specified, the added mass is assumed to be zero below the range of user-specified frequencies.

1.8.10.30 *HYDRODYNAMIC SETS

PURPOSE

To assign hydrodynamic coefficients to element sets.

THEORY

Refer to Hydrodynamic Loading for further information on this feature.

KEYWORD OVERVIEW

Hydrodynamic coefficients can be specified as being (i) constant throughout an analysis, or (ii) can vary as a function of Reynolds number Re, or (iii) can be defined for sets which may be contained in a moonpool. The actual coefficients are specified on two lines (for constant or moonpool values), on three or more lines (for values varying with Re), or four lines (for buoy structure properties). Blocks of lines may be repeated to allow a combination of specifications. If coefficients vary as a function of Reynolds number, an optional line may be included at the beginning of the keyword to indicate that the hydrodynamic forces are to be computed as a function of the instantaneous Reynolds number.

MORISON'S EQUATION COMPUTATION

Purpose

To specify whether element drag (used by default) or buoyancy diameters should be used during the computation of the added mass and inertia terms in Morison's Equation.

Keyword Format

An additional DIAMETER= line at the beginning of the keyword.

```
[DIAMETER=Diameter]
```

Diameter may be either DRAG (the default) or BUOYANCY.

Table Input

Input:	Description
Diameter:	The options are *Drag* (the default) and *Buoyancy*.

Notes

(a) The program hydrodynamic force formulation is outlined in Hydrodynamic Loading. Traditionally, Flexcom has based the calculation of added mass and inertia loading on drag diameter, and this behaviour is retained as the default in order to maintain compatibility with earlier versions. However, strictly speaking, the added mass and inertia terms should be based on displaced volume (i.e. buoyancy diameter) as opposed to projected area (i.e. drag diameter), and the *Buoyancy Diameter* option facilitates this alternative approach.

CONSTANT HYDRODYNAMIC COEFFICIENTS

Purpose

To assign hydrodynamic coefficients to element sets where these coefficients are independent of Reynolds number.

Keyword Format

The two lines that make up the block for constant hydrodynamic properties, specifying either no drag lift or a linear drag lift coefficient:

```
SET=Set Name [, TYPE=CONSTANT]
Cd Normal [, Cd Tangential] [, Cm Normal] [, Ca Tangential] [, Ca Normal] [,
```

`Drag Lift` is specified for linear drag lift, whereas a `Curve Name` is specified for non-linear drag lift. If a curve name is specified, then it must be defined using *DRAG LIFT.

Table Input

Input:	Description
Set Name:	The element set to which the hydrodynamic coefficients are to be assigned. This defaults to all elements.

Normal Drag:	The drag coefficient for the direction normal to the section, denoted C^n_d.
Tangential Drag:	The drag coefficient for the direction tangential to the section, denoted C^t_d. This entry is optional, and if omitted defaults to zero.
Normal Inertia:	The inertia coefficient in the direction normal to the section, denoted C^n_m. This entry is optional, and if omitted defaults to 2.0. See Note (b).
Tangential Added Mass:	The added mass coefficient in the direction tangential to the section, denoted C^t_a. This entry is optional, and if omitted defaults to zero.
Normal Added Mass:	The added mass coefficient in the direction normal to the section, denoted C^n_a. This entry is optional, and if omitted defaults to (C^n_m-1). See Note (c).
Drag Lift:	The drag lift coefficient for the elements of the set or the name of a non-linear drag lift curve. This input is optional. See Note (d). If you specify the name of a non-linear curve here, you must define the non-linear curve using the *Non-linear Drag Lift* table

Notes

(a) Refer to Constant Hydrodynamic Coefficients for further information on this feature.

REYNOLDS DEPENDENT HYDRODYNAMIC COEFFICIENTS

Purpose

To assign hydrodynamic coefficients to element sets where these coefficients are a function of Reynolds number (Re).

Keyword Format

The two lines that make up the block for coefficients which vary with Reynolds number:

```
SET=Set Name, TYPE=REYNOLDS
```

```
Re, Cd Normal [, Cd Tangential] [, Cm Normal] [, Ca Tangential] [, Ca Normal]
```

The second line can be repeated as many times as necessary to define properties for different Reynolds numbers.

Table Input

Input:	Description
Set Name:	The element set to which the hydrodynamic coefficients are to be assigned. This defaults to all elements.
Reynolds No:	The value of Reynolds number at which coefficients are being defined
Normal Drag:	The drag coefficient for the direction normal to the section, denoted C^n_d.
Tangential Drag:	The drag coefficient for the direction tangential to the section, denoted C^t_d. This entry is optional, and if omitted defaults to zero.
Normal Inertia:	The inertia coefficient in the direction normal to the section, denoted C^n_m. This entry is optional, and if omitted defaults to 2.0.
Tangential Added Mass:	The added mass coefficient in the direction tangential to the section, denoted C^t_a. This entry is optional, and if omitted defaults to zero.
Normal Added Mass:	The added mass coefficient in the direction normal to the section, denoted C^n_a. This entry is optional, and if omitted defaults to (C^n_m-1).

Notes

(a) Refer to Reynolds Number Dependent Coefficients for further information on this feature.

(b) Use as many lines as you need to completely define the hydrodynamic coefficients for a particular element set. Simply leave Column 1 blank for second and subsequent lines. For subsequent sets, put the set name in Column 1 and specify the coefficients in the same way.

(c) For values of Re intermediate to the values you specify, the hydrodynamic coefficients are calculated by linear interpolation.

REYNOLDS NUMBER COMPUTATION

Purpose

To specify how the dependence of hydrodynamic coefficients on Reynolds number is to be calculated.

Keyword Format

The presence of the OPTION=INSTANTANEOUS line at the beginning of the keyword indicates that the hydrodynamic forces are to be computed as a function of the instantaneous Reynold’s number. If this line is omitted, the normal approach is adopted whereby Reynold’s number remains constant over each wave period.

```
[OPTION=INSTANTANEOUS]
```

Table Input

Input:	Description
Reynolds Number Computation Type:	The options are *Constant* (the default) and *Instantaneous*.

Notes

(a) Refer to Reynolds Number Dependent Coefficients for further information on this feature.

MOONPOOL HYDRODYNAMIC COEFFICIENTS

Purpose

To assign hydrodynamic coefficients to elements of the structure that may be subjected to hydrodynamic loading within a vessel moonpool.

Keyword Format

The two lines that make up the block for moonpool hydrodynamic properties:

```
SET=Set Name, TYPE=MOONPOOL
Cd Normal [, Cd Tangential] [, Cm Normal] [, Ca Tangential] [, Ca Normal]
```

Table Input

Input:	Description
Set Name:	The element set to which the hydrodynamic coefficients are to be assigned. This defaults to all elements.
Normal Drag:	The drag coefficient for the direction normal to the section, denoted $C^n{}_d$.
Tangential Drag:	The drag coefficient for the direction tangential to the section, denoted $C^t{}_d$. This entry is optional, and if omitted defaults to zero.
Normal Inertia:	The inertia coefficient in the direction normal to the section, denoted $C^n{}_m$. This entry is optional, and if omitted defaults to 2.0.
Tangential Added Mass:	The added mass coefficient in the direction tangential to the section, denoted $C^t{}_a$. This entry is optional, and if omitted defaults to zero.
Normal Added Mass:	The added mass coefficient in the direction normal to the section, denoted $C^n{}_a$. This entry is optional, and if omitted defaults to ($C^n{}_m$-1).

Notes

(a) Refer to Moonpool Hydrodynamics for further information on this feature.

(b) Only constant hydrodynamic coefficients may be specified for moonpool hydrodynamic sets, i.e. hydrodynamic coefficients may not be specified as a function of Reynolds number.

BUOY STRUCTURE HYDRODYNAMIC COEFFICIENTS

Purpose

To assign hydrodynamic properties to a set of elements that collectively model a subsea buoy.

Keyword Format

The four lines that make up the block for buoy structure hydrodynamic properties:

```
SET=Set Name, TYPE=BUOY [, AXIS=Axis Name]
X:CdAd, X:CmVin, X:CaVin
Y:CdAd, Y:CmVin, Y:CaVin
Z:CdAd, Z:CmVin, Z:CaVin
```

Axis Name can be any arbitrary local axis system defined under *LOCAL AXIS SYSTEM. If the axis system for the buoy is initially aligned with the global axis system, you can simply specify AXIS=GLOBAL and a local buoy axis system which is aligned with the global axis system will be created internally for you.

Table Input

Input:	Description
Set Name:	The element set defining the buoy structure to which the hydrodynamic properties are to be assigned.
Axis:	The name of a local axis system to be used in the computation of hydrodynamic loading on the buoy. This entry is optional and the global axis system is used by default. See Note (c).
X: CdAd:	The product of the buoy frontal area and drag coefficient in the X direction.
X: CmVin:	The product of the buoy reference volume and inertia coefficient in the X direction.

X: CaVin:	The product of the buoy reference volume and added mass coefficient in the X direction.
Y: CdAd:	The product of the buoy frontal area and drag coefficient in the Y direction.
Y: CmVin:	The product of the buoy reference volume and inertia coefficient in the Y direction.
Y: CaVin:	The product of the buoy reference volume and added mass coefficient in the Y direction.
Z: CdAd:	The product of the buoy frontal area and drag coefficient in the Z direction.
Z: CmVin:	The product of the buoy reference volume and inertia coefficient in the Z direction.
Z: CaVin:	The product of the buoy reference volume and added mass coefficient in the Z direction

Notes

(a) Refer to Subsea Buoys for further information on this feature.

(b) Note that all of the hydrodynamic properties for the elements of the subsea buoy set are specified here. It is incorrect to assign hydrodynamic properties to elements using *Hydrodynamic – Constant* or *Hydrodynamic – Reynolds No.*, and then to also assign such properties using this *Hydrodynamic – Buoy Structure* facility.

(c) The Axis input gives you a choice of methods for specifying the *Hydrodynamic – Buoy Structure* coefficients. These can be specified in the Global axes or in an axis system which is Local to the buoy. If a local axis system referenced, it must be defined using the `*LOCAL AXIS SYSTEM` keyword. Coefficients specified in either axis system account for changes in the orientation of the buoy as it displaces and rotates in space.

1.8.10.31 *LABEL

PURPOSE

To associate a descriptive name with a node or element of the finite element model.

THEORY

Refer to Node and Element Labels for further information on this feature.

KEYWORD FORMAT

A block of lines which defines a label, repeated as often as necessary.

```
TYPE=Type
Number, LABEL=Label
```

Notes: *Type* may be either NODE or ELEMENT. The second line may be repeated to define the second and successive labels of a particular type. The first line may be repeated to switch between node and element label definitions.

TABLE INPUT

Node Labels

Input:	Description
Node:	The number of the node for which the label is being defined.
Label:	The corresponding node label.

Element Labels

Input:	Description
Element:	The number of the element for which the label is being defined.
Label:	The corresponding element label.

Notes

(a) You can refer to labels in other areas of the program. For example, you can refer to node labels when you are assigning boundary conditions, or defining nodal equivalences, rather than referencing explicit node numbers, in the normal fashion.

(b) It is important to define meaningful names for ease of reference. Care should be taken also to ensure that the location names are unique, in order to avoid any possible ambiguity regarding the label definitions.

1.8.10.32 *LINE LOCATIONS

PURPOSE

To define named locations along a line, allowing you to request that nodes be positioned exactly at specific lengths along a line.

THEORY

Refer to Line Mesh Generation for further information on this feature.

KEYWORD FORMAT

A block of lines defining locations along a line. The block begins with a line defining the line name. This is followed by one or more lines defining named locations in terms of distance along the line. The entire block of lines may be repeated as often as necessary to define all the required locations along lines in the model.

Line defining the line name:

`LINE=`*`Line Name`*

`Line` defining named locations in terms of distance along the line (measured from the start of the line). This line may be repeated as often as necessary to define all the required locations along the line:

`LABEL=`*Node Label, Distance*

Any *`Line Name`* which is referenced must be defined under `*LINES`.

TABLE INPUT

Input:	Description

Line Name:	The name of the line.
Location Name:	The name of location point.
Distance Along Line:	The distance along the line (measured from the start of the line) to the location point

NOTES

(a) Any `Line Name` you reference must be defined in the Lines table

(b) It is important to associate meaningful names with line locations, as these names will be used by the program to automatically create relevant node labels for you, which you can subsequently reference. For example, if you wish to apply a point load at a certain distance along a line, or to connect another component (e.g. another line) to a line at a specific location, you would typically position a node on the line at the required location.

(c) As well as defining node labels, the program will automatically create a pair of element labels on the elements at either side of the location. These element labels will be given the same name as the node label, but with the suffixes "_Before" and "_After" appended. In particular, this greatly simplifies the post-processing of element based results at the location.

1.8.10.33 *LINE SECTION GROUPS

PURPOSE

To define groups of line sections which may be reused and repeated within sections defined in `*LINES`.

THEORY

Refer to Modelling Repeating Sub-Sections for further information on this feature.

KEYWORD FORMAT

A block of lines defining a line section group and its associated sections. The entire block of lines defining the individual line section group may be repeated as often as necessary to define all the required line section groups in the model.

The first line defines a name for the line section group.

```
NAME=Line Section Group Name
```

A block of one to three lines defining a section. The first line defines a set name for elements within the section, and the length of the section. The optional second line defines an additional set or sets to which the elements within the sub-section are to be added. The optional third line defines suggested minimum and maximum element lengths for the sub-section to be used during mesh generation. This block may be repeated as often as necessary to define all the required sections in the group.

```
SECTION=Section Set Name, Length
[ADDITIONAL=Set Name, Set Name, Set Name etc.]
Minimum Element Length (at section start) [, Maximum Element Length] [Minimum
```

TABLE INPUT

Input:	Description
Group Name:	Line section group name.
Section Set Name:	The element set name for the line section definition.
Length:	Length of the section.
Min. Element Length at Start:	The minimum element length to be used in meshing the line section. The first element at the start of the line section will be assigned a length which is identical or very close to this value.

Max. Element Length:	The maximum element length to be used in meshing the line section. The lengths of successive elements beginning from the start of the line section are gradually increased up to this value. This entry is optional, and if omitted, a uniform mesh density is typically created, based on the specified *Min. Element Length at Start*.
Min. Element Length at End:	The minimum element length to be used in meshing the line section towards the end of its length. The last element at the end of the line section will be assigned a length which is identical or very close to this value. This entry is optional, and if omitted, it defaults to the *Min. Element Length at Start* entry for the next section. This allows the meshing algorithm to automatically blend sections into one another.
Additional Sets for this Section:	Additional set or sets to which the elements within the sub-section are to be added. This may be useful in combining different sections which have similar structural or hydrodynamic properties. If more than one set is referenced, use commas to separate out the set names.

Notes

(a) Use as many rows as you need to completely define all the sections within a particular group. Simply leave Column 1 blank for the second and subsequent sections. For subsequent groups, put the *Line Name* in Column 1 and specify the line sections in the same way.

(b) It is important to define meaningful names (for *Section Set Name* entries, and any *Additional Set* names if applicable), as these names will be used by the program to automatically create relevant element sets for you, which you can subsequently reference when you are assigning structural and hydrodynamic properties to your model.

(c) Refer to Modelling Repeating Sub-Sections for further background information on the use of line section groups.

(d) Refer to Line Mesh Generation for a detailed discussion of the automatic mesh creation facility.

1.8.10.34 *LINES

PURPOSE

To define a line (and sections within that line), by specifying relevant set names, lengths, start and end locations, and mesh generation settings.

THEORY

Refer to Lines for further information on this feature.

KEYWORD FORMAT

A block of lines defining an individual line and its associated settings. The block begins with a line defining the line name, and (optionally) its length. This is followed by an optional line defining a default set name for elements within the line which are not part of any sub-sections (if present). The next line defines a node label for the start of the line, along with its coordinates, followed by a similar definition for the end of the line. The next line contains suggested minimum and maximum element lengths for the line to be used during mesh generation. This is followed by an optional line defining undeformed orientation of the elements used to model the line. An optional block consisting of two to three lines defining sub-sections may then follow, and this block may be repeated as often as necessary to define all the required sub-sections in the line. The entire block of lines defining the individual line may then be repeated as often as necessary to define all the required lines in the model.

Line defining the line name, and (optionally) its length. A further option controls whether single or multiple cables are used to model subsections of the line which have different structural properties. Omitting this entry is suitable in the vast majority of circumstances.

```
LINE=Line Name [, Line Length] [, FORCE=SINGLE]
```

Optional line defining a default set name for elements within the line which are not part of any sub-sections (if present):

```
[NON SECTION=Default Set Name]
```

Line defining a node label for the start of the line, along with its coordinates:

```
START=Start Node Label, X [,Y] [,Z]
```

Line defining a node label for the end of the line, along with its coordinates:

```
END=End Node Label, X [,Y] [,Z]
```

Line defining suggested minimum and maximum element lengths for the line to be used during mesh generation:

```
Minimum Element Length (at line start) [, Maximum Element Length] [Minimum El
```

Optional line defining undeformed orientation of the elements used to model the line:

```
[v1, v2, v3, w1, w2, w3]
```

Optional block consisting of one to three lines defining sub-sections. The first line defines a set name for elements within the sub-section, the start and end distances of the sub-section along the line (measured from the start of the line) and optionally the line section group to be repeated across the section. Any Line Section Group Name which is referenced must be defined under *LINE SECTION GROUPS. The optional second line defines an additional set or sets to which the elements within the sub-section are to be added. The optional third line defines suggested minimum and maximum element lengths for the sub-section to be used during mesh generation. This block may be repeated as often as necessary to define all the required sub-sections in the line.

```
SECTION=Section Set Name, Start Distance [, End Distance] [, REPEAT=Line Sect
[ADDITIONAL=Set Name, Set Name, Set Name etc.]
Minimum Element Length (at section start) [, Maximum Element Length] [Minimum
```

If `Line Length` is omitted, is equal to the straight line distance between the line's end points. Y and Z default to zero if omitted.

If the `Minimum Element Length` (at line or section start) is the only meshing parameter specified, then a uniform mesh density is created based on this length. `Aspect Ratio` may be STANDARD (default), FINE or SUPERFINE. If the `Maximum Element Length` is omitted, a uniform mesh density is typically created, based on the specified `Minimum Element Length (at start)`. If the `Minimum Element Length (at line end)` is omitted, it defaults to the corresponding `Minimum Element Length (at line start)`. If the `Minimum Element Length (at section end)` is omitted, it defaults to the `Minimum Element Length (at section start)` for the next section (or the end of the line itself, if it is the last section). This allows the meshing algorithm to automatically blend sections into one another.

The default values of v1, v2, v3, w1, w2, w3 depend on whether the line is modelled internally using a cable or straight section. Refer to Undeformed Versus Initial Positions for a detailed discussion on this topic.

LINES

Purpose

To define a line, by specifying relevant set names, line length, start and end locations, and mesh generation settings.

Table Input

Input:	Description
Line Name:	The name of the line.
Non-Section Set Name:	The element set name for any portions of the line which are not included in any line section definitions. Sections are defined using the *Line Sections* table.
Length:	The length of the line. This is an optional input, and if omitted, it is equal to the straight line distance between the line's end points.
Start Node Label:	The node label for the start of the line.
Start (X):	The global X coordinate of the start of the line.
Start (Y):	The global Y coordinate of the start of the line. This entry is optional and defaults to zero if omitted.
Start (Z):	The global Z coordinate of the start of the line. This entry is optional and defaults to zero if omitted.
End Node Label:	The node label for the end of the line.
End (X):	The global X coordinate of the end of the line.
End (Y):	The global Y coordinate of the end of the line. This entry is optional and defaults to zero if omitted.

End (Z):	The global Z coordinate of the end of the line. This entry is optional and defaults to zero if omitted.
Min. Element Length at Start:	The minimum element length to be used in meshing the line. The first element at the start of the line will be assigned a length which is identical or very close to this value.
Max. Element Length:	The maximum element length to be used in meshing the line. The lengths of successive elements beginning from the start of the line are gradually increased up to this value. This entry is optional, and if omitted, a uniform mesh density is typically created, based on the specified *Min. Element Length at Start*.
Min. Element Length at End:	The minimum element length to be used in meshing the line towards the end of its length. The last element at the end of the line will be assigned a length which is identical or very close to this value. This entry is optional, and if omitted, it defaults to the corresponding *Min. Element Length at Start* entry.
V1, V2, V3:	The components in the global coordinate axes of a vector V, (one of two) defining the undeformed orientation of the elements used to model the line. These entries are optional and will not be invoked in the vast majority of cases.
W1, W2, W3:	The components in the global coordinate axes of a vector W, (one of two) defining the undeformed orientation of the elements used to model the line. These entries are optional and will not be invoked in the vast majority of cases.
Mesh Ratio:	The meshing algorithm ensures that the ratio between the lengths of adjacent elements cannot exceed a certain value. This option allows you to set the value of an optimum ratio. See Note (h).
Force Single Catenary:	This option controls whether single or multiple cables are used to model subsections of the line which have different structural properties. The default option (*No*) is suitable in the vast majority of circumstances. See Note (i).

Notes

(a) Refer to Line Mesh Generation for a detailed discussion of the automatic mesh creation facility.

(b) It is important to define meaningful names (for *Line Name* and *Non-Section Set Name* entries), as these names will be used by the program to automatically create relevant element sets for you, which you can subsequently reference when you are assigning structural and hydrodynamic properties to your model.

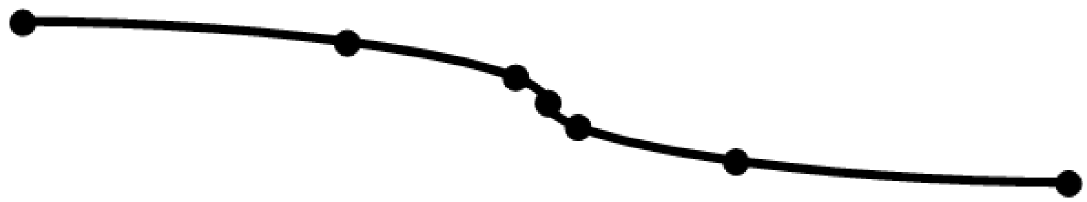
(c) It is important to associate meaningful labels with the line start and end locations, as these labels will be used by the program to automatically create relevant node labels for you, which you can subsequently reference when you are applying boundary conditions to your model.

(d) Based on the length of a line, and the straight line distance between its start and end locations, Flexcom automatically determines whether the line should be modelled internally using a curved (cable) or straight section.

(e) You have control over the distribution of the elements along the line, via the specification of desired maximum and minimum element lengths, and also the division of the line into several subsections if required. The meshing algorithm automatically generates a finite element discretisation based on the guidelines you provide. It also attempts to prevent large changes in relative element length across the finite element mesh by gradually stepping up and down element lengths along the structure, and to avoid over-meshing by using longer elements in the middle of long sections of continuous properties.

(f) The specification of the components of the vectors V and W is related to the Flexcom facility for analysing a structure which is initially deformed. Basically, you specify V and W explicitly when the configuration defined by the nodal coordinates you have input does not represent a stress-free structure orientation. However, in the majority of analyses, this is not the case, and the specified nodal coordinates do represent the undeformed as well as the initial position. In this situation, the specification of V and W is optional, and should in general be omitted. If you want to enter V and W values then all of the values V1, V2, V3, W1, W2, and W3 must be entered. When you do not define V and W explicitly, Flexcom calculates nominal values for these vectors using a default algorithm, based on the specified nodal coordinates. Refer to Undeformed Versus Initial Positions for a detailed discussion on this topic.

(g) A simple example might be that of the curved flowline as shown below. In order to follow a pre-defined lay path, the flowline might be modelled using a series of straight lines, each representing a section aligned in a particular direction (note that each line could be comprised of several elements). The lines would be connected at the intersection point using the equivalent nodes facility. By default, the program will assign local undeformed axes to each element within the model based on the coordinates of its end nodes. This will result in a unique stress-free orientation for each element, and mean that the flowline is completely unstressed in its as-laid configuration as shown. In such circumstances, it is desirable to assign a common local undeformed axis system to each element (defined via lines in this instance), so that while the flowline is restricted (via seabed friction for example) in its as-laid configuration, the required bending moment distribution is captured.



(h) The meshing algorithm attempts to prevent large changes in relative element length across the finite element mesh by gradually stepping up and down element lengths along the structure. The algorithm ensures that the ratio between the lengths of adjacent elements cannot exceed a certain value. The maximum ratio defaults to 1.5 (*Standard* option), but may be reduced to 1.25 (*Fine* option) or 1.1 (*Super Fine* option). The actual value of the ratio is automatically selected to correctly fill the meshed section with elements.

(i) Flexcom will typically use a separate cable to model each subsection of a line which has different structural properties. In the case of a steep wave riser for example, this is perfectly reasonable. However, in very occasional circumstances, it may be desirable to model several discrete subsections using a single catenary representation. For example, if you are modelling a buoyant section of riser which is comprised of standard material supported by numerous discrete buoyancy modules along its length, a single catenary model is preferable. Otherwise a large number of separate cables will be used internally to model what is essentially a simple catenary, and some or all of the individual cables are likely to violate the basic assumptions associated with catenary equations. In such circumstances, you can use the *Force Single Catenary* option to explicitly request that an entire line is to modelled using one single catenary/cable.

LINES SECTIONS

Purpose

To define sections within a line, by specifying relevant set names, start and end distances, and mesh generation settings.

Table Input

Input:	Description
Line Name:	The name of the line. See Note (b).
Section Set Name:	The element set name for the line section definition.
Start Distance:	The distance along the line (measured from the start of the line) to the start of the line section. A negative value may be used to define a distance measured from the end of the line.
End Distance:	The distance along the line (measured from the start of the line) to the end of the line section. A negative value may be used to define a distance measured from the end of the line.

Min. Element Length at Start:	The minimum element length to be used in meshing the line section. The first element at the start of the line section will be assigned a length which is identical or very close to this value.
Max. Element Length:	The maximum element length to be used in meshing the line section. The lengths of successive elements beginning from the start of the line section are gradually increased up to this value. This entry is optional, and if omitted, a uniform mesh density is typically created, based on the specified *Min. Element Length at Start*.
Min. Element Length at End:	The minimum element length to be used in meshing the line section towards the end of its length. The last element at the end of the line section will be assigned a length which is identical or very close to this value. This entry is optional, and if omitted, it defaults to the *Min. Element Length at Start* entry for the next section (or the *Min. Element Length at End* of the line itself, if it is the last section). This allows the meshing algorithm to automatically blend sections into one another.
Additional Sets for this Section:	Additional set or sets to which the elements within the sub-section are to be added. This may be useful in combining different sections which have similar structural or hydrodynamic properties. If more than one set is referenced, use commas to separate out the set names.
Repeat Group:	The line section group which will be repeated over the length of the section.

Notes

(a) Any *Line Name* you reference must be defined in the *Lines* table. Use as many rows as you need to completely define all the sections within a particular line. Simply leave Column 1 blank for the second and subsequent sections. For subsequent lines, put the *Line Name* in Column 1 and specify the line sections in the same way.

(b) It is important to define meaningful names (for *Section Set Name* entries, and any *Additional Set* names if applicable), as these names will be used by the program to automatically create relevant element sets for you, which you can subsequently reference when you are assigning structural and hydrodynamic properties to your model.

(c) Refer to Line Mesh Generation for a detailed discussion of the automatic mesh creation facility.

1.8.10.35 *LINES PIP

PURPOSE

To specify that two lines are connected in a pipe-in-pipe (or pipe-on-pipe) configuration.

THEORY

Refer to Pipe-in-Pipe Configurations for further information on this feature.

KEYWORD FORMAT

A block of lines defining connections between two lines. The block begins with a line defining the primary and secondary lines, the configuration type, and an optional offset term. This is optionally followed by one or more lines defining linear, non-linear or rigid connections between the lines at regular intervals in the axial direction. This line may be repeated as often as necessary to define all the required connections between the lines. The entire block of lines may be repeated as often as necessary to define all the required locations along lines in the model.

Line defining the primary and secondary lines, the configuration type, and an optional offset term:

```
PRIMARY=Primary Line Name, SECONDARY=Secondary Line Name, TYPE=Configuration
```

Optional lines defining linear, non-linear or rigid connections between the lines at regular intervals. Distances are measured from the start of the primary line. This line may be repeated as often as necessary to define all the required connections between the lines:

```
[STIFFNESS=Stiffness or CURVE=Curve Name or BULKHEAD, Spacing, Start Distance
```

The Primary Line Name and Secondary Line Name referenced must be defined under *LINES. Configuration may be PIP or POP. Offset defaults to zero if omitted. Any Curve Name referenced must be defined under *PIP STIFFNESS.

PIPE-IN-PIPE SECTIONS

Table Input

Input:	Description
Primary Line Name:	The name of the primary line.
Secondary Line Name:	The name of the secondary line.
Configuration Type:	The configuration type, whether Pipe-in-Pipe (the default) or Pipe-on-Pipe.
Connection Table:	The name of the relevant connection definition.
Offset for Secondary Line:	The difference in length between (i) the distance of the first connection along the *Primary Line* and (ii) the distance of the first connection along the *Secondary Line*. This entry is optional and defaults to zero if omitted

Notes

(a) Refer to Line Mesh Generation for a detailed discussion of the automatic mesh creation facility.

(b) Any *Primary Line Name* or *Secondary Line Name* you reference must be defined in the *Lines* table.

(c) Any *Connection Table* you reference must be defined in the *Lines - Pipe-in-Pipe Connections* table.

(d) The *Offset for Secondary Line* input might be used in situations where the ends of the *Primary Line* and the *Secondary Line* are misaligned in the axial direction. In this case, you may wish to specify an offset term in order to ensure that the actual connections are aligned in the axial direction.

PIPE-IN-PIPE CONNECTIONS

Table Input

Input:	Description
Table Name:	The name of the connection definition. This is referenced by the *Lines - Pipe-in-Pipe Section* table. See Note (b).
Type:	The connection type, whether *Pipe-in-Pipe* (the default) or *Bulkhead*.
Spacing:	The spacing between successive connections.
Start:	The distance along the line (measured from the start of the *Primary* line) to the first connection.
End:	The distance along the line (measured from the start of the *Primary* line) to the last connection.
Stiffness:	The linear stiffness of the connection between these nodes or the name of a non-linear force-deflection curve that defines this stiffness (defined via the *Pipe-in-Pipe Stiffness* table). This entry is not relevant to Bulkhead connections.

Notes

(a) Use as many rows as you need to completely define all the pipe-in-pipe connections within a particular set of connections. Simply leave Column 1 blank for the second and subsequent connections within that set. For subsequent sets, put the *Table Name* in Column 1 and specify the pipe-in-pipe connections in the same way.

1.8.10.36 *LOCAL AXIS SYSTEM

PURPOSE

To define local axis systems.

THEORY

Refer to User-Defined Axes for further information on this feature.

KEYWORD FORMAT

The keyword starts with a line defining the name of the axis system. This is then followed by a line with 6 numbers.

Line defining the axis system name:

```
NAME=AXIS NAME
```

Line defining an axis system:

```
X1, Y1, Z1, X2, Y2, Z2
```

The two vectors defining an axis system must have non-zero length and must be orthogonal.

TABLE INPUT

Input:	Description
Name:	A unique label for the axis system being defined.
X1:	The component in the global X-direction of the local x-axis of the axis system being defined. See Note (a).
Y1:	The component in the global Y-direction of the local x-axis of the axis system being defined. See Note (a).
Z1:	The component in the global Z-direction of the local x-axis of the axis system being defined. See Note (a).
X2:	The component in the global X-direction of the local y-axis of the axis system being defined. See Notes (a) and (b).

Y2:	The component in the global Y-direction of the local y-axis of the axis system being defined. See Notes (a) and (b).
Z2:	The component in the global Z-direction of the local y-axis of the axis system being defined. See Notes (a) and (b).

NOTES

(a) If you do not specifically input a value in any of Columns 2-7, then the corresponding variable defaults to zero. Note however that all three of *X1, Y1* and *Z1* or all three of *X2, Y2* and *Z2* cannot equal zero – the null vector cannot be either the local x-axis or the local y-axis.

(b) Note that the local x-axis and local y-axis that you define here must be *orthogonal*, that is, the true angle between the two vectors defining the local x- and y-axes must be 90°. Flexcom will generate an error message if this is not the case, as vectors that are not orthogonal cannot be used to form a valid axis system.

1.8.10.37 *MASS

PURPOSE

To specify a point mass or a point rotary inertia.

THEORY

Refer to Point Masses and Point Buoys for further information on this feature.

KEYWORD FORMAT

Two types of line which can be repeated as often as necessary.

Line defining a point mass:

```
Node (Number or Label), Mass [, TYPE=MASS]
```

Line defining a point mass without any gravitational force/weight term:

```
Node (Number or Label), Mass, TYPE=NO WEIGHT MASS
```

Line defining a point rotary inertia:

```
Node (Number or Label), Inertia, TYPE=ROTARY INERTIA
```

If you specify a node label rather than a node number, it must be enclosed in {} brackets.

TABLE INPUT

Input:	Description
Node:	The node (number or label) at which the point mass/inertia is located. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Mass or Inertia:	The magnitude of the point mass/inertia.
Type:	This option allows you to choose between a point mass (the default), a point mass without any gravitational force/weight term, and a point rotary inertia.

NOTES

(a) Point masses are typically included in both the force vector (as a weight term) and the mass matrix (as an inertia term), but you can omit the former if you wish. This could be useful in Floating Body analysis, where gravitational (and mean buoyancy) forces acting on the floating body are generally not included. It is typically assumed that the initial position of floating body represents the static equilibrium configuration, hence the static gravity and buoyancy forces cancel each other out, and so are not modelled explicitly. Dynamic effects are included obviously. The `*FLOATING BODY` keyword allows you to model the vessel mass at the centre of gravity of the hull, but you may wish to include other masses at different locations, such as a wind turbine generator.

1.8.10.38 *MOMENT-CURVATURE

PURPOSE

To define moment-curvature curves for non-linear materials.

THEORY

Refer to Non-Linear Elastic materials for further information on this feature.

Note also that the old *STRESS/STRAIN keyword has effectively been superseded by the new non-linear material definition keywords which explicitly distinguish between bending, axial and torsional stiffness. Refer to related keywords *FORCE-STRAIN and *TORQUE-TWIST.

KEYWORD FORMAT

A block of lines that defines a moment-curve curve, repeated as often as necessary. The block begins with a line defining the curve name. It is followed by as many lines as necessary to define each point on the curve.

Line defining the curve name:

```
CURVE=Curve Name
```

Line defining a point on a curve:

```
Moment, Curvature
```

Each curve must have at least two points defined. This type of moment-curvature curve may not be associated with non-linear beam elements which are defined using the rigid riser format for geometric properties specification.

TABLE INPUT

Input:	Description
Curve Name:	The generic name of the moment-curvature curve.
Moment:	A moment value for a point on the curve.
Curvature:	The corresponding curvature value.

NOTES

(a) This keyword is used to define moment-curvature curves that define EIyy and EIzz for a particular set of elements. Moment-curvature curves may be assigned to element sets using the *GEOMETRIC SETS keyword.

(b) Use as many lines as you need to completely define a particular moment-curvature curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent moment-curvature curves, put the curve name in Column 1 and specify the moment-curvature data in the same way.

(c) The points defining the non-linear moment-curvature curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of curvature.

(d) If the curvature in an element lies between the moment-curvature data points you specify, Flexcom uses linear interpolation to determine the relevant stiffness for the element.

(e) If the curvature in the element lies outside the specified range of the moment-curvature curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

(f) If none of the specified curvature terms have a negative value, the curve is assumed to be symmetrical about the origin (i.e. the behaviour of the element is the same for both positive and negative curvatures).

1.8.10.39 *MOONPOOL

PURPOSE

To define the location and extent of the vessel moonpool at solution initiation, for the purpose of applying hydrodynamic loading on elements of the structure contained within the vessel moonpool.

THEORY

Refer to Moonpool Hydrodynamics for further information on this feature.

KEYWORD FORMAT

A block of two lines repeated as many times as necessary.

```
VESSEL=Vessel Name
X0, Y0, Z0, θ, b
```

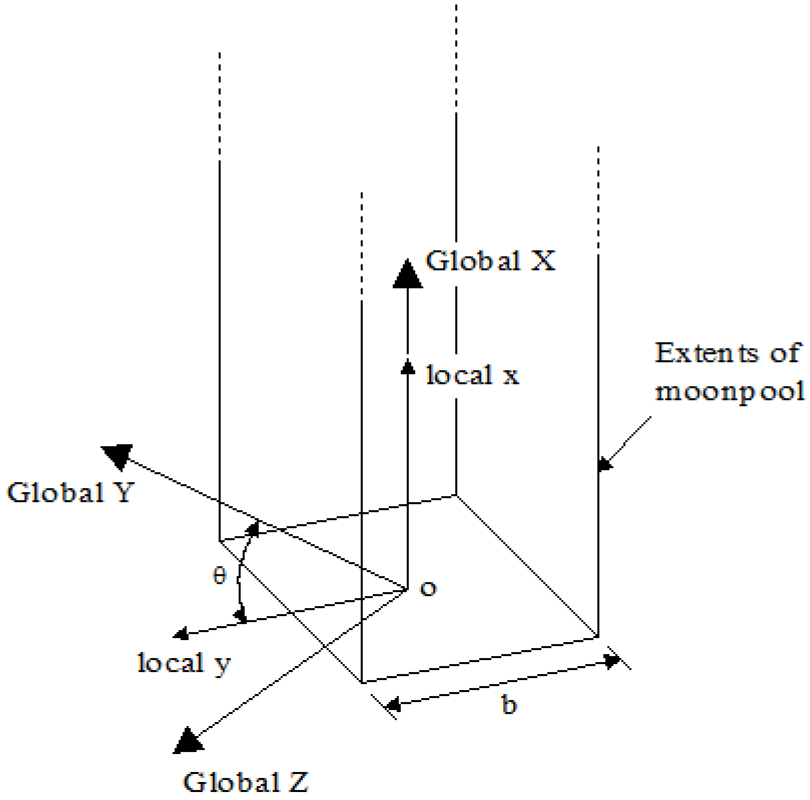
A vessel cannot have more than one moonpool.

TABLE INPUT

Input:	Description
Vessel Name:	The name of the vessel whose motions define the motion of the moonpool.
X0:	The global X-coordinate of the origin of the vessel moonpool at solution initiation. See Note (a).
Y0:	The global Y-coordinate of the origin of the vessel moonpool at solution initiation. See Note (a).
Z0:	The global Z-coordinate of the origin of the vessel moonpool at solution initiation. See Note (a).
Theta:	The initial yaw orientation of the moonpool in degrees measured anticlockwise from the global Y-axis. See Note (a).
Width:	The moonpool width, and overall height of the transition region. See Notes (a) and (c).

NOTES

(a) The vessel moonpool entrains a volume of seawater that is assumed to translate and rotate with the vessel in question. The water particle velocities and accelerations within the area enclosed by the moonpool are calculated from the vessel motions, as opposed to from the ambient wave field. The moonpool is assumed to be square in cross-section and extends from the origin of the moonpool (which would typically be at the same level as the vessel keel) to above the mean water level, as shown below.



The initial location of the volume enclosed by the moonpool is defined by specifying the global X, Y & Z coordinates of the moonpool origin, which is located at the centre of the square that forms the bottom face of the volume enclosed by the moonpool. The orientation of the volume enclosed by the moonpool in the global YZ-plane at solution initiation is defined by θ, which is the angle between the local moonpool y-axis and the global-Y axis (measured anticlockwise from the global-Y axis). In most circumstances, this would be the same as the initial vessel yaw orientation. At solution initiation, the volume enclosed by the moonpool is assumed to be orientated vertically (that is, the local moonpool x-axis is aligned with the global X-axis).

(b) The initial location and orientation of the volume enclosed by the moonpool corresponds to the initial position of the vessel with which the moonpool is associated. Any movement of the vessel from its initial position (including movement due to offset, drift, or vessel RAOs) will cause a corresponding movement of the volume enclosed by the moonpool.

(c) As noted above, the moonpool width *b* also defines the overall height of a transition region between the volume enclosed completed by the moonpool and the ambient wave field. This transition zone, and how Flexcom treats riser element within it, is illustrated and discussed in the notes of the *Moonpool Hydrodynamic Properties* table. You are referred to that table for further details if necessary.

1.8.10.40 *MOORED VESSEL

PURPOSE

To define a moored vessel and its associated properties.

THEORY

Refer to Moored Vessel for further information on this feature.

KEYWORD FORMAT

A block of eight lines defining the vessel parameters with the format:

TYPE=MASS
M_{yy}, $I_{\theta\theta}$, Ma_{yy}, Ma_{zz}, $Ma_{\theta\theta}$, $Ma_{z\theta}$,
TYPE=GEOM
AL, *AT*, L_{fore}, L_{aft}, *T*, *B*
TYPE=DAMP
Byy, *Bzz*, $B_{\theta\theta}$
TYPE=COG
Node (Number or Label) [, θ_G*]*

where:

M_{yy} = the mass of the vessel

$I_{\theta\theta}$ = the vessel moment of inertia in yaw

Ma_{yy} = the vessel added mass in surge

Ma_{zz} = the vessel added mass in sway

$Ma_{\theta\theta}$ = the vessel added mass in yaw

$Ma_{z\theta}$ = the vessel added mass in sway-yaw coupling

AL = the longitudinal area of the vessel exposed to wind action.

AT = the transverse area of the vessel exposed to wind action.

L_{fore} = the length from the vessel centre of gravity to the forward perpendicular

L_{aft} = the length from the vessel centre of gravity to the aft perpendicular

T = the vessel draft.

B = the vessel beam.

B_{yy} = the linear damping coefficient in surge

B_{zz} = the linear damping coefficient in sway

$B_{\theta\theta}$ = the linear damping coefficient in yaw

Node = the node (number or label) located at the vessel COG

θ_G = the heading of the undisplaced vessel (in degrees), measured anticlockwise from the global Y-axis

This keyword is only relevant to mooring analyses. The order in which the TYPE=statements appear is not significant, as long as the relevant data is given under each TYPE=statement. The parameters relating to vessel mass and inertia should be calculated with reference to the vessel centre of gravity. Only one moored vessel may be specified in a mooring analysis and all TYPE=VESSEL boundary conditions and subsequent *RAO data should refer to that vessel. If you specify a node label rather than a node number, it must be enclosed in {} brackets.

MASS

Purpose

To input data defining the inertia of a moored vessel.

Table Input

Input:	Description
Mass of Vessel:	The mass of the vessel.

Yaw Moment of Inertia:	The vessel moment of inertia in yaw.
Surge Added Mass:	The vessel added mass in surge.
Sway Added Mass:	The vessel added mass in sway.
Yaw Added Mass:	The vessel added mass in yaw.
Surge-Yaw Coupling Added Mass:	The vessel added mass in sway-yaw coupling.

GEOMETRY

Purpose

To input data defining the geometry of a moored vessel.

Table Input

Input:	Description
Longitudinal Area of Vessel Exposed to Wind Action:	The longitudinal area of the vessel exposed to wind action.
Transverse Area of Vessel Exposed to Wind Action:	The transverse area of the vessel exposed to wind action.
Length from Vessel COG to Forward Perpendicular:	The length from the vessel centre of gravity to the forward perpendicular.
Length from Vessel COG to Aft Perpendicular:	The length from the vessel centre of gravity to the aft perpendicular.
Vessel Draft:	The vessel draft.
Vessel Beam:	The vessel beam

DAMPING

Purpose

To define linear damping coefficients for a moored vessel.

Table Input

Input:	Description
Surge Linear Damping Coefficient:	The vessel linear damping coefficient in surge.
Sway Linear Damping Coefficient:	The vessel linear damping coefficient in sway.
Yaw Linear Damping Coefficient:	The vessel linear damping coefficient in yaw.

LOCATION

Purpose

To define the location of the vessel centre of gravity and the undisplaced vessel heading.

Table Input

Input:	Description
Node:	The node (number or label) of the finite element mesh located at the vessel COG. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Undisplaced Vessel Heading :	The heading of the undisplaced vessel (in degrees), measured anticlockwise from the global-Y axis.

1.8.10.41 *NODE

PURPOSE

To define nodal coordinates in the global Cartesian coordinate system.

THEORY

Refer to Nodes for further information on this feature.

Note also that the old `*NODE`, `*ELEMENT` and `*CABLE` keywords have been largely superseded by the new `*LINES` keyword. Lines provide an automatic mesh creation facility to greatly expedite the model creation process. Using lines is a fundamentally different approach to working directly with nodes, elements and cables, although the information is ultimately handled in the same fashion internally. Since lines provide automatic mesh generation, you do not to concern yourself with explicit node and element numbering. Indeed the availability of lines makes nodes, elements and cables redundant to some degree, but they are retained for complete generality, and also to maintain downward compatibility with previous program versions. Refer to Lines for further information on this feature.

KEYWORD FORMAT

Two types of lines that may be mixed and/or repeated as often as necessary.

Line defining a single node:

```
Node, X Co-ordinate, Y Co-ordinate [, Z Co-ordinate]
```

Line generating a line of nodes between two existing nodes:

```
GEN=Start Node, End Node [, Node Increment]
```

Node Increment defaults to 1.

TABLE INPUT

Nodes – Define Directly

Input:	Description
Node:	The node (number) being defined.

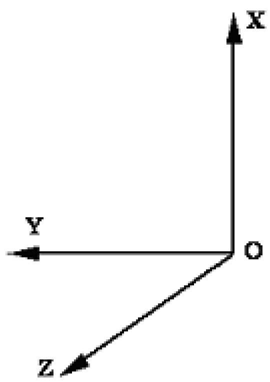
X Coordinate:	The nodal coordinate in the global X-direction.
Y Coordinate:	The nodal coordinate in the global Y-direction.
Z Coordinate:	The nodal coordinate in the global Z-direction. This defaults to a value of 0.

Nodes – Generate

Input:	Description
Start Node:	The node (number) at the start of the straight-line segment.
End Node:	The node (number) at the end of the straight-line segment.
Increment:	The node number increment to be used in assigning numbers to the generated nodes. This input defaults to a value of 1, which will apply in the majority of cases.

NOTES

(a) The global Cartesian coordinate system is shown below and defines a right-handed system. Note that the global X axis is vertical in Flexcom.



(b) Node numbers do not need to be continuous in a Flexcom model - you can use any arbitrary scheme for assigning node numbers. This can be of considerable benefit when defining complex structures, such as for example multi-line riser systems, and can also greatly simplify the refining of a finite element mesh. Furthermore, as Flexcom incorporates automatic internal bandwidth optimisation, the node numbering scheme does not affect the analysis computation time.

(c) In certain circumstances, the coordinates specified here may represent only approximate nodal locations. This is the case when you use cables in defining the model geometry. For a more complete discussion, refer to Approximate Nodal Locations.

(d) The nodal coordinates specified here represent the initial position of the structure. This need not necessarily be an undeformed or stress-free configuration, although it will be in the majority of applications. For a more complete discussion, refer to Undeformed Versus Initial Positions.

(e) Node numbers are assigned to generated nodes by repeatedly adding the specified or default increment to the previous number (obviously starting at the start node), until the end node number is reached or exceeded. So for example, the specification Start node - 1, End Node - 11, Increment - 2 will result in 4 nodes, numbered 3, 5, 7 and 9, and dividing the straight line between nodes 1 and 11 into 5 equal segments.

1.8.10.42 *NODE,AUXILIARY

PURPOSE

To define auxiliary nodal coordinates in the global Cartesian coordinate system.

THEORY

Refer to Auxiliary Bodies for further information on this feature.

Note also that the old auxiliary body keywords (`*NODE,AUXILIARY`, `*ELEMENT,AUXILIARY`, `*AUXILIARY`, `*PANEL,AUXILIARY` and `*PANEL SECTIONS,AUXILIARY`) keywords have effectively been superseded by the integrated vessel/body features which provide a range of standard vessel and subsea component profiles. Refer to `*VESSEL,INTEGRATED` and `*BODY,INTEGRATED` for further information.

KEYWORD FORMAT

One type of line repeated as often as necessary.

```
Node, X Co-ordinate, Y Co-ordinate [, Z Co-ordinate]
```

TABLE INPUT

Input:	Description
Node:	The number of the auxiliary node being defined.
X Coordinate:	The nodal coordinate in the global X-direction.
Y Coordinate:	The nodal coordinate in the global Y-direction.
Z Coordinate:	The nodal coordinate in the global Z-direction. This defaults to a value of 0.

NOTES

(a) Auxiliary nodes, elements and bodies allow you to include objects in your model that are not part of the finite element discretisation, but which are included for illustrative purposes only. Outline vessel models are the most common example of such objects.

(b) Auxiliary nodes and structural nodes (nodes that are included in the finite element discretisation) cannot use the same node number. Each node in the model must be assigned a unique node number irrespective of whether it is a structural node or an auxiliary node.

1.8.10.43 *NODE,CURVILINEAR

PURPOSE

To specify nodes at particular locations along a cable, or to specify a number of equally spaced nodes along a cable segment between two nodes.

THEORY

Refer to Defining Nodes in Terms of Cables for further information on this feature.

Note also that the old *NODE, *ELEMENT and *CABLE keywords have been largely superseded by the new *LINES keyword. Lines provide an automatic mesh creation facility to greatly expedite the model creation process. Using lines is a fundamentally different approach to working directly with nodes, elements and cables, although the information is ultimately handled in the same fashion internally. Since lines provide automatic mesh generation, you do not to concern yourself with explicit node and element numbering. Indeed the availability of lines makes nodes, elements and cables redundant to some degree, but they are retained for complete generality, and also to maintain downward compatibility with previous program versions. Refer to Lines for further information on this feature.

KEYWORD FORMAT

Two types of lines that may be repeated as often as necessary.

Line defining a single node on a cable:

```
Node (Number or Label), Cable no., Distance Along Cable
```

Line generating a number of nodes along a cable.

```
CURV=Start Node (Number or Label), End Node (Number or Label) [, Node Increme
```

Any cable numbers used must be defined under *CABLE. If you specify a node label rather than a node number, it must be enclosed in {} brackets. *Node Increment* defaults to 1.

TABLE INPUT

Cables - Define Nodes

Input:	Description
Node:	The node (number or label) being defined. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Cable Number :	The number of the cable on which the node is being defined.
Distance Along Cable:	The distance along the cable measured from the start node to the node being defined

Cables – Generate Nodes

Input:	Description
Start Node:	The node (number or label) at the start of the cable segment. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
End Node:	The node (number or label) at the end of the cable segment. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Cable Number:	The number of the cable along which the nodes are being generated.
Node Increment:	The node number increment to be used in assigning numbers to the generated nodes. This input defaults to a value of 1, which will apply in the majority of cases

NOTES

(a) The *Cables - Define Nodes* option is commonly used for mesh refinement on a cable.

(b) Node numbers are assigned to generated nodes by repeatedly adding the specified or default increment to the previous number (obviously starting at the start node) until the end node number is reached or exceeded. So for example, the specification Start Node - 1, End Node - 11, Increment - 2 will result in 4 nodes, numbered 3, 5, 7 and 9, and dividing the cable between Nodes 1 and 11 into 5 equal segments.

1.8.10.44 *NODE SPRING

PURPOSE

To specify the stiffness and direction of node springs.

THEORY

Refer to Spring Elements for further information on this feature.

KEYWORD FORMAT

One type of line repeated as often as necessary.

```
Node (Number or Label), DOF, Stiffness
```

If you specify a node label rather than a node number, it must be enclosed in {} brackets.

TABLE INPUT

Input:	Description
Node:	The node (number or label) at which the node spring is located. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global degree of freedom (DOF) defining the spring line of action. Specify a value of 1 for the global X-direction, 2 for the global Y-direction and 3 for the global Z-direction. A value of 4, 5 or 6 gives a rotational node spring.
Stiffness:	The linear spring stiffness of the node spring in the specified DOF. The dimension of spring stiffness in DOFs 1-3 is force per unit length. The dimension of spring stiffness in DOFs 4-6 is moment per degree rotation.

NOTES

(a) The line of action of a node spring remains constant throughout the analysis.

1.8.10.45 *NONLINEAR STIFFENER

PURPOSE

To define a stress-strain curve for a non-linear bend stiffener.

THEORY

Refer to Bend Stiffeners for further information on this feature.

KEYWORD FORMAT

A block of lines that define a stress-strain curve repeated as often as necessary. The block begins with a line defining the curve name. It is followed by as many lines as necessary to define each point on the curve.

Line defining the curve name:

```
CURVE=Curve Name
```

Line defining a point on a curve:

```
Stress, Strain
```

Each curve must have at least two points defined

TABLE INPUT

Input:	Description
Curve Name:	The generic name of the non-linear bend stiffener stress-strain curve.
Stress:	A stress value for a point on the curve.
Strain:	The corresponding strain value.

NOTES

(a) The stress inputs are direct (longitudinal) stresses, and likewise direct strain. You do not specify a moment-curvature relation; rather Flexcom transforms the stress-strain data you specify here into moment-curvature curves for each element of the stiffener, based on the element properties.

(b) Use as many lines as you need to completely define a particular stress-strain curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent stress-strain curves, put the curve name in Column 1 and specify the stress-strain data in the same way.

(c) The points defining the non-linear stress-strain curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of strain.

(d) If the strain in the non-linear bend stiffener element lies between the specified stress-strain data points, Flexcom uses linear interpolation to determine the relevant stiffness of the element.

(e) If the strain in the non-linear bend stiffener element lies outside the specified range of the stress-strain curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

(f) If none of the specified strain terms have a negative value, the curve is assumed to be symmetrical about the origin (i.e. the behaviour of the bend stiffener is the same for both positive and negative strains).

1.8.10.46 *NO OPTIMISE

PURPOSE

To specify whether or not bandwidth optimisation is to be performed.

THEORY

Refer to Bandwidth Optimisation for further information on this feature.

KEYWORD FORMAT

This keyword does not contain any further data.

TABLE INPUT

Input:	Description
Suppress Bandwidth Optimisation:	The options are *No* (the default) and *Yes*.

NOTES

(a) Bandwidth optimisation is performed by default (for non-restart analyses), and is recommended for the vast majority of models. Occasionally, if your model is quite complex, the optimisation can take a long time. To quickly view the results from, for example an initial static analysis, you can use this option to suppress the optimisation. However, suppressing the optimisation is an unusual action.

1.8.10.47 *OCEAN

PURPOSE

To specify general parameters defining the ocean environment.

THEORY

Refer to Environmental Parameters for further information on this feature.

KEYWORD FORMAT

A single line with the following format:

Depth, Density, Acceleration due to Gravity [, *Kinematic Viscosity*]

The Kinematic Viscosity is used only when hydrodynamic coefficients are specified as a function of Reynolds Number. Otherwise this input can be omitted.

TABLE INPUT

Input:	Description
Water Depth:	The water depth.
Water Density:	The value of ρ_w, the mass density (mass per unit volume) of seawater. This defaults to a value of 1025 kg/m^3.
Acceleration due to Gravity:	The value of g, the acceleration due to gravity. This defaults to a value of 9.81m/s^2.
Water Kinematic Viscosity:	The value of ν, the kinematic viscosity of seawater. This defaults to a value of 1.3x10^{-6} m^2/s for a Metric unit system, 1.4x10^{-5} ft^2/s for an Imperial unit system, and 1.0x10^{-10} for a custom unit system. See Note (a).

NOTES

(a) The value of the kinematic viscosity ν is used only to determine the value of Reynolds number in the case where you have specified hydrodynamic coefficients as a function of Reynolds number. If you do not invoke this facility, the value of ν is immaterial and unused.

1.8.10.48 *PANEL,AUXILIARY

PURPOSE

To specify nodes which are connected by an auxiliary panel.

THEORY

Refer to Auxiliary Bodies for further information on this feature.

Note also that the old auxiliary body keywords (`*NODE,AUXILIARY`, `*ELEMENT,AUXILIARY`, `*AUXILIARY`, `*PANEL,AUXILIARY` and `*PANEL SECTIONS,AUXILIARY`) keywords have effectively been superseded by the integrated vessel/body features which provide a range of standard vessel and subsea component profiles. Refer to `*VESSEL,INTEGRATED` and `*BODY,INTEGRATED` for further information.

KEYWORD FORMAT

A single line which may be repeated as often as necessary:

`Panel Number, First Node, Second Node, Third Node`

TABLE INPUT

Input:	Description
Panel Number:	The number of the auxiliary panel being defined.
First Node:	The first node of the panel.
Second Node:	The second node of the panel.
Third Node:	The third node of the panel.

NOTES

(a) The order in which you define panel nodes is important from the point of view of defining the front and back face of the panel. Flexcom assumes that the order of the nodes is clockwise, and determines the front face normal based on this assumption. In the dynamic display, the front face of a panel reflects light (that is, appears bright), whereas the back face reflects no light (appears very dark). This is worth bearing in mind in order, for example, to produce a realistic display of a vessel.

(b) You must assign a unique panel number to each panel in the model. However, the panel numbers do not need to be entered in any numerical order.

(c) It is not necessary (but it is possible) to define auxiliary elements where an auxiliary panel has been specified.

(d) If it is required that a panel can be viewed from both sides, you must define two panels; one defined in reverse order to the other. For example, Panel 1 {Node 1, Node 2, Node 3} and Panel 2 {Node 3, Node 2, Node 1}.

1.8.10.49 *PANEL SECTIONS,AUXILIARY

PURPOSE

To specify the auxiliary panels that make up an auxiliary body.

THEORY

Refer to Auxiliary Bodies for further information on this feature.

Note also that the old auxiliary body keywords (`*NODE,AUXILIARY`, `*ELEMENT,AUXILIARY`, `*AUXILIARY`, `*PANEL,AUXILIARY` and `*PANEL SECTIONS,AUXILIARY`) keywords have effectively been superseded by the integrated vessel/body features which provide a range of standard vessel and subsea component profiles. Refer to `*VESSEL,INTEGRATED` and `*BODY,INTEGRATED` for further information.

KEYWORD FORMAT

A block of lines which define a group of auxiliary panels repeated as often as necessary. The block of lines starts with a line defining the body name and the panel colour. It is followed by either of two types of line defining the auxiliary panels, which can be mixed and repeated as often as necessary.

Line defining the body name and panel colour:

`BODY=`*Vessel Name/Node Number* `[, COLOUR=`*`Colour`*`]`

Two types of lines for defining auxiliary panels:

`List of Panels`

or

`GEN=`*`Start Panel, End Panel`* `[,` *`Panel Increment`*`]`

The list of panels can contain up to 20 panel numbers. *`Panel Increment`* defaults to 1. *`Vessel Name/Node Number`* can be NONE if no motions are to be applied to the auxiliary body. *`Colour`* is entered as an RGB value, either as a hexadecimal number (prefixed by #) or a decimal number. The default colour is silver.

TABLE INPUT

Input:	Description
Body:	This entry allows you to associate the auxiliary body with a vessel (in which case you specify the vessel name) or a structural node (in which case you specify the node number).
Panels:	The auxiliary panels that comprise the auxiliary panel body.
Panel Colour:	The colour of the panel entered as an RGB value, either as a hexadecimal number (prefixed by #) or a decimal number. This input is optional. The default colour is silver. See Note (c).

NOTES

(a) This table allows you to group together auxiliary panels into named *auxiliary panel bodies*. You may also associate the auxiliary body with a vessel or a node of the finite element discretisation data. If this option is invoked, the auxiliary panels translate and rotate with the vessel/node during the course of an analysis or series of analyses (static and dynamic). When viewed in the dynamic display, the motions of the auxiliary body will be clearly visible.

(b) You do not need to explicitly assign all auxiliary panels to auxiliary panel bodies. An auxiliary panel not assigned to a panel body remains stationary throughout all subsequent analysis phases. For example, you might want to visually check for interference between a riser or hose and a nominally fixed structure.

(c) The optional *Panel Colour* column enables you to assign different colours to different parts of the auxiliary body. By default, an auxiliary body is silver throughout. The RGB value refers to the Red, Green and Blue colour model that is widely used in computer graphics. The table below shows 16 standard RGB values in hexadecimal and decimal formats. These values are based on the four numbers: #00, #80, #C0, #FF (0, 128, 192, 255), which are then concatenated in the hexadecimal format to build a total value.

Colour	RGB Value (Hexadecimal)	RGB Value (Decimal)	Colour	RGB Value (Hexadecimal)	RGB Value (Decimal)
Black	#000000	0	Blue	#0000FF	255
Cyan	#00FFFF	65,535	Gray	#808080	8,421,504
Green	#00FF00	65,280	Lime	#008000	32,768
Magenta	#FF00FF	16,711,935	Maroon	#800000	524,288

Navy	#000080	128	Olive	#808000	8,421,376
Purple	#800080	8,388,736	Red	#FF0000	16,711,680
Silver	#C0C0C0	12,632,256	Teal	#008080	32,896
White	#FFFFFF	16,777,215	Yellow	#FFFF00	16,776,960

1.8.10.50 *PIP CONNECTION

PURPOSE

To define pipe-in-pipe connections between nodes of the finite element model.

THEORY

Refer to Standard Connections and Sliding Connections for further information on this feature.

KEYWORD FORMAT

The keyword begins with a number of optional lines to specify miscellaneous parameters. This is followed by various types of line that may be mixed and repeated as often as necessary.

Optional line to specify whether the orientation of the pipe-in-pipe connections is based on the instantaneous positions of the connected nodes or their initial positions.

```
[ORIENTATION=Orientation]
```

Optional line to specify that bandwidth optimisation is to be performed every time a pipe-in-pipe connection changes due to relative motion of the connected pipes in the axial direction (sliding).

```
[BANDWIDTH=UPDATED]
```

Optional line to monitor for occurrences of relative penetration at pipe-in-pipe connections. A warning is issued during the analysis if penetration exceeds a user-specified threshold.

```
[PENETRATION_TOLERANCE=Penetration Tolerance]
```

Line defining an explicit nodal connection:

```
Primary Node (Node or Label), Secondary Node (Node or Label), STIFFNESS=Stifi
```

Line defining multiple nodal connections:

```
GEN=Primary Start Node (Node or Label), Primary End Node (Node or Label)
[, Increment], GEN=Secondary Start Node (Node or Label), Secondary End Node
STIFFNESS=Stiffness or CURVE=Curve Name
```

Line defining an explicit nodal connection, capable of modelling sliding contact:

```
Primary Node (Node or Label), SET=Secondary Element Set,
STIFFNESS=Stiffness or CURVE=Curve Name [, AXIAL_STIFF=Axial Stiffness or AXI
```

Line defining multiple nodal connections, capable of modelling sliding contact:

```
GEN=Primary Start Node (Node or Label), Primary End Node (Node or Label)
[, Increment], SET=Secondary Element Set, STIFFNESS=Stiffness or CURVE=Curve
[, AXIAL_STIFF=Axial Stiffness or AXIAL_NONLINEAR=Axial Curve Name]
```

Orientation may be either INSTANTANEOUS (the default) or INITIAL. If you specify a node label rather than a node number, it must be enclosed in {} brackets. A `Stiffness` is specified when the connection between the pipe-in-pipe sections is linear, whereas a `Curve Name` is specified when the connection is non-linear. Resistance to relative axial motion may be characterised by a linear `Axial Stiffness` or a non-linear `Axial Curve Name`. If a curve name is specified, then the curve must be defined using *PIP STIFFNESS.

`Increment` defaults to 1.

MISCELLANEOUS OPTIONS

PURPOSE

To define miscellaneous options relating to pipe-in-pipe connection modelling.

TABLE INPUT

Input:	Description
Bandwidth:	This option allows you to request that bandwidth optimisation be re-performed every time a changeover occurs between connected nodes in a sliding connection. The options are *No* (the default) and *Yes*.
Orientation:	This option allows you to specify that the orientation of connections be based on the *Instantaneous* (the default) or *Initial* positions of the connected nodes.
Penetration Tolerance:	This option allows you to check if any penetration occurs between the connected pipes during the analysis. Should this happen, the program will issue a warning regarding the relevant connection and time of occurrence. The default value for penetration tolerance is zero. It is also possible to suppress such warnings warnings. See Note (a).

NOTES

(a) If you wish to check contact based on the actual pipe diameters, you should specify a penetration tolerance of zero. Given the numerical model of the pipe-in-pipe connection, power-law curves for example, you may wish to tolerate small levels of penetration, in which case you should specify a small, positive value for the penetration tolerance. If you wish to suppress the warning messages altogether, you should specify an arbitrarily large tolerance. The tolerance parameter may also be used to provide an indication of the connected pipes approaching contact, via the specification of a small, negative tolerance value.

STANDARD PIP CONNECTIONS

PURPOSE

To define standard pipe-in-pipe connections between individual nodes of the finite element model.

TABLE INPUT

Input:	Description
Primary Node:	A node (number or label) on the primary pipe. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Secondary Node:	A node (number or label) on the secondary pipe. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Stiffness:	The linear stiffness of the connection between these nodes or the name of a non-linear force-deflection curve that defines this stiffness.

NOTES

(b) Refer to Standard Connections for further information on the significance of the inputs supplied here.

(c) If a non-linear force-deflection curve is specified for the connection stiffness, then the actual curve must be defined in the *Pipe-in-Pipe Stiffness - Data Pairs* or *Pipe-in-Pipe Stiffness - Power Law* tables.

STANDARD PIP CONNECTIONS - GENERATE

PURPOSE

To generate standard pipe-in-pipe connections between groups of nodes of the finite element model.

TABLE INPUT

Input:	Description
Primary Start Node:	The first node (number or label) in the group of nodes on the primary pipe. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Primary End Node:	The last node (number or label) in the group of nodes on the primary pipe. If you specify a node label rather than a node number, it must be enclosed in {} brackets.

Primary Increment:	The node number increment to be used when generating nodes on the primary pipe which are to be connected to corresponding nodes on the secondary pipe. This input defaults to a value of 1.
Secondary Start Node:	The first node (number or label) in the group of nodes on the secondary pipe. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Secondary End Node:	The last node (number or label) in the group of nodes on the secondary pipe. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Secondary Increment:	The node number increment to be used when generating nodes on the secondary pipe which are to be connected to corresponding nodes on the primary pipe. This input defaults to a value of 1.
Stiffness:	The linear stiffness of the generated connections or the name of a non-linear force-deflection curve that defines this stiffness.

NOTES

(a) Refer to Standard Connections for further information on the significance of the inputs supplied here.

(b) If a non-linear force-deflection curve is specified for the connection stiffness, then the actual curve must be defined in the *Pipe-in-Pipe Stiffness - Data Pairs* or *Pipe-in-Pipe Stiffness - Power Law* tables.

SLIDING PIP CONNECTIONS

PURPOSE

To define sliding pipe-in-pipe connections between individual nodes of the finite element model.

TABLE INPUT

Input:	Description
Primary Node:	A node (number or label) on the primary pipe. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Secondary Element Set:	A set of elements on the secondary pipe. The primary node will be connected to the nearest (in the axial direction) secondary node at all times.
Lateral Stiffness:	The linear stiffness of the connection between these nodes in the lateral direction or the name of a non-linear force-deflection curve that defines this stiffness.
Axial Stiffness:	The linear stiffness of the connection between these nodes in the axial direction or the name of a non-linear force-deflection curve that defines this stiffness.

NOTES

(a) Refer to Sliding Connections for further information on the significance of the inputs supplied here.

(b) If a non-linear force-deflection curve is specified for the connection stiffness, then the actual curve must be defined in the *Pipe-in-Pipe Stiffness - Data Pairs* or *Pipe-in-Pipe Stiffness - Power Law* tables.

SLIDING PIP CONNECTIONS - GENERATE

PURPOSE

To generate sliding pipe-in-pipe connections between groups of nodes of the finite element model.

TABLE INPUT

Input:	Description

Primary Start Node:	The first node (number or label) in the group of nodes on the primary pipe. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Primary End Node:	The last node (number or label) in the group of nodes on the primary pipe. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Primary Increment:	The node number increment to be used when generating nodes on the primary pipe which are to be connected to corresponding nodes on the secondary pipe. This input defaults to a value of 1.
Secondary Element Set:	A set of elements on the secondary pipe. Each primary node will be connected to the nearest (in the axial direction) secondary node at all times.
Lateral Stiffness:	The linear stiffness of the generated connections in the lateral direction or the name of a non-linear force-deflection curve that defines this stiffness.
Axial Stiffness:	The linear stiffness of the generated connections in the axial direction or the name of a non-linear force-deflection curve that defines this stiffness.

NOTES

(a) Refer to Sliding Connections for further information on the significance of the inputs supplied here

(b) If a non-linear force-deflection curve is specified for the connection stiffness, then the actual curve must be defined in the *Pipe-in-Pipe Stiffness - Data Pairs* or *Pipe-in-Pipe Stiffness - Power Law* tables.

1.8.10.51 *PIP SECTION

PURPOSE

To define internal and external pipe sections when part of a pipe model is contained within another.

THEORY

Refer to Pipe-in-Pipe Sections for further information on this feature.

KEYWORD FORMAT

Optional line which provides advanced modelling options regarding drag and inertia loads on inner pipe-in-pipe sections:

```
INNER_HYDRO=Hydrodynamic Solution, AUTO_CREATE=Auto-Create Connections
```

One line that is then repeated as often as necessary to specify all pipe-in-pipe sections:

```
OUTER=Outer Element Set Name, INNER=Inner Element Set Name
```

The inner and outer element sets must be defined under `*ELEMENT SETS`. `INNER_HYDRO` may be either `LHS` (the default) or `RHS`. `AUTO_CREATE` may be either `YES` (the default) or `NO`. These options are highly specialised and should only be explored by power users of the software. The default options are recommended by Wood. Refer to Drag and Inertia on Inner Pipe Sections for further information.

TABLE INPUT

Pipe-in-Pipe Sections

Input:	Description
Outer Element Set:	The element set which represents the external or outer pipe section in the pipe-in-pipe model.
Inner Element Set:	The element set which represents the internal or inner pipe section in the pipe-in-pipe model.

Drag and Inertia on Inner Pipe Sections

Input:	Description

Hydrodynamic Solution:	This options allows you to specify whether drag forces and hydrodynamic inertia on inner pipe-in-pipe sections are modelled as either: • Mass and damping terms on the left hand side of the equations of motion (capturing the required coupling between the outer node's velocity/acceleration and the inner node loading), the default, or... • Force terms on the right hand side of the equations of motion (for compatibility with earlier versions, but theoretically incorrect).
Auto-Create Connections:	This options allows you to specify whether: • Flexcom should automatically insert token connections of zero stiffness where required to ensure that hydrodynamic loading on all inner nodes is modelled, the default, or... • The solution should proceed using only the pipe-in-pipe connections which you have defined explicitly (computationally more efficient in some cases, but there may be a possibility that not all inner nodes will experience hydrodynamic loading at all times during the simulation).

NOTES

(a) A particular *Outer Element Set* can be associated with more than one *Inner Element Set*, for example in the modelling of bundled riser systems. But for obvious reasons a particular *Inner Element Set* can be internal to only one *Outer Element Set*.

(b) A particular element set which is the *Inner Element Set* on one row of the *Pipe-in-Pipe Sections* table can be an *Outer Element Set* on another row, as for example in the analysis of a multi-tube production riser.

(c) The use of the *Pipe-in-Pipe Sections* table is not mandatory when defining a pipe-in-pipe model. For example, in the analysis of piggy-backed systems, it is valid to specify pipe-in-pipe connections without defining internal and external pipe-in-pipe sections.

(d) The *Hydrodynamic Solution* and *Auto-Create Connections* options are highly specialised and should only be explored by power users of the software. The default options are recommended by Wood. Refer to Drag and Inertia on Inner Pipe Sections for further information.

1.8.10.52 *PIP STIFFNESS

PURPOSE

To define force-deflection curves for non-linear pipe-in-pipe connection stiffnesses.

THEORY

Refer to Non-Linear Power-Law Connections for further information on this feature.

KEYWORD FORMAT

A block of lines that define a force-deflection curve repeated as often as necessary. There are two possible types of force-deflection curves: explicit pairs of force-deflection values and power-law curves. The keyword formats are slightly different in each case, so each one is described separately.

Explicit Data Pairs

The first line defines the curve name, and optionally the curve type. This is followed by as many lines as necessary to define each point on the curve.

```
CURVE=Curve Name [, TYPE=DATA PAIRS]
Deflection, Force
```

Each force-deflection curve must contain at least two points, and deflection values should not be negative.

Power Law

The first line defines the curve name, the curve type, and the configuration in which is curve is to be used. The second line contains the maximum contact force, the exponent.

```
CURVE=Curve Name, TYPE=POWER LAW [, CONFIGURATION=Configuration Type]
Maximum Contact Force [, Exponent]
```

Configuration Type may be PIP (default) or POP. Exponent defaults to 5.

PIPE-IN-PIPE STIFFNESS - DATA PAIRS

Purpose

To define force-deflection curves (using explicit data pairs) for non-linear pipe-in-pipe connection stiffnesses.

Table Input

Input:	Description
Curve Name:	The generic name of the force-deflection curve.
Deflection:	A deflection value for a point on the curve.
Force:	The corresponding force value.

Notes

(a) Use as many lines as you need to completely define a particular force-deflection curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent force-deflection curves, put the curve name in Column 1 and specify the force-deflection data in the same way.

(b) The points defining the non-linear force-deflection curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order.

(c) If the deflection in the non-linear connection lies between the specified force-deflection data points, Flexcom uses linear interpolation to determine the relevant stiffness of the connection.

(d) If the deflection in the non-linear connection lies outside the specified range of the force-deflection curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

(e) The nonlinear stiffness of a connection does not depend on the direction of relative displacement of the connected nodes, because of the inherent symmetry of a pipe cross-section. Mathematically, this means the nonlinear stiffness curves should be symmetric about the origin, so you should not specify negative displacement values.

PIPE-IN-PIPE STIFFNESS - POWER LAW

Purpose

To define force-deflection curves (using a power-law approach) for non-linear pipe-in-pipe connection stiffnesses.

Table Input

Input:	Description
Curve Name:	The generic name of the force-deflection curve.
Maximum Contact Force:	The maximum contact force in the lateral direction provided by the connection.
Exponent:	The exponent to be used in the power-law approximation. This entry is optional and defaults to a value of 5 if omitted.
Configuration:	Whether the power-law is to be used in a pipe-in-pipe (default) or pipe-on-pipe type configuration.

Notes

(a) Refer to Non-Linear Power-Law Connections for further information on this feature.:

1.8.10.53 *PLASTIC HARDENING

PURPOSE

To define hardening models for plastic materials.

THEORY

Refer to Linear Elastic with Plastic Hardening for further information on this feature.

KEYWORD FORMAT

A block of lines that defines a plastic hardening model, repeated as often as necessary. The block begins with a line defining the model name and (optionally) a tension variation. It is followed by as many lines as necessary to define each point in the plastic hardening model.

Line defining the model name and tension variation:

```
NAME=Model Name [, TENSION VARIATION=Tension Variation]
```

Line defining a point in the plastic hardening model:

```
Equivalent Stress, Equivalent Plastic Strain
```

Each model must have at least two points defined. Plastic hardening models may only be associated with non-linear beam elements which are defined using the Rigid Riser Format for geometric properties specification.

TABLE INPUT

Input:	Description
Model Name:	The name of the plastic hardening model.
Equivalent Stress:	An equivalent stress value for a point in the model.
Equivalent Plastic Strain:	The corresponding equivalent plastic strain value.
Tension Variation:	An optional value defining the tension variation threshold.

NOTES

(a) This keyword is used to define plastic hardening models that can be associated with a particular set of elements. These models may be assigned to element sets using the `*GEOMETRIC SETS` keyword.

(b) Use as many lines as you need to completely define a particular force-strain curve. Simply leave Columns 1 and 4 blank for second and subsequent lines. For subsequent models, put the model name in Column 1 and specify the equivalent stress and the equivalent plastic strain data in the same way.

(c) The first point in the model defines the yield point. As a result, it is expected that the value for the *Equivalent Plastic Strain* be zero.

(d) If the strain in an element lies between the force-strain data points you specify, Flexcom uses linear interpolation to determine the relevant stiffness for the element.

(e) If the strain in the element lies outside the specified range of the force-strain curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

(f) None of the specified strain terms can have a negative value.

(g) The tension variation value defines the threshold above or below which the plastic hardening model is integrated to determine the remaining bending moment capacity, based on the axial force at a particular location. A value of zero means the integration is performed as soon as a variation is registered. If the tension variation value is omitted, then the plastic hardening model is integrated only once at the beginning of the analysis.

1.8.10.54 *POINT BUOY

PURPOSE

To define point buoys and their associated hydrodynamic properties.

THEORY

Refer to Point Masses and Point Buoys for further information on this feature.

KEYWORD FORMAT

A block of lines used to define a point buoy and its properties, repeated as often as necessary. The block begins with an optional line indicating that rotational terms are being specified. The next line contains the node number, buoyancy, weight, and (optionally) rotational inertia terms and a local axis system. If any of the rotational terms are specified, 6 more lines are required to complete the block. Otherwise 3 more lines are expected.

```
[OPTION=6D_BUOY]
Node (Number or Label), Total Buoyancy, Total Weight [, Ixx] [, Iyy] [, Izz]
```

```
X:CdAd, X:CmVin, X:CaVin
Y:CdAd, Y:CmVin, Y:CaVin
Z:CdAd, Z:CmVin, Z:CaVin
[X:CdDMA, X:CmHI, X:CaHI]
[Y:CdDMA, Y:CmHI, Y:CaHI]
[Z:CdDMA, Z:CmHI, Z:CaHI]
```

You must use the `*LOCAL AXIS SYSTEM` keyword to define any local axis system you reference here. If you specify a node label rather than a node number, it must be enclosed in {} brackets.

POINT BUOYS - 3D

Purpose

To define a 3D point buoy.

Table Input

Input:	Description
Node:	The node (number or label) at which the point buoy is to be positioned. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Total Buoyancy:	The total buoyancy provided by the buoy. See Note (b).
Total Weight:	The total weight (in air) of the buoy. See Note (b).
Buoy ID (Translational):	An identifier or name for the buoy, to be used again when specifying the buoy translational hydrodynamic properties using the Buoy *Properties – Translational (3D)* table.

NOTES

(a) This facility is available to include the effect of a subsea buoy in a model. It is intended as an alternative to modelling the buoy explicitly using beam elements. Using this facility also allows you to specify the buoy hydrodynamic properties independently for the different directions.

(b) The net buoyancy provided by the buoy is naturally the difference between the *Total Buoyancy* and the *Total Weight* values you specify, while the *Total Weight* also defines the inertia contribution of the buoy.

BUOY PROPERTIES - TRANSLATIONAL (3D)

PURPOSE

To specify translational hydrodynamic properties for 3D point buoys.

TABLE INPUT

Input:	Description
Buoy ID:	The name or ID of the buoy whose translational hydrodynamic properties are being defined. This is also used when defining the buoy itself using Point *Buoys – 3D*.
X: CdAd:	The product of the buoy frontal area and drag coefficient in the global X direction.
X: CmVin:	The product of the buoy reference volume and inertia coefficient in the global X direction.
X: CaVin:	The product of the buoy reference volume and added mass coefficient in the global X direction.
Y: CdAd:	The product of the buoy frontal area and drag coefficient in the global Y direction.
Y: CmVin:	The product of the buoy reference volume and inertia coefficient in the global Y direction.
Y: CaVin:	The product of the buoy reference volume and added mass coefficient in the global Y direction.
Z: CdAd:	The product of the buoy frontal area and drag coefficient in the global Z direction.

Z: CmVin:	The product of the buoy reference volume and inertia coefficient in the global Z direction.
Z: CaVin:	The product of the buoy reference volume and added mass coefficient in the global Z direction.

POINT BUOYS - 6D

PURPOSE

To define a 6D point buoy.

TABLE INPUT

Input:	Description
Node:	The node (number or label) at which the point buoy is to be positioned. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
Total Buoyancy:	The total buoyancy provided by the buoy. See Note (b).
Total Weight:	The total weight (in air) of the buoy. See Note (b).
Buoy ID (Translational):	An identifier or name for the buoy, to be used again when specifying the buoy translational hydrodynamic properties using the *Buoy Properties – Translational (6D)* table.
Buoy ID (Rotational):	An identifier or name for the buoy, to be used again when specifying the buoy rotational hydrodynamic properties using the *Buoy Properties – Rotational (6D)* table.
Axis:	The name of a local axis system to be used in the computation of rotational hydrodynamic loading on the buoy. This entry is optional and the global axis system is used by default.

Ixx:	The rotational inertia about the local x-axis for the buoy. This entry is optional and defaults to zero if omitted.
Iyy:	The rotational inertia about the local y-axis for the buoy. This entry is optional and defaults to zero if omitted.
Izz:	The rotational inertia about the local z-axis for the buoy. This entry is optional and defaults to zero if omitted.

NOTES

(a) This facility is available to include the effect of a subsea buoy in a model. It is intended as an alternative to modelling the buoy explicitly using beam elements. Using this facility also allows you to specify the buoy hydrodynamic properties independently for the different directions.

(b) The net buoyancy provided by the buoy is naturally the difference between the *Total Buoyancy* and the *Total Weight* values you specify, while the *Total Weight* also defines the inertia contribution of the buoy.

(c) If a local axis system referenced, it must be defined using the *Local Axes* table.

BUOY PROPERTIES - TRANSLATIONAL (6D)

PURPOSE

To specify translational hydrodynamic properties for 6D point buoys.

TABLE INPUT

Input:	Description
Buoy ID:	The name or ID of the buoy whose translational hydrodynamic properties are being defined. This is also used when defining the buoy itself using Point Buoys – 6D.
X: CdAd:	The product of the buoy frontal area and drag coefficient in the global X direction.

X: CmVin:	The product of the buoy reference volume and inertia coefficient in the global X direction.
X: CaVin:	The product of the buoy reference volume and added mass coefficient in the global X direction.
Y: CdAd:	The product of the buoy frontal area and drag coefficient in the global Y direction.
Y: CmVin:	The product of the buoy reference volume and inertia coefficient in the global Y direction.
Y: CaVin:	The product of the buoy reference volume and added mass coefficient in the global Y direction.
Z: CdAd:	The product of the buoy frontal area and drag coefficient in the global Z direction.
Z: CmVin:	The product of the buoy reference volume and inertia coefficient in the global Z direction.
Z: CaVin:	The product of the buoy reference volume and added mass coefficient in the global Z direction.

BUOY PROPERTIES - ROTATIONAL (6D)

PURPOSE

To specify rotational hydrodynamic properties for 6D point buoys.

TABLE INPUT

Input:	Description
Buoy ID:	The name or ID of the buoy whose rotational hydrodynamic properties are being defined. This is also used when defining the buoy itself using Point Buoys – 6D.

X: CdDMA:	The product of the drag coefficient and the drag moment of area in the X direction.
X: CmHI:	The product of the inertia coefficient and the hydrodynamic inertia in the X direction.
X: CaHI:	The product of the added mass coefficient and the hydrodynamic inertia in the X direction.
Y: CdDMA:	The product of the drag coefficient and the drag moment of area in the Y direction.
Y: CmHI:	The product of the inertia coefficient and the hydrodynamic inertia in the Y direction.
Y: CaHI:	The product of the added mass coefficient and the hydrodynamic inertia in the Y direction.
Z: CdDMA:	The product of the drag coefficient and the drag moment of area in the Z direction.
Z: CmHI:	The product of the inertia coefficient and the hydrodynamic inertia in the Z direction.
Z: CaHI:	The product of the added mass coefficient and the hydrodynamic inertia in the Z direction.

1.8.10.55 *POISSON

PURPOSE

To include Poisson's ratio effects in an analysis.

THEORY

Refer to Poissons Ratio for further information on this feature.

KEYWORD FORMAT

A block of two lines which define the Poisson’s Ratio for all or selected elements (in terms of element sets). If values are defined for individual elements sets, the block of lines can be repeated as often as necessary.

```
SET=Set Name
Poisson's Ratio
```

TABLE INPUT

Input:	Description
Element Set:	The element set to which Poisson’s ratio applies. This defaults to *All*, to indicate all elements.
Poisson's Ratio:	The value of Poisson's ratio. This defaults to a value of 0.

NOTES

(a) By default, Poisson's ratio effects are not included in the analysis. If included, axial changes due to Poisson's ratio and internal and external pressure are calculated.

1.8.10.56 *PROPERTIES

PURPOSE

To assign effective structural properties to element sets for use in calculating stresses.

THEORY

Refer to Stress Properties for further information on this feature.

KEYWORD FORMAT

A block of two lines repeated as often as necessary. Each property specified on the second line can be omitted, in which case a default value will be used for stress calculations:

```
SET=Set Name
Do, Di, A, Iyy, Izz, Tmin
```

The properties specified here are used in calculating timetraces of stresses during an analysis, for use in subsequent Timetrace Postprocessing or Time Domain Fatigue Analysis. This keyword is optional and may be ignored if requested timetrace output does not include stresses, or if you wish to use the actual properties in the computation of stress timetraces. A similar keyword may be included in the Database Postprocessing keyword inputs, and if present, those properties will be used to compute stresses during database postprocessing. However, if the keyword is not included in the database postprocessing keyword file, the values specified here are carried through to the database postprocessor.

If any of the parameters *A, Iyy, Izz* or *Tmin* are omitted, default values are computed based on the specified (or default) values of *Do* and *Di*. The minimum wall thickness (*Tmin*) is only used in the alternative computation of von Mises stress, according to API-2RD.

TABLE INPUT

Input:	Description
Set Name:	The name of the element set to which the stress properties are to be assigned. This defaults to all elements.
Do:	The effective outside diameter for the elements of the set. The default depends on the format used to specify geometric data. If you used the Flexible Riser Format, then the default is the drag diameter for the elements of the set. If you used the Rigid Riser Format or Mooring Line Format, then Do here defaults to Do input in the properties data. See Note (a).
Di:	The effective internal diameter for the elements of the set. The default is the internal diameter specified in the geometric data. See Note (b).
A:	The effective cross-sectional area for the elements of the set. The default is given by: $A = \dfrac{\pi\left(D_o^2 - D_i^2\right)}{4}$

	where Do and Di are the inputs described above, default or otherwise. See Note (a).
Iyy:	The second moment of area about the local y-axis for the elements of the set. The default is given by: $$I_{yy} = \frac{\pi\left(D_o^4 - D_i^4\right)}{64}$$ where Do and Di are the inputs described above, default or otherwise. See Note (a).
Izz:	The second moment of area about the local z axis for the elements of the set. The default is Iyy above.
Min. Wall Thickness :	The minimum wall thickness for the elements of the set. The default is given by: $$T_{min} = \frac{\left(D_o - D_i\right)}{2}$$ where Do and Di are the inputs described above, default or otherwise. See Note (e).

NOTES

(a) The purpose of this table is to input values to be used in calculating timetraces of stresses during an analysis, for use in subsequent Timetrace Postprocessing or Time Domain Fatigue Analysis. Among the parameters you can specify for timetrace output are hoop, bending, axial and von Mises stresses. To calculate these, the program requires inputs whose effective values may be different when used for calculating stresses than they are when calculating, for example, hydrostatic and hydrodynamic forces. For example, to calculate bending stress at the outer surface of the cross section, the effective outer diameter is required, and this can in general be different from either the drag or buoyancy diameters specified as part of the geometric data in Flexible Riser Format.

(b) You will recall that the internal diameter you specify is used only in calculating the effects of internal fluid when you use the *Flexible Riser* format. Therefore, if your structure has no internal fluid present, you can specify an internal diameter of zero as part of the geometric data. However, an effective internal diameter is required to calculate hoop and von Mises stresses, which is why an internal diameter option is provided in this table.

(c) Timetrace output from Flexcom is optional, and therefore so is this table. This table can also be ignored if you request timetrace output which does not include stresses (other outputs available include axial force, effective tension, bending moments, torques and shear forces).

(d) The other type of output produced by Flexcom is database output. Flexcom does not output stresses to the results database. If stresses are required, they are calculated by the Database Postprocessor. For this reason, the *Database Postprocessor* provides a table similar to this one to specify similar inputs. Therefore, if you propose to recover stresses from database output, you do not need to use this table. You may do so if you wish however, and any values specified here will carry through to the *Database Postprocessor* and be used there.

(e) The minimum wall thickness is only used in the computation of von Mises stress according to API-2RD. Refer to Von Mises Stress (API-2RD Method) for further details.

1.8.10.57 *P-Y

PURPOSE

To define P-y curves for modelling soil structure interaction.

THEORY

Refer to Soil Modelling for further information on this feature.

Soil structure interaction can be modelled explicitly by manually specifying P-y curves at different depths, or by utilising one of the three built-in soil models which can generate P-y curves automatically. The three pre-programmed soil models are:

- Sand (O'Neill & Murchinson, 1983)

- Soft Clay (Matlock, 1970)
- Stiff Clay (Reese et al., 1975)

KEYWORD OVERVIEW

The keyword format is a block of lines defining the interaction between a particular structure and the soil. The first line is an optional one which specifies the format in which the properties are input. This is followed by a number of lines defining the actual data. As many blocks as required can be used to specify the soil structure interaction over the complete depth required.

User Format

Data may defined explicitly within the keyword file itself, or defined separately in an external file which is simply referenced using a FILE= entry. The latter approach is highly recommended as it reduces unnecessary clutter in the keyword file, and also helps to improve user interface performance and enhanced user experience.

The format of the block when using directly entered P-y curve data:

```
[TYPE=USER]
SET=Set Name
DEPTH=Depth
Soil Resistance (P), Deflection (y)
```

The first line is optional. The second line indicates the element set for which the subsequent P-y curves apply. The next line defines the depth to which the curve relates and marks the start of a block containing the P-y curve data at that depth. This block can be repeated at as many depths as required. The final line is a pair of P, y values. This line will be repeated several times for describe a P-y curve.

The format of the block when using P-y curve data from an external file:

```
[TYPE=USER]
SET=Set Name
FILE=File Name
```

The first line is optional. The second line indicates the element set for which the P-y curves apply. The third line specifies an external file containing P-y curve data.

Sand Model

The format of the block when using the built-in soil model for sand:

```
TYPE=SAND
```

```
SET=Set Name
DEPTH=Start Depth, End Depth [, Depth Increment]
Diameter [, Water Table Depth]
[SCOUR=Scour Depth]
Depth, Submerged Unit Weight, Angle of Internal Friction, [C1], [C2], [C3],
```

The first line is required and indicates that this block describes the properties of for generating P-y curves based on the standard sand model. P-y curves apply. The second line indicates the element set for which the generated P-y curves apply. The third line indicates the depths at which P-y curves will be generated. The next line defines the nominal element diameter to use when generating the P-y curves and the water table depth, if appropriate. The next line allows the scour depth to be specified, when desired. The final line, which can be repeated as many times as required, allows parameters for the sand to be specified at specific depths.

Soft Clay

The format of the block when using the built-in soil model for soft clay:

```
TYPE=SOFT CLAY
SET=Set Name
DEPTH=Start Depth, End Depth [, Depth Increment]
[LOADING=STATIC/CYCLIC]
Diameter, J
[SCOUR=Scour Depth]
Depth, Undrained Shear Strength, Submerged Unit Weight [, Strain at Half Max
```

The first line is required and indicates that this block describes the properties of for generating P-y curves based on the standard soft clay model. P-y curves apply. The second line indicates the element set for which the generated P-y curves apply. The third line indicates the depths at which P-y curves will be generated. The next line indicates whether or not P-y curves for a static or cyclical loading should be generated; the default is static. The next line defines the nominal element diameter to use when generating the P-y curves and the empirical constant J for soft clays. The next line allows the scour depth to be specified, when desired. The final line, which can be repeated as many times as required, allows parameters for the soft clay to be specified at specific depths.

Stiff Clay Model

The format of the block when using the built-in soil model for stiff clay:

```
TYPE=STIFF CLAY
SET=Set Name
DEPTH=Start Depth, End Depth [, Depth Increment]
[LOADING=STATIC/CYCLIC]
Diameter [, Ks]
[SCOUR=Scour Depth]
```

Depth, Undrained Shear Strength, Submerged Unit Weight, [A], [*Strain at Half*

The first line is required and indicates that this block describes the properties of for generating P-y curves based on the standard stiff clay model. P-y curves apply. The second line indicates the element set for which the generated P-y curves apply. The third line indicates the depths at which P-y curves will be generated. The next line indicates whether or not P-y curves for a static or cyclical loading should be generated; the default is static. The next line defines the nominal element diameter to use when generating the P-y curves and the optional empirical constant Ks for stiff clays. The next line allows the scour depth to be specified, when desired. The final line, which can be repeated as many times as required, allows parameters for the soft clay to be specified at specific depths.

TABLE INPUTS

P-y Curves - Explicit

Input:	Description
Set Name:	The name of the element set for which the P-y curve applies.
Depth:	The depth below the mudline at which the P-y curve is being defined.
Soil Resistance, P:	A soil resistance (P) value for a point on the P-y curve. This has units of force/length.
Deflection, y:	The corresponding deflection (y) value.

P-y Curves - File

Input:	Description
Set Name:	The name of the element set for which the P-y curve applies.
File Name:	The file containing the P-y curve data. This data includes depth and P-y data.

Notes

(a) Use as many lines as you need to completely define a particular P-y curve, for a particular combination of element set and depth. Simply leave the *Set Name* and *Depth* columns blank for second and subsequent lines of the curve data. For subsequent P-y curves, specify a new *Set Name* and/or *Depth*, and specify the P-y data in the same way.

(b) The points defining the P-y curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of deflection.

SAND MODEL

P-y Curves – Sand – General

Input:	Description
Set Name:	The name of the element set for which the P-y curve applies.
Start Depth:	The depth below the mudline at which the first P-y curve is will be generated.
End Depth:	The depth below the mudline at which the last P-y curve is will be generated.
Depth Increment:	The increment in depth, between the requested start and end depths, at which P-y curves will be generated.
Representative Diameter:	The diameter of the structure in the depth region for which P-y curves are being generated. Where this varies, either use multiple blocks with different diameters, or choose a single representative diameter value.
Water Table Depth:	The depth of the water table below the mudline. This entry is optional. This entry only influences the program's calculation of the initial modulus of subgrade reaction, K. If omitted, the program assumes that all heights are above the watertable.

Scour Depth:	Specify a length of the scour region. The program will adjust the generated P-y curves based on this input. No P-y curves will be generated above the scour level and P-y curves below this level will be adjusted to use the adjusted distance below mudline.

P-y Curves – Sand – By Depth

Input:	Description
Depth:	The distance below the mudline.
Submerged Unit Weight:	The soil submerged unit weight.
Angle of Internal Friction:	The angle of internal friction.
C1:	An empirical constant, *C1*. This entry is optional. If omitted, the program supplies an appropriate value from tables provided by O'Neill & Murchinson (1983).
C2:	An empirical constant, *C2*. This entry is optional. If omitted, the program supplies an appropriate value from tables provided by O'Neill & Murchinson (1983).
C3:	An empirical constant, *C2*. This entry is optional. If omitted, the program supplies an appropriate value from tables provided by O'Neill & Murchinson (1983).
K:	The initial modulus of subgrade reaction. This entry is optional. If omitted, the program supplies an appropriate value from tables provided by O'Neill & Murchinson (1983).

SOFT CLAY MODEL

P-y Curves – Soft Clay – General

Input:	Description
Set Name:	The name of the element set for which the P-y curve applies.
Start Depth:	The depth below the mudline at which the first P-y curve is will be generated.
End Depth:	The depth below the mudline at which the last P-y curve is will be generated.
Depth Increment:	The increment in depth, between the requested start and end depths, at which P-y curves will be generated.
Loading:	Generated curves for static and cyclical loading are different. Choose which loading type the generated P-y curve is intended for. Defaults to Static.
Representative Diameter:	The diameter of the structure in the depth region for which P-y curves are being generated. Where this varies, either use multiple blocks with different diameters, or choose a single representative diameter value.
Empirical Constant, J	A dimensionless empirical constant, with values ranging from 0.25 to 0.5, having been determined by field testing. Refer to Matlock (1970) for more details.
Scour Depth:	Specify a length of the scour region. The program will adjust the generated P-y curves based on this input. No P-y curves will be generated above the scour level and P-y curves below this level will be adjusted to use the adjusted distance below mudline.

P-y Curves – Soft Clay – By Depth

Input:	Description
Depth:	The distance below the mudline.

Undrained Shear Strength:	The average undrained soil shear strength at the given depth.
Submerged Unit Weight:	The soil submerged unit weight.
Strain at Half Max Stress:	The strain that occurs at one half of the maximum stress on a laboratory stress-strain curve derived from undrained compression tests of undisturbed soil samples. This entry is optional. The default is 0.01.

STIFF CLAY MODEL

P-y Curves – Stiff Clay – General

Input:	Description
Set Name:	The name of the element set for which the P-y curve applies.
Start Depth:	The depth below the mudline at which the first P-y curve is will be generated.
End Depth:	The depth below the mudline at which the last P-y curve is will be generated.
Depth Increment:	The increment in depth, between the requested start and end depths, at which P-y curves will be generated.
Loading:	Generated curves for static and cyclical loading are different. Choose which loading type the generated P-y curve is intended for. Defaults to Static.
Representativ e Diameter:	The diameter of the structure in the depth region for which P-y curves are being generated. Where this varies, either use multiple blocks with different diameters, or choose a single representative diameter value.

Static Loading Constant:	The static loading constant, K_s. This entry is optional. If omitted, the program supplies an appropriate value from tables provided by Reese et al. (1975).
Scour Depth:	Specify a length of the scour region. The program will adjust the generated P-y curves based on this input. No P-y curves will be generated above the scour level and P-y curves below this level will be adjusted to use the adjusted distance below mudline.

P-y Curves – Stiff Clay – By Depth

Input:	Description
Depth:	The distance below the mudline.
Undrained Shear Strength:	The average undrained soil shear strength at the given depth.
Submerged Unit Weight:	The soil submerged unit weight.
Ultimate Resistance Coefficient:	The ultimate resistance coefficient, A. This entry is optional. If omitted the program supplies an appropriate value from tables provided in Reese et al. (1975).
Strain at Half Max Stress:	The strain that occurs at one half of the maximum stress on a laboratory stress-strain curve derived from undrained compression tests of undisturbed soil samples. This entry is optional. If omitted, the program supplies an appropriate value from tables provided by Reese et al. (1975).

NOTES

(a) Flexcom automatically interpolates the P-y data for nodes located between any two specified depths.

(b) If the deflection at a particular node lies between two P-y data points, Flexcom uses linear interpolation to determine the relevant soil resistance.

(c) If the deflection at the node lies outside the specified range of the P-y curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

1.8.10.58 *RADIATION DAMPING

PURPOSE

To define radiation damping for a floating body.

THEORY

Refer to Wave Radiation Loads and Radiation Damping Coefficients for further information on this feature.

KEYWORD OVERVIEW

Data may defined explicitly within the keyword file itself, or defined separately in an external file which is simply referenced using a FILE= entry. The latter approach is highly recommended as it reduces unnecessary clutter in the keyword file, and also helps to improve user interface performance and enhanced user experience.

Block of data consisting of the floating body name followed by the radiation damping definition. For frequency independent radiation damping, a 6x6 matrix is defined by a single block preceded by a TYPE=CONSTANT line. For frequency dependent radiation damping, multiple 6x6 matrices are defined by several blocks of data, each one preceded by a FREQ= line for the different frequencies, with a TYPE=FREQUENCY line at the beginning. This block is repeated as often as necessary to define the radiation damping over a range of frequencies. Finally, an optional final block of data defining the radiation damping at the cut-off frequency, preceded by TYPE=CUTOFF, may be included. The entire block of data can then be repeated to specify radiation damping for second and subsequent floating bodies.

DATA SPECIFIED IN EXTERNAL FILE

Keyword Format

Line defining floating body name:

```
FLOATING BODY=Floating Body Name
```

Line defining name of external file which contains radiation damping data.

```
FILE=File Name
```

File Name should include the entire path of the file including its extension. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

Table Input

Input:	Description
Floating Body:	The name of the floating body.
File Name:	The name of the external data file. See Note (a).

Notes

(a) Refer to the following sections for further information regarding the required format of data within the external file.

CONSTANT RADIATION DAMPING

Purpose

To define constant (frequency-independent) radiation damping for a floating body.

Keyword Format

Line to define floating body name:

```
FLOATING BODY=Floating Body Name
```

Block of lines defining frequency independent radiation damping:

```
TYPE=CONSTANT
```

C_{11}, C_{12}, C_{13}, C_{14}, C_{15}, C_{16}
C_{21}, C_{22}, C_{23}, C_{24}, C_{25}, C_{26}
C_{31}, C_{32}, C_{33}, C_{34}, C_{35}, C_{36}
C_{41}, C_{42}, C_{43}, C_{44}, C_{45}, C_{46}
C_{51}, C_{52}, C_{53}, C_{54}, C_{55}, C_{56}
C_{61}, C_{62}, C_{63}, C_{64}, C_{65}, C_{66}

Table Input

Input:	Description
Floating Body:	The name of the floating body.
Matrix (6x6):	A 6x6 matrix of radiation damping data. See Note (a).

Notes

(a) Refer to Radiation Damping Coefficients for further information on the layout of the radiation damping terms.

(b) The type of radiation damping specified for a particular floating body, either *Constant* or *Frequency Dependent*, should be consistent with the type of added mass specified for the same floating body.

RADIATION DAMPING - REFERENCE MATRICES

Purpose

To reference frequency dependent and cut-off radiation damping matrices for use with a floating body.

Keyword Format

Line to define floating body name:

```
FLOATING BODY=Floating Body Name
```

Table Input

Input:	Description
Floating Body:	The name of the floating body.
Frequency Dependent Radiation Damping:	The name of the frequency-dependent radiation damping matrix definition.

Cut-Off Radiation Damping:	The name of the cut-off radiation damping matrix definition.

Notes

(a) Any properties which you refer to for a particular floating body must be defined in other tables – i.e. *Body - Frequency Dependent Radiation Damping* and *Body – Cut-Off Radiation Damping*.

FREQUENCY DEPENDENT RADIATION DAMPING

Purpose

To define frequency-dependent radiation damping for with a floating body.

Keyword Format

Block of lines defining frequency dependent radiation damping, preceded by a single TYPE=FREQUENCY line:

```
TYPE=FREQUENCY
FREQ=Frequency
```

C_{11}, C_{12}, C_{13}, C_{14}, C_{15}, C_{16}
C_{21}, C_{22}, C_{23}, C_{24}, C_{25}, C_{26}
C_{31}, C_{32}, C_{33}, C_{34}, C_{35}, C_{36}
C_{41}, C_{42}, C_{43}, C_{44}, C_{45}, C_{46}
C_{51}, C_{52}, C_{53}, C_{54}, C_{55}, C_{56}
C_{61}, C_{62}, C_{63}, C_{64}, C_{65}, C_{66}

Table Input

Input:	Description
Name:	The name of the radiation damping matrix.
Frequency:	The frequency in Hertz to which the 6x6 radiation damping matrix relates.
Matrix (6x6):	A 6x6 matrix of radiation damping data. See Note (a).

Notes

(a) Refer to Radiation Damping Coefficients for further information on the layout of the radiation damping terms.

(b) This table is used in conjunction with the *Body – Cut-Off Radiation Damping* table, in which radiation damping at the cut-off frequency is specified. Radiation damping at the cut-off frequency is completely defined by a single 6x6 matrix. The cut-off frequency itself is specified in the *Body – Frequency* table

(c) For frequencies below the cut-off frequency, the radiation damping corresponds to the radiation damping at the cut-off frequency. If no cut-off frequency is specified, the radiation damping is assumed to be zero below the range of user-specified frequencies.

(d) For frequencies above the range of user-specified frequencies, the radiation damping is assumed to be zero.

(e) For frequencies that do not exactly match one of the frequencies specified, Flexcom linearly interpolates between the nearest frequencies to find the relevant values of radiation damping.

(f) You do not need to specify the frequencies in any particular order; ascending order is not necessary.

(g) The type of radiation damping specified for a particular floating body, either *Constant* or *Frequency Dependent*, should be consistent with the type of added mass specified for the same floating body.

CUT-OFF RADIATION DAMPING

Purpose

To define cut-off radiation damping for a floating body.

Keyword Format

Block of lines defining radiation damping at the cut-off frequency:

TYPE=CUTOFF
C_{11}, C_{12}, C_{13}, C_{14}, C_{15}, C_{16}
C_{21}, C_{22}, C_{23}, C_{24}, C_{25}, C_{26}
C_{31}, C_{32}, C_{33}, C_{34}, C_{35}, C_{36}
C_{41}, C_{42}, C_{43}, C_{44}, C_{45}, C_{46}
C_{51}, C_{52}, C_{53}, C_{54}, C_{55}, C_{56}
C_{61}, C_{62}, C_{63}, C_{64}, C_{65}, C_{66}

Table Input

Input:	Description
Name:	The name of the radiation damping matrix.
Matrix (6x6):	A 6x6 matrix of radiation damping data. See Note (a).

Notes

(a) Refer to Radiation Damping Coefficients for further information on the layout of the radiation damping terms.

(b) This table is used in conjunction with the *Body - Frequency Dependent Radiation Damping* table, in which frequency-dependent radiation damping is specified. The cut-off frequency itself is specified in the *Body – Frequency* table.

(c) For frequencies below the cut-off frequency, the radiation damping corresponds to the radiation damping at the cut-off frequency. If no cut-off frequency is specified, the radiation damping is assumed to be zero below the range of user-specified frequencies.

1.8.10.59 *RAO FORMAT

PURPOSE

To define custom RAO conversion settings.

THEORY

Refer to Custom RAO Conversion for further information on this feature.

KEYWORD FORMAT

Each RAO format definition consists of a block of four lines. This block can be repeated as often as necessary to define further RAO formats.

Line specifying the format name:

```
FORMAT=Format Name
```

Line defining the heading:

```
ZERO=Heading Zero, [MIRROR=Heading Mirror]
```

Line stating the units used:

```
HEADING=Heading Units, [WAVE=Wave Units], [PHASE=Phase Units], [ROT=Rotationa
```

Line defining the positive directions:

```
HEADING=Positive Heading Direction [, ELEV=Positive Wave Elevation Direction]
[, HEAVE=Positive Heave Direction], [, SURGE=Positive Surge Direction] [, SWA
[, YAW=Positive Yaw Direction] [, ROLL=Positive Roll Direction] [, PITCH=Posi
```

Format Name cannot be MCS, AQWA, WAMIT, MOSES or ORCAFLEX – these are predefined formats that use the standard MCS RAO conventions. Each format must have a distinct name. *Heading Mirror* can be SURGE, SWAY, BOTH or NO. *Heading Units, Phase Units* and *Rotational Degree of Freedom Units* are either DEGREES or RADIANS. Wave Units can be HZ, PERIOD or RAD/S. *Positive Heading Direction* must be either CW (clockwise) or ACW (anticlockwise). *Positive Wave Elevation Direction* and *Positive Heave Direction* can be either UP or DOWN. Positive Phase Definition can be LAGGING or LEADING. *Positive Surge Direction* must be FORWARD or AFT. *Positive Sway Direction* can be PORT or STARBOARD. *Positive Yaw Direction, Positive Roll Direction* and *Positive Pitch Direction* can be either RH (right-handed) or LH (left-handed).

Note: If options are not specified they default to those specified by the MCS RAO convention.

RAO CONVERSION - DEFINE

Purpose

To define RAO conversion settings.

Table Input

Input:	Description
Conversion Name:	The name of the RAO conversion.

Mirrored RAOs:	This entry allows you to specify whether or not the RAO values specified in the vessel RAO file are to be mirrored around one or more of the vessel axes. The options are *No* (the default), *Surge, Sway* or *Both*.
Heading Zero Value:	The angle between the negative sense of the local vessel surge axis and a wave with zero incident wave heading, measured in degrees anti-clockwise from the bow in a plan view of the vessel. This entry defaults to zero, which means that a wave incident on the bow represents a wave heading of zero degrees.
Wave Units:	This entry allows you to select whether RAOs are specified as a function of wave frequency (*Frequency [Hz]* or *Frequency [rad/sec]*) or wave period (*Period [s]*) in the vessel RAO file. The default is as a function of wave frequency in Hz (*Frequency [Hz]*).
Phase Units:	This entry allows you to select whether phase angles specified in the vessel RAO file are given in terms of Degrees (the default) or Radians.
Heading Units:	This entry allows you to select whether the incident wave headings specified in the vessel RAO file are given in units of *Degrees* (the default) or *Radians*.
RAO Angular Units:	This entry allows you to select whether the RAO values for the rotational degrees of freedom specified in the vessel RAO file are given in units of *Degrees* (per unit wave elevation) or *Radians* (per unit wave elevation). The default is *Degrees*.
Response Sign:	This entry allows you to select whether phase angles specified in the vessel RAO file define a phase lag (*Lagging*, the default) or a phase lead (*Leading*) relative to the wave elevation.

Wave Elevation Sign:	This entry allows you to select the sign convention for wave elevations. You may specify whether a positive wave elevation is *Up* (the default) or *Down*.
Wave Heading Sign:	This entry allows you to select whether positive wave headings (as specified in the vessel RAO data) are measured *Clockwise* or *Anti-clockwise* (the default) from the point of zero-degree incident wave heading in a plan view of the vessel.
Positive Heave:	This entry allows you to select the direction convention for vessel heave. You may specify whether a positive heave represents vessel motion upwards (*Up*, the default) or downwards (*Down*).
Positive Surge	This entry allows you to select the direction convention for vessel surge. You may specify whether a positive surge represents vessel motion *Forward* (the default) or *Aft*.
Positive Sway:	This entry allows you to select the direction convention for vessel sway. You may specify whether a positive sway represents vessel motion to *Port* (the default) or *Starboard*.
Positive Yaw:	This entry allows you to select the direction convention for vessel yaw. You may specify whether positive yaw is relative to positive heave in a *Right Handed* (the default) or *Left Handed* sense.
Positive Roll:	This entry allows you to select the direction convention for vessel roll. You may specify whether positive roll is relative to positive surge in a *Right Handed* (the default) or *Left Handed* sense.
Positive Pitch:	This entry allows you to select the direction convention for vessel pitch. You may specify whether positive pitch is relative to positive sway in a *Right Handed* (the default) or *Left Handed* sense.

1.8.10.60 *RIGID SURFACE

PURPOSE

To specify the properties of a rigid seabed.

THEORY

Refer to Seabed Interaction for further information on this feature.

Note also that the old `*RIGID SURFACE` keyword has effectively been superseded by the new seabed definition keywords which facilitate the modelling of arbitrary seabed profiles. Refer to `*SEABED PROPERTIES`, `*SEABED PROFILE` and `*SEABED STIFFNESS` for further information.

KEYWORD FORMAT

A single line with the coefficients of friction and maximum characteristic lengths may optionally be followed by a single line (to define a sloping seabed) or several lines (to define an arbitrary seabed profile).

Line defining the seabed properties:

```
Longitudinal Friction Coefficient, Transverse Friction Coefficient [, Maximum
[, Maximum Transverse Characteristic Length] [, Constant Friction Force Coef
Length] [, Transverse Mobilisation Length]
```

Line defining uniform seabed slope:

```
Seabed Slope
```

Line defining a point of the arbitrary seabed:

```
Y Co-ordinate, Seabed Elevation
```

A model may not contain both rigid and elastic surfaces. The *Seabed Slope* is measured in degrees. A sloping seabed is assumed to pass through the global origin. A flat seabed is assumed if no slope is specified.

Both *Maximum Characteristic Length* inputs default to 10ft or 3.048m. Where a *Mobilisation Length* is explicitly specified, it takes precedence over the corresponding *Maximum Characteristic Length* (in this case the latter value is immaterial as it is unused). If a *Mobilisation Length* is omitted, then the mobilisation length is governed by the *Maximum Characteristic Length* and the finite element mesh discretisation. Refer to Seabed Friction for further details.

TABLE INPUT

Input:	Description
Longitudinal Coefficient of Friction:	The coefficient of friction in the longitudinal direction. This defaults to a value of 0.
Transverse Coefficient of Friction:	The coefficient of friction in the transverse direction. This defaults to a value of 0.
Slope:	The slope of a uniformly sloping seabed, in degrees. This defaults to a value of 0o, which gives a horizontal seabed. See Note (e).
Maximum Longitudinal Characteristic Length:	The maximum length used to simulate frictional restraint in the longitudinal direction on the seabed. This entry is optional. See Note (g).
Maximum Transverse Characteristic Length:	The maximum length used to simulate frictional restraint in the transverse direction on the seabed. This entry is optional. See Note (g).
Constant Friction Force Coefficient:	The constant friction force coefficient in the transverse direction to be used in addition to the classical limiting friction. This entry is optional and defaults to 0. See Note (h).
Longitudinal Mobilisation Length:	The mobilisation length used to simulate frictional restraint in the longitudinal direction on the seabed. This entry is optional. See Note (g).
Transverse Mobilisation Length	The mobilisation length used to simulate frictional restraint in the transverse direction on the seabed. This entry is optional. See Note (g).

NOTES

(a) The longitudinal friction coefficient refers to a local direction parallel to the pipe axis, and the transverse coefficient to a direction normal to the pipe axis.

(b) By default, no seabed is included in your analysis.

(c) The default friction coefficients give a smooth seabed, with no frictional resistance.

(d) This keyword and the `*ELASTIC SURFACE` keyword are mutually exclusive. If you have included an *Elastic Surface* in your analysis, then you may not include a *Rigid Surface* also.

(e) The seabed slope is input in degrees; a positive slope defines a seabed sloping upwards in the positive global Y direction, while a negative slope gives a seabed sloping in the opposite direction. A uniformly sloping seabed is assumed to pass through the global origin of co-ordinates. If a non-zero seabed slope is specified, then the definition of an arbitrary seabed bathymetry is not permitted.

(f) Note that, when determining the effect of hydrodynamic loading, Flexcom calculates water particle velocities and accelerations on the basis of the user-specified (constant) water depth, irrespective of whether an arbitrary profile rigid surface is present in the model or not. In other words, the presence of an arbitrary rigid surface is not considered to affect the hydrodynamic loading on a structure.

(g) The *Maximum Characteristic Length* and *Mobilisation Length* inputs relate to the operation of the Flexcom seabed friction model, and allow user control over the characteristics of non-linear springs used to model seabed frictional restraint. All these entries are optional. The *Maximum Characteristic Length* inputs default to 3.048m or 10ft. Where a *Mobilisation Length* is explicitly specified, it takes precedence over the corresponding *Maximum Characteristic Length* (in this case the latter value is immaterial as it is unused). If a *Mobilisation Length* is omitted, then the mobilisation length is governed by the *Maximum Characteristic Length* and the finite element mesh discretisation. Refer to Seabed Friction for further details.

(h) With the *Constant Friction Force Coefficient* input, the total limiting friction, in the transverse direction, is given by:

$$F_{total} = \mu N + CFF$$

where μN is the Coulombian limiting friction force, and *CFF* is the constant friction force. This equation is applicable only to transverse friction. The constant friction force is evaluated for each seabed contact element, and is given by the element length multiplied by the constant friction force coefficient.

1.8.10.61 *SEABED PROFILE

PURPOSE

To specify the seabed profile/bathymetry.

THEORY

Refer to Seabed Interaction for further information on this feature.

KEYWORD FORMAT

A block of two lines which defines the seabed profile. The first line defines the type of specification, *Sloping*, *2D* or *3D*, and the format of the second line depends on the specification type.

For a uniformly sloping seabed, the keyword format is as follows:

```
TYPE=SLOPING
[DIRECTION=Seabed Direction]
SLOPE=Seabed Slope
```

For 2D seabed profiles the keyword format is as follows:

```
TYPE=2D
[INTERPOLATION=Interpolation Method][, DIRECTION=Seabed Direction]
FILE=File Name
```

For 3D seabeds the keyword format is as follows:

```
TYPE=3D
[INTERPOLATION=Interpolation Method]
FILE=File Name
```

The `Seabed Slope` is measured in degrees. A sloping seabed is assumed to pass through the global origin. See Note (b).

The *Seabed Direction* is optional. It is measured in degrees and rotates the seabed profile counter-clockwise about the global X axis.

The *Interpolation Method* is optional. For a 2D seabed it may be `LINEAR` (the default), `CUBIC SPLINE` or `CUBIC BESSEL`. For a 3D seabed it may be `LINEAR` (the default) or `CUBIC`.

For 2D or 3D seabed profiles the file *File Name* is a binary file generated by the Seabed Utility application and has the extension .FCSBD. See Notes (c) and (d).

Additionally, for backward compatibility, the file *File Name* for linear 2D seabed profiles may be specified as a text file. See Note (e). In this case, the format of the external ASCII data file is as follows:

A block of lines defining various points on the arbitrary seabed. The line may be repeated as often as necessary to completely define the seabed profile.

Y Co-ordinate, Seabed Elevation

TABLE INPUTS

Seabed Profile – Sloping

Input:	Description
Seabed Direction:	The orientation of the seabed measured in degrees counter-clockwise about the global X axis. This value defaults to zero if omitted.
Seabed Slope:	The slope of a uniformly sloping seabed, in degrees. This defaults to a value of 0.0 degrees, which gives a horizontal seabed. See Note (b).

Seabed Profile – 2D

Input:	Description

Interpolation Method:	This option allows you to specify how the seabed is interpolated between the seabed data points. The options are `LINEAR` (the default), `CUBIC SPLINE` or `CUBIC BESSEL`.
Seabed Direction:	The orientation of the seabed measured in degrees counter-clockwise about the global X axis. This value defaults to zero if omitted.
Seabed File Name:	The name of the seabed file containing the 2D seabed profile data. See Note (c).

Seabed Profile – 3D

Input:	Description
Interpolation Method:	This option allows you to specify how the seabed is interpolated between the seabed data points. The options are `LINEAR` (the default) or `CUBIC`.
Seabed File Name:	The name of the seabed file containing the 3D seabed profile data. See Note (d).

Notes

(a) If `*SEABED PROPERTIES` is specified but this keyword is omitted, then a default horizontal seabed is assumed. In this case the seabed is coincident with a plane at zero global X elevation (the global YZ plane).

(b) If the sloping seabed option is invoked, a uniformly sloping seabed is modelled which passes through the global origin of co-ordinates. The seabed slope is specified in degrees, and a positive slope defines a seabed sloping upwards in the positive global Y direction, while a negative slope gives a seabed sloping in the opposite direction. Optionally, the sloping seabed can be rotated about the global X axis by specifying a *`Seabed Direction`*.

(c) The profile of the 2D seabed varies in the global XY plane and is then extruded horizontally along the global Z direction. The profile is defined in the seabed file (.FCSBD) by a series of pairs of values. The file is created using the Seabed Utility standalone application. `LINEAR`, `CUBIC SPLINE` or `CUBIC BESSEL` interpolation is used to determine the surface profile between user specified points, while outside the range of points the surface is assumed to be flat and horizontal. Optionally, the 2D seabed can be rotated about the global X axis by specifying a *`Seabed Direction`*.

(d) The profile of the 3D seabed is defined in a binary seabed file (.FCSBD) by a series of global XYZ coordinate values lying on the seabed surface. The seabed file is created using the Seabed Utility standalone application and has the .FCSBD extesion. `LINEAR` or `CUBIC` interpolation may be used to determine the surface elevation between user specified points, while outside the boundary of the point set the surface elevation is assumed to be infinitely deep.

(e) For text based 2D seabed files, the profile is defined by a series of pairs of values. This is the data that is contained in the ASCII data file. Each line in the file defines a point on the profile and contains data in the format *Y-coordinate*, *Seabed Elevation*. Here *Y-coordinate* is self-explanatory, and *Seabed Elevation* is the height of the seabed at this point above the datum X=0.

(e) In the default case of a flat seabed, the seabed is coincident with a plane at zero datum (the global YZ plane).

(e) If the sloping seabed option is invoked, a uniformly sloping seabed is modelled which passes through the global origin of co-ordinates. The seabed slope is specified in degrees, and a positive slope defines a seabed sloping upwards in the positive global Y direction, while a negative slope gives a seabed sloping in the opposite direction.

(e) The profile of the arbitrary 2D seabed varies in the global XY plane and is constant in the global Z direction. The profile is defined by a series of pairs of values, and this is the data that is contained in the ASCII data file. Each line in the file defines a point on the profile and contains data in the format *Y-coordinate*, *Seabed Elevation*. Here *Y-coordinate* is self-explanatory, and *Seabed Elevation* is the height of the seabed at this point above the datum X=0. Linear interpolation is used to determine the surface profile between user-specified points, while outside the range of points the surface is assumed to be flat and horizontal.

1.8.10.62 *SEABED PROPERTIES

PURPOSE

To specify properties such as seabed type (i.e. rigid or elastic) and friction parameters.

THEORY

Refer to Seabed Interaction for further information on this feature.

KEYWORD FORMAT

A block of lines defining the seabed properties.

The block begins with a mandatory line defining the seabed model, whether rigid or elastic.

Mandatory line defining the seabed model:

```
TYPE=Seabed Type
```

This is followed by a block of lines beginning with the element set definition. Next comes an optional line defining seabed stiffness, which is only relevant for an elastic seabed. This is followed by various inputs pertaining to seabed friction. The entire block may then be repeated for subsequent element sets so that different seabed properties may be used with different portions of the model.

Mandatory line specifying the element set:

```
SET=Set Name
```

Optional line specifying the seabed contact stiffness:

```
[STIFFNESS=Stiffness Value or Curve Name]
```

Optional line specifying seabed friction data:

```
[FRICTION=Longitudinal Coefficient, Transverse Coefficient [, Longitudinal Mo
Transverse Mobilisation Length] [, Constant Friction Force Coefficient] [, Lo
Transverse Stiffness]
```

Optional line specifying lateral stiffness (relevant to elastic seabed only):

```
[LATERAL=Lateral Seabed Stiffness]
```

Optional line specifying suction zone parameters (relevant to elastic seabed only):

```
[SUCTION=Suction Stiffness [, Suction Zone Extent]]
```

`Seabed Type` may be RIGID or ELASTIC. Only one of these types is allowed in your model. Any given element may only be assigned seabed properties once. In the event that element sets overlap, the latest definition in the keyword file will take precedence. If a nonlinear seabed stiffness is referenced, it must be defined under the `*SEABED STIFFNESS` keyword. Both `Mobilisation Length` inputs default to 0.15m or the equivalent in feet. Refer to Seabed Friction for further details on the friction model employed by Flexcom.

The `Longitudinal Stiffness` and the Longitudinal Stiffness are only used in frequency domain or modal analyses.

TABLE INPUTS

Seabed Properties – Rigid

Input:	Description
Seabed Stiffness:	The elastic stiffness per unit length of the seabed.
Longitudinal Coefficient of Friction:	The coefficient of friction in the longitudinal direction. This entry is optional and defaults to 0.0 if omitted.
Transverse Coefficient of Friction:	The coefficient of friction in the transverse direction. This entry is optional and defaults to 0.0 if omitted.
Longitudinal Mobilisation Length:	The mobilisation length used to simulate frictional restraint in the longitudinal direction on the seabed. This entry is optional and defaults to 0.15m (or the equivalent in feet) if omitted. See Note (d).

Transverse Mobilisation Length	The mobilisation length used to simulate frictional restraint in the transverse direction on the seabed. This entry is optional and defaults to 0.15m (or the equivalent in feet) if omitted. See Note (d).
Constant Friction Force Coefficient:	The constant friction force coefficient in the transverse direction to be used in addition to the classical limiting friction. This entry is optional and defaults to 0.0 if omitted. See Note (e).
Longitudinal Friction Stiffness:	The value of the longitudinal friction stiffness. This entry is optional and defaults to 0.0 if omitted. See Note (f).
Transverse Friction Stiffness:	The value of the transverse friction stiffness. This entry is optional and defaults to 0.0 if omitted. See Note (f).

Seabed Properties – Elastic

Input:	Description
Seabed Stiffness:	The elastic stiffness per unit length of the seabed.
Longitudinal Coefficient of Friction:	The coefficient of friction in the longitudinal direction. This entry is optional and defaults to 0.0 if omitted.
Transverse Coefficient of Friction:	The coefficient of friction in the transverse direction. This entry is optional and defaults to 0.0 if omitted.
Longitudinal Mobilisation Length:	The mobilisation length used to simulate frictional restraint in the longitudinal direction on the seabed. This entry is optional and defaults to 0.15m (or the equivalent in feet) if omitted. See Note (d).

Transverse Mobilisation Length	The mobilisation length used to simulate frictional restraint in the transverse direction on the seabed. This entry is optional and defaults to 0.15m (or the equivalent in feet) if omitted. See Note (d).
Constant Friction Force Coefficient:	The constant friction force coefficient in the transverse direction to be used in addition to the classical limiting friction. This entry is optional and defaults to 0.0 if omitted. See Note (e).
Longitudinal Friction Stiffness:	The value of the longitudinal friction stiffness. This entry is optional and defaults to 0.0 if omitted. See Note (f).
Transverse Friction Stiffness:	The value of the transverse friction stiffness. This entry is optional and defaults to 0.0 if omitted. See Note (f).
Lateral Seabed Stiffness:	This input allows you to specify a lateral resistance to horizontal motion for the elastic seabed. This entry is optional and defaults to 0.0 if omitted. See Note (g).
Suction Stiffness:	This input allows you to input a seabed stiffness for elements which have moved off an elastic seabed but which are still in a so called "suction zone" between the mudline and the elevation at which suction forces disappear. This entry is optional and defaults to 0.0 if omitted. See Note (h).
Suction Zone Extent:	This is related to the previous input, *Suction Stiffness*. It represents the elevation or height above the mudline at which suction forces become zero. This input is meaningless unless the *Suction Stiffness* is nonzero, but is required if the *Suction Stiffness* is non-zero. See Note (h).

Notes

(a) The longitudinal friction coefficient refers to a local direction parallel to the pipe axis, and the transverse coefficient to a direction normal to the pipe axis.

(b) The default friction coefficients give a smooth seabed, with no frictional resistance.

(c) Linear elastic stiffness is input as a stiffness per unit length of riser or pipeline, in units of [Force]/[Distance]/[Distance] or [Force]/[Distance]^2. If a nonlinear seabed stiffness is referenced, it must be defined via under the *Seabed Stiffness* table.

(d) The *Mobilisation Length* inputs relate to the operation of the Flexcom seabed friction model, and allow user control over the characteristics of non-linear springs used to model seabed frictional restraint. Both *Mobilisation Length* inputs default to 0.15m or the equivalent in feet. Refer to Seabed Friction for further details on the friction model employed by Flexcom.

(e) With the *Constant Friction Force Coefficient* input, the total limiting friction, in the transverse direction, is given by:

$$F_{total} = \mu N + CFF$$

where μN is the Coulombian limiting friction force, and CFF is the constant friction force. This equation is applicable only to transverse friction. The constant friction force is evaluated for each seabed contact element, and is given by the element length multiplied by the constant friction force coefficient.

(f) Regarding the *Friction Stiffness* entries, you can model friction in a frequency domain dynamic analysis using either a total restraint, or a partial restraint modelled using a spring stiffness. For more information, please refer to Seabed Modelling in Frequency Domain Analysis. By default, seabed friction continues to be modelled in the same way as in earlier program versions (the so-called *Fully Restrained Model*). This corresponds to the situation where both stiffness entries are zero. To apply a partial restraint in either the longitudinal or transverse direction, you input a non-zero stiffness in the appropriate direction. Note however that a non-zero value is immaterial and unused if the corresponding friction coefficient is zero. In that case, free motion in the appropriate direction is permitted.

(g) The *Lateral Seabed Stiffness* entry specifies a lateral resistance to horizontal motion on the elastic seabed. Refer to Lateral Resistance for further information on this feature.

(h) The suction or restraining force experienced by a riser element in a “suction zone” just above the mudline is modelled with a linear spring resistance similar to that provided against downward vertical motion by the elastic seabed itself. Refer to Suction Zone for further information on this feature.

1.8.10.63 *SEABED STIFFNESS

PURPOSE

To define nonlinear seabed contact stiffness.

THEORY

Refer to Embedment for further information on this feature.

KEYWORD FORMAT

A block of lines that define a force-embedment curve repeated as often as necessary. The block begins with a line defining the curve name. It is followed by as many lines as necessary to define each point on the curve.

Line defining the element set:

`CURVE=`*`Curve Name`*

Line defining a point on a curve:

`Embedment Ratio, Seabed Force`

Each curve must have at least two points defined. A force-embedment curve may only be specified for an elastic seabed. If a nonlinear seabed stiffness is referenced under the `*SEABED PROPERTIES` keyword, it must be defined here.

TABLE INPUTS

Input:	Description
Curve Name:	The name of the force-embedment curve.
Embedment Ratio:	An embedment ratio value on the curve.
Seabed Force:	The force exerted by the seabed for this embedment ratio.

NOTES

(a) The embedment ratio of an element is defined as the average distance the element centreline lies below the seabed, divided by element external diameter. So embedment ratio is dimensionless. “Average distance” in this context means the average of the distances below the seabed of the two nodes on the element.

(b) How this facility operates is as follows. Flexcom computes the average embedment of an element lying on the elastic seabed at each iteration at each solution time. Then using the embedment curve for the element that you specify here, the tangent stiffness of the curve at that embedment is calculated. This value is used as the vertical seabed stiffness for that element.

(c) You should use as many lines as you need to completely define a force/embedment ratio curve for a particular element set. Simply leave Column 1 blank for second and subsequent lines. For subsequent element sets, put the set name in Column 1 and specify the force/embedment ratio data in the same way. The points defining the force/embedment curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of embedment ratio.

(d) If the computed embedment ratio of an element lies between the specified force/embedment data points that you specify, Flexcom uses linear interpolation to determine the seabed force on the element.

(e) If the computed embedment ratio of an element lies outside the specified range of the force/embedment curve, then Flexcom assumes:

- For embedment ratios greater than the range of the curve, the tangent stiffness of the element is equal to the tangent stiffness between the final two points defined on the curve.
- For embedment ratios less than the range of the curve, the tangent stiffness of the element is equal to the tangent stiffness between the first two points defined on the curve.

1.8.10.64 *SET COLOURS

PURPOSE

To assign a specific colour to an element set or sets.

THEORY

Refer to Model View for further information on this feature.

KEYWORD FORMAT

A single line defining element set and associated colour, repeated as often as necessary.

```
SET=Element Set, COLOUR=Set Colour
```

Note: Set Colour may be either predefined or user-defined. The predefined colour set is composed of 140 standard colours. User-defined colours may be defined using the `*COLOUR DEFINE` keyword.

TABLE INPUT

Input:	Description
Set Name	The element set to which the colours are assigned.
Colour	Name of predefined or user-defined colour. See Note (a).

NOTES

(a) The predefined set of colours and their names are listed in Table 1.

Predefined Colours and Names.

AliceBlue		GhostWhite		Moccasin	
AntiqueWhite		Gold		NavajoWhite	
Aqua		GoldenRod		Navy	
Aquamarine		Gray		OldLace	
Azure		Green		Olive	
Beige		GreenYellow		OliveDrab	
Bisque		HoneyDew		Orange	
Black		HotPink		OrangeRed	
BlanchedAlmond		IndianRed		Orchid	
Blue		Indigo		PaleGoldenRod	
BlueViolet		Ivory		PaleGreen	
Brown		Khaki		PaleTurquoise	
BurlyWood		Lavender		PaleVioletRed	
CadetBlue		LavenderBlush		PapayaWhip	
Chartreuse		LawnGreen		PeachPuff	
Chocolate		LemonChiffon		Peru	
Coral		LightBlue		Pink	
CornflowerBlue		LightCoral		Plum	
Cornsilk		LightCyan		PowderBlue	
Crimson		LightGolden-RodYellow		Purple	
Cyan				Red	
DarkBlue		LightGray		RosyBrown	
DarkCyan		LightGreen		RoyalBlue	
DarkGoldenRod		LightPink		SaddleBrown	
DarkGray		LightSalmon		Salmon	
DarkGreen		LightSeaGreen		SandyBrown	
DarkKhaki		LightSkyBlue		SeaGreen	
DarkMagenta		LightSlateGray		SeaShell	
DarkOliveGreen		LightSteelBlue		Sienna	
Darkorange		LightYellow		Silver	
DarkOrchid		Lime		SkyBlue	
DarkRed		LimeGreen		SlateBlue	
DarkSalmon		Linen		SlateGray	

DarkSeaGreen		Magenta		Snow	
DarkSlateBlue		Maroon		SpringGreen	
DarkSlateGray		MediumAquaMarine		SteelBlue	
DarkTurquoise		MediumBlue		Tan	
DarkViolet		MediumOrchid		Teal	
DeepPink		MediumPurple		Thistle	
DeepSkyBlue		MediumSeaGreen		Tomato	
DimGray		MediumSlateBlue		Turquoise	
DodgerBlue		MediumSpringGreen		Violet	
FireBrick		MediumTurquoise		Wheat	
FloralWhite		MediumVioletRed		White	
ForestGreen		MidnightBlue		WhiteSmoke	
Fuchsia		MintCream		Yellow	
Gainsboro		MistyRose		YellowGreen	

1.8.10.65 *SPRING ELEMENT

PURPOSE

To define the stiffness characteristics of spring elements

THEORY

Refer to Spring Elements for further information on this feature.

KEYWORD FORMAT

Two blocks of lines which define linear and non-linear spring elements and which may be repeated as often as necessary.

Definition of linear spring elements:

```
ELEMENT=List of Elements (Numbers or Labels)
TYPE=LINEAR
Stiffness
```

Definition of non-linear spring elements, with the third line repeated as often as necessary:

```
ELEMENT=List of Elements (Numbers or Labels)
TYPE=NONLINEAR
Displacement, Force
```

Up to 20 elements can be specified after ELEMENT=. At least two points are required for a non-linear spring force-deflection curve. If you specify an element label rather than an element number, it must be enclosed in {} brackets.

TABLE INPUT

Linear Springs (By Element)

Input:	Description
Element:	A list of elements (numbers or labels) whose linear spring stiffness is being defined. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Stiffness:	The linear spring stiffness.

Notes

(a) A spring element is wholly characterised by its spring stiffness, so no geometric or hydrodynamic properties should be attributed to it. Note that spring elements are massless and unaffected by distributed forces.

Linear Springs (By Set)

Input:	Description
Set:	The element set whose linear spring stiffness is being defined.
Stiffness:	The linear spring stiffness.

Notes

(a) A spring element is wholly characterised by its spring stiffness, so no geometric or hydrodynamic properties should be attributed to it. Note that spring elements are massless and unaffected by distributed forces.

Non-Linear Springs (By Element)

Input:	Description
Element:	A list of element (numbers or labels) whose non-linear force-deflection characteristics are being defined. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Displacement:	A displacement value.
Force:	A corresponding force value.

Notes

(a) Use as many lines as you need to completely define the force-deflection relationship for a particular list of elements. Simply leave Column 1 blank for second and subsequent lines. For subsequent lists, put the list in Column 1 and specify the spring characteristics in the same way.

(b) The force-deflection curve for non-linear spring elements may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of deflection.

(c) If the computed deflection of the spring element lies between the specified force-deflection data pairs, then Flexcom interpolates linearly to evaluate the force in the spring element.

(d) If the computed deflection of the spring element lies outside the specified range of the force-deflection curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

(e) If none of the specified displacement terms have a negative value, the curve is assumed to be symmetrical about the origin (i.e. the behaviour of the spring element is the same for both positive and negative deflections).

Non-Linear Springs (By Set)

Input:	Description
Set:	The element set whose non-linear force-deflection characteristics are being defined.

Displacement:	A displacement value.
Force:	A corresponding force value.

Notes

(a) Use as many lines as you need to completely define the force-deflection relationship for a particular set. Simply leave Column 1 blank for second and subsequent lines. For subsequent sets, put the set name in Column 1 and specify the spring characteristics in the same way.

(b) The force-deflection curve for non-linear spring elements may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of deflection.

(c) If the computed deflection of the spring element lies between the specified force-deflection data pairs, then Flexcom interpolates linearly to evaluate the force in the spring element.

(d) If the computed deflection of the spring element lies outside the specified range of the force-deflection curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

(e) If none of the specified displacement terms have a negative value, the curve is assumed to be symmetrical about the origin (i.e. the behaviour of the spring element is the same for both positive and negative deflections).

1.8.10.66 *STIFFENER

PURPOSE

To define the properties of a conical bend stiffener positioned on a flexible riser or pipe.

THEORY

Refer to Bend Stiffeners for further information on this feature.

KEYWORD FORMAT

A block of two lines to define a bend stiffener repeated as often as necessary.

```
SET=Set Name [, E=Curve Name] [, SYMMETRY=Symmetry]
```

Dfree, Dfix, [E], m, Dint[, Mass & Buoyancy Flag]

Mass & Buoyancy Flag defaults to 0, meaning the stiffener mass and buoyancy are not included in the analysis. For a non-linear stiffener the value of E should be omitted. *Symmetry* can be *AUTOMATIC* (the default), *SYMMETRIC* or *ASYMMETRIC*. You must use the *NONLINEAR STIFFENER keyword to define non-linear stiffener material properties if you specify a non-linear stiffener here.

TABLE INPUT

Input:	Description
Set Name:	The element set for which the bend stiffener properties are being defined. See Notes (a) and (b).
Dfree:	The stiffener diameter at the tip or free end of the bend stiffener. See Notes (c) and (d).
Dfix:	The stiffener diameter at the base or fixed end of the bend stiffener. See Notes (c) and (d).
E:	The value of Young's Modulus for the stiffener or the name of a non-linear material stress/strain curve that defines the material bending stiffness. See Note (e).
m:	The total mass (in air) of the stiffener.
Dint:	The internal diameter of the stiffener. See Note (f).
Mass/ Buoyancy	This option allows you to specify whether the mass and buoyancy of the stiffener are to be included in the analysis. See Note (g). The default is No, which means the stiffener mass and buoyancy are not included in the analysis
Symmetry	The bending symmetry model of the nonlinear bend stiffener. See Note (h).

NOTES

(a) The procedure when using this option to define the properties of a conical bend stiffener is as follows. You first define the properties of the riser or pipe in the usual way using the *GEOMETRIC SETS keyword. You then define an element set for the section over which the bend stiffener is to be positioned, and then complete the specification by inputting the actual stiffener properties and dimensions here.

(b) This means that the properties assigned to elements using this menu are additional to the properties defined using the *GEOMETRIC SETS keyword. Flexcom automatically increases the bending stiffness of the elements where the bend stiffener is positioned, using the data you specify here.

(c) The order in which the elements are specified in defining the bend stiffener element set is very important. The first element specified is assumed to be at the free end of the bend stiffener and the last element is assumed to be at the fixed end. Intermediate elements must be specified in order from the free end to the fixed end. Note that the elements in an element set can be specified in any order, including reverse (descending) order, for this reason. Elements can be added to an element set in reverse order by using an Element Increment of -1 under *ELEMENT SETS

(d) The free end diameter must be less than the fixed end diameter.

(e) If a non-linear material curve name is specified for Young's Modulus, then the non-linear material curve must be defined using the the *NONLINEAR STIFFENER keyword.

(f) The specification of an internal diameter for the bend stiffener is optional. If omitted, the internal diameter defaults to the drag diameter of the first element of the stiffener set.

(g) Refer to Bend Stiffeners for a discussion on how bend stiffeners are handled in this program release and earlier versions. Specifically, the mass (inertia), weight and buoyancy of the stiffener are not by default included in an analysis. You do though have the option to instruct Flexcom to include them, which you do using the *Mass/Buoyancy* list.

(h) Refer to the Non-Linear Elastic section for a discussion on nonlinear bending symmetry models. If the bending symmetry of a nonlinear bend stiffener is to be automatically set by Flexcom then the symmetry chosen will depend on the material of the element to which the stiffener is attached. The stiffener bending symmetry will be asymmetric only if the element material is specified using the Flexible Riser Format and is (i) nonlinear asymmetric, i.e. EI_{yy} and EI_{zz} are defined by nonlinear stress/strain curves, or (ii) linear with unequal values of EI_{yy} and EI_{zz}. In all other cases the stiffener bending symmetry will be symmetric.

1.8.10.67 *STRESS/STRAIN

PURPOSE

To define generalised stress-strain curves for non-linear materials.

THEORY

Refer to Non-Linear Elastic materials for further information on this feature.

Note that this keyword has effectively been superseded by the new non-linear material definition keywords which explicitly distinguish between bending, axial and torsional stiffness. Refer to `*MOMENT-CURVATURE`, `*FORCE-STRAIN` and `*TORQUE-TWIST` for further information.

KEYWORD FORMAT

A block of lines that defines a stress-strain curve, repeated as often as necessary. The block begins with a line defining the curve name. It is followed by as many lines as necessary to define each point on the curve.

Line defining the curve name:

```
CURVE=Curve Name
```

Line defining a point on a curve:

```
Generalised Stress, Generalised Strain
```

Each curve must have at least two points defined. This type of stress-strain curve may not be associated with non-linear beam elements which are defined using the rigid riser format for geometric properties specification.

TABLE INPUT

Input:	Description
Curve Name:	The generic name of the generalised stress-strain curve.
Generalised Stress:	A generalised stress value for a point on the curve.
Generalised Strain:	The corresponding generalised strain value.

NOTES

(a) This table is used to define generalised stress-strain curves that define EIyy, EIzz, GJ or EA for a particular set of elements. Generalised stress-strain curves may be assigned to element sets using the Flexible Riser Format.

(b) If the non-linear material curve defines a bending stiffness, then the generalised stress inputs are bending moments and the generalised strains are the corresponding curvature values. If the curve is defining torsional stiffness, then the generalised stress inputs are torsional moments and the generalised strain inputs are the corresponding torsional strains with units [Radian]/[Length]. If the curve is defining axial stiffness, then the generalised stress inputs are axial forces and the generalised strains are the corresponding axial strains.

(c) Use as many lines as you need to completely define a particular stress-strain curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent stress-strain curves, put the curve name in Column 1 and specify the stress-strain data in the same way.

(d) The points defining the non-linear stress-strain curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of strain.

(e) If the strain in an element lies between the stress-strain data points you specify, Flexcom uses linear interpolation to determine the relevant stiffness for the element.

(f) If the strain in the element lies outside the specified range of the stress-strain curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

(g) If none of the specified strain terms have a negative value, the curve is assumed to be symmetrical about the origin (i.e. the behaviour of the element is the same for both positive and negative strains).

1.8.10.68 *STRESS/STRAIN DIRECT

PURPOSE

To define direct stress-strain curves for non-linear materials.

THEORY

Refer to Non-Linear Elastic materials for further information on this feature.

KEYWORD FORMAT

A block of lines that defines a stress-strain curve, repeated as often as necessary. The block begins with a line defining the curve name. It is followed by as many lines as necessary to define each point on the curve.

Line defining the curve name:

`CURVE=`*`Curve Name`*

Line defining a point on a curve:

`Direct Stress, Direct Strain`

Each curve must have at least two points defined. This type of stress-strain curve may not be associated with non-linear beam elements which are defined using the rigid riser format for geometric properties specification.

TABLE INPUT

Input:	Description
Curve Name:	The generic name of the direct stress-strain curve.
Stress:	A direct stress value for a point on the curve.
Strain:	The corresponding direct strain value.

NOTES

(a) This table is used to define direct stress-strain curves that define E for a particular set of elements. Direct stress-strain curves may be assigned to element sets using the Rigid Riser Format table.

(b) Use as many lines as you need to completely define a particular stress-strain curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent stress-strain curves, put the curve name in Column 1 and specify the stress-strain data in the same way.

(c) The points defining the non-linear stress-strain curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of strain.

(d) If the strain in an element lies between the specified stress-strain data points, Flexcom uses linear interpolation to determine the relevant stiffness of the element.

(e) If the strain in the element lies outside the specified range of the stress-strain curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

(f) If none of the specified strain terms have a negative value, the curve is assumed to be symmetrical about the origin (i.e. the behaviour of the element is the same for both positive and negative strains)

1.8.10.69 *T-Z

PURPOSE

To define T-z curves for modelling soil structure interaction.

THEORY

Refer to Soil Modelling for further information on this feature.

Soil structure interaction can be modelled explicitly by manually specifying T-z curves at different depths.

KEYWORD OVERVIEW

The keyword format is a block of lines defining the interaction between a particular structure and the soil. The first line is an optional one which specifies the format in which the properties are input. This is followed by a number of lines defining the actual data. As many blocks as required can be used to specify the soil structure interaction over the complete depth required.

User Format

Data may defined explicitly within the keyword file itself, or defined separately in an external file which is simply referenced using a FILE= entry. The latter approach is highly recommended as it reduces unnecessary clutter in the keyword file, and also helps to improve user interface performance and enhanced user experience.

The format of the block when using directly entered T-z curve data:

```
SET=Set Name
DEPTH=Depth
Soil Resistance (T), Deflection (z)
```

The first line indicates the element set for which the subsequent T-z curves apply. The next line defines the depth to which the curve relates and marks the start of a block containing the T-z curve data at that depth. This block can be repeated at as many depths as required. The final line is a pair of T, z values. This line will be repeated several times for describe a T-z curve.

The format of the block when using T-z curve data from an external file:

```
SET=Set Name
FILE=File Name
```

The first line indicates the element set for which the T-z curves apply. The second line specifies an external file containing T-z curve data.

TABLE INPUTS

T-z Curves - Explicit

Input:	Description
Set Name:	The name of the element set for which the T-z curve applies.
Depth:	The depth below the mudline at which the T-z curve is being defined.

Soil Resistance, T:	A soil resistance (T) value for a point on the T-z curve. This has units of force/length.
Deflection, z:	The corresponding deflection (z) value.

T-z Curves - File

Input:	Description
Set Name:	The name of the element set for which the T-z curve applies.
File Name:	The file containing the T-z curve data. This data includes depth and T-z data.

Notes

(a) Use as many lines as you need to completely define a particular T-z curve, for a particular combination of element set and depth. Simply leave the *Set Name* and *Depth* columns blank for second and subsequent lines of the curve data. For subsequent T-z curves, specify a new *Set Name* and/or *Depth*, and specify the T-z data in the same way.

(b) The points defining the T-z curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of deflection.

NOTES

(a) Flexcom automatically interpolates the T-z data for nodes located between any two specified depths.

(b) If the deflection at a particular node lies between two T-z data points, Flexcom uses linear interpolation to determine the relevant soil resistance.

(c) If the deflection at the node lies outside the specified range of the T-z curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

1.8.10.70 *TAPER

PURPOSE

To assign properties to a set of elements that collectively comprises a tapered stress joint, typically in a model of a rigid riser.

THEORY

Refer to Tapered Joints for further information on this feature.

KEYWORD FORMAT

A block of lines repeated as many times as necessary.

```
SET=Set Name
E, G, rho, Dis, Dos, Doe [, Die, Dbs, Dbe, DDs, Dde ]
[COMPRESSION CHECK=Check Setting]
```

The first six values must be specified. Defaults are calculated for any of the other five if omitted. `Check Setting` may be either AUTOMATIC (the default), NONE or a numeric value.

TABLE INPUT

Input:	Description
Set Name:	The name of the set comprising the tapered stress joint.
E:	The Young's Modulus for the elements of the set.
G:	The shear modulus for the elements of the set.
rho:	The mass density (mass per unit volume) of the material of the set.
Dis:	The riser internal diameter at the start of the tapered stress joint. Dis is used in computing the buoyancy contribution of the internal fluid, if there is any.
Dos:	The riser outer diameter at the start of the tapered stress joint. This is used in computing the area, moment of inertia and polar moment of inertia for the elements of the set.

Doe:	The riser outer diameter at the end of the tapered stress joint. Riser outer diameter is assumed to vary linearly between Dos and Doe.
Die:	The riser internal diameter at the end of the tapered stress joint. This entry is optional, and defaults to Dis. The default value will apply in the vast majority of applications.
Dbs:	The buoyancy diameter at the start of the tapered stress joint. This is the effective outer diameter for buoyancy force calculations. This entry is optional, and defaults to Dos.
Dbe:	The buoyancy diameter at the end of the tapered stress joint. This entry is optional, and defaults to Doe. Buoyancy diameters at intermediate locations are found by linear interpolation between Dbs and Dbe.
Dds:	The drag diameter at the start of the tapered stress joint. This is the effective outer diameter for hydrodynamic force calculations. This entry is optional, and defaults to Dos.
Dde:	The drag diameter at the end of the tapered stress joint. This entry is optional, and defaults to Doe. Drag diameters at intermediate locations are found by linear interpolation between Dds and Dde.
Compression Check:	Flexcom will issue a warning if the compressive load experienced in any element exceeds the critical Euler load. Refer to Compression and Buckling for further information. The options are *Automatic* (the default), *Manual* and *None*. If you opt for *Manual*, you must explicitly set a *Compression Limit* also.
Compression Limit:	A manually defined compression limit to be used in the Euler load check.

NOTES

(a) The procedure when using this option to define the properties of a tapered stress joint is as follows. You first define an element set for the stress joint, and then complete the specification by inputting the actual joint properties and dimensions here. This means that all the geometric properties for the elements of the set are specified here. Unlike the case of a bend stiffener, the properties you specify here are not additional to properties defined using the `*GEOMETRIC SETS` keyword. Indeed, it is an error to assign properties to elements using both the `*GEOMETRIC SETS` and `*TAPER` keywords.

(b) You must specify non-zero values for E, G, rho, Dos and Doe, and a non-zero value for Di would also be normal. All of the other inputs are optional.

1.8.10.71 *TORQUE-TWIST

PURPOSE

To define torque-twist curves for non-linear materials.

THEORY

Refer to Non-Linear Elastic materials for further information on this feature.

Note also that the old `*STRESS/STRAIN` keyword has effectively been superseded by the new non-linear material definition keywords which explicitly distinguish between bending, axial and torsional stiffness. Refer to related keywords `*MOMENT-CURVATURE` and `*FORCE-STRAIN`.

KEYWORD FORMAT

A block of lines that defines a torque-twist curve, repeated as often as necessary. The block begins with a line defining the curve name. It is followed by as many lines as necessary to define each point on the curve.

Line defining the curve name:

```
CURVE=Curve Name
```

Line defining a point on a curve:

```
Torque, Twist
```

Each curve must have at least two points defined. This type of torque-twist curve may not be associated with non-linear beam elements which are defined using the rigid riser format for geometric properties specification.

TABLE INPUT

Input:	Description
Curve Name:	The generic name of the torque-twist curve.
Torque:	A torque value for a point on the curve.
Twist:	The corresponding twist value.

NOTES

(a) This keyword is used to define generalised torque-twist curves that define GJ for a particular set of elements. Torque-twist curves may be assigned to element sets using the `*GEOMETRIC SETS` keyword.

(b) Use as many lines as you need to completely define a particular torque-twist curve. Simply leave Column 1 blank for second and subsequent lines. For subsequent torque-twist curves, put the curve name in Column 1 and specify the torque-twist data in the same way.

(c) The points defining the non-linear torque-twist curve may be specified in any order. Flexcom subsequently sorts the data pairs into ascending order of twist.

(d) If the twist in an element lies between the torque-twist data points you specify, Flexcom uses linear interpolation to determine the relevant stiffness for the element.

(e) If the twist in the element lies outside the specified range of the torque-twist curve, then Flexcom simply extrapolates from the first or last section of the curve as appropriate.

(f) If none of the specified twist terms have a negative value, the curve is assumed to be symmetrical about the origin (i.e. the behaviour of the element is the same for both positive and negative twists).

1.8.10.72 *TURBINE GEOMETRY

PURPOSE

To specify information relating to a wind turbine, including the hub, shaft, bearings and nacelle.

THEORY

Refer to Turbine Geometry for further information.

KEYWORD FORMAT

A block of lines specifying a variety of information relating to a wind turbine.

```
OVERHANG=Overhang distance
SHAFT TILT=Shaft tilt angle
TOWER TOP NODE=Tower top node (Number or Label)
TOWER TO SHAFT=Tower to shaft distance
YAW ANGLE=Yaw angle
AZIMUTH ANGLE=Azimuth Angle
HUB MASS=Hub mass
YAW BEARING MASS=Yaw bearing mass
FRONT BEARING MASS=Shaft front bearing mass
REAR BEARING MASS=Shaft rear bearing mass
NACELLE=Nacelle mass, Nacelle COM X distance, Nacelle COM Y distance, Nacelle
```

TABLE INPUT

Input:	Description
Overhang:	The distance along the (possibly tilted) rotor shaft between the tower centerline and hub center, measured positive downwind. Upwind rotors have a negative overhang. This is illustrated by 'Overhang' in Turbine Geometry and subsequent schematics
Shaft Tilt:	The angle in degrees between the rotor shaft and the horizontal plane. A positive shaft tilt angle means that the downwind end of the shaft is the highest. Upwind turbines have a negative shaft tilt angle for improved tower clearance. This is illustrated by 'ShftTilt' in Turbine Geometry and subsequent schematics.

Tower Top Node:	The location (node number/label) in the finite element model which corresponds to the tower top location.
Tower to Shaft:	The vertical distance from the top of the tower to the shaft axis. This is illustrated by 'Twr2Shft' in Conventional Upwind Turbine Layout.
Yaw Angle:	The initial yaw angle of the turbine about the vertical tower axis. It is positive counterclockwise when looking down on the turbine in plan view.
Azimuth Angle:	The initial azimuth angle for Blade 1. It is positive clockwise when looking downwind.
Hub Mass:	The mass of the hub. This is illustrated by 'Hub C.M.' in Conventional Upwind Turbine Layout.
Yaw Bearing Mass:	The mass of the yaw bearing. This is illustrated by 'Yaw Bearing C.M.' in Conventional Upwind Turbine Layout.
Shaft Front Bearing Mass:	The mass of the front shaft bearing. Note that the front bearing is currently located at a distance along the shaft which is equal to 90% of the overhang distance, in a direction towards the hub from the tower-shaft intersection point.
Shaft Rear Bearing Mass:	The mass of the rear shaft bearing. Note that the rear bearing is currently located at a distance along the shaft which is equal to 20% of the overhang distance, in a direction towards the nacelle from the tower-shaft intersection point.
Nacelle Mass:	The mass of the nacelle. This is illustrated by 'Nacelle C.M.' in Conventional Upwind Turbine Layout.

TABLE INPUT NACELLE COM

Input:	Description

Nacelle CoM X:	This input specifies the downwind distance to the nacelle centre-of-mass, measured from the tower top. See NacCMxn in Conventional Upwind Turbine Layout.
Nacelle CoM Y:	This input specifies the lateral distance to the nacelle centre-of-mass, measured from the tower top. See NacCMyn in Conventional Upwind Turbine Layout.
Nacelle CoM Z:	This input specifies the vertical distances to the nacelle centre-of-mass, measured from the tower top. See NacCMzn in Conventional Upwind Turbine Layout.

1.8.10.73 *TURBINE ROTOR

PURPOSE

To assemble a wind turbine rotor by selecting blades and specifying related information.

THEORY

Refer to Turbine Geometry for further information.

KEYWORD FORMAT

A block comprised of a:

Optional line defining the number of blades:

```
[NUMBER OF BLADES=Number of Blades]
```

Lines selecting the required blade profiles:

```
BLADE1=Blade 1 Name
[BLADE2=Blade 2 Name]
[BLADE3=Blade 3 Name]
```

Line defining the hub radius:

```
HUB RADIUS=Hub Radius
```

Line defining the precone angle of the blade:

```
PRECONE ANGLE=Precone Angle
```

Optional line defining the initial pitch angle of the blade:

```
[PITCH ANGLE=Initial Pitch Angle]
```

Optional line defining the inclusion of pitching-moment terms:

```
[USEBLCM=Include Pitching Moment]
```

Optional line selecting the blade modelling approach, be it rigid or flexible:

```
[BLADE MODEL=Blade Model]
```

Refer to the Turbine Geometry schematics for an illustration of the various inputs. *Number of Blades* defaults to 3 if not specified. Both *Blade 2 Name* and *Blade 3 Name* default to *Blade 1 Name* if not specified. *Initial Pitch Angle* defaults to 0 degrees if not specified. Set *Include Pitching Moment* to 1 to include the pitching-moment terms in the blade aerofoil aerodynamics, or 0 (the default) to neglect them. *Blade Model* can be `RIGID` or `FLEXIBLE`.

The rotor is assembled according to the blade selections and related information defined in the `*TURBINE ROTOR` keyword. Flexcom creates a node in the finite element discretisation corresponding to each blade span position defined in the `*BLADE GEOMETRY` keyword. These nodes are connected sequentially using finite elements whose structural properties are governed by the inputs in the `*BLADE STRUCTURE` keyword. The structural properties are assumed constant along each element, and determined using linear interpolation between the element centrepoint and the nearest available sectional definitions.

TABLE INPUT

Input:	Description
Number of Blades:	The number of blades, which can be 1, 2 or 3 (the default).
Blade 1:	The name of the first blade. Note that the same blade names should be used in the `*BLADE GEOMETRY`, `*BLADE STRUCTURE` and `*TURBINE ROTOR` keywords.
Blade 2:	The name of the second blade. This defaults to the first blade name if not specified.
Blade 3:	The name of the third blade. This defaults to the first blade name if not specified.

Hub Radius:	The radius to the blade root from the center-of-rotation along the (possibly preconed) blade-pitch axis. This is illustrated by 'HubRad' in Turbine Geometry and subsequent schematics.
Precone Angle:	The angle in degrees between a flat rotor disk and the cone swept by the blades, positive downwind. Upwind turbines have a negative precone angle for improved tower clearance. This is illustrated by 'Precone' in Turbine Geometry and subsequent schematics.
Pitch Angle:	The initial blade pitch angle in degrees, positive to feather, leading edge upwind. The axis of rotation is illustrated by 'Blade-Pitch Axis' in Turbine Geometry, while the angle convention is illustrated by 'm: Pitching' in Blade Local Coordinate System. The initial blade pitch angle defaults to 0 degrees if not specified.
Pitching-Moment:	This option allows you to include or exclude (the default) pitching-moment terms in the blade aerofoil aerodynamics.
Blade Model:	This option allows you to choose between a simplistic rigid (the default) or fully flexible model of the rotating blades. See Note (a).

NOTES

(a) Please note that Flexcom 2022.1 uses a simplified rotor nacelle assembly (RNA) model considers that blade deformations under applied loading are negligible, hence the blade geometries are approximated as rigid profiles. This means that the *Blade Model* input must be set to *Rigid* at the moment (i.e. `BLADE MODEL=RIGID`). Work is presently under way to develop a more detailed rotor nacelle assemble (RNA) model that explicitly models the blades with finite elements, thereby allowing blade deformations and rotational inertia to be accurately captured. We hope to make this feature available in the next program release.

(b) You can define as many blades as you wish in the `*BLADE GEOMETRY` and `*BLADE STRUCTURE` keywords. Only the blades which you reference in the `*TURBINE ROTOR` keyword will be used in the model.

(c) If you have selected the *Flexible* blade model, Flexcom creates a node in the finite element discretisation corresponding to each blade span position defined in the `*BLADE GEOMETRY` keyword. These nodes are connected sequentially using finite elements whose structural properties are governed by the inputs in the `*BLADE STRUCTURE` keyword. The structural properties are assumed constant along each element, and determined using linear interpolation between the element centrepoint and the nearest available sectional definitions.

1.8.10.74 *VECTOR

PURPOSE

To define vectors for use with the solution criteria feature.

THEORY

Refer to Solution Criteria Automation for further information on this feature.

KEYWORD FORMAT

The keyword starts with a line defining the name of the vector. This is then followed by a line with three numbers defining the vector. Any further axis systems or vectors can be defined by repeating the two lines as often as necessary.

```
VECTOR=Vector Name
X, Y, Z
```

Any vectors defined must have non-zero length.

TABLE INPUT

Input:	Description
Name:	A unique name for the vector being defined.
X:	The component in the global X-direction of the vector being defined.
Y:	The component in the global Y-direction of the vector being defined.

Z:	The component in the global Z-direction of the vector being defined.

1.8.10.75*VESSEL

PURPOSE

To is used to specify the initial position and undisplaced orientation of a vessel.

THEORY

Refer to Vessels and Vessel Motions for further information on this feature.

Note also that the old `*VESSEL` and `*RAO` keywords have effectively been superseded by the new `*VESSEL,INTEGRATED` keyword, which accepts RAO data also, thereby eliminating the need for a separate `*RAO` keyword – hence the 'integrated' nature of the new keyword. Refer to `*VESSEL,INTEGRATED` for further information.

KEYWORD FORMAT

An optional line to specify the formulation to be used to combine vessel rotations and to calculate the displacements of attached nodes from combined rotations. This is followed by a block of two lines defining a vessel, which is repeated as many times as necessary.

```
[ANGLES=Angle Theory Type]
VESSEL=Vessel Name
INITIAL POSITION=X Co-ordinate, Y Co-ordinate, Z Co-ordinate [, Initial Yaw]
```

Angle Theory Type can be LARGE (the default) or SMALL.

TABLE INPUT

Vessel Motions – Angles Type

Input:	Description
Angles Type:	The options are *Large Angles* (the default) and *Small Angles*.

Vessel – Initial Position

Input:	Description
Vessel:	The name of the attached vessel.
X:	The coordinate in the global X-direction of the initial position of the vessel reference point.
Y:	The coordinate in the global Y-direction of the initial position of the vessel reference point.
Z:	The coordinate in the global Z-direction of the initial position of the vessel reference point.
Theta:	The vessel initial undisplaced orientation, specified in degrees anticlockwise from the global Y-Direction. This defaults to a value of 0°.

1.8.10.76 *VESSEL,INTEGRATED

PURPOSE

To specify all information pertaining to a vessel or vessels.

THEORY

Refer to Vessels and Vessel Motions for further information on this feature.

Note also that the old `*VESSEL` and `*RAO` keywords have effectively been superseded by the new `*VESSEL,INTEGRATED` keyword, which accepts RAO data also, thereby eliminating the need for a separate `*RAO` keyword – hence the 'integrated' nature of the new keyword.

KEYWORD FORMAT

A block of lines defining all information pertaining to a vessel.

The block begins with a mandatory line defining the vessel name. This is followed by an optional line specifying the theory used to combine vessel rotations. Next comes some optional lines associating RAO data with the vessel. This is then followed by further optional lines associating a predefined profile with the vessel. The entire block may then be repeated for subsequent vessels.

Mandatory line defining the vessel name and its physical location:

```
VESSEL=Vessel Name
```

Optional line specifying the theory used to combine vessel rotations:

```
[ANGLES=Angles Theory]
```

Mandatory line defining the initial position of the vessel's focal point, and the initial yaw orientation of the vessel:

```
INITIAL POSITION=X, Y, Z [, Initial Yaw]
```

Optional line associating RAO data with the vessel:

```
[RAO=RAO File Name][, FORMAT=Format Name][, UNITS=Unit Type]
```

Optional line defining separation between the focal point and the RAO reference point (the point on the vessel for which the RAOs are defined), specified in the local vessel heave, surge and sway directions:

```
[REF=REF X, REF Y, REF Z]
```

Optional line associating a predefined profile with the vessel:

```
[PROFILE=Profile Type]
```

If the profile type is user-defined, then another mandatory line immediately follows:

```
FILE=Profile File Name
```

Optional line defining overall dimensions for the vessel profile:

```
[DIMENSIONS=Height, Length, Width]
```

Optional line defining separation between the focal point and the centre of the vessel profile, specified in the local vessel heave, surge and sway directions:

```
[COP=COP X, COP Y, COP Z]
```

`Angles Theory` may be *LARGE* (the default) or *SMALL*. `Initial Yaw` defaults to zero. `Angles Theory` may be *LARGE* (the default) or *SMALL*. If a file name or any part of its path contains spaces then it should be enclosed in double quotation marks. `Format Name` may be one of the standard program formats (MCS, AQWA, WAMIT, MOSES or ORCAFLEX) or the name of a customised format defined under `*RAO FORMAT`. *Unit Type* is only relevant to the standard program formats (apart the standard MCS format). `REF X, REF Y, REF Z, COP X, COP Y` and `COP Z` all default to zero. If dimensions are not explicitly defined, some generic dimensions are applied automatically to the body.

TABLE INPUT

Input:	Description
Vessel Name:	A descriptive name for the vessel.
Focal Point X, Y, Z:	The global X, Y, & Z coordinates of the vessel's focal point at solution initiation. Strictly speaking, it defines the point about which the initial yaw orientation of the vessel is applied. In many cases, the focal point will be coincident with both the RAO reference point (the point on the vessel for which the RAOs are defined) and the centre of the vessel profile (which provides enhanced visual appeal in the structural animation). See Note (a).
Yaw:	The initial undisplaced orientation of the vessel, specified in degrees anticlockwise from the global Y-axis. This defaults to a value of 0°.
RAO File:	The name of the ASCII file containing the RAO data.
Format:	The format of the RAO data. You may select from a range of standard program formats, or alternatively reference a customised RAO format, in which case you specify the name of a predefined *RAO Conversion*. See Note (b).

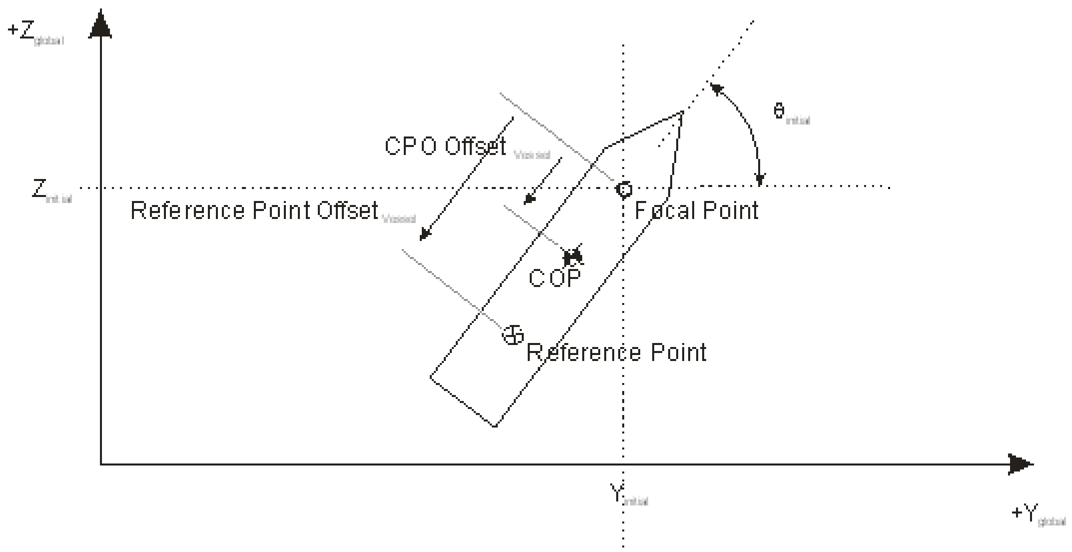
Units:	If you are using one of the standard program formats (apart the standard MCS format), it is advisable to explicitly specify the unit system used in the external RAO file. Flexcom will attempt to ascertain the unit system from the external file, but as this is not always possible, you are advised to explicitly state the unit system in order to avoid any possible ambiguity.
Reference Point Offset X, Y, Z:	The physical separation between the focal point and the RAO reference point (the point on the vessel for which the RAOs are defined), specified in the local vessel heave, surge and sway directions. See Note (a).
Profile:	The profile type used to represent the vessel. You may choose from a range of standard profiles, or opt to use a customised profile.
User Profile:	If you are using a customised profile, this input defines the location of the profile file. See Note (d).
Height, Length, Width:	The overall dimensions of the profile in the local heave, surge and sway axes. If dimensions are not explicitly defined, some generic dimensions are applied automatically to the vessel.
Centre of Profile Offset X, Y, Z:	The physical separation between the focal point and the centre of the vessel profile (which provides enhanced visual appeal in the structural animation), specified in the local vessel heave, surge and sway directions. See Note (a).
Angle Theory:	The theory used to combine vessel rotations. The options are *Large Angles* (the default) and *Small Angles*.

NOTES

(a) The *Focal Point* defines the initial position of the vessel at solution initiation. Strictly speaking, it defines the point about which the initial yaw orientation of the vessel is applied. By default, the focal point is coincident with the RAO reference point (the point on the vessel for which the RAOs are defined), but you may also define *Reference Point Offset* terms if this is not the case. Likewise, the vessel profile is also centred about the focal point (i.e. the origin used in the definition of the profile is assumed to be coincident with the focal point), but again you may define *Centre of Profile Offset* terms if this is not the case.

Note that the *Focal Point* entries define a physical location in global X, Y, & Z coordinates. Both the *Reference Point Offset* and the *Centre of Profile Offset* define physical separations with respect to the focal point, specified in the local vessel heave, surge and sway directions, respectively.

The following figure illustrates the various inputs graphically. It shows a turret moored vessel in plan view. Note that the initial location of the RAO reference point, and the initial configuration of the vessel profile, are both governed by the initial location of the focal point, the applied initial yaw orientation. This facility allows you to adjust the initial orientation of the turret moored vessel without affecting any attached risers, jumpers or mooring lines.



(b) Supported programs include AQWA, WAMIT, MOSES and OrcaFlex. If you specify any of these program names as an RAO format, the program's standard output file may be used directly as a source of RAO data. Flexcom understands the conventions used in these programs and automatically performs the relevant conversions for you. For further details regarding the conventions assumed for each of these programs, refer to RAO Conversions. MCS is also a pre-defined format, which represents the standard RAO conventions traditionally used by Flexcom.

If you RAO data does not correspond to any of these standard conventions, you may explicitly define some customised conventions via an RAO conversion (`*RAO FORMAT` keyword). In this case, you should define the RAO data in an ASCII file, in a layout similar to the standard MCS or Line layouts, and use the Format input above to reference the required RAO conversion. Refer to MCS Layout and Line Layout for further details.

(c) Flexcom uses linear interpolation to calculate RAOs and phase angles at wave headings and frequencies intermediate to those in the RAO file. Outside of the range of user-specified headings and frequencies, RAOs and phase angles are assumed to be zero, so it is important to ensure you cover the full range of conditions likely to be encountered in an analysis when inputting the RAO data.

In the standard MCS convention, wave heading is defined as the angle between the direction of approach of a wave harmonic incident on the vessel and the local surge axis. However, please note that in strict mathematical terms the incident wave heading is the angle between the wave direction drawn at the vessel reference point and the negative direction of the local surge axis. Phase angles are specified in degrees, and represent a phase lag or lead relative to the wave at the vessel reference point. A positive phase angle denotes a phase lag relative to the incident wave harmonic.

(d) The profile is defined in XML format, and several sample profiles are available in your local Flexcom installation directory, under the sub-folder 'StandardProfiles'. Each profile file contains a range of input data, capable of defining standard shapes such as boxes, cylinders etc., as well as arbitrary shapes defined by a series of points about which a mesh is constructed. All locations within the profile file are defined with respect to a local origin (0, 0, 0). The location of this origin in the global axis system is then defined by the *Focal Point* and C*entre of Profile Offset* entries as discussed in Note (a). Should you require any assistance in creating your own customised profile, feel free to contact sw.support@woodplc.com and our support team will discuss the format of the XML information in more detail.

1.8.10.77 *VISCOUS DRAG

PURPOSE

To define viscous drag for a floating body.

THEORY

Refer to Viscous Damping Coefficients for further information on this feature.

KEYWORD FORMAT

Block of data consisting of the floating body name followed by the viscous drag definition. The viscous drag matrix is defined by a block of 6 lines, preceded by a single line which specifies the velocity type used for calculation of viscous forces. The entire block of data can then be repeated to specify viscous drag for second and subsequent floating bodies.

Line defining floating body name:

```
FLOATING BODY=Floating Body Name
```

Line defining viscous drag type:

```
TYPE=MATRIX
```

Line defining velocity type used for calculation of viscous forces:

```
OPTION=ABSOLUTE/RELATIVE
```

Block of lines defining viscous drag matrix:

V_{11}, V_{12}, V_{13}, V_{14}, V_{15}, V_{16}
V_{21}, V_{22}, V_{23}, V_{24}, V_{25}, V_{26}
V_{31}, V_{32}, V_{33}, V_{34}, V_{35}, V_{36}
V_{41}, V_{42}, V_{43}, V_{44}, V_{45}, V_{46}
V_{51}, V_{52}, V_{53}, V_{54}, V_{55}, V_{56}
V_{61}, V_{62}, V_{63}, V_{64}, V_{65}, V_{66}

VISCOUS DRAG - DEFINE

Table Input

Input:	Description
Floating Body:	The name of the floating body.

Velocity Type:	This option allows you to specify the type of velocity to be used in the calculation of viscous drag forces. The options are the *Absolute* velocity of the floating body (default) or the *Relative* velocity between the floating body and the external fluid.
Viscous Drag:	The name of the *Absolute* or *Relative* velocity viscous drag matrix definition.

Notes

(a) The viscous drag matrix which you refer to for a particular floating body must be defined in the *Body - Viscous Drag Coefficients* table.

BODY - VICIOUS DRAG COEFFICIENTS

Table Input

Input:	Description
Name:	The name of the viscous drag coefficient matrix.
Matrix (6x6):	A 6x6 matrix of viscous drag coefficient data. See Note (a).

Notes

(a) The layout of the viscous drag coefficient terms is as follows.

$$
\begin{array}{llllll}
V_{11}, & V_{12}, & V_{13}, & V_{14}, & V_{15}, & V_{16} \\
V_{21}, & V_{22}, & V_{23}, & V_{24}, & V_{25}, & V_{26} \\
V_{31}, & V_{32}, & V_{33}, & V_{34}, & V_{35}, & V_{36} \\
V_{41}, & V_{42}, & V_{43}, & V_{44}, & V_{45}, & V_{46} \\
V_{51}, & V_{52}, & V_{53}, & V_{54}, & V_{55}, & V_{56} \\
V_{61}, & V_{62}, & V_{63}, & V_{64}, & V_{65}, & V_{66}
\end{array}
$$

Each input corresponds to a viscous damping coefficient times an appropriate area term related to the dimensions of the floating body. For example, V_{11} is the viscous damping coefficient in surge times the projected floating body area in the local surge direction.

1.8.10.78 *WAMIT

PURPOSE

To specify that Flexcom is to read floating body data from WAMIT output.

THEORY

Refer to WAMIT Interface for further information on this feature. Note that the WAMIT Interface is a legacy feature which has effectively been superseded by the newer Hydrodynamic Data Importer. This allows allows you to automatically import characteristic data relating to a Floating Body from a range of well-known hydrodynamic simulation packages.

KEYWORD FORMAT

A block of data consisting of TYPE=line to specify whether the data type is single or coupled. This is followed by a single floating body name, or the names of the coupled floating bodies, as appropriate. This is followed by a line to specify the method of RAO calculation, which in turn followed by a line to specify the WAMIT output file name.

Block defining single floating body with radiation RAO:

```
TYPE=SINGLE
FLOATING BODY=Floating Body Name
RAO=RADIATION/DIFFRACTION
FILE=Filename.wmt
```

Block defining coupled floating bodies:

```
TYPE=COUPLED
FLOATING BODY 1=Floating Body Name
FLOATING BODY 2=Floating Body Name
RAO=RADIATION/DIFFRACTION
FILE=Filename.wmt
```

TABLE INPUT

Body - WAMIT Import – Single

Input:	Description
Floating Body:	The name of the floating body for which data is to be retrieved from WAMIT output.

RAO Computation:	This option allows you to specify which force RAOs are to be retrieved from the WAMIT output file and used in the Flexcom analysis. The options are *Radiation* (the default) and *Diffraction*.
File Name:	The name of the WAMIT output file for this floating body

Body - WAMIT Import – Couple

Input:	Description
Floating Body 1:	The name of the first floating body for which data is to be retrieved from WAMIT output.
Floating Body 2:	The name of the second floating body for which data is to be retrieved from WAMIT output.
RAO Computation:	This option allows you to specify which force RAOs are to be retrieved from the WAMIT output file and used in the Flexcom analysis. The options are *Radiation* (the default) and *Diffraction*.
File Name:	The name of the WAMIT output file for this pair of floating bodies.

1.8.11 $MODES

This section corresponds to Modes, an ancillary module of Flexcom, which performs modal analysis. Refer to Modal Analysis for further details.

This section contains the following keywords:

- `*EIGENPAIRS` is used to specify the required number of natural frequencies, and also parameters relating to the subspace iteration algorithm.
- `*ELEMENT SETS` is used to group individual elements into element sets.

- *EXCLUDE MODES is used to specify modes to be excluded from SHEAR7 output.
- *FRICTION is used to specify longitudinal and transverse seabed friction stiffnesses.
- *INCLUDE MODES is used to specify modes to be included in SHEAR7 output.
- *NAME is used to specify a title for a Modes analysis run.
- *OUTPUT OPTIONS is used to specify the eigensolution details to be routed to the output file.
- *PROPERTIES is used to assign effective structural properties to element sets for use in calculating stresses.
- *REPEAT is used to specify that the present Modes analysis is a Repeat Run.
- *REPLACE MODES is used to specify modes to be replaced in SHEAR7 output.
- *RESTART is used to indicate that a Modes analysis is to be restarted from a previous Flexcom run.
- *RISER TYPE is used to specify the riser type and other parameters relating to the Modes facility for generating output for subsequent input to SHEAR7.

1.8.11.1 *EIGENPAIRS

PURPOSE

To specify the required number of natural frequencies, and also parameters relating to the subspace iteration algorithm.

THEORY

Refer to Mathematical Background for further information on this feature.

KEYWORD FORMAT

A single line defining the relevant parameters.

```
No. of Eigenpairs[, Subspace Iteration Convergence
Tolerance] [, Maximum No. of Subspace Iterations] [,
Eigenproblem Shift]
```

`Subspace Iteration Convergence Tolerance` defaults to $1x10^{-6}$. `Maximum No. of Subspace Iterations` defaults to 50. `Eigenproblem Shift` defaults to 1.

TABLE INPUT

Input:	Description
No. of Eigenpairs:	Required number of solution eigenpairs (natural frequencies and mode shapes). See Note (a).
Subspace Iteration Convergence Tolerance:	Subspace iteration convergence tolerance. This entry has a default value of $1x10^{-6}$. See Notes (b) and (c).
Maximum No. of Subspace Iterations:	Maximum number of subspace iterations. This entry defaults to 50 iterations. See Notes (b) and (c).
Eigenproblem Shift:	A shift to apply to the eigenproblem. See Note (d).

NOTES

(a) When specifying the required number of natural frequencies it is recommended that you specify twice the actual required number you are interested in. For example, if the first 5 natural frequencies are of interest, 10 eigenpairs should be requested. This ensures that the actual required values are estimated to an acceptable accuracy.

(b) Convergence of the subspace iteration procedure is deemed to have been achieved when the normalised change in the eigensolution subspace between successive iterations is less than a specified tolerance. This tolerance is the *Subspace Iteration Convergence Tolerance* specified here. The *Maximum No. of Subspace Iterations* input prevents indefinite program looping in non-convergent analyses.

(c) The default values for the convergence measure and maximum number of iterations are sufficient in the vast majority of cases. However, in a small number of cases where the solution is very sensitive, you may need to vary these values.

(d) The basic equation that Modes solves is K v = λ M v, where K and M are stiffness and mass matrices respectively, v is an eigenvector, and λ is the corresponding eigenvalue. Shifting by a value μ means transforming this equation into (K + m M) v = η M v. The eigenvalues of the original and transformed equations are related by $\eta_i = \lambda_i + \mu$ for all i eigenvalues. Eigenvectors are identical. Bathe et al., 1976 claim that applying a shift to an eigenproblem can speed convergence and prevent problems when the stiffness matrix is positive semidefinite. By default Modes applies a shift of 1. In a very small number of analyses increasing this value can guarantee convergence which might otherwise not be achieved, but this option should be very rarely used.

1.8.11.2 *ELEMENT SETS

PURPOSE

To group individual elements into element sets.

THEORY

Refer to Geometric Properties for further information on this feature.

KEYWORD FORMAT

Blocks of lines that define a set repeated as often as necessary. Each block begins with a line defining the set name. This is followed by various types of lines that define the elements in the set. These lines can be mixed and repeated as often as necessary until every element in the set is defined.

Line to define set name:

```
SET=Set Name
```

Line containing a list of elements. This line can contain up to 20 elements (numbers or labels). Any further elements must be defined on a new line.

```
List of Elements (Numbers or Labels)
```

Line defining a sequence of elements:

```
GEN=Start Element (Number or Label), End Element (Number or Label) [, Element
```

Line referencing another set of elements:

```
SUBSET=Subset Set Name
```

The set `ALL` is predefined and cannot be redefined. Every element assigned to a set must be defined using [`*ELEMENT`](). Set names are up to 256 characters long, can include spaces and are case insensitive. If you specify an element label rather than an element number, it must be enclosed in {} brackets. *`Element Increment`* defaults to one.

TABLE INPUT

Input:	Description
Set Name:	A unique label for the element set. Set names are not case sensitive, so the set name 'RISER' is equivalent to 'Riser', which is in turn equivalent to 'riser'.
Elements:	The elements comprising the set. These can be input in three ways, namely: (i) A list of elements (numbers or labels), such as for example "1, 5, 7". (ii) A group of consecutive elements (numbers or labels), input using the format: "11 – 15". This definition specifies Elements 11 to 15 inclusive. The specification "11 - 15 – 2" can be used to specify the Elements 11, 13 and 15 - that is, from Element 11 to Element 15 in steps of 2. (iii) Another set name. For example you might define three sets named SET_1, SET_2 and SET_3, and then combine them in a further set, say ALLSETS, by inputting "SET_1, SET_2, SET_3". If you specify an element label rather than an element number, it must be enclosed in {} brackets. All three specifications can be combined, as for example in "1, 7, 9, 12-15, 17, 20-50-10, RISER". This set combines Elements 1, 7, and 9; elements 12 to 15 inclusive; Element 17; Elements 20, 30, 40 and 50; and the elements comprising the set RISER.

SubSets:	An additional element set or sets whose elements are to be added to the current set definition. If more than one set is referenced, use commas to separate out the set names.

NOTES

(a) Use as many lines as you need to completely define the elements comprising a particular set. Simply leave the first column blank for second and subsequent lines.

(b) If a set name is included in the specification of another set, then obviously the elements comprising that set must be separately defined.

(c) There is one predefined element set in Flexcom, which is named All. Not surprisingly this comprises all of the elements of the finite element discretisation and is the default element set. Note that Flexcom will resist any attempt to redefine the make-up of the set All.

(d) The names and composition of element sets you define in a preceding section carry through to all dependent (restart) sections. For example, if you define an element set in the `$MODEL` section, it will automatically be available in a subsequent `$DATABASE POSTPROCESSING` section. So there is no need to repeat the specifications again – you can just use the set names directly. If you redefine the composition of a set previously defined in a preceding section, Flexcom will output a warning, but will continue with the most recent set definition.

1.8.11.3 *EXCLUDE MODES

PURPOSE

To specify modes to be excluded from SHEAR7 output.

THEORY

Refer to Identifying Mixed Modes for further information on this feature.

KEYWORD FORMAT

A block of two lines, with the second line repeated as often as necessary.

```
[TYPE=Plane Type]
Excluded Mode, Excluded Mode, Excluded Mode, etc.
Excluded Mode, Excluded Mode, Excluded Mode, etc.
```

The TYPE= line is only relevant for SCRs. `Plane Type` can be either IN-PLANE or OUT-OF-PLANE. Up to 20 excluded modes may be specified on a single line.

TABLE INPUT

SCR Modes - Exclude

Input:	Description
Type:	The options are *In-Plane* and *Out-of-Plane*.
Exclude Modes:	Modes to be excluded from the SHEAR7 output. You can list modes on one or more lines. Up to 20 excluded modes may be specified on a single line.

TTR Modes - Exclude

Input:	Description
Exclude Modes:	Modes to be excluded from the SHEAR7 output. You can list modes on one or more lines. Up to 20 excluded modes may be specified on a single line.

1.8.11.4 *FRICTION

PURPOSE

To specify longitudinal and transverse seabed friction stiffnesses.

THEORY

Refer to Seabed Modelling in Modal Analysis for further information on this feature.

KEYWORD FORMAT

One line containing the friction stiffness values:

```
Longitudinal Stiffness, Transverse Stiffness
```

This keyword is only relevant where the initial static analysis has at least one non-zero seabed friction coefficient.

TABLE INPUT

Input:	Description
Longitudinal Friction Stiffness:	The value of the longitudinal friction stiffness. This defaults to a value of 0.
Transverse Friction Stiffness:	The value of the transverse friction stiffness. This defaults to a value of 0.

NOTES

(a) This table is only relevant for a modal analysis, where the initial static analysis has at least one non-zero seabed friction coefficient.

(b) You can model friction in a modal analysis using either a total restraint, or a partial restraint modelled using a spring stiffness. For more information, please refer to Motion in the Plane of the Seabed.

(c) By default, seabed friction continues to be modelled in the same way as in earlier program versions (the so-called *Fully Restrained Model*). This corresponds to the situation where both stiffness entries in this table are zero.

(d) To apply a partial restraint in either the longitudinal or transverse direction, you input a non-zero stiffness in this table in the appropriate direction.

(e) Note however that a non-zero value is immaterial and unused if the corresponding friction coefficient was zero in your initial static analysis. In that case, free motion in the appropriate direction is permitted.

1.8.11.5 *INCLUDE MODES

PURPOSE

To specify modes to be included in SHEAR7 output.

THEORY

Refer to Identifying Mixed Modes for further information on this feature.

KEYWORD FORMAT

A block of two lines, repeated as often as necessary.

```
[TYPE=Plane Type]
Included Mode, Included Mode, Included Mode, etc.
```

The TYPE= line is only relevant for SCRs. `Plane Type` can be either IN-PLANE or OUT-OF-PLANE. Up to 20 included modes may be specified on a single line.

TABLE INPUT

SCR Modes - Include

Input:	Description
Type:	Whether the mode to be included is in-plane or out-of-plane.
Include Modes:	Modes to be included in the SHEAR7 output.

TTR Modes - Include

Input:	Description
Include Modes:	Modes to be included in the SHEAR7 output.

NOTES

(a) You use these tables to instruct Modes to include in SHEAR7 output modes that it would otherwise exclude.

1.8.11.6 *NAME

PURPOSE

To specify a title for a Modes analysis run.

THEORY

Refer to Modal Analysis for further information on this feature.

KEYWORD FORMAT

A single line containing the analysis name.

Name

TABLE INPUT

Input:	Description
Title:	A descriptive title to be associated with the Modes analysis. This entry is optional.

1.8.11.7 *OUTPUT OPTIONS

PURPOSE

To specify the eigensolution details to be routed to the output file.

THEORY

Refer to Additional Output and Repeat Runs for further information on this feature.

KEYWORD FORMAT

A block of four or more lines as follows:

```
[SET=Set Name]
[FACTOR=Scale Factor]
[Static Solution Flag]
```

The fourth and any subsequent lines may be in either of two formats:

```
List of Eigenpairs
```

or

```
GEN=Start Eigenpair, End Eigenpair [, Eigenpair Increment]
```

Set Name defaults to all elements (set ALL). *Scale Factor* defaults to 1.0. *Static Solution Flag* must be 0 if present. The list of eigenpairs can contain up to 20 eigenpair numbers. *Eigenpair Increment* defaults to 1.

TABLE INPUT

Input:	Description

Eigenpairs:	The eigenpairs to include in the output. You can either list these individually, separated by commas or spaces, or request a series of eigenpairs by inputting start and end numbers separated by a hyphen. Both specifications can be combined, as for example in: 1 3 6 10-15 Here details are requested of Eigenpairs 1, 3, 6 and 10 to 15 inclusive. Note that this entry is actually optional. If you invoke the *Output Options - Specify* facility but leave this row blank, then the output file contains details for all eigenpairs in the solution.
Static Solution:	This option is used to signal to Modes that you want details in the output file of the static solution used to determine the stiffness matrix K. The default is No, meaning there are no static solution details in the output file. See Note (b).
Set Name:	The set of elements for which output data is required. This input is optional, and defaults to all elements of your finite element model. See Note (c).
Scale Factor for Stress Output:	A scale factor to be applied to stresses output to the output file. This input is optional, and defaults to 1. See Note (d).

NOTES

(a) The use of this facility is optional. By default the output file contains only a data echo section and a summary list of natural frequencies. A table of maximum curvature for each mode will also be included if your analysis generated SHEAR7 output.

(b) You can request static solution details in a modal analysis. Obviously the static solution will have been performed by Flexcom.

(c) The element set you specify in *Set Name* row can be defined as part of your Modes input data, using the *Define Element Sets* table. Alternatively, it can have been previously defined as part of your Flexcom data.

(d) The *Scale Factor for Stress Output* scales stresses that are included in the static solution details if requested. Stresses are also included in the eigensolution details for individual eigenpairs if the Modes run generates SHEAR7 output, in which case the *Scale Factor for Stress Output* also multiplies the calculated stresses.

1.8.11.8 *PROPERTIES

PURPOSE

To assign effective structural properties to element sets for use in calculating stresses.

THEORY

Refer to Stress Properties for further information on this feature.

KEYWORD FORMAT

A block of two lines repeated as often as necessary. Each property specified on the second line can be omitted, in which case a default value will be used for stress calculations:

```
SET=Set Name
Do, Di, A, Iyy, Izz
```

The properties specified here are used in calculating stresses during modal analysis. This keyword is optional and may be ignored if you wish to use the actual properties (e.g. via the `*GEOMETRIC SETS` keyword in the `$MODEL` section), or properties explicitly specified for stress computations (i.e. via the corresponding `*PROPERTIES` keyword in the `$MODEL` section), as the values specified in the model definition are carried through to Modes.

If any of the parameters `A`, `Iyy` or `Izz` are omitted, default values are computed based on the specified (or default) values of `Do` and `Di`.

TABLE INPUT

Input:	Description

Set Name:	The name of the element set to which the stress properties are to be assigned. This defaults to all elements.
Do:	The effective outside diameter for the elements of the set. The default depends on the format used to specify geometric data. If you used the Flexible Riser Format, then the default is the drag diameter for the elements of the set. If you used the Rigid Riser Format or Mooring Line Format, then Do here defaults to Do input in the properties data
Di:	The effective internal diameter for the elements of the set. The default is the internal diameter specified in the geometric data.
A:	The effective cross-sectional area for the elements of the set. The default is given by: $$A = \frac{\pi\left(D_o^2 - D_i^2\right)}{4}$$ where Do and Di are the inputs described above, default or otherwise.
Iyy:	The second moment of area about the local y-axis for the elements of the set. The default is given by: $$I_{yy} = \frac{\pi\left(D_o^4 - D_i^4\right)}{64}$$ where Do and Di are the inputs described above, default or otherwise.
Izz:	The second moment of area about the local z axis for the elements of the set. The default is Iyy above.

NOTES

(a) This keyword is identical to the `*PROPERTIES` keyword of the main `$MODEL` section. If you use the keyword in `$MODEL` to specify stress properties, then these automatically carry through to Modes and you do not need to repeat the specification here. Refer to Stress Properties for a detailed discussion of the various parameters above.

(b) Modes outputs stress data to the program output file only if you use the *Output Options - Specify* facility to include details of the static solution in this output. Stresses are also included in the eigensolution details for individual eigenpairs if the Modes run generated SHEAR7 output. So the use of this option is meaningless unless either of these conditions apply.

1.8.11.9 *REPEAT

PURPOSE

To specify that the present Modes analysis is a Repeat Run.

THEORY

Refer to Additional Output and Repeat Runs for further information on this feature.

KEYWORD FORMAT

This keyword does not contain any further data.

TABLE INPUT

Input :	Description
Type :	The options are *New Run* (the default) and *Repeat Run*.

NOTES

(a) The default of *New Run* means Modes performs an eigensolution in full. The alternative of *Repeat Run* is appropriate if you want to alter only details of the output produced by your Modes analysis, in which case it is not necessary to perform the eigensolution in full again.

1.8.11.10 *REPLACE MODES

PURPOSE

To specify modes to be replaced in SHEAR7 output.

THEORY

Refer to Identifying Mixed Modes for further information on this feature.

KEYWORD FORMAT

A block of two lines, repeated as often as necessary.

```
[TYPE=Plane Type]
Replaced Mode, Substitute Mode
```

The TYPE= line is only relevant for SCRs. `Plane Type` can be either IN PLANE or OUT OF PLANE.

TABLE INPUT

SCR Modes - Replace

Input:	Description
Type:	Whether the mode to be replaced is in-plane or out of plane.
Mode:	The mode to be replaced.
Replace With:	The mode that will replace the previously specified mode.

TTR Modes - Replace

Input:	Description
Mode:	The mode to be replaced.
Replace With:	The mode that will replace the previously specified mode.

NOTES

(a) You use these tables when you want to 'replace' rather than 'exclude' a specific mode from SHEAR7 output. Replacing a mode means its natural frequency is retained in the SHEAR7 output, but its mode shape is replaced by that of another mode you nominate.

1.8.11.11 *RESTART

PURPOSE

To indicate that a Modes analysis is to be restarted from a previous Flexcom run.

THEORY

Refer to Restart Analyses for further information on this feature.

KEYWORD FORMAT

A single line defining the file name.

```
LAST=File Name
```

TABLE INPUT

Input:	Description
Restart File:	The name of the Flexcom analysis from which the Modes run is to be restarted. This may be input in terms of the analysis name (e.g. *Example1*) or the full path of the analysis (e.g. *C:\Flexcom\Examples\Example1*).

1.8.11.12 *RISER TYPE

PURPOSE

To specify the riser type and other parameters relating to the Modes facility for generating output for subsequent input to SHEAR7.

THEORY

Refer to SHEAR7 Interface for further information on this feature.

KEYWORD FORMAT

A block of three or more lines. The format of the third and subsequent lines depends on the riser type, which is specified on the first line of the keyword.

A block of two lines defining the riser type, element set and VIV option

```
TYPE=Riser Type [, VIV=VIV Option][, TOLERANCE=SCR Plane Tolerance]
```

[SET=*Set Name*]

For TTRs and SCRs, a third and final line is used to specify the number of segments, the number of modes, and a SHEAR7 damping ratio.

No. of Segments, No. of Modes [, *SHEAR7 Damping Ratio*]

For the User riser type option, a third line is used to specify the number of segments and a SHEAR7 damping ratio. This may be followed by two types of lines that may be mixed and/or repeated as often as necessary.

Line to specify the number of segments and a SHEAR7 damping ratio:

No. of Segments [, *SHEAR7 Damping Ratio*]

Line requesting a single mode:

Mode Number

Line requesting a list of modes:

GEN=*Start Mode, End Mode* [, *Mode Increment*]

Riser Type can be SCR, TTR or USER. *VIV Option* is only relevant to SCRs, and may be either *IN-PLANE* (default) or *OUT-OF-PLANE*. *SCR Plane Tolerance* is only relevant to SCRs and defaults to 1×10^{-5}. *Set Name* defaults to all elements (set ALL). *SHEAR7 Damping Ratio* is only relevant to wake interference analyses, and defaults to zero. *Mode Increment* defaults to 1.

RISER TYPE

Purpose

To specify the riser type relating to the Modes facility for generating output for subsequent input to SHEAR7.

Table Input

Input:	Description
Riser Type:	The options are *SCR* (the default), *TTR* and *User*.

VIV DATA - SCR & TTR RISERS

Purpose

To specify various parameters relating to the Modes facility for generating output for subsequent input to SHEAR7.

Table Input

Input:	Description
No. of Modes:	The number of the natural frequencies/mode shapes calculated by Modes to be considered for SHEAR7 output. The default value of 0 signifies that all of the natural frequencies and mode shapes are to be considered, otherwise, the lowest *No. of Modes* natural frequencies/mode shapes are considered.
No. of Segments:	The number of equally spaced segments into which the riser element set is to be divided. A non-zero positive integer value is required.
Set Name:	The set of elements to include in the SHEAR7 output. This input is optional, and can be omitted. See Notes (a) and (b).
SHEAR7 Damping Ratio:	A damping ratio for use with a subsequent SHEAR7 analysis. This input is optional, and can be omitted. See Note (c).
Modes for VIV Analysis:	The SCR modes to be considered for subsequent SHEAR7 analysis. The options are In-Plane (the default) and Out-of-Plane. See Note (d).
SCR Plane Tolerance:	The tolerance value used to determine whether nodal displacements lie in the plane of the SCR (or at 90° to it). This entry has a default value of 1×10^{-5}. See Note (e).

Notes

(a) The element set you specify in the *Set Name* entry can be defined as part of your Modes input data, using the *Define Element Sets* table. Alternatively, it can have been previously defined as part of your Flexcom data.

(b) If you do not specify an element set, the composition of the default set depends on the *Riser Type* under consideration. If the riser is a top tensioned riser (*TTR*) or a user-defined riser type, then the default is all elements of the finite element mesh. If the riser type is a catenary riser (*SCR*), then the default is all elements which are not wholly on the seabed, starting at that element whose first node is the SCR touchdown point.

(c) This input is only relevant to wake interference analyses, where Flexcom can automatically run a SHEAR7 analysis to compute enhanced drag and lift coefficients.

(d) This input is only relevant to SCRs. Modes routes SCR SHEAR7 data to two files, named common.inp (for in-plane modes) and common.out (for the out-of-plane modes). The reason is that the two sets of modes will be excited by different current distributions and so will typically be considered in separate SHEAR7 runs. When doing your SHEAR7 analysis, you will need to rename one or other common.mds as appropriate. You can also use the *Modes for VIV Analysis* input to specify whether you are interested in in-plane or out-of-plane modes, in which case the relevant common file will be renamed automatically for you.

(e) The *SCR Plane Tolerance* input is only relevant to SCRs. When determining whether a mode is In-Plane, Out-of-Plane or Unknown, Modes first determines modal displacements relative to the local SCR axis system. This axis system is defined such that (i) the SCR local X-axis is aligned with the global X-axis, and (ii) the SCR local Y-axis represents the intersection of the horizontal plane with a vertical plane which contains the end points of the SCR. The next step is a normalisation of the local modal displacement vectors at each finite element node. Three conditions are then possible:

i. If all three components of the modal displacement vector (at any finite element node) are larger than the SCR Plane Tolerance value, then the mode is deemed to be Mixed/Unknown.

ii. If all local X and Y displacements are less than the SCR Plane Tolerance value, then the mode is deemed to be Out-of-Plane.

iii. If all local Z displacements are less that the SCR Plane Tolerance value then the mode is deemed to be In-Plane.

VIV DATA - USER RISERS

Purpose

To specify various parameters relating to the Modes facility for generating output for subsequent input to SHEAR7.

Table Input

Input:	Description
No. of Segments:	The number of equally spaced segments into which the riser element set is to be divided. A non-zero positive integer value is required.
Set Name:	The set of elements to include in the SHEAR7 output. This input is optional, and can be omitted. See Notes (a) and (b).
SHEAR7 Damping Ratio:	A damping ratio for use with a subsequent SHEAR7 analysis. This input is optional, and can be omitted. See Note (c).

Notes

(a) The element set you specify in the *Set Name* entry can be defined as part of your Modes input data, using the *Define Element Sets* table. Alternatively, it can have been previously defined as part of your Flexcom data.

(b) If you do not specify an element set, the composition of the default set depends on the *Riser Type* under consideration. If the riser is a top tensioned riser (*TTR*) or a user-defined riser type, then the default is all elements of the finite element mesh. If the riser type is a catenary riser (*SCR*), then the default is all elements which are not wholly on the seabed, starting at that element whose first node is the SCR touchdown point.

(c) This input is only relevant to wake interference analyses, where Flexcom can automatically run a SHEAR7 analysis to compute enhanced drag and lift coefficients.

USER RISER MODES - REQUEST SINGLE

Purpose

To request a single mode to be included in the SHEAR7 output modes.

Table Input

Input:	Description
Mode Number:	The requested mode number.

USER RISER MODES - REQUEST MULTIPLE

Purpose

To request multiple modes to be included in the SHEAR7 output modes.

Table Input

Input:	Description
Start Mode:	The number of the first mode in the series.
End Mode:	The number of the last mode in the series.
Increment:	The mode number increment to be used in assigning numbers to the generated modes. This input defaults to a value of 1, which will apply in the majority of cases.

Notes

(a) Mode numbers are assigned by repeatedly adding the specified or default increment to the previous number (obviously starting at the start mode) until the end mode number is reached or exceeded. So for example, the specification Start Mode - 1, End Mode - 11, Increment - 2 will result in 6 modes, numbered 1, 3, 5, 7, 9 and 11.

1.8.12 $MODES POSTPROCESSING

This section corresponds to the postprocessing of modal analyses. Refer to Modal Analysis Postprocessing for further details.

This section contains the following keywords:

- `*ELEMENT SETS` is used to group individual elements into element sets.
- `*PLOT` is used to specify selected modes for plotting, and the type of plot required.

- `*RESTART` is used to indicate that a Modes postprocessing run is to be restarted from a modal analysis file of different stub name.

1.8.12.1 *ELEMENT SETS

PURPOSE

To group individual elements into element sets.

THEORY

Refer to Geometric Properties for further information on this feature.

KEYWORD FORMAT

Blocks of lines that define a set repeated as often as necessary. Each block begins with a line defining the set name. This is followed by various types of lines that define the elements in the set. These lines can be mixed and repeated as often as necessary until every element in the set is defined.

Line to define set name:

```
SET=Set Name
```

Line containing a list of elements. This line can contain up to 20 elements (numbers or labels). Any further elements must be defined on a new line.

```
List of Elements (Numbers or Labels)
```

Line defining a sequence of elements:

```
GEN=Start Element (Number or Label), End Element (Number or Label) [, Element
```

Line referencing another set of elements:

```
SUBSET=Subset Set Name
```

The set `ALL` is predefined and cannot be redefined. Every element assigned to a set must be defined using `*ELEMENT`. Set names are up to 256 characters long, can include spaces and are case insensitive. If you specify an element label rather than an element number, it must be enclosed in {} brackets. *Element Increment* defaults to one.

TABLE INPUT

Input:	Description

Set Name:	A unique label for the element set. Set names are not case sensitive, so the set name 'RISER' is equivalent to 'Riser', which is in turn equivalent to 'riser'.
Elements:	The elements comprising the set. These can be input in three ways, namely: (i) A list of elements (numbers or labels), such as for example "1, 5, 7". (ii) A group of consecutive elements (numbers or labels), input using the format: "11 – 15". This definition specifies Elements 11 to 15 inclusive. The specification "11 - 15 – 2" can be used to specify the Elements 11, 13 and 15 - that is, from Element 11 to Element 15 in steps of 2. (iii) Another set name. For example you might define three sets named SET_1, SET_2 and SET_3, and then combine them in a further set, say ALLSETS, by inputting "SET_1, SET_2, SET_3". If you specify an element label rather than an element number, it must be enclosed in {} brackets. All three specifications can be combined, as for example in "1, 7, 9, 12-15, 17, 20-50-10, RISER". This set combines Elements 1, 7, and 9; elements 12 to 15 inclusive; Element 17; Elements 20, 30, 40 and 50; and the elements comprising the set RISER.
SubSets:	An additional element set or sets whose elements are to be added to the current set definition. If more than one set is referenced, use commas to separate out the set names.

NOTES

(a) Use as many lines as you need to completely define the elements comprising a particular set. Simply leave the first column blank for second and subsequent lines.

(b) If a set name is included in the specification of another set, then obviously the elements comprising that set must be separately defined.

(c) There is one predefined element set in Flexcom, which is named *All*. Not surprisingly this comprises all of the elements of the finite element discretisation and is the default element set. Note that Flexcom will resist any attempt to redefine the make-up of the set *All*.

(d) The names and composition of element sets you define in a preceding section carry through to all dependent (restart) sections. For example, if you define an element set in the `$MODEL` section, it will automatically be available in a subsequent `$DATABASE POSTPROCESSING` section. So there is no need to repeat the specifications again – you can just use the set names directly. If you redefine the composition of a set previously defined in a preceding section, Flexcom will output a warning, but will continue with the most recent set definition.

1.8.12.2 *PLOT

PURPOSE

To specify selected modes for plotting, and the type of plot required.

THEORY

Refer to Modal Displacement and Curvature for further information on this feature.

KEYWORD FORMAT

A block of two lines to request a plot repeated as often as necessary.

```
TYPE=PLOT TYPE [, SET=Set Name]
Mode Number
```

PLOT TYPE can be DISPLACEMENT or CURVATURE. *Set Name* defaults to all elements (set ALL).

TABLE INPUT

Input:	Description
Modes:	The number of the mode for which a plot is required.

Type:	The options are *Displacement* (the default) and *Curvature*.
Set Name:	The set of elements for which output data is required. This input is optional, and defaults to all elements of your finite element model.

1.8.12.3 *RESTART

PURPOSE

To indicate that a Modes postprocessing run is to be restarted from a modal analysis file of different stub name.

THEORY

Refer to Restart Analyses for further information on this feature.

KEYWORD FORMAT

A single line defining the file name.

```
LAST=File Name
```

The keyword is optional, and is not required if the `$MODES POSTPROCESSING` section is contained within the same file as the actual modal analysis data.

TABLE INPUT

Input:	Description
Restart File:	The name of the modal analysis from which the Modes postprocessing run is to be restarted. This may be input in terms of the analysis name (e.g. *Example1*) or the full path of the analysis (e.g. *C:\Flexcom\Examples\Example1-Static*).

1.8.13 $PREPROCESSOR

This section corresponds to the keyword parameterisation facility, which allows you to model a series of load case variations about a base model. Refer to Keyword Parameterisation for further details.

This section contains the following keywords:

- `*COMBINATIONS` is used to control the names and locations of generated keyword files, along with the required analysis combinations.
- `*EXCEL VARIATIONS` is used to generate keyword files based on a parameter matrix contained in an Excel workbook.
- `*PARAMETERS` is used to define parameters whose names may be referenced subsequently in the definition of other input variables.
- `*VARIATION` is used to define parameter or keyword variations.

1.8.13.1 *COMBINATIONS

PURPOSE

To control the names and locations of generated keyword files, along with the required analysis combinations.

THEORY

Refer to Parameters for further information on this feature.

KEYWORD FORMAT

The keyword begins with an optional line defining the base directory for the generated keyword files. This is followed by a block of lines to define the names and locations of generated keyword files, along with the required analysis combinations. The block begins with a line providing a descriptive comment on the analysis combination, a subdirectory for the generated keyword files, and the file names themselves. This is then followed by one or more lines defining the analysis combinations, via a list of variation names. The entire block may then be repeated for further analysis combinations if required.

Optional line defining the base directory for the generated keyword files.

```
[DIRECTORY=Base Directory]
```

Block of lines to define the names and locations of generated keyword files, along with the required analysis combinations.

Line providing a descriptive comment on the analysis combination, a subdirectory for the generated keyword files, and the file names themselves.

```
DESCRIPTION=Description [, SUBDIRECTORY=Subdirectory] [,
FILENAME=File Name]
```

Line defining the analysis combinations. Every variation may be included in the combination using `ALL`. This line defaults to “VARIATIONS=ALL” if not specified by the user.

```
[VARIATIONS=ALL]
```

Otherwise, up to 20 variation names may be specified on a single line. This is normally sufficient, but the line may be repeated if necessary.

```
VARIATIONS=Variation Name, Variation Name, Variation Name, etc.
```

Base Directory defaults to the directory of the base keyword file if omitted. Although both *Subdirectory* and *File Name* are optional inputs, at least one of these entries should be specified in order to create unique names or locations for the generated keyword files. These entries may contain combinations of standard characters and parameter values (denoted via *%Parameter Name%*). If omitted, *Subdirectory* is the same as the base directory (i.e. no subdirectory is created). If omitted, *File Name* is the same as the base keyword file name. Where there is a conflict between generated names, a counter will be appended to the file name (or directory name) to ensure unique names are used.

Refer to ‘Notes’ below for further details.

TABLE INPUT

Base Directory

Input:	Description
Base Directory:	The base directory for the generated keyword files. This entry is optional and defaults to the directory of the base keyword file if omitted.

Combinations

Input:	Description

Description:	A descriptive comment on the analysis combination. This comment will be included in all generated keyword files for this analysis combination.
Subdirectory:	A subdirectory for the generated keyword files. This entry may contain combinations of standard characters and parameter values. See Note (a).
File Name:	The names of the generated keyword files. This entry may contain combinations of standard characters and parameter values. See Note (a).
Included Variations:	Either ALL or a list of the analysis combinations, defined in terms of *Variation* names. This entry is optional and defaults to ALL if omitted. You can specify up to 20 names per line. This is normally sufficient, but the line may be repeated if necessary. If you are using more than one line for Included Variations, you should leave the preceding columns blank for the second and subsequent lines.

NOTES

(a) The *Subdirectory* and *File Name* entries may contain combinations of standard characters and parameter values (denoted via *%Parameter Name%*). For example, if you had a base file name entitled “Offset.key”, you might wish to create variations such as “25m Offset.key”, “50m Offset.key” etc., in “Near” and “Far” subfolders. This could be achieved by specifying entries such as “Near\%Offset%m Offset” (*Subdirectory*), and “%Offset%m Offset” (*File Name*), assuming an Offset parameter has been defined in the *Parameter Variations* table. The quotation marks must be included in these two entries.

1.8.13.2 *EXCEL VARIATIONS

PURPOSE

To generate keyword files based on a parameter matrix contained in an Excel workbook.

THEORY

Refer to Spreadsheet Based Variations for further information on this feature.

KEYWORD FORMAT

The keyword begins with an optional line defining the base directory for the generated files. This is followed by an optional line to define the names and locations of the generated keyword files. Finally there is a line containing information about the Excel file and the data within it.

Optional line defining the base directory for the generated keyword files.

```
[DIRECTORY=Base Directory]
```

Block of lines to define the names and locations of generated keyword files, along with the required analysis combinations.

Optional line providing a subdirectory for the generated keyword files, and the file names themselves.

```
[SUBDIRECTORY=Subdirectory] [, FILENAME=File Name]
```

Line defining the Excel file name, the sheet name (which defaults to the first sheet), the range where the parameter data is stored, and information about the orientation of the data.

```
EXCEL=Excel Workbook, [SHEET=Worksheet Name], RANGE=Cell Range, [ORIENTATION=
```

`Base Directory` defaults to the directory of the base keyword file if omitted. Although both `Subdirectory` and `File Name` are optional inputs, at least one of these entries should be specified in order to create unique names or locations for the generated keyword files. These entries may contain combinations of standard characters and parameter values (denoted via *%Parameter Name%*). If omitted, `Subdirectory` is the same as the base directory (i.e. no subdirectory is created). If omitted, `File Name` is the same as the base keyword file name. Where there is a conflict between generated names, a counter will be appended to the file name (or directory name) to ensure unique names are used.

`Worksheet Name` is optional and defaults to the first sheet in the Excel workbook. Data Orientation is optional and defaults to HORIZONTAL, meaning parameter names are arranged horizontally across the first row of the data matrix. If ORIENTATION=VERTICAL, then the first column of the data matrix is assumed to contain parameter names.

Refer to ‘Notes’ below for further details.

TABLE INPUT

Input:	Description
Subdirectory:	A subdirectory for the generated keyword files. This entry may contain combinations of standard characters and parameter values. See Note (a).
Generated File Name:	The names of the generated keyword files. This entry may contain combinations of standard characters and parameter values. See Note (a).
Excel Workbook:	The name of the Excel workbook containing the parameter matrix.
Worksheet Name:	The name of the worksheet in the workbook containing the parameter matrix. This entry is optional and defaults to the first sheet in the workbook.
Cell Range:	The range of cells within the worksheet which define the parameter matrix. This must be a valid Excel range specification (e.g. “A1:F22”, “$G$17:$L$40” etc.).
Data Orientation:	The orientation of the parameter matrix. The default is HORIZONTAL, meaning parameter names are arranged horizontally across the first row.
Base Directory:	The base directory for the generated keyword files. This entry is optional and defaults to the directory of the base keyword file.

NOTES

(a) The *Subdirectory* and *File Name* entries may contain combinations of standard characters and parameter values (denoted via %Parameter Name%). For example, if you had a base file name entitled “Offset.key”, you might wish to create variations such as “25m Offset.key”, “50m Offset.key” etc., in “Near” and “Far” subfolders. This could be achieved by specifying entries such as “Near\%Offset%m Offset” (*Subdirectory*), and “%Offset%m Offset” (*File Name*), assuming an Offset parameter has been defined in the Parameter Definitions table. The quotation marks must be included in these two entries.

(b) If a file name includes spaces (e.g. sample 1.xlsx) then it must be enclosed by inverted commas.

1.8.13.3 *PARAMETERS

PURPOSE

To define parameters whose names may be referenced subsequently in the definition of other input variables.

THEORY

Refer to Parameters and Equations for further information on this feature.

KEYWORD FORMAT

Two types of lines that may be mixed and/or repeated as often as necessary.

A line defining a single parameter:

`Parameter Name, Value`

A line defining a list type parameter, that generates a number of single parameters:

`Parameter Name Prefix, /List of Named and/or Unnamed Values/`

The list part of the list type parameter must be enclosed in a pair of forward slashes. The *`List of Named and/or Unnamed Values`* is a comma or space separated list of parameters to be generated from the list. For *`Named Values`* the semicolon is used to separate the name from the value and the name of the resultant parameter is a concatenation of the *`Parameter Name Prefix`* and the name of the *`Named Value`*. For *`Unnamed Values`* the resultant parameter name is a concatenation of the *`Parameter Name Prefix`* and the index of the *`Unnamed Value`* in the list.

There can be as many *`Named and/or Unnamed Values`* in the list as desired. Equations may be used to define the values. If the value is a character expression (as opposed to a numerical value or equation) and contains a space, tab or a comma, then it must be enclosed in quotation marks.

For example the following list type parameter:

```
Point, /X:11, Y:22, Z:=[Point_X + 10], "unnamed value one", "unnamed value tw
```

results in the following individual parameters generated as if they were specified as separate single parameters:

```
Point_X, 11
Point_Y, 22
Point_Z, =[Point_X + 10]
Point_1, "unnamed value one"
Point_2, "unnamed value two"
```

Refer to 'Notes' below for further details.

TABLE INPUT

Input:	Description
Parameter:	The name of the parameter being defined. See Note (a) below for details.
Value:	The value of the parameter being defined. Both single and list type parameters may be defined as described above. See Notes (a) and (b) below for details.

NOTES

(a) The parameter name and the named value name should not contain white spaces or tabs and must start with a letter character.

(b) If the value is a character expression (as opposed to a numerical value or equation) that contains spaces or tabs then it must be enclosed in quotation marks.

(c) When referencing a parameter, the variable definition must be preceded by the characters `=[` and followed by the `]` character. Specifically the variable definition will appear in the format `=[Parameter]`, rather than just having an explicitly specified value. For example, you could define a parameter called `"OceanDepth"`, assign it a numerical value, and reference it subsequently using `"=[OceanDepth]"` rather than explicitly specifying a numerical value.

1.8.13.4 *VARIATION

PURPOSE

To define parameter or keyword variations.

THEORY

Refer to Keyword Based Variations for further information on this feature.

KEYWORD FORMAT

A block of two lines to define the relevant variation. The block begins with a line defining the variation name. The following lines are used to define the actual variations. The format of these lines depend on whether parameter or keyword variations are being defined. The entire block may then be repeated as often as necessary to define all the required variations.

Line defining the variation name:

```
NAME=Variation Name
```

Line defining parameter variations explicitly. This line may be repeated if required.

```
PARA=Parameter Name, VALUE=Value 1 [, Value 2] [, Value 3] etc.
```

Line defining a range of parameter variation via generation option. This line may be repeated if required.

```
PARA=Parameter Name, GEN=Start Value, End Value [, Increment] etc.
```

Line defining a keyword variation. This line may be repeated if required.

```
ACTION=ADD/REMOVE, KEYWORD=Keyword Name [, FILE=File Name] [, SECTION=Sectio
```

Any `Parameter Name` referenced must be defined under `*PARAMETERS`. Generation option is only valid for numerical values. `Increment` defaults to 1.

Refer to the ‘Notes’ section below for further details.

TABLE INPUT

Parameter Variations

Input:	Description

Variation Name:	The name of the variation being defined.
Parameter Name:	The name of the parameter being varied.
Value(s):	The value or values being assigned to the parameter. These can be input in two different ways, namely: (i) A list of values, such as for example "4, 5, 6". (ii) A group of consecutive values, input using the format: "7 – 9". This definition specifies values of 7 to 9 inclusive. The specification "7 – 11 – 2" can be used to specify the values 7, 9 and 11 - that is, from 7 to 11 in steps of 2. Naturally if the value is a character expression (as opposed to a numerical value or equation), the consecutive type input is not relevant. Note also that character expression must be closed in quotation marks.

Keyword Variations

Input:	Description
Variation Name:	The name of the variation being defined.
Action:	The action to be taken on the keyword. The options are Add (the default) and *Remove*. Note that if the keyword is already present, the Add option essentially means *Replace*.
Keyword:	The relevant keyword in question.
File:	The file containing the keyword. This entry is optional and is only relevant when the *Action* is set to *Add*. When the action is set to *Remove*, naturally the *File* in question is the current one.

Section:	The section of the file containing the keyword. This entry is optional and is only relevant when the *Action* is set to *Add*. If omitted, the first occurrence of the keyword in the file is taken if the keyword appears more than once (but this would be unusual).

NOTES

(a) Variations in particular parameters (e.g. vessel offset terms) would typically be the most commonly used variations. However, for complete generality, it is also possible to add or remove entire keywords. This facility is intended to cater for variations which are difficult to accommodate using parameter variations, but the option is probably rarely invoked in practice. For example, if you wished to examine a different wave discretisation option, it might be more straightforward to replace the relevant wave loading keyword, rather than attempting to parameterise several different options within it.

1.8.14 $SHEAR7

This section corresponds to prediction of VIV fatigue damage by interfacing with SHEAR7 program developed by a team at MIT. Refer to SHEAR7 Interface for further details.

This section contains the following keywords:

- `*CURRENT` is used to specify the current to be considered in the SHEAR7 analysis.
- `*CUTTOFF MODES` is used to define the SHEAR7 parameters “Power Cutoff” and “Primary Zone Amplitude Limit”. These parameters are applicable to SHEAR7 4.5 and later. The SHEAR7 User Manual provides a detailed description of these parameters.
- `*DAMPING RATIO` is used to define the SHEAR7 structural damping ratio.
- `*FATIGUE DATA` is used to assign fatigue data to SHEAR7 sets.
- `*FATIGUE OPTIONS` is used to specify fatigue calculation options for SHEAR7 in Block 5.
- `*FOLDER OPTIONS` is used to indicate where the SHEAR7 input files will be generated and run.

- `*HIGHER HARMONICS` is used to define the higher harmonics amplification factor and threshold values.
- `*NAME` is used to specify a title for SHEAR7 analysis run.
- `*NON-ORTHOGONAL DAMPING` is used to define the Non-orthogonal Damping calculation flag.
- `*OUTPUT FILES` is used to request output file types from SHEAR7. The available output file extensions are *.anm, *.scr, *.dmg, *.fat, *.str, *.out, *.out1, and *.out2. Refer to the SHEAR7 User Manual for further information on the output file generation options. The output file options can vary between versions of SHEAR7.
- `*POWER RATIO EXPONENT` is used to define the type of the Power Ratio Exponent for SHEAR7.
- `*REFERENCE DIAMETER` is used to define the Reference Diameter value.
- `*RESPONSE` is used to define the SHEAR7 response definition/resolution to be used in the output file.
- `*RESTART` is used to indicate which modal analysis should be used in order to generate the SHEAR7 model.
- `*S-N CURVE` is used to define S-N curves to be used in SHEAR7 analysis.
- `*SECTION COEFFICIENTS` is used to assign SHEAR7 coefficients to element sets.
- `*SECTION PARAMETERS` is used to assign SHEAR7 parameters to element sets.
- `*SLIP-STICK HYSTERESIS` is used to specify various parameters relating to the SHEAR7 Slip-Stick Hysteresis module.
- `*STRESS TIME HISTORY OUTPUT` is used to specify various parameters relating the stress time history output file (*.s7sth).
- `*VERSION` is used to provide Flexcom with basic SHEAR7 interface information.
- `*ZONES` is used to define element sets that will be used as one zone in SHEAR7.

1.8.14.1 *CURRENT

PURPOSE

To specify the current to be considered in the SHEAR7 analysis.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A block of lines defining the current profile to be used in the SHEAR7 block 3 data.

A line with three optional entries:

```
[PROBABILITY=Probability of Occurrence] [, ID=Profile ID][, OPTION=Specificat
```

A line repeated as often as necessary to define the current profile:

```
Elevation, Velocity
```

The `Probability of Occurrence` of the current profile is an annual probability which must be between 0.0 and 1.0. The default value is 1.0. In SHEAR7 this value is multiplied by the computed damage rate before it is given in the output file. The `Profile ID` is reported in the output file to help users with profile bookkeeping and is not used explicitly by the SHEAR7. The default value for `Profile ID` is 1. See the SHEAR7 User Manual for further information.

At least two points must be defined for a piecewise linear current. The `Specification Option` can be one of three options:

- ASCEND (the default): In term above the datum X=0.
- DESCEND: In terms of distance below the MWL.
- XL : In terms of x/L specification as used by SHEAR7.

TABLE INPUT

CURRENT PROBABILITY

Input:	Description
Probability of Occurrence:	Probability of occurrence of the current profile per year. The default value is 1.0.
Profile ID:	A user specified current profile ID. The default value is 1.

ELEVATION SPECIFICATION

Input:	Description
Elevation	Elevation above the datum.
Velocity	The horizontal current velocity at this point.

DEPTH SPECIFICATION

Input:	Description
Depth	Depth in terms of distance below the MWL.
Velocity	The horizontal current velocity at this point.

X/L SPECIFICATION

Input:	Description
x/L	x/L as used by SHEAR7.
Velocity	The horizontal current velocity at this point.

1.8.14.2 *CUTOFF MODES

PURPOSE

To define the SHEAR7 parameters “Power Cutoff” and “Primary Zone Amplitude Limit”. These parameters are applicable to SHEAR7 version 4.5 and later.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A single line with two optional entries:

```
[Power Cutoff] [, Primary Zone Amplitude Limit]
```

The allowed range for `Power Cutoff` is 0 to 1.0 and the default value is 0.05. The allowed range for `Primary Zone Amplitude Limit` is 0 to 1.0 and the default value is 0.3.

TABLE INPUT

Input:	Description
Power Cut-off Level:	Power Cutoff parameter for SHEAR7 input file. The allowed range is 0 to 1.0 and the default value is 0.05. See the SHEAR7 User Manual for further details.
Primary Zone Amplitude Limit:	Primary Zone Amplitude Limit parameter for SHEAR7 input file. The allowed range is 0 to 1.0 and the default value is 0.3. See the SHEAR7 User Manual for further details.

1.8.14.3 *DAMPING RATIO

PURPOSE

To define the SHEAR7 structural damping ratio.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

One line containing the desired structural damping ratio:

```
Structural Damping Ratio
```

The allowed range for the Structural Damping Ratio is 0 to 1.0 and the default value is 0.003. For more guidance on suggest values refer to the SHEAR7 User Manual.

TABLE INPUT

Input:	Description
Structural Damping Ratio:	Structural Damping Ratio for SHEAR7 input file. The allowed range is 0 to 1.0 and the default value is 0.003.

1.8.14.4 *FATIGUE DATA

PURPOSE

To assign fatigue data to SHEAR7 sets.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A line defining the global stress concentration factor (SCF).

```
GLOBAL_SCF=SCF Value
```

A block of two lines to specify the fatigue data for a set. The first line defines the name of the set to which the fatigue data will be assigned. The second line specifies the fatigue data for the fatigue sets in the following format.

```
SET=Set Name
CURVE=S-N Curve Name
```

Set Name must be a set that appears in your static model. S-N Curve Name must be defined using *S-N CURVE.

TABLE INPUT

Input:	Description
Global SCF:	The stress concentration factor (SCF) to be used in fatigue calculations.
Set Name:	The set to which the fatigue properties are to be assigned. If elements included in the model are not defined here, Flexcom will assume to correspond to the first S-N curve specified.
S-N Curve:	The name of a defined S-N curve.

1.8.14.5 *FATIGUE OPTIONS

PURPOSE

To specify fatigue calculation options for SHEAR7 in Block 5.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A single line with four optional entries:

```
[ZERO-CROSSING=Zero-Crossing Method Flag] [, FIRST_MODE=First Mode In-line Fa
```

The Zero-Crossing and First Mode flags can be either YES or NO to indicate whether the specific calculation type is requested or not. The default value for both of flags is NO. These flags are required for SHEAR7 Version 4.7 and above. The Beta Control Number must be an integer from 0 to 10. The default value for this parameter is 4. This flag is required for SHEAR7 Version 4.8 and above. The Riser Fatigue Diameter is a flag used to identify which diameter is considered for riser fatigue. The default value for this parameter is OUTER. This flag is required for SHEAR7 Version 4.8 and above.

TABLE INPUT

Input:	Description
Zero-Crossing:	Flag for calculating fatigue with zero-crossing method (YES or NO). The default value is NO. SHEAR7 defines this line as “flag for calculating fatigue with zero crossing method”. YES will correspond to a value of 1. NO will correspond to a value of 0. See the SHEAR7 User Manual for further details. Note (a).
First Mode:	Flag for calculating first mode in-line fatigue damage (YES or NO). The default value is NO. SHEAR7 defines this line as “flag for calculating first mode in-line fatigue damage”. YES will correspond to a value of 1. NO will correspond to a value of 0. See the SHEAR7 User Manual for further details. Note (a).
Beta Control Number:	This value is used by SHEAR7 as an upper limit for the number of iterations used to calculate beta. A default of 4 beta iteration will be used for SHEAR7 V4.8. If a value of 0 is used, no beta iterations will be performed which is consistent with previous versions (SHEAR7 v4.7 and older). See the SHEAR7 User Manual for further details.

Riser Fatigue Diameter:	Flag for specifying which diameter (INNER or OUTER) should be used by SHEAR7 for fatigue calculations. The default value is OUTER. SHEAR7 defines this line as “flag for selecting the riser diameter for fatigue”. OUTER will correspond to a value of 0. INNER will correspond to a value of 1. See the SHEAR7 User Manual for further details.

NOTES

(a) It is worth noting that zero-crossing method and first mode in-line fatigue cannot be run together.

1.8.14.6 *FOLDER OPTIONS

PURPOSE

To indicate where the SHEAR7 input files will be generated and run.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A single line defining the folder path name:

```
FOLDER=Folder Path
```

This path will be the location of where SHEAR7 input is placed and where the SHEAR7 analysis will be performed. If the folder does not exist, it will be created. If this keyword is not specified, a default subdirectory named “Shear7” will be created and used.

TABLE INPUT

Input:	Description

Folder:	The folder location where the SHEAR7 input should be generated and SHEAR7 analysis performed. This may be input in terms of a relative path name (e.g. \Example1-SHEAR7) or the full folder path (e.g. C:\Flexcom\Examples\Example1-SHEAR7).

1.8.14.7 *HIGHER HARMONICS

PURPOSE

To define the SHEAR7 higher harmonics amplification factor and threshold values. This value is relevant to Shear 7 V4.9 onwards.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

One line containing the desired amplification factor and threshold values:

```
Amplification Factor, Threshold
```

To run the program without activating the Higher Harmonic correction the Higher Harmonics amplification factor, α, should be set to zero. For more guidance on suggested values refer to the SHEAR7 User Manual.

TABLE INPUT

Input:	Description
Amplification Factor:	Higher Harmonics amplification factor.. See the SHEAR7 User Manual for further details.

Threshold:	Higher Harmonics threshold, above which a Higher Harmonics amplification factor gets applied. The default value is 0.4. See the SHEAR7 User Manual for further details.

1.8.14.8 *NAME

PURPOSE

To specify a title for SHEAR7 analysis run.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A single line containing the SHEAR7 analysis name:

```
Name
```

TABLE INPUT

Input:	Description
Title:	A descriptive title to be associated with the SHEAR7 analysis. This entry is optional.

1.8.14.9 *NON-ORTHOGONAL DAMPING

PURPOSE

To set the SHEAR7 Non-orthogonal Damping calculation flag. This value is relevant to Shear 7 V4.10 onwards.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

One line containing the desired Non-Orthogonal Damping flag value:

```
ENABLED=Non-orthogonal Damping Flag
```

A YES or NO flag to enable the non-orthogonal damping when calculating the VIV response. See the SHEAR7 User Manual for further details.

TABLE INPUT

Input:	Description
Non-Orthogonal Damping Flag:	YES/NO -- SHEAR7 will/will not model the non-orthogonal damping terms. See the SHEAR7 User Manual for further details.

1.8.14.10 *OUTPUT FILES

PURPOSE

To request output file types from SHEAR7. The output file options can vary between versions of SHEAR7.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A single line with all optional entries, specifying the extensions of the output files to be generated by SHEAR7:

[ANM=*ANM Flag*] [, SCR=*SCR Flag*] [, DMG=*DMG Flag*] [, FAT=*FAT Flag*] [, STR=*STR

The flags can be either YES or NO to indicate whether the file is requested or not. The default for *SCR Flag*, *DMG Flag*, and *OUT Flag* is YES. The default for *ANM Flag, FAT Flag, STR Flag, OUT1 Flag, OUT1_OUT2 Flag, CURV Flag, ZETAHYST Flag* and *STH Flag* is NO.

TABLE INPUT

Input:	Description
ANM File:	Flag to request MATLAB animation file (YES or NO). The default value is NO. SHEAR7 defines this line as “flag for MATLAB animation data output”. YES will correspond to a value of 1. NO will correspond to a value of 0.
SCR File:	Flag to request a debugging file (YES or NO). The default value is YES. SHEAR7 defines this line as “flag for generating *.scr file”. YES will correspond to a value of 1. NO will correspond to a value of 0.
DMG File:	Flag to request file listing the Rayleigh fatigue per year for all resonant mode numbers, at each node (YES or NO). The default value is YES. SHEAR7 defines this line as “flag for generating *.dmg file”. YES will correspond to a value of 1. NO will correspond to a value of 0.
FAT File:	Flag to request file containing the total response amplitude and phase angle for every node, for every response frequency (YES or NO). The default value is NO. SHEAR7 defines this line as “flag for generating *.fat file”. YES will correspond to a value of 1. NO will correspond to a value of 0.
STR File:	Flag to request file containing RMS stress for each participating resonant mode (YES or NO). The default value is NO. SHEAR7 defines this line as “flag for generating *.str file”. YES will correspond to a value of 1. NO will correspond to a value of 0.

OUT1 File:	Flag to request a file containing the lift coefficient, non-dimensional frequency, and reduced velocity in the power-in zones (YES or NO). The default value is NO.
OUT1_OUT2 File:	Flag to request the *.out1 file and an additional *.out2 file containing the lift coefficient, non-dimensional frequency, and reduced velocity in the power-in modes at each node of the structure (YES or NO). The default value is NO.
CURV File:	Select *Yes* if you would like Shear7 to generate the *.s7curv file. This file reports the peak curvature of each resonant mode for each node.
ZETA-HYST File:	Select *Yes* if you would like Shear7 to generate the .s7zeta-hyst file. This file reports the stick-slip damping ratio for a range of tensions and curvatures.
STH File:	Select *Yes* if you would like Shear7 to generate the stress time history output file, *.s7sth. The output is stress versus time for each node in the modeled structure. This information can be post-processed into a loading histogram, for inclusion in external fatigue calculations.

1.8.14.11 *POWER RATIO EXPONENT

PURPOSE

To define the type of the Power Ratio Exponent for SHEAR7.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A single line specifying the type of power ratio exponent:

```
TYPE=Power Ratio Exponent Type
```

Power Ratio Exponent Type can be either POWER RATIO or EQUAL PROBABILITIES. The default type is set to POWER RATIO.

TABLE INPUT

Input:	Description
Power Ratio Exponent Type:	The Power Ratio Exponent type for SHEAR7 analysis. The default type is set to Power Ratio. SHEAR7 sometimes refers to this value as “power value exponent”. POWER RATIO will correspond to a value of 1 (default). EQUAL PROBABILITIES will correspond to a value of 0. See SHEAR7 User Manual for more details.

1.8.14.12 *REFERENCE DIAMETER

PURPOSE

To define the SHEAR7 Reference Diameter value. This value is relevant to Shear 7 V4.9 onwards.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

One line containing the desired reference diameter value:

```
Reference Diameter
```

The diameter provided is used in calculating the non-dimensional parameter A_f*. See the SHEAR7 User Manual for further details.

TABLE INPUT

Input:	Description
Reference Diameter:	The diameter provided is used in calculating the non-dimensional parameter A_f*. See the SHEAR7 User Manual for further details.

1.8.14.13 *RESPONSE

PURPOSE

To define the SHEAR7 response definition/resolution to be used in the output file.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

One line containing the desired response definition/resolution for output:

```
Response Definition
```

The allowed range for Response Definition is 0.0 to 1.0 and the default value is 0.1.

TABLE INPUT

Input:	Description
Response Definition:	Output response definition/resolution parameter. The allowed range is 0.0 to 1.0 and the default value is 0.1. See the SHEAR7 User Manual for further details.

1.8.14.14 *RESTART

PURPOSE

To indicate which modal analysis should be used in order to generate the SHEAR7 model.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A single line defining the file name:

```
MODES=File Name
```

The element set used to generate your *.mds file will be consistently used to construct the elements used in SHEAR7.

TABLE INPUT

Input:	Description
Modal Analysis File:	The name of the modal analysis from which the SHEAR7 model should be based on. This may be input in terms of the analysis name (e.g. Example1-Modal) or the full path of the analysis (e.g. C:\Flexcom\Examples\Example1-Modal).

1.8.14.15 *S-N CURVE

PURPOSE

To define S-N curves to be used in SHEAR7 analysis.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

Blocks of lines that define an S-N curve repeated as often as necessary. Each block begins with a line defining the S-N curve name and endurance limit. This is then followed by lines to define the S-N curve.

Line to define S-N curve name and type:

```
CURVE=S-N Curve Name, ENDURANCE_LIMIT=Stress Range
```

Lines to define a curve directly in terms of (S, N) data pairs (the line is repeated as often as necessary):

```
S, N
```

The whole block can be repeated to define multiple curves.

TABLE INPUT

Input:	Description
Curve Name:	A unique name for the S-N curve. See Note (a).
Endurance Limit:	A stress range value below which no fatigue damage occurs, regardless of the number of cycles. The default value is 0.
S:	A stress range value. See Note (a). Stress ranges are considered to be ksi for imperial units and pascals for metric units.
N:	The number of cycles to cause failure at this stress range. See Note (a).

NOTES

(a) As many as 10 segments can be used to define a particular S-N curve. It should be noted that for a single segment S-N curve. There are 2 S-N pairs required, for 2 segments S-N curve it is required to have 3 S-N pairs, etc. On a log-log plot the SN curve is assumed to be linear between these points. The stress range for the different points must be in increasing order. See the SHEAR7 User Manual for further details.

1.8.14.16*SECTION COEFFICIENTS

PURPOSE

To assign SHEAR7 coefficients to element sets.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A block of two lines defining the section coefficients. The block may be repeated to allow for the specification of different coefficients to different sets.

Two lines that make up the block:

```
SET=Set Name
[Normal Added Mass][, Damping Coefficient 0][, Damping Coeffient 1] [, Dampin
```

If not specified, the *Normal Added Mass* will default to that specified in the static model. *Damping Coefficient 0* is the Reynolds Still Water Damping Coefficient which defaults to 1.0. This specification is required for SHEAR7 Version 4.8 and above. If this value is specified for older version of the software, the value will be ignored. *Damping Coefficient 1* is the A/D Still Water Damping Coefficient which defaults to 0.20. *Damping Coefficient 2* is the Low Relative Velocity (V_R) Region Damping Coefficient which defaults to 0.18. *Damping Coefficient 3* is the High Relative Velocity (V_R) Region Damping Coefficient which defaults to 0.20. *Damping Coefficient 4* is the Axial Flow Region which defaults to 0.0. This specification is required for SHEAR7 Version 4.8 and above. If this value is specified for older version of the software, the value will be ignored.

TABLE INPUT

Input:	Description
Set Name:	The element set to which the coefficients are to be assigned. This defaults to all elements.

Added Mass Coefficient:	The Normal Added Mass coefficient to be assigned to this set. If a value is not specified, the normal added mass coefficient specified in the Model section is used.
Damping Coefficient 0:	Reynolds Still Water Damping Coefficient to be assigned to this set. The default value for this coefficient is 1.0. See the SHEAR7 User Manual for further details. See Note (a) and (b).
Damping Coefficient 1:	A/D Still Water Damping Coefficient to be assigned to this set. The default value for this coefficient is 0.20. See the SHEAR7 User Manual for further details. See Note (a).
Damping Coefficient 2:	Low Relative Velocity (VR) Region Damping Coefficient to be assigned to this set. The default value for this coefficient is 0.18. See the SHEAR7 User Manual for further details. See Note (a).
Damping Coefficient 3:	High Relative Velocity (VR) Region Damping Coefficient to be assigned to this set. The default value for this coefficient is 0.20. See the SHEAR7 User Manual for further details. See Note (a).
Damping Coefficient 4:	Axial Flow Region Damping Coefficient to be assigned to this set. The default value for this coefficient is 0.0. See the SHEAR7 User Manual for further details. See Note (a) and (b).

NOTES

(a) All default damping coefficients are based on SHEAR7's recommendation for a bare riser.

(b) These coefficients were introduced in SHEAR7 Version 4.8. If these values are specified for old versions, the values will be ignored.

1.8.14.17 *SECTION PARAMETERS

PURPOSE

To assign SHEAR7 parameters to element sets.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A block of two lines defining the section parameters. The block may be repeated to allow for the specification of different parameters to different sets.

Two lines that make up the block:

```
SET=Set Name
[Reduced Velocity Bandwidth][, Strouhal Number][, CL Reduction Factor] [, Zo
```

TABLE INPUT

Input:	Description
Set Name:	The element set to which the parameters are to be assigned. This defaults to all elements.
Reduced Velocity Bandwidth:	The Reduced Velocity Bandwidth (dVR) is the width of the band, expressed as a fraction of the critical relative velocity that will be used to define which possible modes could lock-in. The value defaults to 0.4 if not specified. See the SHEAR7 User Manual for further details See Note (a).
Strouhal Number:	The Strouhal number uniquely defines the relationship of flow velocity and cylinder diameter to the local vortex shedding frequency. The default value for this coefficient is 0.18. See the SHEAR7 User Manual for further details. See Note (a).

CL Reduction Factor:	The CL Reduction Factor allows the user to modify the lift coefficient iteration scheme. The lift coefficient reduction factor is multiplied by the CL value in the CL tables. The default value for this coefficient is 1.0. See the SHEAR7 User Manual for further details. See Note (a).
Zone CL Type:	The Lift Coefficient Table specifies which CL table from the common.cl table. The default value for this coefficient is 1. See the SHEAR7 User Manual for further details. See Note (a).

NOTES

(a) All default parameters are based on SHEAR7's recommendation for a bare riser.

1.8.14.18 *SLIP-STICK HYSTERESIS

PURPOSE

To specify various parameters relating to the SHEAR7 Slip-Stick Hysteresis module. This keyword is relevant for Shear7 V4.11 and later versions.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

Several optional lines relating to the Slip-Stick Hysteresis module:

```
[MODEL_HYSTERESIS=Model Slip-Stick Hysteresis]
[INPUT_FILE=Hysteresis Input File Name]
[TOLERANCE=Convergence Tolerance]
```

`Model Slip-Stick Hysteresis` can be `NO` (the default) or `YES`. `Hysteresis Input File Name` can be explicitly specified or left blank, in which case it defaults to the same name as the Flexcom keyword file name, but with a .s7inhyst file extension. `Convergence Tolerance` defaults to 0.001 if unspecified.

TABLE INPUT

Input:	Description
Model Slip-Stick Hysteresis:	Select *Yes* if you would like Shear7 to account for damping from stick-slip hysteresis when calculating the VIV response.
Hysteresis Input File Name:	The name of the file containing the input hysteresis curves. If this entry is left blank, the hysteresis file name defaults to the same name as the Flexcom keyword file name, but with a .s7inhyst file extension. If specifying a file name explicitly, do not include any file extension, and this will be added automatically for you.
Convergence Tolerance:	The convergence tolerance defaults to 0.001 if unspecified. The iteration convergence tolerance is calculated in Shear7 as the percentage change of the stick-slip damping coefficient.

1.8.14.19 *STRESS TIME HISTORY OUTPUT

PURPOSE

To specify various parameters relating the stress time history output file (*.s7sth). This keyword is relevant for Shear7 V4.11 and later versions.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

Several optional lines relating to the stress time history output file :

```
[TOTAL_TIME=Total Time]
[SAMPLE_PERIOD=Sample Period]
[NODE_RANGES=Node Ranges]
[RANDOM_SEED=Random Seed]
```

`Node Ranges` can be specified as a series of nodes separated by commas (with no spaces), or as ranges, so 1:5 would be the same as 1,2,3,4,5. Here is a sample specification-> `NODE_RANGES=1:5,251,1884`. If `Node Ranges` are not specified, Shear7 will outputs results for all nodes, and will issue a a warning saying that this could result in a very large .s7sth file.

TABLE INPUT

Input:	Description
Total Time:	The total time required for stress output. If not specified, Shear7 will use a total time which ensures there are at least 10 periods for the mode with the longest period.
Sample Period:	The sample time for stress output. If not specified, Shear7 will use a sample time which ensures at least 10 samples per period for the mode with the shortest period.
Node Ranges:	A list of nodes used to to calculate and output the time histories. This can be specified as a series of nodes separated by commas (with no spaces). They can also be specified as ranges so 1:5 would be the same as 1,2,3,4,5. Here is a sample specification-> 1:5,251,1884. If not specified, Shear7 will outputs results for all nodes, and will issue a a warning saying that this could result in a very large .s7sth file.
Random Seed:	For each repeat period the order of the frequencies gets randomly allocated. A seed number can be specified to ensure the ordering is repeatable.

1.8.14.20 *VERSION

PURPOSE

To provide Flexcom with basic SHEAR7 interface information.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A number of optional lines to specify basic information required for the interface:

```
[SHEAR7_PATH=Full Path of SHEAR7 Executable]
[SHEAR7_VERSION=Version Number]
[CL_PATH=CL File Path],[CL_NAME=CL File Name]
```

SHEAR7 PATH defaults to:

"C:\Program Files\Shear7\4.11b\Bin\shear7_4.11b.exe".

Version Number is specified in the form of X.Y where X and Y are integer values, and defaults to 4.11 if not explicitly specified. The available options are 4.11 (default), 4.10, 4.9, 4.8, 4.7 & 4.6.

The CL file contains the lift coefficient tables. You can specify the *CL File Path* (defaults to current directory) and the *CL File Name* (defaults to "common.s7CL" for Version 4.11 onwards, and "common.CL" for earlier versions).

TABLE INPUT

Input:	Description
SHEAR7 Installation Directory:	The path to the SHEAR7 software program. The default path is "C:\Program Files\Shear7\4.11b\Bin\shear7_4.11b.exe".
SHEAR7 Version:	The version of SHEAR7. The supported versions of SHEAR7 are 4.11 (default), 4.10, 4.9, 4.8, 4.7 & 4.6.
CL File Path:	The file path defaults to the current directory.

CL File Name:	The file name defaults to “common.s7CL” for Version 4.11 onwards, and “common.CL” for earlier versions. If specifying a file name explicitly, do not include any file extension, and this will be added automatically for you.

1.8.14.21 *ZONES

PURPOSE

To define element sets that will be used as one zone in SHEAR7.

THEORY

Refer to SHEAR7 Data Specification for further information on this feature. For more information, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

KEYWORD FORMAT

A repeatable line with one entry:

```
SET=Set Name
```

SHEAR7 recommends not attempting to model too much detail in zones. The zone capability in SHEAR7 should not be used to include small structural details. On the otherhand, Flexcom will seek to capture even slight changes in geometry when it builds your SHEAR7 model. This can be problematic in areas like tapered joints with changing geometry may cause the generation of multiple zones. In order to allow the user to best capture these geometric changes `*ZONES` can be used to generate sections with equivalent properties.

In generation of these equivalent sections the maximum drag diameter is used while a weighted average of other parameters are used. It’s worth noting that when internal fluid is present the total weight of the internal fluid will be calculated and smeared over the section. Additionally, the user should be careful as to which set they assign as a zone as the set does not have to be consecutive.

TABLE INPUT

Input:	Description

Set Name:	The element set which the user wish have specified as one zone in SHEAR7.

1.8.15 $SUMMARY COLLATE

This section corresponds to the summary collation postprocessing module, which allows you to collate the summary postprocessing results across a range of different time domain analyses. Refer to Summary Postprocessing Collation for further details.

This section contains the following keywords:

- `*COLLATE` is used to specify the summary postprocessing data to be collated and any exclusion criteria. This keyword is optional, and if you do not explicitly designate certain parameters for collation purposes, Flexcom will attempt to collate all available data.
- `*IDENTIFY` is used to identify output files for inclusion in summary postprocessing collation.
- `*OUTPUT FILES` is used to specify the type of output files required from summary collation.
- `*PLOT` is used to request the creation of a Summary Collation Plot which graphically presents the variation of any summary postprocessing output against key driving parameters.

1.8.15.1 *COLLATE

PURPOSE

To specify the summary postprocessing data to be collated and any exclusion criteria. This keyword is optional, and if you do not explicitly designate certain parameters for collation purposes, Flexcom will attempt to collate all available data.

THEORY

Refer to Summary Postprocessing Collation for further information on this feature.

KEYWORD FORMAT

One optional line followed by blocks of two lines repeated as necessary. The second line in each block is optional.

```
TITLE=Output Title [, UNITS=Unit]
[MIN=Minimum Value], [MAX=Maximum Value], [STDDEV=Standard Deviation]
```

The first data line, in the block, specifies the title of the result to be included in the collation and the second data line the exclusion criteria that apply for those results. All results sharing the same title are collated. Summary outputs with no title are excluded from the collation. The `Unit` entry is described in Note (c). If a `Minimum` is specified then results minima less than this value are not collated. If a `Maximum` is specified then results maxima greater than this value are not collated. If a `Standard Deviation` is specified then results whose standard deviation is greater than this value are not collated. If this keyword is not specified, Flexcom attempts to collate all available data from successful analyses.

The extension of the file containing the `$SUMMARY COLLATE` section and the extension of the analysis files to be collated must all have the same file extension i.e. a metric load case analysis (*.keyxm) can only be collated by `$SUMMARY COLLATE` contained in a *.keyxm file. Analysis with other keyword file extensions will be ignored and a warning issued.

TABLE INPUT

Input:	Description
Output Title:	A descriptive title associated with the parameter to be collated. See Note (a).
Unit:	The units to be used for the output. This entry is optional, and defaults to the base units for this output type. See Note (b).
Minimum Value:	A threshold level for minima. This entry is optional. See Note (c).
Maximum Value:	A threshold level for maxima. This entry is optional. See Note (c).
Standard Deviation:	A threshold level for standard deviation. This entry is optional. See Note (c).

NOTES

(a) You have control over what parameters you wish to collate, based on the descriptive titles assigned to various outputs during the creation of the *Summary Output File* for each individual analysis. This entry is optional however, and if you do not specify any descriptive titles for collation purposes, Flexcom will attempt to collate all available data. Only parameters which share the same descriptive title are compared (any outputs with blank titles are ignored).

(b) The Units entry explicitly specifies what units are to be assigned to the output. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid.

(c) You may also specify exclusion criteria to be applied, in terms of threshold levels for minima, maxima or standard deviation. If you choose to specify a threshold minimum, then any analysis where the parameter under consideration is less than the threshold is omitted. Likewise, if you opt to specify a threshold maximum or standard deviation, then any individual analysis results where the relevant parameter exceeds the threshold level is omitted. An output must pass all specified filters before it is included in the collated results. If specified, the units for the minima, maxima and standard deviation are taken from the Units entry. If no Unit is specified the unit is the base unit for that collation title.

1.8.15.2 *IDENTIFY

PURPOSE

To identify output files for inclusion in summary postprocessing collation.

THEORY

Refer to Summary Postprocessing Collation for further information on this feature.

KEYWORD FORMAT

A block of lines, all of which are optional. The first line can be repeated as often as necessary.

```
[FOLDER=Folder]
[OPTION=SUBFOLDERS]
[TAG=Substring]
```

The *`Folder`* should include the full path to the relevant folder, must end in a backslash, and should be enclosed in quotation marks if it contains any spaces. If not specified, *`Folder`* defaults to the current folder where the keyword file is located. If more than one `FOLDER` entry is present, multiple collation will occur across several root folders. `OPTION=SUBFOLDERS` instructs the search engine to look in all subfolders (except hidden folders, system folders and symbolic links). *`Substring`* can be any combination of characters, but cannot include any of the following characters: \, /, :, *, ?, “, <, > and |. In order to be identified, a file must match all specified substrings. If a substring is not specified, Flexcom attempts to identify all summary output files in the current folder.

TABLE INPUT

Input:	Description
Path(s) to Database Files:	The master or "root level" directory in which Flexcom is to search for relevant files. If not specified, this defaults to the current folder where the keyword file is located.
Search Subfolders:	This option allows you to indicate that you wish to include all subdirectories of the master folder in the search process also.
Search Tag:	If you invoke this option, Flexcom searches through the files, retaining only those which contain the specified tag in their file names. In order for a file to be considered, its file name must contain all the specified tags. For example, this would allow you to easily sort "near" current load cases from "far" cases.

1.8.15.3 *OUTPUT FILES

PURPOSE

To specify the type of output files required from summary collation.

THEORY

Refer to Spreadsheet Output for further information on this feature.

KEYWORD FORMAT

A single line specifying if Excel output is required.

```
[EXCEL=Excel Option]
```

If the keyword is not present, then Excel output is enabled. *Excel Option* can be `YES` (the default) or `NO`.

TABLE INPUT

Input:	Description
Excel:	Invoking this option means that spreadsheet based output will be produced in the form of Excel files (with the file extension XLSX).

1.8.15.4 *PLOT

PURPOSE

To request the creation of a 2D Plot or a 3D Plot which graphically presents the variation of a set of summary postprocessing results as a function of user specified values. For example, you can plot the maximum effective tension as a function of both wave period and incident wave heading in a 3-dimensional space.

THEORY

Refer to Summary Collation Plots for further information on this feature.

KEYWORD FORMAT

A single line defining the following parameters.

```
TITLE=Output Title, PRIMARY AXIS=Primary Axis, [SECONDARY AXIS=Secondary Axis
```

This line can be repeated as often as necessary to request additional plots.

The significance of each parameter is as follows.

- *Output Title* specifies the title of the Summary Postprocessing result to be plotted on the vertical axis.

- `Primary Axis` defines the primary horizontal axis along which the set of Summary Postprocessing results will be plotted.

- `Secondary Axis` defines the secondary horizontal axis along which the set of Summary Postprocessing results will be plotted in 3D. This entry is optional. A 2D Plot will be created if this entry is omitted.

- `Statistic` refers to the required statistical measurement of the Summary Postprocessing output. The options are `MAX` (default), `MIN`, `MEAN`, `RANGE` or `DEVIATION`.

- `Plot Title` refers to the plot title. This entry is optional and defaults to `Output Title` if omitted.

TABLE INPUT

Input:	Description
Output Title:	The name of the Summary Postprocessing output title to be plotted on the vertical axis. Any variable name referenced here must have been explicitly requested for inclusion in summary postprocessing, via the `TITLE` entry under any of the `$SUMMARY POSTPROCESSING` keywords. See Note (a).
Primary Axis:	The name of the primary horizontal axis along which the collated data is to be plotted. Any axis referenced here must have been explicitly defined via the `*COLLATE PLOT AXES` keyword under the $SUMMARY POSTPROCESSING section.
Secondary Axis:	The name of the secondary horizontal axis along which the collated data is to be plotted in 3D. Any axis referenced here must have been explicitly defined via the `*COLLATE PLOT AXES` keyword under the $SUMMARY POSTPROCESSING section. This entry is optional, and a 2D Plot will be created if this entry is omitted.

Statistic:	The required statistical measurement of the Summary Postprocessing output. The options are *Maximum* (default), *Minimum*, *Mean*, *Range* or *Deviation*.
Plot Title:	A descriptive title to be displayed on the plot. This entry is optional and defaults to *Output Title* if omitted.

NOTES

(a) In cases where the `*STANDARD OUTPUT` keyword has been used to request summary postprocessing output, then the name of the output variables are automatically populated by Flexcom. If you are unsure about their exact titles, they may be examined in the Summary Output File. However, they are always of the form...

- `Effective Tension (Set Name)`
- `Resultant Moment  (Set Name)`
- `Von Mises Stress  (Set Name)`

1.8.16 $SUMMARY POSTPROCESSING

This section corresponds to the summary postprocessing facility, which allows you to extract pertinent results (e.g. maximum/minimum values) from time domain analyses in succinct tabular format. Refer to Summary Postprocessing for further details.

This section contains the following keywords:

- `*AXIS/VECTOR` is used to define axis systems and vectors for use in postprocessing.
- `*COLLATE PLOT AXES` is used to define a number of key parameters which uniquely identify a particular simulation within a load case matrix. This is a prerequisite for generating 3-dimensional Summary Collation Plots.
- `*ELEMENT SETS` is used to group individual elements into element sets.
- `*OPTIONS` is used to select a number of options relating to summary postprocessing.

- *PARA ANGLE AXIS is used to request summary output of the angle between an element and either a vector or an axis system.
- *PARA ANGLE ELEMENT is used to request summary output of the angle between two elements.
- *PARA ANGLE TENSION is used to request summary output of angle/tension/pseudo-curvature.
- *PARA FORCE is used to request summary output of element restoring forces.
- *PARA FORCE ENVELOPE is used to request summary output of statistics of element restoring forces.
- *PARA KINEMATIC is used to request summary output of nodal motions.
- *PARA REACTION is used to request summary output of nodal reactions.
- *PARA SEABED is used to request summary output of statistics of parameters related to seabed contact.
- *RESTART is used to indicate that a postprocessing run is to be restarted from an analysis file of different stub name.
- *STANDARD OUTPUT is used to quickly request a selection of commonly used outputs.
- *TIME is used to specify the time interval over which the summary output statistics are to be calculated.

1.8.16.1 *AXIS/VECTOR

PURPOSE

To define axis systems and vectors for use in postprocessing.

THEORY

Refer to Angles Output for further information on this feature.

KEYWORD FORMAT

The keyword starts with a line defining the name of the axis system or vector. This is then followed by a line with either seven numbers or four numbers, depending on whether an axis system or vector is being defined. Any further axis systems or vectors can be defined by repeating the two lines as often as necessary.

Lines defining an axis system:

```
AXIS=Axis Name
Origin Node (Number or Label), X1, Y1, Z1, X2, Y2, Z2
```

Lines defining a vector:

```
VECTOR=Vector Name
Origin Node (Number or Label), X, Y, Z
```

If you specify a node label rather than a node number, it must be enclosed in {} brackets. The two vectors defining an axis system must have non-zero length and must be orthogonal. Any vectors defined must have non-zero length.

AXIS SYSTEMS

Table Input

Input:	Description
Name:	A unique name for the axis system being defined.
Origin Node:	The node (number or label) at which the axis system is being positioned. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
X1:	The component in the global X-direction of the local x-axis of the axis system being defined.
Y1:	The component in the global Y-direction of the local x-axis of the axis system being defined.
Z1:	The component in the global Z-direction of the local x-axis of the axis system being defined.
X2:	The component in the global X-direction of the local y-axis of the axis system being defined.

Y2:	The component in the global Y-direction of the local y-axis of the axis system being defined.
Z2:	The component in the global Z-direction of the local y-axis of the axis system being defined.

Notes

(a) If you do not specifically input a value in any of Columns 3-8, then the corresponding variable defaults to 0. Note however that all three of *X1, Y1* and *Z1* or all three of *X2, Y2* and *Z2* cannot equal 0 – the null vector cannot be either the local x-axis or the local y-axis.

(b) Note that the local x-axis and local y-axis that you define here must be *orthogonal*, that is, the true angle between the two vectors defining the local x- and y-axes must be 90°. Flexcom will generate an error message if this is not the case, as vectors that are not orthogonal cannot be used to form a valid axis system.

VECTORS

Table Input

Input:	Description
Name :	A unique name for the vector being defined.
Origin Node:	The node (number or label) at which the vector is being positioned. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
X:	The component in the global X-direction of the vector being defined.
Y:	The component in the global Y-direction of the vector being defined.
Z:	The component in the global Z-direction of the vector being defined.

Notes

(a) If you do not specifically input a value in any of Columns 3, 4 or 5, then the corresponding variable defaults to 0. Note however that all three of X, Y and Z cannot equal 0 – the angle between an element and the null vector is not a valid output.

1.8.16.2 *COLLATE PLOT AXES

PURPOSE

To define plot axes and their respective values which uniquely identify this Summary Postprocessing result within a matrix of Summary Postprocessing results for the purposes of generating plots of collated data. Each axis/value pair is defined in terms of a descriptive name and an associated value. If the same axis name is used in other Summary Postprocessing results, and assigned different values, it will be possible to plot the variation of any requested Summary Postprocessing output along the defined axis during the Summary Collation phase. For example, you can plot maximum effective tension as a function of both wave period and incident wave heading in a 3-dimensional space. Any plot axis defined using this keyword may subsequently be referenced by the *PLOT keyword (under the $SUMMARY COLLATE section).

THEORY

Refer to Summary Collation Plots for further information on this feature.

KEYWORD FORMAT

A single line defining the identifier name and its numeric value. This line can be repeated as often as necessary:

```
Plot Axis Name, Value
```

TABLE INPUT

Input:	Description
Plot Axis Name:	The name of the plot axis whose values uniquely identifies these summary postprocessing results.

Value:	The value associated with this plot axis. 'Value' will be interpreted as a real number so the input should not contain any character expressions.

NOTES

(a) When defining a plot axis it is important to define meaningful names. For example, "Wave Period" or "Wave Heading" clearly indicate meaning. Any named defined here may be referenced subsequently when creating Summary Collation Plots. The Summary Postprocessing result of interest may then be plotted at the horizontal location as defined by the user specified plot axis value.

(b) If a plot axis name contains spaces, then quotation marks must be used to denote the bounds of the name e.g. "Wave Period", 'Wave Heading'.

1.8.16.3 *ELEMENT SETS

PURPOSE

To group individual elements into element sets.

THEORY

Refer to Summary Postprocessing for further information on this feature.

KEYWORD FORMAT

Blocks of lines that define a set repeated as often as necessary. Each block begins with a line defining the set name. This is followed by various types of lines that define the elements in the set. These lines can be mixed and repeated as often as necessary until every element in the set is defined.

Line to define set name:

```
SET=Set Name
```

Line containing a list of elements. This line can contain up to 20 elements (numbers or labels). Any further elements must be defined on a new line.

```
List of Elements (Numbers or Labels)
```

Line defining a sequence of elements:

GEN=*Start Element (Number or Label)*, *End Element (Number or Label)* [, *Element*

Line referencing another set of elements:

SUBSET=*Subset Set Name*

The set ALL is predefined and cannot be redefined. Every element assigned to a set must be defined using *ELEMENT. Set names are up to 256 characters long, can include spaces and are case insensitive. If you specify an element label rather than an element number, it must be enclosed in {} brackets. *Element Increment* defaults to one.

TABLE INPUT

Input:	Description
Set Name:	A unique label for the element set. Set names are not case sensitive, so the set name 'RISER' is equivalent to 'Riser', which is in turn equivalent to 'riser'.
Elements:	The elements comprising the set. These can be input in three ways, namely: (i) A list of elements (numbers or labels), such as for example "1, 5, 7". (ii) A group of consecutive elements (numbers or labels), input using the format: "11 – 15". This definition specifies Elements 11 to 15 inclusive. The specification "11 - 15 – 2" can be used to specify the Elements 11, 13 and 15 - that is, from Element 11 to Element 15 in steps of 2. (iii) Another set name. For example you might define three sets named SET_1, SET_2 and SET_3, and then combine them in a further set, say ALLSETS, by inputting "SET_1, SET_2, SET_3". If you specify an element label rather than an element number, it must be enclosed in {} brackets.

	All three specifications can be combined, as for example in "1, 7, 9, 12-15, 17, 20-50-10, RISER". This set combines Elements 1, 7, and 9; elements 12 to 15 inclusive; Element 17; Elements 20, 30, 40 and 50; and the elements comprising the set RISER.
SubSets:	An additional element set or sets whose elements are to be added to the current set definition. If more than one set is referenced, use commas to separate out the set names.

NOTES

(a) Use as many lines as you need to completely define the elements comprising a particular set. Simply leave the first column blank for second and subsequent lines.

(b) If a set name is included in the specification of another set, then obviously the elements comprising that set must be separately defined.

(c) There is one predefined element set in Flexcom, which is named *All*. Not surprisingly this comprises all of the elements of the finite element discretisation and is the default element set. Note that Flexcom will resist any attempt to redefine the make-up of the set *All*.

(d) The names and composition of element sets you define in a preceding section carry through to all dependent (restart) sections. For example, if you define an element set in the `$MODEL` section, it will automatically be available in a subsequent `$DATABASE POSTPROCESSING` section. So there is no need to repeat the specifications again – you can just use the set names directly. If you redefine the composition of a set previously defined in a preceding section, Flexcom will output a warning, but will continue with the most recent set definition.

1.8.16.4 *OPTIONS

PURPOSE

To select a number of options relating to summary postprocessing.

THEORY

Refer to Summary Postprocessing for further information on this feature.

KEYWORD FORMAT

Three lines specifying options in the following format:

```
WARNINGS=[Warnings Option]
DELETEDATA=[Delete Data Option]
PSEUDOCURVATURE=[Pseudo Curvature Option]
```

`Warnings Option` can be YES (the default) or NO. `Delete Data Option` can be YES or NO (the default). `Pseudo Curvature Option` can be EFF (denoting effective tension, the default) or AXI (denoting axial force).

TABLE INPUT

Input:	Description
Pseudo Curvature:	This option allows you to specify whether Effective Tension (the default) or Axial Force is to be used in the calculation of pseudo-curvature.
Warnings:	This option allows you to suppress warning messages (they are activated by default).
Delete Database:	This option allows you automatically delete results databases (they are retained by default).

NOTES

(a) The *Pseudo Curvature* option is immaterial if you do not specify pseudo-curvature as one of your summary output parameters.

(b) When *Warnings* are activated, Flexcom issues a warning in the summary output file if the maximum or minimum of a particular parameter is more than two standard deviations from the mean value. This facility is intended to alert users to the possibility that a maximum or minimum value is perhaps a numerical “spike” rather than a genuine extreme. To confirm whether or not this is the case, you would look at the time history of the response.

(c) The *Delete Database* option might be useful when a large number of analyses are being performed. The results database could automatically be used to generate the summary output file for each analysis and each database could then be deleted, freeing disk space for subsequent analyses.

1.8.16.5 *PARA ANGLE AXIS

PURPOSE

To request summary output of the angle between an element and either a vector or an axis system.

THEORY

Refer to Angles Output for further information on this feature.

KEYWORD FORMAT

A block of lines as follows:

```
[TITLE=Title Name]
[SCALE=Scale Factor] [, UNITS=Unit]
[PLOT=Plot Option]
DATA1=Element 1 (Number or Label)
DATA2=Axis/Vector Name or GLOBAL
DATA3=Angle Option
```

`Axis/Vector Name` must be defined using `*AXIS/VECTOR`. Title Name defaults to blank. `Scale Factor` defaults to 1. The `Unit` entry is described in Note (b). `Plot Option` can be YES or NO (the default). `Angle Option` can be 1 (for actual angle), 2 (for Projected-xy angle), 3 (for Projected-yz angle) or 4 (for Projected-xz angle). If you specify an element label rather than an element number, it must be enclosed in {} brackets.

TABLE INPUT

Input:	Description
Element:	The number/label of the element. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Axis/Vector:	The axis or vector to be used with the element you specify in computing the required angle.

Angle Type:	The type of angle output required. This input is meaningful only when you are requesting statistics of the angle between an element and an axis system.
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the output. This entry is optional, and defaults to the base units for this output type. See Note (b).
Plot:	This option allows you to request a timetrace plot, in addition to the summary output. By default, no plot is created.

NOTES

(a) The units entry explicitly specifies what units are to be assigned to the output. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, angle can be specified in Radian, degree, etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Radian, rad, etc. are all equally valid.

1.8.16.6 *PARA ANGLE ELEMENT

PURPOSE

To request summary output of the angle between two elements.

THEORY

Refer to Angles Output for further information on this feature.

KEYWORD FORMAT

A block of lines as follows:

```
[TITLE=Title Name]
[SCALE=Scale Factor] [, UNITS=Unit]
[PLOT=Plot Option]
```

```
DATA1=Element 1 (Number or Label)
DATA2=Element 2 (Number or Label)
```

`Title Name` defaults to blank. `Scale Factor` defaults to 1. The `Unit` entry is described in Note (b). `Plot Option` can be `YES` or `NO` (the default). If you specify an element label rather than an element number, it must be enclosed in {} brackets.

TABLE INPUT

Input:	Description
Element 1:	The first element (number or label). If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Element 2:	The second element (number or label). If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the output. This entry is optional, and defaults to the base units for this output type. See Note (b).
Plot:	This option allows you to request a timetrace plot, in addition to the summary output. By default, no plot is created.

NOTES

(a) The units entry explicitly specifies what units are to be assigned to the output. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid.

1.8.16.7 *PARA ANGLE TENSION

PURPOSE

To request summary output of angle/tension/pseudo-curvature.

THEORY

Refer to Angles Output for further information on this feature.

KEYWORD FORMAT

A block of lines as follows:

```
[TITLE=Title Name]
[SCALE=Scale Factor] [, UNITS=Unit]
[PLOT=Plot Option]
DATA1=Element (Number or Label)
DATA2=Local Node No.
DATA3=Element (Number or Label) or Axis/Vector Name or GLOBAL
```

Refer to Force Variable Input for further information on acceptable *Variable* values. Refer to Location Parameter Input for further information on the significance of the *Location* input. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis.

Axis/Vector Name must be defined using `*AXIS/VECTOR`. *Title Name* defaults to blank. *Scale Factor* defaults to 1. The *Unit* entry is described in Note (b). *Plot Option* can be `YES` or `NO` (the default). If you specify an element label rather than an element number, it must be enclosed in {} brackets.

TABLE INPUT

Input:	Description
Element:	The element (number or label) for which the summary output is required. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Local Node:	This option allows you to choose between three locations on the specified element. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis.

Angle:	The entry can be either i) an element (number or label), ii) the name of a vector, or iii) the name of an axis system. See Note (a). If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the output. This entry is optional, and defaults to the base units for this output type. See Note (b).
Plot:	This option allows you to request a timetrace plot, in addition to the summary output. By default, no plot is created.

NOTES

(a) The units entry explicitly specifies what units are to be assigned to the output. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid.

1.8.16.8 *PARA FORCE

PURPOSE

To request summary output of element restoring forces.

THEORY

Refer to Output Parameters for further information on this feature.

KEYWORD FORMAT

A block of lines as follows:

```
[TITLE=Title Name]
```

```
[SCALE=Scale Factor] [, UNITS=Unit]
[PLOT=Plot Option]
DATA1=Element (Number or Label)
DATA2=Local Node No.
DATA3=Variable [, Location]
```

Refer to Force Variable Input for further information on acceptable *Variable* values. Refer to Location Parameter Input for further information on the significance of the *Location* input. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis.

`Title Name` defaults to blank. `Scale Factor` defaults to 1. The `Unit` entry is described in Note (c). `Plot Option` can be YES or NO (the default). If you specify an element label rather than an element number, it must be enclosed in {} brackets.

TABLE INPUT

Input:	Description
Element:	The element (number or label) for which the summary output is required. If you specify an element label rather than an element number, it must be enclosed in {} brackets.
Local Node:	This option allows you to choose between three locations on the specified element. The significance of the Local Node Input used during postprocessing depends on whether element based outputs are stored on a node or integration point basis.
Variable:	This list allows you to select a relevant output parameter. The entries are largely self-explanatory. See Note (a).
Location:	This parameter is appropriate when you request bending stress, bending strain, von Mises stress, or pressure. See Note (b).
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.

Unit:	The units to be used for the output. This entry is optional, and defaults to the base units for this output type. See Note (c).
Plot:	This option allows you to request a timetrace plot, in addition to the summary output. By default, no plot is created.

NOTES

(a) Refer to Force Variable Input for further information on acceptable *Variable* values.

(b) Refer to Location Parameter Input for further information on the significance of the *Location* input.

(c) The units entry explicitly specifies what units are to be assigned to the output. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid.

1.8.16.9 *PARA FORCE ENVELOPE

PURPOSE

To request summary output of statistics of element restoring forces.

THEORY

Refer to Output Parameters for further information on this feature.

KEYWORD FORMAT

A block of lines as follows:

```
[TITLE=Title Name]
[SCALE=Scale Factor] [, UNITS=Unit]
[PLOT=Plot Option]
DATA1=Set Name or ALL
DATA2=Variable [, Location]
```

Refer to Force Variable Input for further information on acceptable *Variable* values. Refer to Location Parameter Input for further information on the significance of the *Location* input.

`Title Name` defaults to blank. `Scale Factor` defaults to 1. The `Unit` entry is described in Note (c). `Plot Option` can be YES or NO (the default). `Set Name` must be defined using `*ELEMENT SETS`.

TABLE INPUT

Input:	Description
Element Set:	The element set for which the summary output is required.
Variable:	This list allows you to select a relevant output parameter. The entries are largely self-explanatory. See Note (a).
Location:	This parameter is appropriate when you request bending stress, bending strain, von Mises stress, or pressure. See Note (b).
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the output. This entry is optional, and defaults to the base units for this output type. See Note (c).
Plot:	This option allows you to request a timetrace plot, in addition to the summary output. By default, no plot is created.

NOTES

(a) Refer to Force Variable Input for further information on acceptable *Variable* values.

(b) Refer to Location Parameter Input for further information on the significance of the *Location* input.

(c) The units entry explicitly specifies what units are to be assigned to the output. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid.

1.8.16.10 *PARA KINEMATIC

PURPOSE

To request summary output of nodal motions.

THEORY

Refer to Output Parameters for further information on this feature.

KEYWORD FORMAT

A block of lines as follows:

```
[TITLE=Title Name]
[SCALE=Scale Factor] [, UNITS=Unit]
[PLOT=Plot Option]
DATA1=Node (Number or Label)
DATA2=DOF
DATA3=Type
```

`Title Name` defaults to blank. `Scale Factor` defaults to 1. The `Unit` entry is described in Note (a). `Plot Option` can be YES or NO (the default). If you specify a node label rather than a node number, it must be enclosed in {} brackets. `Type` can be 1 (motion), 2 (displacement), 3 (velocity) or 4 (acceleration).

TABLE INPUT

Input:	Description
Node:	The node (number or label) for which the summary output is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.

DOF:	The global degree of freedom (DOF) at this node. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, or 3 for the global Z-direction; 4, 5 or 6 for the components of the rotation vector in the global X, Y and Z axes respectively; or 7 for the magnitude of rotation.
Type:	The (kinematic) type can be *Motion, Displacement, Velocity, or Acceleration*.
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the output. This entry is optional, and defaults to the base units for this output type. See Note (a).
Plot:	This option allows you to request a timetrace plot, in addition to the summary output. By default, no plot is created.

NOTES

(a) The units entry explicitly specifies what units are to be assigned to the output. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, length can be specified in metres, inches, etc. The program is quite flexible in terms of accepting unit descriptions – e.g. metres, m, etc. are all equally valid.

1.8.16.11 *PARA REACTION

PURPOSE

To request summary output of nodal reactions.

THEORY

Refer to Output Parameters for further information on this feature.

KEYWORD FORMAT

A block of lines as follows:

```
[TITLE=Title Name]
[SCALE=Scale Factor] [, UNITS=Unit]
[PLOT=Plot Option]
DATA1=Node (Number or Label)
DATA2=DOF
DATA3=Axis Name or GLOBAL
```

Title Name defaults to blank. *Scale Factor* defaults to 1. The *Unit* entry is described in Note (a). *Plot Option* can be YES or NO (the default). If you specify a node label rather than a node number, it must be enclosed in {} brackets. *Axis/Vector Name* must be defined using *AXIS/VECTOR.

TABLE INPUT

Input:	Description
Node:	The node (number or label) for which the summary output is required. If you specify a node label rather than a node number, it must be enclosed in {} brackets.
DOF:	The global degree of freedom (DOF) at this node. Specify a value of 1 for the global X-direction, 2 for the global Y-direction, or 3 for the global Z-direction; 4, 5 or 6 for moments about the global X, Y and Z axes respectively.
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Axis:	The name of the axis system for which the angle timetrace is required. This entry defaults to the global XYZ axes.
Unit:	The units to be used for the output. This entry is optional, and defaults to the base units for this output type. See Note (a).

Plot:	This option allows you to request a timetrace plot, in addition to the summary output. By default, no plot is created.

NOTES

(a) The units entry explicitly specifies what units are to be assigned to the output. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, force can be specified in Newtons, Kips etc. The program is quite flexible in terms of accepting unit descriptions – e.g. Newtons, Newton, N etc. are all equally valid.

1.8.16.12 *PARA SEABED

PURPOSE

To request summary output of statistics of parameters related to seabed contact.

THEORY

Refer to Output Parameters for further information on this feature.

KEYWORD FORMAT

A block of lines as follows:

```
[TITLE=Title Name]
[SCALE=Scale Factor] [, UNITS=Unit]
[PLOT=Plot Option]
DATA1=Set Name or ALL
DATA2=Seabed Parameter
```

`Title Name` defaults to blank. `Scale Factor` defaults to 1. The `Unit` entry is described in Note (d). `Plot Option` can be `YES` or `NO` (the default). `Set Name` must be defined using `*ELEMENT SETS`. `Seabed Parameter` can be 1 (for seabed clearance), 2 (for length on seabed) or 3 (for maximum span, average gap, and average gap over central third of maximum span).

TABLE INPUT

Input:	Description

Element Set:	The element set for which the summary output is required.
Parameter:	The options are *Clearance*, *Length on Seabed*, and *Maximum Span and Average Gap*. The options are largely self-explanatory. See Notes (a) - (c).
Title:	A descriptive title to be associated with the output. This entry is optional.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Unit:	The units to be used for the output. This entry is optional, and defaults to the base units for this output type. See Note (d).
Plot:	This option allows you to request a timetrace plot, in addition to the summary output. By default, no plot is created.

NOTES

(a) *Clearance* presents the statistics of clearance between the range of elements and the seabed. What actually happens is that for the element set you specify the program first attempts to identify “sagging” and “hogging” sections within the set. The minimum clearance calculations are then based on the sagging sections and the maximum clearance calculations on hogging sections. Where no such sections can be identified, the calculations cannot be performed, and a zero value is returned by default. Consider for example the case of a free-hanging catenary partly lying on the seabed, where seabed clearance is requested for a set of elements well above the touchdown zone. The summary output will report zero clearance in this case, because no sagging and hogging regions are present. Although this might appear counter-intuitive the program is operating as intended.

(b) *Length on Seabed* calculates the statistics of the length of riser in the element range lying on the seabed during the analysis.

(c) *Maximum Span* is defined as the maximum continuous length of pipeline that is not supported by the seabed. *Average Gap* is defined as the average clearance between the pipeline and the seabed within the region of the maximum span. The *Average Gap* is also presented over the central third of the maximum span.

(d) The units entry explicitly specifies what units are to be assigned to the output. This entry is optional, and if omitted, Flexcom simply assigns suitable units depending on the base units for the particular output type. If you do explicitly specify a unit, then it must be consistent with the dimension of the output type. For example, length can be specified in metres, inches, etc. The program is quite flexible in terms of accepting unit descriptions – e.g. metres, m, etc. are all equally valid.

1.8.16.13 *RESTART

PURPOSE

To indicate that a postprocessing run is to be restarted from an analysis file of different stub name.

THEORY

Refer to Restart Analyses for further information on this feature.

KEYWORD FORMAT

A single line defining the file name.

```
LAST=File Name
```

The keyword is optional, and is not required if the `$SUMMARY POSTPROCESSING` section is contained within the same file as the actual analysis data.

TABLE INPUT

Input:	Description
Restart File:	The name of the analysis from which the postprocessing run is to be restarted. This may be input in terms of the analysis name (e.g. *Example1*) or the full path of the analysis (e.g. *C:\Flexcom\Examples\Example1-Static*).

1.8.16.14 *STANDARD OUTPUT

PURPOSE

To quickly request a selection of commonly used outputs, namely effective tension, resultant bending moment and von Mises stress.

THEORY

Refer to Standard Output for further information on this feature.

KEYWORD FORMAT

A single line defining the element set for which the standard output is required, repeated as often as necessary. If no element set is listed, it is assumed that output is required for all elements (i.e. SET=ALL).

```
[SET=Set Name]
```

TABLE INPUT

Input:	Description
Element Set:	The element set for which the standard output is required. If no element sets are listed, it is assumed that output is required for all elements (i.e. element set *All*).

1.8.16.15 *TIME

PURPOSE

To specify the time interval over which the summary output statistics are to be calculated.

THEORY

Refer to Summary Postprocessing for further information on this feature.

KEYWORD FORMAT

Two lines specifying the start and end times:

```
[START=Start Time]
[END=End Time]
```

Start Time defaults to the analysis start time. *End Time* defaults to the analysis end time.

TABLE INPUT

Input:	Description
Start Time:	The start time for the calculation of statistics. This entry is optional, and defaults to the analysis start time if omitted.
End Time:	The end time for the calculation of statistics. This entry is optional, and defaults to the analysis end time if omitted.

NOTES

(a) It would be usual to specify a start time explicitly, to exclude initial transients from statistics calculations.

1.8.17 $SUMMARY WAVE SCATTER

This section corresponds to the Summary Wave Scatter feature, which allows you to generate Summary Database Files for seastate combinations which you have not actually simulated. Based on some selected 'reference seastates' in your scatter diagram, Flexcom estimates simulation results for adjacent seastates based on an extrapolation technique. You may then use the Summary Postprocessing Collation to collate information from all seastates.

The simulation of an entire scatter diagram in the time domain can be quite time consuming, so this a highly efficient solution technique which can save you considerable computational time. However given the inherent approximation involved, caution is strongly advised regarding the user application of this feature. Refer to Summary Wave Scatter for further details.

This section contains the following keywords:

- `*SCATTER DIAGRAM` is used to specify the wave scatter diagram, including the grouping of similar seastates into blocks and the nomination of appropriate reference seastates.

- `*OPTIONS` is used to specify some miscellaneous options regarding the operation of the Summary Wave Scatter feature.
- `*WAVE SPECTRUM` is used to specify the wave spectrum type for use in the Summary Wave Scatter computations.

1.8.17.1 *SCATTER DIAGRAM

PURPOSE

To specify a wave scatter diagram, including the grouping of similar seastates into blocks, and the nomination of appropriate reference seastates

THEORY

Refer to Summary Wave Scatter for further information on this feature.

KEYWORD FORMAT

The wave scatter diagram is specified in sub-sections called blocks. Each block is defined by a single reference seastate and several other seastates within that same block.

For reference seastates, the data values are preceded by the REFERENCE tag, and the line also specifies the file name which corresponds to the time domain simulation for the reference seastate.

```
REFERENCE=Hs, Tz/Tp/Te, Number of Occurrences, FILE=File Name
```

Non-reference seastates within the same block as the reference seastate are then listed underneath, with the line repeated for as many seastates as you want to include within that seastate block.

```
Hs, Tz/Tp/Te, Number of Occurrences
```

The next block may then be defined, starting with the reference seastate followed by the other seastates, and so on until the entire wave scatter diagram has been defined.

TABLE INPUT

REFERENCE SEASTATE ANALYSES

Input:	Description
Hs:	The reference seastate significant wave height Hs.
Tz/Tp/Te:	The reference seastate mean zero up-crossing period Tz, or peak period Tp, or energy period Te.
Flexcom Analysis	The file name which corresponds to the time domain simulation for the reference seastate.

SEASTATE SCATTER DIAGRAM

Input:	Description
Hs:	The seastate significant wave height Hs.
Tz/Tp/Te:	The seastate mean zero up-crossing period Tz, or peak period Tp, or energy period Te.
No. of Occurrences:	The number of occurrences in a given period (typically one year) of the particular combination of H_s and $T_z/T_p/T_e$ values which that cell represents.

Notes

(a) The scatter diagram is defined in terms of significant wave height (H_s) and mean zero up-crossing period (T_z), or peak period (T_p), or energy period (T_e). If there is no occurrence of a particular combination of H_s and $T_z/T_p/T_e$, you simply leave the corresponding cell blank.

(b) The number of occurrences is not currently used by Flexcom, but wave scatter diagrams are typically defined in this form anyway, and it should help to guide you in the creation of blocks and selection of reference seastates.

1.8.17.2 *OPTIONS

PURPOSE

To specify some miscellaneous options regarding the operation of the Summary Wave Scatter feature

THEORY

Refer to Summary Wave Scatter for further information on this feature.

KEYWORD FORMAT

A number of data lines specifying options in the following format:

```
OVERWRITE DATA=[Overwrite Data Option]
POWER RATING=[Maximum Power Rating]
```

Overwrite Data Option can be YES or NO (the default). If specified, *Maximum Power Rating* must be a number.

TABLE INPUT

Input:	Description
Overwrite Data:	This option allows you to automatically overwrite results databases which may have already been created by Summary Postprocessing in the normal fashion (they are not overwritten by default).
Power Rating:	The upper limit to cap any average power values before they are written to the Summary Database File. This entry is optional and if unspecified no capping is applied.

NOTES

(a) When Summary Postprocessing is performed after a simulation has completed, Flexcom produces a text-based Summary Output File for visual inspection, but more importantly it also creates a Summary Database File, which is effectively a binary version of the same data, for subsequent access by the Summary Collation facility. The Summary Wave Scatter feature also allows you to generate Summary Database Files, but for seastate combinations for which you have not actually performed a numerical simulation. It is possible (although somewhat unlikely) that you would have originally performed numerical simulations and summary postprocessing for some seastates which are not marked as reference seastates in the wave scatter diagram. In such circumstances there is a potential conflict between data files which have been created in standard Flexcom fashion following analysis and postprocessing, and data files generated by the Summary Wave Scatter feature (which estimates simulation results based on an extrapolation technique). By default all 'genuine' data is retained, but you do have the option of overwriting existing summary database files with extrapolated ones.

1.8.17.3 *WAVE SPECTRUM

PURPOSE

To specify the wave spectrum type for use in the Summary Wave Scatter computations

THEORY

Refer to Summary Wave Scatter for further information on this feature.

KEYWORD FORMAT

One line specifying the spectrum type.

```
TYPE=Spectrum Type [, OPTION=Option]
```

Spectrum Type can be PIERSON-MOSKOWITZ or JONSWAP. *Option* can be `TZ` (default) or `TP` or `TE`.

TABLE INPUT

Input:	Description

Spectrum Type:	The options are Pierson-Moskowitz and Jonswap.
Period Type:	The options are T_z (mean zero up-crossing period, the default), T_p (peak period) and T_e (energy period).

NOTES

(a) The spectrum type is a necessary input to allow the Summary Wave Scatter feature to generate response spectra for for seastate combinations for which you have not actually performed a numerical simulation.

(b) Flexcom has traditionally modelled wave spectra in terms of either T_z or T_p. If you specify a scatter diagram in terms of T_e, Flexcom converts T_e into T_z. Refer to Wave Energy Period for further information on the conversion process.

(c) When you specify Jonswap spectra, Flexcom uses the algorithm described in Jonswap Wave to calculate the Jonswap parameters f_p, α and γ from the H_s and T_z/T_p values.

1.8.18 $TIMETRACE POSTPROCESSING

This section corresponds to the timetrace postprocessing facility, which is mainly used in the area of time domain fatigue analysis. Refer to Timetrace Postprocessing for further details.

This section contains the following keywords:

- `*PARAMETERS` is used to define miscellaneous parameters for use in timetrace postprocessing.
- `*PLOT` is used to request the creation of plots using the timetrace postprocessing.
- `*RESTART` is used to indicate that a postprocessing run is to be restarted from an analysis file of different stub name.

1.8.18.1 *PARAMETERS

PURPOSE

To define miscellaneous parameters for use in timetrace postprocessing.

THEORY

Refer to Timetrace Postprocessing for further information on this feature.

KEYWORD FORMAT

A block of two or three lines, the first of which is mandatory. The first line specifies data relating to the calculation of statistics and spectra by the *Timetrace Postprocessor*.

```
GENERAL=[Start Time] [, Time Step] [, No. of Ensembles]
```

`Start Time` defaults to the analysis start time. A `Time Step` value is required in a random sea dynamic analysis with a variable time-step, while it is ignored in the case of an analysis with a fixed step. A `No. of Ensembles` is required in a random sea dynamic analysis, while it is ignored in the case of a regular wave analysis.

The second line, which is optional, specifies the procedure to be used in the computation of extreme values by the *Timetrace Postprocessor*.

```
[EXTREME=Calculation Procedure]
```

The third line is required if the line with EXTREME= is present. The format of the third line depends on the *Calculation Procedure* specified – the options are RAYLEIGH (the default) and WEIBULL. For RAYLEIGH extrema postprocessing, the third line takes the following format:

```
DATA=[Storm Duration] [, Probability]
```

For WEIBULL extrema postprocessing, the third line takes the following format:

```
DATA=[Storm Duration] [, Probability] [, Threshold]
```

Storm Duration defaults to 3 hours and Probability to 0.01 (1%). Threshold defaults to 1.

STATISTICS

Purpose

To specify the start time for the calculation of response statistics.

Table Input

Input:	Description
Start Time:	The start time for the calculation of statistics. Any values before this time are excluded. Use this entry to exclude initial transients from statistics calculations.

Notes

(a) The use of this table is optional. By default, the start time for the calculation of statistics is the first timetrace output time.

(b) The entry you specify here is also used as the start time for the calculation of response spectra in a random sea analysis.

SPECTRA

Purpose

To define parameters relating to the calculation of response spectra.

Table Input

Input:	Description
No. of Ensembles:	The number of ensembles to be used in calculating spectra. See Note (a).
Time Step:	The time step to be used when calculating spectra from the results of a random sea dynamic analysis with a variable time step. See Note (b). If the analysis used a fixed time step then this entry is not required, and any value you specify is ignored.

Notes

(a) The procedure used by Flexcom in calculating spectra is as follows. Firstly, the output timetrace is divided equally into a number of smaller timetraces or ensembles. A spectrum for each ensemble is then calculated using the Fast Fourier Transform (FFT) algorithm. Finally, the actual spectrum to be output is found as an average of the spectra calculated for each ensemble. This standard procedure minimises the statistical error associated with the FFT process. You specify the number of ensembles to be used in this process using the No. of Ensembles entry above. This value should always be greater than 1.

(b) The FFT algorithm requires a record with a fixed time step. When you perform a variable step analysis, obviously such a record is not available, and so Flexcom must synthesise one by interpolating from the variable step record. The Time Step entry tells Flexcom what time step to use in the synthesised record.

EXTREME VALUES - RAYLEIGH

Purpose

To specify parameter values to be used in calculating extreme values with the Rayleigh distribution.

Table Input

Input:	Description
Storm Duration:	The storm duration in hours to be used when calculating extreme values. The default is 3 hours.
Probability:	The exceedance probability to be used when calculating extreme values. The default is 0.01 (= 1%).

Notes

(a) Refer to Extreme Values for a detailed discussion of all aspects of specifying data relating to extreme value postprocessing.

(b) Note that the Extreme Values – Rayleigh and Extreme Values – Weibull tables are mutually exclusive. Specification of data in either table indicates the extreme value postprocessing approach which you wish to use.

EXTREME VALUES - WEIBULL

Purpose

To specify parameter values to be used in calculating extreme values with the Weibull distribution.

Table Input

Input:	Description
Duration:	The duration in hours to be used when calculating extreme values. The default is 3 hours
Probability:	The exceedance probability to be used when calculating extreme values. The default is 0.01 (= 1%).
Threshold:	The proportion of largest maxima /smallest minima to be used when calculating extreme values. The default is 1. See Note (b).

Notes

(a) Refer to Extreme Values for a detailed discussion of all aspects of specifying data relating to extreme value postprocessing.

(b) A threshold equal to 1 denotes all the maxima/minima in the timetrace will be used to calculate the extreme values. A threshold equal to 1/n (e.g. 1/3) uses the upper 1/n of largest maxima and lowest 1/n of smallest minima to calculate the extreme value.

(c) Note that the Extreme Values – Rayleigh and Extreme Values – Weibull tables are mutually exclusive. Specification of data in either table indicates the extreme value postprocessing approach which you wish to use.

1.8.18.2 *PLOT

PURPOSE

To request the creation of plots using the timetrace postprocessing.

THEORY

Refer to Timetrace Postprocessing for further information on this feature.

KEYWORD FORMAT

A single line repeated as often as necessary to indicating the required plots.

```
CHANNEL=Channel Number [, PLOT=NONE/TIMETRACE/SPECTRUM/BOTH] [, SCALE=Scale]
```

Scale defaults to 1. *Figure Title* defaults to blank.

TABLE INPUT

Input:	Description
Channel Number:	The number of the channel. See Note (a).
Plot Type:	The options are *None* (the default), *Timetrace*, *Spectrum* and *Both*. The last two options are not relevant for regular wave analysis.
Scale:	A scale factor to apply to the data. This entry is optional, and defaults to a value of 1.
Title:	A descriptive title to be associated with the output. This entry is optional.

NOTES

(a) “Channel” in this context refers to the list of timetrace outputs requested before the analysis was performed. The timetrace storage file contains time histories of the variables of interest, or channels, which you reference here in the same order as they were originally requested. Note that a time history of wave elevation is stored automatically as the first channel, so you should bear this in mind when nominating channels of interest. If you are unsure, you can always check the tabular output file, which contains a list of channel numbers and corresponding descriptions.

(b) The timetrace postprocessor creates a tabular summary of the statistics for all of the output channels. Additionally, one or possibly two plot files may be created (depending on the *Plot Type* specification) for some or all of the output channels.

1.8.18.3 *RESTART

PURPOSE

To indicate that a postprocessing run is to be restarted from an analysis file of different stub name.

THEORY

Refer to Restart Analyses for further information on this feature.

KEYWORD FORMAT

A single line defining the file name.

```
LAST=File Name
```

The keyword is optional, and is not required if the `$TIMETRACE POSTPROCESSING` section is contained within the same file as the actual analysis data.

TABLE INPUT

Input:	Description
Restart File:	The name of the analysis from which the postprocessing run is to be restarted. This may be input in terms of the analysis name (e.g. *Example1-Static*) or the full path of the analysis (e.g. *C:\Flexcom\Example 1\Example1-Static*).

1.8.19 UNIVERSAL

This section contains univeral keywords/directives which can occur in any section of the keyword file (e.g. `$MODEL, $LOAD CASE` etc.).

This section contains the following keywords:

- `#INCLUDE` is used to specify the name of an external keyword file to be included in the present keyword file.

1.8.19.1 #INCLUDE

PURPOSE

NOTE: Use of #INCLUDE can result in very slow program performance. As a result of this it has been deprecated as of Flexcom 8.4 and it's use is strongly discouraged. It will be removed in a future version of the software. The alternative is to use the FILE= option in keywords, where available.

To specify the name of an external keyword file to be included in the present keyword file.

`#INCLUDE` is typically used to simplify the specification of input data, and to reduce the complexity of the main keyword file. For example, vessel data (such as RAOs and auxiliary profile data) are typically consistent across all models/analyses which include the same vessel, so it makes sense to remove such data into external files, thus reducing the risk of any errors occurring in data input specification.

THEORY

Refer to Keyword Editor for further information on this feature.

KEYWORD FORMAT

A single line containing the name of the keyword file to be included.

```
#INCLUDE File Name
```

It should be noted that strictly speaking `#INCLUDE` is not a keyword – rather it is a directive which instructs Flexcom to insert the entire contents of an additional file at a specific location within a keyword file. Unlike standard keywords which only occur once within a single keyword file, `#INCLUDE` can occur at multiple locations, as many times as necessary. Given that the directive is universal – it can occur in any section of the keyword file (e.g. `$MODEL`, `$LOAD CASE` etc.), and is repeatable, it does not have any corresponding input in the table view.

File Name should be comprised of the name of the included file with its extension. It may optionally include the entire path of the file, if it is located in a different directory to the current keyword file. If the file name or any part of its path contains spaces then it should be enclosed in double quotation marks.

1.9 Theory

This section provides a comprehensive background on the modelling and analytical capabilities of the software. It is not intended to be read from cover-to-cover. You should refer to it when interested in further detail on any particular aspect of the program.

- Fundamentals covers some basic topics which all Flexcom users should be familiar with, namely Unit Systems, Keyword Parameterisation and Axes and Displacements. It also discusses the Finite Element Formulation in detail for interested readers.

- Model Building presents information on all aspects relating to model creation in Flexcom, covering areas such as Geometry, Geometric Properties, Hydrodynamic Properties, Special Element Types, Pipe-in-Pipe, Enviroment, Seabed Interaction, Vessels and Vessel Motions, Contact Surfaces and Line Clashing.

- Applied Loading discussed the various options regarding load application in Flexcom, including Internal Fluid, Current Loading, Wave Loading, Additional Loading Options such as user-subroutines. It also discusses the application of Boundary Conditions, and more specialised areas such as Wake Interference and Coupled Analysis.

- Postprocessing summarises the comprehensive range of postprocessing options available in Flexcom, and includes sections on Database Postprocessing, Summary Postprocessing, Summary Postprocessing Collation, Clearance & Interference Postprocessing and Code Checking. Custom Postprocessing provides additional options for advanced users, facilitating the development of specialised postprocessing tools tailored to meet specific requirements, including areas such as the Excel Add-In, Database Access Routines, and VBA code.

1.9.1 Fundamentals

This section contains information on the following topics.

- Unit Systems explains how unit systems work in Flexcom. It outlines the differences between files employing user-defined units and those with an explicit unit system.

- Keyword Parameterisation discusses the powerful parameterisation facility. Pre-defined input parameters form an integral part of a typical model specification. Once a parameter is altered, all dependent parameters are automatically updated, with consequent savings for model setup effort. Environmental simulation is also seamless, as one parameterised master template keyword file may be used to generate all the required input files (e.g. to accommodate a matrix of varying wave periods and headings).

- Axes and Displacements introduces some concepts which are fundamental to the internal operation of Flexcom.

- Finite Element Formulation discusses Flexcom's finite element formulation in detail.

1.9.1.1 Units Systems

This section explains how unit systems work in Flexcom. It outlines the differences between files employing user-defined units and those with an explicit unit system. It also explains how an existing project, containing files employing user-defined units, may be converted to one with an explicit unit system.

Further information is contained in the following sections:

- Standard Unit Systems introduces the standard Metric and standard Imperial unit systems, distinguishes between base and derived units, and also provides a complete Units Reference Guide.

- Files With Explicit Units discusses keyword files which use either the standard metric (.KEYXM file extension) or standard imperial (.KEYXI file extension) unit system.

- Files With User-Defined Units discusses legacy keyword files which do not include explicit units (.KEYX file extension).

- Units Project Importer describes the convertor program which allows you to convert legacy projects, which do not include explicit units, into one of the standard unit systems.

Standard Unit Systems

BASE UNITS

Units systems are composed of base units, measures which cannot be represented in any other way, and derived units. Derived units can be represented in terms of a combination of base units.

Flexcom requires base units for the dimensions of length, mass, time and temperature. All other units in the program are derived from these four base units. In fact, the use of temperature is so extremely limited that we will confine our discussion to length, mass and time from here on.

STANDARD UNIT SYSTEMS

There are two base unit systems commonly used in Flexcom. A metric system based on meter-kilogram-second and an imperial system based on foot-slug-second. We will refer to these two systems as Standard Metric and Standard Imperial.

Unit System	Length [L]	Mass [M]	Time [T]
Standard Metric	Meter	Kilogram	Second
Standard Imperial	Foot	Slug	Second

DERIVED UNITS

Taking the above systems as examples, we can see how derived units are obtained. Acceleration, for example, is length over time squared. Dimensionally this can be represented as $[L][T]^{-2}$. In the Standard Metric system this gives acceleration units of m/s^2. In Standard Imperial the units of acceleration would be ft/s^2. Building on our definition of acceleration, we know that force is mass times acceleration. Dimensionally, we can take our representation for acceleration and add mass, giving $[M][L][T]^{-2}$. This means that, in the Standard Metric and Standard Imperial systems, force units are $kg.m/s^2$ (i.e. Newtons) and $slugs.ft/s^2$ respectively.

KEYWORD FILE EXTENSIONS

Keyword files which use either the Standard Metric or Standard Imperial unit system have file extensions of .KEYXM (Metric) or .KEYXI (Imperial), respectively. Refer to Files with Explicit Units for further information.

Note also that Flexcom keyword files with the .KEYX extension are said to be using User-Defined Units. These are typically legacy keyword files which were created with an earlier version of the software prior to the release of Flexcom 8.4. Recommended practice is to convert any such files into one of the Standard Unit Systems using the Units Project Importer.

UNITS REFERENCE GUIDE

For a full list of the units accepted by Flexcom, in both metric and imperial systems, refer to Units Reference Guide.

Units Reference Guide

INTRODUCTION

This section present a complete list of all units accepted by Flexcom.

When it comes to suffix in files with explicit units, the system accepts a very wide range of variation, so you should be able to type values in naturally without worrying about what the program wants. Variations such as plurals, spacing and even alternative spellings are accommodated. For example, the following variations are all accepted by Flexcom...

- `m`
- `metre`
- `meter`
- `metres`
- `meters`

e.g. the following keyword is valid to define 4 nodes with x, y, z coordinates in a file with explicitly defined units (.KEYXM or .KEYXI):

```
*NODE
 1, <10m>, <20 ft>, <15 meter>
 2, <12 feet>, <18 metres>, <15'>
 3, <1200cm>, <800in>, <210">
 4, <8 foot>, <12000 mm>, <300 inches>
```

METRIC SYSTEM

Dimension	Metric	
	Standard	Others include
Length	Metre (m)	Millimetre (mm) Centimetre (cm) Kilometre (km)
Mass	Kilogram (Kg)	Gram (g) Tonne
Time	Second (s)	Minute (min) Hour (h) Day (d)
Temperature	Degree Celsius (°C)	
Angle	Degree (°)	Radian (rad)
Area	Square metre (m^2)	Square millimetre (mm^2)
Volume	Cubic metre (m^3)	Litre (L) Cubic centimetre (cm^3)
Velocity	Metre/Second (m/s)	Kilometres per hour (kph) Metres per minute (mpm)
Acceleration	Metre/second squared (m/s^2)	
Force	Newton (N)	Kilonewton (kN) Meganewton (MN)
Moment	Newton metre (N.m)	Kilonewton metre (kN.m) Meganewton metre (MN.m)
Mass density	Kilogram/metre cubed (kg/m^3)	Tonne/metre cubed ($tonne/m^3$)
Mass/length	Kilogram/metre (kg/m)	Tonne/metre (tonne/m)

Inertia	Kilogram metre squared (kg.m^2)	Tonne metre squared (tonne.m^2)
Inertia/length	Kilogram metre (kg.m)	Tonne metre (tonne.m)
Stress/Pressure	Newton/metre squared (N/m^2)	Pascal (Pa) Kilopascal (kPa) Bar Megapascal (Mpa) Gigapascal (GPa)
Curvature	Inverse metres (1/m)	
Twist	Radian per metre (rad/m)	
Frequency	Hertz (Hz)	Radians/second (rad/s)
Viscosity	Metres squared/second (m^2/s)	
Stiffness	Newton/metre (N/m)	Kilonewton/metre (kN/m) Meganewton/metre (MN/m) Newton metre/metre squared (N.m/m^2) Kilonewton metre/metre squared (kN.m/m^2)
Thermal expansion	Per degree celsius (1/°C)	
Bending stiffness	Newton metre squared (N.m^2)	Kilonewton metre squared (kN.m^2) Meganewton metre squared (MN.m^2)
Stiffness/length	Newton/metre/metre (N/m/m)	Newton/metre squared (N/m^2) Kilonewton/metre/metre (kN/m/m) Kilonewton/metre squared (kN/m^2)
Torsional stiffness	Newton metre squared/radian (Nm^2/rad)	Kilonewton metre squared/radian (kNm^2/rad)
Second moment of area	metres to the four (ft^4)	

Drag moment of area	metres to the five (ft^5)	
Roll and pitch stiffness	Newton metre/degree (Nm/°)	Kilonewton metre/degree (kN.m/°)
Damping	Newton seconds/metre (N.s/m)	Kilonewton seconds/metre (kN.s/m)
Quadratic damping	Newton seconds squared/metre squared ($N.s^2/m^2$)	Kilonewton seconds squared/metres squared ($kN.s^2/m^2$)
Pitch damping	Newton metre second/degree (Nm.s/°)	Kilonewton metre second/degree (kNm.s/°)
Couple damping	Newton second/degree (N.s/°)	Kilonewton second/degree (kN.s/°)
Spectral value	metre squared seconds ($m^2.s$)	
RAO rotational	Degrees/second (°/s)	Radian/second (°/s)
Rotational acceleration	Degree/second squared ($°/s^2$)	Radian/second squared (rad/s^2)
Weight/length	Newton/metre (N/m)	Kilonewton/metre (kN/m)
Circular frequency	Degree/second (°/s)	Radian/second (rad/s)
Impulse	Newton seconds (N.s)	Kilonewton seconds (kN.s) Meganewton seconds (MN.s)
Power	Watts (W)	Newton metre/second (N.m/s) Kilowatts (kW) Megawatts (MW)

IMPERIAL SYSTEM

Dimension	Imperial	
	Standard	Others include
Length	Feet (ft)	Inch (in) Yard (yd)

		Mile (mi)
Mass	Slug	Pound mass (lbm) Ton
Time	Second (s)	Minute (min) Hour (h) Day (d)
Temperature	Degree Fahrenheit (°F)	
Angle	Degree (°)	Radian (rad)
Area	Square foot (ft^2)	Square inch (in^2)
Volume	Cubic foot (ft^3)	Cubic inch (in^3) US gallon Oil barrel (bbl)
Velocity	Foot/Second (ft/s)	Miles per hour (mph) Feet per minute (fpm)
Acceleration	Foot/second squared (ft/s^2)	
Force	Pounds force (lbf)	Kilo pound force (Kip)
Moment	Pound force feet (lbf.ft)	Kilo pound force feet (kip.ft)
Mass density	Slug/foot cubed ($slug/ft^3$)	Pound mass/foot cubed (lbm/ft^3) Pound mass/US gallon cubed (ppg) Ton/foot cubed (ton/ft^3)
Mass/length	Slug/foot (slug/ft)	Pound mass/foot (lbm/ft) Ton/foot (ton/ft)
Inertia	Slugs feet squared ($slugs.ft^2$)	Ton feet squared ($ton.ft^2$)
Inertia/length	Slugs feet (slugs.ft)	Ton feet (ton.ft)
Stress/Pressure	Pounds force/foot squared (lbm/ft^2)	Pound force per inch squared (Psi)

		Kilopound force per inch squared (Ksi) Kilopound force/Foot Squared (kip/ft^2)
Curvature	Inverse feet (1/ft)	
Twist	Radian per feet (rad/ft)	
Frequency	Hertz (Hz)	Radians/second (rad/s)
Viscosity	Feet squared/second (ft^2/s)	
Stiffness	Pounds force/foot (lbf/ft)	Kilo pound force/foot (kips/ft) Pounds force foot/foot squared (lbf.ft/ft^2) Kilo pound force foot/foot squared (kip.ft/ft^2)
Thermal expansion	Per degree fahrenheit (1/°F)	
Bending stiffness	Pounds force foot squared (lbf.ft^2)	Kilo pound force foot squared (kips.ft^2)
Stiffness/length	Pounds force/foot/foot (lbf/ft/ft)	Pounds force/foot squared (lbf/ft^2) Kilo pound force /foot/foot (kips/ft/ft) Kilo pound force/foot squared (kips/ft^2)
Torsional stiffness	Pounds force foot squared/radians (lbf.ft^2/rad)	Kilo pound force food squared/radian (kips.ft^2/rad)
Second moment of area	Feet to the four (ft^4)	
Drag moment of area	Feet to the five (ft^5)	
Roll and pitch stiffness	Pounds force feet/degree (lbf.ft/°)	Kilo pound forcefeet/degree (kips.ft/°)

Damping	Pounds force seconds/foot (lbf.s/ft)	Kilo pound force second/foot (kips.s/ft)
Quadratic damping	Pounds force seconds squared/feet squared (lbf.s^2/ft^2)	Kilo pound force seconds squared/feet squared (kips.s^2/ft^2)
Pitch damping	Pounds force foot second/degree (lbf.ft.s/°)	Kilo pound force foot seconds /degree (kips.ft.s/°)
Couple damping	Pounds force seconds/degree (lbf.s/°)	Kilo pound force seconds/degree (kips.s/°)
Spectral value	Feet squared seconds (ft^2.s)	
RAO rotational	Degrees/second (°/s)	Radian/second (°/s)
Rotational acceleration	Degree/second squared (°/s^2)	Radian/second squared (rad/s^2)
Weight/length	Pounds force/foot (lbf/ft)	Kilo pound force/foot (kips/ft)
Circular frequency	Degree/second (°/s)	Radian/second (rad/s)
Impulse	Pound force seconds (lbf.s)	Kilo pound force second (kips.s)
Power	Pound force foot/second (lbf.ft/s)	Horsepower (hp)

Files With Explicit Units

OVERVIEW

Since Flexcom 8.4, it is possible to specify explicitly that a keyword file is in either the Standard Metric or Standard Imperial unit system. This is done by naming the file with the .KEYXM (Metric) or .KEYXI (Imperial) extension, as required.

There are several benefits to doing this. Because the program now knows with certainty what your inputs are, it can report units across the program with confidence and allow you to work with non-standards units in an entirely new and extremely flexible way.

One of the consequences of starting to work with explicit units is that you must have consistency along your restart chain. For example, if your initial static file is .KEYXM (Metric), then all subsequent restarts from this analysis must also be .KEYXM.

BASE UNITS

By default, all entries in the keyword file, when entered numerically, are still in unscaled base or derived units. This means that you don't have try to remember if diameter is specified in millimetres, centimetres or meters. If it's a length, it's meters. This also means, if your existing files employ user-based units, but conform to either the Standard Metric or Standard Imperial system, only a name-change is required to convert them.

To illustrate, consider this geometric properties specification from a Metric file:

```
*GEOMETRIC SETS
 OPTION=FLEXIBLE
 SET=mat_Production
 114000, 114000, 1.3e+007, 1.8e+009, 171, 1, 0.254, 0.3456, 0,3456, 0
```

CONVENIENT UNITS

What however if you have a keyword file, and you'd like to specify diameter in millimetres? That's easily done. A new syntax has been developed, which works equally in the keyword editor or in the table view. Here's an equivalent specification where EA is specified in Meganewtons and all diameters are specified in millimetres:

```
*GEOMETRIC SETS
 OPTION=FLEXIBLE
 SET=mat_Production
 114000, 114000, 1.3e+007, <1800MN>, 171, 1, <254mm>, <345.6mm>, <345.6mm>, 0
```

You can see how the <value-unit suffix> syntax can be used to specify a different unit to that expected. This system is highly flexible and understands most commonly used units.

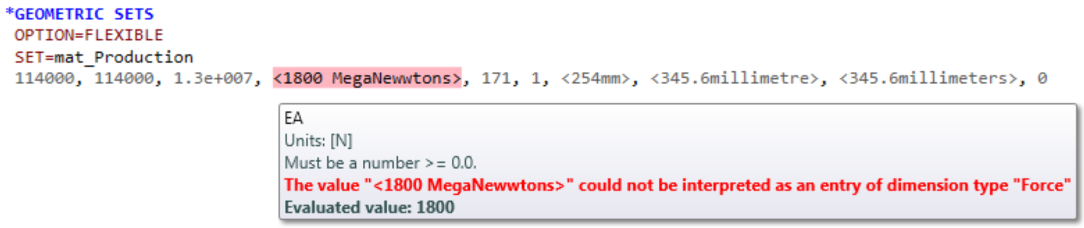
UNIT SUFFIXES

When it comes to suffix, the system accepts a very wide range of variation, so you should be able to type values in naturally without worrying about what the program wants. Variations such as plurals, spacing and even alternative spellings (meter/metre) are accommodated.

For examples of this, take a look at this equally valid specification:

```
*GEOMETRIC SETS
 OPTION=FLEXIBLE
 SET=mat_Production
 114000, 114000, 1.3e+007, <1800 MN>, 171, 1, <254mm>, <345.6millimetre>, <34
```

You need not worry about making a mistake. If you type in something which the software does not recognise, the relevant input will be highlighted, and you will receive a warning message in the Keyword Syntax Issues window.



MIXING UNIT SYSTEMS

The more eagle-eyed reader may have noticed that the first diameter specification looks like it has been converted from an imperial value (254mm = 10inches), and indeed it is. Here is where the system becomes even more useful. You can use Imperial units in a Metric file and vice versa. Here is the example specification again:

```
*GEOMETRIC SETS
 OPTION=FLEXIBLE
 SET=mat_Production
 114000, 114000, 1.3e+007, <1800 MN>, 171, 1, <10">, <13.6in>, <13.6inch>, 0
```

Now all diameters have been specified directly in inches. Furthermore they've been specified with three equally valid suffixes to further emphasise the system's flexibility.

Being able to enter values exactly as you've received them eliminates a potential source of error in your model, especially when a conversion between unit systems was previously required.

PARAMETERS AND EQUATIONS

Some consideration needs to be given to the use of parameters and equations in a file with explicit units. Numerical parameters defined using the `*PARAMETERS` keyword do not have any units associated with them, and you should be mindful of this when using them later in the file.

Let's consider an example where the inner diameter, intended to be in inches, is specified as a parameter

```
*PARAMETERS
 Di, <10in>
```

In this case, the value <10in> is recognised as a length and automatically converted to the appropriate base unit of length, metres in this instance. So the value of Di is 0.254.

If we are to use that value directly in the *GEOMETRIC SETS keyword thus:

```
*GEOMETRIC SETS
 OPTION=FLEXIBLE
 SET=mat_Production
 114000, 114000, 1.3e+007, 1.8e+009, 171, 1, =[Di], 0.3456, 0,3456, 0
```

This will be correct, because the program will consider this the same as typing the value 0.254 directly into that keyword.

However if we were using this value to set something else, such as the name of the analysis:

```
*NAME
 =["Analysis " + Do + " inch Pipe"]
```

That would resolve to “Analysis 0.254 inch Pipe” which is not what is intended, so you must be mindful of this.

An alternative approach, equally valid, is to specify 10 as a number, with no units:

```
*PARAMETERS
 Di, 10
```

And then specify the conversion at the point of use:

```
*GEOMETRIC SETS
 OPTION=FLEXIBLE
 SET=mat_Production
 114000, 114000, 1.3e+007, 1.8e+009, 171, 1, <=[Di]in>, 0.3456, 0,3456, 0
```

If we took this approach then, the analysis name would also be constructed correctly:

```
*NAME
 =["Analysis " + Do + " inch Pipe"]
```

That would resolve to “Analysis 10 inch Pipe” which is consistent with requirements.

In theory both approaches are valid. However the first approach is strongly recommended. Where the relevant units are identified at source, any potential for error in subsequent data specification is greatly reduced (in the second approach, you have to remember to insert the correct units every time the parameter is referenced). Further information is also provided in Adding Units to Parameters.

GENERIC UNIT CONVERSIONS

All of the examples considered up to this point are intrinsic conversions, where Flexcom automatically converts a user-defined value into the base units for that parameter type in the relevant standard unit system. For example, if you are using a metric file, then <1kN> is automatically converted to 1000N. Similarly, <1kip> would be automatically converted to 4448.2N.

For complete flexibility, Flexcom also provides a Generic Conversion Function which allows you to convert a user-defined value into any other acceptable unit type for that parameter, which need not necessarily correspond to the base unit for that parameter. For example, you could convert a value from kN to kips, as follows:

```
$PREPROCESSOR
*PARAMETERS
 Tension, =[convert(1,"kN","kips")]
```

POSTPROCESSING

Another very useful feature is the ability to postprocess results into any desired units, regardless of the unit system used for the actual analysis. For example, if a system was modelled by an engineer in the US using imperial units, a colleague in the UK can still take those results and examine tension in Kilonewtons and stress in Megapascals.

For illustrative purposes, the keywords shown below are used to request two snapshot plots of effective tension, the first in Kilonewtons (kN) and the second in Kilo Pounds Force (kips).

```
$DATABASE POSTPROCESSING
*SNAPSHOT
 TYPE=FORCE, SET=All
 7, 1, 1.0, , UNITS=kN
 TITLE=Effective Tension (kN)
 TYPE=FORCE, SET=All
 7, 1, 1.0, , UNITS=kips
 TITLE=Effective Tension (kips)
```

The keyword commands shown above will work in either a metric (.keyxm) or an imperial (.keyxi) postprocessing file.

Files With User-Defined Units

DATA INPUT

Flexcom keyword files with the .KEYX extension are said to be using user-defined units. These are typically legacy keyword files which were created with an earlier version of the software prior to the release of Flexcom 8.4. Recommended practice is to convert any such files into one of the Standard Unit Systems using the Units Project Importer.

When performing data-input the UI will suggest units for keyword entries, consistent with the Standard Metric and Standard Imperial systems. The user is free to ignore these suggestions as long as they maintain consistency between different entries.

Although it is possible to adopt a unit system based on a non-standard series of base units, centimetre-kilogram-hour for example, in practice this almost never happens. To do so requires enormous care to establish the correct units for every required program input. Therefore, most keyword files, even when using user-defined units, tend to conform to either the Standard Metric system or Standard Imperial system.

POST-PROCESSING

Deviation from the two standard unit systems is extremely rare. Because of this, even when using user-defined units, Flexcom has evolved to make assumptions, when post-processing, about the actual unit system being used. This allows it to label and scale plots appropriately.

Flexcom makes this assumption based on the value of the acceleration due to gravity entry in the `*OCEAN` keyword. The value is compared to the Standard Metric value (9.81m/s^2) and the Standard Imperial value (32.2ft/s^2), if it falls within a given tolerance of either of these two values then the respective unit system is assumed.

Units Project Importer

OVERVIEW

If you have an existing project which uses user-defined units, but you would like to take advantage of the new unit related features that come from specifying an explicit unit system, then there is a convertor program you can use to copy your project and convert it to the new file extensions. All you must ensure is that your existing project already conforms to the Standard Metric system or the Standard Imperial system. Most projects do.

OPERATION

The Units Importer copies all your project files from one directory to another, preserving the project structure as it does so. It renames the files from .KEYX to .KEYXM or .KEYXI, depending on the specified unit system. It also copies the project file and any additional input files as it goes. Where file references inside the copied files explicitly reference the .KEYX extension, these references will be updated.

As it copies files, the program will check any instances of the `*OCEAN` keyword it finds to ensure that the value of gravity seems to correspond to the selected unit system. If it does not, a warning will be issued.

What the Units Importer does not do is change any numerical values in the files. It cannot be used to convert from one unit system to another.

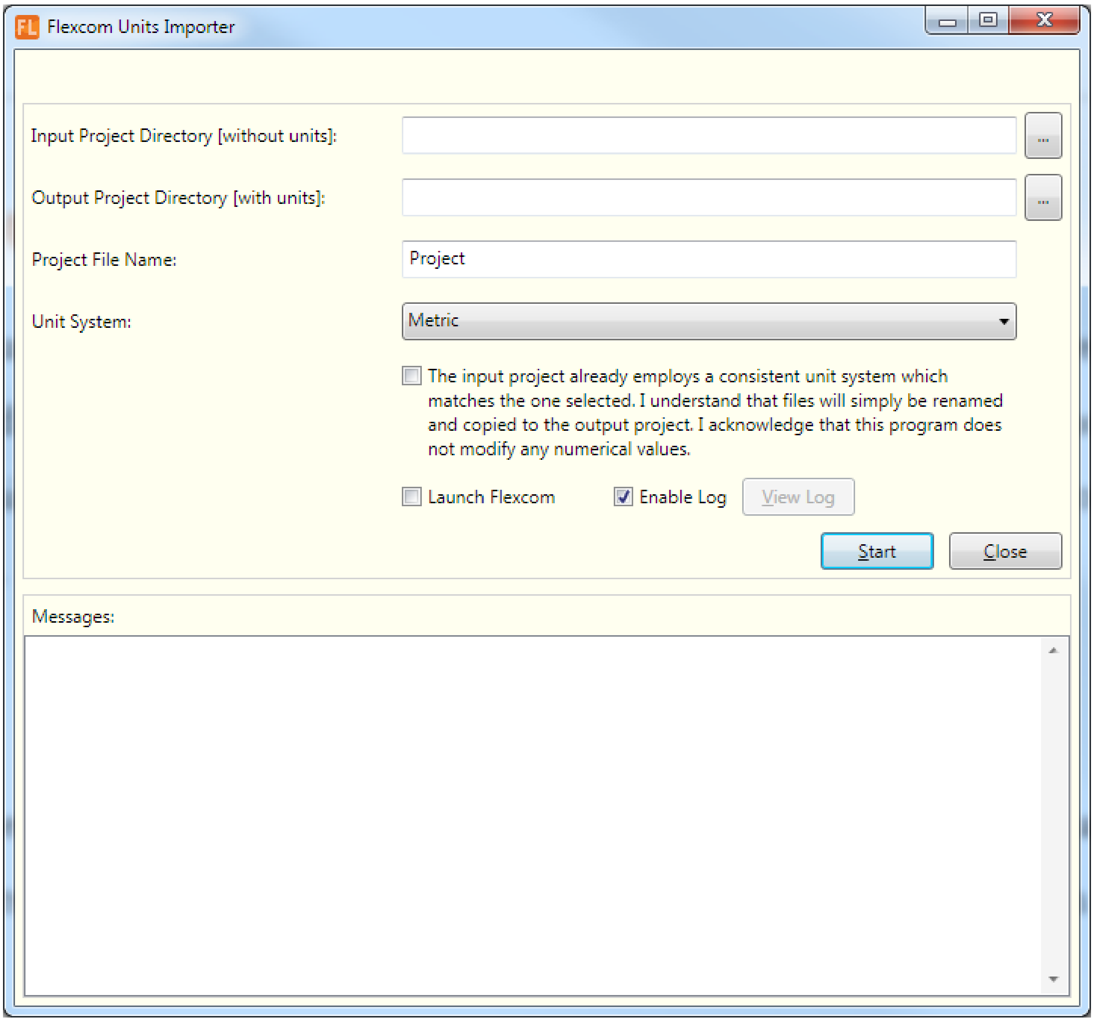
Refer to Running the Units Importer for further information about this feature.

Running the Units Importer

The Units Importer can be found in the Windows Start menu or under the Tools menu in the Flexcom Main UI.

The program asks you to pick an input directory, containing files with the .KEYX extension and no explicit unit system, and an output directory to place the copied and renamed files. These directories must be distinct and cannot be nested within each other.

If there is a project file in the existing input directory, the program will automatically pick this up and use its name by default. You can override this value if you choose.



The unit system you choose must correspond to the unit system actually in use in the input files. You must confirm that you understand the operation of the program and that no actual unit conversion is taking place by ticking the appropriate check box.

The new project can optionally be launched in Flexcom once import is complete. Check the appropriate box to enable this option.

By default the program maintains a log as it works. This is in addition to messages shown on the dialog and is more detailed. It is recommended that you use this option and press the “View Log” button after import to check for messages, warning and errors.

Once you are ready, press the “Start” button. Importation will take from between a few seconds to a few minutes, depending on the size of your project.

1.9.1.2 Keyword Parameterisation

OVERVIEW

The rationale behind keyword parameterisation is best illustrated by considering a sample design scenario. Supposing you have performed an initial static analysis of an offshore structure, and you wish to examine the structural response to a series of regular wave load cases, each having a unique wave period. In earlier versions of Flexcom, this would require the creation and storage of separate keyword files for each regular wave load case. The situation becomes even more complex when separate vessel offset and current load cases are also taken into consideration. So the entire process could be quite time consuming, particularly for large load cases matrices, with the software offering little to assist you by way of automation.

The keyword parameterisation facility means that a single keyword file is all that is required to model a series of load case variations about a base model. Flexcom will then automatically create all the keyword files required to examine each load case.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Parameters
- Equations
- Keyword Based Variations
- Spreadsheet Based Variations

RELEVANT KEYWORDS

Parameters

- `*PARAMETERS` is used to define parameters whose names may be referenced subsequently in the definition of other input variables.

Keyword Based Variations

- `*VARIATION` is used to define parameter or keyword variations.

- *COMBINATIONS is used to control the names and locations of generated keyword files, along with the required analysis combinations.

If you would like to see an example of how these keywords are used in practice, refer to K02 - Worked Example - Complex.

Spreadsheet Based Variations

- *EXCEL VARIATIONS is used to generate keyword files based on a parameter matrix contained in an Excel workbook.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser.

Parameters

OVERVIEW

You may define constant parameters within a keyword file, and reference these parameters when defining input variables. If you choose to define any such parameter, their definitions must appear at the beginning of the keyword file, under the $PREPROCESSOR section. When referencing a parameter, the variable definition must be preceded by the characters =[and followed by the] character. Specifically the variable definition will appear in the format =[Parameter], rather than just having an explicitly specified value. The facility is best illustrated via a simple example. Consider the following environmental specification:

– Ocean depth = 500m

– Seawater density = 1025 kg/m^3

– Acceleration due to Gravity = 9.81m/s

This may be specified directly in numerical format as follows:

```
$MODEL
*OCEAN
 500, 1025.0, 9.81
```

Alternatively, parameters may be defined initially, and then referenced subsequently:

```
$PREPROCESSOR
*PARAMETERS
 OceanDepth, 500
 SeawaterDensity, 1025
 Gravity, 9.81
```

```
$MODEL
*OCEAN
 =[OceanDepth], =[SeawaterDensity], =[Gravity]
```

You can mix parameters with standard inputs also – the following is another valid specification:

```
$PREPROCESSOR
*PARAMETERS
 OceanDepth, 500
 Gravity, 9.81

$MODEL
*OCEAN
 =[OceanDepth], 1025, =[Gravity]
```

Note also that there are some pre-defined System Parameters. You should not attempt to redefine these as parameter variables.

ADDING UNITS TO PARAMETERS

Unlike most inputs in Flexcom (e.g. line length, element diameter), user-defined parameters do not have any default units associated with them. This is because it is possible in theory to use a single parameter to define more than one, possibly unrelated, entries in the Flexcom input file. For example, you could define a parameter called 'General', and assign it a value of 5. This could then be used to define a length of 5m, a mass of 5kg, or even a time of 5s. Needless to say, this is not good practice.

In order to eliminate any possible ambiguity, it is highly recommended that you explicitly associate units with every parameter definition (assuming you are using a Keyword File with Explicit Units). For example:

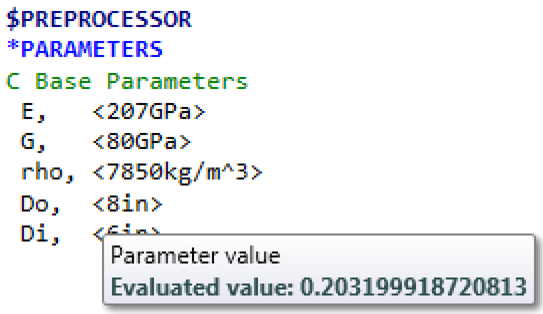
```
$PREPROCESSOR
*PARAMETERS
C Base Parameters
 E,   <207GPa>
 G,   <80GPa>
 rho, <7850kg/m^3>
 Do,  <8in>
 Di,  <6in>
```

All parameters are then automatically converted by Flexcom into the relevant base units. Given that the above is a metric file (file extension .keyxm), Flexcom automatically converts these parameters into SI base units as follows:

```
E     207000000000 Pa
```

```
G     80000000000 Pa
rho   7850 kg/m^3
Do    0.203199918720813 m
Di    0.15239993904061 m
```

Any of these intrinsic conversions may be verified by hovering the mouse cursor over a parameter definition in the Keyword Editor or Table Editor, or perhaps more conveniently, all of them may be checked via the Parameter Values window. For example, Do has been converted from 8 inches to 0.203m in the following example...



Even though rho is already in standard units, it is good practice to explicitly state the relevant units for clarity. This will also prevent rho being used subsequently to define any Flexcom input where the base units are not [mass/length^3].

Naturally it follows that you may wish to define some derived parameters based on these fundamental entries. For example:

```
$PREPROCESSOR
*PARAMETERS
C Base Parameters
 E,   <207GPa>
 G,   <80GPa>
 rho, <7850kg/m^3>
 Do,  <8in>
 Di,  <6in>

C Derived Parameters
 I,  =[PI/64*(Do^4-Di^4)]
 J,  =[2*I]
 A,  =[PI/4*(Do^2-Di^2)]
 EI, =[E*I]
 EA, =[E*A]
 GJ, =[G*J]
```

Explicitly specifying units along with parameter definitions ensures consistency of any further computations and derived parameters. For example, even though Do and Di have both been specified in inches, they are immediately converted internally into metres (the base unit for length in the SI system), and consequently any derived parameters such as I, J and A are computed correctly.

DEFINING LISTS OF PARAMETERS

If you have several parameters which are related in some way, it may be more convenient to define them in a list format. This allows you to assign a common prefix to each parameter, and is a more economical way of writing as several parameter definitions may be included on the same line. For example:

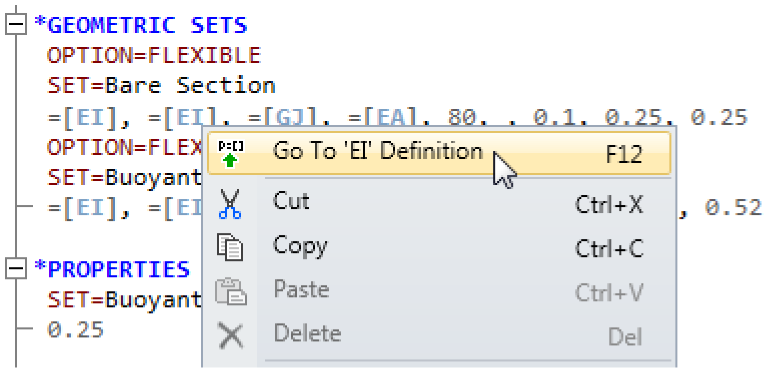
```
$PREPROCESSOR
*PARAMETERS
 Pipe,  / Do:<8in>, Di:<6in> /
 Cable, / Do:<2in>, Di:<0in> /
```

In this case Flexcom creates four separate parameters - Pipe_Do, Pipe_Di, Cable_Do and Cable_Di. This is equivalent to the following individual definitions, but the list type above possibly looks neater.

```
$PREPROCESSOR
*PARAMETERS
 Pipe_Do,  <8in>
 Pipe_Di,  <6in>
 Cable_Do, <2in>
 Cable_Di, <0in>
```

CHECKING PARAMETER DEFINITION

When you are referencing a parameter, you can quickly check how the parameter has been defined. Simply right-click on the parameter reference in the Keyword Editor, and select *Go to 'Parameter' Definition,* as illustrated below...



You can also use the Smart Select feature to highlight all instances of the parameter within the keyword file.

RELEVANT KEYWORDS

- `*PARAMETERS` is used to define parameters whose names may be referenced subsequently in the definition of other input variables.

If you would like to see an example of how this keyword is used in practice, refer to C02 - Multi-Line Flexible System or C03 - Turret Disconnect.

System Parameters

The following are pre-defined constants. You should not attempt to redefine these are parameter variables.

System Parameters

Parameter Name	Variable Type	Assigned Value
PI	Real number	3.14159265358979
TRUE	Character	True
FALSE	Character	False
ApplicationPlatform	Character	"32bit" or "64bit"

Equations

MATHEMATICAL EQUATIONS

Numerical variables may also be replaced by mathematical equations. For example, you might like to position a vessel in the region of the mean water line, as follows:

```
$PREPROCESSOR
*PARAMETERS
 OceanDepth, 500

$MODEL
*VESSEL
 VESSEL=FPSO
 INITIAL POSITION==[OceanDepth+5], 0, 0, 0
```

This means that the vessel reference point will always be positioned at 5m above the mean water line, regardless of the water depth.

Refer to Mathematical Operators and Mathematical Functions for a full list of the mathematical features supported by Flexcom.

LOGICAL EXPRESSIONS

It is also possible to use logical expressions. Let's suppose you would like to alternate between metric and imperial unit systems. Unit conversion is now handled more conveniently with the CONVERT function (see below) and dedicated keyword file extensions (keyxm and keyxi) but this is useful to illustrate logical expressions. The following specification converts water depth from metres to feet, and seawater density from kg/m^3 to slugs/ft^3.

```
$PREPROCESSOR
*PARAMETERS
 Gravity, 9.81
 LengthFactor, 3.28084
 DensityFactor, 0.0019403233
 OceanDepth, =[if(Gravity<10, 500, 500*LengthFactor) ]
 SeawaterDensity, =[if(Gravity<10, 1025, 1025*DensityFactor) ]

$MODEL
*OCEAN
 =[OceanDepth], =[SeawaterDensity], =[Gravity]
```

Refer to Logical Operators and Functions for a full list of the logical features supported by Flexcom.

STRING EXPRESSIONS

The preceding sections have focused primarily on the use of parameters in defining numerical values, or expressions which equate to a numerical value. It is also possible to use parameters to define string expressions; for example:

```
$PREPROCESSOR
*PARAMETERS
 Offset, -100
 OffsetFolder, Near
```

In this simple example, the parameter OffsetFolder has the value Near. If however the folder name is more complex and contains spaces, then it needs to be wrapped in double quotation marks:

```
$PREPROCESSOR
*PARAMETERS
 Offset, -100
 OffsetFolder, "C:\Test Case\Near"
```

Without quotation marks, the program would interpret this as a pair of values (C:\Test and Case\Near) following OffsetFolder with a space between them, not what is intended. The quotation marks are therefore required in this case.

When constructing a path name containing spaces using an equation, you need a way to make sure that it's wrapped in double quotation marks. That is straightforward to do; in an equation, the built-in constant QQ represents a double-quotation mark. For example:

```
$PREPROCESSOR
*PARAMETERS
BaseFolder, "C:\Test Case\"
OffsetFolder, =[QQ + BaseFolder + "Near" + QQ]
```

This will result in the OffsetFolder having the value of "C:\Test Case\Near", including quotation marks.

Note also that any unusual characters may be defined explicitly using ASCII character codes, using the general function char(N), where N is the ASCII code corresponding to the required character. In fact, QQ is defined internally using char(34).

Refer to String Operators and Functions for a full list of the logical features supported by Flexcom.

RELEVANT KEYWORDS

- `*PARAMETERS` is used to define parameters whose names may be referenced subsequently in the definition of other input variables.

If you would like to see an example of how mathematical equations are used in practice, refer to C02 - Multi-Line Flexible System or C03 - Turret Disconnect.

If you would like to see an example of how string expression are used in practice, refer to K02 - Worked Example - Complex.

Mathematical Operators

The available mathematical operators are summarised in the table below.

Mathematical Operators (In Order of Evaluation Precedence)

Function	Description
A ^ B	A to the power of B
A * B	A multiplied by B
A / B	A divided by B
A – B	A minus B
A + B	A plus B.

Mathematical Functions

The available mathematical functions are summarised in the table below.

Mathematical Functions

Function	Description
ABS(A)	Returns the absolute value of A

ACOS(A)	Returns the arccosine of A, in radians
ACOSH(A)	Returns the inverse hyperbolic cosine of A, in radians
ASIN(A)	Returns the arcsine of A, in radians
ASINH(A)	Returns the inverse hyperbolic sine of A, in radians
ATAN(A)	Returns the arctangent of A, in radians
ATANH(A)	Returns the inverse hyperbolic tangent of A, in radians
ATANTWO(A,B)	Returns the arctangent treating A and B as x- and y-coordinates, in radians
CEIL(A)	Returns the nearest integer value greater than or equal to A.
CHAR(A)	Returns the character corresponding to the ASCII code for integer A
COS(A)	Returns the cosine of A, where A is in radians
COSH(A)	Returns the hyperbolic cosine of A, where A is in radians
CONVERT(A,B,C)	Convert a numeric value from one dimension to another. The original numeric value is A. B and C are strings which represent the current dimension and the dimension to convert to, respectively.

	For example, to convert 21 inches to an equivalent value in metres: CONVERT(21, “in”, “m”) = 0.5334
DEGREES(A)	Returns the angle in degrees, where A is in radians
EXP(A)	Returns e raised to the power of A
FLOOR(A)	Returns the nearest integer value less than or equal to A.
INT(A)	Returns the nearest integer value to A.
LN(A)	Returns the natural logarithm of A
LOG(A)	Returns the base-10 logarithm of A
MAX(A,B)	Returns the larger of A or B
MIN(A,B)	Returns the smaller of A or B
MOD(A,B)	Returns the remainder after A is divided by B
RADIANS(A)	Returns the angle in radians, where A is in degrees
SQRT(A)	Returns the positive square root of A
SIN (A)	Returns the sine of A, where A is in radians
SINH (A)	Returns the hyperbolic sine of A, where A is in radians

TAN(A)	Returns the tangent of A, where A is in radians
TANH(A)	Returns the hyperbolic tangent of A, where A is in radians

Logical Operators and Functions

The available logical operators and functions are summarised in the table below.

Logical Operators and Functions

Function	Description
A < B	Returns TRUE if A less than B
A <= B	Returns TRUE if A less than or equal to B
A > B	Returns TRUE if A greater than B
A >= B	Returns TRUE if A greater than or equal to B
A = B	Returns TRUE if A equals B
AND(A, B)	Returns TRUE if A and B are TRUE
IF(A, B, C)	If A is true, returns the value of B, else returns the value of C

NOT(A)	If A is TRUE, returns FALSE. If A is FALSE, returns TRUE/
OR(A, B)	Returns TRUE if either A or B are TRUE

String Operators and Functions

The available string operators and functions are summarised in the table below.

String Operators and Functions

Function	**Description**
A + B	Where either A and/or B are string values, they are concatenated.
FORMAT(A, B)	A is a numeric value to be formatted; B is a string describing the format to employ. Standard formats: E/e: Scientific Notation (upper and lower case 'E') F: Floating point G: General: Floating point or scientific notation, depending on value N: Floating point with digit grouping Standard formats can be followed by a number to indicate precision. For example, “F2” indicates floating point style with two digits after the decimal point. Fully customisable number formats are also available where:

	'0' represents the corresponding digit, or '0' if there is none '#' represents the corresponding digit '.' represents position of the decimal separator Other custom format options are available. See Microsoft .NET documentation for Double.ToString() for full details.

Keyword Based Variations

THEORY

While the use of equations may be helpful in certain circumstances, the main benefit of parameter definition probably lies in the area of load case variations. This area of application is illustrated in the following simple example. Supposing you wish to examine a range of near and far vessel offset cases, spanning ±100m, at 25m intervals. In earlier versions of Flexcom, this would require the creation and storage of a separate keyword file for each offset, but the keyword parameterisation facility means that a single keyword file is all that is required to model each load case. So you include all the required data in a single base keyword file, called “Offset.keyx”, restarting from an initial static analysis keyword file called “Static.keyx”, as follows:

```
$PREPROCESSOR
C Set up a vessel offset parameter
*PARAMETERS
 Offset, -100
 OffsetFolder, "Near"

C Set up the vessel offset variations
*VARIATION
 NAME=NearOffsets
 PARA=Offset, GEN=-100, -25, 25
 PARA=OffsetFolder, VALUE=Near
 NAME=FarOffsets
 PARA=Offset, GEN=25, 100, 25
 PARA=OffsetFolder, VALUE=Far

C Set up the analysis combinations
*COMBINATIONS
```

```
 DESCRIPTION=Offset Analyses, SUBDIRECTORY="%OffsetFolder%\%Offset%m  Offset'
 VARIATIONS=ALL
$LOAD CASE
*RESTART
 TYPE=DYNAMIC/STATIC, LAST="Static"

C Reference the (variable) vessel offset parameter
*OFFSET
 VESSEL=FPSO
 OFFSET=0.0, =[Offset], 0.0, 0.0, 0.0, 0.0
```

Flexcom will then automatically create a separate keyword file for you corresponding to each vessel offset, and the *OFFSET keyword will contain the relevant offset value. The names and locations of the files are controlled by the *COMBINATIONS keyword. For the sample data shown above, separate subdirectories are created using a combination of the parameter name (i.e. “%Offset%” in this case) and standard characters (i.e. “m Offset” in this case). Specifically, you will obtain 8 separate subdirectories, entitled “Near\-25 m Offset”, “Far\75m Offset” etc., located in the same directory as the base file. The keyword file names will be entitled “-25m Offset.keyx”, “75m Offset.keyx” etc.

You might typically proceed to defining a range of regular wave load cases, restarting from the vessel offset cases. Supposing you wish examine a range of regular waves, with periods spanning 6-24s, at 2s intervals. When these 10 regular waves are combined with the 8 vessel offsets, this results in a load case matrix totalling 80 separate dynamic analyses. To set up all of these keyword files individually would be quite tedious and time consuming, and this is where the keyword parameterisation facility really comes into its own. It’s possible to set up all the required analyses using a single base keyword file, called “Wave.keyx”, as follows:

```
$PREPROCESSOR
C Set up wave period and vessel offset parameters
*PARAMETERS
 Period, 6
 Offset, -100
 OffsetFolder, "Near"

*VARIATION
C Set up the wave period variations
 NAME=Waves
 PARA=Period, GEN=6, 24, 2

C Set up the vessel offset variations (copied from previous file)
  NAME=NearOffsets
  PARA=Offset, GEN=-100, -25, 25
  PARA=OffsetFolder, VALUE=Near
  NAME=FarOffsets
  PARA=Offset, GEN=25, 100, 25
  PARA=OffsetFolder, VALUE=Far
```

```
C Set up the analysis combinations
*COMBINATIONS
 DESCRIPTION=Wave Analyses for Offsets, SUBDIRECTORY="%OffsetFolder%\%Offset%
 VARIATIONS=ALL

$LOAD CASE
C Reference the (variable) wave period parameter
*WAVE-REGULAR
 1.0, =[Period], 0.0, 0.0

C Reference the (variable) wave period parameter
*TIME
 STEP=FIXED
 2, =[(Period*5)+2], =[Period*0.05], =[Period]

C Reference the (variable) vessel offset parameter
*RESTART
 TYPE=DYNAMIC/STATIC, LAST==[Offset + "m Offset"]
```

For the above specification, you will obtain 10 separate wave analyses, entitled "6s Wave.keyx", "8s Wave.keyx" etc., located in each of the 8 subdirectories created previously for the vessel offset analyses. Each dynamic analysis uses an appropriate regular wave period. A unit wave amplitude is used in all cases. Appropriate time variables are computed based on the regular wave period. Each dynamic analysis runs for 5 wave periods, using a time step of 1/20th of the wave period, with the loads ramped on over the first wave period.

The load case matrix is automatically determined by Flexcom based on the specified parameters and combinations. For the sample case described above, 10 different wave periods are combined with 8 different preceding vessel offset analyses, resulting in a load case matrix totalling 80 separate dynamic analyses.

For complete generality, it is also possible to add or remove entire keywords. This facility is intended to cater for variations which are difficult to accommodate by the standard parameter variations, but the option is probably rarely invoked in practice. For example, if you wished to examine different wave spectra or discretisation options, it might be more straightforward to replace the entire `*WAVE` keyword, rather than attempting to parameterise several different options within it.

While the preceding discussion provides a representative sample of the modelling possibilities using keyword parameterisation, it is by no means exhaustive. You could also use parameterisation to examine different finite element mesh densities, or to automatically compute structural properties required by Flexcom from the engineering inputs available to you. You can make the parameterisation definitions as simple or complex as you wish – the possibilities are endless really – feel free to experiment.

Note that the above example is ideally suited to regular waves, where the wave period tends to vary in fixed increments. Random seastates, which are typically characterised by a mean up-crossing period Tz (this does not vary in fixed increments), are better suited to the alternative Spreadsheet Based Variations modelling approach.

RELEVANT KEYWORDS

- `*PARAMETERS` is used to define parameters whose names may be referenced subsequently in the definition of other input variables.
- `*VARIATION` is used to define parameter or keyword variations.
- `*COMBINATIONS` is used to control the names and locations of generated keyword files, along with the required analysis combinations.

If you would like to see an example of how these keywords are used in practice, refer to J03 - Summary Postprocessing Collation.

Spreadsheet Based Variations

THEORY

The preceding example which illustrates Keyword Based Variations is relatively simplistic, in that the parameters vary in fixed increments. For example, regular wave periods tend to wave between 5s and 30s, in steps of 1.0s or 0.5s. However, random seastates are usually classified according to mean up-crossing period (Tz). The Tz parameter is used in conjunction with significant wave height (Hs), and both are typically presented together on a Scatter Diagram. The Hs and Tz values are generally not round integers, nor do they vary in fixed increments.

Spreadsheet based variations are much more powerful and flexible than keyword based variations. The parameters, and all of their associated variations, are first defined in a spreadsheet. The `*EXCEL VARIATIONS` keyword is then used to inform Flexcom where the relevant input data is located, in terms of an Excel workbook, a worksheet name, and a cell range within that worksheet. The keyword is also used to define the names and locations of all the generated keyword files. When you run a parameterised file which contains spreadsheet based variations, Flexcom opens the spreadsheet, extracts the relevant data, and then proceeds to generate all the required keyword files based on the master template.

RELEVANT KEYWORDS

Parameters

- *PARAMETERS is used to define parameters whose names may be referenced subsequently in the definition of other input variables.
- *EXCEL VARIATIONS is used to generate keyword files based on a parameter matrix contained in an Excel workbook.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser.

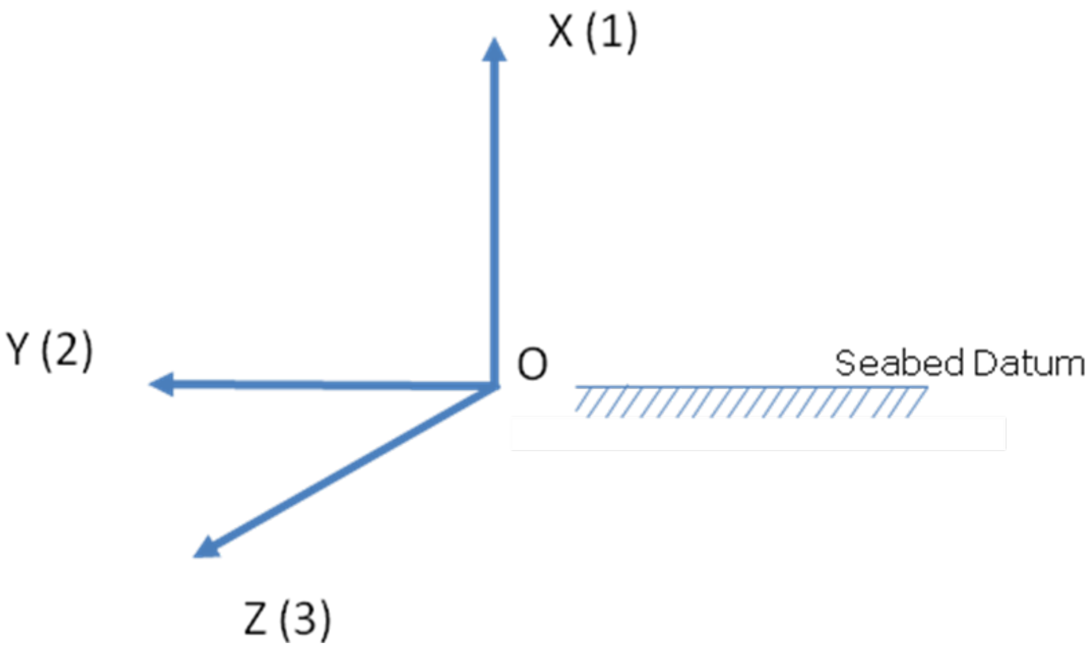
1.9.1.3 Axes and Displacements

The following sections introduce some concepts which are fundamental to the internal operation of Flexcom.

- Global Axes
- Global Degrees of Freedom
- User-Defined Axes

Global Axes

Flexcom uses a number of axis systems. Of these, the most important are the global or XYZ axes. These are the axes in which many (though by no means all) coordinates and loads are input, they are the axes in which the total system equations of motion are assembled and solved internally by Flexcom, and they are finally the axes in which motions and reactions at restrained locations are output. In this system, the global X axis is vertical, with the global Y and Z axes lying in a horizontal plane such that the system is right-handed. Zero datum is located at X=0, so for example, if you are using the default flat seabed profile, then the global Y and Z-axes both lie on the seabed. See below:



Global Axes

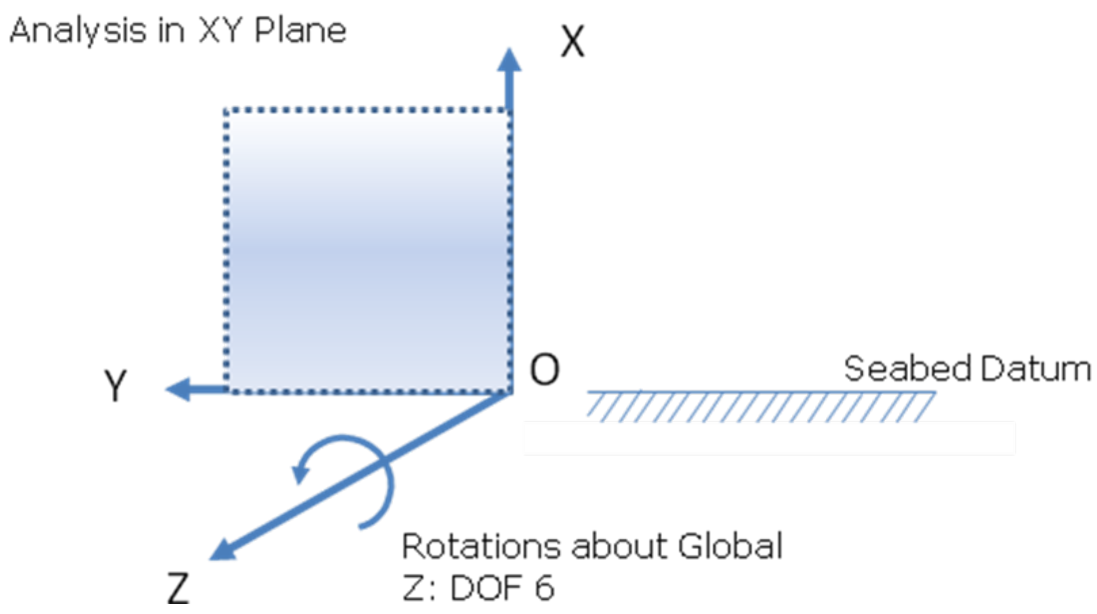
Global Degrees of Freedom

The independent components required to define the movement or displacement of a point on a structure are known as its degrees of freedom, or DOFs for short. A point requires 6 DOFs to define its movement – three translations (one vertical, two horizontal) and three rotations. In the Flexcom convention, a movement or displacement in the global X direction is denoted a DOF 1 translation, a displacement in global Y is a DOF 2 translation, while DOF 3 is used for a translation in the global Z direction.

The rotational displacements are denoted DOFs 4-6 in Flexcom. Explaining what these represent physically is a little more complicated than the translations case. Taken together, they represent the rotation in 3D space of a point from its initial orientation. You will no doubt be aware that finite (non-infinitesimal) 3D rotations cannot be represented as vectors, since the addition of such vectors is non-commutative, that is, the order in which the rotations are taken affects the result. In Flexcom this problem is handled by means of a consistent 3D kinematics formulation based on the correspondence between a specially defined rotation vector and a transformation matrix. Manipulation of the rotation is undertaken by means of operations on the transformation matrices. (You will note that as required matrix multiplication is non-commutative.)

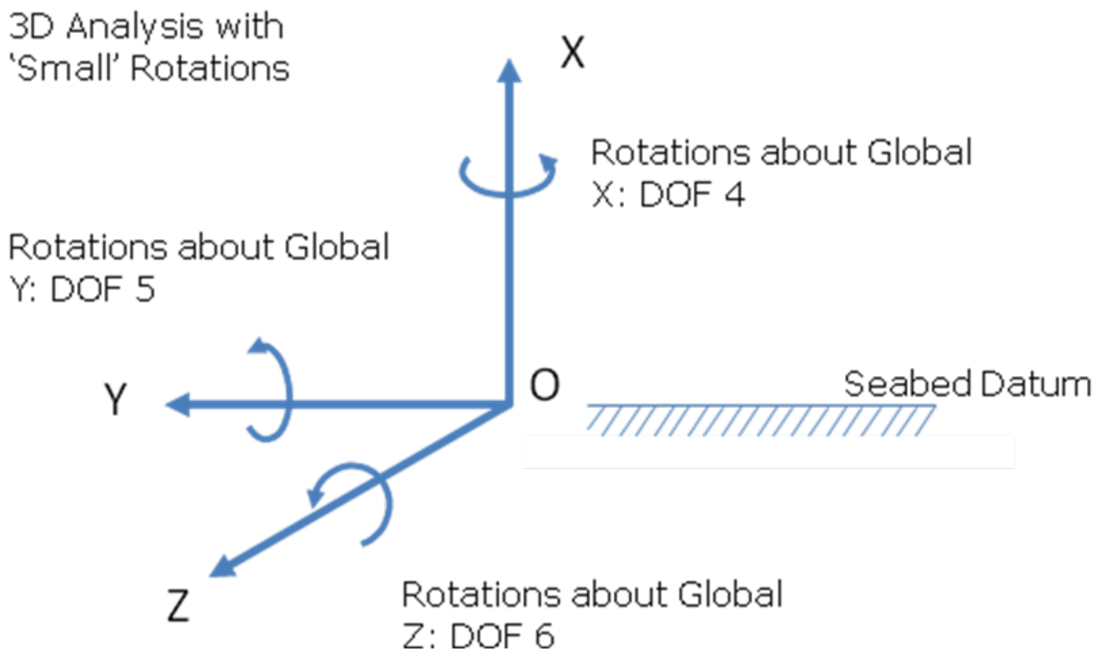
The cornerstone of this formulation is the definition of the rotation vector referred to above. The full details are omitted here, but, in summary, a rotation is represented by a vector such that i) the magnitude of the vector represents the magnitude of the rotation, and ii) the direction of the vector represents the axis of rotation. DOFs 4-6 in Flexcom are the components of this rotation vector. This means that in general they do not represent individual rotations about the global X, Y and Z axes.

While this is true in general, in many cases the situation is somewhat simplified. Firstly, in an analysis where the structure and environment are in either the XY or XZ plane, then in this 2D case there is obviously only one rotation component active (for example, DOF 6 if structure and environment are in the vertical XY plane, as illustrated below).



2D Analysis in XY Plane

Secondly, in a 3D analysis, the individual components will represent individual rotations about the global axes if the overall rotation is ‘small’, say less than 10°. See below, which shows incidentally the positive sense of these rotations.

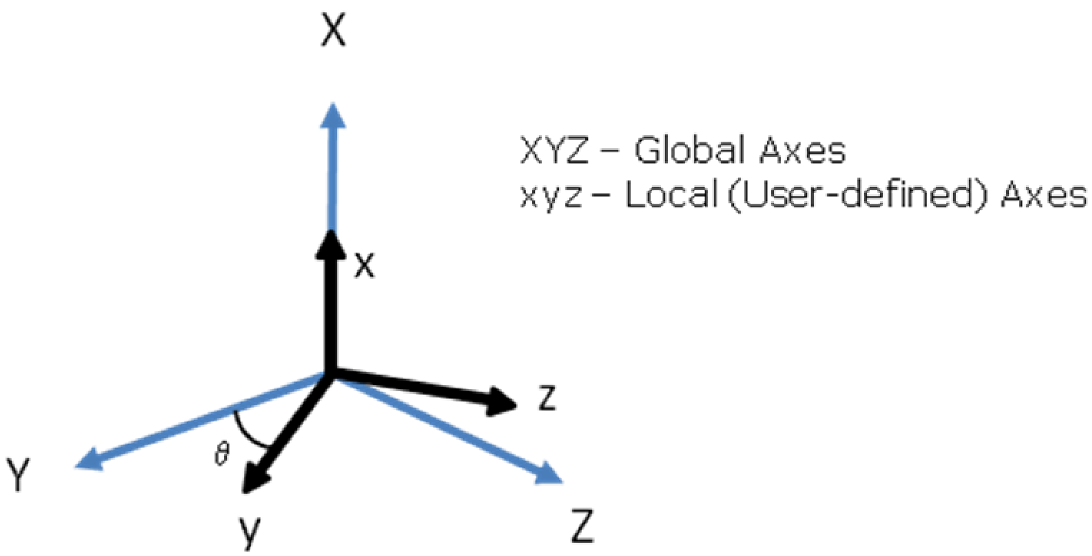


3D Analysis with Small Rotations

The only time the Flexcom user really needs to be aware of these rotation definitions and conventions is if you are applying non-zero rotational restraints or boundary conditions. It is important if specifying displacements in DOFs 4-6 to understand the significance of these degrees of freedom, as described above, and to be clear that they do not in the most general case represent individual rotations about the global axes. In particular, this means you should be wary of specifying only 1 or 2 rotation components (that is, 1 or 2 only of DOFs 4-6) where large rotations are involved, and you should only do so if you understand exactly the effect of what you are doing. Of course, if you are not specifying non-zero rotational boundary conditions, then this is not an issue.

User-Defined Axes

In addition to the global and other pre-defined axes in Flexcom, the program does provide the option for you to define an arbitrary or ‘local’ system of axes. For example, rotational hydrodynamic coefficients for a point buoy may be defined with respect to a local axis system. The local axes are defined by inputting, as components in the global axes, two of the three vectors defining the axes, namely the local x-axis and y-axis. These must be orthogonal, and provided they are, Flexcom works out the component of the local z-axis using the right-hand rule. For example, in the simplified specification below where the local axes simply represent a rotation of the global axes about global X, appropriate inputs for the local x-axis would be (1, 0, 0) while the local y-axis could be defined as (0, cos θ, sin θ).



User-Defined or Local Axes

1.9.1.4 Finite Element Formulation

INTRODUCTION

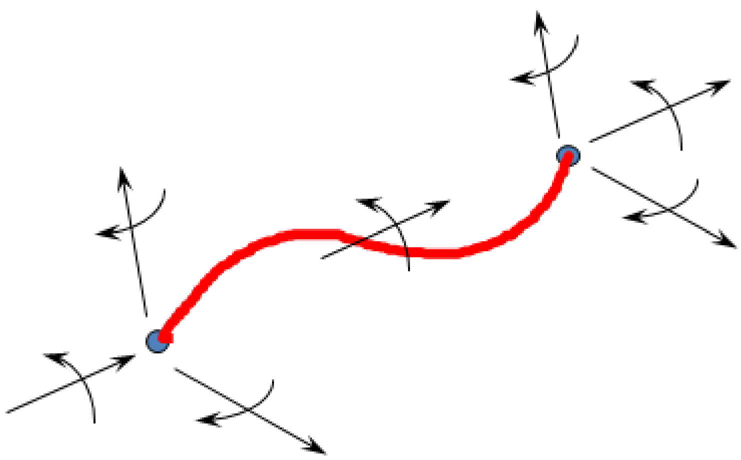
Flexcom’s advanced computational technique provides supreme confidence in the engineering design. The software uses an industry-proven finite element formulation, incorporating a hybrid beam-column element with fully coupled axial, bending and torque forces. Up to 10 integration points are used within each finite element to ensure a precise distribution of applied forces. Third order shape functions are used to predict solution variable (e.g. moment, curvature) variations within each element.

A truss element has also been added in more recent versions of Flexcom. This is essentially a simplified version of the traditional beam element (it does not solve for nodal rotations) which is better suited to structures with very low structural bending stiffness such as mooring chains.

HYBRID BEAM-COLUMN ELEMENT

In order for Flexcom to be capable of analysing flexible risers, flowlines and mooring lines, in addition to more traditional structures such rigid risers, TLP tethers and pipelines, a numerical solution scheme is required which caters for (i) bending stiffnesses which are much lower than both axial stiffness and torsional stiffness values, and (ii) arbitrarily large and nonlinear displacements and rotations in three dimensions.

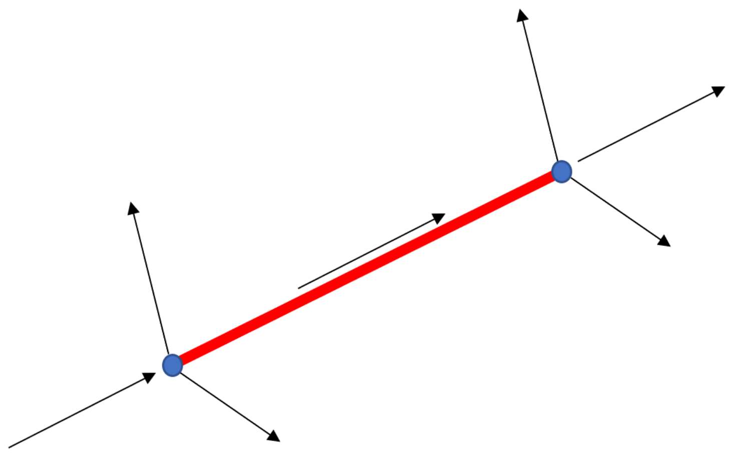
To accommodate the low or zero bending stiffness problem, early versions of Flexcom were based on a 2D hybrid beam-column element (McNamara et al., 1988). In this approach the axial force appeared as an explicit nodal solution variable, and was interpolated independently of the axial strain. The stress-strain compatibility relationship was applied outside of the virtual work statement by means of a Lagrangian constraint. This proved an accurate approach, and was subsequently extended to three dimensions (O'Brien & McNamara, 1988). Further developments resulted in the addition of an extra Lagrangian constraint on the torque degree of freedom, in order to make the scheme more robust and accurate when this variable plays a significant role in the solution (O'Brien. et al., 1991). This leads to a 14-DOF hybrid finite element with two end nodes, where the axial force and torque are added to the usual form of a three-dimensional beam element. The fundamental finite element equations for this element are derived in the following sections.



14-DOF Hybrid Finite Element

TRUSS ELEMENT

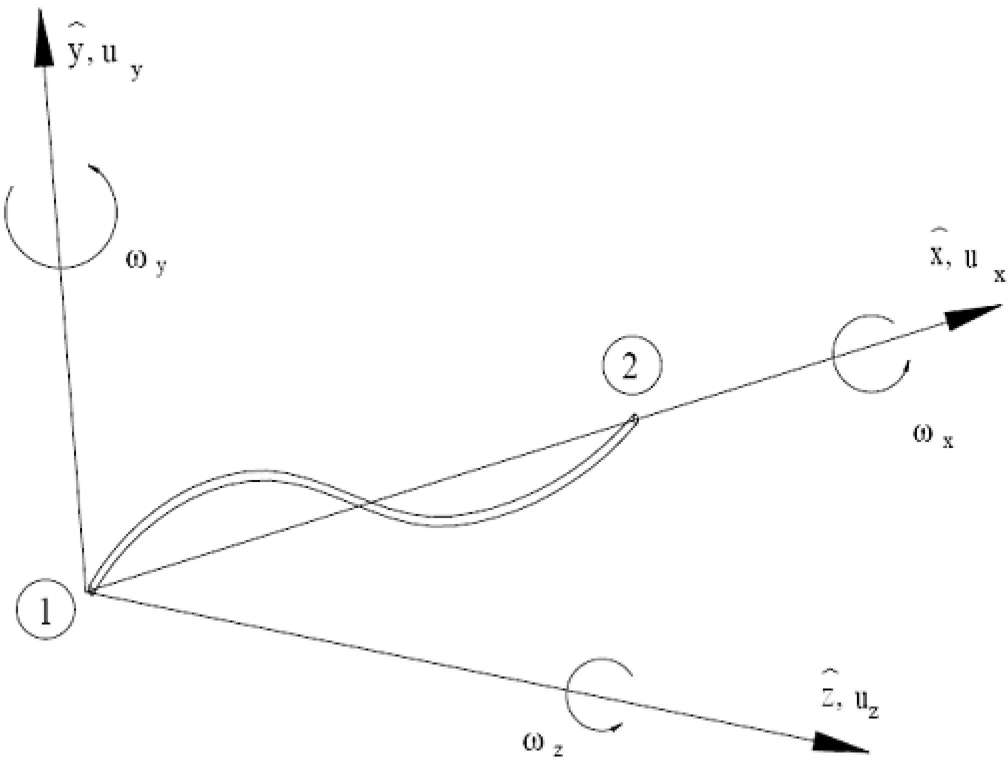
The truss element has 3 translational degrees of freedom at each node, and deforms only in the axial direction (it does not deform in bending or torsion). As it does not solve for nodal rotations, the connection at each node is essentially a pure hinge. The axial force penalty term is retained making the truss element a 7-DOF hybrid finite element with two end nodes. The truss element is designed specifically for modelling structures which have very low levels of structural bending stiffness (such as mooring chains) and is essentially a simplified version of the standard beam-column element employed by Flexcom. Refer to Truss Element for further information.



7-DOF Truss Element

CONVECTED AXIS SYSTEM

Flexcom uses detailed kinematics for finite rotations in three dimensions. This is based on the use of convected coordinate axes in developing the equations of motion, a technique also developed initially in 2D and subsequently extended to 3D (O'Brien and McNamara, 1988), (O'Brien. et al., 1988), (McNamara and O'Brien, 1986). Each element of the finite element discretisation has a convected axis system associated with it, which moves with the element as it displaces in space. The convected system is denoted in the following as the $\hat{x}\hat{y}\hat{z}$ system. (By contrast the global axes, which are common for all elements and are the system in which the equilibrium equations are assembled and solved, is denoted the XYZ system) The instantaneous orientation of the convected axes for an element is found as follows, with reference to the below. The convected $\hat{x}$ axis always joins the first local node of the element to the second node; the $\hat{y}$ and $\hat{z}$ axes are oriented in such a way that at the first node the local twist rotation of the deformed element relative to the convected axes is zero (O'Brien. et al., 1988).



Convected Axis System

The finite element development in the following is based on the assumption that strains are small and elastic under arbitrarily large three-dimensional displacements and rotations. The internal and external virtual work statements are written in the convected system; deformations along the element relative to this system are assumed to be moderate. Further details are provided in the following sections, and are also available in Chaudhuri. et al., (1987), O'Brien, and McNamara, (1989), Karve et al., (1988), O'Brien et al., (2002), O'Brien et al., (2003).

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Nomenclature explains the matrix algebra notation used extensively throughout this chapter.
- Internal Virtual Work Statement derives the internal incremental virtual work statement for the hybrid beam formulation.
- Finite Element Model describes the nodal solution vector.
- Strain Increments describes the strain vector, in terms of linear and nonlinear strain terms.
- External Virtual Work Statement presents the external virtual work for the beam element.
- Equations of Motions presents the finite element equations of motion, based on equating internal and external virtual work expressions.
- Finite Rotation Kinematics presents the Flexcom kinematics for finite rotations in three dimensions.
- Truss Element discusses the truss element in more detail.

RELEVANT KEYWORDS

- `*PRINT` is used to request additional printed output to the main output file. Specifically, the `OUTPUT=CONVECTED ELEMENT AXES` option is used to request the local convected axes for each element as a function of time.

Nomenclature

Matrix algebra is used extensively throughout the derivations which follow. An underlined variable such as $\underset{-}{K}$ represents a rectangular array or tensor, that is, an array comprised of m rows and n columns, with m>1 and n>1. A variable of the form $\underset{\sim}{F}$ represents a vector, that is, a matrix comprising m rows and 1 column only.

Matrices may be expressed in either global or local (convected) axes, so it is necessary to distinguish between them. In this manual a quantity with the ^ superscript, such as $\underset{\sim}{\hat{F}}$, represents a matrix or vector in local axes. A quantity without the ^ superscript, such as $\underset{\sim}{F}$, is a global matrix. In the case of a small number of matrices the question of local or global axes is immaterial, and these are written without the ^ superscript by default (an example is the matrix of finite element interpolation functions $\underset{-}{N}$).

The nomenclature may be summarised as follows:

- $\underset{-}{K}$ is a tensor in global axes
- $\underset{-}{\hat{K}}$ is a tensor in local axes
- $\underset{\sim}{F}$ is a vector in global axes
- $\underset{\sim}{\hat{F}}$ is a vector in local axes

Internal Virtual Work Statement

The generalised stress field $\underset{\sim}{\sigma}$ for a three dimensional beam, as illustrated in the Convected Axis System, is given by:

$$\underset{\sim}{\sigma} = \lfloor n \;\; m_x \;\; m_y \;\; m_z \rfloor^T \qquad (1)$$

where:

- n is axial force, and
- m_x, m_y, m_z are the moments about the three axes

Since it is intended to apply penalties to the axial force and torque moment quantities using Lagrange constraints, an additional vector $\underset{\sim p}{\sigma}$ is defined as:

$$\underset{\sim p}{\sigma} = \left\lfloor \tilde{n} \;\; \tilde{m}_x \; 0 \;\; 0 \right\rfloor^T \qquad (2)$$

where $\tilde{n}$ and $\tilde{m}_x$ are independent variables and are solved for directly in the set of finite element equations.

The generalised strains $\underset{\sim}{\varepsilon}$ conjugate to the stresses in the above Equation are written in terms of linear and nonlinear components as follows:

$$\underset{\sim}{\varepsilon} = \underset{\sim L}{\varepsilon} + \underset{\sim NL}{\varepsilon} \qquad (3)$$

Here:

$$\underset{\sim L}{\varepsilon} = \left\lfloor u_x^{'} \;\; \omega_x^{'} \;\; u_z^{''} \;\; u_y^{''} \right\rfloor^T \qquad (4)$$

and:

$$\underset{\sim NL}{\varepsilon} = \left\lfloor \frac{1}{2} \left\{ (u_y^{'})^2 + (u_z^{'})^2 \right\} 0 \;\; 0 \;\; 0 \right\rfloor^T \qquad (5)$$

A prime denotes differentiation with respect to the coordinate $\hat{x}$ along the local convected beam axis shown in the Convected Axis System. The nonlinear strains in the equation directly above indicate that moderate rotations with respect to the convected axes are included. They may be expressed more conveniently in the form:

$$\underset{\sim NL}{\varepsilon} = \frac{1}{2} \underset{\sim}{A} \underset{\sim}{\theta} \qquad (6)$$

where:

$$\underset{\sim}{A} = \begin{bmatrix} u_y' & u_z' \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad (7)$$

and:

$$\underset{\sim}{\theta} = \begin{bmatrix} u_y' \\ u_z' \end{bmatrix} \qquad (8)$$

Finally, the stress-strain relationship is written simply as:

$$\underset{\sim}{\sigma} = \underset{-}{D} \underset{\sim}{\varepsilon} \qquad (9)$$

where $\underset{-}{D}$ is an elastic constitutive matrix constructed as:

$$\underline{D} = \begin{bmatrix} EA & 0 & 0 & 0 \\ 0 & GJ & 0 & 0 \\ 0 & 0 & EI_{yy} & 0 \\ 0 & 0 & 0 & EI_{zz} \end{bmatrix} \qquad (10)$$

The internal incremental virtual work statement for this hybrid formulation is now expressed as:

$$\delta W_I = \int_0^L \delta \underset{\sim}{\varepsilon}^T \underset{-}{\Gamma} \underset{\sim}{\sigma} d\hat{x} + \int_0^L \delta \underset{\sim}{\varepsilon}^T (\underset{-}{I} - \underset{-}{\Gamma}) \underset{\sim p}{\sigma} d\hat{x}$$
$$+ \int_0^L \delta \underset{\sim p}{\sigma}^T (\underset{-}{I} - \underset{-}{\Gamma})(\underset{\sim}{\varepsilon} - \underset{-}{D}^{-1} \underset{\sim p}{\sigma}) d\hat{x} \qquad (11)$$

where:

- δW_I is an increment of internal virtual work
- L is the length of the beam element
- $\underset{\sim}{I}$ is the identity matrix

and:

$$\underset{\sim}{\Gamma} = \begin{bmatrix} \gamma_1 & 0 & 0 & 0 \\ 0 & \gamma_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (12)$$

The matrix $\underset{\sim}{\Gamma}$ is a weighting matrix, and the values of γ_1 and γ_2 depend on the flexibilities of particular elements. The third integral in the above equation is the Lagrangian constraint imposing the penalty terms on the axial force and torque variables.

Finite Element Model

The local displacement vector relative to the convected axes $\underset{\sim}{\overset{\wedge def}{u}}$ for a point on the pipe element shown in the Convected Axis System is defined as:

$$\underset{\sim}{\overset{\wedge def}{u}} = \lfloor u_x \; u_y \; u_z \rfloor^T \qquad (1)$$

Similarly, the local rotation vector $\underset{\sim}{\overset{\wedge def}{\omega}}$ is given as:

$$\underset{\sim}{\overset{\wedge def}{\omega}} = \lfloor \omega_x \; \omega_y \; \omega_z \rfloor^T \qquad (2)$$

Since the rotations are assumed small relative to the convected axes, the above equation may be written as:

$$\hat{\underset{\sim}{\omega}}^{def} = \left\lfloor \omega_x \; -u'_z \; u'_y \right\rfloor^T \qquad (3)$$

The total local nodal solution vector $\hat{\underset{\sim}{d}}^{def}$ for the two-noded pipe element may be written as:

$$\hat{\underset{\sim}{d}}^{def} = [u_{x1} \; u_{y1} \; u_{z1} \; \omega_{x1} \; -u'_{z1} \; u'_{y1} \; u_{x2} \; u_{y2} \; u_{z2}$$

$$\omega_{x2} \; -u'_{z2} \; u'_{y2} \; \tilde{n} \; \tilde{m}_x]^T \qquad (4)$$

The standard twelve degrees of freedom are augmented in this case by the independent axial force and torque variables.

The vector of the solution variables at any point on the pipe element $\hat{\underset{\sim}{u}}$ is found by interpolating the nodal vector as follows:

$$\hat{\underset{\sim}{u}} = \underset{-}{N} \, \hat{\underset{\sim}{d}}^{def} \qquad (5)$$

where linear interpolation is used for the axial components u_x and ω_x ; cubic functions for the transverse displacements u_y and u_z ; and the generalised forces $\tilde{n}$ and $\tilde{m}_x$ are assumed constant over the element. The rotations u'_y and $\tilde{m}_x$ may also be expressed in terms of the nodal vector where:

$$\underset{\sim}{\theta} = \underset{-}{G} \, \hat{\underset{\sim}{d}}^{def} \qquad (6)$$

Clearly, $\underset{-}{G}$ is obtained directly from the functions given by $\underset{-}{N}$. Similarly, the penalty terms may be written as:

$$\underset{\sim}{\sigma}_p = \underline{Q}\,\underset{\sim}{\hat{d}}^{def} \qquad (7)$$

Strain Increments

Both total and incremental forms of the strain vector are required for substitution in Internal Virtual Work Statement (Eq.10). The linear strains are expressed in the usual way in terms of the nodal variables by:

$$\underset{\sim}{\varepsilon}_L = \underline{B}_L\,\underset{\sim}{\hat{d}}^{def} \qquad (2)$$

where $\underline{B}_L$ is the standard “beta” matrix relating strains to displacements. The incremental form is simply:

$$\delta\underset{\sim}{\varepsilon}_L = \underline{B}_L\,\delta\underset{\sim}{\hat{d}}^{def} \qquad (3)$$

Similarly for the nonlinear strain, the total strain is given by Internal Virtual Work Statement (Eq.6), and the incremental value is derived as:

$$\delta\underset{\sim}{\varepsilon}_{NL} = \frac{1}{2}\delta\underline{A}\underset{\sim}{\theta} + \frac{1}{2}\underline{A}\delta\underset{\sim}{\theta} = \underline{A}\delta\underset{\sim}{\theta} \qquad (4)$$

which gives:

$$\delta\underset{\sim}{\varepsilon}_{NL} = \underline{B}_{NL}\,\delta\underset{\sim}{\hat{d}}^{def} \qquad (5)$$

where $\underline{B}_{NL} = \underline{A}\,\underline{G}$. Additionally, the total nonlinear term can be written in the form:

$$\underset{\sim}{\varepsilon}_{NL} = \frac{1}{2}\underline{B}_{NL}\,\underset{\sim}{\hat{d}}^{def} \qquad (6)$$

External Virtual Work Statement

The external virtual work statement for an accelerating pipe element is given by:

$$\delta W_E = \int_0^L \delta \hat{\underset{\sim}{u}}^T \hat{\underset{\sim}{f}} \, d\hat{x} - \int_0^L m \, \delta \hat{\underset{\sim}{u}}^T \hat{\ddot{\underset{\sim}{u}}} \, d\hat{x} - \int_0^L p \, \delta \hat{\psi} \hat{\ddot{\psi}} \, d\hat{x} \qquad (1)$$

where:

- δW_E is the increment of external virtual work
- $\hat{\underset{\sim}{f}}$ is the distributed load vector in convected axes
- m is the mass per unit length of element
- $\hat{\ddot{\underset{\sim}{u}}}$ is the translational acceleration vector in convected axes, $\left\lfloor \hat{\ddot{u}}_x \; \hat{\ddot{u}}_y \; \hat{\ddot{u}}_z \right\rfloor^T$
- p is the polar moment of inertia per unit length about element axis
- $\hat{\ddot{\psi}}$ is the angular acceleration about convected $\hat{x}$ axis

Substituting for displacements and accelerations in terms of nodal quantities and interpolation functions, in standard finite element fashion, gives the finite element form of δW_E, namely:

$$\delta W_E = \delta \hat{\underset{\sim}{d}}^T \left(\hat{\underset{\sim}{F}} - \hat{\underset{-}{M}} \hat{\ddot{\underset{\sim}{d}}} \right) \qquad (2)$$

where:

- $\hat{\underset{\sim}{F}}$ is the vector of equivalent nodal loads in convected axes
- $\hat{\underset{-}{M}}$ is the element mass matrix in convected axes

- $\underset{\sim}{\hat{\ddot{d}}}$ is the vector of nodal accelerations in convected axes

Equations of Motion

This section presents the finite element equations of motion, based on equating internal and external virtual work expressions.

By substituting for the stress and strain expressions in the virtual work statement given by Internal Virtual Work Statement (Eq.11), and equating internal and external work increments given by Internal Virtual Work Statement (Eq.11) and External Virtual Work Statement (Eq.2), the local form of the equations of motion are derived as:

$$\underset{-}{\hat{M}}\underset{\sim}{\hat{\ddot{d}}} + \underset{-}{\hat{K}}\underset{\sim}{\hat{d}}^{def} = \underset{\sim}{\hat{F}} \qquad (1)$$

The local hybrid stiffness matrix is decomposed to:

$$\underset{-}{\hat{K}} = \hat{K}_{-L} + \hat{K}_{-D} + \hat{K}_{-G} \qquad (2)$$

where $\hat{K}_{-L}$ is the standard beam-column linear stiffness matrix, and $\hat{K}_{-D}$ and $\hat{K}_{-G}$ are given by:

$$\hat{K}_{-D} = \frac{1}{2}\left[\int_{0}^{L} \underset{-}{B}^{T}\underset{-}{\Gamma}\underset{-}{D}\underset{-}{B}_{NL}\, d\hat{x} + \int_{0}^{L} \underset{-}{Q}^{T}(\underset{-}{I}-\underset{-}{\Gamma})\underset{-}{B}_{NL}\, d\hat{x}\right] \qquad (3)$$

and:

$$\hat{K}_{-G} = \int_{0}^{L} \underset{-}{G}^{T}\underset{-}{H}\underset{-}{G}\, d\hat{x} \qquad (4)$$

where:

$$\underline{H} = \begin{bmatrix} n & 0 \\ 0 & n \end{bmatrix} \qquad (5)$$

$\hat{\underline{K}}_D$ and $\hat{\underline{K}}_G$ are respectively the initial displacement matrix and the geometric stiffness matrix. Equation (1) above can now be transformed from the local convected to the global axes resulting in the equation:

$$\underline{M}\,\ddot{\underset{\sim}{d}} + \underline{K}\,\underset{\sim}{d}^{def} = \underset{\sim}{F} \qquad (6)$$

where:

$$\underline{M} = \underline{R}^T \,\hat{\underline{M}}\, \underline{R} \qquad (7)$$

$$\underline{K} = \underline{R}^T \,\hat{\underline{K}}\, \underline{R} \qquad (8)$$

$$\underset{\sim}{F} = \underline{R}^T \,\hat{\underset{\sim}{F}},\quad \ddot{\underset{\sim}{d}} = \underline{R}^T \,\hat{\ddot{\underset{\sim}{d}}},\quad \underset{\sim}{d}^{def} = \underline{R}^T \,\hat{\underset{\sim}{d}}^{def} \qquad (9)$$

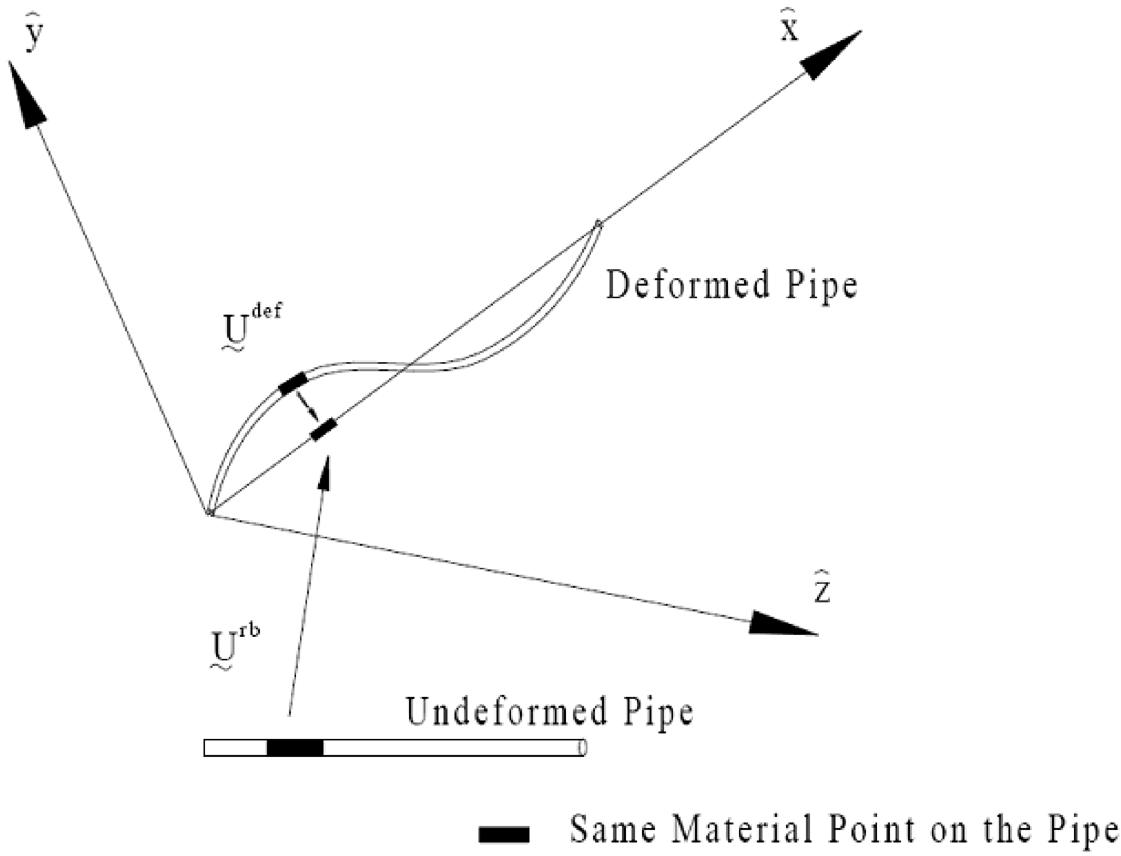
and $\underline{R}$ is the transformation matrix from local to global axes.

Finite Rotation Kinematics

The coefficient vector associated with the stiffness matrix in the Equations of Motion (Eq.6) is written in terms of the deformations with respect to the convected axes. In order to use this equation to solve the complete system, Equations of Motion (Eq.6) must be expressed in terms of the total nodal solution vector $\underset{\sim}{d}$. With reference to the figure below, the translation deformation components of a material point on a pipeline may be written as:

$$\underset{\sim}{u} = \underset{\sim}{u}^{def} + \underset{\sim}{u}^{rb} \qquad (1)$$

where $\underset{\sim}{u}^{rb}$ is the rigid body translation vector of a material point on the convected axes.



Translation of Material Point

However, the same simple decomposition cannot be applied to the finite rotational components since they do not behave as vectors.

The finite rotation of a material point on the pipeline may be represented by the line segment $\underset{\sim}{\omega}$, whose magnitude is the amount of the rotation and whose direction is the axis about which the rotation occurs and is in the sense defined by the right-hand rule. The line segment $\underset{\sim}{\omega}$ is not a true vector since addition is not commutative. A finite rotation can also be represented by an orthogonal matrix $\underset{-}{T}$ whose components are uniquely determined by the components of $\underset{\sim}{\omega}$. We can incorporate the rigid body rotation of the convected axes from the undeformed pipe orientation by the equation:

$$\underset{-}{T} = \underset{-}{T}^{def}\,\underset{-}{T}^{rb} \qquad (2)$$

where $\underset{-}{T}^{rb}$ corresponds to the rigid body rotation $\underset{\sim}{\omega}^{rb}$ of the convected axes, and $\underset{-}{T}^{def}$ corresponds to the rotation $\underset{\sim}{\omega}^{def}$ of the material point from the convected axes.

Now consider two material points on the beam a and b at a distance Δs apart, and write that:

$$\underset{-}{T}_b = \Delta\underset{-}{T}\,\underset{-}{T}_a \qquad (3)$$

where $\underset{-}{T}_b$ corresponds to a rotation $\underset{\sim}{\omega}_b$, $\underset{-}{T}_a$ corresponds to a rotation $\underset{\sim}{\omega}_a$, and $\Delta\underset{-}{T}$ corresponds to an incremental rotation $\Delta\underset{\sim}{\omega}$. Note that $\Delta\underset{\sim}{\omega}$ is not the algebraic difference between $\underset{\sim}{\omega}_a$ and $\underset{\sim}{\omega}_b$, but must be found from Equation (3). A rate of rotation vector $\underset{\sim}{\Omega}$ may now be defined as:

$$\underset{\sim}{\Omega} = \lim_{\Delta s \to 0} \frac{\Delta\underset{\sim}{\omega}}{\Delta s} \qquad (4)$$

and is a true vector quantity in the limit of small rotations.

An examination of Equations (3) and (4) reveals that the definition of the rate of rotation is independent of the reference configuration from which the rotation is measured. Therefore it is possible to define two vector quantities $\underset{\sim}{W}$ and $\underset{\sim}{W}^{def}$ such that:

$$\frac{\partial \underset{\sim}{W}}{\partial s} = \frac{\partial \underset{\sim}{W}^{def}}{\partial s} = \underset{\sim}{\Omega} \qquad (5)$$

Additionally, rotations $\underset{\sim}{\omega}^{def}$ relative to the convected axes are small by definition and so behave as true vectors; therefore it is possible to define $\underset{\sim}{W}^{def}$ as:

$$\underset{\sim}{W}^{def} = \underset{\sim}{\omega}^{def} \qquad (6)$$

Integrating Equation (5) and incorporating Equation (6) yields

$$\underset{\sim}{W} = \underset{\sim}{\omega}^{def} + \underset{\sim}{W}^{rb} \qquad (7)$$

where $\underset{\sim}{W}^{rb}$ is a constant vector determined from the integration of Equation (5) and associated with the rigid body rotation of the convected axes.

Note that $\underset{\sim}{W}$ and $\underset{\sim}{W}^{rb}$ are true vectors, but are not the same as the actual rotations $\underset{\sim}{\omega}$ and $\underset{\sim}{\omega}^{rb}$, respectively. However, Equation (7) shows that by using the so-called quasi-rotation vector $\underset{\sim}{W}$ it is possible to isolate a rigid body term $\underset{\sim}{W}^{rb}$ associated with the rigid body rotation of the convected axes. The decomposition represented by Equation (7) is also of the same vector form as Equation (1).

If the quantities in Equation (7) are re-interpreted as nodal rotational components, the following decomposition is possible for the total nodal vector:

$$\underset{\sim}{d} = \underset{\sim}{d}^{def} + \underset{\sim}{d}^{rb} \qquad (8)$$

For completeness, the terms in $\underset{\sim}{d}$ are listed as:

$$\underset{\sim}{d} = [d_{X1}\ d_{Y1}\ d_{Z1}\ W_{X1}\ W_{Y1}\ W_{Z1}\ d_{X2}\ d_{Y2}\ d_{Z2}$$
$$W_{X2}\ W_{Y2}\ W_{Z2}\ \tilde{n}\ \tilde{m}_x]^T \qquad (9)$$

where:

- d_{Xi}, d_{Yi}, d_{Zi} are the components in global axes of displacement of Node i from undeformed position
- W_{Xi}, W_{Yi}, W_{Zi} are the components in global axes of quasi-rotation vector at Node i

Substituting Equation (8) into Equation (5), the final form of the finite element equations of motion is obtained as:

$$\underline{M}\,\underset{\sim}{\ddot{d}} + \underline{K}\,\underset{\sim}{d} = \underset{\sim}{F} + \underline{K}\,\underset{\sim}{d}^{rb} \qquad (10)$$

Stiffness and mass proportional damping terms may also be included in the equations of motion, in which case Equation (10) is modified to become:

$$\underline{M}\,\underset{\sim}{\ddot{d}} + (\lambda\,\underline{K} + \mu\,\underline{M})\underset{\sim}{\dot{d}} + \underline{K}\,\underset{\sim}{d} = \underset{\sim}{F} + \underline{K}\,\underset{\sim}{d}^{rb} \qquad (11)$$

where λ and μ are damping coefficients and $\underset{\sim}{\dot{d}}$ is the vector of nodal velocities. The equation for a pure static analysis is given by:

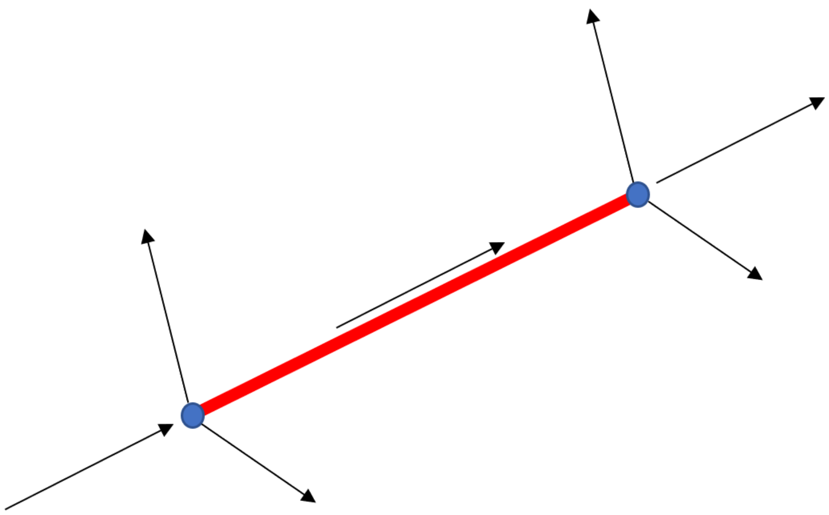
$$\underline{K}\,\underset{\sim}{d} = \underset{\sim}{F} + \underline{K}\,\underset{\sim}{d}^{rb} \qquad (12)$$

where the load vector $\underset{\sim}{F}$ in this case contains static loading only.

Truss Element

OVERVIEW

The truss element has 3 translational degrees of freedom at each node, and deforms only in the axial direction (it does not deform in bending or torsion). As it does not solve for nodal rotations, the connection at each node is essentially a pure hinge. The axial force penalty term is retained making the truss element a 7-DOF hybrid finite element with two end nodes. The truss element is designed specifically for modelling structures which have very low levels of structural bending stiffness (such as mooring chains) and is essentially a simplified version of the standard beam-column element employed by Flexcom.



7-DOF Truss Element

Modelling of flexible lines can be quite challenging, especially in severe dynamic environments, where there is a tendency for the lines to go slack intermittently. Due to the low structural bending stiffness, the standard beam column elements employed by Flexcom are not well suited to this type of scenario. When the beam element attempts to solve for nodal rotations, the lack of bending resistance can lead to a solution indeterminacy. In such circumstances, truss elements offer both increased solution robustness and increased computational efficiency. Faster computation times are due to the reduced number of degrees of freedom in the model, and the ability to run at larger time-steps (particularly when lines go slack).

The general finite element formulation for truss elements is similar to that of beam elements, which has been discussed in some detail in the preceding articles. For example, each truss element has its own Convected Axis System and the Equations of Motion are applied in a consistent manner regardless of whether a model consists of beam elements, truss elements or a combination of both. Due to the fewer active degrees of freedom, there are some differences in the formation of the constitutive matrices for truss and beam elements. The following sections present the truss element formulation in more detail.

Note that Truss Elements supersede the Mooring Line modelling feature which is now outdated. In this older approach, the mooring line is modelled using a combination of standard beam elements interspersed with hinge elements. While this hybrid approach offers increased solution robustness over traditional beam models, it represents an inefficient solution procedure with the model containing many more elements than an equivalent truss element model.

FINITE ELEMENT FORMULATION

Shape Functions

The local deformation vector at any point along the element, $u_{def}(x)$, is related to the nodal deformation vector of the element, d_{nd}, by the matrix of shape functions, [N(x)].

$$\vec{u}_{def}(x) = \left[N(x)\right]\vec{d}_{nd} \qquad (1)$$

The truss element undergoes axial deformation only so the element shape is a straight line between the two end nodes. Therefore linear shape functions are used to interpolate for deflections at points intermediate to the end-nodes.

$$\left[N(x)\right] = \begin{bmatrix} N_1(x) & 0 & 0 & 0 & 0 & 0 & N_2(x) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_1(x) & 0 & 0 & 0 & 0 & 0 & N_2(x) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & N_1(x) & 0 & 0 & 0 & 0 & 0 & N_2(x) & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (2)$$

where:

$$N_1(\mathrm{x}) = 1 - \frac{x}{L} \qquad (3)$$

and:

$$N_2(\mathrm{x}) = \frac{x}{L} \qquad (4)$$

are the linear interpolation (shape) functions for the truss element and L is the element length. Note that these differ from the shape functions that are used for a beam-column element, in particular the truss element uses linear functions to interpolate for the lateral deflections, v(x) and w(x), whereas the beam-column element uses cubic polynomial (Hermitian) interpolation functions.

Structural (Linear) Stiffness Matrix

The linear stiffness matrix [K_L] is defined as:

$$[K_L] = \int_0^L [B(x)]^T [D][B(\mathrm{x})]dx \qquad (5)$$

Here [D] is the constitutive matrix and [B(x)] is the matrix of functions relating the linear strain to the nodal deformation vector. For a truss element with only axial deformation, the constitutive matrix contains only a single element, which is the axial stiffness EA. Note that in the current version of Flexcom, EA must have a constant (linear) value. Non-linear axial stiffness will be supported in a future version.

The linear stiffness matrix [K_L], including the axial penalty terms, may be derived as:

$$[K_L]=\begin{bmatrix} \gamma\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & -\gamma\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-(1-\gamma)}{\rho} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\gamma\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & \gamma\frac{EA}{L} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{(1-\gamma)}{\rho} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{-(1-\gamma)}{\rho} & 0 & 0 & 0 & 0 & 0 & \frac{(1-\gamma)}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-(1-\gamma)\,\mathrm{L}}{\rho^2 EA} \end{bmatrix}$$

(6)

For beam-column elements, γ is related to the ratio of the bending to axial stiffness (γ = EI / $EA.L^2$). However as truss elements have no bending stiffness a fixed value of $\gamma = 1 \times 10^{-5}$ is used (the same value is used for spring elements which have essentially the same form of linear stiffness matrix). The parameter ρ is a flexibility coefficient with a default value of 1×10^{-3}.

Geometric Stiffness Matrix

The geometric stiffness matrix $[K_G]$ is defined as:

$$[K_G]=\int_0^L [G(x)]^T [A_\sigma][G(\mathrm{x})]dx$$

(7)

Here $[A_\sigma]$ is a 2x2 matrix based on effective tension:

$$A_\sigma=\begin{bmatrix} T_{eff} & 0 \\ 0 & T_{eff} \end{bmatrix}$$

(8)

and [G(x)] is the matrix that relates a vector of slopes to the nodal deformation vector. [G(x)] is found by differentiating the second and third rows of the shape function matrix [N(x)] with respect to x and is given by:

$$G(\mathrm{x}) = \begin{bmatrix} 0 & N_1^{'}(x) & 0 & 0 & 0 & 0 & 0 & N_2^{'}(x) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & N_1^{'}(x) & 0 & 0 & 0 & 0 & 0 & N_2^{'}(x) & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(9)

The geometric stiffness matrix $[K_G]$ may be derived as:

$$[K_G] = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{T_{eff}}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{T_{eff}}{L} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{T_{eff}}{L} & 0 & 0 & 0 & 0 & 0 & -\frac{T_{eff}}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{T_{eff}}{L} & 0 & 0 & 0 & 0 & 0 & \frac{T_{eff}}{L} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{T_{eff}}{L} & 0 & 0 & 0 & 0 & 0 & \frac{T_{eff}}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(10)

Note that the above exact form of the geometric stiffness matrix assumes that the effective tension T_{eff} is constant along the element. This is not the case in general, so the geometric stiffness matrix is found using numerical integration.

It is also important to note that the contribution from the geometric stiffness matrix is included only if T_{eff} is positive, as flexible lines (modelled using truss elements) cannot carry compressive loads. If T_{eff} is negative at any specific integration point, the geometric stiffness contribution is assumed to be zero. Intuitively one would expect that a truss element should never experience effective compression, but it is mathematically possible in a numerical model. Refer to Compression in Truss Elements for further discussion on this topic.

The first solution iteration of an initial static analysis is a special case. Geometric stiffness relies on the presence of a positive effective tension, but no effective tension results are formally computed until after the first solution iteration has been completed.

- If you are modelling a catenary section, using either the `*CABLE` or `*LINES` keywords, the Cable Pre-Static Step is invoked. Using cable catenary equations, this step provides the finite element solver with approximate nodal coordinates, and it also provides an estimation of the effective tension in the line. This tension value ensures that the geometric stiffness contribution is included in the first solution iteration.

- If you are modelling a straight section, using either `*NODE`/`*ELEMENT` or `*LINES` keywords, Flexcom automatically assumes a nominal effective tension value of 1N (or 1lbf) to facilitate the inclusion of a token geometric stiffness contribution. This is generally sufficient to prevent a solution indeterminacy in the initial static analysis. If not however, then you could (i) add some node springs of very low stiffness or (ii) attempt to solve the model quasi-statically rather than statically, perhaps including a small level of damping to aid numerical stability.

Initial Displacement Matrix

The initial displacement matrix, $[K_D]$, as presented in Equations of Motion (Eq.3) for beam elements, is not constructed for a truss element. This aspect of the finite element formulation is only meaningful when bending and torsional degrees of freedom are active and so is reserved for beam elements only. It is not possible to explicitly specify a stress-free orientation for a truss element and so both (i) the specification of V and W vectors under the `*LINES` or `*ELEMENT` keywords and (ii) the discussion around Undeformed Versus Initial Positions are not relevant for truss elements.

Mass Matrix

The mass matrix [M] is defined as:

$$[M]=\int_0^L [N(x)]^T [m][N(x)]dx \qquad (11)$$

where the density matrix [m] is defined as:

$$[m]=\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} \qquad (12)$$

and m is the mass per unit length of the element (which includes contributions from internal fluid and hydrodynamic added mass, if appropriate).

Hence the mass matrix [M] may be derived as:

$$[M]=\begin{bmatrix}
\frac{mL}{3} & 0 & 0 & 0 & 0 & 0 & \frac{mL}{6} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{mL}{3} & 0 & 0 & 0 & 0 & 0 & \frac{mL}{6} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{mL}{3} & 0 & 0 & 0 & 0 & 0 & \frac{mL}{6} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{mL}{6} & 0 & 0 & 0 & 0 & 0 & \frac{mL}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{mL}{6} & 0 & 0 & 0 & 0 & 0 & \frac{mL}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{mL}{6} & 0 & 0 & 0 & 0 & 0 & \frac{mL}{3} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix} \qquad (13)$$

Note that the above exact form of the mass matrix assumes that the mass (including contributions from internal fluid and hydrodynamic added mass) is constant along the element. This is not the case in general (for example, in the case of changing internal fluid properties or surface-piercing elements), so the mass matrix is found using numerical integration.

Damping Matrix

The Damping matrix [C] for a truss element is formed as a linear combination of the stiffness and mass matrices in the same way as for beam-column elements. Specifically:

$$[C] = \lambda[\underline{K}] + \mu[\underline{M}] \qquad (14)$$

The effects of Deformation Mode Damping for a truss element are limited to the axial degree of freedom for the structural stiffness contribution and the bending degrees of freedom for the geometric stiffness contribution. Hence:

$$[\underline{C}] = \lambda_{Axial}\left[\underline{K}^{Axial}_{Lin}\right] + \lambda_{Bending}\left[\underline{K}^{Bending}_{Geom}\right] + \mu[\underline{M}] \qquad (15)$$

Solution Parameters

The table below compares the range of solution parameters available for beam and truss elements.

A limited range of output variables is available for a truss element. In theoretical terms, the generalised stress field presented in Internal Virtual Work Statement (Eq.1) is simplified as:

$$\underset{\sim}{\sigma} = \begin{bmatrix} n & 0 & 0 & 0 \end{bmatrix}^T \qquad (16)$$

where *n* is axial force (moments about the local axes are all zero). So the truss element is incapable of presenting any data relating to shear force, bending moment, bending stress or torque. Likewise strain terms such as bending strain, curvature and bend radius are unavailable. For ease of post-processing and compatibility of databases, you may request any of these parameters and the program will simply return zero values rather than issuing an error message.

Note however that although the truss element does not solve for rotational degrees of freedom, results for nodal rotations in DOFs 4, 5 & 6 are available for convenience. These are derived from the element orientations in time and space, which in turn are governed by the translational degrees of freedom of the element end nodes.

Category	Parameter	Beam Element	Truss Element
Motion	DOF1-3	✓	✓
	DOF4-6	✓	✓
Force	Axial force	✓	✓
	Effective tension	✓	✓
	Shear force	✓	✗
Moment	Torque	✓	✗
	Bending moment	✓	✗
Stress	Axial stress	✓	✓
	Bending stress	✓	✗
	Von Mises stress	✓	✓
	Hoop stress	✓	✓
Strain	Axial strain	✓	✓
	Curvature	✓	✗
	Bending strain	✓	✗
	Bending radius	✓	✗
General	Temperature	✓	✓
	Pressure	✓	✓

RELEVANT KEYWORDS

- `*GEOMETRIC SETS` is used to assign structural properties, including those of truss elements.

If you would like to see an example of truss elements used in practice, refer to either E01 - CALM Buoy - Simple or L04 - UMaine VolturnUS-S IEA15MW.

1.9.2 Model Building

This section contains information on the following topics.

- Geometry discusses the various program options for defining model geometry, including the fundamental building blocks of Nodes and Elements, and the powerful Lines modelling feature. Undeformed Versus Initial Positions discusses some of the more advanced program facilities, specifically with regard to specifying an initial deformed structure configuration.
- Geometric Properties discusses the different formats for specifying structure geometric properties, namely the Flexible Riser, Rigid Riser and Mooring Line formats.
- Hydrodynamic Properties discusses the various Flexcom options for defining hydrodynamic properties, including sections on Constant Coefficients and Reynolds Number of Dependent Coefficients, plus some more Advanced Topics.
- Special Element Types discusses a range of practical modelling features provided by Flexcom. Specialised elements include Springs, Hinges and Flex joints, Tapered Stress Joints, Bend Stiffeners, Dampers, Winches, Point Masses and Point Buoys and Drag Chains.
- Pipe-in-Pipe discusses Flexcom's pipe-in-pipe modelling feature.
- Enviroment describes the ambient ocean environment, including sections on Environmental Parameters, Buoyancy Forces and Hydrodynamic Loading.
- Seabed Interaction presents a range of options for modelling seabed interaction. Rigid Seabeds may be flat, sloping, or have an arbitrary 2D piecewise linear bathymetry. Elastic Seabed geometries may be flat, sloping, or have arbitrary bathymetry specified in 2D or 3D.

* Vessels and Vessel Motions presents the various program options for modelling vessels and associated vessel motions. Vessel motions include Vessel Offsets, Low Frequency Drift Motions and High Frequency RAO Motions.

* Contact Surfaces are suitable for modelling contact between a line and a vessel, for example, contact between a riser and a spar hull.

* Line Clashing is used to model intermittent contact between adjacent lines.

1.9.2.1 Geometry

Finite element geometry in Flexcom is defined in terms of basic components or building blocks, namely nodes, elements, and lines. Finite element analysts will, of course, be familiar with the concept of nodes and elements. The use of cables in Flexcom is related to the program's cable pre-static step, which is described in detail later. Lines are a relatively recent addition to Flexcom, and essentially provide an automatic mesh creation facility whereby nodes, elements and cables can be generated quite easily. Flexcom also provides a number of options for defining auxiliary bodies, used to produce a visual representation of an object (such as a floating vessel).

The various program options for defining model geometry are discussed in the following sections:

* Nodes and Elements summarises the explicit specification of nodal coordinates and finite element connectivity.

* Cables describes the Flexcom cable pre-static step.

* Equivalent Nodes describes a facility for identifying coincident nodes as being equivalent.

* Node and Element Labels introduces the concept of node and element labelling.

- Lines describes the use of lines for model building. Lines provide an automatic mesh creation facility to greatly expedite the model creation process. Using lines is a fundamentally different approach to working directly with nodes, elements and cables, although the information is ultimately handled in the same fashion internally. Since lines provide automatic mesh generation, you do not to concern yourself with explicit node and element numbering. Indeed the availability of lines makes nodes, elements and cables redundant to some degree, but they are retained for complete generality, and also to maintain downward compatibility with previous program versions.

- Auxiliary Bodies describes a facility for including components in a model for visual or illustrative purposes only, typically a floating vessel. Auxiliary bodies do not form part of the finite element model of the structure.

- Undeformed Versus Initial Positions discusses some of the more advanced program facilities, specifically with regard to specifying an initial deformed structure configuration.

Nodes

OVERVIEW

Nodal locations may be specified in any of the following ways in Flexcom:

i. Directly in terms of global coordinates (using the `*NODE` keyword)

ii. Generated along a straight line between existing nodes (using the `*NODE` keyword)

iii. Directly in terms of distance along a cable (using the `*NODE,CURVILINEAR` keyword)

iv. Generated along a cable (using the `*NODE,CURVILINEAR` keyword)

v. Automatically created at the start or end point of a line (using the `*LINES` keyword)

vi. Automatically created during line meshing (using the `*LINES` keyword)

vii. Automatically positioned at a designated location in terms of distance along a line (using the `*LINE LOCATIONS` keyword)

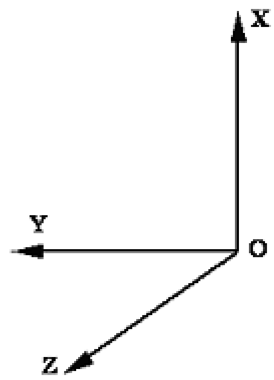
Methods (iii) and (iv) relate to the use of cables, and are discussed later in Cables. The use of cables has to a large extent been superseded by the Lines capability, so you can effectively disregard these options as being obsolete. Additionally, generation of straight lines is also possible with lines, so you can ignore method (ii) also.

Methods (v) to (vii) relate to the line generation facility, and are discussed later in Lines.

Hence, method (i) assumes the primary focus of this article.

EXPLICITLY DEFINED NODES

The specification of nodal locations in terms of global coordinates is self-explanatory. For this option, you define the node directly in terms of a node number and global coordinates. The Flexcom global coordinate system is shown below. Note that the global X axis is vertical in Flexcom. Explicitly defined nodes are useful when modelling structures which do not constitute long slender bodies such as risers and mooring lines. For example, you may wish to model a Floating Body using an arbitrary assemblage of rigid massless beam elements, in which case Nodes and Elements would typically represent the quickest approach.



Global Cartesian Coordinate System

AUTOMATICALLY GENERATED NODES

You can also specify nodes as being equally spaced along a straight line joining two nodes. This corresponds to method (ii) above. For this option, you specify the numbers of the end nodes. These end nodes must themselves be defined directly, or in a separate generation. You can also optionally specify the increment to be used in assigning node numbers to generated nodes, or simply let the program use the default value of 1.

When nodes are automatically generated from Lines, you do not concern yourself with explicit node and element numbering. The meshing process is similar to what you could achieve manually, but without all the unnecessary hassle.

Note that Flexcom automatically performs bandwidth optimisation to reduce the bandwidth of the system matrices if possible. This means that any arbitrary node numbering scheme may be used without adversely affecting the speed of execution of a particular Flexcom analysis.

Elements

THEORY

Flexcom has two options for defining the finite element model connectivity:

- Direct specification
- Element generation

These options are largely self-explanatory.

Direct specification involves specifying an element number and the two Nodes on the element. For the element generation option, you specify a master element (which must be directly specified) and instruct the program to generate a number of similar elements by copying the master element node numbering pattern. As an example, if Element 1 is defined as connecting Nodes 4 and 6, and this is then specified as a master element, then the first generated element connects Nodes 6 and 8, the second Nodes 8 and 10, and so on up to the specified Number of Elements.

Note that (i) this Number of Elements includes the master element, and (ii) you may optionally specify the increment to be used in assigning element numbers to the generated elements (the default value is 1). So in the example above, if you specify 4 as the number of elements, then 3 elements are generated in addition to the master element. If the user-specified increment is 2, these are numbered Elements 3, 5 and 7.

RELEVANT KEYWORDS

- `*ELEMENT` is used to specify the finite element connectivity of the structural model.
- `*LINES` is used to define a line (and sections within that line), by specifying relevant set names, lengths, start and end locations, and mesh generation settings.

Cables

OVERVIEW

A powerful feature in Flexcom is the cable pre-static step, which you can invoke to provide the full finite element analysis with a first approximation to the static configuration of a structure with low bending stiffness. To invoke this option, you identify all or parts of a structure as forming a cable or a number of cables between specified end points, these end points being nodes of the finite element discretisation. The static cable profile between the end points is then computed by the pre-static step, using the cable catenary equations, prior to a full finite element analysis with bending and torsion included.

The advantage of this facility is that it allows you to define nodes in terms of an (unknown) cable profile. Without this feature, you would have no alternative other than to perform an initial static analysis in a known configuration (typically a straight line) where you would be able to define the nodal coordinates using one of the methods described above. From this you would then move the cable to the required catenary configuration. This point is discussed in considerable detail in Cable Pre-Static Step.

The rest of this section goes into considerable detail about cables and the Flexcom cable pre-static step, and provides guidelines and recommendations for their use. Much of this information is useful in understanding the internal workings of Flexcom and in correctly employing cables in your model building. However, the use of cables has to a large extent been superseded by the Lines capability, and the majority of guidelines presented in the rest of this section are automatically adhered to when cables are generated automatically from lines. So unless you propose to use cables in preference to lines, you can skip reading this section on cables and proceed directly to Lines.

Further information on this topic is contained in the following sections:

- Cable Data

- Cable Pre-Static Step
- Approximate Nodal Locations
- Exact Nodal Location
- Using Multiple Cables
- Cables on Seabed
- Defining Nodes in Terms of Cables

RELEVANT KEYWORDS

- `*CABLE` is used to specify that part of the structure forms a cable between specified end points
- `*NODE,CURVILINEAR` is used to specify nodes at particular locations along a cable, or to specify a number of equally spaced nodes along a cable segment between two nodes.

Cable Data

Four data items are required from the user to define a cable, namely

i. a cable number (for identification purposes)

ii. the cable start node

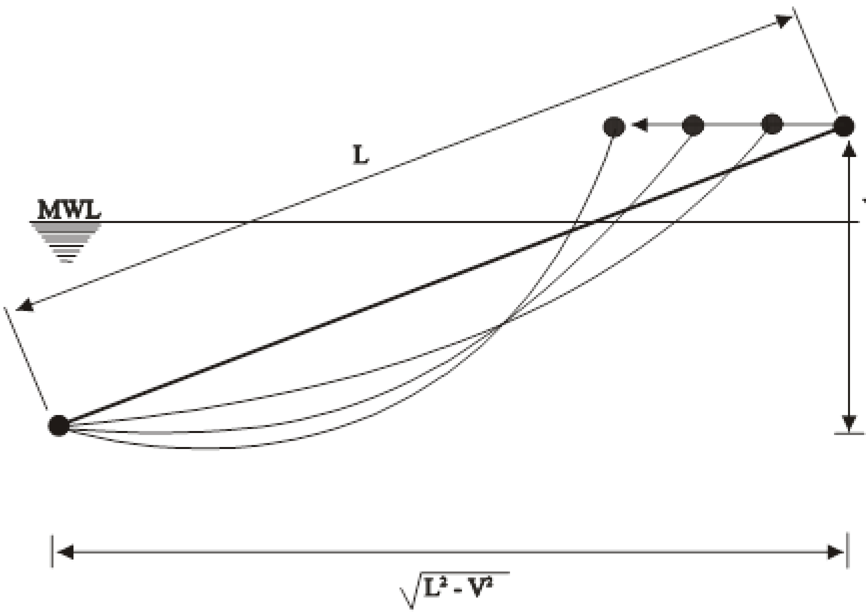
iii. the cable end node

iv. the cable length

The start and end nodes must be nodes of the finite element discretisation, defined using either of the methods described earlier. However, the nodal coordinates input for the cable start and end nodes are generally approximate locations only (with two exceptions), and merely constitute an initial guess or approximation to the actual position. The reason for this is that in general these actual positions are a function of the cable profile, which is not known a priori. This point is discussed in detail in Approximate Nodal Locations.

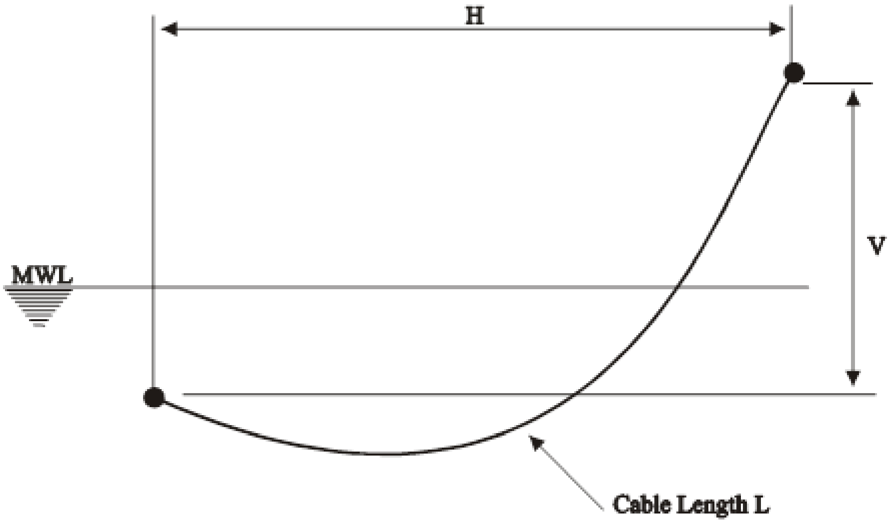
Cable Pre-Static Step

The use of the cable pre-static step is best illustrated by means of an example. Suppose that you want to perform an analysis of a simple free hanging catenary system such as that shown in the Example Catenary Configuration below. For simplicity, assume that you require the static configuration due to gravity and buoyancy only. What you would know in advance of setting up this analysis would be the length of the cable, the various cable properties (mass and stiffness, assumed uniform for simplicity), and the horizontal and vertical distances between the catenary end points (denoted H and V on the Example Catenary Configuration). How would you then go about performing this analysis?



Example Catenary Configuration

Obviously the cable profile is part of the solution sought, so defining the finite element nodal coordinates using only the standard features described earlier (define directly and generate on a straight line) presents difficulties. Using this method the best option would be to locate one end of the cable at some arbitrary position in which the cable is stretched out in a straight line, such as that shown in the Initial Static Analysis below This position is easy to calculate, and specification of the nodal coordinates is then straightforward. However, an initial analysis must now be performed to move this end into its required starting position. Although this procedure is perfectly valid in Flexcom, it does add an additional step simply to arrive at the required starting position. For complicated and/or sensitive systems, performing this initial static solution can sometimes prove difficult.

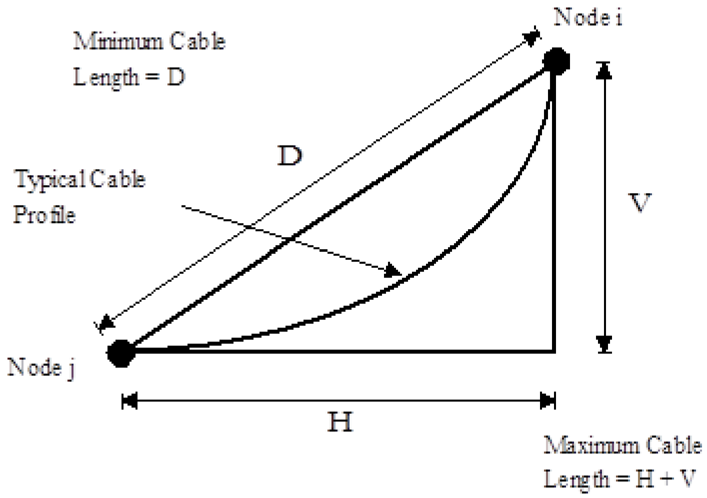


Initial Static Analysis

The Flexcom pre-static step is designed to eliminate this unwanted analysis stage. The procedure in the example above would be as follows. Firstly, you would define the coordinates of the start and end nodes in the actual initial static configuration shown in the above Example Catenary Configuration. You would then specify that the structure forms a cable of specified length between these points. You would next use the options for Defining Nodes In Terms Of Cables to distribute nodes along the cable, although its profile is not yet known. Following specification of the connectivity, properties and remaining data, you would run Flexcom. The program pre-static step would run automatically, and calculate an initial estimate of the locations of the catenary intermediate nodes using the cable catenary equations, neglecting bending and torsion. These estimates would then be passed to the main analysis module, which would solve the full finite element equations with bending and torsion included, for the actual static configuration. Convergence would typically be achieved in a small number of iterations.

Approximate Nodal Locations

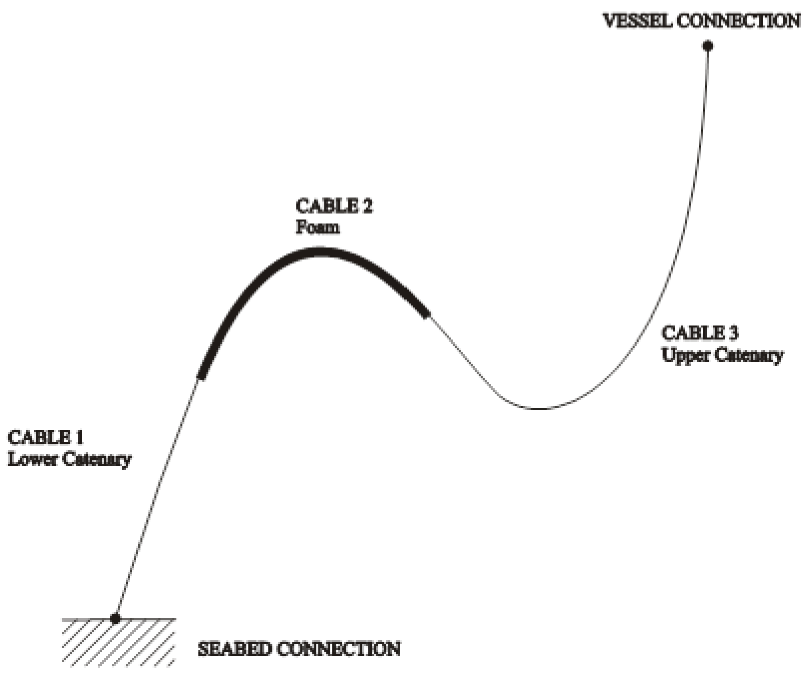
When specifying approximate nodal locations, Flexcom requires only that you satisfy two common sense conditions, which can be simply expressed as follows. Firstly, the distance between cable end nodes must not be greater than the length of the cable joining them - the alternative is physically impossible. Secondly, if one of the cable end nodes is on the seabed, the sum of the vertical and horizontal distances between cable end nodes must be greater than the cable length (otherwise the cable profile cannot be found using the catenary equations). These conditions are illustrated schematically below.



Maximum and Minimum Lengths for Given Nodes

Here the conditions are illustrated in slightly reverse order, by showing the maximum (if Node j is on the seabed) and minimum cable lengths possible for a given nodal separation. It is important to understand that when you calculate approximate locations it is not necessary for program convergence that they are close to the eventual positions. As long as the above conditions are satisfied, the program should converge quickly to the correct locations.

Why the actual coordinates specified for the cable start and end nodes generally represent only approximate locations is again best illustrated by means of an example – consider the case of a steep wave flexible riser illustrated schematically below.



Steep Wave Riser

In defining this system in terms of cables, you would normally use three cables, the first for the lower catenary section from the seabed connection to the start of the buoyant section, the second to represent the buoyant section, and the third for the upper catenary, from the end of the buoyant section to the vessel connection point. To define these three cables, you would need to first define four nodes, at the seabed and vessel connection points, and at the changeover between the buoyant section and the lower and upper catenaries respectively. The coordinates of the two connection points will, of course, be known in advance. Not so the coordinates of the changeover points, because again these locations are part of the solution and will change if the various cable lengths and properties change. How then to define nodal coordinates for these points? The answer is that Flexcom requires only approximate locations for these nodes that satisfy the common-sense conditions described above.

Focusing on the steep wave riser example, the following illustrates how coordinates for the changeover nodes might be calculated for the following values:

Cable lengths:	Lower catenary	80m
	Buoyant section	130m
	Upper catenary	240m

Coordinates: Seabed (0,0,0)

Vessel (310,120,0)

Since the lower catenary will be nearly vertical in the static offset position, possible values for the coordinates of the lower catenary/buoyant section changeover point might be (77,20,0). For these coordinates, the straight-line distance between ends is 77.13m (less than 80m), and the sum of the horizontal and vertical separations is 97m (greater than 80m). So both criteria are satisfied, and (77,20,0) represents an acceptable approximation. For the upper changeover point, possible coordinates might be (110,70,0) – you might want to prove to yourself that these are again acceptable values.

Although it might appear that calculating approximate coordinates in this way requires some ingenuity, in general acceptable values can usually be quickly found from simple geometrical considerations. Finally, it is worth commenting again that where coordinates are approximate, it is not necessary for program convergence that the values specified be close to the eventual correct values (which, of course, are unknown). As long as the approximate coordinates are physically reasonable, the program will converge quickly to the correct solution.

Exact Nodal Location

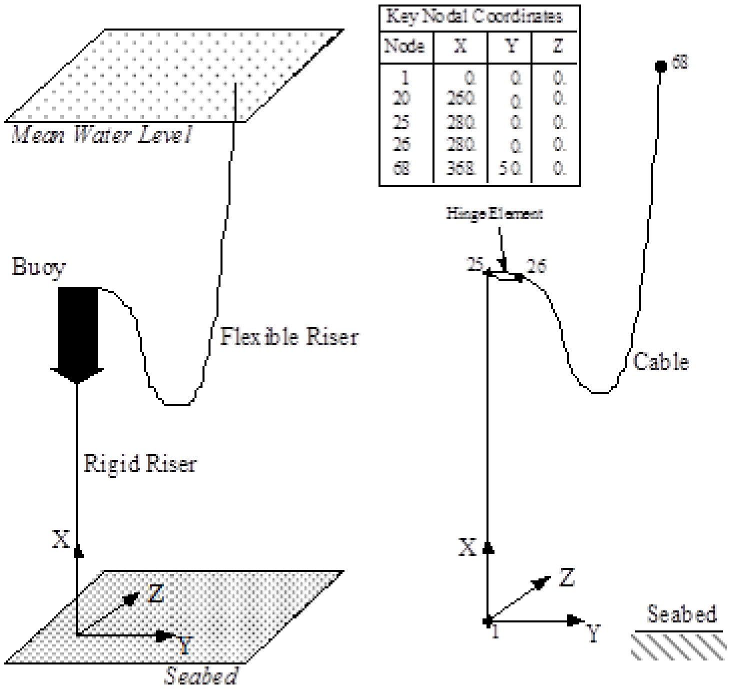
As mentioned, there are two exceptions where nodal locations are exact rather than approximate. These are:

i. where a particular node is a boundary node, specifically a node at which a translational boundary condition is applied

ii. where a node is also on an element which is not part of a cable (for convenience, an element which is not on a cable is denoted a rigid element, regardless of its actual stiffness - in fact these elements are often hinges)

In these situations, the specified nodal coordinates are exact locations. These two exceptions are illustrated in the figures below.

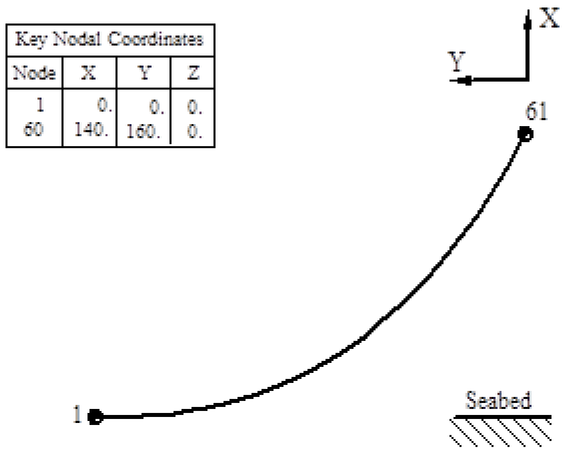
The first figure shows a simple catenary riser. To specify this system you would input the coordinates of the end nodes, define a cable between them, and apply restraints at the two ends. Obviously in this situation the locations of the end nodes are exact rather than approximate.



Node	X	Y	Z
1	0.	0.	0.
20	260.	0.	0.
25	280.	0.	0.
26	280.	0.	0.
68	368.	50.	0.

Boundary Nodes

The second case is illustrated by the figure below, which shows a hybrid riser system comprising an underwater rigid riser supported by a subsea buoy, and a flexible riser descending to the buoy from the Mean Water Line (MWL). The flexible and buoy are connected by a hinge. In modelling this system, the flexible is identified as a cable stretching from the hinge to the MWL. For the purposes of the cable pre-static step, the location of the node at which the hinge and the flexible meet is considered to be exact, and in fact the cable module applies a boundary condition here and finds the cable catenary between this fixed point and the vessel connection point at the MWL. Of course in the subsequent full finite element analysis this node is again free to displace (unless restrained by the user).



Intersection of Cable and Non-Cable Elements

It is important to remember that cables you define connect the specified end points for the purposes of the cable step only. For the purposes of the complete finite element analysis with bending and torsion, the full connectivity of all elements comprising each cable must be input using either of the methods discussed in Elements.

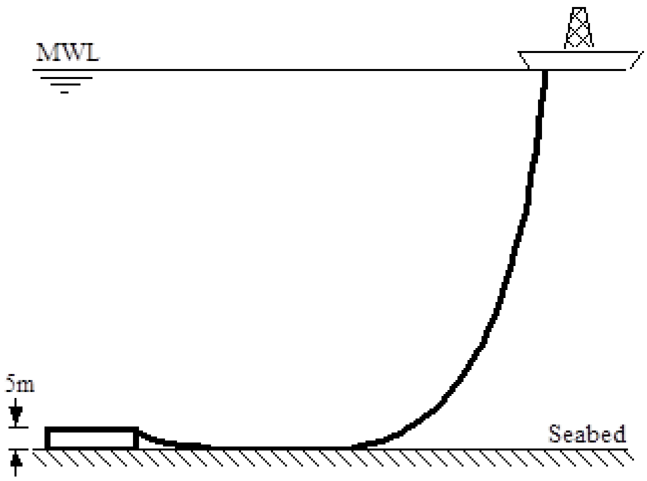
Using Multiple Cables

The data required by the cable pre-static step for each cable are its axial stiffness and its apparent weight (weight in water) per unit length. These are automatically retrieved or calculated from the data you specify under the `*GEOMETRIC SETS` keyword. One important point to note is that the cable properties are assumed constant over each cable. Where a cable is comprised of elements with different stiffnesses and/or weights, constant values are automatically computed as an average of the various values for these elements. You should therefore not group large numbers of elements with very different properties together in single cables, since a particular average might give a poor representation of the distribution of a property over the cable.

Cables on Seabed

If part of a cable is expected to lie on the seabed, then the internal operation of the cable pre-static step imposes a minor constraint on the specification of the end nodes for that cable. Specifically, the node at the lower end of the cable must be positioned initially on the seabed. If the actual final position of this node is not on the seabed, but rather some distance above it, this must be achieved by the application of a boundary condition.

Consider for example the simple catenary shown below. It is required that the riser be connected to the PLEM at an elevation of 5m above the seabed. To use a cable in this analysis, you must specify that it extends from the node at the PLEM (Node 1 in this case) to the node at the vessel (Node 61). However, if you locate Node 1 at (5, 0, 0), Node 61 at (140,160,0) and define a cable of length 260m between them, Flexcom may fail to converge. You must instead locate Node 1 at (0,0,0) and apply a vertical displacement of 5m as a boundary condition. Convergence will then be achieved in a small number of iterations.



Cable Specification for Sections with Partial Seabed Contact

The use of the cable pre-static step is of course optional, and should only be used where the cable catenary equations can genuinely provide an initial approximation to the final static configuration. As an extreme example, a pipe or cable lying completely on the seabed cannot be analysed by the cable pre-static step. Note also that the cable pre-static step is automatically run, without user interference, whenever you define a cable or cables.

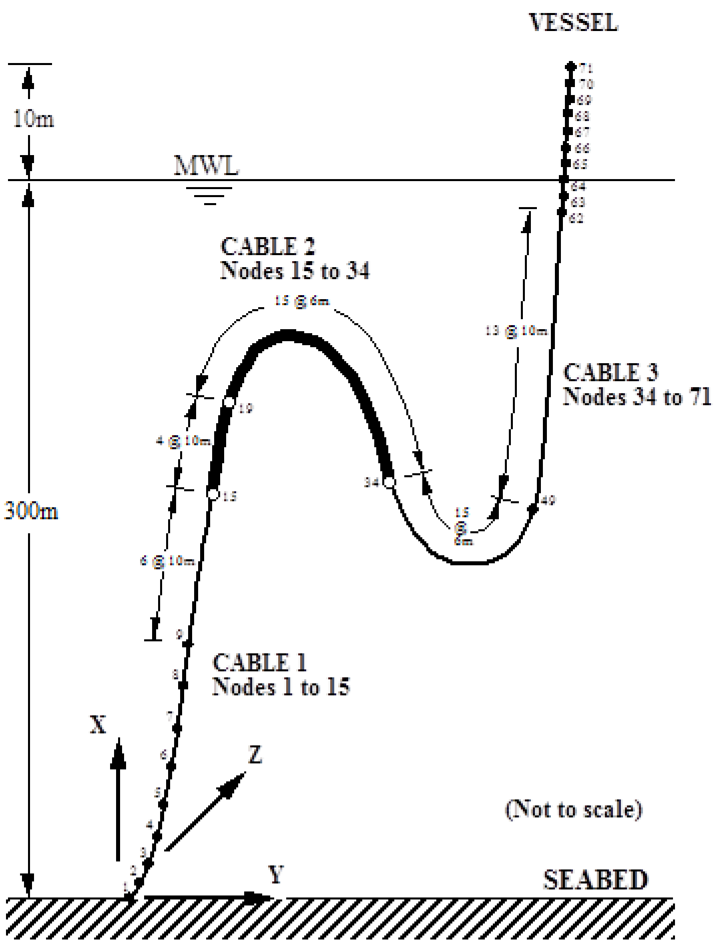
Defining Nodes in Terms of Cables

Two methods were noted earlier for defining nodal locations in terms of cables, denoted (iii) directly in terms of distance along a cable, and (iv) generated along a cable. It is in fact these facilities which are the purpose of using cables in the first place.

The first of these facilities is largely self-explanatory, and requires three data inputs, namely the node number, the number of the cable on which it is located, and the required distance along the cable, which is measured from the specified cable start node.

The generation option permits nodes to be uniformly distributed between two nodes on a particular cable. The user inputs in this case are the cable number, the first node number, the last node number, and the increment for assigning node numbers to the generated nodes (which defaults to a value of 1). The first and last nodes are not necessarily the cable start and end nodes, but if they are not, their location must be defined directly using the method just discussed.

The operation of both of these facilities, and indeed of the whole cable pre-static step, is illustrated below. Here a steep wave riser is defined in terms of 3 cables, a lower catenary from Nodes 1 to 15, a section with buoyancy modules attached extending from Nodes 15 to 34, and an upper catenary from Nodes 34 to 71. The coordinates of four cable end nodes (1, 15, 34 and 71) are required inputs for this riser. Of these, two are exact locations (Nodes 1 and 71), since boundary conditions will be applied here. On the other hand, the exact locations of Nodes 15 and 34 are a function of the unknown static structure configuration, and so can be specified only as approximate locations which satisfy the conditions discussed in the previous section.



Key Nodal Coordinates			
Node	X	Y	Z
1	0	0	0
15	77	20	0
34	110	70	0
71	310	120	0

Cable	Length (m)
1	80
2	130
3	240

Node	Cable	Distance
2	1	0.25
3	1	0.50
4	1	1.00
5	1	1.50
6	1	2.50
7	1	5.00
8	1	10.00
9	1	20.00
19	2	40.00
49	3	90.00
62	3	220.00
63	3	225.00
64	3	230.00
65	3	235.00
66	3	237.50
67	3	238.50
68	3	239.00
69	3	239.50
70	3	239.75

Defining Nodes In Terms Of Cables

Equivalent Nodes

OVERVIEW

A Flexcom feature that can simplify setting up and refining finite element models is the option of equivalent nodes. This allows you to specify that two individual nodes, which have been previously defined, are in fact the same node from the point of view of the finite element discretisation.

The usefulness of this feature is best illustrated by an example. Suppose we have defined two cables. The node numbers on the first cable run from 1 to 51, while those on the second cable run from 101 to 151. If we wished to join these cables end to end, we could change the node numbering on the first cable to run from 1 to 50, with the last node numbered 101. However this is not very convenient, and destroys the sequential node numbering on the first cable.

Instead, we can simply define that Nodes 51 and 101 are equivalent nodes, that is, we can specify that they are actually the same node. This has the effect of joining the two cables end to end, while preserving the sequential node numbering scheme on both cables. This makes subsequent refinement of the finite element mesh on either cable a much simpler task. Of course, for two nodes to be defined as equivalent, they must have the same nodal coordinates.

The use of equivalent nodes is particularly associated with the line generation facility (which is described in Lines). Individual lines are defined separately in terms of start and end points during the model building process. Where two lines are to be connected together, the relevant points at the intersection (e.g. the end of the first line and the start of the second line) are identified as equivalent, which facilitates the finite element mesh generation.

RELEVANT KEYWORDS

- `*EQUIVALENT` is used to specify that two individual nodes are a single equivalent node.

If you would like to see an example of how this keyword is used in practice, refer to Example A01 - Deepwater Drilling Riser.

Node and Element Labels

Another capability closely associated with the operation of the lines facility is the use of node labels. This allows you to associate a descriptive name with a node of the finite element model. Thereafter you can refer to the node label rather than the node number in other areas of the program – for example, when you are assigning boundary conditions, or defining nodal equivalences. This is very useful when combined with the lines capability, as described shortly (in particular in Start and End Locations and Boundary Conditions). In a similar manner, element labels may be used to associate a descriptive name with an element of the finite element model. Again refer to Line Start and End Locations for further details.

Lines

OVERVIEW

The lines functionality is a relatively recent addition to Flexcom, and provides an automatic mesh creation facility for nodes, elements and cables. (It is worth noting that the word ‘line’ used in this context does not imply a straight line section; a catenary section is equally valid, as demonstrated shortly.)

The main aspects to creating a line are as follows:

- Associating a descriptive name with the line
- Defining start and end locations
- Specifying the line length
- Specifying line mesh generation settings
- Assigning structural and hydrodynamic properties
- Defining boundary conditions at line end points

Some advanced topics which merit further discussion also include:

- Connecting lines together
- Modelling repeating sub-sections within a single line
- Using the line facility to create pipe-in-pipe configurations
- The significance of local undeformed element axes, and how this pertains to lines

Each of these topics is discussed in further detail in the following sections.

RELEVANT KEYWORDS

- `*LINES` is used to define a line (and sections within that line), by specifying relevant set names, lengths, start and end locations, and mesh generation settings.
- `*LINE LOCATIONS` is used to define named locations along a line, allowing you to request that nodes be positioned exactly at specific lengths along a line.
- `*LINE SECTION GROUPS` is used to define groups of line sections which may be reused and repeated within sections defined in *LINES.
- `*LINES PIP` is used to specify that two lines are connected in a pipe-in-pipe (or pipe-on-pipe) configuration. `*LINES PIP` provides a high degree of automation in terms of setting up pipe-in-pipe configuration, but for complete generality and increased user control, the `*PIP CONNECTION` and `*PIP SECTION` keywords are recommended.

If you would like to see an example of how these keywords are used in practice, refer to Example H02 - J-Tube Pull-In.

Descriptive Name

Considering again the Steep Wave Riser example – if this were to be modelled using lines, you would probably create a single line with three subsections. In fact, this is another advantage of using lines over cables. As described in Using Multiple Cables earlier, it is not advisable to group sections of very different properties together in to a single cable, as the average properties used by the cable pre-static step may lead to an inaccurate initial estimate of the actual configuration. So the steep wave riser requires three separate cables, whereas a single line definition is sufficient. Furthermore, the use of lines means that you do not have to manually estimate initial coordinates for the cable intersection points (i.e. the ends of the buoyant section).

The line might be called “Riser”, with subsections entitled “Lower Catenary”, “Buoyant Section” and “Upper Catenary”, respectively. It is important to define meaningful names, as these names will be used by the program to automatically create relevant element sets for you, which you can subsequently reference when you are assigning structural and hydrodynamic properties to your model. This aspect is discussed in more detail later in Structural and Hydrodynamic Properties.

Start and End Locations

Start and end locations for lines are defined directly in terms of global coordinates. Definition of these location points is quite similar to the explicit definition of finite element nodes discussed previously, with the exception that the nodes are assigned descriptive names rather than node numbers. It is important to associate meaningful names with line start and end locations, as these names will be used by the program to automatically create relevant node and element labels. For example, you can subsequently reference these node labels when you are applying boundary conditions to your model. This aspect is discussed in more detail shortly, and in the ‘Boundary Conditions’ section later.

Referring again to the Steep Wave Riser example, you would define two location points to define the start and end points of the “Riser” line. The coordinates of these locations would correspond to the original cable end nodes (i.e. Nodes 1 and 71) in the example. The location points would have associated names such as “Riser Start” and “Riser End”, respectively.

The significance of these names is that Flexcom will assign a node label with each name to the nodes that it generates at these locations during mesh generation. Thereafter you can specify, for example, a boundary condition at a location although you have no idea of the actual number of the node there. This is very powerful, and means that you can change minimum and maximum element lengths without, say, any boundary condition or point load data being lost or having to be redefined.

Element labels are also automatically created to identify the first and last elements of each line (and subsection of each line if appropriate) in your model. This feature is particularly useful during postprocessing – for example, you could monitor effective tension at the end of a line without having knowledge of the actual element numbers used in the finite element mesh. In the case of the steep wave riser example, Flexcom will automatically create element labels called “Lower Catenary_First”, “Lower Catenary_Last”, “Buoyant Section_First” etc. and associate these labels with the relevant elements in the finite element mesh.

Line Length

The line length would be specified as 450m, corresponding to the total riser length shown in the Steep Wave Riser example. Based on the length of a line, and the straight line distance between its start and end locations, Flexcom automatically determines whether the line should be modelled internally using a curved (cable) or a straight section. If the straight line distance between the start and end locations is less than the specified line length, a cable (or several cables) is used. Otherwise the line is modelled using a straight line of nodes and elements. Note that line length is an optional input, and if no length is specified, then the length is equal to the straight line distance between its end points. Obviously cables will be automatically used by Flexcom in the case of the steep wave riser example.

Flexcom will typically use a separate cable to model each subsection of a line which has different structural properties. In the case of a steep wave riser for example, this is perfectly reasonable. However, in very occasional circumstances, it may be desirable to model several discrete subsections using a single catenary representation. For example, if you are modelling a buoyant section of riser which is comprised of standard material supported by numerous discrete buoyancy modules along its length, a single catenary model is preferable. Otherwise a large number of separate cables will be used internally to model what is essentially a simple catenary, and some or all of the individual cables are likely to violate the basic assumptions associated with catenary equations. In such circumstances, you can use the Force Single Catenary option to explicitly request that an entire line is to modelled using one single catenary/cable.

Line Mesh Generation

MESH GENERATION

You have control over the distribution of the elements along the line, via the specification of desired maximum and minimum element lengths, and also the division of the line into several subsections if required. The meshing algorithm automatically generates a finite element discretisation based on the guidelines you provide. Ideally, the aim is to create a mesh which is sufficiently dense to accurately capture structural behaviour, while not being unduly complex and resulting in lengthy simulation times. For example, if defining a steel catenary riser, you would typically require a relatively fine mesh in the touchdown region, while a more coarse mesh would suffice for the portion of riser lying flat on the seabed, or the portion extending upwards through the water column. If your model is likely to experience significant compressive loads, you will need to use a relatively fine mesh in order to ensure that the critical Euler buckling load is not exceeded - refer to Compression and Buckling for further details.

The meshing algorithm also attempts to prevent large changes in relative element length across the finite element mesh by gradually stepping up and down element lengths along the structure, and to avoid over-meshing by using longer elements in the middle of long sections of continuous properties. Intersections between different lines, or between different sections within a single line, are generally the points of most interest and therefore have a more refined mesh. It is desirable to step up the element lengths away from intersection regions and use larger elements towards the centre of homogenous sections. The meshing algorithm ensures that the ratio between the lengths of adjacent elements cannot exceed a certain value. The maximum ratio defaults to 1.5 (Standard option), but may be reduced to 1.25 (Fine option) or 1.1 (Super Fine option). The actual value of the ratio is automatically selected to correctly fill the meshed section with elements.

Referring again to the Steep Wave Riser example, you might specify minimum and maximum element lengths of 1m and 5m, respectively, for each of the three subsections, “Lower Catenary”, “Buoyant Section” and “Upper Catenary”. Focusing on the lower catenary section, the element length at the seabed connection would be in the region of 1m, would increase gradually (subject to the Standard aspect ratio constraint mentioned above) to 5m with increasing elevation above the mudline, before gradually reducing again approaching the buoyant section to a length of 1m. This would lead to a total of 21 elements, with corresponding lengths as presented below:

Element Number	Element Length (m)	Total Length (m)
1	0.985	0.985
2	1.477	2.462
3	2.215	4.677
4	3.323	8.000
5	4.809	12.809
6	4.809	17.618
7	4.809	22.426
8	4.809	27.235
9	4.809	32.044
10	4.809	36.853
11	4.809	41.662
12	4.809	46.470
13	4.809	51.279
14	4.809	56.088
15	4.809	60.897
16	4.809	65.705
17	4.809	70.514
18	4.809	75.323
19	2.215	77.538
20	1.477	79.015
21	0.985	80.000

Sample Element Lengths

LINE LOCATIONS

One further point to note is that the automatic meshing algorithm allows you to request that nodes be positioned exactly at specific lengths along a line. This is in addition to the guidelines you provide in relation to desired maximum and minimum element lengths. It is not particularly important in the context of the steep wave example, but can be useful if in certain circumstances. For example, if you wish to apply a point load at a certain distance along a line, or to connect another component (e.g. another line) to a line at a specific location, you would typically position a node on the line at the required location. It is also important in the context of pipe-in-pipe models, although the process in that case is automatic and seamless to the user.

Refer to Meshing Algorithm for further information on this feature.

Occasionally you may be trying to exercise very precise control over the mesh density in a particular line. In these cases a more detailed understanding of how the meshing algorithm works may be required to achieve the precise spacing you require.

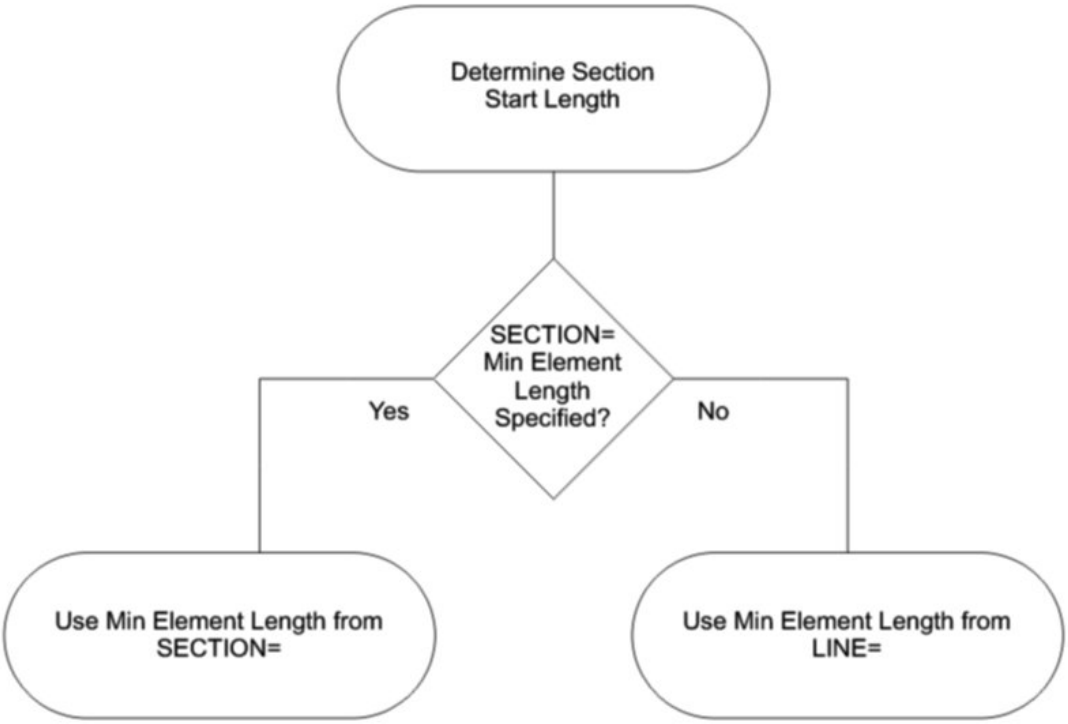
The meshing algorithm views the line as a series of sections. Some sections are explicitly defined by the user. Parts of the line not covered by an explicit section are treated as implicit sections, with minimum and maximum element lengths as for the line itself.

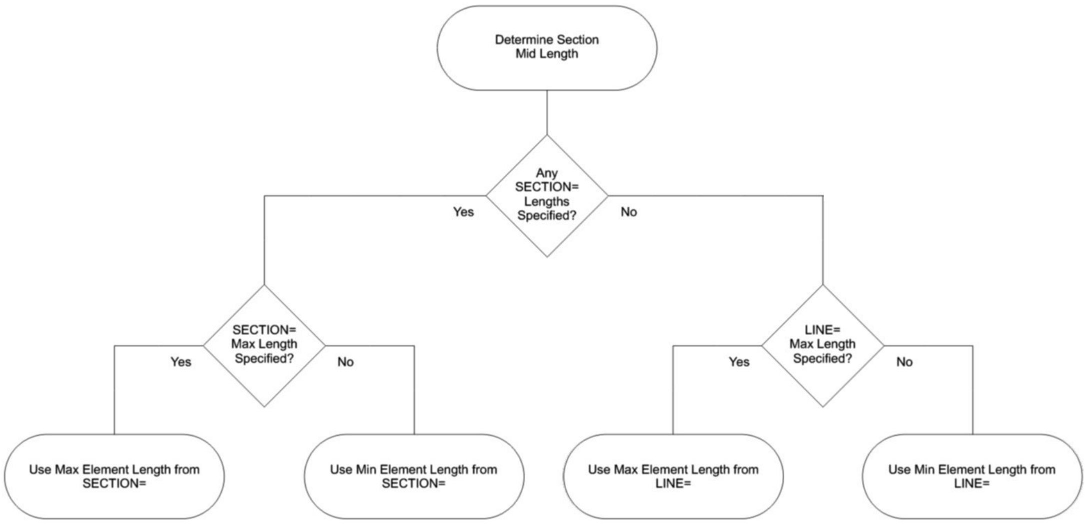
When meshing a section, the meshing algorithm treats the section as if it has three characteristic lengths:

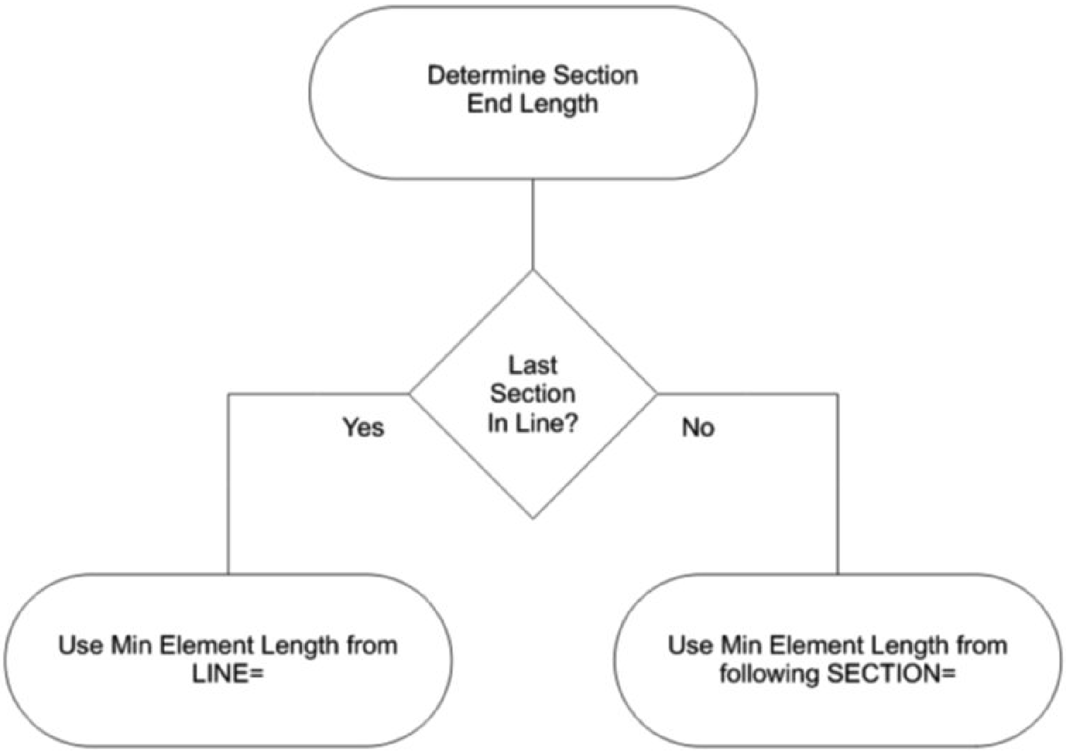
1. The length at the start of the section. This is a mandatory input which corresponds to the Min. Element Length at Section Start.
2. The length in the middle of the section. This is an optional input which corresponds to the Max. Element Length in Section. If not explicitly specified, a uniform mesh density is typically created, based on the specified length at the start of the section.
3. The length at the end of the section. This is an optional input which corresponds to the Min. Element Length at Section End. If not explicitly specified, the element length at the end of a particular section is aligned with the element length at the start of the next section (or the end of the line itself, if it is the last section). This allows the meshing algorithm to automatically blend sections into one another.

Assuming the section is long enough, the element lengths will vary across the section blending between these lengths as the algorithm meshes from start to end.

The following flowcharts illustrate how the meshing algorithm operates.







If in doubt, consult the Line Generation Report to see what your element lengths are.

Structural and Hydrodynamic Properties

The preceding discussion focused mainly on the operation of the meshing algorithm and the resulting finite element mesh. Obviously the next step is the assignment of appropriate structural and hydrodynamic properties to the elements themselves. The various program options for structural and hydrodynamic properties are discussed in detail in Geometric Properties and Hydrodynamic Properties, respectively, but a brief introduction to the topic is appropriate at this juncture.

Flexcom uses the concept of element sets to define the physical properties of the finite element model. Groups of elements with identical properties are logically combined into named element sets, properties are then assigned on an element set-by-set basis, and the program automatically associates these properties with the individual elements of each set. Typical structural properties would include bending, torsional and axial stiffnesses, mass per unit length etc., while hydrodynamic properties are defined in terms of various drag and inertia coefficients. If you choose to define elements directly (as discussed in the sections earlier covering Nodes and Elements and Cables), then the onus is on you to create the relevant element sets manually. However, if you are utilising lines to create your model, relevant element sets are created automatically for you, based on the names of each line (and subsection of each line if appropriate) in your model. You can then refer to these element sets when you are assigning structural and hydrodynamic properties in the normal fashion. This is why it was suggested earlier that it is important to define meaningful names for ease of reference. Care should be taken also to ensure that the line names are unique, in order to avoid any possible ambiguity regarding the element set definitions.

Referring again to the Steep Wave Riser example, four element sets will be automatically created in this case, called “Riser”, “Lower Catenary”, “Buoyant Section” and “Upper Catenary”, respectively, corresponding to the line and its subsections defined in the model. The composition of each set will be a list of elements used internally to model the relevant section. For example, the set “Lower Catenary” will contain the elements numbered 1-21, as illustrated in the Sample Element Lengths table.

Boundary Conditions

As mentioned previously, you assign line start and end locations descriptive names, and Flexcom automatically creates relevant node labels for you, which are the as the names of each line start and end location. Node labels are also assigned to intersection points between various subsections within each line if present, plus to any additional locations you define. You can then refer to these node labels when you are assigning boundary conditions, or defining nodal equivalences.

Referring again to the Steep Wave Riser example, the model will automatically have two labelled nodes, namely “Riser Start” (referring to Node 1), and “Riser End” (referring to Node 71). Typical constraints for the steep wave riser would be a pinned connection at “Riser Start”, and a built-in vessel attachment at “Riser End”. Note that while the relevant node labels are created automatically for you, it is your responsibility to assign appropriate boundary conditions to them if required.

Connecting Lines

Occasionally you may wish to connect a line to some other component, such as another line, or perhaps an element which has been created explicitly (in terms of its end nodes, or as part of a cable definition). A useful feature in this context is that of equivalent nodes (as previously described in Equivalent Nodes). This allows you to specify that two individual nodes, which have been previously defined, are in fact the same node from the point of view of the finite element discretisation. If you are working with explicitly defined nodes, you reference the actual node numbers when defining nodal equivalences. If you are utilising the line facility to create your model, you can refer to the relevant node labels rather than node numbers, and define nodal equivalences in the normal fashion.

Modelling Repeating Sub-Sections

OVERVIEW

While the Lines modelling feature is far more powerful than the traditional approach of working directly with nodes, elements and cables, using the `*LINES` keyword alone does not result in optimum user experience. Specifically, each sub-section within a line (i) has to be defined explicitly, and (ii) requires a unique name.

Where a model contains a large number of similar sections, another complementary feature is provided which allows repeated sections to be modelled quickly and easily. A group of sub-sections may be defined under a `*LINE SECTION GROUPS` keyword, which may then be subsequently referenced under the normal *LINES keyword as required.

When the repeat section group option is invoked, Flexcom automatically populates each line section with as many of the sub-section groups as will fit between the specified start and end distance for that line section. If the repeated group length doesn’t divide into the distance evenly, a notification message is issued to that effect, and the last section group truncated appropriately.

Consider for example the sample floating hose model which is shipped with Flexcom. It contains 52 identical sections of buoyed hose, separated by standard/unbuoyed hose sections in between. This geometry would require over 250 lines of keyword data to construct using *LINES alone. However, using *LINES SECTION GROUPS in conjunction with *LINES results in an over 80% reduction in volume of required keyword data.

RELEVANT KEYWORDS

* `*LINES` is used to define a line (and sections within that line), by specifying relevant set names, lengths, start and end locations, and mesh generation settings.

* `*LINE SECTION GROUPS` is used to define groups of line sections which may be reused and repeated along line sections.

If you would like to see an example of how these keywords are used in practice, refer to E03 - Floating Hose.

Pipe-in-Pipe Configurations

A further level of complexity afforded by the line modelling facility is the ability to model pipe-in-pipe (or pipe-on-pipe) scenarios. You model the inner and outer pipes using separate lines as described in the preceding sections, and then specify that you would like both pipes to be connected in a pipe-in-pipe type configuration. One line is designated the primary line (typically the outer line), and the other termed the secondary line. Note that while the remainder of this section refers to dual pipe scenarios, it is also possible to model riser bundles (with one primary line and several secondary lines) or pipe-in-pipe-in-pipe configurations (with a primary line whose secondary line also serves as a primary line for another secondary line).

The interaction between inner and outer pipes falls into three categories – linear, non-linear and rigid connections. Both linear and non-linear connections may be considered to operate in a similar manner to spring elements, providing resistance to relative motion of the inner and outer pipes in the transverse plane. Linear connections are typically used to model centralisers, which are characterised by a relatively high linear stiffness. Away from centralisers, the inner and outer pipes are free to move radially relative to one another. Generally speaking, the modelling arrangement should be such that when inner and outer pipes come into contact, there is no penetration of the outer by the inner, and vice versa, and this is achieved by means of non-linear connections. The (user-defined) force-deflection relationship typically have a very low stiffness when the pipes are in their respective un-deformed positions, but the stiffness increases exponentially as the gap between pipes approaches zero. Note that neither linear nor non-linear connections provide any resistance to relative axial motion of the inner and outer pipes. Rigid constraints are typically used to model bulkheads, where the inner and outer pipes are rigidly connected to each other. In the case of pipe-in-pipe model, this is achieved via the nodal equivalence facility discussed previously, but in this case the process is performed automatically by Flexcom, so you are not required to explicitly define any nodal equivalence. For pipe-on-pipe cases, rigid massless elements are inserted between the pipes at appropriate locations.

All connections (whether linear, non-linear or rigid) are defined in terms of distances along the line. You specify a start length, and end length and a spacing when generating connections along the line. As mentioned previously, the automatic meshing algorithm allows you to request that nodes be positioned exactly at specific lengths along a line (in addition to the guidelines you provide in relation to desired maximum and minimum element lengths). When you specify pipe-in-pipe connections, you are effectively requesting that nodes be placed at appropriate locations, and although the process may appear seamless, this is how your request is handled internally.

Refer to Pipe-in-Pipe for further details on pipe-in-pipe modelling in Flexcom.

Auxiliary Bodies

OVERVIEW

One additional feature relevant to setting up models in Flexcom is the use of so-called auxiliary bodies. Auxiliary bodies are not part of the finite element discretisation, but may be used to give a visual indication of the location of objects (such as floating vessels). So they are included for illustrative purposes only – they have no structural properties, you are not interested in determining forces or stresses within them, and they have no structural function whatsoever.

AUXILIARY NODES, ELEMENTS AND PANELS

The coordinates of auxiliary nodes are defined in the global coordinate system in the normal fashion. Auxiliary nodes may be connected using auxiliary elements or auxiliary panels. Auxiliary elements are used for defining lines (each element connects two auxiliary nodes), while auxiliary panels are used for defining solid triangular surfaces (each panel connects three auxiliary nodes). It’s also possible to assign different colours (using standard RGB values) to different panels for enhanced visual displays. The coordinates of all auxiliary nodes and the connectivity of all auxiliary elements and panels must be defined directly – there are no associated generation facilities.

A group of auxiliary elements and/or panels can be grouped into an auxiliary body. Such bodies can then be associated with a vessel for which Response Amplitude Operators (RAOs) have been defined. The auxiliary elements and/or panels will translate and rotate with the vessel giving a clear representation of the motion of the vessel when viewed using the display facility.

INTEGRATED VESSEL/COMPONENT PROFILES

Creation of individual auxiliary nodes, elements and panels can be quite tedious and time consuming. These concepts are somewhat outdated, and have effectively been superseded by the integrated vessel/body features which provide a range of standard vessel and subsea component profiles.

The profile is defined in XML format, and several sample profiles are available in your local Flexcom installation directory, under the sub-folder ‘StandardProfiles’. Each profile file contains a range of input data, capable of defining standard shapes such as boxes, cylinders etc., as well as arbitrary shapes defined by a series of points about which a mesh is constructed. All locations within the profile file are defined with respect to a local origin (0, 0, 0). The location of this origin in the global axis system is then defined by the X, Y, & Z coordinates of the body centre.

Refer to Creating Customised Object Profiles for further information on building your own models.

Sample Helicopter Profile

POST-PROCESSING

Auxiliary elements have existed since early versions of Flexcom. In terms of visualisation purposes, they have effectively been superseded by auxiliary panels and integrated vessel/component profiles. It should be noted however, that auxiliary elements do have one additional advantage – they may be referenced during postprocessing in the computation of element angles. For example, you may wish to monitor the angle between the riser at the vessel connection and some part of the vessel itself. Flexcom provides a range of outputs for this purpose under the general heading of Angles; a detailed discussion is provided in Angles Output.

RELEVANT KEYWORDS

Older keywords:

- *NODE,AUXILIARY is used to define auxiliary nodal coordinates in the global Cartesian coordinate system.
- *ELEMENT,AUXILIARY is used to specify the connectivity of auxiliary elements in the model.
- *AUXILIARY is used to specify the auxiliary elements which comprise an auxiliary body.
- *PANEL,AUXILIARY is used to specify nodes which are connected by an auxiliary panel.
- *PANEL SECTIONS,AUXILIARY is used to specify the auxiliary panels that make up an auxiliary body.

Newer keywords:

- *VESSEL,INTEGRATED is used to specify all information pertaining to a vessel. It offers a range of standard vessel profiles to provide enhanced visual appeal in the structural animation. You may choose to insert an FPSO, Semisub, Drill Ship, Spar, TLP, Installation Vessel etc. or opt to define your own customised vessel shape.
- *BODY,INTEGRATED is used to add subsea components to the structural animation for enhanced visual appeal.

If you would like to see an example of how these keywords are used in practice, refer to Example A01 - Deepwater Drilling Riser. Some samples images are shown for the Drill Ship and LMRP, BOP & Wellhead.

Creating Customised Object Profiles

INTRODUCTION TO XML

XML (eXtensible Markup Language) is a programming language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. Syntax recognition and error highlighting in Flexcom's keyword editor is achieved in XML for example, as the user interface understands the various keyword formats accepted by Flexcom based on pre-defined rules which have been implemented by the software developers. When defining a customised object profile for use in Flexcom, you are essentially communicating to the Model View what to display on screen, via a profile file which is written in XML format.

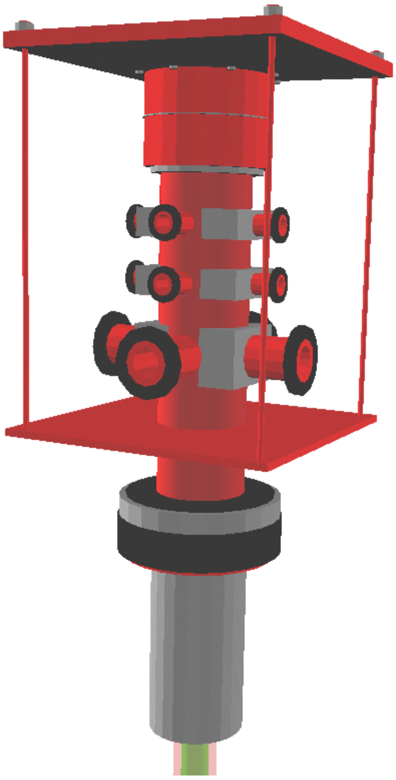
XML may look a little complicated to a first time user, but it is actually quite straightforward to interpret, and you will be confidently building your own profiles before long. If this is your first time working with XML, we suggest that you search for an introductory tutorial online - there is plenty of free training material and tutorial videos available.

SAMPLE OBJECT PROFILE

Let's look at a sample object profile to help illustrate the use of XML. The Wellhead component provided with Flexcom (Wellhead.fcvpx) is probably the simplest one to start with. It is actually very readable as you can see. It consists of 3 separate cylinders, each defined by a central axis, a centre point, a colour, a height/length, an external diameter, and a number of segments (which defines the number of flat panels which are used to represent the cylindrical surface). Beneath the XML, you can see how the component looks in the Model View, with the wellhead component (largely grey) beneath the BOP (largely red).

```
<Body Name="Wellhead">
    <!--A combination of cylinder shapes used to create the Wellhead-->
    <Cylinder Axis="1 0 0" Centre="7.5 0 0" Colour="Grey" Height="15" OuterDiameter="3.5" InnerDiameter="1.25"
    <Cylinder Axis="1 0 0" Centre="15.2 0 0" Colour="Red" Height="0.4" OuterDiameter="5.5" Segments="24"/>
    <Cylinder Axis="1 0 0" Centre="16.9 0 0" Colour="Black" Height="3" OuterDiameter="6" Segments="24"/>
</Body>
```

XML file for Wellhead Component



Wellhead (lower) & BOP (upper) Components

Even this very simple example illustrates a number of key features of XML...

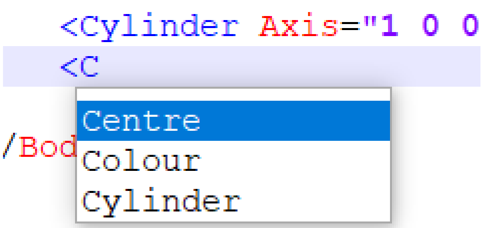
- XML documents are composed of blocks of information known as elements. The file above has 4 elements, the wellhead, and the 3 cylinders which comprise it.
- When viewed in a compatible file viewer, XML documents are neatly coloured for ease of reading and understanding. While they can be viewed in a standard internet browser, we would recommend using a standard text editing software which supports both viewing and editing. Notepad++ is a very useful text editor which is free to download and install.
- All XML documents must have a root element. This acts as the parent element for all sub-elements which it contains. The root element in the file above, and indeed all Flexcom object profiles, is denoted by "Body".
- XML elements are created using an opening tag (the "<" symbol) followed by the element type. For example, "<Body".

- All XML elements must have a closing tag. Element definitions typically span several lines so there is a separate closing tag (e.g. "</Body>") on another line further down. Simpler element definitions may be written on a single line, but the closing tag must still be present.

- Note that XML tags are case sensitive. Opening and closing tags must be written with the same case.

- User defined properties associated with elements, known as attributes, must always be enclosed in quotation marks. For example, the name of the body ("Wellhead") and any associated properties such as dimensions, colours etc. are all enclosed in quotation marks in the image above.

- Most text editors provide helpful tooltips while you type. The image below shows that once you add an opening tag and start typing, the editor suggests as many element or attribute types as it can.



- Indentation is highly recommended for clarity and ease of reading. Notice how the cylinder element tags above are indented by a few spaces to the right of the body element tags.

- The use of annotational comments are also highly recommended. Like everything else in XML, comments must appear in a very specific format, for example "<!-- Enter comment here -->".

- Element definitions may be shown in collapsed format for convenience if you are reading a large XML file. Simply press the minus symbol to the left of the element's opening tag. This will allow you to compress chunks of XML data so that they remain hidden from view. To expand these again later, press the plus symbol beside the opening tag.

- Note that you must adhere rigidly to the XML document rules. For example, if you forget a closing tag, the entire XML document may be rendered unreadable to the software. Advanced software developer tools known as IDEs (Integrated Development Environments) will actually highlight incorrect XML syntax. Regardless, the golden rule is to build up your syntax gradually and carefully.

BUILDING BLOCKS

The wellhead profile above was created using cylinder objects only. Naturally Flexcom supports many more object types which you can use to create your customised model. Some of the more popular ones are summarised in the following table.

Object	Description	Attributes	Sample Syntax
Box	Rectangular box	Name (text), Centre (float, 3 values), Height (float), Width (float), Breadth (float), Colour (text)	<Box Centre="0 0 0" Height="7.1" Width="53.8" Breadth="10.6" Colour="Red"/>
Cylinder	Cylindrical surface	Name (text), Centre (float, 3 values), Axis (float, 3 values). Height (float), OuterDiameter (float), InnerDiameter (float), Segments (integer), Colour (text)	<Cylinder Name="Circle" Axis="1 0 0" Centre="0.02 0 0" Height="0.005" OuterDiameter="0.7" InnerDiameter="0.6" Segments="16"/>
Quad	Quadrilateral surface defined by 3 points in space	Name (text), Positions (3 values, repeated 3 times), Colour (text)	<Quad Colour="Black" Positions="1532.5 2273.9 -49.1,1535 2273.9 -49.1, 1535 2281.9 -49.1"/>

Object	Description	Attributes	Sample Syntax
Line	Straight line between 2 points	Name (text), Positions (3 values, repeated 2 times), Colour (text)	<Line Positions="1538.3 2274 -20.35, 1547.3 2274 -20.35" />
LineMesh	Mesh of straight lines using any number of points in space	Name (text), Positions (3 values, repeated as many times as required), Indices (2 values, repeated as many times as required), Colour (text) Each pair of indices creates a straight line between the relevant positions. The index refers to the list order in which the points were defined (note: starting index is 0 rather than 1).	<LineMesh Positions="1527.2 2273.9 -25 1529 2273.9 -25 1527.2 2273.9 -19.0 1529 2273.9 -19.05 1527.2 2273.9 -13.1 1529 2273.9 -13.1" Indices="0 1, 2 3, 4 5, 1 5"/>
PolyMesh	Mesh of triangular panels, each defined using any 3 points in space	Name (text), Positions (3 values, repeated as many times as required), Indices (3 values, repeated as many times as required), Colour (text)	<PolyMesh Name="Funnel" Colour="White" Positions="-16 13.75 -11

Object	Description	Attributes	Sample Syntax
		Each set of 3 indices creates a triangular panel between the relevant positions. The index refers to the list order in which the points were defined (note: starting index is 0 rather than 1). Depending on the order of the indices in the triangular definition, the panel can be either forward or backward facing, based on the direction of the normal vector, and this can affect how the panels appear in the Model View. Prior to the definition of panels, you can nominate a colour which is used for all the subsequent panels, up to the next colour statement.	16 2.75 -5.5 16 2.75 5.5 -16 13.75 11 -16 - 8.25 -11 16 - 13.75 -5.5 -16 - 8.25 11 16 - 13.75 5.5" Indices="0 1 2, 0 2 3, 4 5 1, 4 1 0, 6 7 5, 6 5 4, 2 7 6, 2 6 3, 1 5 2, 2 5 7"/>

Other possible object types include Fan, Strip, Polygon, RotationSurface, SweptSurface etc. You can also create named template objects which you can reuse multiple times (e.g. the same object could appear several times in your model view, with different dimensions and/or locations). Contact us if you are interested in further details - we can supply and describe the XSD schema which governs the XML file contents via the application of syntactical constraints.

IMPORTING OBJECT PROFILES FROM OTHER SOFTWARE

The PolyMesh command is highly generic and very powerful - this sample helicopter profile was built using PolyMesh. If you have access to graphical building software, they typically allow export of model data in text style format (e.g. OBJ files). These files contain coordinate and connectivity data which can be readily converted it into the PolyMesh format accepted by Flexcom, including intermediary manipulating steps via Excel or NotePad++ if necessary. If you find yourself performing this operation repeatedly over time, it might be useful to create a script or macro to help automate the process. We plan to incorporate an OBJ file importer tool into a future version of Flexcom.

MODEL BUILDING SUGGESTIONS

- Make sure your profile file has an .xml file extension. Object profile files in Flexcom have the file extension .fcvpx (FlexCom Vessel Profile eXtended) by default. However this file extension will not be recognised by web browsers or text editing software, so you should ensure to give your file an .xml file extension instead.
- Open your XML file in a text editor such as Notepad++, this will neatly colour code the XML syntax for ease of reading and understanding. If you have access to a software developer tool this will be even better, as it will highlight any errors in your XML syntax
- Build up your syntax gradually, taking care to adhere to the XML rules, and ensuring the integrity of your XML document.
- Link your XML file to a Flexcom model, via the `*BODY,INTEGRATED` or `*VESSEL,INTEGRATED` keywords. This will allow you to view the work-in-progress object in the Model View, giving you instantaneous visual feedback following every addition.

Undeformed Versus Initial Positions

Another innovative facility offered by Flexcom is a capability for the specification of undeformed element orientations. In most applications, the geometry you define represents the structure undeformed or stress-free configuration. However, this need not necessarily be the case, and the position defined by the specified nodal coordinates may represent the configuration of a deformed structure. In this general case the definition of the stress free orientation and the definition of the nodal coordinates are completely independent, with the stress free orientation being defined on an element-by-element basis when specifying the structure connectivity. This facility offers considerable flexibility, but is of interest only to a limited number of users.

When you define the connectivity of an element, in the majority of cases you only specify the element number and its start and end nodes. However, you may optionally define an additional six inputs, indicating to Flexcom the stress free orientation of the element. This section explains the significance of these inputs, and illustrates the circumstances when you might be required to specify such data. The inputs are entitled V1, V2, V3, W1, W2 and W3. Together these inputs define two vectors V and W, which in their turn define the undeformed orientation of the element you are inputting. To understand the significance of this, a brief overview of the Flexcom coordinate axis systems is provided.

Further information on this topic is contained in the following sections:

- Flexcom Coordinate Axes
- Structures without Cables
- Structures with Cables
- Conclusion

RELEVANT KEYWORDS

- If you have constructed your model using lines, then the stress free orientation of any line may be explicitly defined by specifying the V1, V2, V3, W1, W2 and W3 entries under the `*LINES` keyword.

- If you have built your model using an older approach (i.e. via the old `*NODE`, `*ELEMENT` and `*CABLE` keywords), then the stress free orientation of any element may be explicitly defined by specifying the V1, V2, V3, W1, W2 and W3 entries under the `*ELEMENT` keyword. When the undeformed orientation of an element is explicitly defined and that element is then identified as the Master Element in generating further elements, the undeformed orientation of the master element becomes by default the undeformed orientation of the generated elements.

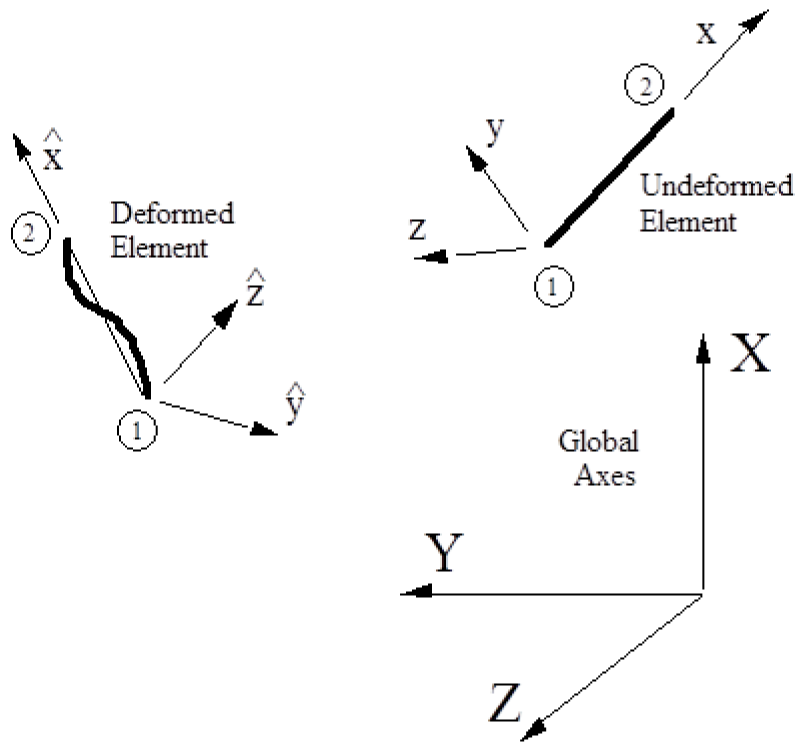
If you would like to see an example of how the stress free orientation of several lines are explicitly defined, refer to H04 - Pipe Laying.

Flexcom Coordinate Axes

OVERVIEW

There are three coordinate axis systems in Flexcom, termed the global axes, the local undeformed axes, and the convected axes. Of these, the global axes are the same for all elements of the finite element model. The others are individually defined for each element.

The three coordinate axes are illustrated schematically below.



Global, Local Undeformed and Convected Coordinate Systems

GLOBAL AXES

The global axes (XYZ) are the axes in which the total system equations of motion are assembled and solved, and in which the structure motions and reactions at restrained nodes are output.

CONVECTED AXES

Each element has associated with it a convected ($\hat{x}\,\hat{y}\,\hat{z}$) coordinate axis system. The convected axes displace and rotate with the element - these are the axes in which the instantaneous equations of motion at a particular solution time are written for the element. The definition of the convected axes, and the transformation of the equations of motion from the convected to the global axes, is one of the cornerstones of the Flexcom kinematics formulation for handling finite three-dimensional rotations. This is described in many publications by <%COMPANYNAME%> personnel (for example, O'Brien et al., 2002 and O'Brien et al., 2003), and the details are not elaborated further here. Copies of the two papers referenced are available on request from <%COMPANYNAME%>.

LOCAL UNDEFORMED AXES

Each element also has an associated local undeformed (xyz) axis system. The local undeformed axes define the orientation of each element when that element is undeformed or stress free. Restoring forces are set up in an element when it deforms relative to its local undeformed axes. The discussion of local undeformed axes that follows is in two parts. The next section, entitled Structures without Cables, discusses these axes in the case of structures defined without invoking the cable pre-static step. The presence of cables complicates the discussion of undeformed configurations somewhat, so a discussion of local undeformed axes in the case of structures with cables is postponed until the following section, entitled Structures with Cables.

Structures without Cables

Two definitions will help to clarify some issues here. The initial position of an element is the position defined by the user-specified nodal coordinates. The element undeformed orientation is the orientation in which there are no stresses or restoring forces in the element. The undeformed orientation is also termed the stress free configuration. Naturally, the local undeformed axes for an element define the undeformed orientation.

In the great majority of analyses, particularly in the case of structures defined without cables (the subject of this section), the initial position and the undeformed orientation are the same.

When that is the case, you are not required to specify the orientation of the undeformed axes for the element. Specifically, you should not specify the vectors V and W when defining elements directly. As mentioned already and discussed in detail below, the vectors V and W define the undeformed orientation for an element. This definition is unnecessary when the undeformed orientation coincides with the initial position of the element. So when you do not explicitly define V and W, this is your way of informing Flexcom that initial and undeformed configurations are coincident.

Further information on this topic is contained in the following sections:

- Default Local Undeformed Axes
- When Initial is not Undeformed
- Sample Application

Like all three-dimensional finite element software, Flexcom of course needs a local undeformed axis system for every element. If you don’t explicitly define it, Flexcom uses a default algorithm to calculate the vectors of the local undeformed triad. The line joining the two nodes on an element defines the undeformed X-axis and the local Y and Z axes are chosen to give a right-handed system.

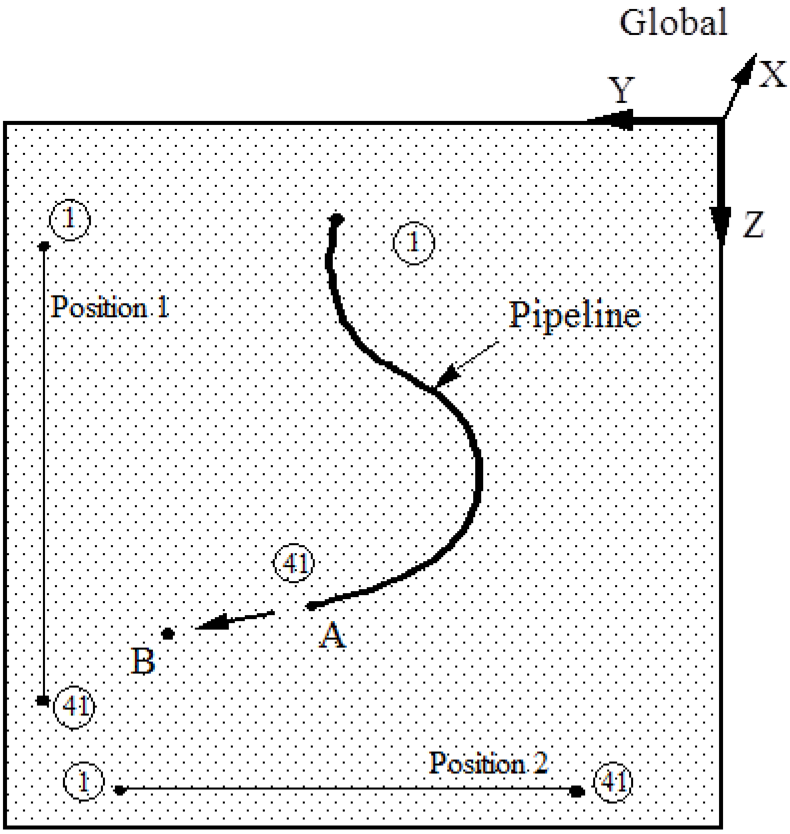
If you want to know what x, y and z axes Flexcom has calculated for a particular element, you will find the data you need in the program output file (jobname.out). In practice, though, this information is of very limited interest. When you are analysing a structure whose initial position defines its stress free or undeformed orientation, it is sufficient to know that the program uses a consistent scheme to calculate local undeformed axes for each element, and thereafter automatically (and correctly) evaluates the restoring forces and stresses caused by deformations relative to these axes.

One of the more innovative features of Flexcom is that you can define an initial position which is not stress free, and the program will automatically calculate the initial stress distribution, as well of course as additional or further stresses caused by motions from or about this initial position.

What happens in this case is that the specification of the initial position and the undeformed configuration become completely separate and independent. The initial position is described via the nodal coordinates in the usual way. The undeformed configuration is defined by using the components of the V and W vectors to define the undeformed orientation of elements.

Note the use of the term orientation in that last sentence, and indeed throughout earlier sections. The actual location (relative, say, to the global coordinate system) of the undeformed elements is not the main concern in defining the stress free structure configuration. It is instead the orientation of these elements, particularly relative to one another, which is important. This is best illustrated by means of an example - let's consider a sample pipeline model.

Consider the situation below, which shows a plan view of a pipeline lying on the seabed (which for convenience is assumed here to be smooth). Suppose the initial position shown by the heavy line can be defined in terms of the nodal coordinates of, say, Nodes 1 to 41, but that it is now required to examine the effect on the pipeline of pulling Node 41 from Point A to Point B. How might this analysis be performed in Flexcom?

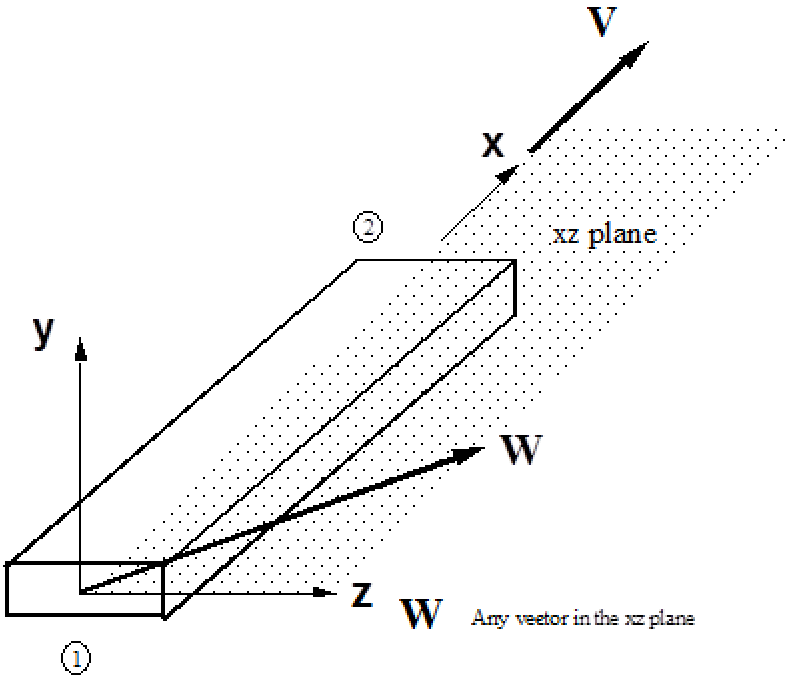


When Initial is Not Undeformed

Specification of the nodal coordinates (presumed known) and finite element connectivity is straightforward. Obviously though the initial position defined by these coordinates cannot represent a stress free configuration. The pipeline is stress free when it is stretched out in a straight line to its full length, and this is what Flexcom needs to be told. Where the pipe actually is in space when stretched out is of no consequence. For example, either of the positions identified as Position 1 or Position 2 above could be considered the stress free position of the pipeline. Motion of the pipeline from either to the initial position of the analysis induces the same stresses. Conversely, motion from Position 1 to Position 2 or vice versa induces no stresses in the pipe, since so-called ‘rigid body motions’ do not result in stresses. What Positions 1 and 2 have in common is that in both the elements all have the same relative orientation, and this is what is important. This is the reason why Flexcom requires you to specify vectors defining the stress free element orientation, rather than coordinates defining stress free position.

Obviously three vectors make up the local undeformed axes, but knowledge of two of them is sufficient because the three are orthogonal. This is why Flexcom requires only the specification of the components of two vectors, V and W. The V vector defines the x axis, that is, it is directed from the first node of the element to the second node, in the undeformed position. W is then any vector in the xz plane (other than V obviously). This definition of V and W is illustrated below. Note that neither V nor W is required to be a unit vector, nor are they required to be orthogonal (though they frequently are). While W can be any vector is the xz plane, a vector directed along the z axis is often a natural choice.

Consider again the sample pipeline model. For Position 1, V could be defined as (0, 0, 1) (directed along global Z) and for Position 2 as (0, -1, 0) (directed along negative global Y). For W, suitable choices for Position 1 could be (-1, 0, 0) and for Position 2 (0, 0, 1). In actual fact, in light of the above, the actual choices for V and W are nearly immaterial. The important thing is that all elements have the same undeformed orientation.



Element Undeformed Orientation Definition

For completeness the following is a summary of the actual mechanics of specifying the data for the analysis of the sample pipeline model. It is assumed that the coordinates (in the initial stressed or deformed position) of Nodes 1 to 41 are known, and can be input in the usual way by defining these nodes directly.

Specification of the finite element mesh connectivity would be relatively straight forward in the absence of the necessity to input the components of V and W. You would simply define Element 1 as joining Nodes 1 and 2, and then generate the remainder of the mesh using the element generate facility. However the process is not overly complicated by the need to input V and W. The actual procedure is that you specify the connectivity of Element 1 and also the components of V and W, using either of the specifications of the previous section (or indeed any other similarly valid scheme). You then generate the remaining elements as before. When the undeformed orientation of an element is explicitly defined and that element is then identified as the Master Element in generating further elements, the undeformed orientation of the master element becomes by default the undeformed orientation of the generated elements.

In fact, this is the only way to specify the undeformed orientation of generated elements. This feature is in reality a bonus, because what you are usually trying to do when specifying undeformed orientations is to ensure all elements have the same orientation - the orientation itself is frequently (though not always) immaterial.

There is one final point to make about the nodal coordinates for the initial position of the sample pipeline model. Obviously the configuration in the plot represents a position in which the pipe restoring forces or stresses are in equilibrium with the applied loading and boundary conditions. The likelihood is that in general the coordinates which give this exact equilibrium position will not be known. It is probably possible to make a reasonable guess or approximation to the actual equilibrium position, but that is all it is – an approximation.

In fact, whenever you perform an analysis in which the initial position is not undeformed or stress free, Flexcom assumes the nodal coordinates specified are approximate. In such an analysis Flexcom begins by finding the exact equilibrium position and the corresponding initial stresses. The approximate nodal coordinates do not even have to be very close to the actual equilibrium position. The closer they are, the fewer iterations that will be required to find the exact equilibrium position, but this rate of convergence is the only thing that is affected by the nodal coordinates input.

Structures with Cables

This section discusses the specification of undeformed orientations in the case of structures that include cables. Although it was noted in an earlier section that this complicated issues slightly, it should be emphasised that in the overwhelming majority of analyses where cables are used, the undeformed orientation of elements is again of no consequence, and the specification of the components of the V and W vectors is unnecessary.

This section begins with a discussion of why the presence of cables should complicate matters at all. It then proceeds to examine in some detail the two situations where the presence of cables means consideration of undeformed orientations is necessary and indeed essential. For the record, the two situations are (i) where a structure combines elements which are defined on cables with elements which are not on cables, and (ii) when specifying rotational boundary conditions in a model containing cables.

Note that the comments in this section apply equally regardless of whether you defined the cables in a structure explicitly yourself, or Flexcom automatically generated them when you defined a structure with lines. As noted previously in the section ‘Lines’, Flexcom automatically determines whether a line should be modelled using a straight section or a cable based on the length of the line and the straight line distance between its start and end locations. So if you know Flexcom is going to use a cable or cables, you may occasionally need to consider the issues discussed here.

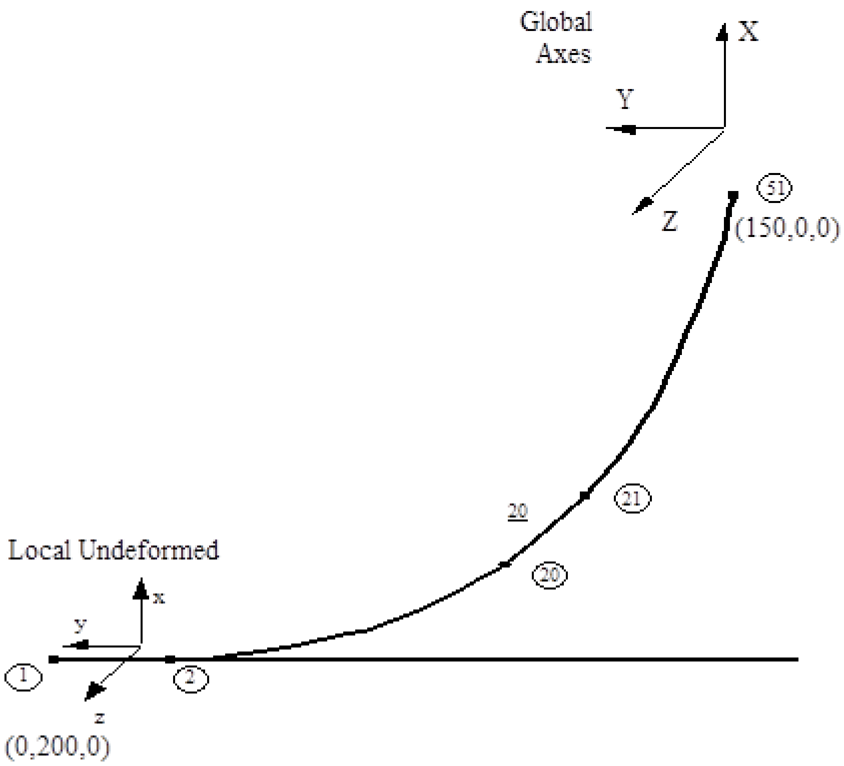
Further information on this topic is contained in the following sections:

- Why Cables Introduce Complications
- Mixing Cable and Rigid Elements
- Rotational Boundary Conditions and Cables

The underlying principles are again best illustrated by means of an example. Consider the free hanging catenary below. In specifying the finite element geometry data for this structure you would normally:

- Specify the coordinates of two nodes, denoted 1 and 51
- Define one cable suspended between these two nodes
- Specify the connectivity of one element, typically Element 1 from Node 1 to Node 2
- Generate the remaining elements using 1 as the master element. You would not normally define the undeformed orientation of Element 1, or consequently, of any of the other elements of the finite element mesh.

Flexcom still requires local undeformed axes for each element of the model, but now is at something of a disadvantage in choosing them. Consider Element 20 of the model, from Node 20 to 21. Default Local Undeformed Axes outlined how Flexcom begins defining default axes by joining the two nodes on an element to obtain the local x axis. The program cannot do that here. The locations of Nodes 20 and 21 are initially unknown, indeed they form part of the solution. So when an element is defined as being on a cable and the user does not explicitly define the undeformed element orientation, Flexcom cannot use the default algorithm used in the case of elements which are not on cables to find the local undeformed axes.



Local Undeformed Axes on a Cable

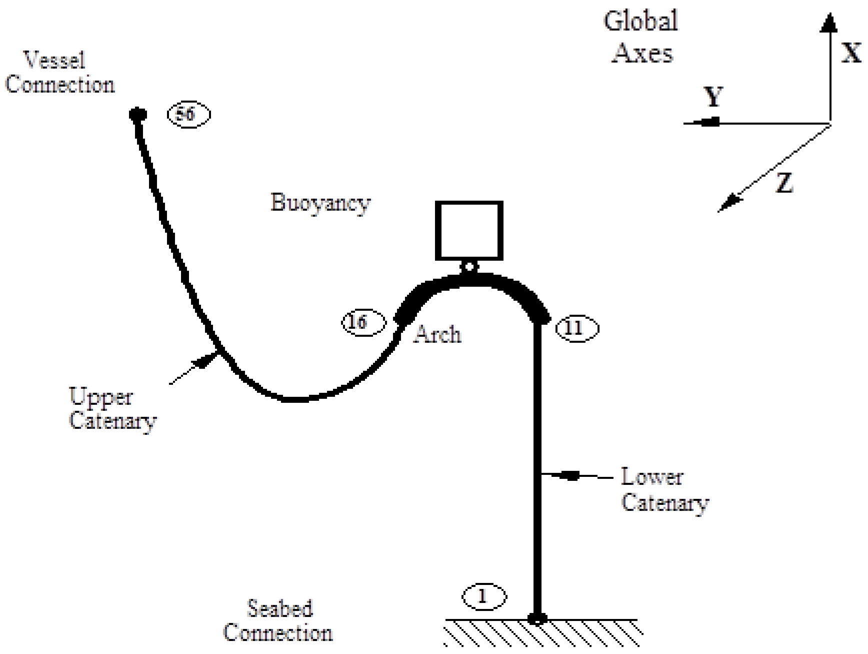
Flexcom must use instead an algorithm or strategy that does not involve knowledge of the element nodal coordinates. The actual strategy used is this: the local x axis is assumed to be coincident with the global X axis, and the plane formed by the local x and y axes is the plane of the cable itself. Where the cable is defined as initially in the global XY plane, which is frequently the case, the local undeformed axes are coincident with the global axes.

This is the case in the example catenary above. The Flexcom default local undeformed axes are illustrated for Element 1. They are identical for all other elements. Two observations from Default Local Undeformed Axes are appropriate here also. Firstly, if you want to know the orientation calculated by Flexcom for the local undeformed axes of an element defined on a cable, the information you need is in the output file jobname.out. But secondly, and perhaps more importantly, this information is of no particular significance. Looking again at the catenary above, it is clear that it is of little consequence how the undeformed axes of the individual elements are aligned. The important point, as before, is that the undeformed orientation of all elements is the same (the catenary is undeformed when stretched out in a straight line). Flexcom ensures this is the case by default, and will automatically calculate the restoring forces caused by subsequent deformations from this configuration.

One situation where it is necessary to consider carefully the undeformed orientation of cables is where a structure combines elements that are defined on a cable or cables (we call these cable elements for convenience), and elements that are not defined on a cable. For convenience, we term the latter rigid elements, although this is certainly not intended to suggest they cannot be, for example, sections of flexible riser.

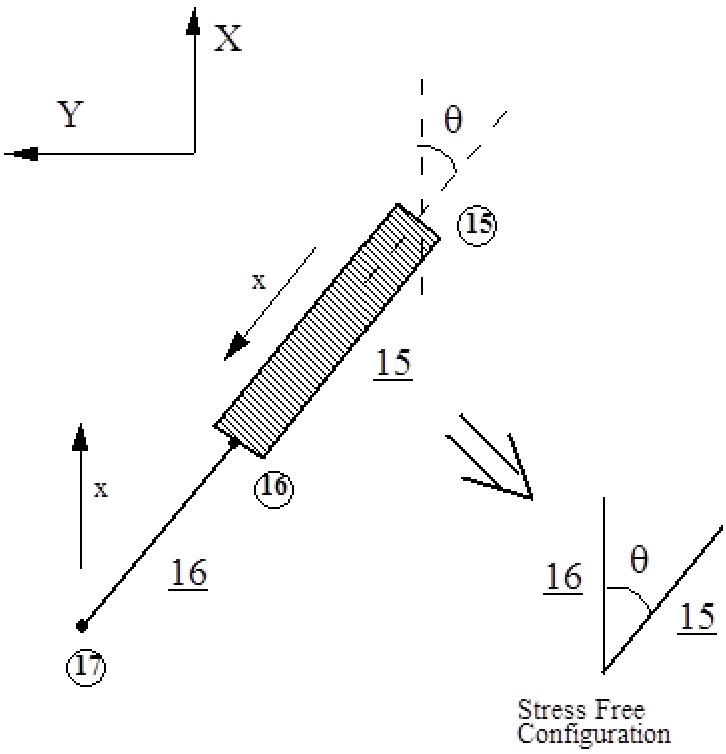
An example is again useful in illustrating the principles involved. The diagram below shows a certain type of Steep-S Flexible Riser. A lower catenary section extends from the seabed to a rigid arch or 'gooseneck'. A longer upper catenary extends from the gooseneck to the vessel connection point. A buoyancy module is connected to the arch by means of a hinge or flex joint, but the modelling of this part of the structure is of no consequence to the discussion to follow and so is omitted for clarity.

In developing a Flexcom model of this system, a cable would normally be used only for the upper catenary section. The lower catenary would typically be specified as initially vertical. The gooseneck would be modelled using, say, 5 elements with coordinates directly specified (these elements are undeformed in the curved shape below). So the model would contain one cable only between, say, Nodes 16 and 56. (This would be the case whether or not you used lines to define this model, an application for which they would be ideally suited, or you defined the geometry yourself explicitly).



Steep-S Flexible Riser

It is important to understand what happens if you do not explicitly define the undeformed orientation of the elements on this cable. The following diagram shows the situation at Node 16. The local undeformed axes of Element 15 are evaluated using the program default algorithm based on the nodal coordinates. The local x axis only is shown for convenience, pointing from the first node of the element (15) to the second (16). On the other hand, the program default algorithm for cable elements defines the local undeformed axes for Element 16. The local x axis in this case is vertical. What this means is that the stress free structure configuration is assumed to have a bend or kink at Node 16, which will result in large restoring forces being induced as the structure takes up its static configuration. This is obviously incorrect.



Situation at Node 16

The answer lies in associating the undeformed orientation of Element 15 with all of the Elements 16 to 55. This will mean both that the undeformed structure will not have a kink at Node 16, and equally that the undeformed orientation of the cable of the upper catenary will be with the riser stretched out in a straight line as required. With this in mind, the following are the steps for defining a typical finite element geometry for the structure. Nodes 1, 11 to 16, and 56 are defined explicitly in terms of coordinates. Nodes are generated between 1 and 11 using the (straight line) node generate facility. A cable is defined between Nodes 16 and 56, and intermediate nodes are distributed along this cable using the (cable) node generate facility.

The final step is the specification of element connectivity and undeformed orientation (where required). Elements 1 to 15 pose no problems. Element 1 is defined as joining Node 1 to Node 2, without specifying V or W. You then generate the connectivity of Elements 2 to 15 by identifying Element 1 as the Master Element and inputting 15 as the Number of Elements.

The most important step comes next, the specification of the connectivity of Element 16. The First Node and Last Node entries are straightforward, being 16 and 17 respectively. In this case however it is important to specify V and W, for reasons just discussed. V as previously outlined is a vector along the local x axis. For Element 15 of the model, this will be (X16 - X15, Y16 - Y15, 0), where X15 and X16 are respectively the coordinates in the global X direction of Nodes 15 and 16, and Y15 and Y16 are the corresponding coordinates in the Y directions. These are the values which must be input for Element 16.

The specification of the components of W is slightly more complicated. It is based on knowing what Flexcom calculates as the W vector for Element 15. The following summarises the program operation without discussion, since the program reasoning is irrelevant to the matter in hand. For a system defined in the XY plane as shown in the Steep-S Flexible Riser figure above, the components in the X and Y directions of V and W are the same, while the component in the Z direction is equal to the length of the element.

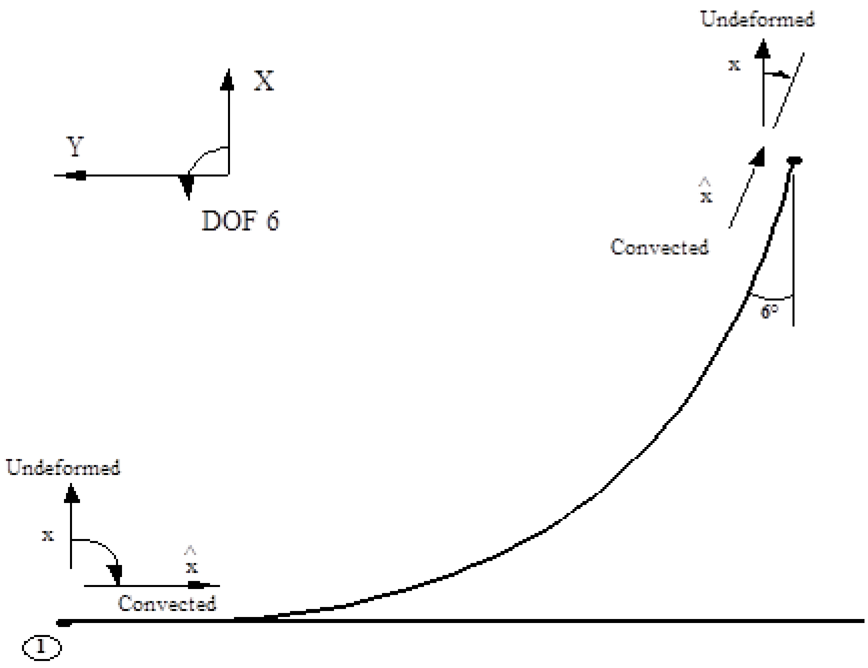
Suppose for example that in our example, Node 15 is at (95,40,0) while Node 16 is at (92,42,0), giving an element of length 3.6m. Flexcom would calculate for Element 15 a V vector with components in the global axes of (-3,2,0), while the corresponding W vector would have components (-3,2,3.6).

It is worth noting that while calculating V and W for an element such as Element 15 might appear complicated, Flexcom does output its default vectors to the jobname.out output file. It is always possible (indeed recommended) to check this file to ensure you have indeed input the same V and W components for Element 16 as Flexcom calculates for Element 15.

The definition of the finite element mesh for the steep-S system is completed by generating the connectivity of Elements 17 to 55 using the generate elements facility. Element 16 is identified as the Master Element and the Number of Elements input is 30. This automatically associates the orientation of Element 16 with the generated elements, which is exactly what is required. This completes the specification of the steep-S finite element model, other than the modelling of the buoyancy tank/hinge assembly, which is not discussed here.

The section concludes with a discussion of one last area where it is necessary to take account of the undeformed orientation of cable elements. This is in the specification of rotational boundary conditions, that is, boundary conditions in any or all of degrees of freedom (DOFs) 4, 5 or 6. Once again the principles involved are best illustrated by means of an example. This is based again on the free hanging catenary of Local Undeformed Axes on a Cable figure; for simplicity it is assumed that a static analysis of the riser in the 2D XY plane is being performed. Suppose it is required to apply boundary condition which will a) prevent rotations about the Z axis (rotations in DOF 6) at Node 1, and b) restrain the riser angle to the vertical to be 6° at Node 51. What values should be specified as the associated Displacement terms for these boundary conditions to apply the desired restraints? It is assumed by that the orientation of the local undeformed axes for the elements of this model is calculated by Flexcom using the program’s default algorithm.

Specifying constant rotational boundary conditions at a node is effectively the same as specifying the orientation of the convected axes for the appropriate element. Because the example is 2D only, the situation here is quite straightforward in that the discussion need be concerned only with specifying the orientation of the appropriate $\hat{x}$ axes. Consider for example the boundary condition to be applied at Node 1. The restraint proposed is effectively the same as saying the $\hat{x}$ axis for Element 1 is to be directed along the seabed as shown below. The following diagram shows the orientation of the x (local undeformed) axis for Element 1, as calculated by Flexcom. From the diagram below it is clear that the rotation required to change the undeformed to the convected orientation is -90° (note the sign - a positive rotation in DOF 6 is in the direction from global X to global Y). So at Node 1 the DOF 6 boundary condition is -90°.



Specification of Rotational Boundary Conditions

A similar reasoning can be used to work out the boundary condition required at Node 51. The orientation of the $\hat{x}$ axis for the required restraint is shown above. The rotation required to align local undeformed with convected axes is -6° (again note the sign), so this is the value to be specified as the associated Displacement term for the boundary condition.

Conclusion

This concludes the discussion on Undeformed Versus Initial Positions. The following is a short summary:

- A local undeformed axis system is associated with each element of a Flexcom model.
- You have the option of using the components of two vectors V and W to specify the orientation of the local undeformed axes for a particular element.
- If you do not invoke this option, Flexcom calculates the undeformed orientation of elements using a default algorithm. This will be perfectly satisfactory in the vast majority of Flexcom analyses.

- The specification of the undeformed orientation of elements is most frequently used when you want to specify that the position of the structure defined by the nodal coordinates you have input is not stress free. This is a powerful program facility, but of interest only in a minority of analyses.

- When a structure includes a cable or cables, it is occasionally necessary to be aware of the undeformed orientation of elements on the cable(s), especially (i) where elements on a cable join directly to elements that are not defined on a cable and (ii) when specifying rotational boundary conditions.

1.9.2.2 Geometric Properties

Flexcom provides different formats for specifying structure geometric properties, although ultimately the information is the same and is used in the same way by the program. These are termed Flexible Riser, Rigid Riser and Mooring Line formats.

- The Flexible Riser format is so called because it is the format commonly used when defining flexible risers. Specifically, the data to be specified in this case comprises bending stiffnesses about two axes, torsional stiffness, axial stiffness, mass per unit length etc.

- When you use the Rigid Riser format on the other hand, you input internal and external diameters, Young’s modulus, shear modulus, mass density etc. – data for the analysis of a rigid riser (whether a top tensioned riser or an SCR) would normally be available in this format.

- The Mooring Line format is similar to the Flexible Riser format, but fewer input terms are required. It is geared towards cables and wires which typically have very low resistance to bending and torsion, hence the governing inputs are axial stiffness and mass per unit length.

It is important to stress that there is of course no requirement that the structure you are analysing with Flexcom be a flexible riser, rigid riser or mooring line. These terms are used simply because the formats are traditionally used for these types of structure.

- Stress Properties details the use of property data in calculating cross-section stresses.

- Coatings describes the coatings facility and how coatings affect element properties.

- Poisson’s Ratio describes how Poisson’s ratio effects are modelled.

- Tangent and Secant Stiffness describes the traditional and alternative approaches for modelling non-linear materials.
- Non-linear Material Force Term explains the theory behind the right-hand side force term which accompanies stiffness terms for non-linear material elements.
- Compression and Buckling discusses the critical Euler load, the maximum compressive load which a beam can sustain without buckling laterally.

Geometric Properties in Flexible Riser Format

OVERVIEW

When using the Flexible Riser format you specify geometric property data under four headings:

- Structural properties: there are three Material Models available, Linear Elastic, Non-linear Elastic, and Hysteretic Bending
- Mass and Polar Inertia per Unit Length
- Diameter Inputs: there are four external diameter inputs and one internal diameter input
- Buoyancy Formulations: the options are Default and Distributed

All of these are now discussed in more detail. Note that the choice of buoyancy formulation has significance for some of the external diameter inputs, so you may wish to read the article on Buoyancy Formulations before reading the section discussing Diameter Inputs.

RELEVANT KEYWORDS

- `*GEOMETRIC SETS` is used to assign geometric properties to element sets (see OPTION=FLEXIBLE).
- `*MOMENT-CURVATURE` is used to define moment-curvature curves for non-linear materials.
- `*FORCE-STRAIN` is used to define force-strain curves for non-linear materials.

- *TORQUE-TWIST is used to define torque-twist curves for non-linear materials.

Note also that the old *STRESS/STRAIN keyword has effectively been superseded by these new non-linear material definition keywords which explicitly distinguish between bending, axial and torsional stiffness.

If you would like to see an example of how structural properties are defined in the Flexible Riser format, refer to C02 - Multi-Line Flexible System.

Material Models

In the case of the Flexible Riser format, there are three material models available to you:

- Linear Elastic
- Non-linear Elastic
- Hysteretic Bending

The simplest material model is linear elastic. In this case, you specify a linear (single) value for each of the bending stiffnesses EI_{yy} and EI_{zz}, the torsional stiffness GJ, and the axial stiffness EA.

OVERVIEW

User Inputs

You may specify a non-linear relationship for any or all of the bending stiffnesses EI_{yy} and EI_{zz}, the torsional stiffness GJ, and the axial stiffness EA. To use this option you define the relationship between “generalised stress” and “generalised strain”. The significance of “generalised stress” and “generalised strain” depends on the stiffness being defined. For bending stiffness, stress is bending moment M and strain is curvature κ, so the input is an M-κ curve. For torsional stiffness, stress is torque and strain is torsional strain (twist / length); while for axial stiffness, stress is axial force and strain is axial strain (extension / length).

Note that you can combine linear and non-linear specifications for the same element set; so for example you might specify a non-linear bending response, while GJ and EA are linear (single-valued).

Symmetric and Asymmetric Bending Models

A further option is available to you if you define a non-linear bending stiffness relation. Generally speaking, you input the same non-linear relationship for EI_{yy} and EI_{zz}. In the program terminology, the Symmetric non-linear bending model involves the computation of a single bending stiffness EI at any solution time based on the total curvature κ, with both EI_{yy} and EI_{zz} being set equal to EI. This means that the element stiffness will be same in any situation where κ is the same, regardless of the values of the individual curvature components.

An option to model Asymmetric non-linear bending behaviour is also provided. In this case, EI_{yy} at any solution time is found from the curve you input for that stiffness term based on the instantaneous value of the corresponding local curvature term κ_y, and likewise EI_{zz} is found based on instantaneous κ_z. This is true even if you input the same curve name for EI_{yy} and EI_{zz}. This can mean that bending response can differ for the same loading depending on element orientation; an orientation where κ_y and κ_z are non-zero can give a different response to an orientation with either κ_y or κ_z equal to zero.

Curvature Slippage

It is possible to change material bending properties from linear to non-linear between successive analysis stages. Furthermore, the user-defined (non-linear) moment-curvature relationship used in a restart stage can be adjusted such that it is offset by moment-curvature values obtained from the preceding solution. Refer to Curvature Slippage for further information.

THEORY

Refer to Non-linear Material Force Term for further information on how the Flexcom solver accounts for non-linear materials in the equilibrium equations.

RELEVANT KEYWORDS

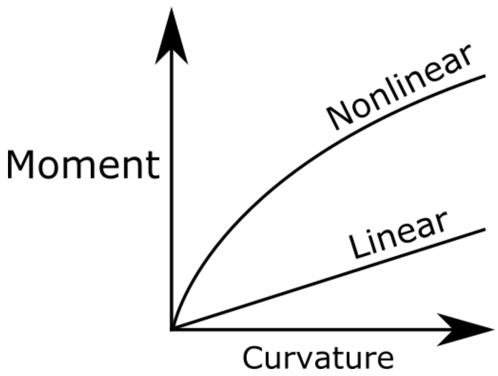
- `*MOMENT-CURVATURE` is used to define moment-curvature curves for non-linear materials.
- `*FORCE-STRAIN` is used to define force-strain curves for non-linear materials.
- `*TORQUE-TWIST` is used to define torque-twist curves for non-linear materials.
- `*NONLINEAR MODEL` is used to specify a modelling approach for non-linear materials.

THEORY

It is possible to change material bending properties from linear to non-linear between successive analysis stages. Furthermore, the user-defined (non-linear) moment-curvature relationship used in a restart stage can be adjusted such that it is offset by moment-curvature values obtained from the preceding solution. This modelling behaviour is termed *curvature slippage*.

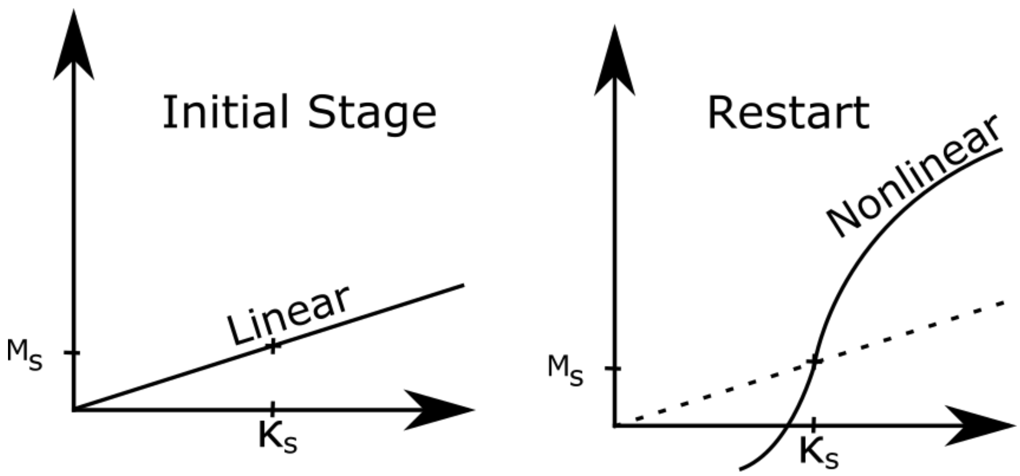
In some respects, curvature slippage is similar to Hysteretic Bending, but bending reversals do not cause the structure to alternate between pre-slip and post-slip conditions. Rather, a single slip is modelled, and there is no loop behaviour from that point onwards.

The following figure shows some sample linear and non-linear material properties, each of which define some relationship between bending moment and curvature. In the context of curvature slippage, the elastic/unpressurised state is represented by the linear relationship, while the friction/pressurised state is represented by the non-linear relationship.



Sample Moment-Curvature Relationships

The following figure shows some actual material properties used in a notional Flexcom analysis. The left image shows the initial linear moment-curvature relationship, while the right image shows the modified non-linear moment-curvature relationship which is used in a restart analysis. The intersection point of both lines corresponds to the state of slippage $\{K_s, M_s\}$ values.



Curvature Slippage Stages

USER INPUTS

Initial Static Analysis

Contrary to conventional Flexcom models, both linear and non-linear material properties are defined in the initial static stage. Specifically, the `*GEOMETRIC SETS` keyword should contain numerical inputs which represent the bending stiffness about the local y and z axes, and the name (or names) of non-linear moment-curvature relationships for these same degrees of freedom. Any non-linear relationships which are referenced should be defined under the `*MOMENT-CURVATURE` keyword. Finally, the `*NONLINEAR MODEL` keyword should be included, and the `EI` input set to `LINEAR`.

Restart Analysis

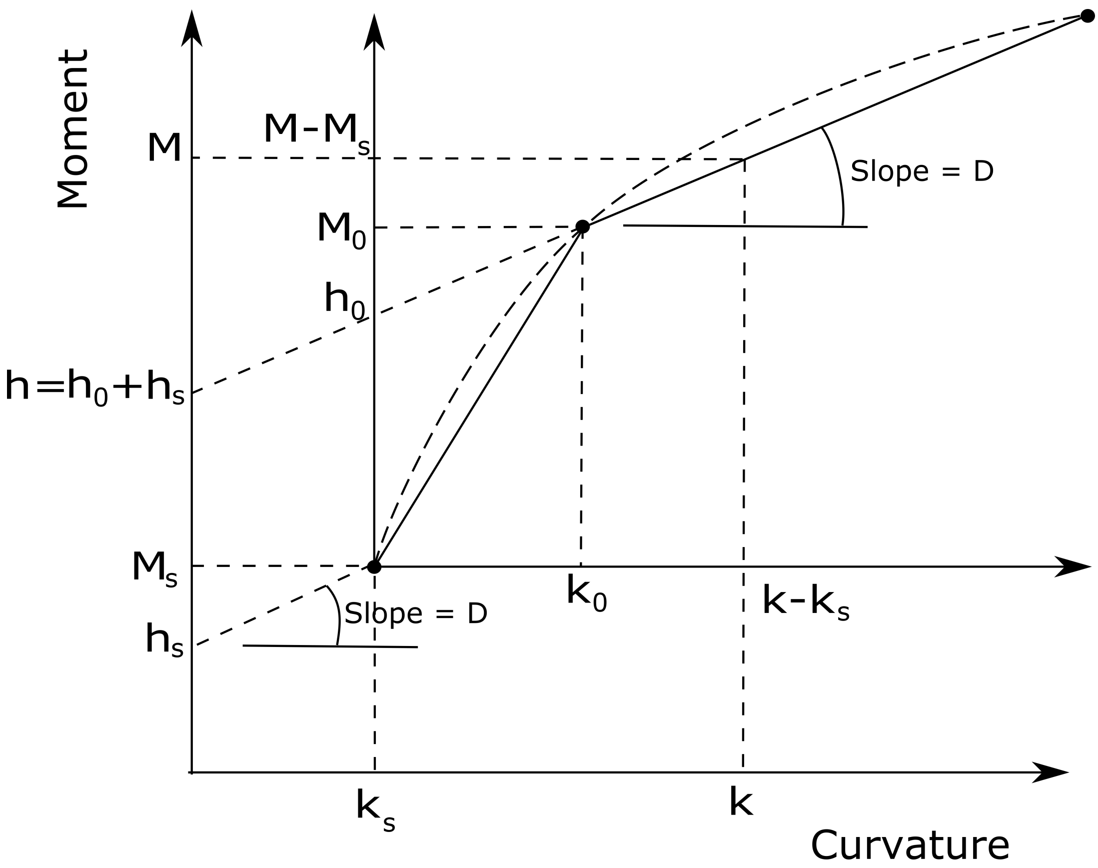
The *NONLINEAR MODEL keyword should be included, and the EI= input set to NONLINEAR. Additionally, the NONLINEAR= input under the same keyword should be set to CURVATURE_SLIPPAGE.

SYMMETRIC AND ASYMMETRIC BENDING MODELS

The following section provide some further theory on the distinction between the Symmetric and Asymmetric Bending Models, and its significance in the context of the curvature slippage feature.

Asymmetric Bending

For asymmetric bending with slippage, the stiffness, *D*, and Nonlinear Material Force, *h*, for each local *y* and *z* axis are calculated independently. For a given local axis, the determination of *D* and *h* is illustrated as follows.



Asymmetric Curvature Slippage

where *h* is calculated as:

$$h = h_0 + h_s \quad (1)$$

and

$$h_s = Dk_s - M_s \quad (2)$$

The stiffness, *D*, is calculated using either the tangent (default) or secant method (refer to Tangent and Secant Stiffness for further information). If the secant stiffness method is specified using the `*NONLINEAR MODEL` keyword, then h_0 will be zero and hence the non-linear material force will be equal to h_s.

Symmetric Bending

For the case of symmetric bending the calculation of *D* and *h* is described as follows.

Firstly, the resultant relative curvature, k_r, is determined:

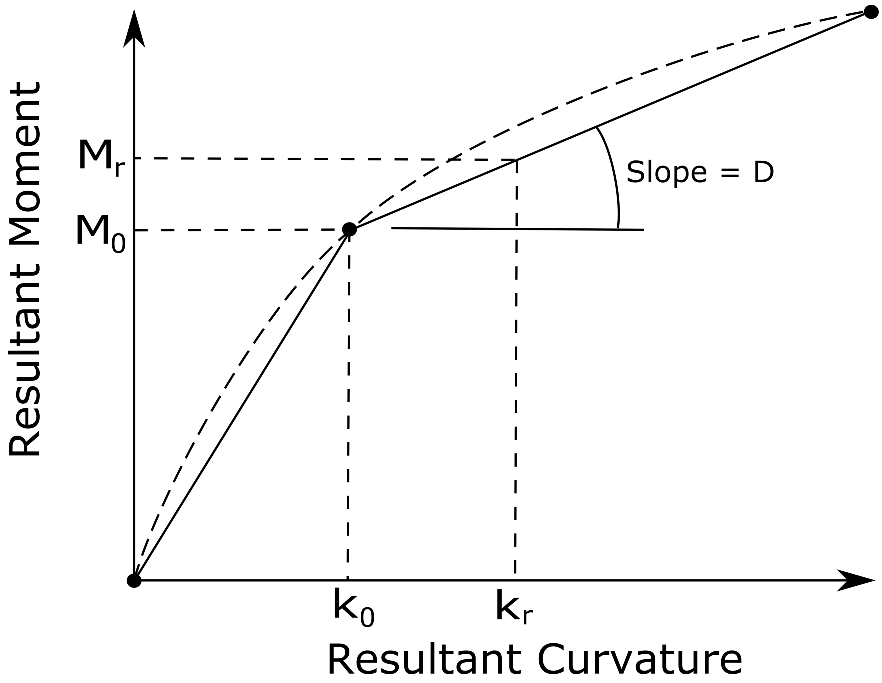
$$k_r = \sqrt{(k_y - k_{ys})^2 + (k_z - k_{zs})^2} \quad (3)$$

where:

- k_y is the instantaneous local *y* curvature
- k_z is the instantaneous local *z* curvature
- k_{ys} is the slipped local *y* curvature
- k_{zs} is the slipped local *z* curvature

A resultant relative bending moment, M_r, is then obtained from the non-linear moment-curvature relationship as follows:

$$M_r = M_0 + D(k_r - k_0) \quad (4)$$



Symmetric Curvature Slippage

The stiffness, D, is calculated using either the tangent (default) or secant method (refer to Tangent and Secant Stiffness for further information), depending on the option specified in the `*NONLINEAR MODEL` keyword.

Local *y* and *z* moments are determined by apportioning the resultant moment, M_r, by the ratio of the change in local curvature (relative to the slipped position), k-k_s, to the resultant relative curvature, k_r, as follows:

$$M_y = M_r \frac{k_y - k_{ys}}{k_r} + M_{ys} \qquad (5)$$

and

$$M_z = M_r \frac{k_z - k_{zs}}{k_r} + M_{zs} \qquad (6)$$

with the addition of M_{ys} and M_{zs} which are the local *y* and *z* slipped moments, respectively.

Once the local moments are determined, the local Non-linear Material Force Terms (h_y and h_z) are obtained as follows:

$$h_y = Dk_y - M_y \qquad (7)$$

and

$$h_z = Dk_z - M_z \qquad (8)$$

RELEVANT KEYWORDS

- `*GEOMETRIC SETS` is used to assign geometric properties to element sets.
- `*MOMENT-CURVATURE` is used to define moment-curvature curves for non-linear materials.
- `*NONLINEAR MODEL` is used to specify a modelling approach for non-linear materials.

OVERVIEW

Flexcom provides an option to specify non-linear bending behaviour with hysteresis. This is an important effect observed in flexible risers during normal operation. Internal pressure in the riser causes interlayer friction to build up between the tensile armour layers leading to a hysteretic stick-slip bending response.

Flexcom employs a sophisticated algorithm that computes bending hysteresis for all types of loading. The input data is a backbone curve comprising a series of piecewise-linear moment/curvature values, which is representative of the operating conditions. The input format is similar to the specification of a non-linear elastic bending stiffness relationship as described above. Flexcom modifies the hysteresis curve as required to initiate the step change in non-linear stiffness when the bending reverses.

The hysteresis option is by default included in both static and dynamic analyses. An option is available to suppress hysteresis in a static analysis, since bending hysteresis effects occur after a pipe becomes pressurised. When this option is invoked, a linear elastic bending stiffness is used for static analyses that precede the first dynamic analysis. The linear bending stiffness is assigned as the last slope of the moment/curvature backbone curve. The hysteresis bending stiffness is Symmetric by the definition above.

When conducting a global analysis of a flexible riser, a cautious approach is often adopted whereby only the relatively small elastic component of the pipe bending stiffness is included in the analysis. This comparatively small bending stiffness produces conservative over-estimates of the pipe bending curvature, which is normally acceptable provided the extreme and fatigue design criteria (limit states) are not exceeded. However, in recent years this simplified design approach has been shown to exceed the design criteria at the touchdown of deepwater catenary risers and in fatigue analysis involving wet armour wires.

When fully pressurised, the interlayer friction between the tensile armour layers causes the dynamic bending response of flexible risers to become highly damped. During dynamic excitation, the pressurised pipe bending stiffness is characterised by a stick-slip (high-low) effect as adjacent armour layers slide over one another. This stick-slip effect produces hysteresis loops during cyclic loading, unloading and load reversal.

Further information on this topic is contained in the following sections:

- Hysteresis Theory
- Flexcom Methodology
- Deepwater Catenary Example
- Recommendations

RELEVANT KEYWORDS

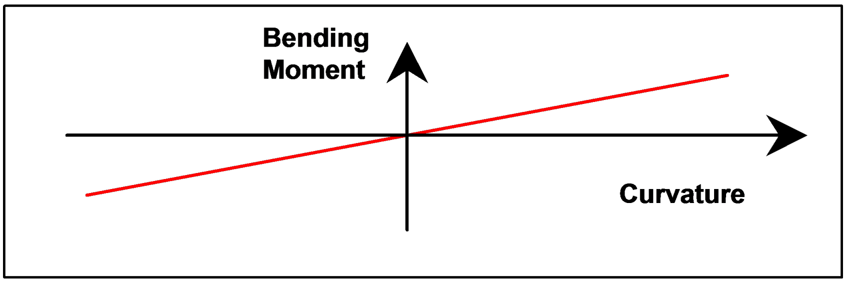
- `*GEOMETRIC SETS` is used to assign geometric properties to element sets.
- `*BENDING HYSTERESIS` is used to define hysteresis moment-curvature backbone curves for non-linear materials.
- `*NO HYSTERESIS` is used to suppress bending hysteresis effects.

The bending stiffness of a flexible riser differs largely for non-operating (depressurised) and operating (pressurised) conditions. The riser has a comparatively small bending stiffness when depressurised, for example during reeling before shipment and during subsequent installation at the field. The bending stiffness in this case is mainly dependent on the external sheath and, to a smaller extent, the internal sheath(s). The tensile armour layers in a depressurised riser provide little resistance to pipe bending due to free slip of each armour strip. A linear bending stiffness of between 10 kNm^2 to 100 kNm^2 is representative of depressurised flexible pipes.

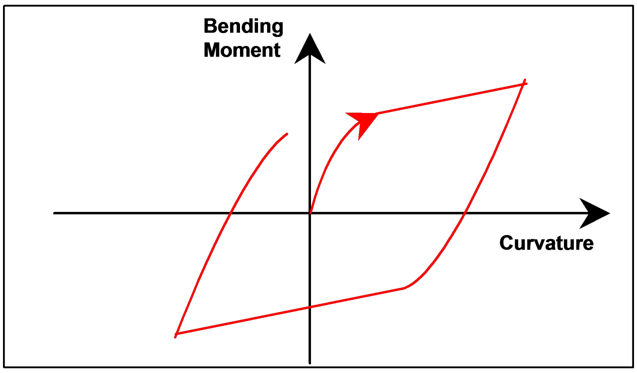
The bending stiffness of the depressurised riser is normally used in design for computing the pipe bending radius in extreme and fatigue wave loadings. The comparatively small bending stiffness produces conservative over-estimates of the pipe bending curvature, which is normally acceptable, provided the extreme and fatigue design criteria are not exceeded. In recent years, this simplified design approach has been shown to exceed the design criteria at the touchdown of deepwater catenary risers and in fatigue analysis involving wet armour wires.

The bending stiffness of a pressurised flexible riser is characterised by a complex non-linear bending stiffness relationship known as hysteresis. Pressurising a flexible riser produces large contact pressures between the various layers in the pipe construction. The resulting frictional resistance that develops during bending prevents free slip of the tensile armour strips. The friction increases the bending stiffness by two to three orders of magnitude for small changes of pipe bending curvature. Larger changes in the bending curvature overcome the initial frictional resistance and the pipe stiffness reduces (softens) towards the depressurised condition as the tensile armour begins to slip. The riser bending stiffness is generally divided between pre-slip and post-slip conditions. A bending reversal of the riser (as induced by the dynamic wave loading) causes the stiffness to return to the pre-slip condition.

The two below figures compare the conventional depressurised (linear) and pressurised (hysteresis) moment curvature relationships of a flexible riser.



Linear Moment Curvature Relationship



Hysteresis Moment Curvature Relationship

Hysteresis has two important effects that reduce the bending motion of a riser. Firstly, the comparatively large pre-slip bending stiffness is supported for small changes of curvature at each loading reversal (and at the commencement of bending motion). Secondly, the area within a closed (or almost closed) loop represents a loss of bending energy due to friction. This is a structural damping effect that is modelled directly by Flexcom without converting it to an equivalent form of viscous damping. These two effects significantly reduce the bending motions in a pressurised riser and can greatly assist in satisfying the design criteria for the extreme bending radius and fatigue life.

THEORY

Hysteresis requires two modifications to the general Flexcom analysis methodology. The first allows a changeover from depressurised to pressurised operating conditions. The second modification improves the convergence of the time-step iteration method as the stiffness undergoes large changes between pre- and post-slip conditions.

Depressurised and Pressurised States

Hysteresis generally applies to a pressurised flexible riser after it has been installed and become operational. Flexcom therefore allows the effects of hysteresis to be excluded from an initial series of static or quasi-static analyses (via the `*NO HYSTERESIS` keyword) so that the initial riser configuration is determined using the depressurised (linear elastic) bending stiffness. The stiffness for the static analysis when this option is selected equals the slope of the last segment in the moment-curvature input data described below. The final computed static configuration is then used as a relative reference from which to switch on the hysteresis stiffness.

The `*NO HYSTERESIS` keyword allows you to explicitly exclude hysteresis effects from any individual analysis (including dynamic, although this would represent an unusual request). Note that this input is not carried through to any subsequent restart analysis – it must be explicitly specified in every analysis stage in which it is warranted.

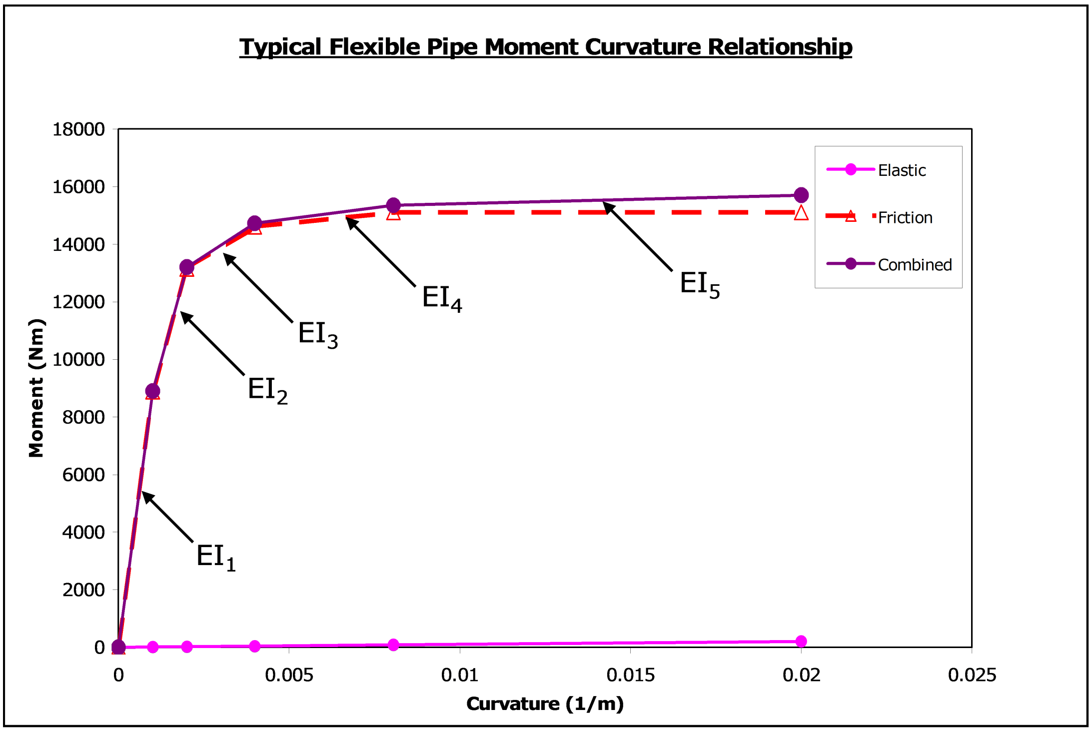
Pre-Hysteresis Analysis

Hysteresis can cause the stiffness terms to repeatedly toggle between high pre-slip and low post-slip values during iteration at a particular solution time, which can prevent convergence. The problem is circumvented by solving each time step twice, first with a pre-hysteresis step and then a hysteresis analysis. The pre-hysteresis analysis keeps the stiffness unchanged from the previous time step. The solution of the pre-hysteresis analysis is then used to determine an expected stiffness value based on the rules of hysteresis, that is, pre-slip, post-slip or reversal. The hysteresis analysis resumes the iteration and will decrease the expected stiffness if required, but will not increase an expected post-slip stiffness to a pre-slip value during the iteration. The two successive analyses carry a small computational overhead but ensure convergence of the iteration method at each solution time.

Backbone Curve

When invoking the hysteresis option in Flexcom, a moment-curvature relationship for the riser under consideration is required. This is specified in the form of a backbone curve which defines the moment experienced by the pipe for increasing values of curvature. The figure below shows an example of a typical backbone curve (and its piecewise linear components).

The elastic curve is based on the depressurised riser and is comparatively small. The friction curve is the main component of the pressurised condition. The large moments on this curve result from the inter-layer friction on the tensile amour layers. The friction curve levels off towards a zero stiffness (slope) for increasing curvatures. The friction curve is determined from test data or a local structural analysis of the riser cross section. The combined curve is the sum of the elastic and friction curves and provides the input data required by Flexcom.

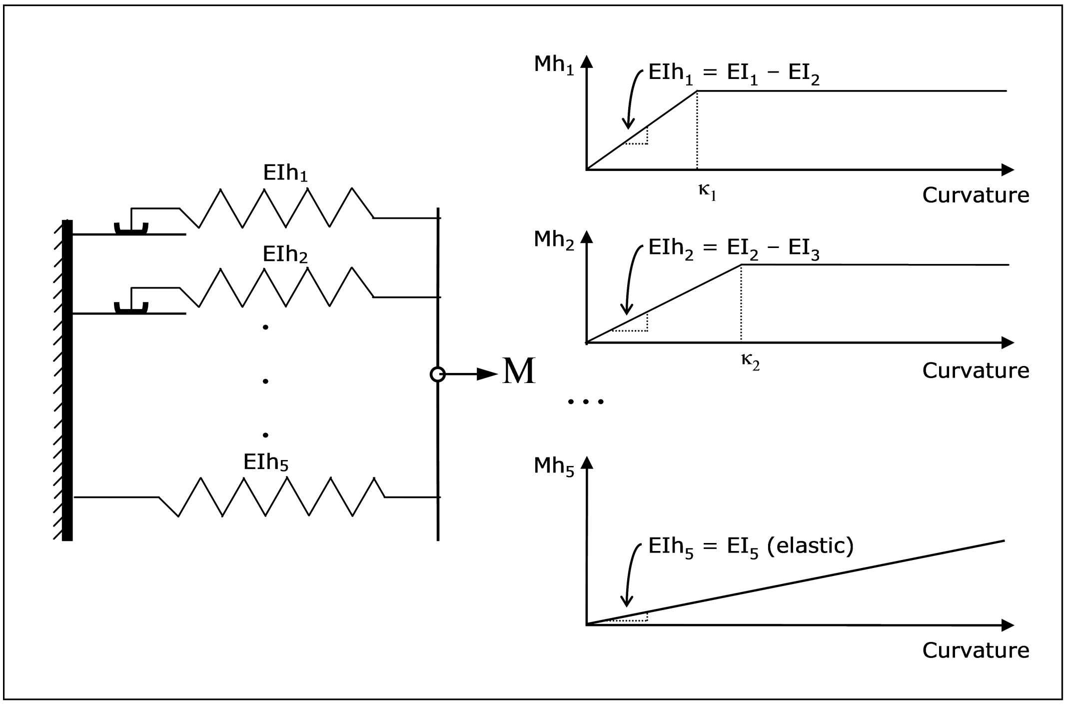


Hysteresis Backbone Curve

The Flexcom hysteresis analysis can handle any arbitrary number of input segments starting from bi-linear (two segments). A close approximation of a smooth backbone curve is usually achieved with approximately five segments as shown above.

Elastoplastic Model – Parallel Spring Analogy

An important part of the hysteresis analysis is the decomposition of the backbone curve. This is achieved using an analogous elastoplastic system of parallel springs as shown in the figure below. Each linear segment of the backbone curve is replaced by an elastoplastic stiffness component.



Assembly of Elastoplastic Stiffness Components

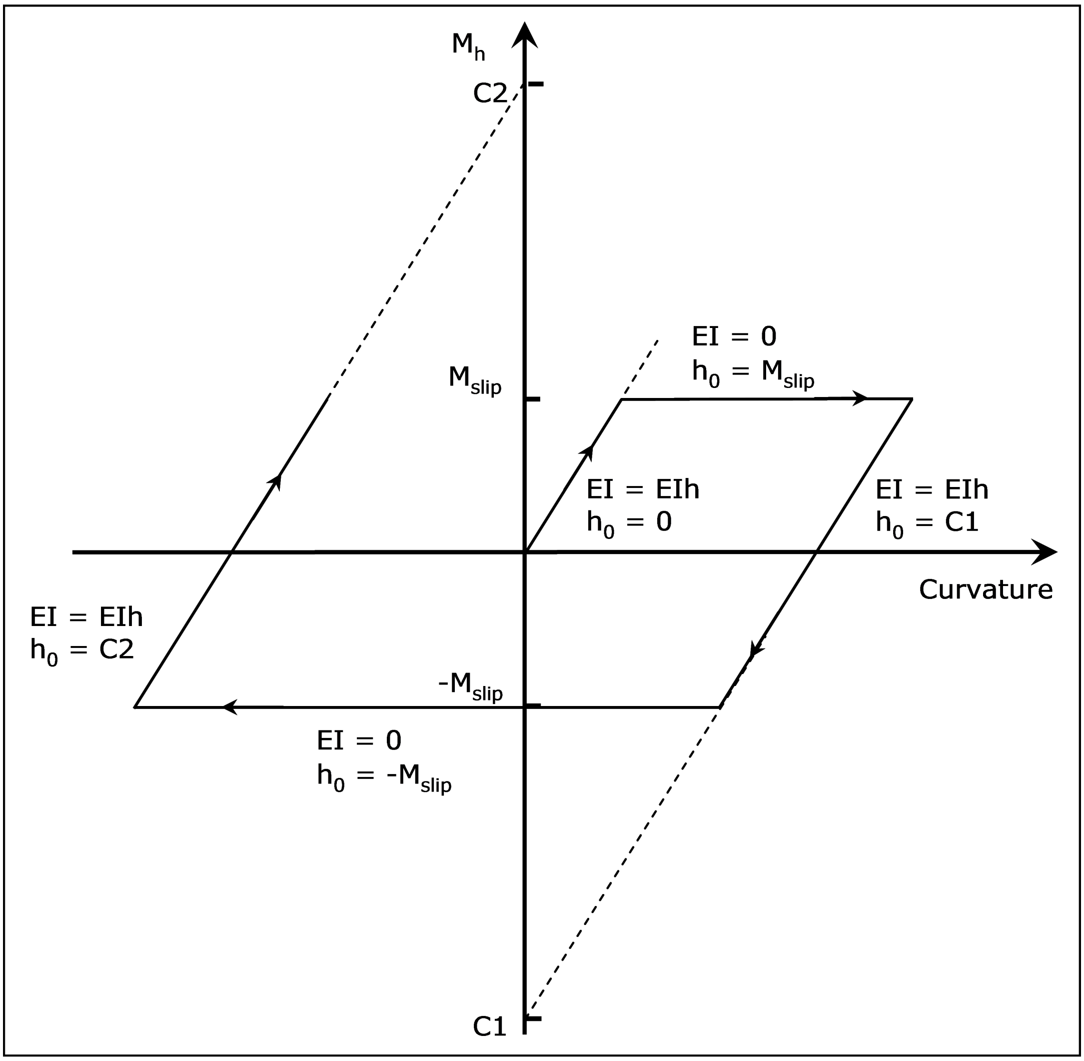
The pre-slip stiffness assigned to each spring is based on the difference in the stiffness of adjacent segments as shown in the above figure, e.g. $EIh_1 = EI_1 - EI_2$. The post-slip stiffness of each spring is zero, i.e. perfectly plastic. The slip curvature of each spring is equal to curvature at the end of the related segment on the backbone curve of the pipe moment. The first segment has the smallest slip curvature and the last (elastic) segment never slips. The pipe bending stiffness at the current time step is the sum of the elastoplastic components. The rules governing each elastoplastic component for two-dimensional bending are described in the next section, followed by their extension to three-dimensions.

Two-Dimensional Bending – Uniaxial

The curves on the right-side of the previous figure define the backbone curves of the elastoplastic stiffness components. The moment curvature relationship at each time step is derived from these backbone curves. The hysteresis relationship for an elastoplastic component is defined by:

$$M = EI\kappa + h \qquad (1)$$

where *M* denotes the bending moment, *EI* bending stiffness, κ bending curvature and *h* is the Nonlinear Material Force intercept with the moment axis as shown below:



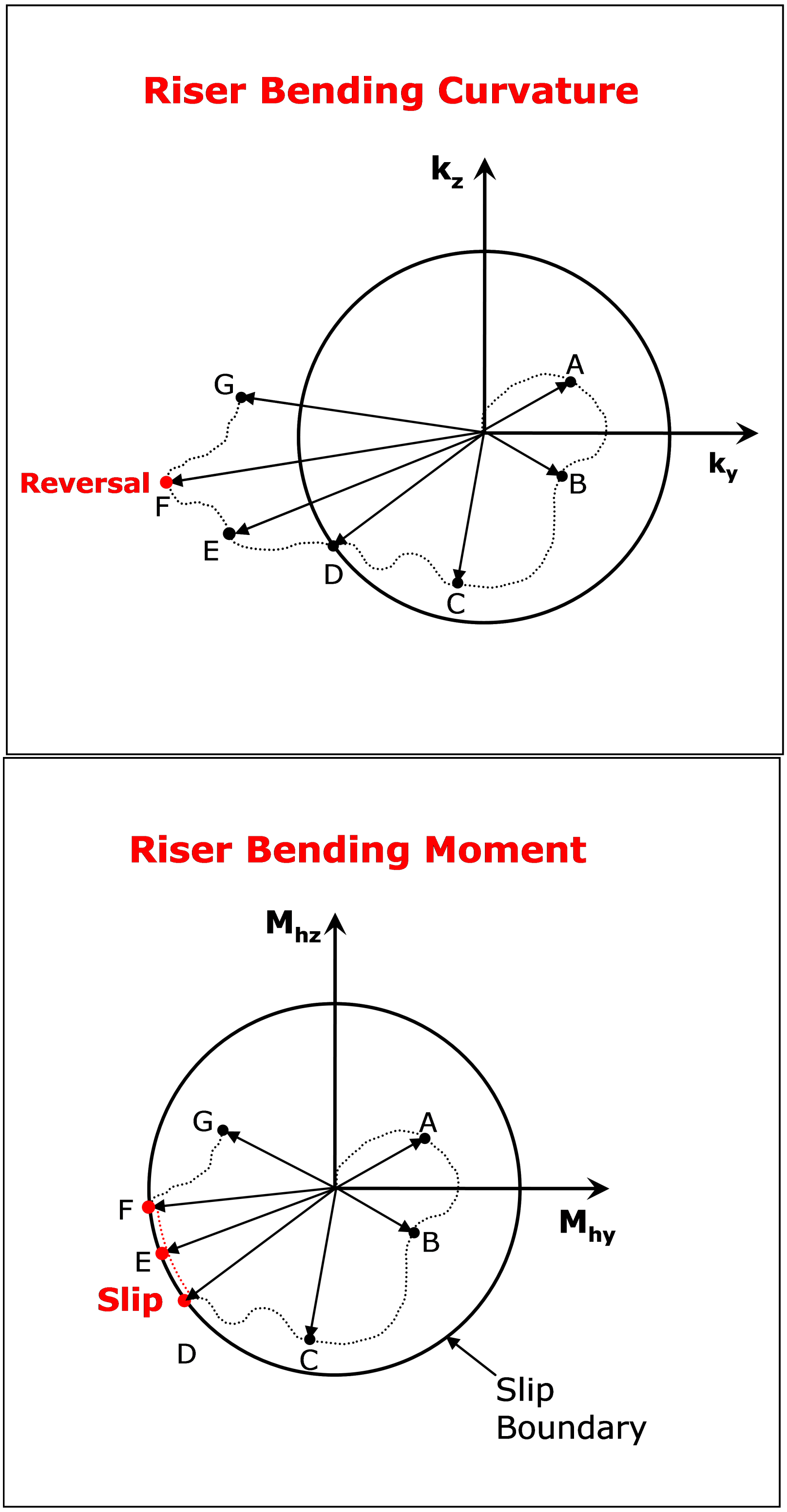
Uniaxial Bending Relationship

The values of M, EI, κ and h are iteratively updated at each time step. The moment update is based on the increment of curvature from the previous time step and the expected stiffness as computed from the pre-hysteresis analysis. The moment is clipped if necessary to stay within the elastoplastic limit ($\pm M_{slip} = \pm EIh\ \kappa_{slip}$) and the stiffness is adjusted to take account of the corrected moment. The stiffness generally equals the pre-slip or post-slip values (*EIh* or *0)*, or an interim value if the moment changes between elastic and plastic values. The intercept term h is computed from the curvature and moment at the previous time step and the current stiffness. The intercept term maintains continuity of the bending moment as the stiffness changes between pre- and post-slip values. The finite element formulation assembles the intercept term with the global load vector on the right-hand side of the equation of motion as previously described.

The above figure also shows that the slip moment M_{slip} provides a natural parameter for defining the elastoplastic limit. The bending curvature is not constrained by the elastoplastic relationship but the moment is restricted to stay to between $\pm M_{slip}$.

Three-Dimensional Bending – Biaxial

An elastoplastic component in a three-dimensional analysis supports biaxial loading with curvature $\boldsymbol{\kappa} = (\kappa_y, \kappa_z)$ and moment $\mathbf{M_h} = (M_{hy}, M_{hz})$. The figures below show the key concepts of the biaxial elastoplastic methodology.



Biaxial Bending Relationship

The figure on the left shows the riser bending curvature with local components κ_y and κ_z. The circle denotes the (initial) slip curvature as the elastoplastic component bends from straight. The figure on the right shows the bending moment of an elastoplastic component with moments M_{hy} and M_{hz}. The circle denotes the slip moment of the elastoplastic component. The moment is elastic when inside the circle and plastic when on the boundary. The moment cannot pass outside this circle.

The curve *A* to *G* shows an example of the loading. The loading starts from the origin and passes through points *A*, *B* and *C* with an elastic response and continues to point *D* at the slip boundary. The moment becomes plastic at *D* and continues along the slip boundary through *E* until point *F*. The plastic moment maintains constant magnitude but tracks the direction of the curvature. The curvature reverses at *F* (magnitude reduces) and the moment returns to inside the slip boundary.

The hysteresis relationship for a biaxial elastoplastic component is defined by:

$$\mathbf{M} = EI\boldsymbol{\kappa} + \mathbf{h} \qquad (2)$$

where **M**, **κ** and **h** denote the moment, curvature and intercept vectors and *EI* is a scalar (symmetric) bending stiffness that applies to both axes.

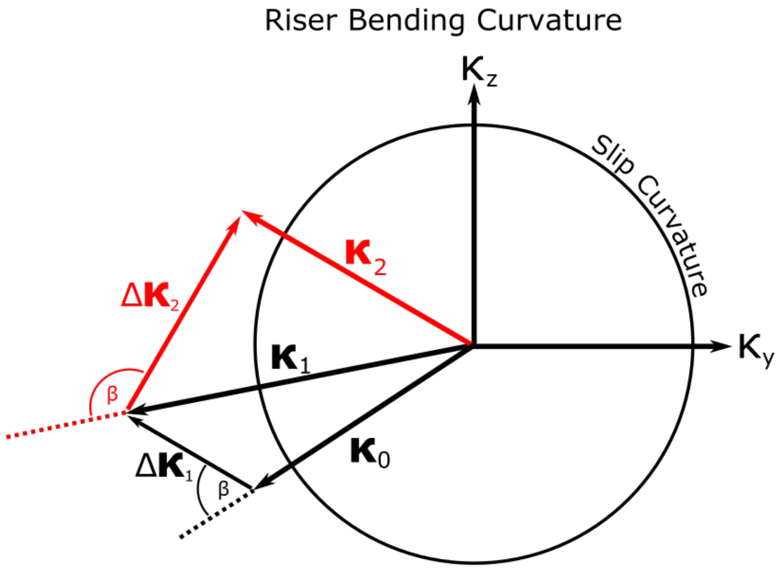
The values of **M**, *EI*, **κ** and **h** are iteratively updated at each time step in a similar manner to the uniaxial elastoplastic component. The moment is first updated using the increment of curvature from the previous time step and the expected stiffness from the pre-hysteresis analysis. The magnitude of the updated moment vector is reduced if necessary to stay within the elastoplastic circular limit.

Curvature Reversal

A curvature reversal occurs when:

1. The magnitude of the curvature, $|\boldsymbol{\kappa}|$, from the previous time step exceeds the spring slip curvature.
2. The angle, β, between the curvature at the previous time step, $\boldsymbol{\kappa}_{t-1}$, and current change in curvature, $\Delta\boldsymbol{\kappa}_t = \boldsymbol{\kappa}_t - \boldsymbol{\kappa}_{t-1}$ is between 90º and 270º.

These two conditions are illustrated in the figure below where a series of curvatures are shown at time *t = 0*, *t = 1* and *t = 2*:



Curvature Reversal

A curvature reversal occurs between *t = 1* and *t = 2* because the curvature κ_1 is outside the slip circle, and the angle between κ_1 and $\Delta\kappa_2$ is between 90º and 270º.

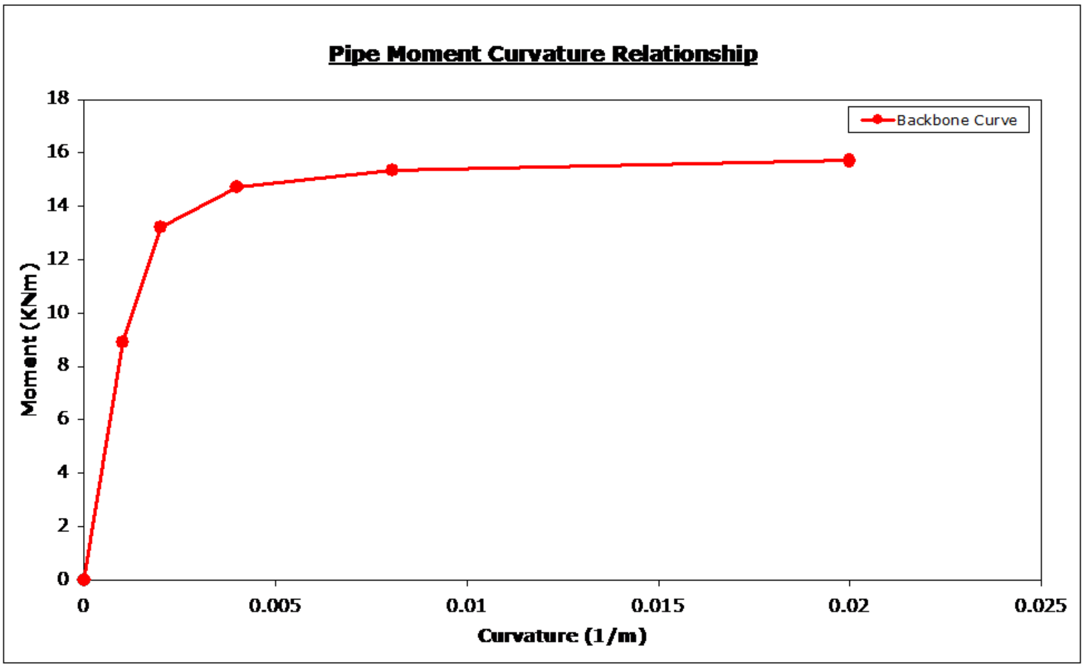
Further Information

More details on how Flexcom models hysteretic bending is provided in Smith. et al., (2007).

RELEVANT KEYWORDS

- `*BENDING HYSTERESIS` is used to define hysteresis moment-curvature backbone curves for non-linear materials.
- `*NO HYSTERESIS` is used to suppress bending hysteresis effects.

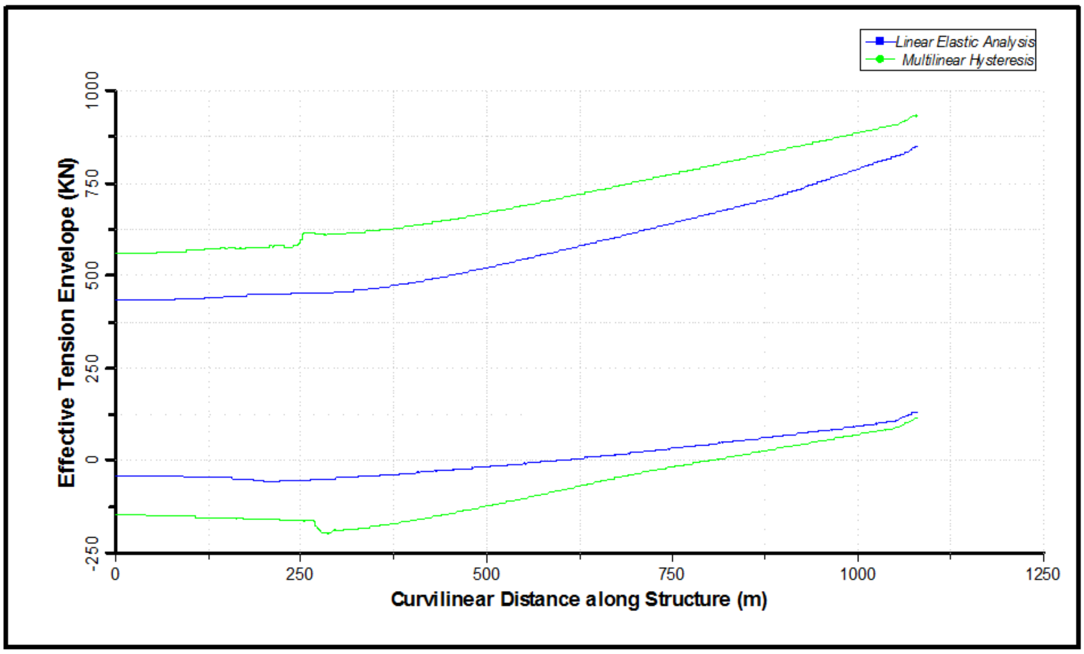
Design predictions of excess bending in the touchdown zone of deepwater catenary risers have been of concern in recent years. To illustrate the benefits of incorporating bending hysteresis effects in the global analysis, the analysis of a deepwater catenary is considered. The riser is a 6” production riser deployed from an FPSO in 705m of water, similar to the Free Hanging Catenary example from the standard Flexcom examples set. The hysteresis option is invoked by specifying an appropriate backbone curve as described above. The backbone curve used for this particular example is shown below.



Hysteresis Backbone Curve for Deepwater Catenary

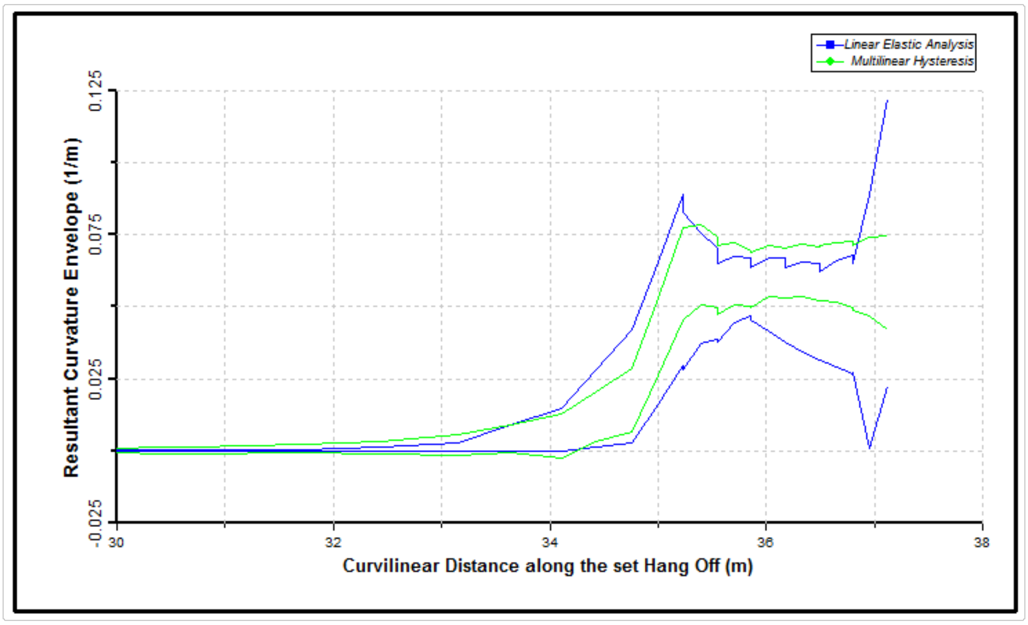
The wave loading applied to the riser system is a regular wave of amplitude 5m and period 14s. This represents an extreme condition at a benign offshore location. Bending curvatures in the touchdown zone and at the riser hang-off are compared with results obtained from a conventional linear elastic analysis.

The below figure compares the envelopes of effective tension for the linear elastic and non-linear hysteresis models. Significant compression occurs in the riser in both cases, with the level of compression increased in the hysteresis case.

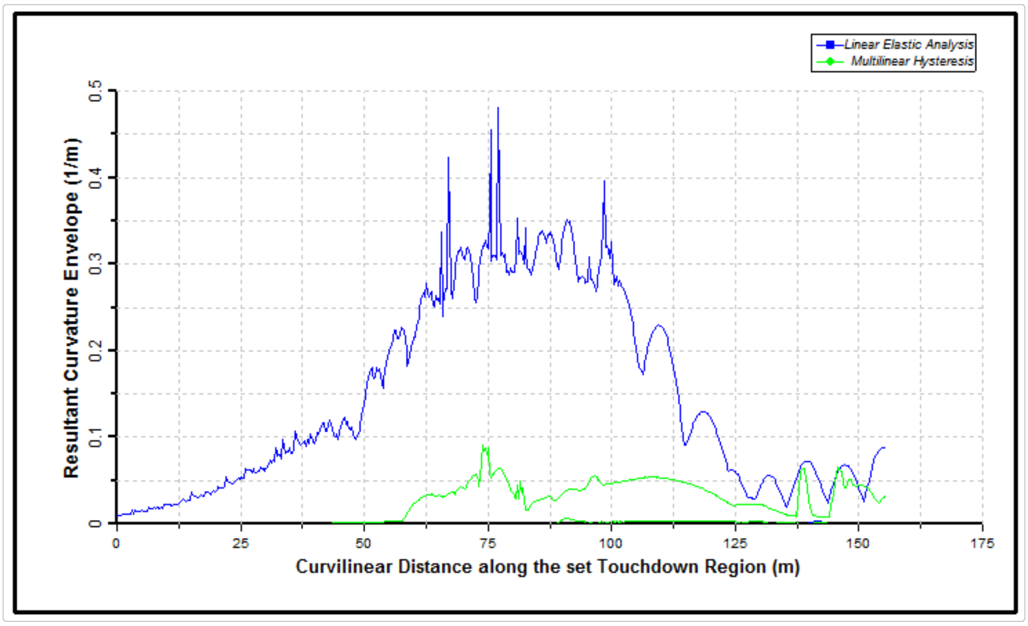


Effective Tension Envelope

The following two figures compare envelopes of curvature at the riser hang-off point and in the riser touchdown zone respectively. The touchdown curvatures based on the linear elastic bending exceed design requirements in several locations. In contrast, curvatures based on the hysteretic backbone curve give notably smaller touchdown and hang-off curvatures which would enable the riser design to pass.

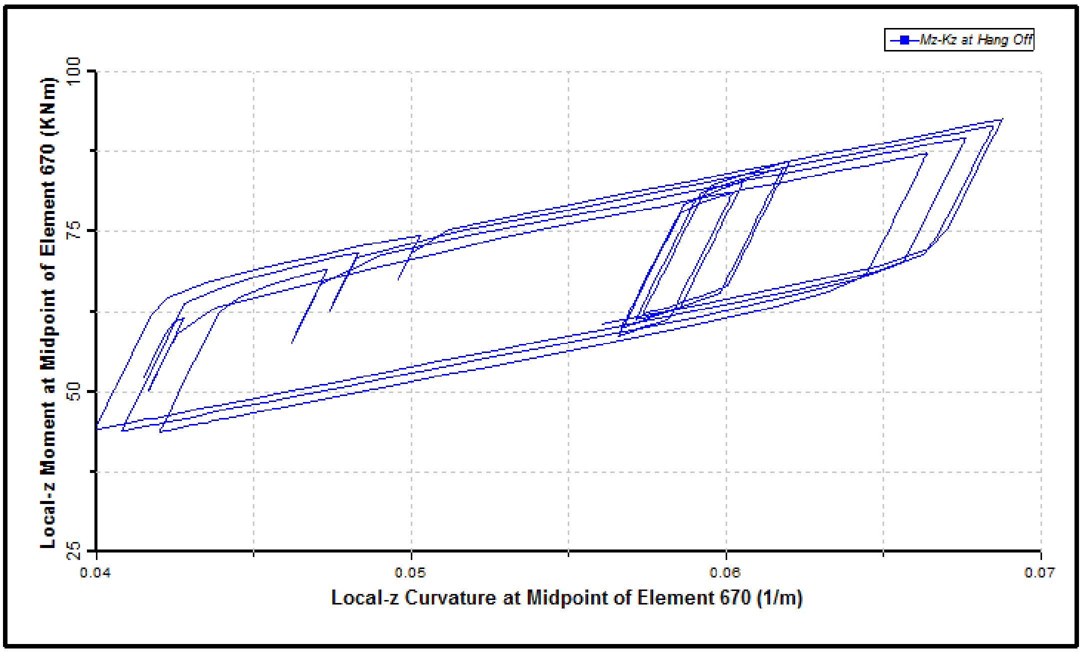


Dynamic Curvature Envelope at Riser Hang-off



Dynamic Curvature Envelope in Touchdown Zone

The below figure shows a variable/variable plot of moments and curvatures at the hang off point. The plot demonstrates that the response is hysteretic.



Moment-Curvature Response in Touchdown Zone

OVERVIEW

It is recommended that in analyses in which the bending response of a flexible riser is characterised by hysteresis, that the seabed contact model used be elastic (rather than rigid). Experience has shown that the convergence characteristics of analyses which include hysteresis are greatly improved if an elastic seabed is used. Refer to Elastic Seabed Profile for options available for defining an elastic seabed bathymetry

Based on simple test cases performed during the original implementation of the hysteresis modelling feature in Flexcom, it was felt preferable initially to use the Constant (rather than the default Updated) damping formulation, where damping effects were included. Refer to Damping for further information on the various damping options in Flexcom. The rationale was that as the damping matrix is directly proportional to the stiffness matrix, instantaneous changes in damping forces could be introduced by the slip-stick nature of hysteretic response, which could potentially affect the solution. However, based on experienced gained with realistic flexible riser models since then, using the Updated formulation is now recommended practice. The stiffness of a flexible riser includes a significant contribution from the effective tension distribution, and as the effective tension varies during a dynamic analysis, so too does the riser stiffness. Consider the case of a flexible riser in a simple catenary configuration – if effective compression occurs intermittently in the touchdown region, the geometric stiffness contribution can alternate between substantial and negligible values. Furthermore, if the model is susceptible to buckling, the orientation of elements in the touchdown zone can vary quickly and significantly. Both of these issues will effectively be ignored in the assembly of the structural damping matrix if a Constant formulation is used.

RELEVANT KEYWORDS

- `*SEABED PROPERTIES` is used to specify properties such as seabed type (i.e. rigid or elastic) and friction parameters.
- `*DAMPING FORMULATION` is used to specify the damping formulation to be used in a time domain dynamic analysis.

Mass and Polar Inertia per Unit Length

MASS AND POLAR INERTIA

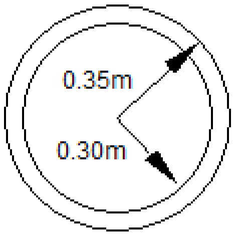
The mass per unit length and the polar inertia per unit length are related terms. The mass per unit length is a measure of the inertia of the structure with respect to translation. The polar inertia is a measure of the inertia of the structure with respect to torsional rotations about the cross section axis.

The mass per unit length m is calculated as the product of the cross-sectional area A and the mass density (mass per unit volume) r of the material. Similarly the polar inertia per unit length p is the product of the polar moment J of the cross-section and the mass density r. The dimensions of p are [Mass][Length]. In the SI system of units, p is typically in kg.m. In Imperial units, p is typically in slugs.ft.

Note that the influence of the value of p on the solution in very many cases is small. Also, in a 2D analysis, the value of p (and indeed the value of GJ) is of course immaterial.

EXAMPLE POLAR INERTIA CALCULATION

Consider a steel pipe as shown in the figure below. The density ρ of the pipe material is 7850 kg/m^3. The pipe inside and outside diameters are 0.30m and 0.35m respectively. The calculation of p is as follows:



Example Polar Inertia Calculation

Polar moment, J, is given by:

$$\frac{\pi\left(D_o^4 - D_i^4\right)}{32} = \frac{\pi\left(0.35^4 - 0.30^4\right)}{32}$$

$= 6.78 \times 10^{-4}$ m^4

Polar moment of inertia per unit length, p = J. ρ

$= (6.78 \times 10^{-4}).(7850)$ kg.m

= 5.32 kg.m

Diameter Inputs

There are a number of different element diameters you can specify when inputting geometric properties in the Flexible Riser format. Specifically you can input values for:

- Internal diameter
- Drag diameter

- Buoyancy diameter
- Outer diameter
- Contact diameter

This section discusses the significance of each of these diameter inputs in order to eliminate any possible ambiguity or confusion. Diameters Summary shows this information in a convenient table format.

In the Flexible Riser format, diameter data you specify is used in three ways. Firstly, it is used in calculating forces on an element from a variety of sources as part of setting up the finite element equilibrium equations. Secondly, it is used in calculating hoop, bending and von Mises stress output from the results of an analysis, this being a purely postprocessing operation. Thirdly, it is used to calculate the effect of coatings. Coatings are the subject of a separate section.

In the case of geometric properties in the Flexible Riser format, the internal diameter (Di) is used for computing the buoyancy contribution of internal fluid, if present. It is also used as the inner diameter for stress calculations unless you specify otherwise. Di defaults to a value of zero, although that would be an unusual value.

The drag diameter (Dd) is the effective external diameter for hydrodynamic force evaluation using Morison’s Equation. It is also used as the external diameter for stress calculations unless you specify otherwise.

The buoyancy diameter (Db) is the effective external diameter for buoyancy force calculations. If you are using the Default buoyancy option, the applied buoyancy force is based only on Db. If you are using the Distributed option, then the buoyancy force is calculated using Db and the outer diameter Do, as now described. Db is also used in some circumstances in the calculation of the added mass/inertia components of the Morison's Equation hydrodynamic force – this is discussed in Element Diameter and Hydrodynamic Forces.

For the Flexible Riser format, the outer diameter Do is an appropriate input only if:

- You are using the Distributed buoyancy option, or
- External coatings are specified

In either of these cases Do is a required input, and is taken to represent the actual outer diameter of the riser (without for example buoyancy material).

In terms of buoyancy, Flexcom uses Do to calculate the buoyancy loads generated by the riser itself, and these are summed into the element load matrix in the usual way. The difference between the buoyancy loading calculated using Do, and the buoyancy loading calculated using Db, is then applied as a uniformly distributed load in the element axial direction, as discussed earlier in this section.

If you are using the Default buoyancy option and you have not specified any external coatings, then any value you specify for Do is immaterial and unused. It is because this entry is optional in the majority of cases that the external diameter for the calculation of stresses defaults to Dd when you use the Flexible Riser format.

The contact diameter (Dc) is the effective diameter for contact calculations, and is relevant only when your Flexcom analysis includes guide surface contact, line clashing or pipe-in-pipe contact. Flexcom uses this input to determine when contact occurs. Specification of a contact diameter is optional, and Dc defaults to the largest of Db, Dd or Do if omitted.

Diameter	**Inner**	**Outer**	**Drag**	**Buoyancy**	**Contact**
Symbol	Di	Do	Dd	Db	Dc

Used for **(All formats)**	Computing the mass/buoyancy contribution of any internal fluid The inner diameter for stress calculations unless you specify otherwise under *PROPERTIES The base diameter onto which any internal coatings specified with *COATINGS will be applied.	The base diameter onto which any external coatings specified with *COATINGS will be applied. If you invoke the Distributed Buoyancy option, then the outer diameter Do is used in calculating buoyancy forces due to the structure	The effective outer diameter for hydrodynamic force evaluation using Morison's Equation Strictly speaking the inertia/added mass components in Morison's Equation should be based on the buoyancy diameter (Db). This can be set under *HYDRODYNAMIC SETS with the DIAMETER=BUOYANCY tag	The effective outer diameter for buoyancy force calculations If you invoke the *Distributed Buoyancy* option, then the difference between the buoyancy diameter Db and the outer diameter Do, is used in calculating buoyancy forces due to the buoyancy material only Strictly speaking the inertia/added mass components in Morison's Equation should be based on the buoyancy diameter (Db). This can be set under *HYDRODYNAMIC SETS with the DIAMETER=BUOYANCY tag	The effective diameter for contact calculations (guide surface contact, line clashing or pipe-in-pipe contact) Optionally used for seabed contact under *SEABED PROPERTIES The diameter of elements displayed in the Model View
Additionally Used For **(Rigid Riser Format)**	Used for computing cross-section area, moment of inertia and polar moment of inertia, if these are not specified directly	Used for computing cross-section area, moment of inertia and polar moment of inertia, if these are not specified directly The outer diameter for stress calculations unless you specify otherwise under *PROPERTIES The diameter used for CLEAR if not specified otherwise under *PROPERTIES or explicitly specified in the clear analysis input			
Additionally Used For **(Flexible Riser Format)**			The outer diameter for stress calculations unless you specify otherwise under *PROPERTIES The diameter used for CLEAR if not specified		

			otherwise under *PROPERTIES or explicitly specified in the clear analysis input		
Default Value (Rigid Riser Format)	0.0	Required Input	Outer Diameter (Do)	Outer Diameter (Do)	Max(Db, Dd, Do)
Default Value (Flexible Riser Format)	0.0	Required Input (for limited cases only)	0.0	0.0	Max(Db, Dd, Do)

Geometric Properties in Rigid Riser Format

OVERVIEW

When using the Rigid Riser format you specify geometric property data under five headings:

- Structural properties: there are three material models available, Linear Elastic, Non-linear Elastic, and Linear Elastic with Plastic Hardening.
- Diameter Inputs: there are four external diameter inputs and one internal diameter input
- Mass density
- Cross-section properties
- Buoyancy Formulations: the options are Default and Distributed

All of these are discussed in more detail in the proceeding sections.

RELEVANT KEYWORDS

- `*GEOMETRIC SETS` is used to assign geometric properties to element sets (see OPTION=RIGID).
- `*STRESS/STRAIN DIRECT` is used to define stress-strain curves for non-linear materials.

If you would like to see an example of how structural properties are defined in the Rigid Riser format, refer to A01 - Deepwater Drilling Riser.

Material Models

If the case of the Rigid Riser format, there are three material models available to you:

- Linear Elastic
- Non-linear Elastic
- Linear Elastic with Plastic Hardening

The various stiffness terms (bending stiffnesses EI_{yy} and EI_{zz}, torsional stiffness GJ, and axial stiffness EA) are calculated from values you specify for Young's modulus E and shear modulus G, and from the values of the cross-section properties I, A and J. EI_{yy} and EI_{zz} are necessarily equal in this case.

To specify non-linear elastic behaviour, rather than inputting a single value of Young's modulus you input instead a direct stress-strain (σ-ε) relationship. This defines both non-linear bending and non-linear axial response. You are still required to input a (single) value for shear modulus – the option to specify non-linear torsional response is not available with this format, but this option in any event has limited usage. Note also that you cannot combine non-linear bending response with linear axial behaviour, and vice versa.

What actually happens in this case is that Flexcom transforms the direct stress-strain relationship into moment/curvature and axial force/axial strain curves for each element of the set in question, based on I and A values for that set. Thereafter the operation is the same as for the Flexible Riser format. Note however that non-linear bending response with the Rigid Riser format is (necessarily) Symmetric by the above definition.

Refer to Non-linear Material Force Term for further information on how the Flexcom solver accounts for non-linear materials in the equilibrium equations.

OVERVIEW

The mechanical properties used in an elastic-plastic loading scenario are the elasticity constants Young's modulus (*E*) and Poisson's ratio (*v*), and the stress-strain curve (obtained from a tensile test). The latter is used to determine the yield point and the nonlinear isotropic hardening of the material. This information can be also expressed in the form of a stress-plastic strain curve, which starts at the yield point and terminates at the point of fracture. The plastic strain (ε^{pl}) can be expressed as:

$$\varepsilon^{pl} = \varepsilon - \frac{\sigma}{E} \tag{1}$$

where σ is stress and ε is total strain.
The constitutive equations governing the 3D beam element are expressed in terms of forces and moments (generalised stresses) and their conjugates strains and curvatures (generalised strains), respectively. This requires that any nonlinear behaviour in bending, axial or torsion be explicitly expressed either independently of each other (Nonlinear Flexible Riser Format) or in some form of dependence (Nonlinear Rigid Riser Format). Elastic deformations can be superposed as they are reversible, but plastic deformations cannot. Plastic deformation is path dependent, which means that the order in which loads are applied to the body is important and will lead to a different yield state.

Flexcom uses a hybrid beam element, where constitutive equations are specified in terms of forces and moments and not stresses. There is no established measure of yielding expressed in terms of forces and moments. The J_2 flow theory can be applied selectively to each mode of deformation to control the yielding independently. In this situation, the balancing of the beam capacity has to be performed explicitly. In other words, the bending, axial and torsion loading capacity of the beam must always sum to the same overall value. For example, should the beam becom overstretched, then it cannot support the same bending moment as if it was not loaded at all. Adjusting the capacity is done automatically by the program, which assumes that torsion loads do not produce any plastic deformations. This is reasonable given the nature of the engineering problem, where bending and tensile loads would typically produce plastic deformations. It is considered that the bending moment capacity is dependent on the amount of resultant axial force in the cross-section, and which is calculated by integrating the stress-strain curve over the cross-section. The bending capacity is adjusted at least once at the beginning of an analysis for all beam elements in the model. These adjustments can be performed dynamically at the run-time if the tension changes by more than a specified amount.

Plastic hardening is implemented for the bending and axial responses, while the torsional response is considered to be linear elastic. The shear modulus (G) is calculated from:

$$G = \frac{E}{2(1+\nu)} \tag{2}$$

The effects of internal and external pressure are included after the axial or bending deformation has taken place. Note that Flexcom does not readjust bending moment capacity as a result of the applied pressure.

If you would like to see an example of the plastic hardening material model in use please refer to Standard Example H05 - Steel Pipe Installation with Plastic Deformation.

RELEVANT KEYWORDS

- `*PLASTIC HARDENING` is used to define hardening models for plastic materials.

The bending response of the beam is determined using the coupled local-y and local-z bending. The following quantities are defined, in the context of the J_2 flow theory.

Term	Definition
$M_Y = i$	Moment-equivalent plastic curvature (input)
$h = \frac{\partial M}{\partial \bar{k}}$	Hardening modulus
EI^{el}	Elasticity modulus, analogous to the Young's modulus (input)
$\Delta\mathbf{k}$	Increment in total curvatures from beginning of the time-step (input)

$\mathbf{k}_{t+\Delta t}^{el}$, $\mathbf{k}_{t}^{el}$	Elastic curvatures at end of the time-step (output) and beginning of the time-step (input)
$\mathbf{k}_{t+\Delta t}^{pl}$, $\mathbf{k}_{t}^{pl}$	Plastic curvatures at end of the time-step (output) and beginning of the time-step (input)
$\overline{k}_{t+\Delta t}^{pl}$, $\overline{k}_{t}^{pl}$	Equivalent plastic curvature at end of the time-step (output) and beginning of the time-step (input)
$\Delta\overline{k}^{pl}$	Increment in equivalent plastic curvature
$\mathbf{M}_{t+\Delta t}$, $\mathbf{M}_{t}$	Moments at end of the time-step (output) and beginning of the time-step (input)
$\mathbf{M}^{pr}$, $\overline{M}^{pr}$, M_Y	Predictor moment, effective predictor moment and yield moment (output)
EI_{eff}^{el}, h_{eff}	Effective elastic and hardening moduli
$\frac{\partial\Delta\mathbf{M}}{\partial\Delta\mathbf{k}}$	Material Jacobian (output)
h_0	Zero curvature intercept (output)

The total local-y and local-z curvatures are used as the main input to determine, if yielding has occurred at the current time-step. The algorithm updates the moments and curvatures, but also the material Jacobian and zero curvature intercept which are needed in the assembly of the global system of finite element equations. The algorithm used is based on a “predictor” moment, which is calculated purely on elastic behaviour. Therefore, the material Jacobian is defined as:

$$\frac{\partial \Delta \mathbf{M}}{\partial \Delta \mathbf{k}} = diag\left(EI^{el} \quad EI^{el}\right) \tag{1}$$

$\frac{\partial \Delta \mathbf{M}}{\partial \Delta \mathbf{k}}$ is defined in elastic terms only, thus linking the increment in the moment, $\Delta \mathbf{M}$, to the increment in curvature, $\Delta \mathbf{k}$. It follows that “predictor” moment is:

$$\mathbf{M}^{pr} = \mathbf{M}_t + \frac{\partial \Delta \mathbf{M}}{\partial \Delta \mathbf{k}} \Delta \mathbf{k} \tag{2}$$

The effective "predictor" moment is:

$$\overline{M}^{pr} = \sqrt{\left(M_y^{pr}\right)^2 + \left(M_z^{pr}\right)^2} \tag{3}$$

The elastic curvature at the beginning of the time-step, $\mathbf{k}_t^{el}$, is updated with the increment in curvature:

$$\mathbf{k}_{t+\Delta t}^{el} = \mathbf{k}_t^{el} + \Delta \mathbf{k} \tag{4}$$

Using equation (3), the effective “predictor” moment, $\overline{M}^{pr}$, is calculated and compared against the yield moment at the beginning of the time-step $M_{Y,t}$ in order to determine if the material has yielded. If yielding does not occur, $\overline{M}^{pr} \le M_{Y,t}$, then the deformation is entirely elastic. Hence, the new values for moment and elastic curvature are those given by equations (2) and (4), respectively. However, if the effective “predictor” moment is larger than the calculated yield moment at the beginning of the time-step, $\overline{M}^{pr} > M_{Y,t}$, then the components of the elastic and plastic curvatures tensors as well as the moment tensor need to be corrected. In this case, the following equation is solved iteratively for the increment in the equivalent plastic curvature, $\Delta \overline{k}^{pl}$.

$$\overline{M}^{pr} - EI^{el}\Delta\overline{k}^{pl} - M_Y\left(\overline{k}^{pl}\right) = 0 \tag{5}$$

Once $\Delta\overline{k}^{pl}$ is known, and hence the equivalent plastic curvature at end of the time-step is known, the solution is fully defined and $\mathbf{M}_{Y,t+\Delta t}$ is known. The moments can be updated to the end of the time-step as:

$$\mathbf{M} = \frac{\mathbf{M}^{pr}}{\overline{M}^{pr}} M_{Y,t+\Delta t} \tag{6}$$

The increment in plastic curvature is:

$$\Delta\mathbf{k}^{pl} = \frac{\mathbf{M}^{pr}}{\overline{M}^{pr}} \Delta\overline{k}^{pl} \tag{7}$$

Therefore, the elastic curvature at the end of the time-step is corrected to:

$$\mathbf{k}^{el}_{t+\Delta t} = \mathbf{k}^{el}_t + \Delta\mathbf{k} - \Delta\mathbf{k}^{pl} \tag{8}$$

The plastic and equivalent plastic curvatures at the end of the time-step are found to be:

$$\begin{aligned} \mathbf{k}^{pl}_{t+\Delta t} &= \mathbf{k}^{pl}_t + \Delta\mathbf{k}^{pl} \\ \overline{k}^{pl}_{t+\Delta t} &= \overline{k}^{pl}_t + \Delta\overline{k}^{pl} \end{aligned} \tag{9}$$

The material Jacobian needs to be updated to reflect the new hardening moduli:

$$\frac{\partial\Delta\mathbf{M}}{\partial\Delta\mathbf{k}} = diag\left(EI^{el}_{eff} \quad EI^{el}_{eff}\right) + h_{eff}\,\frac{M^{pr}_i}{\overline{M}^{pr}}\frac{M^{pr}_j}{\overline{M}^{pr}} \tag{10}$$

where *i* and *j* are placeholders for *y* and *z*,

$$h_{eff} = \frac{EI^{el}h}{EI^{el} + h} - EI^{el}_{eff} \tag{11}$$

and

$$EI_{eff}^{el} = EI^{el} \frac{M_{Y,t+\Delta t}}{\overline{M}^{pr}} \tag{12}$$

Finally, the zero curvature intercept if found as

$$\mathbf{h}_0 = \mathbf{M} - \frac{\partial \Delta \mathbf{M}}{\partial \Delta \mathbf{k}} \left(\mathbf{k}_{t+\Delta t}^{el} + \mathbf{k}_{t+\Delta t}^{pl} \right) \tag{13}$$

The axial response of the beam is described below using uni-dimensional equations. The following quantities are defined, in the context of the J_2 flow theory.

Term	Definition
$F_Y = F$	Force-equivalent plastic axial strain (input)
$h = \frac{\partial F}{\partial \bar{\varepsilon}}$	Hardening modulus
EA^{el}	Elasticity modulus, analogous to the Young's modulus (input)
$\Delta\varepsilon$	Increment in total axial strain from beginning of the time-step (input)
$\varepsilon_{t+\Delta t}^{el}$, ε_t^{el}	Elastic axial strain at end of the time-step (output) and beginning of the time-step (input)
$\varepsilon_{t+\Delta t}^{pl}$, ε_t^{pl}	Plastic axial strain at end of the time-step (output) and beginning of the time-step (input)
$\bar{\varepsilon}_{t+\Delta t}^{pl}$, $\bar{\varepsilon}_t^{pl}$	Equivalent plastic axial strain at end of the time-step (output) and beginning of the time-step (input)

$\Delta\bar{\varepsilon}^{pl}$	Increment in equivalent plastic axial strain
$F_{t+\Delta t}$, F_t	Force at end of the time-step (output) and beginning of the time-step (input)
F^{pr}, $\overline{F}^{pr}$, F_Y	Predictor force, effective predictor force and yield force (output)
EA_{eff}^{el}, h_{eff}	Effective elastic and hardening moduli
$\frac{\partial \Delta F}{\partial \Delta \varepsilon}$	Material Jacobian (output)
h_0	Zero axial strain intercept (output)

The total axial strain is used as the main input to determine, if yielding has occurred at the current time-step. The algorithm updates the force and the axial strains, but also the material Jacobian and zero axial strain intercept which are needed in the assembly of the global system of finite element equations. The algorithm used is based on a “predictor” force, which is calculated purely on elastic behaviour. Therefore, the material Jacobian is defined as:

$$\frac{\partial \Delta F}{\partial \Delta \varepsilon} = EA^{el} \tag{1}$$

$\frac{\partial \Delta F}{\partial \Delta \varepsilon}$ is defined in elastic terms only, thus linking the increment in force, ΔF, to the increment in axial strain, $\Delta \varepsilon$. It follows that “predictor” force is:

$$F^{pr} = F_t + \frac{\partial \Delta F}{\partial \Delta \varepsilon} \Delta \varepsilon \tag{2}$$

The effective "predictor" force is:

$$\overline{F}^{pr} = \left| F^{pr} \right| \tag{3}$$

The elastic axial strain at the beginning of the time-step, ε_t^{el}, is updated with the increment in axial strain:

$$\varepsilon_{t+\Delta t}^{el} = \varepsilon_t^{el} + \Delta\varepsilon \tag{4}$$

Using equation (3), the effective “predictor” force, $\overline{F}^{pr}$, is calculated and compared against the yield force at the beginning of the time-step $F_{Y,t}$ in order to determine if the material has yielded. If yielding does not occur, $\overline{F}^{pr} \le F_{Y,t}$, then the deformation is entirely elastic. Hence, the new values for axial force and elastic axial strain are those given by equations (2) and (4), respectively. However, if the effective “predictor” force is larger than the calculated yield axial force at the beginning of the time-step, $\overline{F}^{pr} > F_{Y,t}$, then the elastic and plastic axial strains as well as the force need to be corrected. In this case, the following equation is solved iteratively for the increment in the equivalent plastic axial strain, $\Delta\overline{\varepsilon}^{pl}$.

$$\overline{F}^{pr} - EA^{el}\Delta\overline{\varepsilon}^{pl} - F_Y\left(\overline{\varepsilon}^{pl}\right) = 0 \tag{5}$$

Once $\Delta\overline{\varepsilon}^{pl}$ is known, and hence the equivalent plastic axial strain at end of the time-step is known, the solution is fully defined and $F_{Y,t+\Delta t}$ is known. The force can be updated to the end of the time-step as:

$$F = \frac{F^{pr}}{\overline{F}^{pr}} F_{Y,t+\Delta t} \tag{6}$$

The increment in plastic axial strain is:

$$\Delta\varepsilon^{pl} = \frac{F^{pr}}{\overline{F}^{pr}} \Delta\overline{\varepsilon}^{pl} \tag{7}$$

Therefore, the elastic axial at the end of the time-step is corrected to:

$$\varepsilon_{t+\Delta t}^{el} = \varepsilon_t^{el} + \Delta\varepsilon - \Delta\varepsilon^{pl} \quad (8)$$

The plastic and equivalent plastic axial strains at the end of the time-step are found to be:

$$\varepsilon_{t+\Delta t}^{pl} = \varepsilon_t^{pl} + \Delta\varepsilon^{pl} \quad (9)$$
$$\bar{\varepsilon}_{t+\Delta t}^{pl} = \bar{\varepsilon}_t^{pl} + \Delta\bar{\varepsilon}^{pl}$$

The material Jacobian needs to be updated to reflect the new hardening moduli:

$$\frac{\partial\Delta F}{\partial\Delta\varepsilon} = EA_{eff}^{el} + h_{eff}\frac{F^{pr}}{\bar{F}^{pr}}\frac{F^{pr}}{\bar{F}^{pr}} \quad (10)$$

where

$$h_{eff} = \frac{EA^{el}h}{EA^{el}+h} - EA_{eff}^{el} \quad (11)$$

and

$$EA_{eff}^{el} = EA^{el}\frac{F_{Y,t+\Delta t}}{\bar{F}^{pr}} \quad (12)$$

Finally, the zero curvature intercept if found as

$$h_0 = F - \frac{\partial\Delta F}{\partial\Delta\varepsilon}\left(\varepsilon_{t+\Delta t}^{el} + \varepsilon_{t+\Delta t}^{pl}\right) \quad (13)$$

Diameter Inputs

There are a number of different element diameters you can specify when inputting geometric properties in the Rigid Riser format.

Specifically you can input values for:

- Outer diameter
- Internal diameter

- Drag diameter
- Buoyancy diameter
- Contact diameter

This section discusses the significance of each of these diameter inputs in order to eliminate any possible ambiguity or confusion. Diameters Summary shows this information in a convenient table format.

In the Rigid Riser format, diameter data you specify is used in four ways. Firstly, it is used in calculating forces on an element from a variety of sources as part of setting up the finite element equilibrium equations. Secondly, some of the values can be used in calculating cross-section properties. Thirdly, diameter values are used in calculating hoop, bending and von Mises stress output from the results of an analysis, this being a purely postprocessing operation. Fourthly, it is used to calculate the effect of coatings. Coatings are the subject of a separate section.

In the case of geometric properties in the Rigid Riser format, the outer diameter Do is a required input. Dd and Db both default to the value you specify for Do unless you input values directly. Do is also used (with Di) for computing the cross-section area, moment of inertia and polar moment of inertia if these are not specified directly. Finally, it is used as the external diameter for stress calculations unless you specify otherwise.

The internal diameter (Di) is used for computing the buoyancy contribution of internal fluid, if present. The internal diameter is also used for computing cross-section area, moment of inertia and polar moment of inertia, if these are not specified directly. Di is used as the inner diameter for stress calculations unless you specify otherwise. Di defaults to a value of zero, although that would be an unusual value.

The drag diameter (Dd) is the effective outer diameter for hydrodynamic force evaluation using Morison’s Equation. For geometric properties in the Rigid Riser format, Dd defaults to Do, but you can override this.

The buoyancy diameter (Db) is the effective outer diameter for buoyancy force calculations. For geometric properties in the Rigid Riser format, Db defaults to Do, but you can override this. Db is also used in some circumstances in the calculation of the added mass/inertia components of the Morison's Equation hydrodynamic force – this is discussed in Element Diameter and Hydrodynamic Forces.

If you specify the Distributed buoyancy option, then specification of a value for Db greater than Do would be typical. Do is used to calculate the buoyancy loads generated by the riser itself, while the difference between the buoyancy loading calculated using Do and the buoyancy loading calculated using Db is then applied as a uniformly distributed load in the element axial direction, as discussed earlier in this section.

The contact diameter (Dc) is the effective diameter for contact calculations, and is relevant only when your Flexcom analysis includes guide surface contact, line clashing or pipe-in-pipe contact. Flexcom uses this input to determine when contact occurs. Specification of a contact diameter is optional, and Dc defaults to the largest of buoyancy Db, Dd or Do if omitted.

Diameter	**Inner**	**Outer**	**Drag**	**Buoyancy**	**Contact**
Symbol	Di	Do	Dd	Db	Dc
Used for (All formats)	Computing the mass/buoyancy contribution of any internal fluid The inner diameter for stress calculations unless you specify otherwise under *PROPERTIES The base diameter onto which any internal coatings specified with *COATINGS will be applied.	The base diameter onto which any external coatings specified with *COATINGS will be applied. If you invoke the Distributed Buoyancy option, then the outer diameter Do is used in calculating buoyancy forces due to the structure	The effective outer diameter for hydrodynamic force evaluation using Morison's Equation Strictly speaking the inertia/added mass components in Morison's Equation should be based on the buoyancy diameter (Db). This can be set under *HYDRODYNAMIC SETS with the DIAMETER=BUOYANCY tag	The effective outer diameter for buoyancy force calculations If you invoke the *Distributed Buoyancy* option, then the difference between the buoyancy diameter Db and the outer diameter Do, is used in calculating buoyancy forces due to the buoyancy material only Strictly speaking the inertia/added mass components in Morison's Equation should be based on the buoyancy diameter (Db). This can be set under *HYDRODYNAMIC SETS with the	The effective diameter for contact calculations (guide surface contact, line clashing or pipe-in-pipe contact) Optionally used for seabed contact under *SEABED PROPERTIES The diameter of elements displayed in the Model View

				DIAMETER=BUOYANCY tag	
Additionally Used For (Rigid Riser Format)	Used for computing cross-section area, moment of inertia and polar moment of inertia, if these are not specified directly	Used for computing cross-section area, moment of inertia and polar moment of inertia, if these are not specified directly The outer diameter for stress calculations unless you specify otherwise under *PROPERTIES The diameter used for CLEAR if not specified otherwise under *PROPERTIES or explicitly specified in the clear analysis input			
Additionally Used For (Flexible Riser Format)			The outer diameter for stress calculations unless you specify otherwise under *PROPERTIES The diameter used for CLEAR if not specified otherwise under *PROPERTIES or explicitly specified in the clear analysis input		
Default Value (Rigid Riser Format)	0.0	Required Input	Outer Diameter (Do)	Outer Diameter (Do)	Max(Db, Dd, Do)
Default Value	0.0	Required Input (for limited cases only)	0.0	0.0	Max(Db, Dd, Do)

(Flexible Riser Format)					

Mass Density

Mass is input in the Rigid Riser format as mass density, defined as mass per unit volume and denoted r. This defines both the section mass per unit length (m = A. r) and the polar inertia per unit length (p = J. r).

Cross Section Properties

You can specify values for the cross-section area A, the moment of inertia I and the polar moment of inertia J. Input of values for all of these is optional. Any you choose not to define explicitly will be calculated by default from element set internal and outer diameters.

Geometric Properties in Mooring Line Format

OVERVIEW

When using the Mooring Line format you specify geometric property data under three headings:

- Axial Stiffness
- Mass per Unit Length
- Various Diameter Inputs

RELEVANT KEYWORDS

- `*GEOMETRIC SETS` is used to assign geometric properties to element sets (see OPTION=MOORING).
- `*FORCE-STRAIN` is used to define force-strain curves for non-linear materials.

Axial Stiffness

In the case of the Mooring Line format, the material model is linear elastic. The governing input is axial stiffness (EA), with both bending (EI) and torsional stiffness (GJ) assumed to be negligible. While it is possible to specify very low values of the EI and GJ stiffness terms in the Flexible Riser format, this is not advisable. Recommended practice is to use the dedicated Mooring Line format, for reasons now outlined.

Modelling of mooring lines can be quite challenging, especially in severe dynamic environments, where there is a tendency for the mooring lines to go into compression. As the mooring lines typically have virtually zero structural bending stiffness, the standard beam column elements traditionally employed by Flexcom are not ideally suited for modelling this type of structure.

When you use the Mooring Line format, a slightly different solution procedure is applied internally within the program. Flexcom uses a special type of arrangement which effectively does not consider bending or torsional stiffness terms. Specifically, the mooring line is modelled using a combination of standard beam-column elements interspersed with hinge/articulation elements. This specialised combination means that while the mooring line has a realistic axial stiffness, it has effectively little or no associated bending or torsional resistance. This provides a robust method for modelling mooring lines without any significant loss in accuracy. The approach also results in increased computational efficiency over the standard Flexible Riser format elements.

Mass per unit Length

Similar to the Flexible Riser format, the mass per unit length is a user input in the Mooring Line format. The mass per unit length is a measure of the inertia of the structure with respect to translation.

Similarly, the polar inertia is a measure of the inertia of the structure with respect to torsional rotations about the cross section axis. In practice however, the influence of polar inertia on the solution in very many cases is small, and so for convenience, a nominal value of 1.0 is assumed for the Mooring Line format. The dimensions of p are [Mass][Length]. In the SI system of units, p is typically in kg.m. In Imperial units, p is typically in slugs.ft.

Buoyancy Modelling

For elements whose structural properties are defined using the Mooring Line format, the Default buoyancy formulation is used. Discrete buoyancy units are not typically associated with mooring lines, and in the unlikely event that these are required, they should be modelled explicitly. If you require further information on the Distributed buoyancy formulation, or buoyancy modelling in general, refer to Buoyancy Formulations.

Diameter Inputs

DIAMETER SPECIFICATIONS

There are a number of different element diameters relating to the Mooring Line format specification of geometric properties.

Specifically you can input values for:

- Outer diameter
- Drag diameter
- Buoyancy diameter

Values are automatically assigned to:

- Internal diameter
- Contact diameter

This section discusses the significance of each of these diameter inputs in order to eliminate any possible ambiguity or confusion.

In the Mooring Line format, diameter data you specify is used in four ways. Firstly, it is used in calculating forces on an element from a variety of sources as part of setting up the finite element equilibrium equations. Secondly, some of the values can be used in calculating cross-section properties. Thirdly, diameter values are used in calculating hoop, bending and von Mises stress output from the results of an analysis, this being a purely postprocessing operation. Fourthly, it is used to calculate the effect of coatings. Coatings are the subject of a separate section.

In the case of geometric properties in the Mooring Line format, the outer diameter Do is a required input. Dd and Db both default to the value you specify for Do unless you input values directly. Do is also used for computing the cross-section area, and it is used as the external diameter for stress calculations unless you specify otherwise.

In the case of geometric properties in the Mooring Line format, the internal diameter (Di) is assumed to be equal to zero.

The drag diameter (Dd) is the effective outer diameter for hydrodynamic force evaluation using Morison's Equation. For geometric properties in the Mooring Line format, Dd defaults to Do, but you can override this.

The buoyancy diameter (Db) is the effective outer diameter for buoyancy force calculations. For geometric properties in the Mooring Line format, Db defaults to Do, but you can override this. Db is also used in some circumstances in the calculation of the added mass/inertia components of the Morison's Equation hydrodynamic force – this is discussed in Element Diameter and Hydrodynamic Forces.

The contact diameter (Dc) is the effective diameter for contact calculations, and is relevant only when your Flexcom analysis includes guide surface contact, line clashing or pipe-in-pipe contact. Flexcom uses this input to determine when contact occurs. Explicit specification of a contact diameter is not possible with the Mooring Line format, rather Dc is automatically set equal to the largest of Db, Dd or Do.

Geometric Properties for Truss Elements

USER INPUTS

The following user inputs are used to assign geometric property data to Truss Elements.

- **Axial Stiffness**: Most simulations typically use a linear elastic material model, where the the governing input is a linear axial stiffness (EA). Alternatively you may specify a non-linear relationship for axial stiffness EA, which relates axial force and axial strain.
- **Mass**: Mass is input per unit length. This is is a measure of the inertia of the structure with respect to translation.

- **Various Diameters**

  - **Outer diameter Do:** This is a required input. Dd, Db and Dc all default to Do if not explicitly specified. Do is also used for computing the cross-section area, and it is used as the external diameter for stress calculations unless you specify otherwise.

  - **Internal diameter (Di)**: This is used for computing the buoyancy contribution of internal fluid, if present. It is also used as the inner diameter for stress calculations unless you specify otherwise. Di defaults to a value of zero.

  - **Drag diameter (Dd)**: This is the effective outer diameter for hydrodynamic force evaluation using Morison's Equation. Dd defaults to Do if not explicitly specified.

  - **Buoyancy diameter (Db)**: This is the effective outer diameter for buoyancy force calculations. Db defaults to Do if not explicitly specified. Db is also used in some circumstances in the calculation of the added mass/inertia components of the Morison's Equation hydrodynamic force – this is discussed in Element Diameter and Hydrodynamic Forces.

  - **Contact diameter (Dc)**: This is the effective diameter for contact calculations, and is relevant only when your Flexcom analysis includes guide surface contact, line clashing or pipe-in-pipe contact. Flexcom uses this input to determine when contact occurs. Dc defaults to the largest of Db, Dd or Do if not explicitly specified.

RELEVANT KEYWORDS

- `*GEOMETRIC SETS` is used to assign geometric properties to element sets (see OPTION=TRUSS).

- `*FORCE-STRAIN` is used to define force-strain curves for non-linear materials.

Stress Properties

OVERVIEW

The outputs from a Flexcom analysis include bending, axial, hoop and von Mises stresses. How the program calculates these is described in detail in Force and Stress Outputs. Those calculations though use for each element the external diameter De, the internal diameter Di, the cross-section area A and the moment of inertia I. What values Flexcom selects for these is the subject of this short section.

The first point to make is that you have the option to define any or all of these so-called stress properties independently of defining geometric properties. You can define stress properties as part of your finite element model definition, or you can do so when defining data for postprocessing. (Using the parlance of keyword files, you can define stress properties either in your '`$MODEL`' data or in your '`$DATABASE POSTPROCESSING`' data.)

The following summarises what happens if you do not explicitly define any of De, Di, A or I. The program default operation depends in the case of De on whether you used the Flexible Riser Format, Rigid Riser Format or Mooring Line Format in defining geometric properties for a particular element; for the other properties it is the same regardless.

Further information on this topic is contained in the following sections:

- External Diameter De
- Internal Diameter Di
- Cross-section Area A
- Moment of Inertia I

RELEVANT KEYWORDS

There are several keywords which may be used to assign effective structural properties to element sets for use in calculating stresses.

- `*PROPERTIES` in the `$MODEL` section.
- `*PROPERTIES` in the `$DATABASE POSTPROCESSING` section.
- `*PROPERTIES` in the `$LIFETIME FATIGUE` section.

- *PROPERTIES in the $LIFEFREQUENCY section.

External Diameter De

In the case of properties defined using the Flexible Riser format, De defaults to the drag diameter Dd. For properties defined using the Rigid Riser or Mooring Line formats, De defaults to the outer diameter Do.

Internal Diameter Di

The internal diameter for stress calculations defaults to the Di value you specify in defining geometric properties (in the case of the Mooring Line format, this default will be zero).

Cross-section Area A

A always defaults to $\pi[(De)^2 – (Di)^2]/4$. This is true even if you explicitly define a value for A in inputting geometric properties using the Rigid Riser format. De and Di can have user-defined or default values (in the case of the Mooring Line format, Di will be zero).

Moment of Inertia I

I always defaults to $\pi[(De)^4 – (Di)^4]/64$. This is true even if you explicitly define a value for I in inputting geometric properties using the Rigid Riser format. De and Di can have user-defined or default values (in the case of the Mooring Line format, Di will be zero).

Coatings

THEORY

Flexcom provides a facility to apply coatings on an element set basis. This feature is useful when modelling, for example, the effect of insulation or marine growth.

Multiple coatings can be applied. Each coating is characterised by its name, type (whether internal or external), thickness, mass density, and the element set to which it applies. Note that coating names are intended for descriptive purposes only and have no effect on the internal computations.

When multiple coatings are assigned, the coatings are assumed to occur in the order specified from the base section – inwards in the case of internal coatings and outwards for external coatings. Some caution is required where there is overlap between coated element sets. If an element is present in more than one coated element set, coatings from each set will be applied, and the order will be consistent with the set definitions referenced during the definition of coatings.

The application of coatings causes an increase in mass per unit length or mass density, while stiffness properties remain unchanged. The various diameters (drag, buoyancy, outer, contact and internal) are also adjusted as appropriate. Specifically, the internal diameter is reduced where one or more internal coatings are present, and the outer diameter is increased if one or more external coating is present. The remaining diameters (drag, buoyancy and contact) are only adjusted where the relevant user specified value is exceeded by the outer diameter plus any external coatings. It is important to note that the diameters used for stress computations are unaffected by the presence of coatings, and the values specified for the base section take priority.

The option to specify coatings is independent of the geometric format option used to define the base element set, whether Flexible Riser, Rigid Riser or Mooring Line. Note however that if you are using the Flexible Riser format, then the outer diameter Do (which is immaterial in the majority of cases) becomes a required input.

RELEVANT KEYWORDS

- `*GEOMETRIC SETS` is used to assign geometric properties to element sets.
- `*COATINGS` is used to apply coatings on the basis of element set.

If you would like to see an example of how this keyword is used in practice, refer to K02 - Worked Example - Complex.

Poissons Ratio

THEORY

This section describes how Poisson’s ratio effects are modelled.

Poisson’s ratio is defined as the ratio of transverse strain (perpendicular to the applied load), to axial strain (in the direction of the applied load). In Flexcom, changes due to Poisson’s ratio and internal and external pressure are calculated using the following relationship:

$$F_A = (P_o A_o - P_i A_i)(1 - 2\nu)$$

where F_A is the axial load due to Poisson’s ratio effects, P_o and P_i are external the internal pressures respectively, A_o and A_i are the corresponding cross sectional areas of the element, and is the specified Poisson’s ratio. Note also that Poisson’s ratio may be specified on an element set-by-set basis in Flexcom, or simply specified as a global parameter applicable to all elements in the model.

RELEVANT KEYWORDS

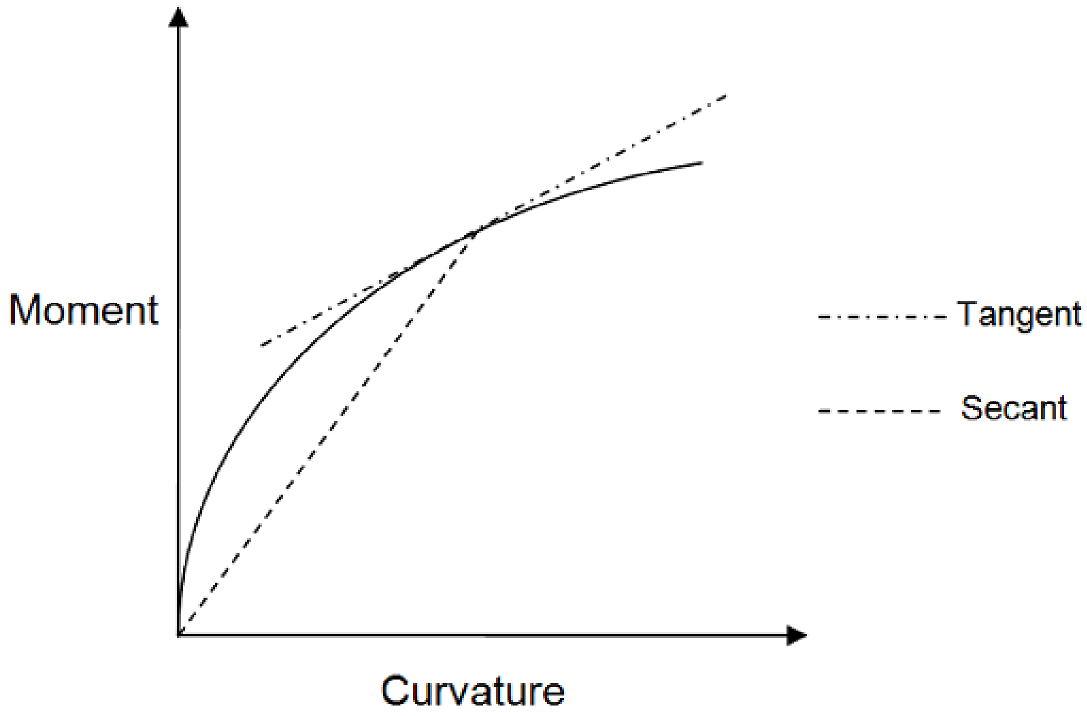
- `*GEOMETRIC SETS` is used to assign geometric properties to element sets.
- `*POISSON` is used to include Poisson's ratio effects in an analysis.

If you would like to see an example of how this keyword is used in practice, refer to C03 - Turret Disconnect.

Tangent and Secant Stiffness

THEORY

Traditionally, Flexcom has employed a tangent method for modelling non-linear materials. An alternative secant approach is now also available. This may be helpful in situations where the default tangent method leads to solution robustness issues. The option relates to non-linear structural properties, whether defined in the Flexible Riser or the Rigid Riser format, and non-linear bend stiffener elements. Both methods are illustrated graphically in the figure below for a sample moment-curvature relationship.



Secant Stiffness Method

Under normal circumstances, both methods should produce exactly the same results, but the secant method may occasionally provide additional robustness. For example, the tangent method generally requires that the slope should either monotonically increase or monotonically decrease with increasing displacement. Otherwise, there may be more than one location along the non-linear relationship which results in the same restoring force, and this may contribute to solution instability. The tangent method is still retained as the default option, as the secant method should, in theory at least, typically require a slightly larger number of iterations for solution convergence.

APPLICATIONS

- The secant stiffness option is available for Static Analysis and Time Domain Analysis only, and does not apply to Frequency Domain Analysis.
- The secant stiffness method applies to Non-Linear Elastic material properties and non-linear Bend Stiffeners.

- Many other non-linear modelling features, such as Spring Elements, Pipe-in-Pipe Connections, Seabed Embedment etc. still employ the tangent stiffness method even if secant stiffness is specified.

- P-y curves used in Soil Modelling are an exception in that they are always modelled using the secant method (where the secant approach typically provides greater numerical stability).

RELEVANT KEYWORDS

- `*GEOMETRIC SETS` is used to assign geometric properties to element sets.

- `*STIFFENER` is used to define the properties of a conical bend stiffener positioned on a flexible riser or pipe.

- `*NONLINEAR MODEL` is used to specify a modelling approach for non-linear materials.

Non-linear Material Force Term

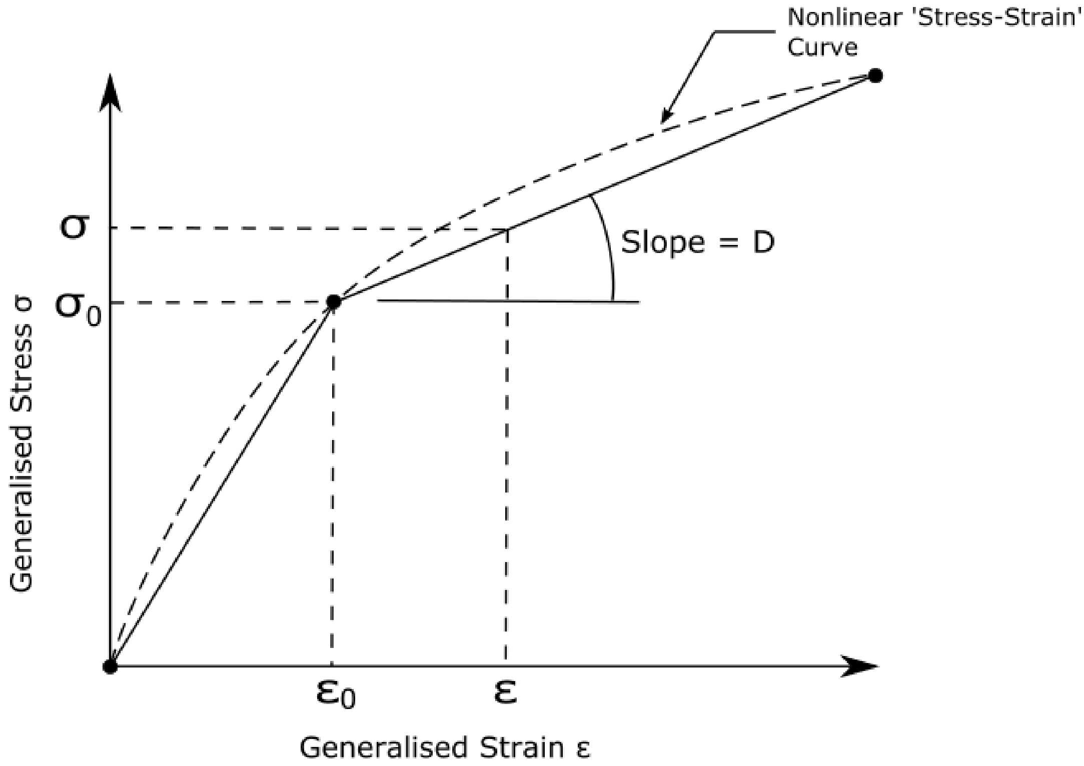
THEORY

The equilibrium equations outlined in the Finite Element Formulation section are developed for the case where all the generalised stress components are linearly related to their corresponding strain components. This linear stress-strain relationship is given by:

$$\sigma = D\,\varepsilon \qquad (1)$$

where σ and ε are generalised stress and strain components, respectively, and D is a component of the constitutive (stiffness) matrix.

Consider the more general case where the stress-strain relationship is nonlinear. Such a relationship is presented below.



Non-linear Material Model

This curve could represent any of the component stress-strain functions, such as axial force v axial strain, bending moment v curvature, or torque v twist relationships. The non-linear curve (shown dotted above) may be approximated as a series of straight-line segments. For a specified strain component, ε, the stress component, σ, is defined by the equation:

$$\sigma = \sigma_0 + D(\varepsilon - \varepsilon_0) \qquad (2)$$

where:

ε_0 is the nearest strain value below ε where the line segment intersects with the stress-strain curve.

σ_0 is the stress component corresponding to ε_0.

D is the slope of the line segment.

Equation (2) may be rearranged to give:

$$\sigma = D\,\varepsilon - h_0 \qquad (3)$$

where:

$$h_0 = D\,\varepsilon_0 - \sigma_0 \qquad (4)$$

In Flexcom terminology, h_0 is known as the non-linear material force.

Equation (4) is now in the same form as the linear relationship, with the only change being the subtraction of a new non-linear material force term from the right hand side of the equation. At any given iteration of the solution of the equilibrium equations, the non-linear material force is computed from Equation (4), and is accounted for in the Flexcom solution by putting it on the right hand side of the equilibrium equations.

RELEVANT KEYWORDS

- `*MOMENT-CURVATURE` is used to define moment-curvature curves for non-linear materials.
- `*FORCE-STRAIN` is used to define force-strain curves for non-linear materials.
- `*TORQUE-TWIST` is used to define torque-twist curves for non-linear materials.

Element Diameter and Hydrodynamic Forces

THEORY

Traditionally, Flexcom has based the calculation of both the drag and the inertia/added mass components of the Morison's Equation hydrodynamic force on the drag diameter Dd. This is accurate and reasonable for many riser systems and is retained as the default operation to maintain compatibility with older program versions. However, strictly speaking the added mass and inertia terms should be based on displaced volume (defined by the buoyancy diameter) as opposed to projected area (defined by the drag diameter). Flexcom provides an option for you to instruct the program to use Db rather than Dd for the added mass and inertia components of Morison's Equation. This choice is actually made when defining hydrodynamic rather than geometric properties, since this relates to hydrodynamic force computations. Refer to Hydrodynamic Properties for more details.

RELEVANT KEYWORDS

- `*HYDRODYNAMIC SETS` is used to assign hydrodynamic coefficients to element sets. Specifically, the `DIAMETER=` input is used to specify whether element drag (used by default) or buoyancy diameters should be used during the computation of the added mass and inertia terms in Morison’s Equation.

Compression and Buckling

INTRODUCTION

There are two main types of element of Flexcom (ignoring Special Elements such as hinges, springs, dampers, winches etc.):

- 14-DOF beam element with fully coupled axial, bending and torque forces
- 7-DOF truss element which has no structural resistance to bending or torsion

Compression and buckling are handled differently by these element types so a separate discussion is included on each.

COMPRESSION IN BEAM ELEMENTS

Euler's Critical Load

Flexcom is capable of modelling buckling of a line section under compressive axial loads, provided that the relevant beam element lengths and the solution time-step are sufficiently refined. With regards to the element length, the critical Euler load is an important parameter. This is defined as the maximum compressive load which a beam can sustain without buckling laterally.

The critical Euler load in a beam element is defined as:

$$F_{Euler} = \left(\frac{\pi^2 EI}{L^2} \right) \qquad (1)$$

where:

- F_{Euler} is the critical Euler load
- EI is the bending stiffness
- L is the element length

Identification and Resolution

Flexcom will issue a warning if the compressive load experienced in any given beam element exceeds the critical Euler load. Where this occurs, you are advised to use a refined mesh density in the region where the excessive compression is being experienced (the warning message will direct you to the relevant element number, and if you have built your model using lines, the line section in which it is located). You are free to ignore the warning message, but in doing so you accept the risk that the model is not sufficiently refined to accurately capture the localised buckling. By default, the critical load is defined by the formula above, but you can explicitly specify a different limit if you wish. It is also possible to suppress the warning messages completely, should you decide that (i) you are happy with the model definition and/or (ii) that the Euler limit is exceeded occasionally during a simulation rather than regularly.

COMPRESSION IN TRUSS ELEMENTS

Background

A truss element has zero structural bending stiffness, so in theory it has no resistance to bending and is incapable of supporting any compressive load no matter how small. However, a numerical solver such as Flexcom can predict effective compression in certain circumstances. It is important to note that bending stiffness is not solely limited to structural properties, and resistance to bending can also come from several other sources:

- The truss element formulation includes a geometric stiffness term when the element is in tension. Although this stiffness contribution should be inactive when a truss element goes into compression, it is possible that adjacent elements may still be in tension, particularly in models which are subjected to strong and rapidly changing dynamic excitation.

- A truss element experiences drag and inertial loads the same way as a beam does. These loads can inhibit the ability of a truss element to move laterally with respect to adjacent elements, particularly if the structural velocities and accelerations are significant. Buoyancy forces can also have a similar effect albeit less significant.

- Bending stiffness may be provided by a supporting structure, such as seabed contact or guide surface contact.

- Local frictional effects related to contact can provide additional resistance to perpendicular movement of a truss element.

Consider a simple catenary mooring chain which is attached to a vessel experiencing severe wave loading. The vessel will heave and pitch in accordance with the vessel RAO definitions in a decoupled analysis, or be subjected to hydrodynamic loads in a coupled analysis which induce vessel motions. During the downward stroke, the movement of the fairlead forces the mooring line downwards. Resistance in the water column is relatively low, comprising largely of axial drag forces on the line, so that much of the resistance to the mooring line inertia occurs close to the seabed. In the real world, the mooring chain will naturally collapse in this region. In the numerical environment of Flexcom, successive elements should move in different directions to reflect the local buckling. Because the mooring chain can adopt an almost infinite number of configurations on the seabed, the numerical solver is not solving for the exact configuration (which would be an indeterminate problem) but rather a representative solution. The local buckling details are immaterial as the only variable of interest is mooring line tension, but the key point is that the mathematical model is not necessarily an exact representation of the physical scenario. Intuitively one would expect that a truss element should never experience effective compression, but it is mathematically possible in a numerical model. Small values of compression may usually be ignored, but significant compression is possible given the sensitive nature of this issue.

Identification

Flexcom will automatically issue a warning message if compression is experienced by a structure which is modelled using truss elements. This message contains information regarding:

- The element set in which the compression occurs.

- The maximum amount of effective compression which is experienced over the course of the simulation.

- The element number which corresponds to the maximum value.

- The length along the element set where this element is located. This is typically located in the touchdown zone.

- The first element of the set which experiences any compressive loading. If compression occurs in the touchdown zone, some level of compression may extend all the way to the anchor point, particularly if a smooth seabed is used.

- The last element of the set which experiences any compressive loading. This is usually located in the water column, but compression can potentially extend up towards the fairlead point.

- The total length of line which experiences effective compression at any point during the simulation. This is the difference between the lengths of the first and last elements respectively along the element set.

You could also request an envelope plot of effective tension using the Database Postprocessor for a graphical overview of the extent of the compression.

Resolution

There are several options available to you if compression in truss elements is posing problems.

- **Damping**. Applying some level of stiffness proportional damping to the truss element set is generally recommended. This can greatly reduce the effects of high frequency noise in the numerical solution. However, it does not fully address the fundamental problem of converged solutions which include compression in truss elements.

- **Mesh Refinement**. Experience suggests that this is a critical modelling aspect - having a sufficiently refined finite element mesh is key to the elimination of compression. Shorter elements afford the numerical solver with greater opportunity for allowing adjacent truss elements to displace laterally with respect to each other. However, this solution comes at the expense of larger models and slower run times. It is important to note that the full structure does not need to be remeshed, and detailed refinement of a carefully selected sub-section centred around the location of maximum compression is recommended. The most critical location is obvious from the warning message and mesh refinement is very easy if your model is built using the *LINES keyword.

- **Time Stepping**. Time step size does not appear to have a major impact in this area. In fact, noisy solutions are sometimes exacerbated by the use of smaller time steps. However, every model is different and a quick sensitivity study on time step size could yield positive results for very little effort.

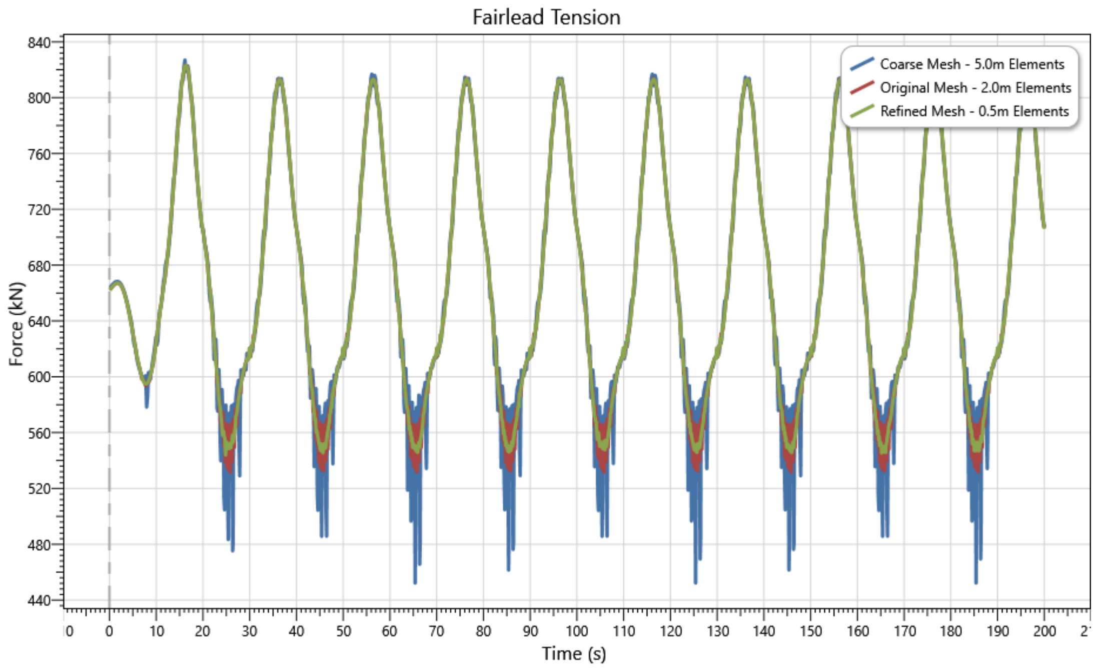
- **3D Loading**. If the model and applied loading is contained within a particular plane such as the global XY plane, then the solution is effectively a 2D one. Compression is avoided when truss elements move laterally with respect to each other (laterally in this context means perpendicular to the truss element's primary axis which runs from start node to end node). If there is no out-of-plane loading present, there is no incentive for any of the truss nodes to move out of plane. This is a quirk of the mathematical solver which can be readily avoided by including some additional loading. For example, if the simulation contains wave loading which is aligned with the plane of the model, you could include a small amount of current loading in the perpendicular direction. If the model already contains both wave and current loading, then you could introduce a slight separation between the wave and current directions, one acting slightly to the left in plan view and the other acting slightly to the right.

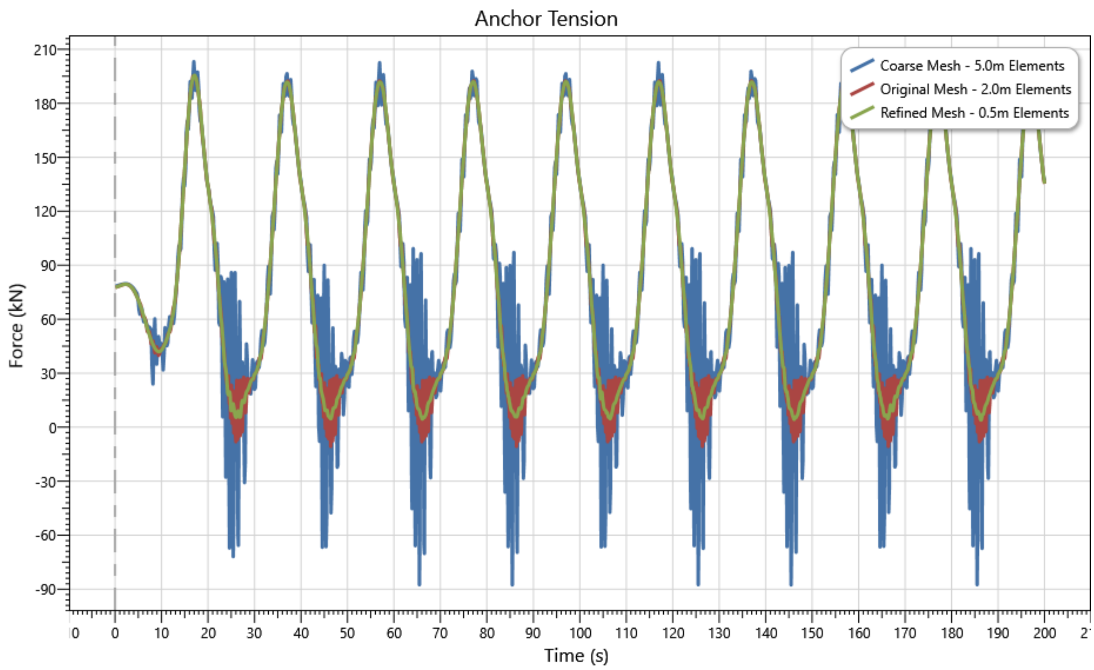
Sample Illustration

This hypothetical example illustrates the benefits of mesh refinement. The model was specifically created to demonstrate the issue of compression in truss elements. It is a simple catenary mooring line (375m long) in relatively shallow water (100m). The chain is heavy (685kg/m) and has a high extensional stiffness (3270MN). The vertical section of the line is quite steep (just 38m layback/horizontal separation between the fairlead and the touchdown point). This serves to lower the static pre-tension in the line and render it more susceptible to going slack when dynamic excitation is applied. A smooth flat elastic seabed is used. Out of plane current loading is applied (0.4m/s, uniform with depth). Sizeable vessel motions (10m heave and 5m surge) are applied to the fairlead in a dynamic simulation which is run for 10 periods, with the prescribed motions ramped up over the first period. A small amount of damping (coefficient = 0.05) is included to help reduce any high frequency noise.

The following table and accompanying figures present some key findings from the finite element mesh sensitivity study.

Parameter	Base Case	Refined Mesh	Coarse Mesh
Element Length	2.0m	0.5m	5.0m
Number of Elements	187	632	75
Run Time	12s	52s	5s
Minimum Anchor Tension	-10.7kN	4.0kN	-87.5kN
Length of line experiencing compression	278.7m	7.5m	290.0m
Maximum Fairlead Tension	813.9kN	813.3kN	816.9kN





Some observations are as follows:

- Refining the mesh density reduces the amount of effective compression in the mooring line.

- Maximum fairlead tensions are practically independent of the mesh density used i.e. the presence of effective compression in the seabed or touchdown regions is immaterial.

- Refining the mesh density creates a computational overhead which may not be justified. However, the trends may be case specific so users are advised to perform sensitivity studies using their own custom models.

RELEVANT KEYWORDS

- `*GEOMETRIC SETS` is used to assign geometric properties to element sets, and includes an option to set or disable the Euler compression check for beam elements.

FURTHER INFORMATION

Beam Elements

- Example C01 - Free Hanging Catenary considers a flexible catenary riser which experiences compression in the touchdown zone. This example is based on a publication by McCann et al., (2003).

- Example F02 - Upheaval Buckling simulates a subsea pipeline which experiences compressive axial loads leading to upheaval buckling.

- As part of their studies, O'Brien et al., (2003) consider the prediction of buckling onset and subsequent post-buckling response of an elastic column.

1.9.2.3 Hydrodynamic Properties

OVERVIEW

The various Flexcom options for defining hydrodynamic properties are discussed in the following sections:

- Constant Coefficients describes the simplest hydrodynamic property specification.

- Reynolds Number of Dependent Coefficients describes the facility to specify hydrodynamic coefficients as a function of Reynolds number.

- Advanced Topics describes advanced hydrodynamic modelling features such as Subsea Buoys, Moonpool Hydrodynamics and Drag Lift.

RELEVANT KEYWORDS

- `*HYDRODYNAMIC SETS` is used to assign hydrodynamic coefficients to element sets.
- `*MOONPOOL` is used to define the location and extent of the vessel moonpool.

Constant Hydrodynamic Coefficients

THEORY

The simplest hydrodynamic property specification is that of constant coefficients which remain the same throughout an analysis. Elements are grouped together into element sets in the usual manner, and the following five hydrodynamic coefficients for Morison's Equation are specified on a set by set basis:

- Normal Drag. The drag coefficient for the direction normal to the section, denoted C_d^n.
- Tangential Drag. The drag coefficient for the direction tangential to the section, denoted C_d^t. This entry is optional, and if omitted defaults to zero.
- Normal Inertia. The inertia coefficient in the direction normal to the section, denoted C_m^n. This entry is optional, and if omitted defaults to 2.0. Cnm is used in calculating the contribution to the inertia force normal to the element from the water particle acceleration. This term is integrated in the force vector on the right hand side of the equations of motion
- Tangential Added Mass. The added mass coefficient in the direction tangential to the section, denoted C_a^t. This entry is optional, and if omitted defaults to zero.

- Normal Added Mass. The added mass coefficient in the direction normal to the section, denoted C_a^n. This entry is optional, and if omitted defaults to ($C_m^n - 1$). Cna is used in calculating the contribution to the inertia force normal to the element from the structure acceleration. This term is integrated into the mass matrix on the left hand side of the equations of motion.

- Drag Lift Coefficient. This input is optional and by default no drag lift is modelled. Refer to Drag Lift for further information on this feature.

The hydrodynamic force formulation in Flexcom is outlined in Hydrodynamic Loading.

RELEVANT KEYWORDS

- `*HYDRODYNAMIC SETS` is used to assign hydrodynamic coefficients to element sets. Specifically, the `TYPE=CONSTANT` input is used to assign hydrodynamic coefficients to element sets where these coefficients are independent of Reynolds number.

Reynolds Number Dependent Coefficients

THEORY

Flexcom also provides a facility to specify hydrodynamic coefficients as a function of Reynolds number (Re). The same five hydrodynamic coefficients are specified on a set by set basis as before, but the terms are repeatedly input for a range of Reynolds numbers to build up a table to data. For values of Re intermediate to the values you specify, the hydrodynamic coefficients are calculated by linear interpolation.

When hydrodynamic coefficients are specified as a function of Re, Flexcom calculates the value of Re at each integration point at each iteration, using the following relation:

$$\mathrm{Re} = \frac{V_r D_d}{\nu} \qquad (1)$$

Here D_d is drag diameter and V is the kinematic viscosity of seawater. Vr represents the magnitude of the relative fluid/structure velocity in the direction normal to the element. The manner in which Vr is calculated depends on whether you invoke the Constant or Instantaneous Reynolds number computation option.

The default option is Constant, which is valid only for analyses with a single regular Airy wave or a Stokes V waves. When Constant is selected, the computation procedure is as follows. During the first two wave periods, the profile of Re with depth is based only on water particle velocity due to current. During the second wave period the value of Vr at each integration point is monitored and the maximum value for the integration point stored. These stored maxima are then used in calculating the Re profile for third wave period, during which the value of Vr is again monitored and stored; and so on until the analysis is completed.

If on the other hand you choose the Instantaneous option, hydrodynamic coefficients are computed as a function of the instantaneous value of Re. In this case Vr is the magnitude of the instantaneous relative fluid/structure velocity normal to the element at each solution time. Because Re is continuously recomputed, this procedure is valid for any seastate specification, whether regular or random or combinations of these.

One further point to note is that Re is defined in terms of the (unknown) structure velocity. For this reason, regular wave analyses with the default Constant procedure may take longer to reach steady state that would be the case were the hydrodynamic coefficients independent of Re. You should be careful to ensure steady state conditions have indeed been achieved when interpreting the results of regular wave runs that use this facility.

Taking the Pierson-Moskowitz spectrum as an example, the `FREQUENCY=AREA` input is used to request equal area discretisation, while the `FREQUENCY=GP` input is used to request geometric progression discretisation.

RELEVANT KEYWORDS

- `*HYDRODYNAMIC SETS` is used to assign hydrodynamic coefficients to element sets. Specifically, the `TYPE=REYNOLDS` input is used to assign hydrodynamic coefficients to element sets where these coefficients are a function of Reynolds number.

- `*PRINT` is used to request additional printed output to the main output file. Specifically, the `OUTPUT=REYNOLDS COF OUTPUT` option is used to request output of the instantaneous hydrodynamic coefficients as a function of time. For analyses where you specify that hydrodynamic coefficients are to be computed as a function of instantaneous Re, then for QA purposes a table may be automatically created showing the drag and inertia coefficients actually used by the program in calculating hydrodynamic forces at each integration point on each element. This output is routed to a separate output file named jobname.cof. In the case of a static analysis, the output is produced at the last solution time only. In a dynamic analysis, the output is generated at each solution time which lies within a user-specified range. The default range is from the start of the analysis to the end; if you want to change this, you do so when specifying time variables. The data is in tabular format, in eight columns, these being the following for each location:

    1. Element number
    2. Integration point number
    3. Instantaneous elevation above the mudline
    4. Instantaneous water particle velocity
    5. Instantaneous structural velocity
    6. Instantaneous relative velocity
    7. Corresponding normal drag coefficient
    8. Corresponding normal inertia coefficient

Advanced Topics

This section contains information on:

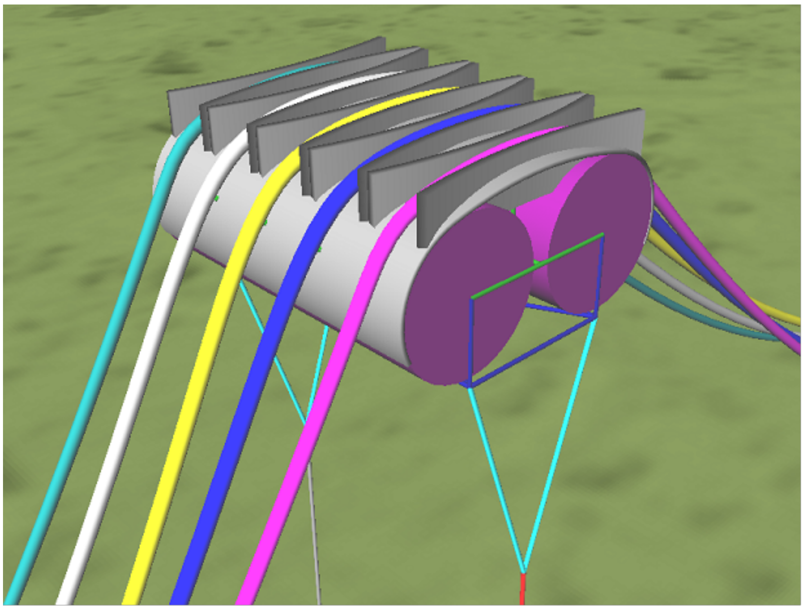
- Subsea Buoys describes the buoy structure modelling facility, which facilitates the accurate modelling of subsea buoyancy structures.
- Moonpool Hydrodynamics describes modelling the hydrodynamic loading on sections of the structure that are contained within a Spar hull or moonpool.
- Drag Lift describes option to model drag lift forces.

Subsea Buoys

OVERVIEW

Many offshore structures have subsea buoyancy structures incorporated into their design. How these structures are modelled can have a significant effect on the outcome of analyses. This relates in particular to the hydrodynamic properties of the buoy, since modelling the correct buoyancy force is relatively straightforward.

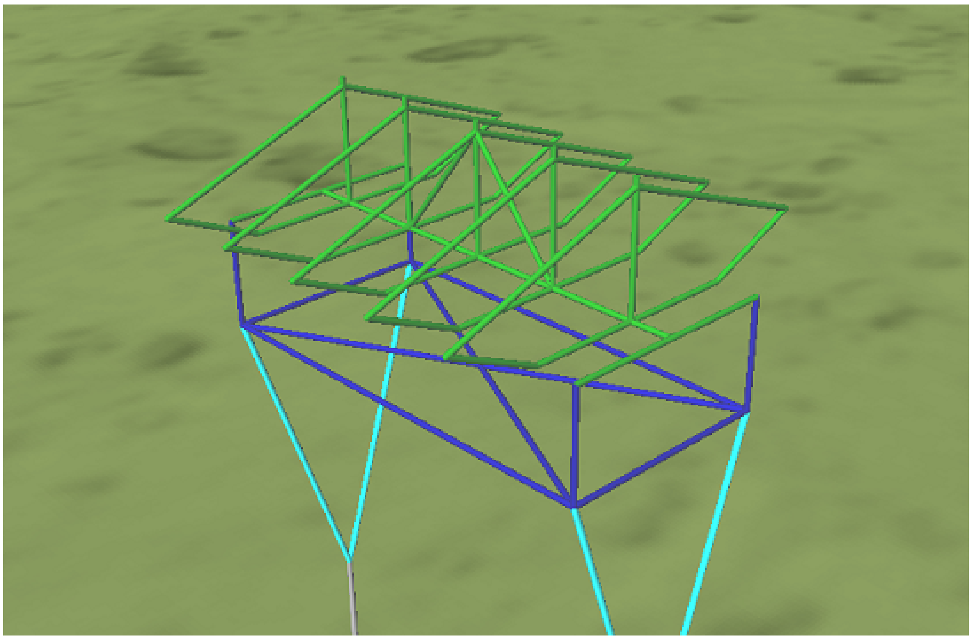
To facilitate the accurate modelling of subsea buoyancy structures, such as a subsea buoy/arch arrangement in a flexible riser lazy-S configuration, Flexcom incorporates a Buoy Structure facility. You use this facility to assign the properties that define the hydrodynamic response of the buoy to a set of elements that collectively represent the buoy.



Typical Subsea Buoy

A typical subsea buoy is shown in the figure above. Obviously the hydrodynamic properties vary depending on direction. For this reason the buoy hydrodynamic coefficients are normally specified separately for each of three directions (one vertical, two horizontal). Furthermore, for each direction the data is often specified as three products, namely i) C_dA_d (drag coefficient by frontal area), for the hydrodynamic loading due to relative fluid/buoy velocity; ii) C_mV_{in} (inertia coefficient by reference volume), for the hydrodynamic loading due to water particle acceleration; and iii) C_aV_{in} (added mass coefficient by reference volume), for the hydrodynamic loading due to buoy acceleration. This means a subsea buoy may be characterised by nine hydrodynamic coefficients or products.

The basic modelling procedure in Flexcom is that a subsea buoy is modelled using an assemblage of beam elements. All of these elements are then combined into one or more element sets in the usual way. Finally, the buoy hydrodynamic properties are specified for each buoy structure element set. The figure below shows a sample assemblage of beam elements.



Assemblage of Beam Elements

What happens in the actual Flexcom analysis is that the program computes the overall loading on the buoy, and then distributes the forces to the constituent elements. In this way phase differences between wave forces on different parts of the buoy, and also the changes in the buoy hydrodynamic properties as the buoy displaces and rotates, are accurately modelled.

It should be noted that each element of the buoy structure is a normal beam element in its own right, and must have structural (geometric) properties assigned to it in the usual way. Typically the beam elements are rigid, and broadly speaking, approximate the mass distribution around the structure.

CHOICE OF AXIS SYSTEM

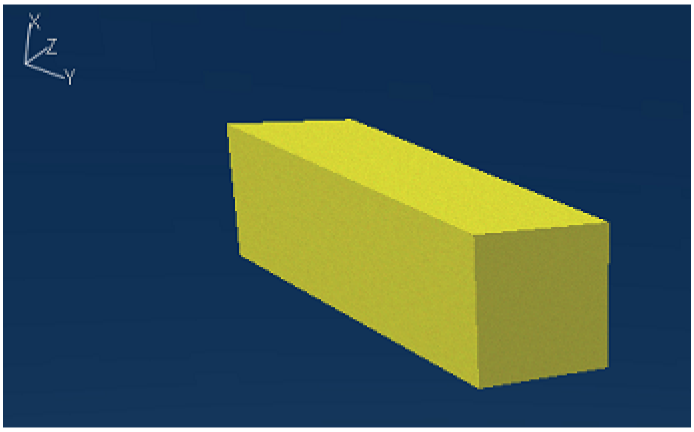
If the primary axes of the buoy structure happen to be aligned with the global XYZ axes of the software model, then the Global axis system option should be used. However, in many cases it will be necessary to define a Local axis system, which describes how the buoy structure is initially aligned with respect to the global XYZ coordinate system. Typically the local X axis will coincide with the global X axis, as the buoy will be horizontal initially, and local Y and Z axes will reflect the orientation of the buoy.

With regard to the treatment of axis system, improvements to the software, first introduced in Flexcom 8.3.5, are noteworthy. In earlier versions, the user did not explicitly define the local axis system. Rather the program assumed it to be coincident with the local axis system of the buoy structure elements. Given that several elements are typically contained within the buoy structure element set, the potential for conflicting definitions was ever present. So the hydrodynamic force terms could only be modelled correctly if (i) a single element was used to model the buoy structure, or (ii) several elements aligned in a uniform direction were used. Potential discrepancies have since been eliminated via the explicit specification of the local axis system.

APPLICATION OF FORCE TERMS

Hydrodynamic forces on the buoy structure are computed via Morison’s Equation. Three separate force terms (i.e. drag, added mass and added inertia) are computed in three separate directions, based on the user-specified hydrodynamic coefficients, and the instantaneous structural and fluid motions. These forces are computed with respect to the local buoy structure axis system, and then rotated into the global axis system. Even if the buoy is initially aligned with the global axis system (i.e. the Global specification option is used), some rotation is usually necessary as the buoy structure will displace and rotate over time subject to the hydrodynamic excitation induced by environmental loading.

The total forces are distributed evenly amongst the constituent elements, based on their projected lengths onto the local XY, YZ & XZ planes of the buoy structure. Consider drag loading in the local X direction as an example. Each element will be projected onto the local YZ plane, where it will be assigned a certain projected length. This length, expressed as a percentage of the total sum of all the projected lengths, governs the fraction of the total force which is applied to any given element. For the simple case of a cuboid shown in the below figure, the total force in the vertical (X) direction is distributed amongst the horizontal elements which border the upper and lower faces of the cuboid. The vertical elements which border the side faces receive no loading in this direction.



Drag Loading on Cuboid Shape

RELEVANT KEYWORDS

- `*HYDRODYNAMIC SETS` is used to assign hydrodynamic coefficients to element sets. Specifically, the `TYPE=BUOY` input is used to assign hydrodynamic properties to a set of elements that collectively model a subsea buoy.

If you would like to see an example of how this keyword is used in practice, refer to C02 - Multi-Line Flexible System.

Moonpool Hydrodynamics

THEORY

Flexcom provides comprehensive modelling capabilities intended specifically for the analysis of Spar or DDCV risers. One of these is the ability to specify that hydrodynamic loads on sections of riser contained within the Spar moonpool are based on the water particle velocities and accelerations within the moonpool. Flexcom assumes that the vessel moonpool entrains a body of water, that is, that the water within the vessel moonpool moves with the vessel. So water particle velocities and accelerations used in calculating Morison's Equation forces are calculated from the vessel motions, rather than from the ambient wave field. The moonpool is assumed to have a square cross-section and extends from a user-specified level up to the Mean Water Line.

There are two main steps involved in modelling the hydrodynamic loading on sections of the structure contained within a moonpool. The first step is to define the volume contained within the vessel moonpool at the initial static analysis stage. This is done by specifying the following data:

(i) The vessel with which the moonpool is associated. The volume enclosed by the moonpool moves with this vessel throughout the analysis, and the water particle velocities and accelerations within the moonpool are calculated from the motion of this vessel.

(ii) The location of the moonpool at the start of the analysis. This is specified by inputting the global coordinates of a particular point at the bottom of the moonpool when defining model geometry and properties.

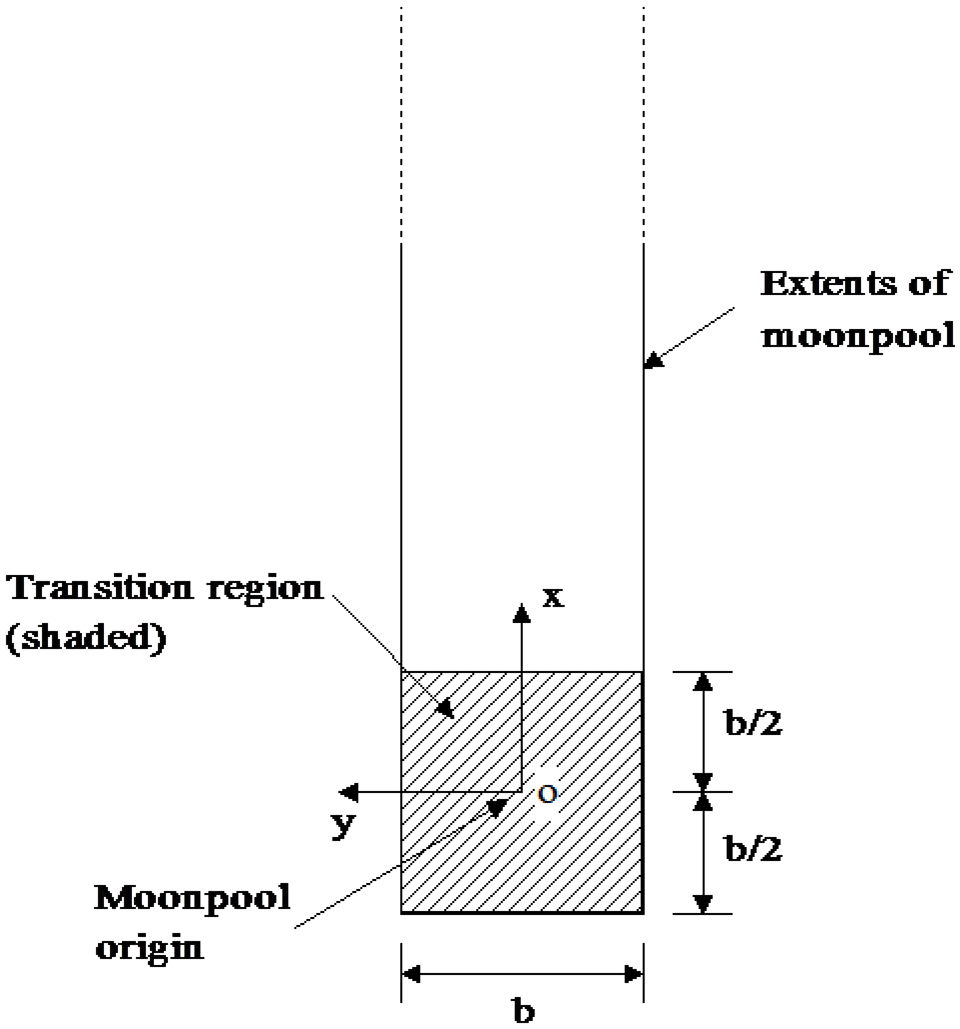
(iii) The initial yaw orientation of the moonpool. As the moonpool is assumed to have a square cross-section, this entry simply defines the orientation of the moonpool in the global-YZ plane. This is usually the same as the yaw orientation of the relevant vessel.

(iv) The width of the moonpool.

The second step in modelling moonpool hydrodynamics is to identify elements of the structure that may be subject to hydrodynamic loading within the vessel moonpool.

What happens when elements are assigned moonpool hydrodynamic coefficients is that the program checks at each solution time whether each integration point on a given element is within the volume enclosed by the vessel moonpool at that particular time, or if it is in the so-called transition region, or if it is completely outside the influence of the vessel moonpool. If it is within the volume enclosed by the moonpool, then the water particle velocities and accelerations used in calculating the Morison’s Equation force at the integration point are calculated from the vessel motions. If the integration point is outside the influence of the moonpool, the water particle velocities and accelerations are calculated from the ambient wave field. If the integration point is within the transition region, then the water particle velocities and accelerations are interpolated linearly from those in the moonpool and those in the wave field. In this way, the program automatically accounts for elements that may move in and out of the vessel moonpool during the course of an analysis.

In order to avoid a discontinuity in the water particle velocities and accelerations at the boundary between the volume enclosed by the moonpool and the ambient wave field (which could result in a discontinuity in the hydrodynamic loads applied to the structure), a transition region is defined at this location. Across this transition region, the water particle velocities and accelerations vary linearly from those in the moonpool to those in the ambient wave field. The transition region extends from a distance equal to one-half the moonpool width below the moonpool origin to one-half the moonpool width above the moonpool origin, as shown in the below figure.



Volume Enclosed by Moonpool

A final point to note is that only constant hydrodynamic coefficients may be specified for moonpool hydrodynamic sets; hydrodynamic coefficients may not be specified as a function of Reynolds number.

RELEVANT KEYWORDS

- `*MOONPOOL` is used to define the location and extent of the vessel moonpool.
- `*HYDRODYNAMIC SETS` is used to assign hydrodynamic coefficients to element sets. Specifically, the `TYPE=MOONPOOL` input is used to assign hydrodynamic coefficients to elements of the structure that may be subjected to hydrodynamic loading within a vessel moonpool.

If you would like to see an example of how these keywords are used in practice, refer to A02 - Spar Production Riser.

Drag Lift

Flexcom provides an option to model drag lift forces. This is most typically used to compute lift forces on a pipeline from current and wave caused by the asymmetry of flow around the pipeline in an on-bottom or near-bottom configuration. The magnitude of the drag lift force is given by the following relation:

$$F_L = \frac{1}{2}\rho_W C_L D_d \left(v_W^N - v_P^N\right)^2 \qquad (1)$$

where ρ_W is the density of the external fluid (typically seawater), C_L is the drag lift coefficient, D_d is the drag diameter, v_W^N and v_P^N are the normal components of fluid and structure velocity with respect to the orientation of the structure (the axis of the element). The lift coefficient C_L may be specified as a constant value or expressed as a function of the ratio of the gap between structure and seabed to the contact diameter of the element.

The direction of the lift force is perpendicular to the element axis and to the water particle velocity. Specifically, the lift force acts parallel to the normal ($\overline{n}$) of the plane defined by the vector of the local convected x axis of the element ($\overline{x}$) and the vector of the water velocity ($\overline{v_W}$). The normal ($\overline{n}$) is he normalised cross-product $\overline{x} \times \overline{v_W}$.

The sense of direction for the lift force is defined as follows:

(i) If the angle between the normal ($\overline{n}$) and the global X vector is strictly less than 90°, then the force has the same direction as the normal ($\overline{n}$).

(ii) If the angle between the normal ($\overline{n}$) and the global X vector is strictly more than 90°, then the force is in the opposite direction to the normal ($\overline{n}$).

(iii)If the angle between the normal ($\bar{n}$) and the global X vector is exactly 90°, then the force direction is given by the direction of ($\bar{n}$).

1.9.2.4 Special Element Types

OVERVIEW

Flexcom is based on a 3D hybrid beam-column element with fully coupled axial, bending and torque forces, for reliable and accurate modelling of slender offshore structures. A truss element is also available which is better suited to structures with very low structural bending stiffness. Additionally, a wealth of practical modelling features means that almost any offshore system can be simulated. Specialised elements include:

- Springs describes the spring element facility.
- Hinges and Flex joints describes these two articulation-type elements.
- Tapered Stress Joints describes the facility to model a section of linearly increasing diameter.
- Bend Stiffeners describes the facility to model a bend stiffener such as is typically used to prevent excessive bending at the termination of a flexible riser.
- Dampers describes the facility to directly model local damping effects.
- Winches describes these unique beam elements whose lengths can vary during an analysis.
- Point Masses and Point Buoys describes these two features which simulate concentrated mass. The point buoy offers comprehensive hydrodynamic modelling options.
- Drag Chains describes the facility to simulate the presence of chains connected to a near-bottom structure.

RELEVANT KEYWORDS

- `*SPRING ELEMENT` is used to define the stiffness characteristics of spring elements.

- *NODE SPRING is used to specify the stiffness and direction of node springs.
- *HINGE is used to specify hinge elements.
- *FLEX JOINT is used to define flex joint elements.
  - *TAPER is used to assign properties to a set of elements that collectively comprises a tapered stress joint, typically in a model of a rigid riser.
  - *STIFFENER is used to define the properties of a conical bend stiffener positioned on a flexible riser or pipe.
- *NONLINEAR STIFFENER is used to define a stress-strain curve for a non-linear bend stiffener.
- *DAMPER is used to define damper elements.
- *DAMPER DATA is used to define time-varying coefficients for use with damper elements.
- *WINCH is used to define winch elements.
- *MASS is used to specify a point mass or a point rotary inertia.
- *POINT BUOY is used to define point buoys and their associated hydrodynamic properties.
- *DRAG CHAIN is used to include drag chains in the structural model.

Spring Elements

THEORY

A spring element provides a restoring force which is directly proportional to the relative movement of its end nodes. The magnitude of the force is defined according to the following relationship:

$$F = -k\,x \qquad (1)$$

where F is the restoring force, k is the spring stiffness and x is the relative displacement between the end nodes of the spring element (the change in length of the spring element from its original length).

The spring may be characterised by a linear (constant) stiffness value, or defined in terms of a non-linear force-displacement relationship. The line of action of the spring is always aligned with the element axis, and the restoring force is such that it opposes the change in length – so the spring is in tension when the element extends, and in compression when the element contracts. The behaviour of a linear spring element is the same in tension and contraction, but the non-linear spring element offers complete generality in this regard.

In terms of directionality, the convention in Flexcom is that spring extensions constitute positive displacements. Spring compression (i.e. element contraction) is considered a negative displacement. This convention should be borne in mind when defining non-linear force-displacement relationships for non-linear springs.

RELEVANT KEYWORDS

- `*SPRING ELEMENT` is used to define the stiffness characteristics of spring elements.

If you would like to see an example of how this keyword is used in practice, refer to H03 - Articulated Stinger.

- `*NODE SPRING` is used to specify the stiffness and direction of node springs.

Hinge and Flex Joints

INTRODUCTION

Hinges and flex joints are considered as separate entities in Flexcom. A hinge in an articulation with typically a zero or low rotational stiffness value, while a flex joint has a finite linear or nonlinear stiffness. A hinge has zero length and acts at a single point in the model, whereas a flex joint has a short but non-zero length. A hinge constrains the displacements of its two end nodes to be exactly the same, while a flex joint includes a physical separation between its ends nodes. If a hinge is assigned a rotational stiffness, it must be a linear value. Flex joints accommodate a linear stiffness value or a non-linear moment-angle relationship. Hinges are massless (although you could include a point mass in conjunction with a hinge) while flex joints may have associated weights in air and in water.

The following table summarises the differences between hinges and flex joints.

Feature	Hinge	Flex Joint
Length	Zero	Short
Stiffness	Zero or low	Linear or non-linear
Mass/buoyancy	Zero	Wet and dry weights
Finite element modelling	Single point	Short beam element

HINGE JOINTS

All hinge elements should have zero length in your Flexcom model (the program will issue an appropriate warning message if any hinge element has a non-zero length). This ensures that the displacements of its two end nodes are exactly the same. Rotational stiffness, if specified, acts at a single point in the model, ensuring a pure hinge is modelled. The rotational stiffness acts equally in all directions.

FLEX JOINTS

Flex joints are modelled in Flexcom using (typically short) beam elements. Using a combination of the user-specified rotational stiffness and the length of joint, the software automatically computes an appropriate bending stiffness for a beam element used to represent the flex joint. The bending moment across the flex joint must be replicated by the moment across the equivalent beam element, therefore:

$$M = k\theta = EI\kappa \quad (1)$$

where M is the applied moment, k is the rotational stiffness of the flex joint, θ is the rotation (in degrees) across the joint, EI is the bending stiffness of the equivalent beam element, and κ is the curvature of the element.

Considering a segment of a circle, the radian measure of an angle is defined as:

$$\theta = L / r \quad (2)$$

where θ is the angle (in radians), L is the arc length subtended by the angle, and r is the radius of the circle. Assuming L is the length of a curved beam element, relating bend radius to curvature, converting radians to degrees, and substituting for θ in the first Equation above gives:

$$EI = kL \qquad (3)$$

The torsional stiffness of the beam element used to represent the flex joint is assigned the same value as the bending stiffness. The axial stiffness is set equal to the larger axial stiffness of the beam elements connected by the flex joint.

RELEVANT KEYWORDS

- `*HINGE` is used to specify hinge elements.
- `*FLEX JOINT` is used to define flex joint elements.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser.

Tapered Joints

OVERVIEW

A tapered stress joint or TSJ is assumed to be conical in shape. The specification of geometric properties for a set of elements which together comprise a TSJ is reasonably similar to that of a set of standard beam elements when the Rigid Riser format is used. Specifically, you input Young's modulus, shear modulus, mass density etc. However, instead of specifying single (constant) values for the various diameters (drag, buoyancy etc.), you actually enter two values for each diameter, one for the start of the tapered stress joint and one for the end. The diameters vary linearly from the start to the end of the TSJ.

TAPERED JOINT MODELLING

It is important to note that all of the properties of the elements of the tapered stress joint set are specified using tapered stress joint data - no properties are assigned to these elements using the normal facilities for beam-column elements. This reflects the reality that a tapered stress joint is normally an integral part of a riser system (as opposed to a bend stiffener, which provides additional stiffness to a section of riser). The calculation of the standard beam-column properties from the tapered stress joint data is straight-forward. Note that as all beam elements in Flexcom have uniform diameter along their length, tapered stress joints are in fact represented using a series of beam elements of varying diameter. External diameters at the start and end of an element of the set are found by linear interpolation, and these are averaged to give an equivalent external diameter for the element. The process is repeated to find equivalent internal diameter, drag diameter and buoyancy diameter values if necessary. The average external and internal diameter values for an element are used to calculate A, I and J, and these are combined with the user-specified values for Young’s modulus and shear modulus to get bending, axial and torsional stiffnesses for the element. Multiplying the material mass density by A and J in turn gives respectively the mass per unit length and the polar inertia per unit length. From this point on the tapered stress joint element is treated exactly like any other beam-column element.

RELEVANT KEYWORDS

- `*TAPER` is used to assign properties to a set of elements that collectively comprises a tapered stress joint, typically in a model of a rigid riser.

If you would like to see an example of how this keyword is used in practice, refer to A02 - Spar Production Riser.

Bend Stiffeners

OVERVIEW

Bend stiffeners are typically positioned at the termination of a flexible riser to prevent excessive bending occurring. Like tapered stress joints, they are assumed to be conical in shape. The user inputs include the external diameters at the two ends of the stiffener, the Young's modulus of the stiffener material, the total mass of the stiffener, and the stiffener internal diameter. These properties are assigned to a named element set, and the elements comprising the set are defined separately. One important point to note about a bend stiffener is that it provides additional bending resistance to the elements over which it is positioned. So the elements of the bend stiffener set must also be assigned properties appropriate to normal beam-column elements. The order in which the elements are specified in defining the bend stiffener element set is very important. The first element specified is assumed to be at the free end (smaller diameter) of the bend stiffener and the last element is assumed to be at the fixed end (larger diameter). Intermediate elements must be specified in order from free end to fixed end.

BEND STIFFENER MODELLING

Flexcom uses the specified bend stiffener properties as follows. The external diameters at stiffener interior nodes are found by linear interpolation between the stiffener start and end values. If the external diameters at the ends of a particular stiffener element are d_{oj} and d_{ok}, and the stiffener internal diameter is d_i, then the moments of inertia at the element ends I_j and I_k can be calculated in the usual way - for example:

$$I_j = \frac{\pi}{64}(d_{oj}^4 - d_i^4) \qquad (1)$$

An average value for the element I_{av} is calculated as:

$$I_{av} = \frac{(Ij + Ik)}{2} \qquad (2)$$

The user-specified bending stiffnesses EI_{yy} and EI_{zz} for the element are augmented by the addition of $E_{BS}I_{av}$, where E_{BS} is the Young's modulus of the stiffener material. The bend stiffener is assumed to provide no additional axial or torsional stiffness to the riser. Note that you have the option to specify a non-linear material model for your bend stiffener, which is defined in terms of a direct stress-strain curve. If you invoke that option, then the value of E_{BS} will be an instantaneous rather than a constant value.

In earlier versions of Flexcom, when you specified a mass for a bend stiffener this mass was distributed between the elements protected by the stiffener, which therefore had a mass increased from that specified by the user. Likewise the buoyancy diameter for these elements was increased to add the volume of water displaced by the bend stiffener to the volume displaced by the riser elements themselves. The current modelling approach differs in two respects. Firstly, by default (that is, unless you specify otherwise), the mass and buoyancy of the bend stiffener are not now included in the analysis loading. The rationale for this is that these loads are rarely transferred to the riser itself, but are in fact carried by an end fitting. You do though have the option to override this: if you exercise this option, the second change in program operation occurs. Instead of the weight and buoyancy of the stiffener being distributed over the enclosed elements, these are now applied as point loads at the stiffener end where the outer diameter is a maximum. (In fact Flexcom positions a point buoy with zero hydrodynamic coefficients at this location, which is effectively the same thing.) The program does apply one check as follows: if you specify a stiffener mass but do not change from the default program mode of operation, then you are warned that the weight of the stiffener will not in fact be included in the analysis.

COMPARISON WITH TAPERED STRESS JOINT

To summarise, a bend stiffener specification is similar to that of a tapered stress joint described earlier, but differs in a couple of aspects:

(i) The properties assigned to the bend stiffener are additional to the properties defined for the standard beam element which it encloses. Flexcom automatically increases the bending stiffness of the elements where the bend stiffener is positioned, using the relevant bend stiffener properties.

(ii) Bend stiffeners can be either linear or non-linear. Young's Modulus can be defined using a single (constant) value, or characterised in terms of a non-linear direct stress-strain relationship.

RELEVANT KEYWORDS

- *STIFFENER is used to define the properties of a conical bend stiffener positioned on a flexible riser or pipe.
- *NONLINEAR STIFFENER is used to define a stress-strain curve for a non-linear bend stiffener.

If you would like to see an example of how these keywords are used in practice, refer to C01 - Free Hanging Catenary.

Damper Elements

OVERVIEW

Damper or dashpot elements can be used to directly model local damping effects. Example applications would be in the analysis of a drilling riser model that includes a crown heave motion compensator, or the damping of SCR motions by soil in the touchdown zone. The damper element is completely defined by its damping coefficients (constant, linear and quadratic options are available), and there are no associated mass or stiffness inputs.

Damper elements can be applied in a translational or rotational context.

Note that damper elements are fully-fledged elements of the finite element model, whose geometry (nodal coordinates and connectivity) must be defined in any of the standard ways.

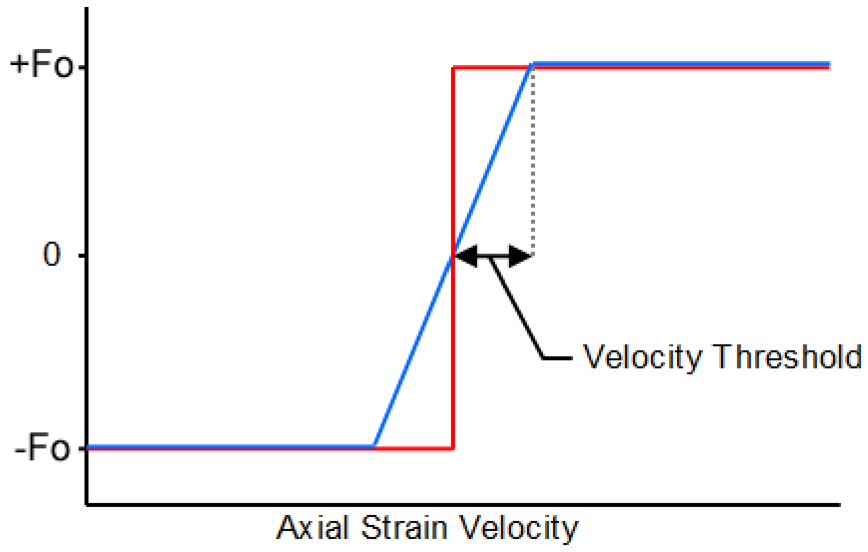
TRANSLATIONAL DAMPER

The axial force exerted by a translational damper element, F, is defined as follows:

$$F = -(F_0 + C_1 v + C_2 v^2) \qquad (1)$$

where v is the relative velocity between the element end nodes in the axial direction of the element, F_0 is the constant damping force, and C_1 & C_2 are the linear and quadratic damping coefficients respectively. Each component term may be specified as a single (constant) value, or as a series of values which depend on time or velocity. F_0, C_1 & C_2 can each vary as a function of time, and C_1 can also vary as a function of velocity.

It is possible to control the application of the constant damping force (F_0 term) by specifying a velocity threshold. By default no ramping is applied, so the damping force is applied in full at all times. The significance of the velocity threshold is illustrated in the figure below. Rather than switching instantaneously between positive and negative force values, the constant force term is gradually ramped up over a finite range of relative velocity. This tends to smooth out the applied damping forces and eliminates the possibility of a dramatic alternation between positive and negative force terms for small extensions and contractions of the damper element.



Velocity Threshold

Damper elements are sometimes used to simulate power extraction. In this context, Flexcom provides post-processing options for damper power. The power produced in each damper element is calculated as follows:

$$P = F*(-v) \qquad (2)$$

See the Force Variable Input section for details on how to output damper power.

ROTATIONAL DAMPER

For rotational damper elements, the moment exerted by the element, M, is defined as follows:

$$M = -(C_1 v) \qquad (3)$$

where ν is the relative rotational velocity between the element end nodes and C_1 is the linear damping coefficient.

RELEVANT KEYWORDS

- `*DAMPER` is used to define damper elements.
- `*DAMPER DATA` is used to define coefficients for use with damper elements.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser.

Winch Elements

OVERVIEW

Winch elements are normal beam-column elements which have the unique property that their lengths can vary during an analysis. In a dynamic analysis, the variation in length may be defined in terms of a standard velocity profile or an arbitrary payout as a function of time. The operation in a static analysis is less complex, with an overall change in length being applied linearly from analysis start time to analysis end time.

The standard velocity profile is defined in terms of a maximum winch velocity and a winching time sequence. The time sequence consists of (i) a ramp-up time when winch velocity is increased from zero to the maximum value; (ii) a time during which the velocity remains at this maximum, and (iii) a ramp-down time when the velocity returns to zero. In the case of an arbitrary payout definition, a time history of extension/contraction is read directly from an external ASCII file.

Winch elements can be used, for example, in pipelaying applications, in simulating the transfer of an SCR from a lay vessel to a TLP or semi-sub.

RELEVANT KEYWORDS

- `*WINCH` is used to define winch elements.

If you would like to see an example of how this keyword is used in practice, refer to H04 - Pipe Laying.

Point Masses and Point Buoys

OVERVIEW

A point mass is used to model a concentrated mass or rotational inertia. While a point buoy is a reasonably similar entity, it is more complex in a number of respects:

(i) The buoy is characterised by a weight in air and a total buoyancy force, and the buoyancy force is activated/deactivated depending on the instantaneous location of the buoy with respect to the ambient water surface.

(ii) Different rotational inertias can be specified about the individual local buoy x-, y- and z-axes.

(iii) Hydrodynamic loading on the buoy can be modelled, and comprehensive specification options (similar to that of the Buoy Structure facility) are provided.

POINT BUOY HYDRODYNAMICS

The point buoy facility might typically be used to model the effects of a subsea buoy in a flexible riser model. It is intended as an alternative to modelling the buoy explicitly using beam elements – a point buoy is specified at a node of the discretisation, but is not an integral part of the structure model.

Typically the hydrodynamic properties of a subsea buoy are very different in the vertical and horizontal directions. For this reason the buoy hydrodynamic coefficients are normally specified separately for each of three directions (one vertical, two horizontal). Furthermore, for each direction the data is often specified as three products, namely i) C_dA_d (drag coefficient by frontal area), for the hydrodynamic loading due to relative fluid/buoy velocity; ii) C_mV_{in} (inertia coefficient by reference volume), for the hydrodynamic loading due to water particle acceleration; and iii) C_aV_{in} (added mass coefficient by reference volume), for the hydrodynamic loading due to buoy acceleration. This means a subsea buoy may be characterised by nine hydrodynamic coefficients or products.

Additionally you can also define rotational hydrodynamic coefficients for the point buoy. Similar to the translational terms, coefficients for each direction are specified as three products, namely i) C_dDMA (product of drag coefficient and drag moment of area); ii) C_mHI (product of inertia coefficient and hydrodynamic inertia); and iii) C_aHI (product of added mass coefficient and hydrodynamic inertia). Rotational hydrodynamics are applied with respect to the global axis system by default, but a facility is also provided to define a local axis system and use this as a reference for the rotational hydrodynamics.

RELEVANT KEYWORDS

- `*MASS` is used to specify a point mass or a point rotary inertia.
- `*POINT BUOY` is used to define point buoys and their associated hydrodynamic properties.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser (`*MASS`) and A02 - Spar Production Riser (`*POINT BUOY`).

Drag Chains

THEORY

A drag chain is used to simulate the presence of a chain connected to a structure, where a portion of the chain lies along the seabed. The chain itself is not modelled physically using standard elements, rather gravity and lift forces for the drag chain are computed and applied as point loads at the specified node of the structure. The gravity load is calculated from the length of drag chain not lying on the seabed. This length is calculated from the elevation above the seabed of the chain node (the chain is assumed to hang vertically) and the chain length, which is a user input. The lift force on each chain is calculated from the relative fluid/structure velocity at the chain node and the specified lift coefficient. The lift force is computed as follows:

$$F_L = \tfrac{1}{2} C_L \rho_w u^2 l^2 \qquad (1)$$

where F_L is the total lift force, C_L is the specified drag chain lift coefficient, ρ_w is the mass density of seawater, u is the magnitude of the horizontal component of relative fluid/structure velocity at the drag chain node, and is the length of drag chain not lying on the seabed.

RELEVANT KEYWORDS

- *DRAG CHAIN is used to include drag chains in the structural model.

1.9.2.5 Pipe-in-Pipe

OVERVIEW

A useful feature in Flexcom is the ability to model pipe-in-pipe configurations. Internal and external sections are modelled separately, and interact with each other by means of special connections simulating linear and non-linear resistance to relative motion as well as internal fluid hydrodynamic forces. Although the capability is called ‘pipe-in-pipe’, it can also be used in the modelling of ‘pipe-on-pipe’ or piggy-backed systems - there is no requirement for one section to be physically contained within another.

When referring to pipe-in-pipe sections, the terms ‘outer’ and ‘inner’ are used regularly. This makes sense as the purpose of the *PIP SECTION keyword is to identify which pipe sections are contained within other pipe sections. When referring to pipe-in-pipe connections, the terms ‘primary’ and ‘secondary’ are used, reflecting the fact the connections may represent either pipe-in-pipe or pipe-on-pipe configurations (indeed the same *PIP CONNECTION keyword is used to define both).

- Model Set-up Guidelines provides guidance to assist new users with model creation and help to avoid common pitfalls.

- Pipe-in-Pipe Sections describes how one pipe is contained within another, and explains the implications of this for buoyancy and hydrodynamic forces.

- Standard Connections outlines the operation of the standard pipe-in-pipe contact model at a high level.

- Sliding Connections explains the additional functionality offered by sliding connections, and suggests scenarios for which it is appropriate.

- Contact Modelling discusses the operation of the pipe-in-pipe contact modelling algorithm in detail.
- Hydrodynamic Forces outlines the main fluid forces modelled by Flexcom (including buoyancy, hydrodynamic and internal fluid) and how these are affected by the presence of pipe-in-pipe configurations.

APPLICATION

Flexcom's pipe-in-pipe facility has been used extensively by Wood engineering services division. For example, Flexcom was an integral part of the FEED and detailed engineering design phases for a Bundled Hybrid Offset Riser (BHOR) configuration deployed offshore west of Africa, and the software was also central to the complicated installation process involving shallow water tow out and deep water upending.

Liu et al. (2013) compare Flexcom and Abaqus for a pipe-in-pipe configuration, outlining independent research conducted by IntecSea and FloaTEC. A static in-situ analysis of an SCR within a pull-tube, and a dynamic pull-through analysis are considered. The results show very close agreement for the static case, and strong agreement for the dynamic case (including sliding pipe-in-pipe contact) in terms of the bending response. Flexcom does not yet model frictional effects in the pipe-in-pipe model, but this was not an issue in this case study, as bending is typically the principle design driver for pull-tube scenarios rather than tension. Furthermore, the authors conclude that the solution offered by Flexcom offers greater computational efficiency than the Abaqus simulation (i.e. shorter run-times).

RELEVANT KEYWORDS

- `*PIP SECTION` is used to define internal and external pipe sections when part of a pipe-in-pipe model is contained within another.
- `*PIP CONNECTION` is used to define pipe-in-pipe connections between nodes of the finite element model.
- `*NO PIP SLIDING` is used to disable the interchangeable nature of sliding pipe-in-pipe connections.
- `*PIP STIFFNESS` is used to define force-deflection curves for non-linear pipe-in-pipe connection stiffnesses.

- *LINES PIP is used to specify that two lines are connected in a pipe-in-pipe (or pipe-on-pipe) configuration.

If you would like to see an example of how these keywords are used in practice, refer to A03 - Pipe-in-Pipe Production Riser (Standard Connections) and H02 - J-Tube Pull-In (Sliding Connections).

Model Set-up Guidelines

Pipe-in-pipe models tend to be quite complex, so the following guidance is provided to assist new users with model creation and help to avoid common pitfalls. Some of the following information may be very familiar to existing Flexcom users, but a coherent summary nonetheless serves as a helpful reference point.

BASIC MODEL CONSTRUCTION

- Construct the basic model first before concerning yourself with any pipe-in-pipe interaction. Most or all of the model may be constructed using the *LINES keyword but occasionally you may need to define nodes and elements explicitly using the *NODE and *ELEMENT keywords.

- If you are using the *LINES keyword exclusively to construct your model, it should be possible to use the *LINES PIP keyword to define the pipe-in-pipe interactions. This creates all the relevant pipe-in-pipe section definitions, standard connections, and nodal equivalences (to represent bulkheads). This is a very convenient approach as it requires very little user input, but you may find it overly restrictive as you cannot perform any personalised adjustments to the model subsequently. Many users, particularly experienced ones, will prefer to disregard the *LINES PIP keyword and opt to define the various pipe-in-pipe inputs individually. Much of the remainder of this article assumes that you are using *LINES (and possibly *NODE and *ELEMENT also) rather than *LINES PIP.

- Pipe-in-pipe models should always be set up such that primary and secondary sections are initially concentric. It is important to note that while Flexcom monitors the relative displacement of the connected nodes in the lateral direction (as outlined in Contact Modelling), it does not take any directionality into account. For illustrative purposes, let's consider a dual-bore top tensioned riser. The sign of the displacement term is always positive when examined in plan view, regardless of the whether the inner section moves from right to left, left to right, upwards to downwards, or downwards to upwards, within the outer section. If you are modelling an eccentric pipe-in-pipe configuration, where the inner pipe is not concentric with the outer pipe, then you should set up the model such that both sections are concentric initially, and then apply a desired offset to the inner pipe subsequently.
    - A related point therefore is that Power Law connections are recommended, as all displacement and force terms are inherently positive by definition. If you are using force-deflection curves which are explicitly defined using Data Pairs, then you should ensure that all specified deflection values are zero or positive.

IDENTIFYING OUTER AND INNER SECTIONS

- Use the `*PIP SECTION` keyword to identify which pipe sections are contained within which. Pipe-in-pipe section data is used in the calculation of buoyancy and hydrodynamic forces. If you are defining a pipe-on-pipe configuration, then you will not require the `*PIP SECTION` keyword. If you are defining a pipe-in-pipe bundle, then some pipes may act as both inner and outer sections for other pipes.

IDENTIFYING NODES WHICH CAN COME INTO CONTACT

- Use the `*PIP CONNECTION` keyword to define pipe-in-pipe connections between nodes on the primary and secondary pipes. Connections govern the physical interaction between the pipes. Recommended practice is to designate the larger/stronger pipe as the primary pipe, and the smaller/dependent pipe as the secondary pipe. The primary elements are used to determine the orientation vectors of the stiffness connections between the connected pipes, and it is important that these vectors remain perpendicular to the overall pipe geometry in so far as possible. Designating the smaller/dependent pipe as the primary pipe is not recommended, as the consistency of the orientation vectors could be compromised due to its flexibility.

- If you are expecting little or no relative sliding between the pipes in the axial direction, standard connections are the recommended option. They are computationally efficient as each pair of connected primary and secondary nodes remain connected throughout the entire simulation and the model is effectively fixed from the outset.
- If the primary and secondary pipes share a common finite element mesh, then it is very straightforward to generate all the required standard connections using the `GEN=`*`Primary Nodes`*, `GEN=`*`Secondary Nodes`* input format under the `*PIP CONNECTION` keyword. In practice however, this is rarely the case so this approach is typically suitable for very simple models only.
- For models of any significant complexity, the mesh densities on the primary and secondary pipes will vary, both in terms of the overall number of nodes, and the distribution of these nodes along the pipe lengths. So identifying all the required one-to-one connections as suggested in the previous point may be time consuming and tedious, and simply unfeasible in some cases.
- So even if you are not expecting any relative sliding, it can be advantageous to create a series of sliding connections to help assemble the model connectivity initially, and to subsequently designate these connections as non-sliding once the overall model configuration has been established. Sliding connections may be created using the `GEN=`*`Primary Nodes`*, `SET=`*`Secondary Element Set`* input format under the `*PIP CONNECTION` keyword. This means that the program will automatically determine the optimum nodal connectivity. This considerably reduces effort on your part as the software determines which primary and secondary nodes are to be connected.
- Sliding connections can also be very useful in situations where the meshes are consistent, but where significant axial motion occurs during an initial alignment stage. For example, when setting up a landing string model, the initial set-down of the drill string within the marine riser can make it very difficult to manually identify the optimal set of pipe-in-pipe connections in advance of the initial static analysis. From a dynamic perspective, if relative sliding is not anticipated, the connections can be designated as non-sliding subsequently.

- Assuming that relative sliding is not anticipated from a dynamic perspective, you should ensure to designate the connections as non-sliding prior to any dynamic simulations. This is achieved via the inclusion of the `*NO PIP SLIDING` keyword. This means that the connections are effectively treated as standard rather than sliding from that point onwards. This eliminates any overhead associated with the monitoring of relative nodal locations over time, and can possibly provide enhanced numerical stability as the model connectivity remains consistent throughout the entire simulation. Note that you must explicitly specify the `*NO PIP SLIDING` keyword in every dynamic simulation. If you perform dynamic continuation stages, the keyword must be included in any restart keyword files also.

- Finally, and this may already be obvious, if you are genuinely expecting relative sliding between the primary and secondary pipes, you should define sliding connections as discussed above, and disregard any comments relating to the reassignment of the connections as non-sliding. Sliding connections are ideal for situations where continuous realignment of the pipe-in-pipe connections is required, for example in the case of a J-Tube Pull-In.

FINITE ELEMENT MESH DISCRETISATION

- Strictly speaking, it is not necessary to have finite element mesh alignment between the primary and secondary pipes. The contact stiffness at each pipe-in-pipe connection acts in a direction which is perpendicular to the Primary Element, so in theory the contact model should be independent of any axial separation which exists between connected nodes. However in practice, having a reasonably well aligned model is generally recommended, and is actually necessary for models which incorporate significant variations in local curvature.

- Line mesh generation is performed individually for each line, independently of all other lines in the model. So if you are using the `*LINES` keyword to construct your model, there is no automatic way to enforce mesh consistency between the primary and secondary pipes. If you feel this is essential to the success of your model, you could include additional sub-sections on one or both pipes, even if the structural properties of these sub-sections are identical to adjacent sub-sections, in order to replicate the same finite element mesh across both pipes. Although possible, this may be quite tedious to do in practice, and it is likely to add considerable effort to your model construction phase. Furthermore, it would not necessarily ensure mesh alignment in models where relative axial motion occurs during an initial alignment stage (e.g. initial set-down of a drill string within a marine riser). So it is mentioned here for completeness only.

- By default, drag forces and hydrodynamic inertia on inner pipe-in-pipe elements are modelled as mass and damping terms on the left hand side of the equations of motion, capturing the required coupling between the outer node’s velocity/acceleration and the inner node loading. This necessitates the existence of suitable pipe-in-pipe connections in the global connectivity matrix. In situations where there is no user-defined connection guaranteed to exist between an inner node and the outer pipe, the software automatically inserts a token connection of zero stiffness where required to ensure that hydrodynamic loading on the inner node can be captured. This may lead to the creation of a large number of additional connections, which can have a negative impact on run-time performance. This approach may be viewed as overly conservative by some software users, who may instead opt to suppress the creation of the additional connections via the `*PIP SECTION` keyword (`AUTO_CREATE` option). Before you choose this option, you are advised to carefully read the article on Additional Connections to Support Inner Pipe Hydrodynamics and become fully aware of its potential implications.

MODELLING CENTRALISERS, BULKHEADS AND WALL-TO-WALL CONTACT

- Where you are using linear connections (e.g. to model centralisers), the linear contact stiffness is defined under the `*PIP CONNECTION` keyword so no further user input is required.

- In regions where the connected pipes are free to move radially relative to one another, a non-linear contact relationship should be defined under the `*PIP STIFFNESS` keyword. Recommended procedure is to use the Power-Law approach which ensures a gradual transition between regions of low (i.e. free movement zone) and high stiffness (i.e. wall-to-wall contact) and tends to aid solution robustness. Using this approach, the instantaneous contact stiffness is derived from a power-law equation which is governed by the relative spacing between the pipes in the lateral direction, and some user-defined coefficients such as a maximum contact force and a power exponent. As this is a crucial aspect of pipe-in-pipe modelling, you are advised to read the guidance regarding contact force and exponent terms.
    - For pipe-in-pipe models, you should use the `CONFIGURATION=PIP` option under the `*PIP STIFFNESS` keyword.
    - For pipe-on-pipe models, you should use the `CONFIGURATION=POP` option under the `*PIP STIFFNESS` keyword.
- In situations where the pipes are rigidly connected to each other, via a bulkhead for example, you will need to create a rigid constraint.
    - For pipe-in-pipe models, you should use the `*EQUIVALENT` keyword to specify that the inner and outer nodes are modelled a single equivalent node.
    - For pipe-on-pipe models, you should use the `*ELEMENT` keyword to insert a rigid massless element between the primary and secondary nodes.

DOUBLE-CHECKING YOUR MODEL

- Given the complexity associated with pipe-in-pipe modelling - as reflected by the amount of user documentation on this topic - you are encouraged to inspect your model in some detail to ensure that it is operating as expected and meets your requirements. You can insert the `*PRINT` keyword (sub-entry `OUTPUT=PIP CONNECTIONS`) at any simulation stage to request additional information to be printed to the main output file. This facilitates a detailed inspection of the pipe-in-pipe configuration at any point during the simulation, and provides greater transparency regarding the internal workings of the software. Some sample information is shown below.

```
                       *** ADDITIONAL PRINTED OUTPUT ***
                       ---------------------------------
  Pipe-in-Pipe Connections:
```

```
  Number     Type    Primary Node   Secondary Node    Status    Primary El
 ---------------------------------------------------------------------------
     1      Sliding       1033           1002         Active          10
     2      Sliding       1034           1004         Active          10
     3      Sliding       1035           1005         Active          10
     4      Sliding       1036           1006         Active          10
     5      Sliding       1037           1008         Active          10
```

FURTHER INFORMATION

To gain further insight and a complete understanding of the internal operation of the pipe-in-pipe modelling algorithms in Flexcom, you are encouraged to read these related articles also...

- Pipe-in-Pipe Sections describes how one pipe is contained within another, and explains the implications of this for buoyancy and hydrodynamic forces.
- Standard Connections outlines the operation of the standard pipe-in-pipe contact model at a high level.
- Sliding Connections explains the additional functionality offered by sliding connections, and suggests scenarios for which it is appropriate.
- Contact Modelling discusses the operation of the pipe-in-pipe contact modelling algorithm in detail.
- Hydrodynamic Forces outlines the main fluid forces modelled by Flexcom (including buoyancy, hydrodynamic and internal fluid) and how these are affected by the presence of pipe-in-pipe configurations.

FOOTNOTE

- **Cable Bundles**. If you are defining pipe-in-pipe connections in a model which includes connections along a curved section, you need to ensure that both pipes adopt a consistent profile initially. Consider the case of a steel catenary riser which has an inner tubing enclosed within it. If you define the geometry of the inner and outer pipes separately via the `*LINES` or `*CABLE` keywords, then the solutions from the Cable Pre-Static Step for each pipe will be independent. As the properties of the riser and tubing will be very different, the cable solutions will naturally be quite different also. Note that the cable pre-static solution is based solely on apparent weight and axial stiffness, and does not take account of any pipe-in-pipe connections. The cable pre-static solution is used as a first approximation to the full finite element solution, and the pipe-in-pipe connections are initialised on that basis also. This means that any initial deviations of the inner pipe away from the centreline of the outer pipe will be maintained throughout the entire duration of the analysis, which is probably an undesirable effect. To cater for such circumstances, you can use the `*CABLE BUNDLE` keyword to specify that two or more cables are to be grouped together within a single bundle for the purposes of the cable pre-static step only. This means that internally, all cables within a bundle share common properties for the purposes of the cable computations, ensuring that each cable adopts a similar profile before the full finite element solution initiates. Note that if you are using the `*LINES PIP` keyword to set up your model, then you do not need to concern yourself with the issue of cable bundles, as it is handled automatically by the software.

Pipe-in-Pipe Sections

THEORY

You define pipe-in-pipe section data very simply by defining a pair or pairs of outer and inner element sets, the composition of the sets being defined in the normal fashion. Pipe-in-pipe section data is used in the calculation of buoyancy and hydrodynamic forces on the relevant sets.

With regard to buoyancy forces, the internal fluid of the outer set of a pair becomes the external fluid for the inner set, and pressure forces are calculated on that basis. In terms of hydrodynamic loading, the annular fluid acts as the external fluid for the inner pipe. This means that the inner pipe is not subjected to any hydrodynamic loading from the external ambient environment. Specifically, wave and current loads are experienced by the outer pipe only, and do not (directly at least) apply any hydrodynamic loading to the inner pipe.

Note also that the annular fluid is assumed to move rigidly with the outer pipe as it is entrained within it. So the velocity and acceleration terms for the annular fluid are assumed equal to those of the outer pipe. Therefore, the drag and hydrodynamic inertia forces are themselves dependent on the unknown velocity and acceleration solution variables. For this reason, Flexcom automatically creates a direct finite element coupling between the inner and outer nodes (using token pipe-in-pipe connections of zero stiffness) so that the relevant drag and hydrodynamic inertia forces may be computed for any inner nodes which are not already connected to outer nodes via Standard or Sliding Connections. In earlier versions, it was the responsibility of the user to create these additional connections, but this is no longer necessary. Hydrodynamic forces on the inner pipe are then computed in the normal fashion via Morison's Equation. Recommended practice is to specify the hydrodynamic properties of both pipes as normal, and allow Flexcom to handle the pipe-in-pipe modelling aspects of the configuration. Refer to Hydrodynamic Forces for more detailed information on this topic.

If you are defining a pipe-on-pipe (or piggy-backed) system, rather than a genuine pipe-in-pipe one where one set is contained within the other, then the specification of pipe-in-pipe sections is unnecessary. Buoyancy and hydrodynamic forces on all elements will then be calculated in the normal way.

A particular outer element set can contain more than one inner element set, but naturally a particular inner element set can have only one associated outer element set. Also, the inner element set of one pair can also be the outer element set of another pair. An example might be a multi-tube production riser, consisting perhaps of outer and inner risers and production tubing. The inner riser is contained within the outer riser, but also contains the tubing, so it would be both an inner element set and an outer element set. In the likely event of the inner riser sliding axially out of the outer riser while the tubing remains in place, then the fluid contents of the outer riser would define the hydrodynamic loading on the tubing.

RELEVANT KEYWORDS

- `*PIP SECTION` is used to define internal and external pipe sections when part of a pipe-in-pipe model is contained within another.

If you would like to see an example of how this keyword is used in practice, refer to A03 - Pipe-in-Pipe Production Riser.

Standard Connections

INTRODUCTION

You define standard pipe-in-pipe connections between a pair of nodes when setting up your model, and those two nodes remain connected throughout all subsequent analysis stages. You can define standard connections in two ways: you can identify the pair of nodes directly, or you can connect pairs of nodes from two groups of elements using a generate option. It is important to be clear that even when you use the latter method, a pair of nodes connected initially remain connected thereafter.

OPERATION

Pipe-in-pipe connections can be considered to operate like Spring Elements, whether linear or non-linear. However the springs in this context do not directly connect the two nodes (in many cases the nodes are coincident, at least initially, and so cannot be connected by a spring). What happens is that a spring stiffness is added directly into the stiffness matrix at locations corresponding to the appropriate motions of the two nodes. The direction of the spring is computed as being normal to the primary node.

Pipe-in-pipe connections are primarily designed to resist relative lateral motion of the connected nodes. However, it is also possible to simulate some resistance to motion in the axial/longitudinal direction, which may be useful in certain circumstances.

Note also that the stiffness contribution is always computed and added, no matter how far apart the two nodes move axially. If the degree of relative motion between the inner and outer pipes in the axial direction is significant, then you should define Sliding Connections instead.

PRIMARY AND SECONDARY PIPES

Strictly speaking, there are no mandatory requirements regarding which pipe should be designated as primary and which as secondary. Recommended practice is to designate the larger/stronger pipe as the primary pipe, and the smaller/dependent pipe as the secondary pipe. The following examples provide some practical illustration.

- For pipe-in-pipe configurations, the outer pipe is normally designated as the primary pipe, and the inner pipe as the secondary pipe. The outer pipe is naturally larger in diameter and usually has a higher bending stiffness. Top tensioned risers for example, typically have a steel outer pipe with a flexible inner tubing. In this case, the steel outer pipe should be designated as primary, with the inner tubing denoted secondary.

- For pipe-on-pipe configurations, the main/carrier pipe is normally designated as the primary pipe, with the attached pipe as the secondary pipe. Again the carrier pipe is typically larger and stronger than any attached lines. Hybrid riser towers for example, typically have a central steel tendon with a number of attached lines to support production, water injection etc. In this case, the steel tendon should be designated as primary, with any attached lines denoted secondary.

The primary elements are used to determine the orientation vectors of the stiffness connections between the connected pipes, and it is important that these vectors remain perpendicular to the overall pipe geometry in so far as possible. Designating the smaller/dependent pipe as the primary pipe is not recommended, as the consistency of the orientation vectors could be compromised due to the flexibility of the inner pipe. Considering the top tensioned riser for example, if the flexible inner tubing is allowed to set down under self weight inside the steel pipe, the local orientation vectors could be non-horizontal and also vary considerably at different elevations.

RELATIONSHIP BETWEEN CONNECTIONS AND SECTIONS

Pipe-in-pipe drag and hydrodynamic inertia forces due to the presence annular fluid are dependent on velocity and acceleration solution variables (the fluid is assumed to move rigidly with the outer pipe nodes). For this reason, a direct finite element coupling between the inner and outer pipe-in-pipe nodes must be explicitly specified using pipe-in-pipe connections between the inner and outer nodes for the drag and hydrodynamic inertia forces to be computed. So Flexcom automatically inserts token connections of zero stiffness where required to ensure that hydrodynamic loading on all inner nodes is modelled, even if there is no user-defined connection between all outer and inner nodes. Refer to Hydrodynamic Forces for further details.

RELEVANT KEYWORDS

- `*PIP CONNECTION` is used to define pipe-in-pipe connections between nodes of the finite element model.

- `*PIP STIFFNESS` is used to define force-deflection curves for non-linear pipe-in-pipe connection stiffnesses.

If you would like to see an example of how these keywords are used in practice, refer to A03 - Pipe-in-Pipe Production Riser.

Sliding Connections

OVERVIEW

Sliding connections are similar to standard ones, with the added advantage that the connections are interchangeable. This is appropriate for modelling scenarios where there is significant relative axial motion between primary and secondary structures.

When setting up sliding connections, you associate a node on the primary structure (or a group of nodes if you avail of the generate option) with a set of nodes on the secondary structure. You may have noticed that there is currently no facility for defining node sets in Flexcom, although the use of element sets is widespread throughout the program. So you actually define a relevant element set instead, and an appropriate node set is created internally for you. Based on the proximity of the various nodes at the beginning of the analysis, the program creates an initial set of (effectively standard) connections between the primary and secondary structures. The modelling procedure is then very similar to that of the standard connections – resistance to relative motion of the pipes in the lateral plane is achieved via an appropriate augmentation of the global stiffness matrix. The main difference lies in the fact that the connections are interchangeable. The program continually monitors the relative axial locations of the connected nodes over the course of the analysis, and the set of connections is updated as and when required.

Another advantage of using sliding connections is that it is also possible to model a degree of resistance to relative motion of the pipes in the axial direction. This may be may characterised by a linear axial stiffness value or a non-linear force-deflection relationship.

OPERATION

Connections are constructed based on the instantaneous locations of the primary and secondary nodes. The distance between a primary and a secondary node is computed in a local axis system which is formed based on the primary element. The modelling procedure is similar to that discussed in Contact Modelling, and interested readers are referred to that section for further details as required.

Once the initial set of connections has been established the analysis proceeds as normal. After every solution time step, the set of connections is revised and updated to reflect the latest nodal positions. A margin of 5% is used to avoid connections alternating back and forth at the changeover point, when there are two secondary nodes which are almost equidistant axially from the primary node. So once a connection has been established between a primary node and a secondary node, a competing secondary node must be at least 5% closer to the primary node before the interchange happens.

By definition, sliding connections are based on primary nodes. It is also important to note that each primary node can only be connected to one secondary node at any given time. So for example, if the mesh density on the secondary section is more refined than the primary section, the number of active connections will be equal to the number of primary nodes. If the primary section has a higher mesh density, then it is possible that some secondary nodes can be connected to more than one primary node. Advanced users who wish to examine the active connections at any point during the simulation may use the `*PRINT` keyword.

INACTIVE CONNECTIONS

A situation can sometimes occur where a secondary section is being inserted into a primary section (such as a J-Tube Pull-In), or being removed from it. In such circumstances, some or all of the secondary section is completely detached from primary section, and the connections are deemed to be inactive. For each primary node, Flexcom computes the axial distance between it and the nearest secondary node. If this distance is greater than a threshold value, then the connection is deemed to be inactive. The threshold value is equal to the length of the longest element in the secondary element set. Assuming for simplicity that all elements in the primary and secondary sections have equal length, connections start to become inactive when the nearest secondary node is more than one element length away from the primary node.

While a connection is inactive, it makes no contribution to the global stiffness matrix. Refer to Contact Modelling for further details. Note also that even if a connection becomes inactive at a particular time, it may become active again subsequently.

SAMPLE APPLICATIONS

Sliding connections are appropriate for modelling scenarios where there is significant relative axial motion between the primary and secondary structures. For example, J-Tube Pull-In scenarios are well suited to sliding connections.

In certain circumstances, it may be desirable to allow the software to initially determine appropriate connections between a primary and secondary section, and to subsequently treat these connections as standard connections. For example, when setting up a landing string model, the initial set-down of the drill string within the marine riser can make it very difficult to manually identify the optimal set of pipe-in-pipe connections in advance of the initial static analysis. Designating the connections as sliding allows the program to automatically determine the optimal connections, minimising effort on the part of the user. While there may be significant relative axial motion between the initial model definition and the converged static solution, there is comparatively little axial motion during the actual simulation itself (e.g. when the model is subjected to wave loading). Invoking the `*NO PIP SLIDING` keyword in subsequent restart analyses thereby ensures computational efficiency (any overhead associated with the monitoring of nodal locations is eliminated), and can also provide enhanced numerical stability (connectivity of the finite element model remains consistent throughout the simulation). For example, Completion/Workover Risers would be suitable for modelling using Sliding Connections, in conjunction with the `*NO PIP SLIDING` keyword.

Regarding sliding connections, there may be a slight computational overhead associated with the continual monitoring of relative axial locations over the course of the analysis. For this reason, it is recommended that you utilise standard connections in situations where significant relative axial motion is unlikely to occur – for example, if there are bulkheads present. For example, a Top-Tensioned Production Riser would be suitable for modelling using Standard Connections.

BANDWIDTH OPTIMISATION & PIP SLIDING

Flexcom, like most comparable finite element codes, has a bandwidth optimisation facility that internally renumbers the node and elements of the finite element mesh to minimise the required array storage. By default this minimisation occurs once only (in the initial static analysis) because mesh connectivity does not change thereafter in the overwhelming majority of cases. One exception is when sliding pipe-in-pipe connections change nodes, as discussed above. Again by default, Flexcom does not repeat bandwidth optimisation every time this happens, because the potential saving in array storage is likely to be small, and the computing overhead associated with the process could significantly affect analysis run-time. But you can override this default and instruct Flexcom to repeatedly perform bandwidth optimisation, via the `BANDWIDTH=UPDATED` option under the `*PIP CONNECTION` keyword. Doing this only make senses if (i) sliding pipe-in-pipe connections change between nodes only intermittently, but (ii) the magnitude of the change when it occurs is significant enough to make the optimisation worth the computing effort. As these conditions will be very rarely met, the option will be very rarely invoked.

RELEVANT KEYWORDS

- `*PIP CONNECTION` is used to define pipe-in-pipe connections between nodes of the finite element model.
- `*PIP STIFFNESS` is used to define force-deflection curves for non-linear pipe-in-pipe connection stiffnesses.
- `*NO PIP SLIDING` is used to disable the interchangeable nature of sliding pipe-in-pipe connections.
- `*PRINT` is used to request additional printed output to the main output file. Specifically, the `OUTPUT=PIP CONNECTIONS` option facilitates a detailed inspection of the connected nodes at any point during the simulation, and provides greater transparency regarding the internal workings of the software.

If you would like to see an example of how these keywords are used in practice, refer to H02 - J-Tube Pull-In.

Contact Modelling

INTRODUCTION

The pipe-in-pipe contact modelling algorithm performs a number of sequential stages. These same stages are applied to every pipe-in-pipe connection, at every solution iteration. The only exception being Inactive Connections - these are sliding connections which are deemed to have become inactive following excessive relative axial motion between the connected pipes.

PRIMARY AND SECONDARY NODES

Firstly, the relevant finite element nodes in the pipe-in-pipe connection are identified. For standard connections, the primary and secondary nodes are defined during model set-up, and remain the same throughout the analysis. For sliding connections, each connection is defined by a primary node which remains consistent, and a secondary node whose identity may vary from one solution time step to the next.

PRIMARY ELEMENT

A primary element is also identified for each pipe-in-pipe connection. Identification of the primary elements is important, as the local contact stiffness terms are applied in a direction which is perpendicular to each primary element. The primary element must contain the primary node as one of its end nodes. Where a contact node is shared between two or more elements, the first (lowest numbered) element is selected.

For top tensioned risers which are numbered from bottom to top, the primary elements typically lie immediately underneath the primary nodes. In such circumstances, structural curvatures generally vary little from element to element, so the distinction between a primary element which is just above or just below the connected nodes is immaterial. However, you are naturally free to alter the model set-up (which is very easy if the model is built using Lines) should you wish to influence the selection of the primary element.

TRANSFORMATION MATRIX

Next a 3x3 transformation matrix is assembled for the pipe-in-pipe connection. Its purpose is to facilitate transformation of information between the local axis system associated with the pipe-in-pipe connection, and the global axis system in which Flexcom computes its solution. Each row of the matrix corresponds to a local vector, local-x, local-y and local-z.

The local-x vector corresponds to the instantaneous (convected) local-x axis of the Primary Element. Specifically the local-x vector points directly between the start node and the end node of the primary element. If the element length is zero for some reason (this would be expected for Hinge Elements, and theoretically possible for beam or spring elements experiencing severe compressive strain), then the local-x vector is assumed to be aligned with the global-X vector. Finally the local-x vector is normalised.

The local-y vector is defined in a plane which is perpendicular to the local-x vector. A simple procedure is used as follows. If the first component of the local-x vector (x_1) is non-zero, then the local-y vector is defined as $\{-x_2/x_1,1,0\}$. Otherwise, if the second component of the local-x vector (x_2) is non-zero, then the local-y vector is defined as $\{0,-x_3/x_2,1\}$. Otherwise, the local-y vector is defined as $\{1,0,-x_1/x_3\}$. Finally the local-y vector is normalised. Although the procedure may seem somewhat arbitrary, the only important aspect at this stage is the orthogonality of the local-x and local-y vectors.

The local-z vector is formed from the cross product of the local-x and local-y vectors. The 3 local vectors are then inserted into the 3 rows of the (global to local) transformation matrix. An inverse matrix is also created to facilitate reverse operations, from local to global.

CONNECTION VECTOR

A new vector is now assembled which points directly between the instantaneous positions of the primary and secondary nodes of the pipe-in-pipe connection. This vector is then transformed from the global to local axis system using the assembled transformation matrix. The axial component of this local vector is then suppressed by setting the local-x term to zero, which equates to a projection of the vector onto the local y-z plane. This vector is then transformed back into the global axis system, where it effectively represents a vector which is (i) perpendicular to the primary element and (ii) lies in a plane which contains both the primary and secondary nodes of the pipe-in-pipe connection.

By default, the connection vector is calculated on the basis of the instantaneous structure configuration. It is possible to instruct Flexcom to base the connection vector on the initial structure configuration (via the `ORIENTATION=INITIAL` option under the `*PIP CONNECTION` keyword), and this may help solution stability in exceptional circumstances. For example, if the lateral movement of the inner pipe with respect to the outer pipe is small in magnitude but variable in direction, this can cause the instantaneous orientation of the connection vector to continually change between successive solution times. However this is very much the exception, and the default operation is the preferred and recommended method for the majority of models.

LOCAL AXIS SYSTEM

Now a newer, more refined transformation matrix is assembled for the pipe-in-pipe connection. This matrix is central to the integration of the pipe-in-pipe module into the global solution. It has 3 components, each of which corresponds directly to one of the local degrees of freedom of the pipe-in-pipe connection. Hence the terms *longitudinal*, *normal* and *transverse* are used from this point onwards, rather than generic terms such as *local-x*, *local-y* and *local-z*.

- The longitudinal axis of the local axis system corresponds to the instantaneous (convected) local-x axis of the primary element, as discussed in the Transformation Matrix section.
- The normal axis of the local axis system is perpendicular to the local-x vector, and lies in a plane which contains both the primary and secondary nodes, as discussed in the Connection Vector section.
- The transverse axis of the local axis system completes the right handed system.

$$\begin{bmatrix} L_x, L_y, L_z \\ N_x, N_y, N_z \\ T_x, T_y, T_z \end{bmatrix}$$

Transformation Matrix

An inverse transformation matrix is also created to facilitate reverse operations.

CONTACT STIFFNESS

For pipe-in-pipe connections which have been assigned a linear connection stiffness, the relevant stiffness term is quickly identified from the relevant keyword entry (i.e. the numerical value specified following the `STIFFNESS=` option under the `*PIP CONNECTION` keyword).

For pipe-in-pipe connections which have been assigned a non-linear connection stiffness (using the `CURVE=` option under the `*PIP CONNECTION` keyword, and the `*PIP STIFFNESS` keyword), the procedure is slightly more involved. Firstly, the relative displacement of the primary and secondary nodes is computed based on the instantaneous positions of these nodes with respect to their initial positions. Displacements are solved in the global axis system, so the relative displacement vector is computed in a global context, and then transformed into the local axis system using the pre-defined Transformation Matrix. The normal component of this vector, representing the relative displacement of the connected nodes in the lateral direction, is then used in conjunction with the relevant non-linear force-deflection curve to determine the instantaneous contact stiffness.

If the non-linear force-deflection curve has been defined using explicit Data Pairs, Flexcom loops through the deflection values searching for two consecutive values which lie either side of the lateral displacement term. Linear interpolation is used to determine the relevant stiffness of the connection. If the lateral displacement term lies outside the specified range of the force-deflection values, extrapolation is used, based on the force-deflection terms at the lower or upper extremity as appropriate.

If the non-linear force-deflection curve has been defined using the Power-Law approach, the instantaneous stiffness is readily derived from the power-law equation.

- For pipe-in-pipe models (i.e. `*PIP STIFFNESS` -> `TYPE=POWER LAW` -> `CONFIGURATION=PIP`), the maximum lateral displacement is defined as...

$$d_{\max} = \left(\frac{Dio - Doi}{2} \right)$$

where Dio is the internal diameter of the outer pipe, and Doi is the external (contact) diameter of the inner pipe.

- The lateral displacement term is expressed as a percentage of maximum displacement. Where relative penetration has occurred (percentages greater than 100%), the percentage value is reset to the maximum value of 100%. This prevents the numerical model from creating contact forces which are greater than the user specified Maximum Contact Force. A pair of force-deflection values is computed in the region of the percentage value, corresponding to the nearest integer percentage values. For example, if the instantaneous lateral displacement is 82.4% of the maximum displacement value, then actual deflection values are defined at 82% and 83% of maximum displacement. Corresponding force values are then computed using the Pipe-in-Pipe Power-Law equation. Linear interpolation is used to determine the instantaneous stiffness of the connection. Note that pipe-in-pipe contact is based on the Tangent Stiffness method.

- For pipe-on-pipe models (i.e. `*PIP STIFFNESS` -> `TYPE=POWER LAW` -> `CONFIGURATION=POP`), the maximum lateral displacement is defined as the initial separation between the connected nodes in the lateral direction, taking external (contact) diameters into account. Again the lateral displacement term is expressed as a percentage of maximum displacement. Thereafter the procedure is very similar to that just described for the pipe-in-pipe case, except that the Pipe-on-Pipe Power-Law equation is used instead.

Consistent with the Local Axis System discussion, the contact stiffness as just described is denoted the normal stiffness, K_n.

AXIAL STIFFNESS

It is not currently possible to model the effects of axial friction between sliding pipes. However some degree of resistance to relative axial motion may be simulated using the axial stiffness inputs. Linear axial stiffness values may be specified via `*PIP CONNECTION` -> `AXIAL_STIFF`, while non-linear axial force-deflection relationships may be specified via `*PIP CONNECTION` -> `AXIAL_NONLINEAR`.

For pipe-in-pipe connections which have been assigned a linear axial stiffness, the relevant stiffness term is quickly identified from the relevant keyword entry.

For pipe-in-pipe connections which have been assigned a non-linear axial stiffness, the procedure is slightly more involved, but is conceptually similar to that discussed in the Contact Stiffness section. Firstly, the relative displacement of the primary and secondary nodes is computed based on the instantaneous positions of these nodes with respect to their initial positions. Displacements are solved in the global axis system, so the relative displacement vector is computed in a global context, and then transformed into the local axis system using the pre-defined Transformation Matrix. The axial component of this vector, representing the relative displacement of the connected nodes in the axial direction, is then used in conjunction with the relevant non-linear force-deflection curve to determine the instantaneous axial stiffness. The linear interpolation technique used is identical to that discussed previously.

Consistent with the Local Axis System discussion, the axial stiffness is denoted the longitudinal stiffness, K_l.

TRANSVERSE STIFFNESS

The governing entries are naturally the lateral and axial stiffness discussed in the previous sections. There is one further term, called the transverse stiffness, K_t. For reasons of numerical stability, and to a lesser extent to represent a physical resistance to transverse motion when two pipes have come into contact in a normal direction, a transverse stiffness term is computed and inserted at 90° to the normal stiffness term.

For pipe-in-pipe connections which have been assigned a linear connection stiffness, the same stiffness term is used in both lateral degrees of freedom. Specifically, $K_t = K_n$.

For pipe-in-pipe connections which have been assigned a non-linear connection stiffness, the transverse stiffness depends on the instantaneous configuration. If the connected nodes experience some degree of physical separation in the lateral plane, the Local Axis System will have already been determined with absolute certainty. In this case, the same stiffness term is used in both lateral degrees of freedom. Specifically, $K_t = K_n$. If the connected nodes are perfectly coincident, which might be the case for concentric pipe sections prior to the application of any external loading, the local axis system may be somewhat arbitrary. In this case, the transverse and normal directions are treated independently, and the transverse stiffness is computed using the procedure already documented in the Contact Stiffness section.

ASSEMBLY OF GLOBAL STIFFNESS MATRIX

A local 12x12 stiffness matrix is assembled for the pipe-in-pipe connection. Flexcom actually uses a fourteen degree of freedom Hybrid Beam Element, where the axial force and torque are added to the usual form of a three-dimensional beam element. However these additional degrees of freedom are not relevant to pipe-in-pipe connections.

The local stiffness matrix is populated as follows.

$$\begin{bmatrix} K_l, 0, 0, 0, 0, 0, -K_l, 0, 0, 0, 0, 0 \\ 0, K_n, 0, 0, 0, 0, 0, -K_n, 0, 0, 0, 0 \\ 0, 0, 0, K_t, 0, 0, 0, 0, -K_t, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ -K_l, 0, 0, 0, 0, 0, K_l, 0, 0, 0, 0, 0 \\ 0, -K_n, 0, 0, 0, 0, 0, K_n, 0, 0, 0, 0 \\ 0, 0, 0, -K_t, 0, 0, 0, 0, K_t, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \\ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 \end{bmatrix}$$

Local Stiffness Matrix

The local stiffness matrix is then transformed into the global axis system, using a 12x12 version of the (inverse) Transformation Matrix defined earlier. This is then used to augment the global finite element stiffness matrix at the appropriate locations.

GLOBAL FORCE VECTOR

All of the terms discussed up to this point relate to the stiffness matrix which appears on the left hand side of the Finite Element Equations of Motion. Some additional terms also need to be assembled and inserted into the global force vector on the right hand side.

There are two separate items which fall into this category, stemming from rigid body motions, and non-linear pipe-in-pipe connections.

- Because Flexcom solves for position rather than displacement, forces corresponding to rigid body motions must be included in the global force vector. This approach applies to all elements in the Flexcom solution, but the steps relevant to pipe-in-pipe connections are outlined here. The initial separation between the connected nodes is assembled into a local vector, with longitudinal, normal and transverse (zero by definition) terms, consistent with the definition of the Local Axis System outlined earlier. This initial separation vector is then transformed into the global axis system using the inverse Transformation Matrix, where it is then added into a 12x1 column vector at appropriate locations. This displacement vector is then multiplied by the global stiffness matrix for the pipe-in-pipe connection to produce the relevant right hand side force vector. This is then used to augment the global force vector at the appropriate locations.

- Although not explicitly mentioned in the preceding sections on Contact Stiffness and Axial Stiffness, a force term is also computed and stored during the same operation which returns the instantaneous stiffness of non-linear connections. As noted earlier, the stiffness term is equal to the tangent slope of the force-deflection relationship at the relevant displacement. The force term is the point on the force axis where the tangent slope intersects it. This is conceptually similar to the Non-linear Material Force Term associated with non-linear materials. The force terms are used to populate a 12x1 column vector at appropriate locations. Assuming the normal and transverse force terms are identical, the local force vector would be assembled as follows.

$$\begin{bmatrix} F_l \\ F_n \\ F_t \\ 0 \\ 0 \\ 0 \\ -F_l \\ -F_n \\ -F_t \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Local Force Vector

The local force vector is then transformed into the global axis system, using a 12x12 version of the (inverse) Transformation Matrix defined earlier. This is then used to augment the global force vector at the appropriate locations.

Non-Linear Power-Law Connections

OVERVIEW

Non-linear connections, whether standard or sliding, can now be defined using a power-law approach, defined in terms of a maximum contact force and an exponential coefficient (the latter parameter is optional and defaults to a value of 10 if omitted). Earlier versions of Flexcom required that you explicitly defined non-linear connections as force-deflection pairs, and while this approach is still retained for complete generality, the power-law approach is more convenient, requiring far less input on the part of the user.

- The user does not have to manually define a force-deflection relationship in terms of explicit data points
- It ensures a gradual transition between regions of low and high stiffness, which aids solution robustness
- It ensures that the slope of the force-deflection relationship is monotonically increasing with increasing displacement, which again aids solution robustness, particularly given that the pipe-in-pipe contact is based on the Tangent Stiffness method.
- For cases of Sliding Connections, for example a J-Tube Pull-In, the force-deflection relationship is automatically updated during a simulation to reflect contact diameters of connected elements at different points in time

OPERATION

Pipe-in-Pipe

For pipe-in-pipe type configurations, Flexcom automatically generates a non-linear relationship of the form:

$$F = F_{\max}\left(\frac{d}{d_{\max}}\right)^{Exp}$$

where:

- F is the instantaneous resistive force in the lateral direction provided by the connection
- $F_{\max}$ is the user-specified maximum contact force
- d is the relative displacement of the connected nodes in the lateral direction
- $d_{\max}$ is the maximum displacement of the connected nodes in the lateral direction before the inner and outer pipes come into contact. Specifically, $d_{\max} = \left(\frac{Dio - Doi}{2}\right)$, where Dio is the internal diameter of the outer pipe, and Doi is the external (contact) diameter of the inner pipe.
- The $d_{\max}$ parameter is automatically computed by Flexcom based on the instantaneous diameters at any point in time. This means that the non-linear relationship remains accurate at all times, even for sliding connections where the relevant diameters may vary as the pipes move in the axial direction.
- Exp is the user-specified exponent

Pipe-on-Pipe

For pipe-on-pipe type configurations, Flexcom automatically generates a non-linear relationship of the form:

$$F = F_{\max}\left(\frac{d_{init} - d_{inst}}{d_{init}}\right)^{Exp}$$

where:

- F is the instantaneous resistive force in the lateral direction provided by the connection

- $F_{\max}$ is the user-specified maximum contact force

- d_{init} is the initial separation between the connected nodes in the lateral direction, taking external (contact) diameters into account

- d_{inst} is the instantaneous separation between the connected nodes in the lateral direction, taking external (contact) diameters into account

- Exp is the user-specified exponent

Note that, given the above definitions, non-linear connections defined using a power-law approach are only meaningful for modelling resistance for pipe-in/on-pipe connections in the lateral direction. The option should not be used for modelling axial resistance.

GUIDANCE REGARDING CONTACT FORCE AND EXPONENT TERMS

Flexcom users regularly seek guidance regarding suitable values of both the maximum contact force and exponent entries. It is difficult to provide a definitive answer to this question as the optimal parameters may vary between different models. However, the following guidelines should be helpful.

- Using a high exponent value will align the analytical and theoretical solutions. However, the rationale behind the exponent input is that it allows the user to control the application of the contact stiffness term. Very high exponents can lead to solution convergence issues, as the contact stiffness can vary significantly over extremely small ranges of displacement. It could also cause an increase in run-time, as more solution iterations may be required to achieve convergence at every solution time step. Using an exponent term affords a reasonable compromise between theoretical accuracy and solution efficiency/robustness.

- In order to facilitate inspection of the automatically generated power-law relationship, the Flexcom output file presents a table illustrating the variation in contact force with lateral displacement, in both percentage and absolute terms. If you would like to visually check the force-deflection relationship for different values of exponent, this table may be easily imported into a spreadsheet and plotted graphically.

- The maximum contact force needs to be sufficiently high to prevent any significant penetration between the pipes (which would represent a physically unreasonable configuration). What constitutes an acceptable level of penetration is very much a subjective decision on the part of the user. Remember that the exponent value tends to compensate against any penetration also, so both inputs are interdependent to some degree. Rigid pipes may require a higher contact force than more flexible ones.

- From an engineering perspective, any possible misinterpretation of results would represent a more important concern than a slight compromise of modelling accuracy (e.g. in an effort to aid solution robustness). This point merits further elaboration.

    - Minor adjustments to the maximum force and exponent inputs may not make any significant difference in the context of a global analysis. For example, the overall model configuration should look largely similar, regardless of these specific inputs. Assuming the overall displacements are consistent with expectations, then parameters like bending moment and curvature should be reasonably accurate also. In summary, using moderate, rather than extreme, values of maximum force and exponent should not lead to any appreciable loss in precision.

    - If the solution exhibits high frequency noise, this is most likely a numerical issue which does not represent a physical phenomenon. Unexpected peaks in bending moment for example, may be induced by using excessively large maximum force and/or exponent terms. Many users employ Summary Postprocessing and Collation, and some users may rarely examine individual time history plots. Spurious peaks, going unnoticed, could conceivably distort the overall structural design. In summary, it is preferable to accept a slight modelling compromise, rather than striving for theoretical accuracy, which may come at the expense of introducing uncertainties.

- If you are concerned about optimal values of maximum force and/or exponent, performing some sensitivity studies may provide greater insight. For example, you could examine the effect of varying these parameters on bending moment in the connected pipes. Considering just one dynamic load case should serve as a useful starting point. This approach could also provide a validation of the advice presented in the previous point.

Based on the experience of the Flexcom technical support team, the following rules of thumb may be useful.

- A maximum contact force of 1.0E+06 N is recommended. Using higher values have been shown to contribute to numerical noise.
- An exponent value of between 10 (lower bound) and 50 (upper bound) is recommended.

IDENTIFYING CONTACT NODES

Switching on node numbers in the Model View will allow you to identify Contact Nodes. Once the separation between the connected nodes (as per the definitions above) has reduced to 10% of its initial value (i.e. 90% gap closure), both nodes in the contact pair are highlighted.

RELEVANT KEYWORDS

- `*PIP CONNECTION` is used to define pipe-in-pipe connections between nodes of the finite element model.
- `*PIP STIFFNESS` is used to define force-deflection curves for non-linear pipe-in-pipe connection stiffnesses. Specifically, the `TYPE=POWER LAW` input is used to create a non-linear contact relationship based on a power law approach.

If you would like to see an example of how these keywords are used in practice, refer to H02 - J-Tube Pull-In.

Hydrodynamic Forces

OVERVIEW

The main fluid forces modelled by Flexcom include buoyancy, hydrodynamic and internal fluid.

- Buoyancy forces effectively represent the resultant of the sum of all the pressure forces acting on a particular element. Theoretically, the net buoyancy force may be shown to correspond exactly to the volume of displaced fluid, and this is the modelling approach adopted by Flexcom. Refer to Buoyancy Forces for further details.
- Hydrodynamic forces are computed based on Morison’s Equation, and include three distinct components, drag, added mass and hydrodynamic inertia. Refer to Hydrodynamic Loading for further details.

* Flexcom provides a comprehensive internal fluid modelling capability. Stationary internal fluids, uniform steady state internal fluid flow and multi-phase slug flow may all be modelled. Refer to Internal Fluid for further details.

Each of these three areas is also influenced by the presence of pipe-in-pipe data. In the absence of pipe-in-pipe, the program operates in standard fashion, as outlined in the relevant sections quoted above. Pipe-in-pipe introduces some subtle differences over the standard modelling procedures, as now documented.

PIPE-ON-PIPE CONFIGURATIONS

For pipe-on-pipe set-ups, the primary and secondary sections are both exposed to the ambient environment. There are no ‘outer’ or ‘inner’ sections, as none have been defined using `*PIP SECTION`. So the program operation in terms of buoyancy, hydrodynamics and internal fluid is the same as normal, and the remainder of this page is immaterial.

OUTER AND INNER ‘ELEMENT PAIRS’

For pipe-in-pipe configurations, there are clearly defined ‘outer’ or ‘inner’ sections, as defined via the `*PIP SECTION` keyword. This naturally leads on to the concept of outer and inner ‘element pairs’.

* Each element in the inner set is associated with a corresponding element on the outer set through pipe-in-pipe connections, if present.

* The outer element is chosen such that the physical separation between the mid-point of both inner and outer elements is as little as possible.

* Given that the pipes can move independently over the course of a simulation, the optimal inner/outer element pairs are revised at every solution time step.

BUOYANCY FORCES

* Buoyancy forces on outer elements are computed in the standard fashion based on the density of the ambient fluid (typically seawater).

* Buoyancy forces on inner elements are computed based on the density of the annular fluid.

HYDRODYNAMIC FORCES

- Hydrodynamic forces on elements that are exposed to prescribed water particle motions are computed in the standard fashion based on Morison’s Equation.
- Drag forces on inner elements are based on the density of the annular fluid, and the relative velocity between the inner element and the outer element (the annular fluid motion is assumed to correspond with that of the outer element).
- Hydrodynamic inertia terms associated with the inner elements are based on the annular fluid, and the structural acceleration of the outer element (the annular fluid motion is assumed to correspond with that of the outer element).
- Added mass terms associated with the inner elements are based on the density of the annular fluid, and the structural acceleration of the inner element.

Refer to Drag and Inertia on Inner Pipe Sections for further information on this important topic.

INTERNAL FLUID

The presence of internal fluid induces the following load terms.

- Gravitational and Inertial Forces
- Centrifugal Forces
- Coriolis Force
- Hydrostatic Pressure
- Dynamic Pressure

The modelling of internal fluid within a particular pipe section is unaffected by any inputs pertaining to pipe-in-pipe. Each pipe is filled with internal fluid as dictated by the `*INTERNAL FLUID` keyword, and all computations are performed in standard fashion. Consider the mass of annular fluid for example, which contributes to static (gravitational) and dynamic (inertial) loads. The mass of annular fluid contained within an outer pipe section, is defined as the internal area (derived from internal diameter) of the outer pipe, times the pipe length, times the fluid density. So the internal volume in this case is the total internal volume, rather than the annular volume (i.e. the volume computation disregards the presence of the inner pipe). Conceptually this may sound counter intuitive, but it is theoretically consistent with the computation of the buoyancy force, which is based on pressure differentials around the inner surface of the outer pipe. The pressure exerted on the inner wall of the outer pipe section is unaffected by the presence or absence of the inner pipe. This computational methodology has also been investigated and verified using a range of unit test cases.

Drag and Inertia on Inner Pipe Sections

OVERVIEW

Unlike elements that are exposed to the prescribed water particle motions, the drag, hydrodynamic inertia and added mass terms of the inner pipe-in-pipe elements are accommodated into the Finite Element Equation of Motion via the global damping matrix (for drag forces) and global mass matrix (for the hydrodynamic inertia and added mass). This is necessary because the fluid motion used to determine these loads depends on unknown solution variables that are not prescribed (as is the case for the ambient fluid). Further detail on the relevant implementation is outlined in the following points:

- Drag and hydrodynamic inertia loads depend on the motion of the surrounding fluid, typically the prescribed motion of the ambient sea water. The instantaneous hydrodynamic drag and inertia of a standard element surrounded by ambient seawater is therefore a known external force as it does not depend on any solution variables. In this case, the drag and hydrodynamic inertia is computed directly, and added to the global force vector on the right hand side of the finite element equations of motion.

- In the case of an inner pipe-in-pipe element, the motion of the annular fluid is assumed to be consistent with that of the outer element. Nodal velocities and accelerations in Flexcom are computed by taking derivatives of the instantaneous solution displacement variables. Therefore any terms dependent on the instantaneous motion of a finite element node are calculated as part of the finite element solution itself. For example, the added mass is computed alongside the element mass, and is included in the global mass matrix, as it depends on the instantaneous nodal acceleration which is itself a solution variable dependent term.

- While conceptually similar to added mass, incorporation of the hydrodynamic inertia load for inner elements is not quite so straightforward. Rather than the inner pipe acceleration, it is actually the acceleration of the outer pipe that is used to calculate the hydrodynamic inertia on the inner pipe.

- In a way similar to the hydrodynamic inertia above, the drag load applied to the inner elements has a dependency on the outer node velocity and modifications to the finite element connectivity for inner elements are made to capture this dependency also.

ADDITIONAL CONNECTIONS TO SUPPORT INNER PIPE HYDRODYNAMICS

As discussed above, drag forces and hydrodynamic inertia on inner pipe-in-pipe elements are modelled as mass and damping terms on the left hand side of the equations of motion, capturing the required coupling between the outer node’s velocity/acceleration and the inner node loading. However this was not always the case, and in versions prior to Flexcom 8.6.3, drag and inertia were included directly as force terms on the right hand side of the equations of motion. This approach was theoretically incorrect, as it meant that the outer node’s velocity and acceleration at the previous time step was used to compute the drag forces and hydrodynamic inertia on the inner node at the current time step. Notwithstanding the theoretical limitation, it is conceivable that some software users may occasionally wish to invoke this modelling approach (for example, to quantify any level of inaccuracy associated with an earlier study). For this reason, it is possible to select the modelling approach (be it left hand side or right hand side) via the `*PIP SECTION` keyword (`INNER_HYDRO` option).

An inherent prerequisite for the left hand side modelling approach is that pipe-in-pipe connections must exist in the global connectivity matrix. In versions prior to Flexcom 8.10.1, such connections had to be explicitly created by the user, if there was no physical connection between all inner nodes and the outer pipe. In more recent versions of the software, token connections of zero stiffness are automatically inserted where required to ensure that hydrodynamic loading on all inner nodes is modelled. The procedure is as follows:

- Flexcom firstly identifies all potentially 'isolated inner nodes'. These are inner nodes which are not guaranteed to have a pipe-in-pipe connection to an outer node at all times during the simulation i.e. they do not form part of any standard connection, and they do not act as the primary node of any sliding connection (the latter case is highly unlikely, as recommended practice is to designate nodes on the outer pipe as primary nodes - refer to Primary and Secondary Pipes for further details).
- Secondly, a sliding connection is automatically created between each 'isolated inner node' and a suitable outer element set. A suitable element set is one which has been designated as an outer set via the `*PIP SECTION` keyword, and whose corresponding inner element set includes the 'isolated inner node'.

One disadvantage of this modelling procedure is that it may create a large number of additional pipe-in-pipe connections, which can have a negative impact on run-time performance. For example, let's suppose you have created a pipe-in-pipe model which has 100 nodes on the outer pipe and 100 nodes on the outer pipe. If the mesh densities are identical on both pipes, and you are expecting little or no relative axial sliding, then you would typically create 100 standard connections between the pipes. In this case, all inner nodes are guaranteed to have a pipe-in-pipe connection to an outer node at all times during the simulation, and no additional connections are created by the software.

If the mesh densities are similar on both pipes, but not identical, you might create 100 sliding connections, with each one linking an outer node with a set of inner elements. This means that the program will automatically determine the optimum configuration thereby reducing effort on your part (i.e. Flexcom figures out which nodes should be connected). As you are not anticipating any relative sliding, you would typically include the `*NO PIP SLIDING` keyword also for computational efficiency. This means that once the optimal configuration has been attained, the connections are effectively treated as standard rather than sliding from that point onwards. However, in order to guarantee with absolute certainty that no inner node is isolated from the outer pipe at any point during the entire simulation, Flexcom will automatically create 100 additional connections, with each one linking an inner node with a set of outer elements. So you now have 200 pipe-in-pipe connections rather than 100. This approach may be viewed as overly conservative by some software users for the following reasons:

- As the mesh densities are similar on both pipes, the majority of inner nodes will have an active connection to an outer node at all times during the simulation regardless.
- The additional connections are created and monitored purely as a fail-safe mechanism, and it is highly likely that most of them will be redundant at any given time during the simulation.
- Assuming that a small number of inner nodes do become 'isolated' at various stages during the simulation, the omission of a small proportion of the overall hydrodynamic loads is arguably unlikely to have a significant effect on the global response of an offshore structure.

In summary, some users may feel that the additional computational effort associated with the additional connections is simply not justified. So you have the option to:

- Suppress the creation of the additional connections via the `*PIP SECTION` keyword (`AUTO_CREATE` option).
- Check if any inner nodes become 'isolated' during the simulation, via the `*PRINT` keyword.

It should be noted that recommended practice is to accept the additional connections which guarantee correct application of hydrodynamic loading to all inner nodes at all times. If you reject the additional connections, then it is your responsibility to ensure that the model behaviour is consistent with expectations. In this case, some sensitivity studies would certainly be advisable.

RELEVANT KEYWORDS

- `*PIP CONNECTION` is used to define pipe-in-pipe connections between nodes of the finite element model.
- `*NO PIP SLIDING` is used to disable the interchangeable nature of sliding pipe-in-pipe connections.
- `*PIP SECTION` is used to define internal and external pipe sections when part of a pipe model is contained within another.

If you would like to see an example of how these keywords are used in practice, refer to A03 - Pipe-in-Pipe Production Riser (Standard Connections) and H02 - J-Tube Pull-In (Sliding Connections).

1.9.2.6 Enviroment

OVERVIEW

The various program options for modelling the ambient ocean environment and its associated loading are discussed in the following sections:

- Environmental Parameters introduces some fundamental parameters such as water depth, density of seawater, the gravitational constant, and the kinematic viscosity of seawater.
- Buoyancy Forces outlines some basic theory relating to buoyancy loads.
- Buoyancy Formulations discusses two distinct options for specifying how buoyancy forces generated by elements are determined, Default and Distributed.
- Hydrodynamic Loading discusses how the various hydrodynamic forces are computed based on Morison’s Equation, including the three distinct components of drag, add mass and added inertia.

- Partially Submerged Elements outlines how the buoyancy and hydrodynamic load computations are adjusted to reflect the reduced areas and volumes associated with partially submerged elements.

RELEVANT KEYWORDS

- `*OCEAN` is used to specify general parameters defining the ocean environment.

Environmental Parameters

OVERVIEW

This section describes the parameters to mandatory provide the minimum environment specification.

Flexcom traditionally groups together four parameters required to provide the minimum environment specification, namely the analysis water depth, the density of seawater ρ_w, the gravitational constant g, and ν, the kinematic viscosity of seawater. Strictly speaking the gravitational constant is not an environmental parameter, but as it is a common parameter which affects the vast majority of all input data, it is grouped together with the ambient ocean parameters for convenience. Note also that the value of the kinematic viscosity is used only to determine the value of Reynolds number in the case where you have specified hydrodynamic coefficients as a function of Reynolds number. If you do not invoke this facility, the value of ν is immaterial and unused.

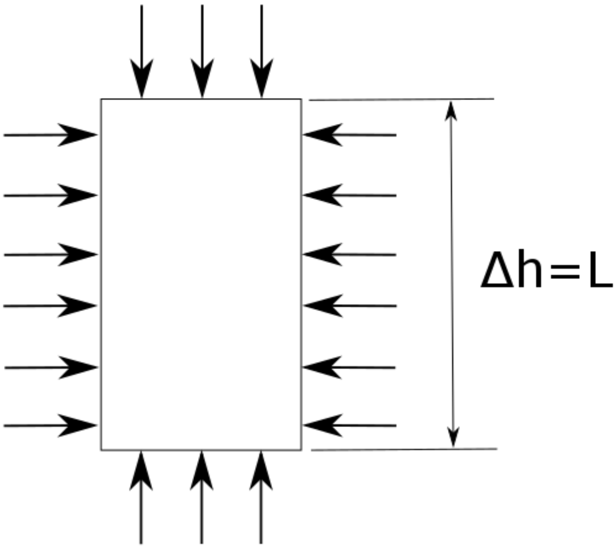
RELEVANT KEYWORDS

- `*OCEAN` is used to specify general parameters defining the ocean environment.

If you would like to see an example of how this keyword is used in practice, refer to any of the standard Flexcom Examples.

Buoyancy Forces

Buoyancy forces effectively represent the resultant of the sum of all the pressure forces acting on a particular element.



Vertical Element

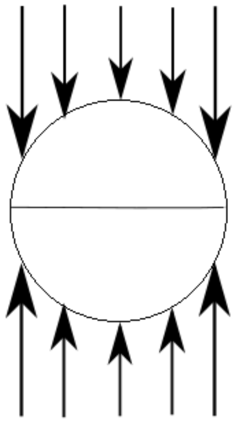
In the simplest possible case of a vertical element, the pressure forces acting around the element circumference cancel each other out so there is no net horizontal force. The pressure forces acting on the element faces produce a net vertical force as follows.

$$F_B = \rho.\underset{\sim}{g}.A.\Delta h \qquad (1)$$

where:

- F_B is the net buoyancy force
- ρ is the density of the ambient fluid (typically seawater)
- g is the acceleration due to gravity
- A is the external cross-sectional area of the element (based on the effective buoyancy diameter)
- Δh represents the change in hydrostatic head (which is equal to the element length in this case)

For a marine structure such a drilling riser, the external cross-sectional area of adjacent elements may be consistent along the riser length, leading to large/concentrated forces at changes in riser cross-section. This point is further elaborated in Buoyancy Formulations.



Horizontal Element

In the case of a horizontally aligned element, the pressure forces acting on the element faces cancel each other out so there is no net horizontal force. The pressure forces acting around the element circumference naturally produce a buoyancy uplift, and while the theoretical derivation of the buoyancy force is more complex that the vertical element case, the fundamental principles of pressure differential are equally valid. Specifically, the total buoyancy force may be found by integrating the pressure differential around the element circumference. Theoretically the net buoyancy force may be shown to correspond exactly to the volume of displaced fluid, and this is the modelling approach adopted by Flexcom.

So whether an element is vertical, horizontal, or inclined, the net buoyancy per unit length applied by Flexcom is equal to

$$F_B = \rho . \underset{\sim}{g} . A \qquad (2)$$

where F_B is the net buoyancy force per unit length, and the other terms are as previously outlined.

This term appears of the right hand side of the Finite Element Equations of Motion. Flexcom evaluates buoyancy forces on an integration point basis, before ultimately transferring the terms to element end nodes.

Buoyancy Formulations

OVERVIEW

Flexcom provides two options for specifying how buoyancy forces on an element are determined, as follows:

- Default buoyancy
- Distributed buoyancy

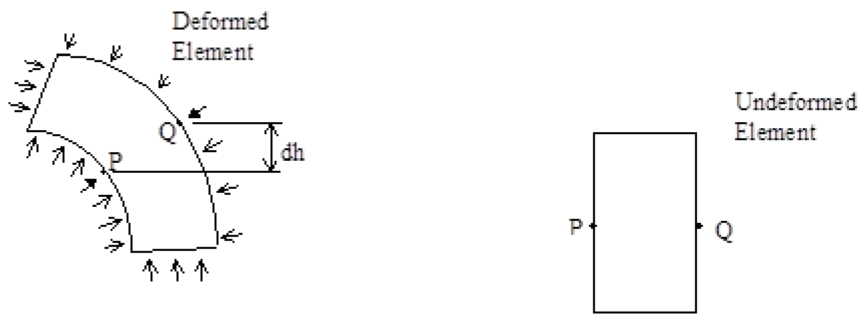
THEORY

Default Buoyancy Formulation

The Default buoyancy formulation, calculates the total buoyancy force on each element as the sum of two components. These are:

i) the pressure forces normal to the riser cross-section, and

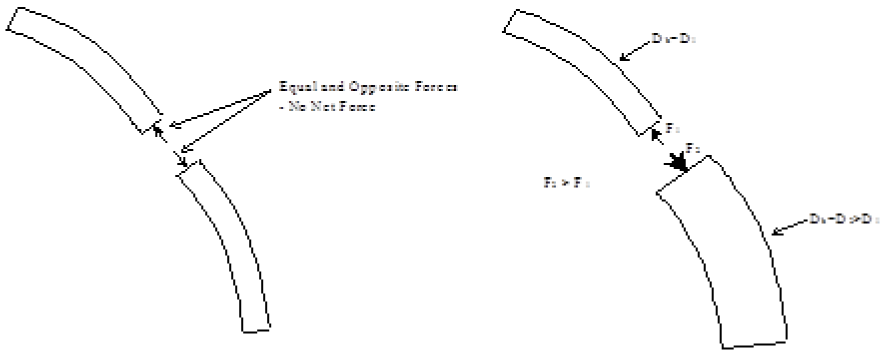
ii) the pressure forces on the ends of the element. These components are illustrated in the figure below.



Buoyancy Components on Riser Element

An expression for the first of these terms is readily derived by integrating around the circumference the net differential pressure experienced between opposite points on the circumference (such as P and Q in the figure above), due to riser rotation and deformation. The second term, the element end forces, is calculated simply as $\rho_w g A_o h$, where ρ_w is the mass density of seawater, **g** is the gravitational constant, A_o is the external area (calculated from the effective buoyancy diameter), and **h** is hydrostatic head.

Buoyancy forces calculated for individual elements are accumulated in the global force vector for the total structure. Where there is no change in external area (Ao) between adjacent elements, the element end forces cancel out when the total force vector is assembled; this is illustrated schematically in Figure (A) below. Where there is a change in cross section, a net force based on the change in areas between the two elements results, as illustrated in Figure (B) below.



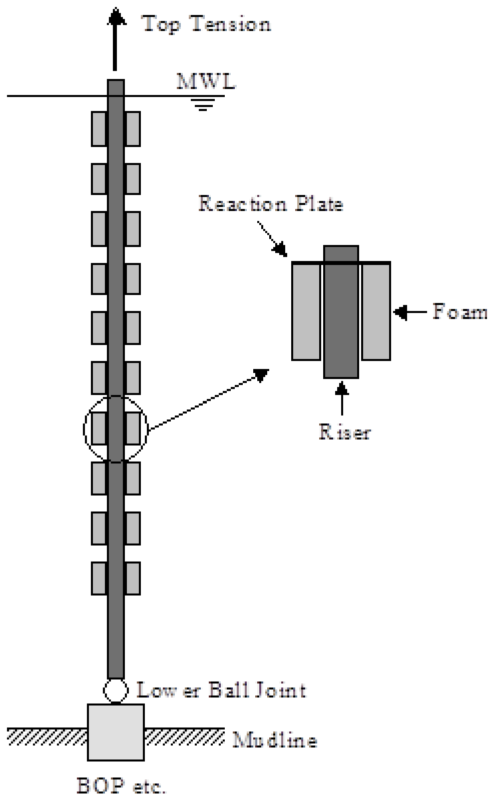
(A) No Change in Cross-Section **(B) Change in Cross-Section**

Forces on Adjacent Elements

For the majority of analyses, this default buoyancy formulation provides the most realistic and accurate approach to modelling the buoyancy forces on the elements. In a small number of cases, however, this may not provide the most realistic buoyancy model.

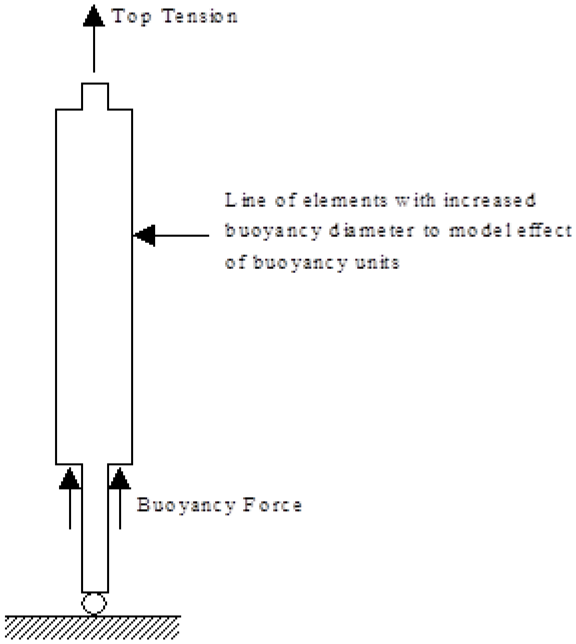
Distributed Buoyancy Formulation

Consider the rigid riser shown schematically in the figure below. It consists of a number of riser joints, many with a foam buoyancy section attached. The buoyancy forces generated by the foam sections are transferred to each riser joint by a reaction plate, located close to the top of the joint. A close-up of a single joint is shown for clarity.



Rigid Riser with Discrete Buoyancy Units

The most convenient way to model this structure using Flexcom is to adopt a smeared buoyancy approach, where the section over which the foam buoyancy units are distributed is modelled as a single line of elements with appropriate buoyancy diameter, as shown in the figure below. If this approach is taken with the Default buoyancy option, it will however result in a large upward buoyancy force at the base of the buoyancy section, with no additional buoyancy forces above this point (because there is no change in section). Such a model will correctly predict the effective tension distribution in the riser, and hence the riser bending response. However the distribution of axial extension and axial force will not be correctly predicted.



Simplified Riser Model with Distributed Buoyancy

This problem could be readily overcome by modelling the riser and buoyancy units separately (but connected at the reaction plates), but this could be quite tedious to set up (particularly in earlier versions of the software, which did not possess the `*LINE SECTION GROUPS` keyword which facilitates the creation of repeating sub-sections). An alternative approach is to use the Distributed buoyancy formulation. With this formulation, the buoyancy forces generated by the riser itself (that is, the riser without buoyancy material attached) are calculated as before, but the buoyancy loads generated by the buoyancy material are applied as a distributed vertical load along the riser. This more closely matches the situation in reality, and correctly predicts the effective tension distribution, the axial force distribution and the axial extension of the riser.

If you use the distributed buoyancy option, then you must provide a value for the actual outer diameter of the riser (without buoyancy material). Flexcom uses this value to calculate the buoyancy loads generated by the riser itself. The program then assumes that buoyancy material completely surrounds the riser, extending out as far as the specified effective buoyancy diameter, and calculates the distributed buoyancy loading generated by this material.

It should be stressed that for the majority of analyses, the default buoyancy option provides the most accurate way of modelling the situation in reality. It is only in a small number of cases (such as that described above) where buoyancy loads generated by a large number of discrete buoyancy units may be approximated by a distributed load, that the distributed buoyancy option should be used.

RELEVANT KEYWORDS

- `*GEOMETRIC SETS` is used to assign geometric properties to element sets, including the specification of buoyancy diameter, and the selection of buoyancy formulation.

If you would like to see an example of how buoyancy modules are modelled using a smeared/distributed buoyancy approach, refer to A01 - Deepwater Drilling Riser.

If you would like to see an example of how buoyancy modules are modelled explicitly, refer to K01 - Worked Example - Simple.

Hydrodynamic Loading

THEORY

Hydrodynamic forces calculated from Morison’s Equation act in general in three dimensions on a structure. The forces combine drag and inertia/added mass components. The drag force at a point on an element is written in terms of the relative fluid/structure velocity $\underset{\sim}{V}_r$ at that point, which is given by

$$\underset{\sim}{V}_r = \underset{\sim}{V}_w + \underset{\sim}{V}_c + \underset{\sim}{V}_u - \underset{\sim}{V}_s \qquad (1)$$

where:

- $\underset{\sim}{V}_w$ is the water particle velocity due to wave action
- $\underset{\sim}{V}_c$ is the current velocity from user-specified constant (time-invariant) current distribution
- $\underset{\sim}{V}_u$ is the velocity specified via a current user subroutine

- $\underset{\sim}{V}_s$ is the structure velocity

Dealing first with $\underset{\sim}{V}_w$ in the case of a regular wave analysis this term is calculated using standard Airy, Stokes V or Dean's Stream Function wave theory, depending on user specification. In the case of a random sea analysis, such as a Pierson-Moskowitz, Jonswap, Ochi-Hubble or Torsethaugen wave spectrum, $\underset{\sim}{V}_w$ is a summation of terms, specifically:

$$\underset{\sim}{V}_w = \sum_{i=1}^{N} \underset{\sim}{V}_{wi} \qquad (2)$$

where:

- $\underset{\sim}{V}_{wi}$ is the water particle velocity due to ith wave harmonic
- N is the number of harmonics into which the wave spectrum is discretised

The spectrum discretisation algorithms are discussed in Spectrum Discretisation. $\underset{\sim}{V}_{wi}$ for each harmonic is calculated using standard linear Airy wave theory, since Stokes or Dean's Stream Function waves cannot be superposed.

The term $\underset{\sim}{V}_s$ is strictly speaking unknown at each solution time. Flexcom uses the structure velocity from the previous iteration in calculating $\underset{\sim}{V}_r$. At the first iteration at a particular solution time, the velocity from the previous solution time is used.

$\underset{\sim}{V}_r$ has components normal and tangential to the element in the convected axes. These are denoted by $\underset{\sim}{\hat{V}}_{rn}$ and $\underset{\sim}{\hat{V}}_{rt}$ respectively, and are given by

$$\underset{\sim}{\hat{V}}_{rn} = \underset{\sim}{V}_r - (\underset{\sim}{V}_r . \underset{\sim}{\hat{i}})\underset{\sim}{\hat{i}} \qquad (3)$$

and

$$\underset{\sim}{\hat{V}}_{rt} = (\underset{\sim}{V}_r \cdot \underset{\sim}{\hat{i}})\underset{\sim}{\hat{i}} \qquad (4)$$

where:

- $\underset{\sim}{\hat{i}}$ is the unit vector in the element axial direction of the convected axes.

  For the inertia/added mass components, expressions are required for water particle and structure acceleration components in the local axes for the Morison's Equation hydrodynamic force. The total water particle acceleration vector in global axes $\underset{\sim}{A}_t$ is defined as

$$\underset{\sim}{A}_t = \underset{\sim}{A}_w + \underset{\sim}{A}_u \qquad (5)$$

where:

- $\underset{\sim}{A}_w$ is the water particle acceleration due to wave action
- $\underset{\sim}{A}_u$ is the acceleration specified via current user subroutine

$\underset{\sim}{A}_w$ is calculated from Airy, Stokes V or Dean's Stream Function wave theory in the case of a regular wave, and from a summation of Airy wave terms in the case of a random sea.

The structure acceleration vector in global axes is denoted $\underset{\sim}{A}_s$. Like $\underset{\sim}{V}_s$, $\underset{\sim}{A}_s$ is unknown at each solution time, but the acceleration term is handled differently to the velocity term, as discussed further below. To calculate hydrodynamic forces, both $\underset{\sim}{A}_t$ and $\underset{\sim}{A}_s$ are finally resolved into components normal and tangential to the convected axes, using analogous expressions to the two Equations above.

The components of the hydrodynamic force in the normal and tangential directions, denoted $\underset{\sim}{\hat{f}}_n$ and $\underset{\sim}{\hat{f}}_t$ respectively, are now defined as

$$\hat{\underset{\sim}{f}}_n = \frac{1}{2}\rho_w d_d C_{dn} \hat{\underset{\sim}{V}}_{rn} \left|\hat{\underset{\sim}{V}}_{rn}\right| + \frac{1}{4}\rho_w \pi d^2 C_{mn} \hat{\underset{\sim}{A}}_{tn} - \frac{1}{4}\rho_w \pi d^2 C_{an} \hat{\underset{\sim}{A}}_{sn} \tag{6}$$

and

$$\hat{\underset{\sim}{f}}_t = \frac{1}{2}\rho_w d_d C_{dt} \hat{\underset{\sim}{V}}_{rt} \left|\hat{\underset{\sim}{V}}_{rt}\right| + \frac{1}{4}\rho_w \pi d^2 C_{mt} \hat{\underset{\sim}{A}}_{tt} - \frac{1}{4}\rho_w \pi d^2 C_{at} \hat{\underset{\sim}{A}}_{st} \tag{7}$$

Morison's Equation

where:

- d_d is the drag diameter of the element
- d is the drag (d_d) or buoyancy (d_b) diameter, depending on user specification
- C_{dn} is the drag coefficient in direction normal to element
- C_{mn} is the inertia coefficient in direction normal to element
- C_{an} is the added mass coefficient in direction normal to element
    - This defaults to $(C_{mn} - 1)$ unless specified otherwise
- $\hat{\underset{\sim}{A}}_{tn}$ is the component of $\underset{\sim}{A}_t$ normal to element
- $\hat{\underset{\sim}{A}}_{sn}$ is the component of $\underset{\sim}{A}_s$ normal to element
- C_{dt} is the drag coefficient in direction tangential to element
- C_{at} is the added mass coefficient in direction tangential to element

- C_{mt} is the inertia coefficient in direction tangential to element
    - note that this term is not a user input – it is assumed to be equal to C_{at}
- $\hat{\underset{\sim}{A}}_{tt}$ is the component of $\underset{\sim}{A}_{t}$ tangential to element
- $\hat{\underset{\sim}{A}}_{st}$ is the component of $\underset{\sim}{A}_{s}$ tangential to element.

The last term in both of these equations is proportional to a component of the structure acceleration $\underset{\sim}{A}_{s}$. These terms are not actually integrated into the right-hand side force vector. Instead they are taken to the left-hand side of the equations of motion and integrated into the element mass matrix.

Traditionally, Flexcom has based the calculation of both the drag and the inertia/added mass components of Morison’s Equation on the drag diameter. This is retained as the default operation to maintain compatibility with older program versions. However, strictly speaking the added mass and inertia terms should be based on displaced volume (defined by the buoyancy diameter) as opposed to projected area (defined by the drag diameter), and an option is now provided to select this approach.

Note that the application of drag and hydrodynamic inertia loads for inner pipe-in-pipe elements requires that pipe-in-pipe connections exist between the inner and outer pipe-in-pipe nodes. Please refer to the Hydrodynamic Forces section for more information on this.

RELEVANT KEYWORDS

Hydrodynamic Coefficients

- *HYDRODYNAMIC SETS is used to assign hydrodynamic coefficients to element sets.

Current

- *CURRENT is used to specify current loading.

Regular Waves

- `*WAVE-REGULAR` is used to specify regular Airy wave loading.
- `*WAVE-STOKES` is used to specify Stokes V regular wave loading.
- `*WAVE-DEANS` is used to specify Dean's Stream regular wave loading.

Random Seas

- `*WAVE-PIERSON-MOSKOWITZ` is used to specify a Pierson-Moskowitz random sea wave spectrum or spectra.
- `*WAVE-JONSWAP` is used to specify a JONSWAP random sea wave spectrum or spectra.
- `*WAVE-OCHI-HUBBLE` is used to specify an Ochi-Hubble random sea wave spectrum or spectra.
- `*WAVE-TORSETHAUGEN` is used to specify a Torsethaugen random sea wave spectrum or spectra.
- `*WAVE-TIME-HISTORY` is used to specify a random seastate in terms of a time history of water surface elevation.
- `*WAVE-USER-DEFINED` is used to specify a User-Defined random sea wave spectrum or spectra.

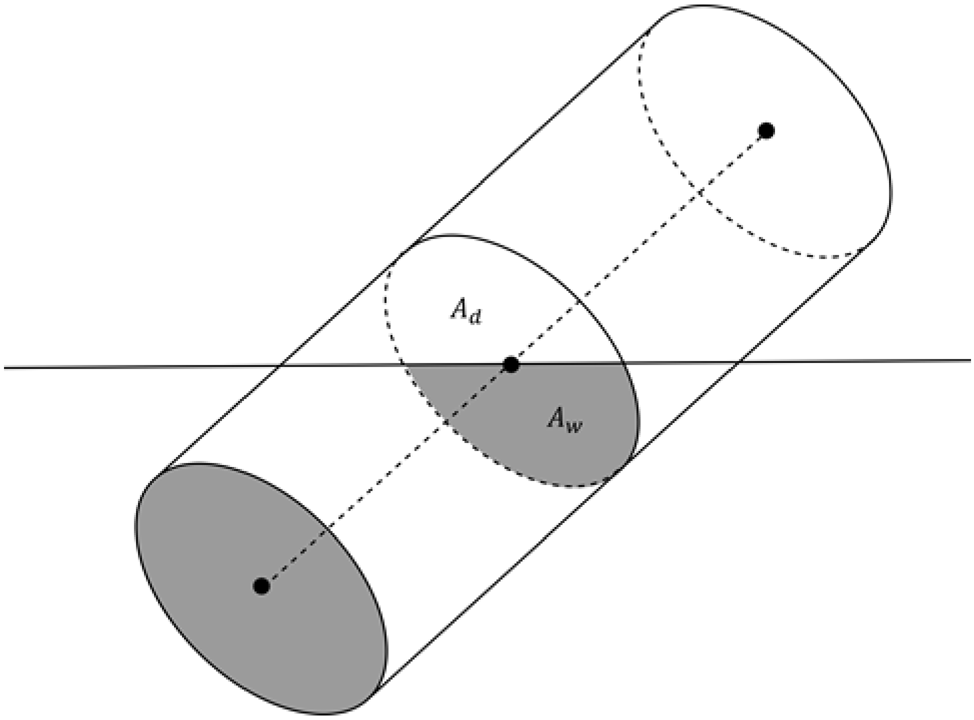
Partially Submerged Elements

When an element is fully submerged, the computation of buoyancy and hydrodynamic loading is relatively straightforward. Specifically…

- The buoyancy forces per unit length are computed based on pressure differentials, and the resultant buoyancy force effectively equates to the volume of displaced fluid. Refer to Buoyancy Forces for further details.
- The hydrodynamic forces are based on Morison's Equation, with separate components in the normal and tangential directions for drag, added mass and added inertia. Refer to Hydrodynamic Loading for further details.

When an element is partially submerged, Flexcom adjusts the various contributions appropriately, to reflect the reduced areas and volumes involved. A number of points are noteworthy…

- Flexcom evaluates all loads on an integration point basis as it is a full finite element solver. These components loads are then transferred to element end nodes using Gaussian Quadrature.
- For surface piercing elements, the number of integration points is temporarily increased to the maximum number (which is equal to 10, assuming the user is not already using this number). This facilitate a precise distribution of the applied forces in the water line region.
- Flexcom computes the instantaneous Water Surface Elevation at each integration point based on the ambient wave loading.
- A planar cross-sectional area is computed at each integration point, normal to local element x-axis (longitudinal), as shown in the schematic below.
- If this cross-sectional area is fully submerged, buoyancy and hydrodynamic loading is applied in full.
- If this cross-sectional area is completely above the water line, buoyancy and hydrodynamic loading are set to zero.
- If this cross-sectional area is partially submerged, the ratio of submerged cross-sectional area (marked Aw in the figure) to total cross-sectional area (Aw+Ad) is used to scale the buoyancy and hydrodynamic loading to account for the effects of partial submergence.



Partially Submerged Element

1.9.2.7 Seabed Interaction

OVERVIEW

Flexcom has a range of options for modelling seabed interaction. Both rigid and elastic seabed models are provided. Rigid Seabeds may be flat, sloping, or have an arbitrary 2D piecewise linear bathymetry. Elastic Seabed geometries may be flat, sloping, or have arbitrary bathymetry specified in 2D or 3D.

Elastic seabed stiffness may be defined using a linear (constant) stiffness or characterised in terms of non-linear force-embedment curves.

Seabeds of either type may be smooth or have longitudinal and/or transverse friction included. For structures running through and below the mudline, soil-structure interaction may also be modelled by the specification of P-y curves.

The various program options for modelling seabed interaction are discussed in the following sections:

- Rigid Seabed Profile describes the various options for defining a rigid seabed bathymetry.

- Elastic Seabed Profile describes additional options available for defining an elastic seabed bathymetry.

- Seabed Contact Modelling describes both the elastic and rigid contact modelling algorithms.

- Seabed Friction describes the Coulomb friction model implemented in Flexcom, and examines the effect of characteristic (mobilisation) length.

- Seabed Modelling in Frequency Domain Analysis discusses how seabed contact is modelled in frequency domain analyses.

- Seabed Modelling in Modal Analysis discusses how seabed contact is modelled in modal analyses.

- Soil Modelling describes the soil-structure interaction modelling facility, for structures running through and below the mudline.

Note that, when determining the effect of hydrodynamic loading, Flexcom calculates water particle velocities and accelerations on the basis of the user-specified (constant) water depth, irrespective of whether an arbitrary profile surface is present in the model or not. In other words, the presence of an arbitrary surface is not considered to affect the hydrodynamic loading on a structure.

RELEVANT KEYWORDS

- `*SEABED PROFILE` is used to specify the seabed profile/bathymetry.

- `*SEABED PROPERTIES` is used to specify properties such as seabed type (i.e. rigid or elastic) and friction parameters.

- `*SEABED STIFFNESS` is used to define nonlinear seabed contact stiffness.

- `*P-Y` is used to define P-y curves for modelling soil structure interaction.

Note also that the old `*ELASTIC SURFACE`, `*RIGID SURFACE` and `*EMBEDMENT` keywords have effectively been superseded by the new seabed definition keywords which facilitate the modelling of arbitrary seabed profiles.

Elastic Seabed Profiles

Elastic seabed profiles may have a bathymetry as described in the following subsections:

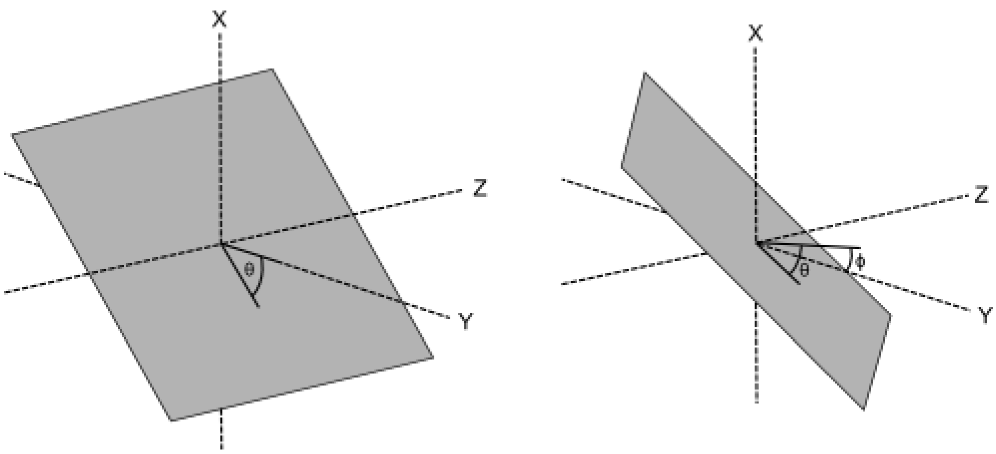
- Sloping Seabed Profile describes the options available for elastic sloping seabeds.
- 2D Seabed Profile describes the Flexcom 2D seabed profile.
- 3D Seabed Profile describes the Flexcom 3D seabed profile.

Sloping Profile

OVERVIEW

If the sloping seabed option is invoked, a uniformly sloping seabed is modelled which passes through the global origin of co-ordinates. The seabed slope is specified in degrees, and a positive slope defines a seabed sloping upwards in the positive global Y direction, while a negative slope gives a seabed sloping in the opposite direction.

In addition to being able to specify the slope, elastic sloping seabed profiles can also be arbitrarily oriented about the global X axis. By default, the profile of a sloping seabed lies in the global XY plane. The orientation can be changed so that the sloping elastic seabed is rotated anti-clockwise about the global X axis by a user specified amount, ?, as shown in the figure below.



Sloping Elastic Seabed Geometry

RELEVANT KEYWORDS

- `*SEABED PROFILE` is used to specify the seabed profile/bathymetry.

- *SEABED PROPERTIES is used to specify properties such as seabed type (i.e. rigid or elastic) and friction parameters.

If you would like to see an example of a sloping seabed profile, refer to C01 - Free Hanging Catenary.

2D Profile

OVERVIEW

In the context of elastic seabed contact, a 2D seabed contact surface is defined by a piecewise curve in the global XY plane horizontally extruded along the global Z axis. The global XY coordinates of the curve segment end points are specified by the user through the Seabed Utility application, which then pre-compiles the input data and generates a file that can be referenced by a Flexcom analysis. Intermediate locations are interpolated.

The 2D seabed file that is generated by the seabed utility is used for:

- Rendering the seabed in the Model View.
- Modeling contact with the seabed during analyses.

See the Seabed Utility section for further details on how to pre-compile a compatible 2D seabed file.

INTERPOLATION OPTIONS

Once the seabed file has been generated it may be referenced in a keyword file under the *SEABED PROFILE keyword. The type of interpolation to be used is specified here also and can be linear, cubic spline or cubic bessel.

Linear

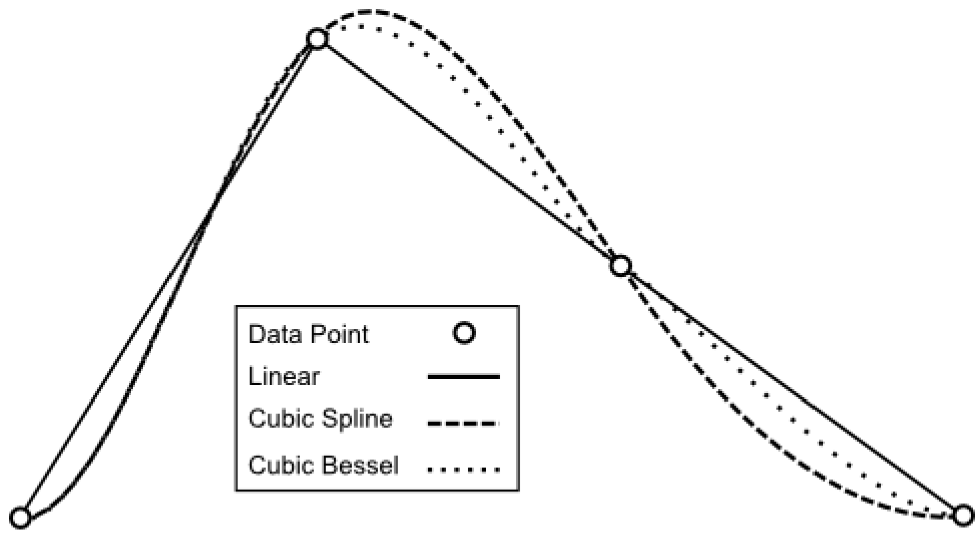
This is the default option and, if selected, the seabed profile is interpolated to lie on the straight line joining the input data points.

Cubic Spline

If this option is selected then a piecewise cubic polynomial which passes through all input points is used to interpolate the seabed profile. The global X elevation and slope at the global YZ interpolation point are taken to be the value and derivative, respectively, of the cubic Hermite spline defined by the curve segment endpoint values and slopes. The slopes at the extreme end points are assumed to be zero. Intermediate input point slopes are obtained by solving a symmetric positive-definite tri-diagonal linear system for the set of slopes associated with a Hermite cubic spline. This method of determining the slopes ensures that the curve has a continuous second derivative.

Cubic Bessel

Also known as parabolic blending, this interpolation option is similar to the cubic spline option in that a cubic Hermite piecewise curve passes through each input data point and the slopes at the extreme end points are set to zero. It differs in the way that the intermediate data point slopes are calculated. The slope at each intermediate data point is determined from the slope of the parabola interpolating the data point and its two neighbouring data points. Because the nodal slopes depend only on adjacent node positions, the cubic Bessel interpolation gives a curve closer in shape to that of the linear interpolation than that of the cubic spline as shown in the figure below.



2D Seabed Profile Interpolation Options

ORIENTATION

The orientation of the 2D seabed profile about the global X axis may be specified also. By default, the profile lies in the global XY plane. The orientation can be changed so that the profile is rotated about the global X axis by a user specified amount in the same way that an elastic sloping profile may be rotated. Finally, the 2D seabed surface is assumed to be horizontal outside the range of the profile and to extend infinitely.

RELEVANT KEYWORDS

- `*SEABED PROFILE` is used to specify the seabed profile/bathymetry.
- `*SEABED PROPERTIES` is used to specify properties such as seabed type (i.e. rigid or elastic) and friction parameters.

If you would like to see an example of a 2D seabed profile, refer to F01 - As-Laid Span Analysis.

3D Profile

OVERVIEW

In the context of Flexcom seabed contact, a 3D seabed profile is defined by a set of 3D scattered data points which lie on the seabed surface. The global XYZ coordinates of the points on the seabed surface must be specified by the user through the Seabed Utility application which then precompiles the data and generates a binary file (with a file extension of .FCSBD) that can be referenced in a Flexcom analysis.

The 3D seabed file that is generated by the seabed utility is used for:

- Rendering the seabed in the Model View.
- Modeling contact with the seabed during analyses.

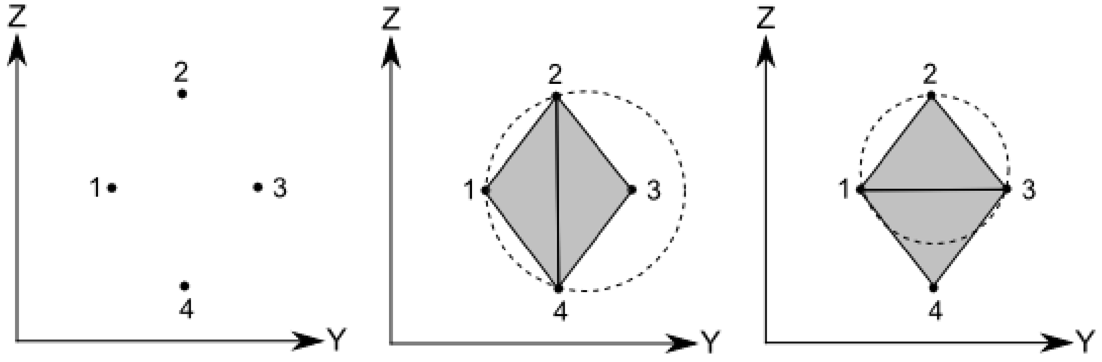
See the Seabed Utility section for further details on how to pre-compile a compatible 3D seabed file.

COMPILATION PROCESS

At a high level, the compilation process involves a number of steps described as follows:

(i) The seabed data points are read from an input text file.

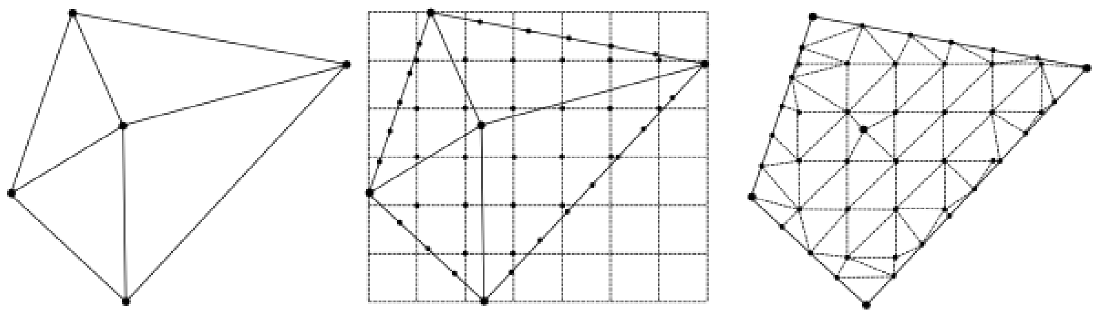
(ii) A Delaunay triangulation on the Flexcom global YZ coordinates of the data points is performed. The resulting triangulation ensures that no data point is inside the circumcircle of any triangle. For example, on the left hand side of the figure below is given the global YZ locations of a set of input data points. There are two possible triangulations of this data as shown in the center and on the right. The Delaunay triangulation of the four points must be the one shown on the right because no data point lies within the circumcircles of the triangles.



Delaunay Triangulation.

(iii) Surface gradients are then determined at each data point for cubic interpolation.

(iv) To render a nonlinear 3D seabed surface in the Model View, the data is further subdivided into regularly spaced points. A Delaunay triangulation is again performed over this new set of gridded points as shown in the figure below and elevations are calculated using the nonlinear interpolation. The grid interval length is chosen based on the user’s maximum triangle side requirement.



Mesh Subdivision for Rendering of Nonlinear 3D Seabed

(v) Finally, gradient and triangulation data both for analysis and rendering purposes is written to the seabed file.

INTERPOLATION OPTIONS

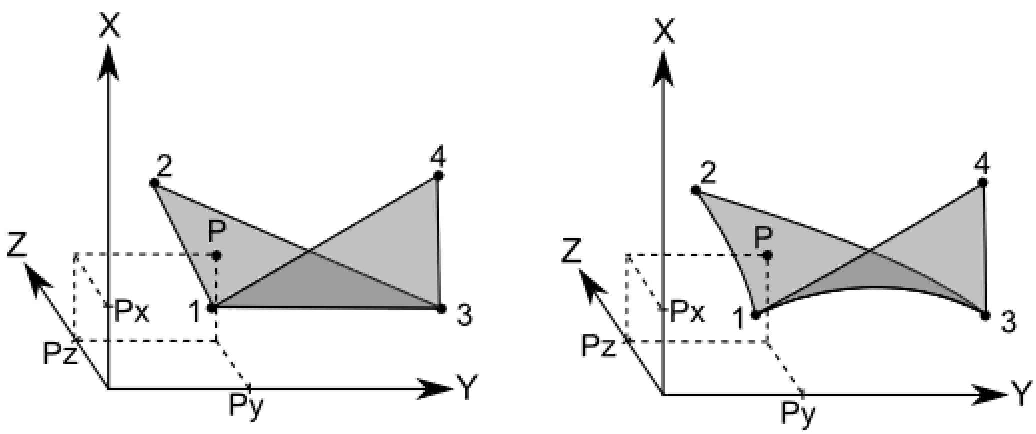
Once the seabed file has been generated it may be referenced in a keyword file under the *SEABED PROFILE keyword. The type of interpolation to be used is specified here also and can be linear or cubic.

Linear

If linear interpolation is selected then the triangles formed during the Delaunay triangulation of the input data form the surface of the seabed. The elevation of each triangle vertex is that specified by the user input data global X coordinate. The elevation of interpolation points whose YZ coordinate lies within a particular triangle is interpolated linearly on the plane of that triangle as shown on the left of the figure below.

Cubic

If cubic interpolation is selected then the triangles formed during the Delaunay triangulation of the input data are used to determine the surface of the seabed. The elevation of each triangle vertex is that specified by the user input data global X coordinate. The elevation of interpolation points whose YZ coordinate lies within a particular triangle is interpolated on a cubic surface over that triangle using the data point surface gradients determined during the seabed file creation process. The cubic surface is defined so that neighbouring triangle edges and gradients are continuous as shown on the right of the below figure.



Linear and Cubic 3D Seabed Interpolation.

Finally, the 3D seabed is assumed to be infinitely deep outside the bounds of the triangulation.

RELEVANT KEYWORDS

- `*SEABED PROFILE` is used to specify the seabed profile/bathymetry.
- `*SEABED PROPERTIES` is used to specify properties such as seabed type (i.e. rigid or elastic) and friction parameters.

If you would like to see an example of a 3D seabed profile, refer to H04 - Pipe Laying.

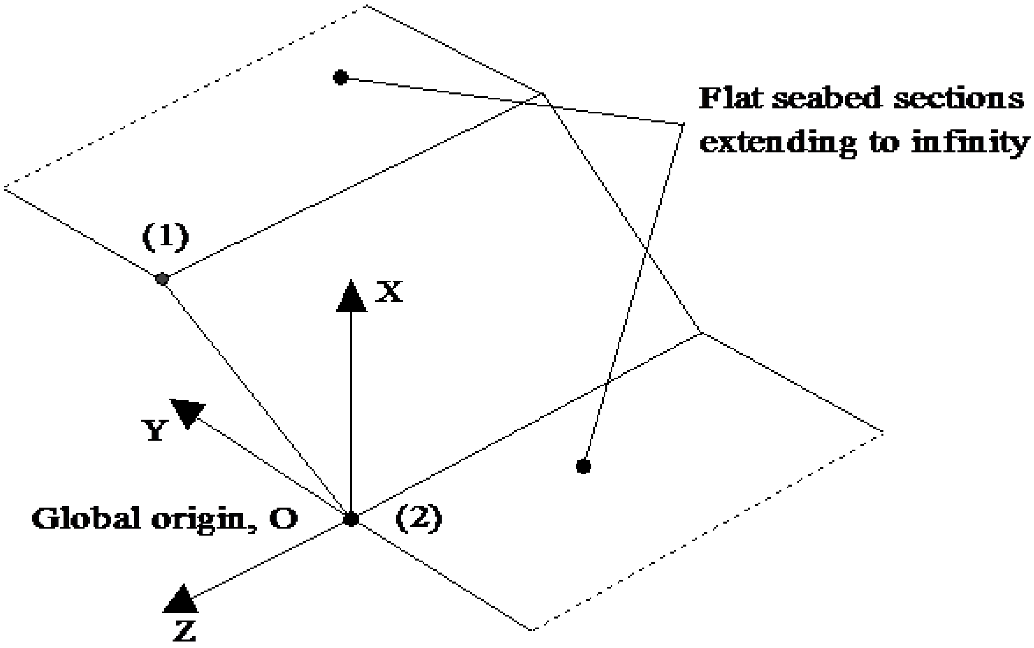
Rigid Seabed Profiles

Rigid seabed profiles may be either flat or sloping, or defined via an arbitrary 2D bathymetry. In the default case of a flat seabed, the seabed is coincident with a plane at zero datum (the global YZ plane). If the sloping seabed option is invoked, a uniformly sloping seabed is modelled which passes through the global origin of co-ordinates. The seabed slope is specified in degrees, and a positive slope defines a seabed sloping upwards in the positive global Y direction, while a negative slope gives a seabed sloping in the opposite direction. For the most general case, Flexcom allows you to define a 2D seabed profile. This is done by specifying an external ASCII data file which contains information on the profile of the seabed surface.

Note that the Elastic Seabed model is the recommended option in Flexcom. Rigid seabed contact involves the application of boundary conditions in a direction normal to the seabed to prevent nodal penetration. In situations where intermittent contact occurs, such as in the touchdown zone of a steel catenary riser, the application and removal of restraints can result in high frequency noise with a consequent requirement for small simulation time steps. Rigid seabed contact is also further complicated by issues of Seabed Penetration and Negative Contact Reactions. In summary, the elastic seabed contact model is the preferred option, except in cases of exceptionally high contact stiffness.

2D Profile

In the context of rigid seabed contact, the profile of the arbitrary 2D surface varies in the global XY plane and is constant in the global Z direction (see figure below). The profile is defined by a series of pairs of values, and this is the data that is contained in the ASCII data file. Each line in the file defines a point on the profile and contains data in the format Y-coordinate, Seabed Elevation. Here Y-coordinate is self-explanatory, and Seabed Elevation is the height of the seabed at this point above the datum X=0. Linear interpolation is used to determine the surface profile between user-specified points, while outside the range of points the surface is assumed to be flat and horizontal.



Definition of Arbitrary Seabed Profile

Two data values are necessary to define the profile shown in the above figure, namely the values for Points (1) and (2). The arbitrary surface file for this example would look like this:

100, 30

0, 0

The first line of data defines Point (1) and the second line defines point (2). Point 1 is a distance of 100 (units) along the positive Y-axis, and the seabed at that point is 30 units above X=0. The seabed slopes uniformly from Point (1) to Point (2). Elsewhere the seabed is flat.

Seabed Contact Modelling

ELASTIC CONTACT MODELLING

Elastic seabed contact is modelled via a direct augmentation of the global stiffness matrix (as opposed to using boundary conditions as in the rigid seabed case) at appropriate locations. Incorporation of some degree of elasticity is typically beneficial when modelling situations of intermittent contact, as the stiffness terms tend to cushion any impact between the structure and the seabed.

The elastic seabed contact algorithm also provides further options for modelling Lateral Resistance to horizontal motion, vertical resistance within a Suction Zone, or non-linear elastic contact via the Embedment facility.

RIGID CONTACT MODELLING

Rigid seabed contact is modelled in Flexcom by checking the positions of all nodes at every iteration. If a node is found to have penetrated the seabed, it is brought back to the mudline, and a boundary condition is applied in the direction normal to the surface to prevent further penetration.

Where there is a large amount of intermittent seabed contact, using a rigid seabed may require relatively small time-steps in order to accurately capture the intermittent impact and avoid high frequency noise, possibly leading to protracted simulation run-times. In such circumstances, it may be beneficial to invoke the elastic seabed option instead, and select a reasonable value of contact stiffness.

Rigid seabed contact is also further complicated by issues of Seabed Penetration, Negative Contact Reactions and the application of Global Boundary Conditions. In summary, the elastic seabed contact model is the preferred option, except in cases of exceptionally high contact stiffness.

IDENTIFYING CONTACT NODES

Switching on node numbers in the Model View will allow you to identify Contact Nodes.

Lateral Resistance

THEORY

The Lateral Seabed Stiffness entry specifies a lateral resistance to horizontal motion on the elastic seabed. The facility is activated by inputting a lateral spring stiffness per unit length, and the resistance this provides is integrated into a seabed stiffness matrix for any element on the mudline in a similar manner to the vertical seabed resistance. Naturally one effect is defined in terms of seabed axes and the other in terms of a local axis system, which may vary from element to element. However this is readily handled by Flexcom. The resistance to horizontal motion provided by the lateral spring replaces rather than augments transverse friction (seabed friction modelling is described later in 'Seabed Friction'). The two effects cannot be combined. If you input a non-zero value for both the lateral seabed stiffness and the transverse friction coefficient, Flexcom outputs a warning and sets the friction coefficient to zero. Note that whichever of these effects is present in an analysis provides lateral resistance to elements on the seabed only, and not to elements in the suction zone.

RELEVANT KEYWORDS

- `*SEABED PROPERTIES` is used to specify properties such as seabed type (i.e. rigid or elastic) and friction parameters. Specifically, the `LATERAL=` input is used to specify a lateral resistance to horizontal motion.

Suction Zone

THEORY

The suction or restraining force experienced by a riser element in a “suction zone” just above the mudline on an elastic seabed is modelled with a linear spring resistance, similar to that provided against downward vertical motion by the elastic seabed itself. The suction force is of course directed downwards whereas the seabed resistance is vertically upwards. The inputs for this facility are suction “spring stiffness” per unit length, and the “suction zone extent” which is defined in terms of the elevation above the mudline at which suction forces disappear. The suction force varies linearly between zero at the mudline and the maximum force experienced at the top of the suction zone. Once an element clears the suction zone the suction force returns to zero. One important point about this facility is that suction forces are applied only to elements moving from the mudline into the suction zone. An element moving into the suction zone from above does not experience any force, nor does an element moving upwards through the suction zone unless it has impacted the mudline before ascending. Finally, the specification of suction forces is optional, and by default no suction effects are included in an analysis.

RELEVANT KEYWORDS

- `*SEABED PROPERTIES` is used to specify properties such as seabed type (i.e. rigid or elastic) and friction parameters. Specifically, the `SUCTION=` input is used to specify suction zone parameters such as suction stiffness and suction zone extent.

Embedment

THEORY

Flexcom provides an option to specify that elastic seabed stiffness varies with the degree of embedment of a riser or pipeline. In this case, seabed stiffness is characterised in terms of non-linear force-embedment curves (as opposed to being defined using a single linear stiffness value).

The embedment ratio of an element is defined as the average distance the element centreline lies below the seabed, divided by element external diameter. So embedment ratio is dimensionless. “Average distance” in this context means the average of the distances below the seabed of the two nodes on the element.

How this facility operates is as follows. Flexcom computes the average embedment of an element lying on the elastic seabed at each iteration at each solution time. Then using the embedment curve for the element, the tangent stiffness of the curve at that embedment is calculated. This value is used as the vertical seabed stiffness for that element.

The facility to specify different embedment curves for different element sets is provided for complete generality. However in the majority of applications the same curve will apply everywhere.

If an element in contact with an elastic seabed is not included in any element set defined for embedment, the global elastic seabed stiffness is used for that element. If no elastic seabed stiffness is specified, then no vertical restraint will be applied at that element, and a solution indeterminacy could potentially occur.

RELEVANT KEYWORDS

- `*SEABED STIFFNESS` is used to define nonlinear seabed contact stiffness.

Note also that the old `*EMBEDMENT` keyword has effectively been superseded by the new `*SEABED STIFFNESS` keyword.

Seabed Penetration

THEORY

Checking for seabed contact with a rigid seabed takes place prior to the solution of the equations of motion. On the other hand, the determination of convergence criteria takes place just after the equations have been solved. A situation can occasionally occur where Flexcom deems convergence to have been achieved, but in fact one or more nodes have penetrated the seabed and would have boundary conditions applied if a further iteration were to occur. This can sometimes lead to unreasonable moments or stresses being reported.

In this case you can use the Rigid Surface Threshold Penetration entry to prevent this occurring. If you input any threshold value, it is an instruction to Flexcom to check all nodes for seabed penetration as part of the convergence checking after solving the equations of motion. Furthermore, if a node has penetrated the seabed to a depth greater than the threshold you specify here, then convergence cannot be deemed to have been achieved, regardless of the maximum convergence measure calculated in the normal fashion. The rationale for specifying a threshold is to prevent unnecessary iterating in the case for example of infinitesimal penetrations.

RELEVANT KEYWORDS

- `*TOLERANCE` is used to define the analysis convergence tolerance measure and related data, including the Rigid Surface Threshold Penetration input.

Negative Contact Reactions

THEORY

For the case of nodes which already have boundary conditions applied due to rigid seabed contact at an earlier solution time, then as part of the seabed monitoring process, Flexcom checks the value of the reaction in the normal direction. If the reaction is positive, this means the node is “pressing down” on the seabed and the boundary condition should be retained. If on the other hand the reaction is negative, then the seabed is “holding onto” the node and the boundary condition should be removed. Again this process is done prior to solving the equations of motion, and again a situation can occasionally arise that convergence is deemed to be achieved when a reaction at a seabed node is negative, and the node would be released if a further iteration were to occur.

You can use the Negative Reaction Threshold entry to prevent this occurring. If you input any threshold value, it is an instruction to Flexcom to check for negative reactions at all seabed nodes as part of the convergence checking after solving the equations of motion. Furthermore, if the magnitude of any reaction is greater than the threshold you specify here, then convergence cannot be deemed to have been achieved, regardless of the maximum convergence measure calculated in the normal fashion. The rationale for specifying a threshold is to prevent unnecessary iterating in the case for example of infinitesimal negative reactions. Note that the threshold value you specify is a magnitude – that is, it should be a positive rather than a negative number.

RELEVANT KEYWORDS

- *TOLERANCE is used to define the analysis convergence tolerance measure and related data, including the Negative Reaction Threshold input.

Global Boundary Conditions

THEORY

Rigid seabed contact is modelled in Flexcom by checking the positions of all nodes at every solution iteration. If a node is found to have penetrated the seabed, it is brought back to the mudline, and a boundary condition is applied in the direction normal to the surface to prevent further penetration. This operation is straight forward for flat seabed surfaces as the local seabed axis system coincides with the global axis system.

This situation is a little more complex for sloping or arbitrary seabeds, where the solution axis system for any contact node is temporarily transformed to local, before responses are transferred back to the global system post-solution. A difficulty arises when a seabed node (e.g. the end of a pipeline) is assigned a constraint (i.e. via the *BOUNDARY keyword) in either DOF1 or DOF2, but not both. Flexcom cannot apply a constraint along an axis which does not coincide with the solution axis system (the seabed axis system in this case), so it simply constrains the node is both local DOFs in such circumstances. Where an additional constraint is applied, the user is notified via an appropriate warning message for transparency.

Note also that where the constrained node lies on a flat section of an arbitrary seabed, the additional constraint is not applied, as it is not actually necessary. For example, this would allow the end of a line to slide horizontally along a flat section of seabed, subject to frictional constraints, if a global constraint was specified in DOF1.

RELEVANT KEYWORDS

- *BOUNDARY is used to define boundary conditions.

Seabed Friction

OVERVIEW

In many areas of marine riser and pipeline analysis, the interaction between the riser or pipeline and the seabed plays an important role in the overall response of the structure. It is important that riser analysis software is capable of accurately modelling this interaction. Flexcom incorporates a robust and accurate seabed friction model that has been developed over successive versions of the program and that has been the subject of extensive verification. The friction model provides an optimum combination of solution accuracy and speed. This section describes the operation of the friction model in detail, and highlights some important practical issues in modelling seabed friction:

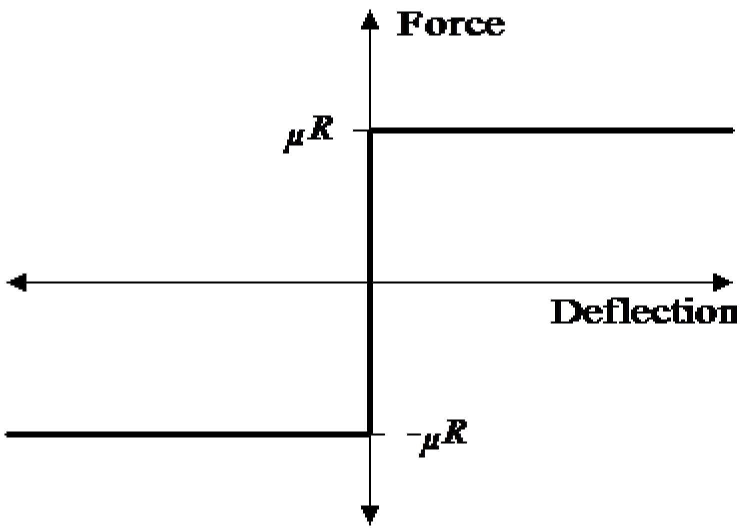
- Friction Modelling Algorithm
- Implications for Seabed Modelling
- Characteristic Length Example

Seabed Friction Modelling Algorithm

THEORY

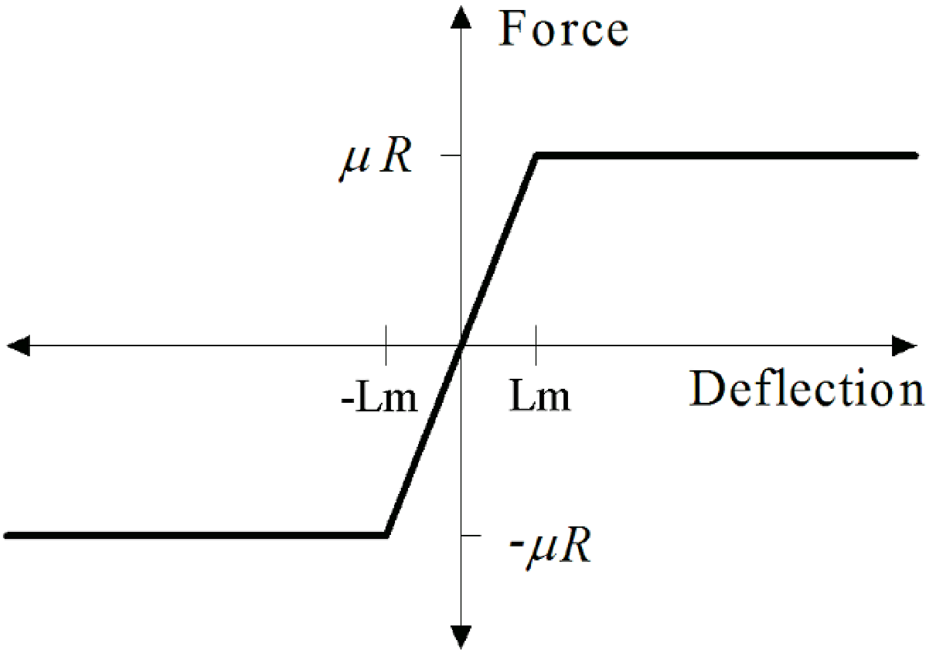
At every iteration at every solution time of a Flexcom analysis, the program monitors all the nodes in the structural model for contact with the seabed (as long as either a rigid or elastic seabed has been specified as part of the model). If a node is in contact with the seabed, and if either or both of the seabed friction coefficients are non-zero, then the program effectively 'attaches' the node to the seabed (in the plane of the seabed). In general, this attachment can be achieved by means of either a boundary condition or a non-linear spring. For a variety of reasons, the approach taken in Flexcom is to use non-linear springs.

In an ideal (that is, Coulomb) friction model, each of these non-linear springs would have a force-deflection relationship such as that shown in the figure below. Consider for the moment that the figure below refers to the longitudinal direction. If there is no longitudinal force on the node, the node does not move (corresponding to zero deflection in the below figure). Indeed, the node should remain in the same location until the total force on the node exceeds the limiting friction force (μR where μ is the friction coefficient and R is the normal reaction), at which point the node may move with this movement resisted by a constant force equal to the limiting friction force.



Ideal Seabed Friction Model

The main difficulty with implementing such a friction model in a finite element program such as Flexcom, which solves for deflections, is that the stiffness of the ideal spring is effectively infinite in the region corresponding to zero deflection. This would make it virtually impossible for the program's iterative solution scheme to converge on the correct solution. To get around this difficulty, a slightly modified non-linear spring characteristic, such as that shown in the figure below, is employed instead.



Flexcom Seabed Friction Model

This spring characteristic has the region around the zero-deflection point replaced by a section of very high (but not infinite) stiffness. This point is crucial to the operation of the seabed friction model. The stiffness of this section of the force-deflection curve is given by the expression:

$$k = \frac{\mu R}{L_m} \qquad (1)$$

Here L_m is what is known as the 'mobilisation length', which is user-configurable.

If you are using the older keyword inputs (i.e. *RIGID SURFACE or *ELASTIC SURFACE), the mobilisation length is based on the 'characteristic length' if it is not explicitly specified. The characteristic length itself is governed by the Maximum Characteristic Length input and the finite element mesh discretisation. Specifically, the characteristic length for a given node is equal to the minimum length of the elements which share the node, or the Maximum Characteristic Length input (which defaults to 3.048m or 10ft), whichever is the lesser. In this case, the mobilisation length is equal to one twentieth (5%) of the characteristic length. Where the Mobilisation Length input is explicitly specified, it takes precedence over the Maximum Characteristic Length entry – in this case the latter value is immaterial as it is unused.

If you are using the newer keyword inputs (i.e. *SEABED PROFILE and *SEABED PROPERTIES), the mobilisation length defaults to 0.15m (or the equivalent in feet) if it is not explicitly specified.

Clearly, the value of L_m affects the stiffness of the non-linear spring – the smaller this value, the greater is the stiffness and the closer the friction model is to an ideal model. However, reducing L_m makes it harder for the program to converge on a solution. Note also that separate mobilisation lengths may be used in both the longitudinal and transverse directions, with the longitudinal value typically being shorter than the transverse one.

The major advantage of using this approach to modelling seabed friction is that it allows the program to find the structure configuration very rapidly (typically in only a few iterations) because it is able to solve directly for displacements. An alternative formulation, using boundary conditions, would in general require significantly more iteration to find the structure configuration. This is because one iteration would be required for each node where limiting friction has been exceeded to be restrained – so for a situation where a substantial section of the model is in contact with the seabed, the number of iterations required could be significant.

RELEVANT KEYWORDS

- *SEABED PROPERTIES is used to specify properties such as seabed type (i.e. rigid or elastic) and friction parameters. Specifically, the FRICTION= input is used to specify friction data such as friction coefficients and mobilisation lengths.

Implications for Seabed Modelling

THEORY

The major implication of this approach to friction modelling is that a node in contact with the seabed has to move a distance equal to the mobilisation length before the full limiting friction force is mobilised. Below this level of deflection, the friction force is proportional to the deflection. This means that the rate of tension dissipation along a length of pipeline on the seabed due to the presence of seabed friction may, in some circumstances, be a function of the mobilisation length. Specifically, there is a transition region in which the tension dissipation will gradually increase from zero (where no friction forces are mobilised) up to the theoretical maximum rate of tension dissipation (where friction forces are fully mobilised).

For the majority of applications, this is of little consequence. There are, however, some situations where it is desirable to be able to limit the maximum value that the characteristic length can take on. Traditionally, Flexcom provided a Maximum Characteristic Length input for this purpose, and while this feature is still retained for compatibility with earlier versions, it has effectively been superseded by an actual Mobilisation Length entry. Both entries basically allow you to control the stiffness of the non-linear springs used to model seabed friction in both longitudinal and transverse directions.

An option is provided to suppress the effects of friction. Very occasionally, it may be desirable to exclude seabed friction, for example, during the initiation of seabed contact in a pipeline analysis. By default frictional effects are modelled (provided at least one non-zero friction coefficient is specified), and this is suitable for the majority of analyses.

RELEVANT KEYWORDS

- `*SEABED PROPERTIES` is used to specify properties such as seabed type (i.e. rigid or elastic) and friction parameters. Specifically, the `FRICTION=` input is used to specify friction data such as friction coefficients and mobilisation lengths.
- `*NO FRICTION` is used to suppress the effects of friction.

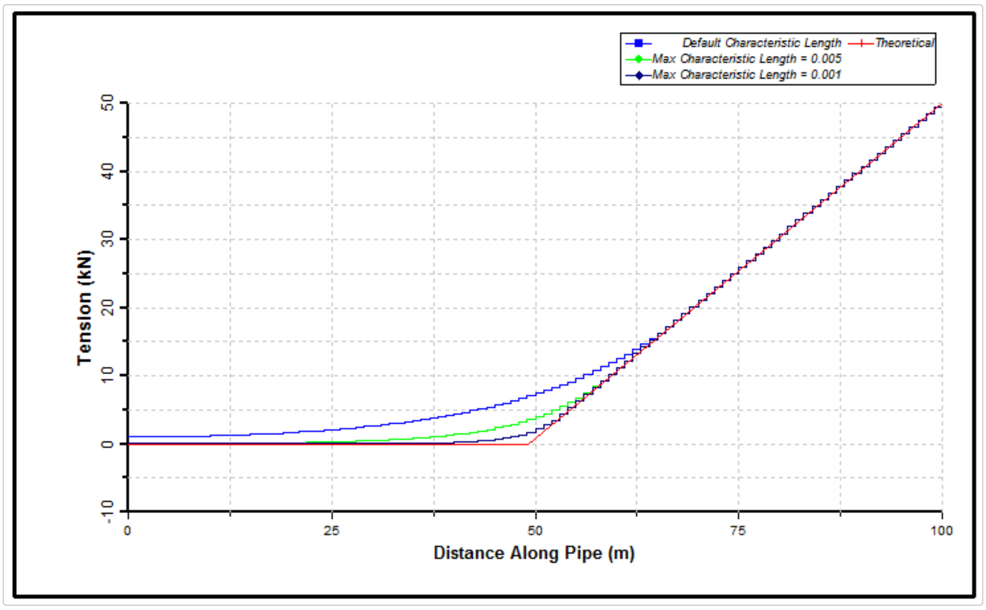
Characteristic Length Example

To illustrate the effect of the Maximum Characteristic Length inputs on the rate of tension dissipation in a section of pipeline lying on the seabed, consider the example of a 100m long section of pipe with a wet weight W of 1962N/m, lying on a seabed with a longitudinal friction coefficient of 0.5.

Now, consider what happens if a 50kN tensile load is applied to one end of the pipe while the other remains fixed. The total tensile load that may be resisted by seabed friction is found from:

$$\begin{aligned} F_{total} &= W\ L\ \mu \\ &= 1962 \times 100 \times 0.5 \\ &= 98.1\,\mathrm{kN} \end{aligned} \qquad (1)$$

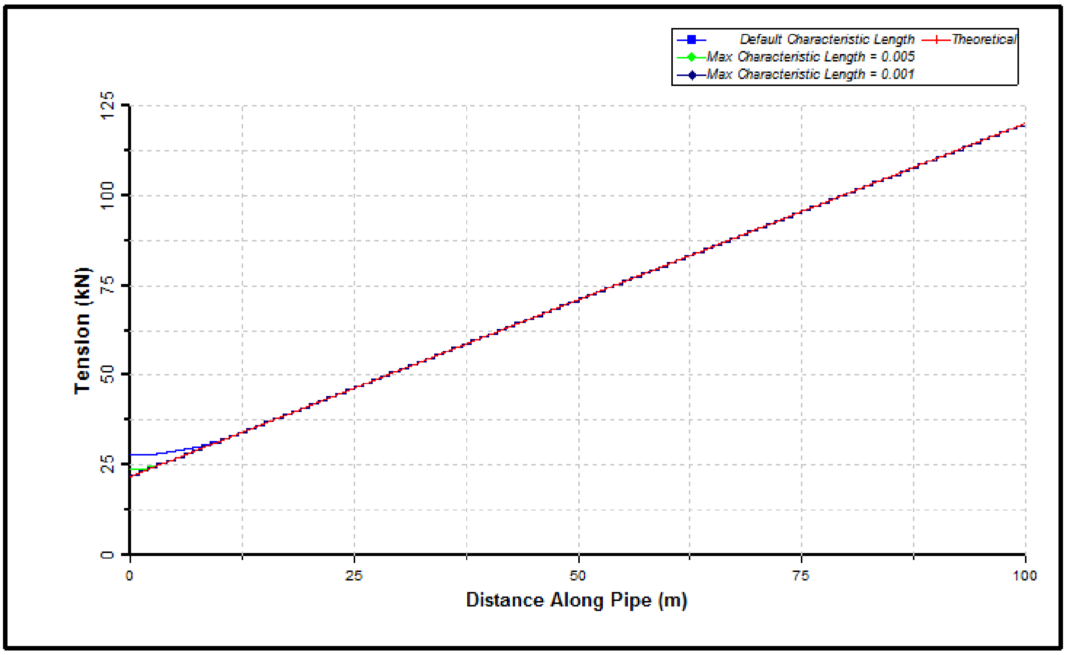
The theoretical maximum rate of tension dissipation is simply m W = 0.981kN/m. The section of pipe is modelled in Flexcom using 100 one-metre elements. The above Equation compares results from three separate analyses of the pipe, one using the default (longitudinal) characteristic length, the second with a user-specified maximum characteristic length of 0.005m and the third with a user-specified maximum characteristic length of 0.001m. The theoretical tension distribution is also shown for comparison purposes.



Effect of Characteristic Length on Tension – 50kN Load

It can be seen that as the characteristic length is reduced, the resulting tension distribution becomes closer to the theoretical distribution, as would be expected. This is, however, at the expense of additional computational effort, as the solution is slower to converge as the characteristic length is reduced.

As the axial load is increased, and the full friction force is mobilised over a larger proportion of the pipe length, the effect on the results of varying the maximum characteristic length is reduced. The above figure shows the tension distribution in the pipe when the axial load is increased to 120kN. It is clear that in this case, each of the tension distributions corresponding to the three values for the maximum characteristic length are very close to the theoretical tension distribution.



Effect of Characteristic Length on Tension – 120kN Load

In conclusion, seabed friction is a highly non-linear and path-dependent effect that can significantly impact on structure response in certain applications. For this reason it is important that riser analysis software is capable of accurately modelling the friction loads caused by the movement of a structure on the seabed. The Flexcom seabed friction model provides a robust, accurate and fast approach to this problem, while simultaneously giving the riser analyst control over the conflicting requirements of accuracy and solution speed.

Seabed Modelling in Frequency Domain Analysis

This section discusses how seabed contact is modelled in frequency domain analyses. If you are analysing an SCR or a flexible riser (e.g. a free hanging, lazy S or lazy wave configuration) in the frequency domain, an important consideration is how the program models the interaction between with the riser and the seabed. There are two aspects to this issue, namely:

(i) Riser motion in the direction normal to the seabed.

(ii) Riser motion tangential to the seabed.

Motion Normal to the Seabed

With regard to motion normal to the seabed, the program operation depends on whether a rigid or elastic seabed is specified.

If the seabed is rigid, then all of the section which the initial static analysis calculates as lying on the seabed is completely restrained in the normal direction during the frequency domain dynamic analysis, as the dynamic phase cannot consider intermittent seabed contact. So nodes which are on the seabed in the initial static solution are restrained there during the dynamic phase, and nodes close to the seabed are free to move in all directions regardless of their proximity to the seafloor. However, during the static analysis phases (initial and full – see Frequency Domain Analysis for further details on frequency domain solution procedure), nodes can move onto or off the seabed. The reaction normal to the seabed at seabed nodes is monitored continuously, and nodes may be released and may move away from the seabed. Likewise nodes which are close to the seabed may move onto the seabed and be restrained there.

If on the other hand, the seabed is elastic, then a seabed stiffness matrix in the direction normal to the seabed is computed in standard fashion for each element calculated by the initial static solution as lying on the seabed. However the nodes on these elements are not restrained in the direction normal to the seabed, but are in fact free to displace.

There is also another additional feature for the case of an elastic seabed. Instead of a constant and single-valued seabed stiffness, you can specify that the stiffness varies as a function of the degree of embedment of a riser or pipeline in the seabed (this is defined in terms of a seabed force/embedment ratio relationship). If you invoke this option, then the seabed will be modelled as a linear elastic contact surface in the frequency domain dynamic analysis (frequency domain analysis is inherently linear by nature), with the stiffness for each contact node being equal to the final contact stiffness attained in the initial static analysis.

Motion Tangential to the Seabed

With regard to motion tangential to the seabed, the program operation in this case depends on whether a smooth seabed or a seabed with friction is specified. If the seabed is smooth, then the section of riser on the seabed is free to displace tangentially to the seabed during the dynamic analysis. If friction is present on the seabed, frictional effects may be modelled using a fully restrained model or a partially restrained model. Each of these is now described in turn.

In the case of the fully restrained model, if friction is present and both friction coefficients are non-zero, each node on the seabed is completely restrained against motion tangential to the seabed during the dynamic analysis. This is true for both rigid and elastic seabeds. If one friction coefficient is zero and one non-zero, the program applies a restraint in the direction (transverse or longitudinal) with the non-zero coefficient, while allowing motion in the other direction. This requires the application of boundary conditions in a local seabed axis system rather than in the global axes.

In the partially restrained model, friction is represented as a stiffness. If friction is present, a stiffness matrix is computed for each element on the seabed in a direction (transverse, longitudinal or both) with a non-zero coefficient. In a direction with a zero coefficient, full free motion is allowed during the dynamic analysis as in the fully restrained model. The longitudinal and transverse ‘friction stiffnesses’ are user specified inputs.

Seabed Modelling in Modal Analysis

If you are analysing an SCR or a flexible riser (e.g. a free hanging, lazy S or lazy wave configuration) in a modal analysis, an important consideration is how the program models the interaction between the riser and the seabed. There are two aspects to this issue, namely:

(i) Riser motion in the direction normal to the seabed.

(ii) Riser motion tangential to the seabed.

Motion Normal to the Seabed

THEORY

With regard to motion normal to the seabed, the program operation depends on whether a rigid or elastic seabed is specified.

If the seabed is rigid, then all of the section which the initial static analysis calculates as lying on the seabed is completely restrained in the normal direction during the modal analysis, as intermittent seabed contact cannot be considered. So nodes which are on the seabed in the initial static solution are restrained there during the modal analysis, and nodes close to the seabed are free to move in all directions regardless of their proximity to the seafloor.

If on the other hand, the seabed is elastic, then a seabed stiffness matrix in the direction normal to the seabed is computed in standard fashion for each element calculated by the initial static solution as lying on the seabed. However the nodes on these elements are not restrained in the direction normal to the seabed, but are in fact free to displace.

There is also another additional feature for the case of an elastic seabed. Instead of a constant and single-valued seabed stiffness, you can specify that the stiffness varies as a function of the degree of embedment of a riser or pipeline in the seabed (this is defined in terms of a seabed force/embedment ratio relationship). If you invoke this option, then the seabed will be modelled as a linear elastic contact surface in the modal analysis (modal analysis is inherently linear by nature), with the stiffness for each contact node being equal to the final contact stiffness attained in the initial static analysis.

RELEVANT KEYWORDS

- `*SEABED PROPERTIES` is used to specify properties such as seabed type (i.e. rigid or elastic) and friction parameters.
- `*SEABED STIFFNESS` is used to define nonlinear seabed contact stiffness.

Motion Tangential to the Seabed

THEORY

With regard to motion tangential to the seabed, the program operation in this case depends on whether a smooth seabed or a seabed with friction is specified. If the seabed is smooth, then the section of riser on the seabed is free to displace tangentially to the seabed during the modal analysis. If friction is present on the seabed, frictional effects may be modelled using a fully restrained model or a partially restrained model. Each of these is now described in turn.

In the case of the fully restrained model, if friction is present and both friction coefficients are non-zero, each node on the seabed is completely restrained against motion tangential to the seabed during the modal analysis. This is true for both rigid and elastic seabeds. If one friction coefficient is zero and one non-zero, the program applies a restraint in the direction (transverse or longitudinal) with the non-zero coefficient, while allowing motion in the other direction. This requires the application of boundary conditions in a local seabed axis system rather than in the global axes.

In the partially restrained model, friction is represented as a stiffness. If friction is present, a stiffness matrix is computed for each element on the seabed in a direction (transverse, longitudinal or both) with a non-zero coefficient. In a direction with a zero coefficient, full free motion is allowed during the modal analysis as in the fully restrained model. The longitudinal and transverse 'friction stiffnesses' are user specified inputs.

RELEVANT KEYWORDS

- `*SEABED PROPERTIES` is used to specify properties such as seabed type (i.e. rigid or elastic) and friction parameters. Specifically, the `FRICTION=` input is used to specify friction data such as friction coefficients and friction stiffness terms.

Soil Modelling

OVERVIEW

For structures running through and below the mudline, soil-structure interaction may be modelled by the specification of P-y and/or T-z curves. These curves define the soil resistance to lateral (P-y) and axial (T-z) motion over a range of depths below the mudline.

Soil structure interaction can be modelled explicitly by manually specifying P-y and/or T-z curves at different depths, or by utilising one of the three built-in soil models which can generate P-y curves automatically. The three pre-programmed soil models are:

- Sand (O'Neill & Murchinson, 1983)
- Soft Clay (Matlock, 1970)
- Stiff Clay (Reese et al., 1975)

Using the P-y and/or T-z curves, Flexcom calculates the appropriate (typically non-linear) soil stiffness to be applied at nodes of the finite element model along the length of the structure below the mudline. Further details about the P-y curves are provided in Operation.

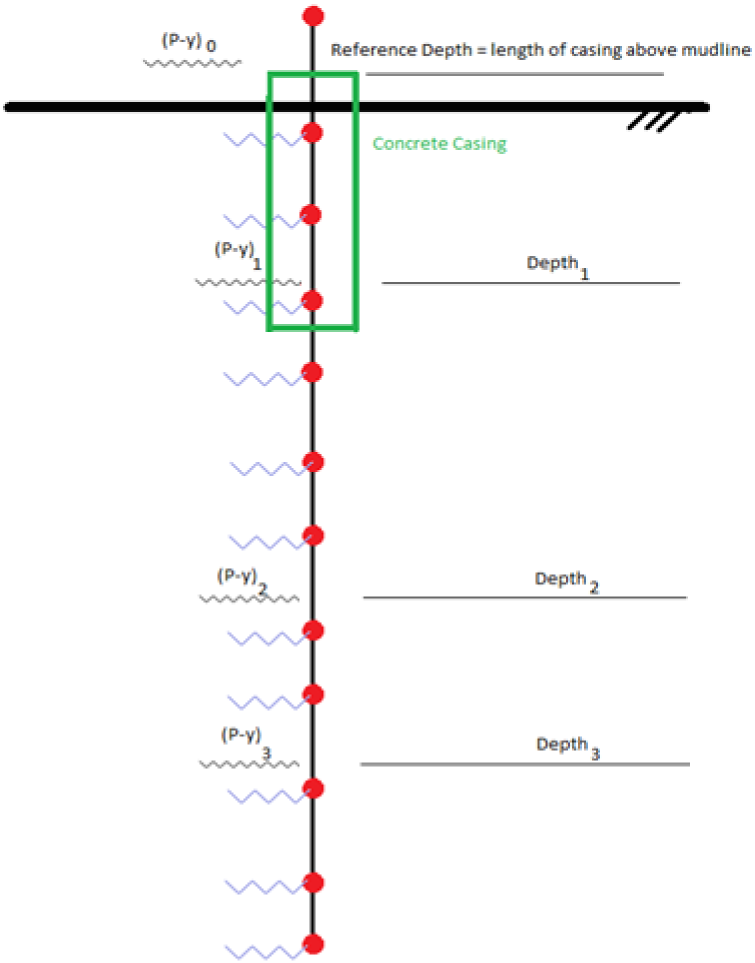
RELEVANT KEYWORDS

- `*P-Y` is used to define P-y curves for modelling soil structure interaction.
- `*T-Z` is used to define T-z curves for modelling soil structure interaction.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser.

Operation

Typically, the uppermost P-y curve data is specified at some level below the mudline, but for complete generality, it is possible to specify P-y curve data at negative depths (i.e. at levels above the mudline). For example, the figure below shows a concrete casing extending above the mudline.



P-y Definition Schematic

A reference depth, d_0, is defined as the depth of the uppermost P-y curve definition, or zero (representing the depth of the mud line), whichever is the higher elevation. Nodes beneath this reference depth are included in the soil resistance model.

Note also that you have the option to apply P-y curves to specific element sets. This allows you to cater for complex models (e.g. pipe-in-pipe configurations), or multiple risers located in different regions with dissimilar soil resistance characteristics. If no such sets are specified, all nodes below the reference depth are considered for soil resistance by default.

The resistance at a depth d between $Depth_1$ *d1* and $Depth_2$ *d2* is for a given lateral displacement y calculated as follows. Firstly the slopes of the P-y curve corresponding d_1 and d_2 are calculated from y: call them S_1 and S_2 respectively. An average slope S(y) is then calculated at d by linear interpolation using:

$$S(y) = w_1(d)S_1(y) + w_2(d)S_2(y) \qquad (1)$$

where w_1 and w_2 are weighting functions defined by:

$$w_1(d) = 1 - \left(\frac{d - d_1}{d_2 - d_1}\right)$$

$$w_2(d) = 1 - w_1(d) \qquad (2)$$

The resistance *P(y)* is then given by:

$$P(y) = S(y).y \qquad (3)$$

The resistance at a depth between the reference depth and the smallest depth for which a specification exists, is calculated as above, assuming a specification exists at the reference depth. If no specification exists at the reference depth (the reference depth could be consistent with the mud line), the above calculation takes place assuming zero resistance at the reference depth. The resistance at a depth greater than the largest depth for which a specification exists, is assumed to remain consistent with the resistance at the largest depth.

The calculated resistance at each node, *P(y)*, is multiplied by the sum of half the projected length on the X-axis of elements sharing that node, to yield the actual resistance force.

One final point to note is that if the soil resistance alters significantly at particular depths, it is good practice to have P-y curves specified at the beginning and end of each different soil section. For example, consider the case of a concrete casing located between depths of 0 and 10, and then mud at all depths greater than 10. In this instance, recommended practice would be to define P-y curves for concrete at depths of 0 and 10, and to define P-y curves for mud at 10.001 (or similar) and then at regular intervals below this depth. This approach should ensure that no nodes are located in a “transition region” between areas of significantly varying soil resistance.

1.9.2.8 Vessels and Vessel Motions

OVERVIEW

The various program options for modelling vessels and associated vessel motions are discussed in the following sections:

- Basic Vessel Concepts introduces the various vessel motion categories and the vessel local axis system.
- Combining Vessel Rotations discusses the available methods (both large and small angle theories) for combining vessel rotations from a number of sources.
- Vessel Offsets discusses the application of static offsets.
- Low Frequency Drift Motions describes the application of second order drift motions.
- High Frequency RAO Motions describes the application of first order (wave frequency) motions.
- Combined High and Low Frequency Motion Timetraces describes the facility to combined low and high frequency vessel motions into motion time histories.
- Summary of Vessel Motion Options summarises the various vessel motion categories, and valid combinations of those divisions.
- Calculating Vessel RAO Response outlines the basic principles underlining the operation of the Flexcom vessel RAO facility.
- RAO Conversions describes the available options to define conventions with respect to the specification of vessel RAO data.
- RAO Layouts describes the different schemes for laying out the data in the RAO file.
- Moonpool Data describes the specification of the vessel moonpool, which is used in the computation of hydrodynamic loading within the moonpool.

RELEVANT KEYWORDS

- `*VESSEL,INTEGRATED` is used to specify all information pertaining to a vessel or vessels.
- `*VESSEL` is used to specify the initial position and undisplaced orientation of a vessel.

- *RAO is used to specify Response Amplitude Operators for a vessel.

Note also that the old *VESSEL and *RAO keywords have effectively been superseded by the new *VESSEL,INTEGRATED keyword, which accepts RAO data also, thereby eliminating the need for a separate *RAO keyword – hence the 'integrated' nature of the new keyword.

- *OFFSET is used to specify an offset of an attached vessel from its initial position.
- *DRIFT is used to define vessel drift motions.
- *VESSEL TIMETRACE is used to specify that the combined high and low frequency motions of a vessel are to be read from an ASCII timetrace data file.
- *RAO FORMAT is used to define custom RAO conversion settings.
  - *MOONPOOL is used to define the location and extent of the vessel moonpool.

If you would like to see an example of how the *VESSEL,INTEGRATED keyword is used in practice, refer to any of the standard Flexcom Examples.

Basic Vessel Concepts

VESSEL MOTION CATEGORIES

You can define vessel motions under five categories in Flexcom, as follows:

- 6 DOF static offset
- Constant vessel forward velocity
- Low frequency drift motions
- High frequency motions calculated from RAOs – for convenience these are termed "RAO motions"
- Combined low and high frequency motions in the form of motion timetraces or time histories – for convenience these are termed "combined motions"

The following sections describe the significance of the various program inputs under each of these headings, and details the conventions used by Flexcom in calculating vessel motions from these user inputs. Many of the inputs and conventions relate to a local axis system centred on the vessel. The inputs defining this local axis system and their significance are first described. Thereafter the procedures used by the program in combining vessel rotations are detailed.

VESSEL SPECIFICATION

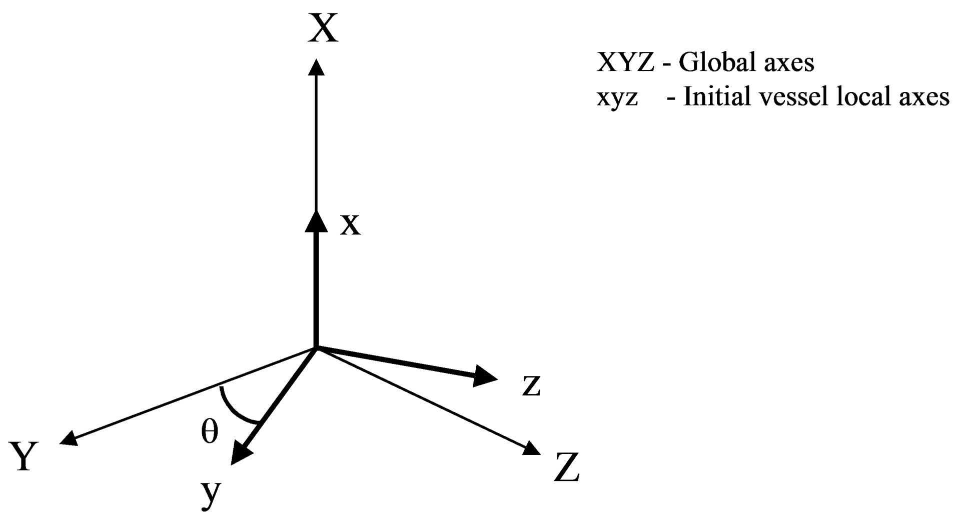
A vessel may be defined using either one of two keywords, `*VESSEL` or `*VESSEL,INTEGRATED`. While the former is perfectly valid, it has effectively been superseded by the latter, which is now the recommended option. `*VESSEL,INTEGRATED` is a more comprehensive keyword and has several advantages over its predecessor:

- It accepts RAO data also, thereby eliminating the need for a separate `*RAO` keyword – hence the 'integrated' nature of the new keyword.
- It offers a range of standard vessel profiles to provide enhanced visual appeal in the structural animation. You may choose to insert an FPSO, Semisub, Drill Ship, Spar, TLP, Installation Vessel etc. or opt to define your own customised vessel shape.
- It allows you to rotate the vessel about a focal point, thereby adjusting its initial orientation without affecting any attached risers, jumpers or mooring lines. This feature is particularly useful for turret moored vessels.

Whichever input strategy you choose, both keywords are responsible for the definition of the initial position of the vessel reference point – this is an important input in Flexcom and represents the point on the vessel for which the RAOs are defined. With `*VESSEL`, this location is defined explicitly using X, Y and Z coordinates in the global axis system. With `*VESSEL,INTEGRATED` the location of vessel's focal point (e.g. turret) is defined via global X, Y and Z coordinates, while the vessel reference point is then defined with respect to the focal point using separations specified in the local vessel heave, surge and sway directions. In many cases however, the focal point will be coincident with the reference point, so the actual inputs in both the new and old formats are in fact identical. Further details are presented in the online help for both keywords. Note also that the keywords are mutually exclusive – the use of one precludes the use of the other in any given analysis or series or analyses.

VESSEL LOCAL AXES

The vessel local axes are centred on the vessel reference point, and initially oriented in a manner which is consistent with the specified initial vessel yaw orientation, which defines the orientation of the vessel local axis system with respect to the global axis system. A single orientation value is sufficient because it is presumed that the local heave axis is vertical in the vessel undisplaced position, so the orientation of the vessel axes is completely defined by the angle (measured positive anti-clockwise) between global Y and the vessel local y or surge axis. See the figure below, where the initial orientation is defined as the angle θ.



Initial Orientation of Vessel Axes

VESSEL RESPONSE PREVIEW OPTIONS

Flexcom provides a helpful Vessel Response Plot option in the user interface. This option displays time histories of the vessel response in all degrees of freedom (heave, surge, sway, yaw, roll and pitch). You are prompted to nominate a particular vessel as you may have more than one vessel defined in your model. The option is particularly useful if you are interested in performing a random sea analysis in the time domain – it affords you an instantaneous preview of vessel motions, without having to undertaking a full 3-hour simulation.

The RAO Response Plot option is a useful feature for verifying that vessel RAO data has been input correctly into Flexcom. In the most general case, RAO magnitudes and phase angles vary with both incident wave heading and wave frequency, so the complete specification of RAO data for any given vessel can be quite extensive. Furthermore, Flexcom has its own set of conventions regarding the specification of RAO data, and the program supports RAO Conversions from third-party software.

Combining Vessel Rotations

LARGE ANGLE VESSEL ROTATIONS

When vessel rotations from a number of sources are specified, an important consideration is how these rotations are combined, and how the displacements of attached nodes resulting from the combined rotations are calculated. This section describes the default Flexcom procedures for these operations, which are based on a large angle theory. The program alternative approach based on classical small angle theory is detailed in the next section. The significance of the various user inputs with respect to these two conventions is then explained in subsequent sections.

The large angle rotations formulation is based on the Flexcom 3D kinematics algorithm. In this, rotations are represented as vectors according to a specific definition. However, because finite (large) rotations do not behave as vector quantities (since they are not commutative), the program does not perform operations directly on these vectors. Instead the following procedure is adopted.

Any rotation “vector” uniquely defines a transformation matrix corresponding to a finite rotation as follows. The transformation matrix defines a rotation about an axis that is given by the direction of the vector, while the actual angle of rotation is given by the magnitude of the rotation vector. So the procedure to combine rotation vectors is that i) the transformation matrix corresponding to each individual rotation is calculated; ii) the transformation matrices are multiplied in a manner consistent with the order of application of the rotations; and finally iii) the combined rotation vector is calculated from the matrix product in a reverse procedure to that used to calculate transformation matrix from rotation vector. How this procedure is used in practice to combine vessel rotations from the various sources available in Flexcom is described in the next section.

At each solution time Flexcom calculates a total transformation matrix corresponding to combined vessel rotations according to the above procedure. This is then used to calculate corresponding motions of an attached node on a riser as follows. If at any time t we denote as $\underset{\sim}{v}(t)$ the vector from the vessel reference point (the point for which the rotations are defined) to the attached node, and if the total transformation matrix is $\underset{=}{T}$, then $\underset{\sim}{v}(t)$ is defined by:

$$\underset{\sim}{v}(t) = \underset{=}{T}\,\underset{\sim}{v}_0 \qquad (1)$$

where $\underset{\sim}{v}_0$ is the vector from initial vessel reference point position to the initial position of the attached node. So the displacement $\underset{\sim}{d}(t)$ of an attached node from its initial position due to vessel rotations is given by:

$$\underset{\sim}{d}(t) = \underset{=}{T}\,\underset{\sim}{v}_0(t) - \underset{\sim}{v}_0 = [\underset{=}{T} - \underset{=}{I}]\,\underset{\sim}{v} \qquad (2)$$

where $\underset{=}{I}$ is the identity matrix.

SMALL ANGLE VESSEL ROTATIONS

When vessel rotations from a number of sources are to be combined using the optional small angle formulation, the procedure is more straightforward. The assumption of small angles means that rotation vectors can indeed be treated as vectors, and so rotations can be combined by the simple addition of rotation vectors. So for example in an analysis with a 6 DOF vessel static offset, drift rotations, and first order rotations calculated from RAOs, the total yaw is the sum of the yaw components from the three sources, and the total roll and pitch are likewise the sums of individual roll and pitch components.

The calculation of attached node displacements corresponding to these rotations is also more straightforward. If we denote as $\underset{\sim}{\theta}(t)$ the total rotation vector at time t whose components are the cumulative vessel yaw, roll and pitch motions, then the attached node displacement vector $\underset{\sim}{d}(t)$ is given by:

$$\underset{\sim}{d}(t) = \underset{\sim}{\theta}(t) \times \underset{\sim 0}{v} \qquad (3)$$

where $\times$ represents the vector cross product.

There is however one circumstance where rotations calculations involving transformation matrices are still used in the small angle case. If a riser attached node is restrained in DOFs 4-6, then the rotations due to vessel motions in any combination at any time are given by $\underset{\sim}{\theta}(t)$. However, in many cases, it is necessary to superpose or add these rotations to mean static riser (rather than vessel) rotations. There is no requirement in general that these riser rotations be small, and so simply adding rotation vectors in this case could lead to errors. So in this case the two rotations vectors are combined via transformation matrices as previously described.

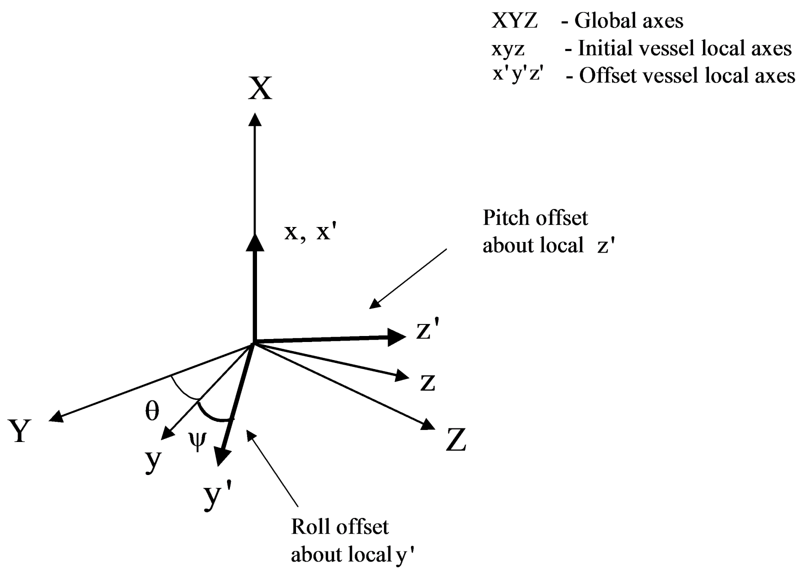
RELEVANT KEYWORDS

- *VESSEL, INTEGRATED is used to specify all information pertaining to a vessel or vessels. Specifically, the ANGLES= input is used to specify the theory used to combine vessel rotations.

Vessel Offsets

LARGE ANGLE THEORY

The Flexcom inputs in this case are three translations and three rotations. The three translations are the components in the global axes of the vessel translational offset. The three rotations are termed the yaw offset, the roll offset and the pitch offset. The yaw offset rotates the vessel and local vessel axes about the vertical or global X direction. See figure below, where the yaw offset is denoted ψ. The orientation of the yawed axes relative to the global axes is simply the sum of θ and ψ.



Vessel Axes following Yaw Offset – Large Angles

The roll offset and the pitch offset define rotations about this yawed vessel axis system.

Specifically, the roll offset defines a rotation about the yawed y' axis in the figure above, while the pitch represents a rotation about the z' axis. In practice these two rotations define two components of a rotation vector in the standard Flexcom kinematics formulation (the third component being zero). Transformations from the vessel yawed axes back to the global axes are then handled in standard Flexcom fashion.

SMALL ANGLE THEORY

The handling of the three static offset displacements is naturally unaffected by the choice of large or small angles, which affects only vessel rotations. However in the small angle case the three rotation offsets, yaw, roll and pitch, define individual rotations about the undisplaced vessel axes.

RELEVANT KEYWORDS

- `*VESSEL, INTEGRATED` is used to specify all information pertaining to a vessel or vessels. Specifically, the `ANGLES=` input is used to specify the theory used to combine vessel rotations.
- `*OFFSET` is used to specify an offset of an attached vessel from its initial position.

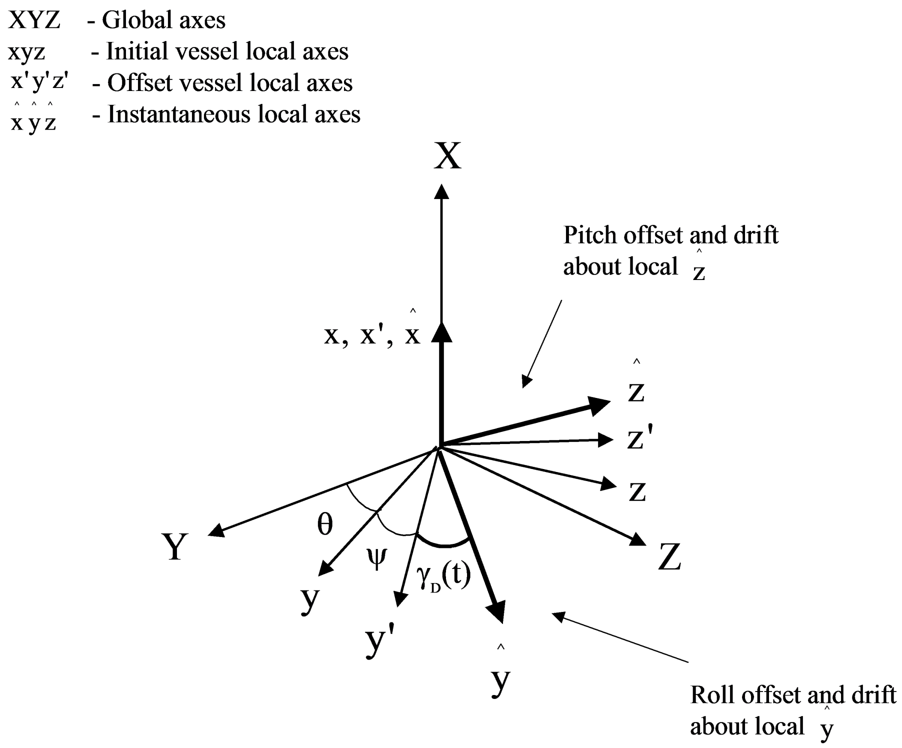
Low Frequency Drift Motions

LARGE ANGLE THEORY

Vessel reference point low frequency or drift motions can be defined in two ways, as sinusoids or via timetrace data read from file, but the actual motions are handled in the same way regardless of which specification you are using. In the most general case you input or define three drift translations and three rotations. The handling of the three displacements is discussed first.

The drift translations can be input in either the global axes or a so-called local system. The handling of drift displacements in the global axes is straight-forward. If on the other hand you input drift displacements in local axes, Flexcom assumes these are relative to the initial orientation of the vessel axes, as defined in the Initial Orientation of Vessel Axes figure.

The significance of the three drift rotations is very similar to the handling of the static offset rotations. Specifically the so-called yaw drift at any time, which we denote $\gamma_D(t)$, represents a further rotation of the vessel and vessel axes about the global X direction. So at any time, the orientation of the vessel $\hat{y}$ axis relative to global Y is simply the sum of the initial yaw θ, the static yaw Ψ and the instantaneous yaw drift $\gamma_D(t)$. See the figure below. The roll and pitch drift rotations are then defined relative to this instantaneous yawed axis system. Again these two rotations define two components of a standard Flexcom rotation vector. They are combined using standard Flexcom techniques with the static roll and pitch, which are themselves continuously defined (or redefined) relative to the instantaneous $\hat{x}\,\hat{y}\,\hat{z}$ axis system.



Instantaneous Vessel Axes – Large Angles

SMALL ANGLE THEORY

The handling of the three drift displacements is likewise unaffected by the choice of large or small angles, which affects only vessel rotations.

The significance of the three drift rotations in the small angle case is very similar to the handling of the small angle static rotations. Specifically the drift yaw, roll and pitch represent further rotations of the vessel about the undisplaced vessel axes. The combined effect of offset and drift rotations at any time is found by simply adding yaw components to get combined yaw, roll components to get combined roll, and pitch components to get combined drift.

RELEVANT KEYWORDS

- `*VESSEL, INTEGRATED` is used to specify all information pertaining to a vessel or vessels. Specifically, the `ANGLES=` input is used to specify the theory used to combine vessel rotations.
- `*DRIFT` is used to define vessel drift motions.

High Frequency RAO Motions

COMPUTATION OF DISPLACEMENTS

First-order wave motions due to the presence of Regular Airy Waves are computed as follows:

$$d_i(t) = \sum_{n=1}^{N} RAO_i(\theta,\omega)\ a_n\ \cos(-\omega_n t + k_n s_n + \varphi_n + \Phi_i(\theta,\omega)) \qquad (1)$$

where:

- $d_i(t)$ is the instantaneous displacement in a particular degree of freedom, i (heave, surge, sway, yaw, roll or pitch)
- t is the instantaneous solution time in seconds
- N is the number of regular Airy waves
- $RAO_i(\theta,\omega)$ is the RAO magnitude in degree of freedom i, which is a function of Incident Wave Heading θ and wave frequency ω
- a_n is the amplitude of the n^{th} regular wave (MWL to crest or trough)
- k_n is the wave number of the n^{th} regular wave ($k = 2\pi/\lambda$, where λ is wavelength)
- s_n is the horizontal distance from the global X axis to the vessel reference point in the direction of wave propagation for the n^{th} regular wave ($s_n = y \cos\theta_n + z \sin\theta_n$, where y and z are the instantaneous coordinates of the vessel reference point in the global Y and Z axes respectively)
- ω_n is the circular frequency of the n^{th} regular wave
- φ_n is the phase of the n^{th} regular wave relative to zero datum
- $\Phi_i(\theta,\omega)$ is the RAO phase angle in degree of freedom i, which is a function of wave heading θ and wave frequency ω. Note that a positive phase angle denotes a phase lag relative to the incident wave harmonic at the vessel reference point.

If there is only a single regular Airy wave present, then the summation from 1 to N in Equation 1 is obviously not required.

All random seas (including Pierson-Moskowitz, Jonswap, Ochi-Hubble, Torsethaugen and User-defined) are simulated using a series of individual component harmonics which are generated using a Spectrum Discretisation technique. Each harmonic is basically a regular Airy wave, so Equation 1 above remains valid.

A slightly different modelling procedure is used for Stokes V and Dean's Stream waves. You should refer to the relevant articles for further information, where the computation of vessel response is discussed for each of the various response settings.

APPLICATION OF DISPLACEMENTS

Large Angle Theory

The handling of first order or high frequency motions calculated via RAOs is slightly different from the handling of drift motions described earlier. This is because the derivation of RAOs is such that they do genuinely represent motions in vessel degrees of freedom (heave, surge, sway, yaw, roll, and pitch).

So dealing first with translations, the three values calculated by Flexcom define heave, surge and sway relative to the instantaneous yawed vessel axis system of the Instantaneous Vessel Axes – Large Angles figure. So heave represents motion in the global X direction. Surge is directed along an axis in the global YZ plane which is the global Y axis rotated through $\{\theta + \Psi + \gamma_D(t)\}$ about global X. The sway axis is likewise the global Z axis rotated through the same angle.

The three RAO rotations likewise represent yaw, roll and pitch relative to the yawed axis in the Instantaneous Vessel Axes – Large Angles figure. These define the components of a rotation vector in the terminology of the Flexcom kinematics algorithm, and are combined with rotations due to static offset and drift using standard Flexcom techniques.

At a given instant in time, Flexcom calculates the amplitudes of the individual rotations from the wave amplitude and the vessel RAOs. So for example, the amplitude of the yaw response (A_{yaw}) is equal to the wave amplitude times the yaw RAO. The magnitudes of the roll (A_{roll}) and pitch (A_{pitch}) responses are computed in a similar manner. Based on A_{yaw}, A_{roll} and A_{pitch}, a single 'unit rotation vector' is established about which a single rotation takes place. The magnitude of this rotation is:

$$A_{total} = \sqrt{A_{yaw}{}^2 + A_{roll}{}^2 + A_{pitch}{}^2} \qquad (2)$$

The unit vector about which this rotation takes place is defined by the following x, y and z components in the vessel axis system:

$$\underset{\sim}{\hat{V}} = \left\{ \frac{A_{yaw}}{A_{total}}, \frac{A_{roll}}{A_{total}}, \frac{A_{pitch}}{A_{total}} \right\} \qquad (3)$$

The sense of the rotation follows the right hand rule. Specifically, when looking along the unit vector (from the origin, looking along the unit vector) the sense of the rotation is clockwise. For illustration purposes, supposing the individual yaw, roll and pitch terms are all equal to 69.28°, then this would represent an overall rotation of 120° about the vector {0.577, 0.577, 0.577}, rotating clockwise, from the perspective of a viewpoint located at the origin and a direction aligned with the vector.

Of course in an analysis with no low frequency drift motions, the RAO motions are relative to the axis system produced by the vessel static offset, which is shown in the Vessel Axes following Yaw Offset – Large Angles figure. Likewise if there is no offset or drift, RAO motions are defined in terms of the undisplaced axis system of the Initial Orientation of Vessel Axes figure.

Finally, in an analysis where RAOs are specified as a function of wave heading, wave headings at any time are calculated relative to the instantaneous orientation of the vessel axes as shown in the Instantaneous Vessel Axes – Large Angles figure. The calculation of wave heading does not take into account the changing orientation of the axes due to first order yaw. In an analysis with no drift, wave headings are invariant with time, and are calculated with respect to the yawed axes of the Vessel Axes following Yaw Offset – Large Angles figure. If there is neither offset nor drift, then the invariant headings are calculated with respect to the undisplaced axes of the Initial Orientation of Vessel Axes figure.

Small Angle Theory

The three translations calculated from vessel RAOs by Flexcom define heave, surge and sway relative to the undisplaced vessel axes of the Instantaneous Vessel Axes – Large Angles figure in the small angle case.

The three RAO rotations likewise represent yaw, roll and pitch relative to the undisplaced vessel axes. Once again they are combined with offset and drift rotations by summing separately the contributions due to each of yaw, roll and pitch. Finally, where vessel RAOs are specified as a function of wave heading, the heading is calculated relative to the undisplaced vessel axes.

RELEVANT KEYWORDS

- `*VESSEL,INTEGRATED` is used to specify all information pertaining to a vessel or vessels. Specifically, the `ANGLES=` input is used to specify the theory used to combine vessel rotations.
- `*VESSEL` is used to specify the initial position and undisplaced orientation of a vessel.
- `*RAO` is used to specify Response Amplitude Operators for a vessel.

Note also that the old `*VESSEL` and `*RAO` keywords have effectively been superseded by the new `*VESSEL,INTEGRATED` keyword, which accepts RAO data also, thereby eliminating the need for a separate `*RAO` keyword – hence the ‘integrated’ nature of the new keyword.

Combined High and Low Frequency Motion Timetraces

INTRODUCTION

Combined low and high frequency motions are specified in the form of vessel motion timetraces or time histories.

Vessel motions are normally due to wave excitation. If you apply a time history of vessel motion, then you must ensure that you also specify the corresponding wave excitation which generated those vessel motions. This is typically specified via a Time History of Water Surface Elevation.

LARGE ANGLE THEORY

The handling of these motions is similar in some respects to the low frequency drift motions. Like the drift translations, combined translations can be specified in global or so-called local axes. Local translations if specified are assumed to be relative to the initial vessel axes (shown in the Initial Orientation of Vessel Axes figure).

The first rotation read from the combined motions timetrace file is assumed to define a yaw rotation which rotates vessel and axes about the vertical axis. And like in the case of drift rotations, the remaining rotations are assumed to define roll and pitch respectively relative to this instantaneous yawed axis system, whose rotation from global Y is the sum of initial rotation θ, the yaw offset ψ and the combined yaw read from the timetrace file. Likewise vessel headings are calculated with respect to the instantaneous yawed axis system.

SMALL ANGLE THEORY

The handling of these motions is very similar to that described earlier for RAO motions. Translations specified in local axes define motions relative to the undisplaced vessel axes. Likewise rotations define yaw, roll and pitch relative to the initial vessel orientation, and are simply added to corresponding static offset rotations if these are non-zero.

RELEVANT KEYWORDS

- `*VESSEL TIMETRACE` is used to specify that the combined high and low frequency motions of a vessel are to be read from an ASCII timetrace data file.

Summary of Vessel Motion Options

To summarise the above, the specification of vessel motion data in Flexcom falls into seven categories or headings, many of which are optional. These are:

(i) Vessel initial position

(ii) Static vessel offset

(iii)Constant forward vessel velocity

(iv)RAO data

(v) Drift data, whether sinusoidal or timetrace

(vi)Combined motion timetrace

(vii)Large or small angles formulation

It must be noted that not all of inputs (i) to (vi) in this list can be used in the same analysis. For example, if you invoke the combined motion timetrace option, then by definition you cannot also specify RAO data or drift data. In practice, options can be specified in only one of two possible combinations, which are defined as follows – the use of the [ ] brackets indicate a particular option is optional.

Combination 1: Vessel initial position [+ static offset] [+ constant velocity] [+ drift] [+ RAOs]

Combination 2: Vessel initial position [+ static offset] [+ combined motion timetrace]

Calculating Vessel RAO Response

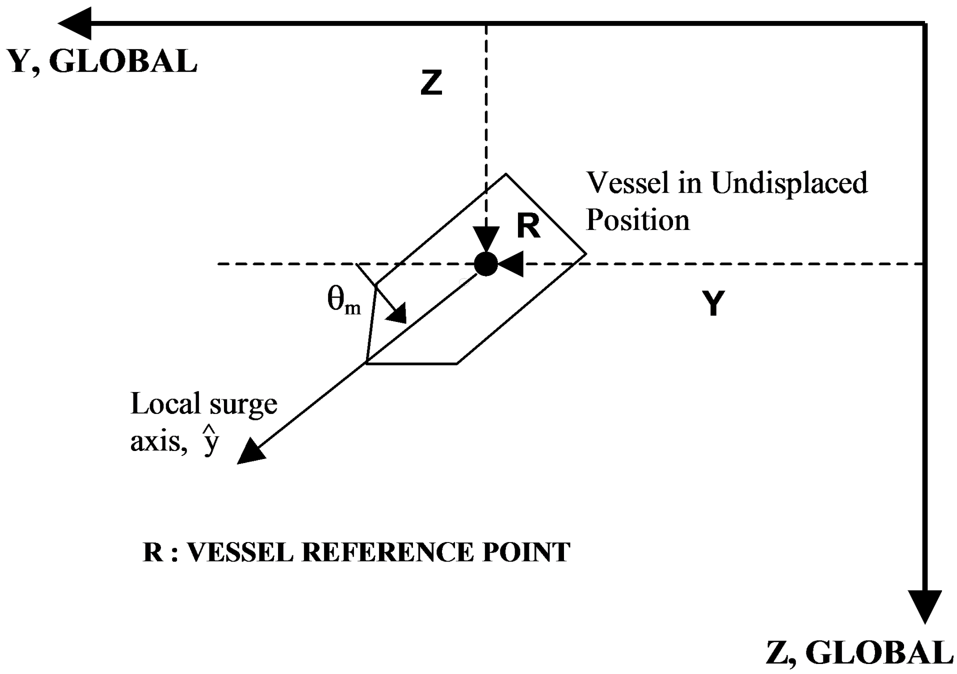
This section outlines the basic principles underlining the operation of the Flexcom vessel RAO facility. The most important point to note is that the vessel RAO data refers exclusively to the attached vessel or vessels, and not to the structure being analysed. The data specified in each RAO file is used to define the motion of a reference point on the vessel. This is normally, but not necessarily, the vessel centre of gravity. This point need not correspond to any point on the actual structure, and indeed in most cases will not do so. The point or points on the structure to which the vessels are attached are specified as boundary nodes. The motion(s) of these points are calculated by the program from the vessel motions, without user intervention and automatically taking into account the effect of any spatial separation between the point(s) on the structure and the vessel reference point.

Further information on this topic is contained in the following sections:

- Relevant Inputs
- Vessel Degree of Freedom
- Vessel RAOs
- Wave Heading
- Flexcom Waves
- Miscellaneous

Relevant Inputs

Three sets of inputs are required to completely define the response of each floating vessel to the ambient wave field (the use of one set is optional). The first set defines the vessel initial position, that is, the location of the vessel prior to the application of any vessel offset or dynamic motions. This location usually corresponds to the structure location in an initial static analysis. There are two inputs in this first set. The first is the location (coordinates) of the vessel reference point in the vessel initial position. The second input is the orientation of the vessel relative to the global coordinate axes in this initial position. These two sets of inputs completely define the vessel configuration prior to the application of any vessel offset. The definition of these parameters is illustrated schematically in the figure below. The reference point location is defined in terms of coordinates in the global coordinate system. The vessel orientation is defined in terms of the angle between the positive direction of the global Y-axis and the $\hat{y}$ direction of a local vessel coordinate system centred on the reference point. The significance of this local axis system is described in the next section.



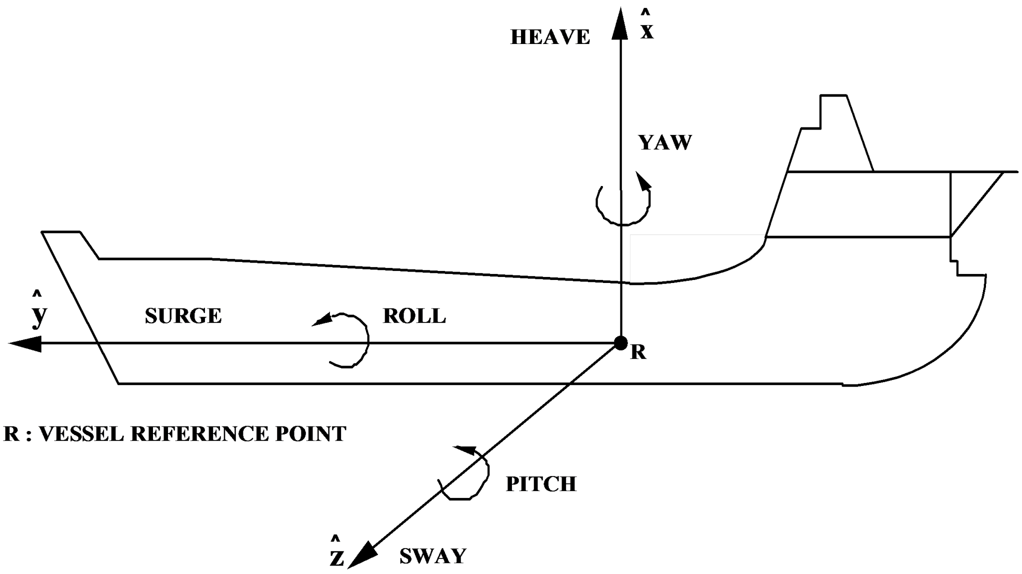
Initial Vessel Configuration

The next (second) set of inputs is used to specify the offset of the vessel from its initial position and orientation. These entries are optional and are only specified if the vessel is offset from its initial location and orientation before the commencement of the dynamic analysis. Flexcom offers a full 6 degree of freedom offset capability, comprising three translations and three rotations. Together, the vessel initial position and offset define the position and orientation of the vessel at the initiation of the dynamic analysis.

The third and final set of inputs is the full vessel RAOs and phase angles in all vessel degrees of freedom. Given the vessel initial position, offset and RAOs, in addition to the time history of wave elevation, the motion of the reference point in response to the wave field throughout an analysis can be calculated, and from this the motion of the point(s) on the structure attached to the vessel.

Vessel Degrees of Freedom

An attached vessel has six degrees of freedom (DOFs), three translations and three rotations. These are defined in Flexcom in terms of a local vessel coordinate axis centred, as noted above, on the reference point. The vessel direction is initially vertical. Motions in this direction represent vessel heave, and rotations about this axis correspond to vessel yaw. The local and directions form a plane that is horizontal in the initial vessel position. Motions in the vessel and directions correspond respectively to vessel surge and sway, while the respective rotations give roll and pitch. These directions and motions are illustrated schematically in the figure below. Note that vessel RAOs are normally defined for all six degrees of freedom.



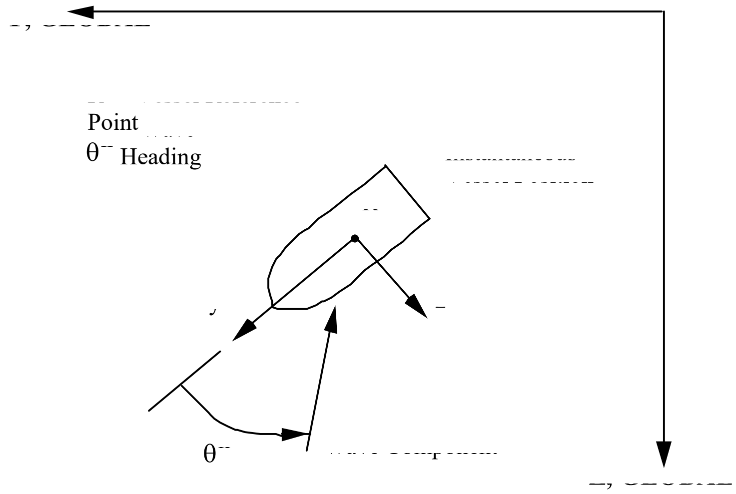
Vessel Degrees of Freedom

Vessel RAOs

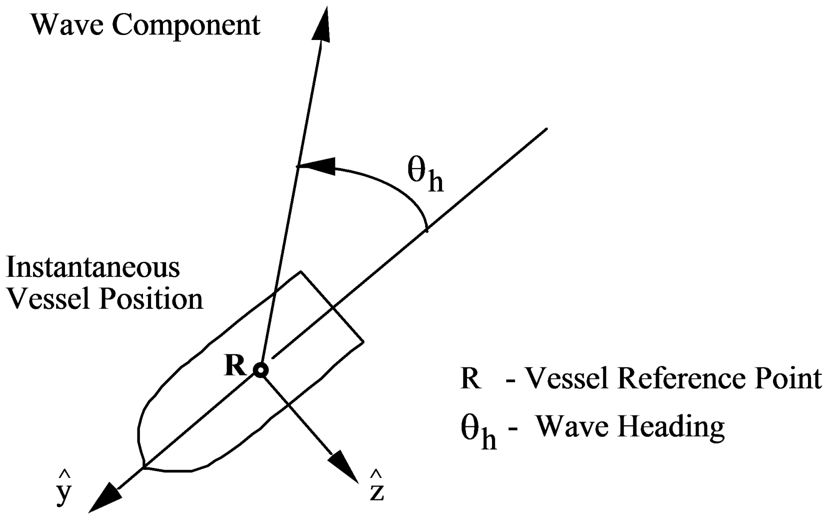
In the most general case vessel RAOs and phase angles vary with i) incident wave heading and ii) frequency. The definition of wave heading is described in detail shortly. The variation of RAOs and phase angles with frequency is well understood. This general variation of RAOs with wave heading and frequency is reflected in the format of the RAO file, where RAOs are defined first in terms of wave heading, and then at each heading value in terms of wave frequency. Of course, in very many analyses, namely those with one or more collinear regular waves or a uni-directional random sea, the question of wave heading does not arise, and this is reflected in Flexcom RAO file format.

Wave Heading

Wave heading is defined as the angle between the direction of approach of a wave harmonic incident on the vessel and the local surge () axis. The calculation of wave heading is illustrated schematically in the first figure below. This definition is one used principally by naval architects, but will be intuitively reasonable to most users. However, please note that in strict mathematical terms the incident wave heading is the angle, measured positive anticlockwise, from the negative direction of the local surge axis to the wave direction drawn at the vessel reference point. This is shown in the second figure below. This definition is slightly at odds with the scheme used elsewhere in Flexcom, where angles are defined in terms of wave and current directions relative to positive global Y-direction. This slight inconsistency results from the definition of wave heading in most common use and best understood by offshore engineers.



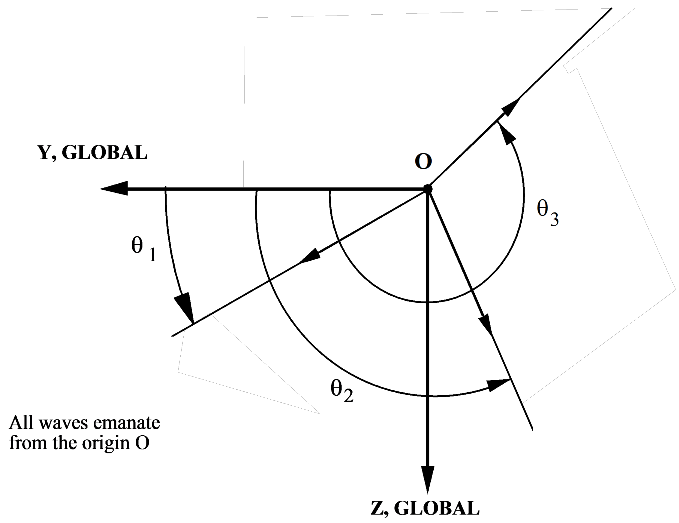
Definition of Wave Heading



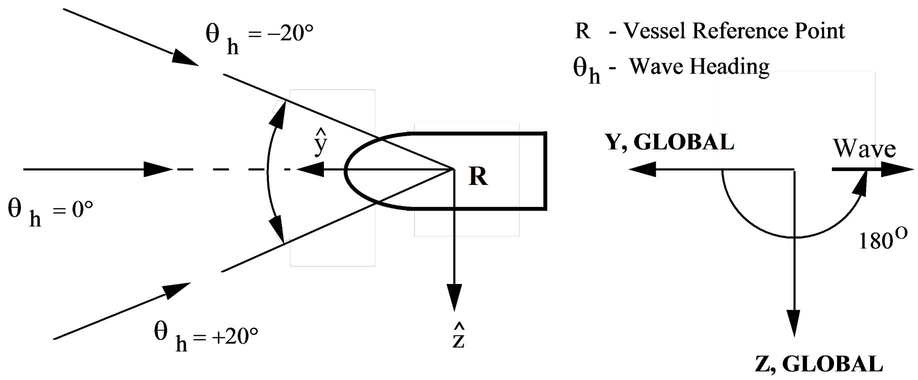
Definition of Wave Heading Used in Flexcom

Flexcom Waves

One very important point to note is that in Flexcom all waves emanate from the origin. A number of examples of wave direction specification are shown in the first figure below. When defining an RAO in terms of wave heading, it is important to ensure that the wave directions specified using any of the various wave specification options produce waves that are, in fact, incident on the vessel between the vessel heading values specified – see the second figure below.



Examples of Wave Direction Specification



Wave Direction and Wave Heading

It is important to be clear on the difference between wave direction and wave heading. Wave direction is the angle between the direction of wave advance and the global axis system; this remains constant throughout an analysis. You specify wave direction data when specifying other wave data, such as amplitude(s) and period(s). Wave direction data is independent of RAO data, because of course an analysis with waves included does not necessarily include vessel motions (although it would in most cases).

Wave heading on the other hand is a method of defining the direction of wave advance relative to a vessel axis system. As such, wave heading definition is intimately tied up with the specification and use of RAOs.

Miscellaneous

A number of final points to note with regard to the RAO data are as follows. Firstly, Flexcom uses linear interpolation to calculate RAOs and phase angles at wave headings and frequencies intermediate to those in the RAO file. Outside of the range of user-specified headings and frequencies, RAOs and phase angles are assumed to be zero, so it is important to ensure you cover the full range of conditions likely to be encountered in an analysis when inputting the RAO data. Secondly, specified phase angles represent a phase lag or lead relative to the wave at the vessel reference point. Thirdly, a positive phase angle denotes a phase lag relative to the incident wave harmonic. Finally, Flexcom requires that you use the same frequencies in specifying RAOs and phase angles at each vessel wave heading value. You do not however have to use the same frequencies (or indeed headings) if you are specifying RAOs for two vessels in an analysis – the RAOs can be at very different heading and frequency values, and in general would almost certainly be the case.

RAO Conversions

OVERVIEW

Flexcom, like most other programs, has its own set of conventions regarding the specification of input data. One area of particular importance is the definition of vessel RAO data. Quite often, RAO data is sourced externally (e.g. from a radiation-diffraction program), and the conventions used are rarely consistent with the standard format expected by Flexcom itself.

In order to minimise effort, Flexcom allows you to import RAO data directly from third-party software data files. If the data source corresponds to one of the standard programs listed below, it is not necessary to explicitly define a conversion scheme – you simply select the relevant program from a list provided. Flexcom understands the conventions used in these programs and automatically performs the relevant conversions for you. The programs currently supported are:

- WAMIT
- ANSYS Aqwa
- MOSES
- OrcaFlex

If you have RAO data which does not originate from any of the above sources, it is possible to define a custom RAO conversion to describe the various conventions pertaining to the RAO data.

RELEVANT KEYWORDS

- `*VESSEL,INTEGRATED` is used to specify all information pertaining to a vessel or vessels. Specifically, the `FORMAT=` input is used to specify the format of the RAO data, and you may select from a range of standard program formats such as WAMIT, AQWA, MOSES or OrcaFlex.
- `*VESSEL` is used to specify the initial position and undisplaced orientation of a vessel.
- `*RAO` is used to specify Response Amplitude Operators for a vessel. Specifically, the `FORMAT=` input is used to specify the format of the RAO data, and you may select from a range of standard program formats such as WAMIT, AQWA, MOSES or OrcaFlex.

Note also that the old `*VESSEL` and `*RAO` keywords have effectively been superseded by the new `*VESSEL,INTEGRATED` keyword, which accepts RAO data also, thereby eliminating the need for a separate `*RAO` keyword – hence the ‘integrated’ nature of the new keyword.

WAMIT

The conventions used in WAMIT are as follows:

- The definition of wave heading is transformed such that a 0 degree heading, which is incident on the stern in the WAMIT format, is incident on the bow in the Flexcom format.
- The order of translational RAOs is changed from WAMIT format (Surge, Sway, Heave) to Flexcom format (Heave, Surge, Sway).
- The order of rotational RAOs is changed from WAMIT format (Roll, Pitch, Yaw) to Flexcom format (Yaw, Roll, Pitch).
- The units of the rotational RAOs are changed from the WAMIT format of radians per unit length to the Flexcom format of degrees per unit length.
- The reference units of the rotational RAOs (i.e. whether the angle is specified per metre or per foot) may be adjusted if the WAMIT units differ from the Flexcom units. Typically the WAMIT units are ascertained automatically from the external file, but as this is not always possible, you are advised to explicitly state the unit system (along with the file name) in order to avoid any possible ambiguity.
- A positive phase angle in WAMIT format is converted from leading to lagging to conform to Flexcom format.

ANSYS Aqwa

The conventions used in ANSYS Aqwa are as follows:

- The definition of wave heading is transformed such that a 0 degree heading, which is incident on the stern in the AQWA format, is incident on the bow in the Flexcom format.
- The order of translational RAOs must be changed from AQWA format (Surge, Sway, Heave) to Flexcom format (Heave, Surge, Sway).
- The order of rotational RAOs must be changed from AQWA format (Roll, Pitch, Yaw) to Flexcom format (Yaw, Roll, Pitch).

- The reference units of the rotational RAOs (i.e. whether the angle is specified per metre or per foot) may be adjusted if the AQWA units differ from the Flexcom units. Typically the AQWA units are ascertained automatically from the external file, but as this is not always possible, you are advised to explicitly state the unit system (along with the file name) in order to avoid any possible ambiguity.

MOSES

THEORY

The conventions used in Moses are as follows:

- The definition of wave heading is transformed such that a 0 degree heading, which is incident on the stern in the MOSES format, is incident on the bow in the Flexcom format.

- Positive wave incidence is converted from MOSES format of clockwise in plan to Flexcom format of anti-clockwise in plan.

- The order of translational RAOs must be changed from MOSES format (Surge, Sway, Heave) to Flexcom format (Heave, Surge, Sway).

- The order of rotational RAOs must be changed from the MOSES format (Roll, Pitch, Yaw) to Flexcom format (Yaw, Roll, Pitch)

- Positive surge is converted from the MOSES format of Aft to the Flexcom format of Forward.

- Positive sway is converted from the MOSES format of Starboard to the Flexcom format of Port.

- Positive roll is converted from the MOSES format of Port Down to the Flexcom format of Port Up.

- Positive pitch is converted from the MOSES format of Bow Up to the Flexcom format of Bow Down.

- The reference units of the rotational RAOs (i.e. whether the angle is specified per metre or per foot) may be adjusted if the MOSES units differ from the Flexcom units. Typically the MOSES units are ascertained automatically from the external file, but as this is not always possible, you are advised to explicitly state the unit system (along with the file name) in order to avoid any possible ambiguity.

- The positive phase angle must be converted from the MOSES format of leading to the Flexcom format of lagging.

FURTHER INFORMATION

Further information may be found in the MOSES Conventions and Coordinates section of the Bentley Systems website.

OrcaFlex

OrcaFlex models can be saved in either binary data files (.dat) or text data files (.yml). Flexcom can import RAO data from an OrcaFlex text data file (.yml) only. The conventions used in Orcaflex are as follows:

- The definition of wave heading is transformed such that a 0 degree heading, which is incident on the stern in the OrcaFlex format, is incident on the bow in the Flexcom format.

- All other formats are set on a file by file basis and the interpretation of the available options in OrcaFlex are as follows:

- RAOResponseUnits: it is assumed that this refers to the units in which the rotational RAOs are referred to in the yml file. If they are stated as being in “radians” then Flexcom multiplies these rotational RAOs by (180/pi). If they are stated as being in “degrees” then no conversion will take place.

- RAOWaveUnit: It is assumed that this is always set to “amplitude” and if not Flexcom produces an error. No conversion will take place based on this convention but Flexcom will produce an error if either “max wave slope” or “wave steepness” is the stated convention.

- WavesReferredToBy: It is assumed that this option is a statement of what the units of wave period are for the RAOS in the yml file. Flexcom converts the values referenced by this convention from the convention stated in the yml file to the Flexcom format. If the waves are referred to in “periods” then Flexcom performs the following conversion (1/period). If the waves are referred to in “frequency (rads/s)” then Flexcom divides them by (2*pi). If the waves are referred to in “frequency (Hz)” then no conversion will take place.

- RAOPhaseConvention: It is assumed that this refers to the positive rao phase angle of the RAOs in the yml file. If the option in OrcaFlex states that the phase angle “leads” then Flexcom will multiply all the phase angles by -1. Otherwise if the option states that it “lags” Flexcom will not perform any conversion.

- RAOPhaseUnitsConvention: It is assumed this refers to the units that the phases in the yml file are in. If the phases are stated as being in “radians” then Flexcom multiplies the phases by (180/pi). If the phases are stated as being in “degrees” then Flexcom will not perform any conversion.

- RAOPhaseRelativeToConvention: It is assumed that this refers to the positive wave elevation. If the phases are stated as being relative to the “trough” then Flexcom subtracts 180 from all of the phases. If the phases are stated as being relative to the “crest” then Flexcom does not perform any conversion. If the phases are stated as being relative to either the “zero up-crossing” or “zero down-crossing” then Flexcom does not continue with the analysis and produces an error.

- SurgePositive: It is assumed that this refers to the positive surge direction. If the positive surge direction is stated as being “aft” then Flexcom converts the surge phase by subtracting 180 from it. If the positive surge direction is stated as being “forward” then Flexcom will not perform any conversion.

- SwayPositive: It is assumed that this refers to the positive sway direction. If the positive sway direction is stated as being “starboard” then Flexcom converts the sway phase by subtracting 180 from it. If the positive sway direction is stated as being “port” then Flexcom will not perform any conversion.

- HeavePositive: It is assumed that this refers to the positive heave direction. If the positive heave direction is stated as being “down” then Flexcom converts the heave phase by subtracting 180 from it. If the positive heave direction is stated as being “up” then Flexcom will not perform any conversion.

- RollPositiveStarboard: It is assumed that this refers to the positive roll direction. If the positive roll direction is stated as being starboard “up” then Flexcom subtracts 180 degrees from the roll phases. If the positive roll direction is stated as being starboard “down” then Flexcom will not perform any conversion.

- PitchPositiveBow: It is assumed that this refers to the positive pitch direction. If the positive pitch direction is stated as being bow “up” then Flexcom subtracts 180 degrees from the pitch phases. If the positive roll direction is stated as being bow “down” then Flexcom will not perform any conversion.

- YawPositiveBow: It is assumed that this refers to the positive yaw direction. If the positive yaw direction is stated as being bow to “starboard” then Flexcom subtracts 180 degrees from the pitch phases. If the positive yaw direction is stated as being bow to “port” then Flexcom will not perform any conversion.

- Symmetry: It is assumed that this refers to the planes in which the stated RAOs are set to be mirrored. If the mirroring option is stated as being in the “XZ plane” then Flexcom will mirror the RAOs about the surge axis. If the mirroring option is stated as being in the “YZ plane” then Flexcom will mirror the RAOs about the sway axis. If the mirroring option is stated as being in the “XZ and YZ plane” then Flexcom will mirror the RAOs about the surge and sway axes.

- The reference units of the rotational RAOs (i.e. whether the angle is specified per metre or per foot) may be adjusted if the OrcaFlex units differ from the Flexcom units. Typically the OrcaFlex units are ascertained automatically from the external file, but as this is not always possible, you are advised to explicitly state the unit system (along with the file name) in order to avoid any possible ambiguity.

Custom RAO Conversion

THEORY

If the RAO data requiring conversion is non-standard, then you are required to explicitly define an RAO conversion scheme using a series of drop-down lists. The inputs, which are largely self-explanatory, are divided into four main categories, as follows:

1. The Zero Degree Heading category allows you to define the conventions used with regard to wave heading. Specifically, you specify how the zero degree incident wave heading is oriented relative to the vessel, and whether or not the specified RAO values are to be mirrored around one or more of the vessel axes.

2. The Unit Convention category allows you to associate units with the RAO data. Specifically, you may specify units for the incident wave (frequency, period or circular frequency), incident wave headings (degrees or radians), phase angles (degrees or radians) and rotational RAO amplitudes (degrees or radians).

3. The Sign Convention category allows you to define various sign conventions used in the specification of the RAO data. Specifically, you may specify whether positive phase angles represent a phase lag or a phase lead, whether positive wave headings are measured clockwise or anti-clockwise from the zero degree heading, and whether a positive wave elevation is up or down.

4. The Direction Convention category allows you to define the sign conventions regarding the responses in each degree of freedom. For example, you can specify whether a positive heave represents vessel motion upwards or downwards, and so on.

RELEVANT KEYWORDS

- `*RAO FORMAT` is used to define custom RAO conversion settings.
- `*VESSEL,INTEGRATED` is used to specify all information pertaining to a vessel or vessels. Specifically, the `FORMAT=` input is used to specify the format of the RAO data.
- `*VESSEL` is used to specify the initial position and undisplaced orientation of a vessel.
- `*RAO` is used to specify Response Amplitude Operators for a vessel. Specifically, the `FORMAT=` input is used to specify the format of the RAO data.

Note also that the old `*VESSEL` and `*RAO` keywords have effectively been superseded by the new `*VESSEL,INTEGRATED` keyword, which accepts RAO data also, thereby eliminating the need for a separate `*RAO` keyword – hence the ‘integrated’ nature of the new keyword.

RAO Layouts

Flexcom supports two different schemes for laying out the data in the RAO file (assuming you are not importing RAO data directly from a third party software such as WAMIT®), namely the MCS layout and the Line layout. This former is the standard layout traditionally used by Flexcom. The Line layout is a more recent addition to the software, and it represents a more general layout that, for example, simplifies copying and pasting RAO data from spreadsheet programs.

Further information on this topic is contained in the following sections:

- Syntax
- MCS Layout
- Line Layout
- Miscellaneous

Syntax

RAO data comprises two types of lines, a data line and a comment line. The letter C (uppercase essential) in Column 1 identifies a comment line. Comment lines are completely ignored by Flexcom, and are intended to allow you to include comments in an RAO file for other users or to assist in later scrutiny. Comment lines can be included at any point in the RAO file.

Any line that is not a comment line is a data line. The actual data values expected on a data line are, of course, a function of position within the file. Numerical inputs on a data line are in free format, and can be specified as floating point numbers with or without exponent. Use an uppercase E when specifying an exponent. Either commas or blanks may separate a number of values on a particular data line; do not use TABs.

MCS Layout

The MCS layout begins with a single line defining the incident wave heading at which the RAOs are being defined. This is followed by a block of three lines specifying the incident wave frequency and the relevant RAO amplitudes and phases for this heading and wave frequency, as shown below. Note that entries shown below in italics should be replaced with their actual numeric values in the RAO file.

```
HEADING=Wave Heading
Wave Frequency
Heave RAO, Surge RAO, Sway RAO, Yaw RAO, Roll RAO, Pitch RAO
Heave Phase, Surge Phase, Sway Phase, Yaw Phase, Roll Phase, Pitch Phase
```

The block of three lines specifying the incident wave frequency, RAOs and phases are repeated as often as necessary until the RAOs are defined over the required range of wave frequencies. To define RAOs at more incident wave headings, you simply repeat the HEADING= line for the new wave heading, and the RAO data for this wave heading is specified as before. If RAOs are being defined for more than one incident wave heading, then they must also be defined for more than one frequency at each wave heading, and the number of frequencies at which the RAOs are defined is the same for each heading. If the RAOs are independent of the incident wave heading, then the HEADING= line should be omitted.

Line Layout

The Line layout is very similar to the standard MCS layout, with the exception that the RAO and phase data for a particular heading and frequency appear on a single line, the format of which is shown below. Note that, for clarity, the data shown below is split over a number of lines, but is in reality specified on a single line of the RAO file.

```
Wave Heading, Wave Frequency, Heave RAO, Heave Phase, Surge RAO, Surge Phase,
Sway RAO, Sway Phase, Yaw RAO, Yaw Phase, Roll RAO, Roll Phase, Pitch RAO,
Pitch Phase
```

This line is repeated for every wave frequency and every wave heading for which RAOs are being defined. As with the MCS layout, if RAOs are being defined at more than one incident wave heading, then they must also be defined at more than one frequency at each wave heading, and the number of frequencies at which the RAOs are defined is the same for each heading. So, for example, if RAOs were to be defined at three wave headings and ten frequencies per heading, then 30 lines of data would be required to specify the RAOs using the Line layout. If the same wave heading is specified on all lines, then Flexcom assumes that the RAOs are independent of wave heading. Note also that the order in which the lines of RAO data appear is not important. Flexcom automatically sorts the RAO data by heading and frequency.

Miscellaneous

Note that the vessel heading is the angle, in degrees, between the instantaneous local vessel surge axis and the line defining the direction of the wave incident on the vessel, as defined in Definition of Wave Heading Used In Flexcom figure. Vessel heading must be defined in the range ±180°.

The RAOs for the three displacements (heave, surge and sway) are input in terms of displacement amplitude per unit wave amplitude. For yaw, roll and pitch, the inputs are in degrees per unit wave amplitude.

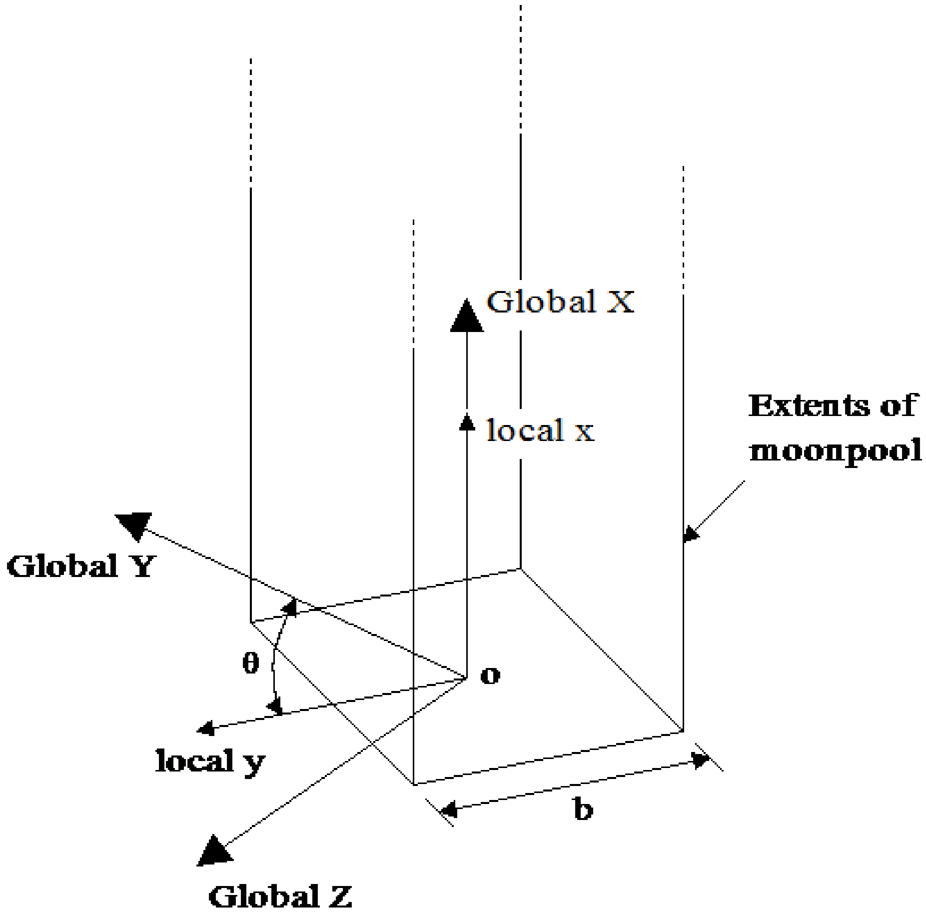
In general, when performing an analysis where RAOs are specified as a function of wave heading, it is always recommended when postprocessing the results that you plot timetraces of the motions of the node(s) attached to floating vessels. In this way you can ensure that the RAO specification did in fact lead to the expected vessel motions.

Moonpool Data

THEORY

Flexcom provides an option to specify that hydrodynamic loads on sections of riser contained within the vessel moonpool are based on the water particle velocities and accelerations within the moonpool. This facility is described in detail in Hydrodynamic Properties, and you are referred to the relevant section there for further details. In terms of vessel data, the location and extent of the vessel moonpool at solution initiation must be specified, for the purpose of applying hydrodynamic loading on structures contained within the vessel moonpool.

The vessel moonpool entrains a volume of seawater that is assumed to translate and rotate with the vessel in question. The water particle velocities and accelerations within the area enclosed by the moonpool are calculated from the vessel motions, as opposed to from the ambient wave field. The moonpool is assumed to be square in cross-section and extends from the origin of the moonpool (which would typically be at the same level as the vessel keel) to above the mean water level, as shown below in the figure below.



Definition of Moonpool Origin and Initial Yaw Orientation

The initial location of the volume enclosed by the moonpool is defined by specifying the global X, Y & Z coordinates of the moonpool origin, which is located at the centre of the square that forms the bottom face of the volume enclosed by the moonpool. The orientation of the volume enclosed by the moonpool in the global YZ-plane at solution initiation is defined by θ, which is the angle between the local moonpool y-axis and the global-Y axis (measured anticlockwise from the global-Y axis), as shown in the figure above. In most circumstances, this would be the same as the initial vessel yaw orientation. At solution initiation, the volume enclosed by the moonpool is assumed to be orientated vertically (that is, the local moonpool x-axis is aligned with the global X-axis, as shown in the figure above).

The moonpool width b also defines the overall height of a transition region between the volume enclosed completed by the moonpool and the ambient wave field. This transition zone, and how Flexcom treats riser elements within it, is illustrated and discussed in Hydrodynamic Properties, and you should referred to the relevant section there for further details if necessary.

RELEVANT KEYWORDS

- `*MOONPOOL` is used to define the location and extent of the vessel moonpool.
- `*HYDRODYNAMIC SETS` is used to assign hydrodynamic coefficients to element sets. Specifically, the `TYPE=MOONPOOL` input is used to assign hydrodynamic coefficients to elements of the structure that may be subjected to hydrodynamic loading within a vessel moonpool.

If you would like to see an example of how these keywords are used in practice, refer to A02 - Spar Production Riser.

1.9.2.9 Contact Surfaces

OVERVIEW

Guide surfaces are typically used to model the intermittent contact between a riser and another structure such as a vessel. Their use is illustrated in some of the standard examples which accompany the software.

- A02 - Spar Production Riser, which uses a series Zero-Gap Guide Surfaces to model hull-mounted guides.
- C02 - Multi-Line Flexible System, which uses an arrangement of Cylindrical Guide Surfaces to model a mid-water arch.
- H03 - Articulated Stinger, which uses a complex arrangement of Flat Guide Surfaces to model the rollerbox supports on an articulated S-Lay stinger.

There are two main steps involved in modelling contact with guide surfaces. The first of these is to define the location and geometry of the guide surfaces themselves. The second step is to define a set of elements which are to be monitored for contact with each guide.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Flat Guide Surfaces
- Cylindrical Guide Surface
- Friction Modelling
- Contact Modelling
- Zero-Gap Guide Surfaces
- Contact Element Sets
- Finite Element Mesh
- Contact Element Diameter
- Dynamic Analysis and Timestep Size
- Exclude Friction Option

RELEVANT KEYWORDS

- `*GUIDE` is used to define guide (contact) surfaces.
- `*CONTACT MODELLING` is used to specify specialised parameters relating to guide surface contact modelling.
- `*LINES` is used to define a line (and sections within that line), by specifying relevant set names, lengths, start and end locations, and mesh generation settings.
- `*TIME` is used to define time parameters for an analysis.
- `*NO FRICTION` is used to suppress the effects of friction.

Flat Guide Surfaces

THEORY

Each flat guide is a rectangular surface. To define a flat guide surface, it is necessary to specify the following pieces of information:

(i) Whether the guide surface is to be associated with a vessel or a structural node, or if it is to remain stationary throughout the analysis.

(ii) The name of a set of elements that are to be monitored for contact with the guide surface. In most analyses, only a small section of the model is ever likely to come into contact with a particular guide surface, and for this reason it is, in general, unnecessary to check every node in the model for contact with a guide surface at every timestep. By associating a set of elements with each guide surface, the number of nodes that must be checked at each solution time is significantly reduced. This leads to faster analyses.

(iii) The location of the flat guide surface at the start of the analysis. This is specified by inputting the global coordinates of a particular point on the flat guide surface at the start of the initial analysis.

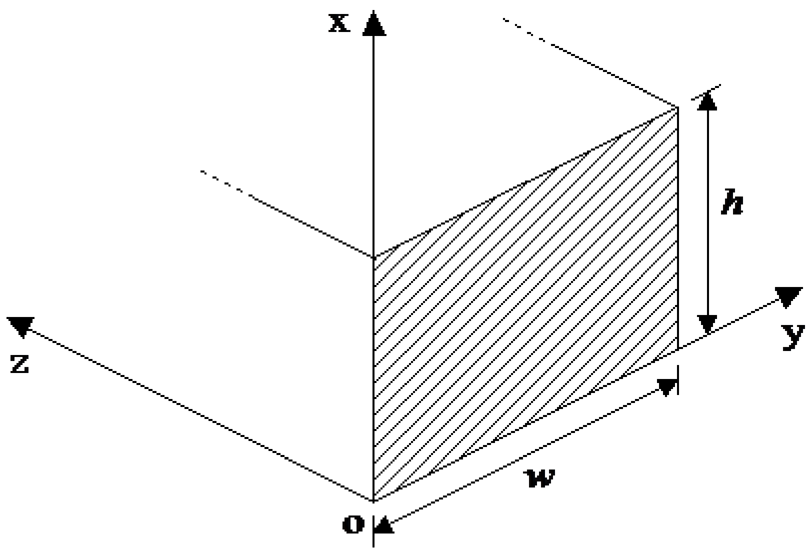
(iv) The orientation of the flat guide surface at the start of the analysis. This is defined by specifying the global components of two vectors that lie in the plane of the surface.

(v) The dimensions (height & width) of the flat guide surface.

(vi) The longitudinal and transverse friction coefficients associated with the contact surface.

(vii) The elastic contact stiffness of the guide surface.

Each contact surface has a local axis system associated with it, as shown in the figure below. The initial location and orientation of the contact surface is defined by specifying the initial coordinates of the origin of this axis system along with the local x- and y-axes.



Flat Guide Surface

The origin of the contact surface is defined as the lower left-hand corner of the surface when the surface is viewed from the side that can come into contact with the structure. To define the local x- and y-axes, you must specify the global X, Y & Z components of two vectors, the first of which is aligned with the local x-axis and the second of which is aligned with the local y-axis. The length of these vectors is not significant – it is their orientation that is important. Naturally, there must be a 90° angle between the local x- and y-axes or the program will generate an error. Once the local x- and y-axes are specified, the program automatically finds the local-z axis using the right-hand rule. Note that the orientation of the local-z axis is significant – it must point away from the side of the surface that can come into contact with the structure.

RELEVANT KEYWORDS

- `*GUIDE` is used to define guide (contact) surfaces. Specifically, the `TYPE=FLAT` inputs are used to define flat guide surfaces.

Cylindrical Guide Surfaces

THEORY

To define a cylindrical guide surface, it is necessary to specify the following pieces of information:

(i) Whether the cylindrical guide contact surface is external (the convex shape against the contacting body), or internal (the concave shape against the contacting body).

(i) The name of a set of elements that are to be monitored for contact with the guide surface. In most analyses, only a small section of the model is ever likely to come into contact with a particular guide surface, and for this reason it is, in general, unnecessary to check every node in the model for contact with a flat guide surface at every timestep. By associating a set of elements with each guide surface, the number of nodes that must be checked at each solution time is significantly reduced. This leads to faster analyses.

(ii) The location of the cylindrical guide surface at the start of the analysis. This is specified by inputting the coordinates of the origin of the contact surface at solution initiation. The origin is located at the centre of curvature, at the first end of the cylinder.

(iii) The orientation of the cylindrical guide surface at the start of the analysis. This is defined by specifying the components of axial and radial vectors.

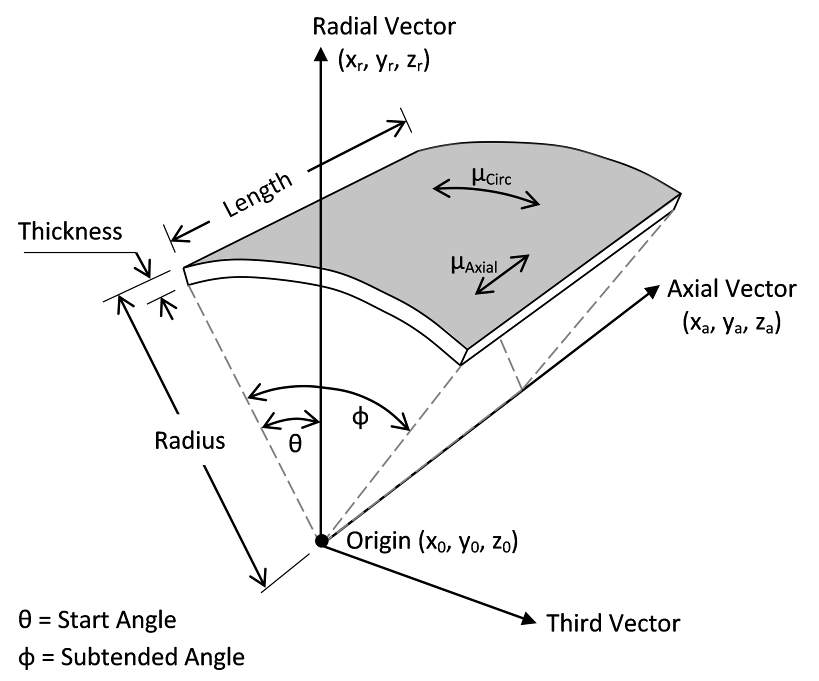
(iv) The dimensions (length, radius and subtended angle) of the cylindrical surface.

(v) The elastic contact stiffness of the guide surface.

(vi) The axial and circumferential friction coefficients associated with the contact surface.

(vii) Whether the cylindrical guide contact surface is external (the convex shape against the contacting body), or internal (the concave shape against the contacting body).

The overall layout of the cylindrical guide surface is illustrated in the figure below.



Cylindrical Guide Surface

Each cylindrical surface has a local axis system associated with it, as shown above. The axial and radial vectors form two components of a right handed system. The length of these vectors is not significant – it is their orientation that is important. Naturally these two vectors should be orthogonal. A third vector is assembled internally using the right-hand rule.

The sign convention for both the Start Angle and the Subtended Angle is consistent with the right handed system. Specifically, positive angles represent rotations of the radial vector toward the third vector. If a Start Angle is not specified, its magnitude is half that of the Subtended Angle, and it differs in sign. In this case, the radial vector bisects the Subtended Angle, and represents an axis of symmetry for the cylindrical surface.

The thickness of the contact surface is used for display purposes only and does not affect the overall operation of the contact modelling algorithm. It is an optional entry and defaults to 0.1m or 0.328ft. For external contact it is measured from the external surface in. For internal contact it is measured from the internal surface out.

RELEVANT KEYWORDS

- `*GUIDE` is used to define guide (contact) surfaces. Specifically, the `TYPE=CYLINDRICAL` inputs are used to define cylindrical guide surfaces.

Guide Surface Friction Modelling

THEORY

Friction on flat or cylindrical guide surfaces is modelled using a very similar approach to that adopted for seabed friction (refer to Seabed Friction Modelling Algorithm for a detailed discussion of the seabed friction model). Specifically, friction is effectively modelled using non-linear springs. In an ideal (that is, Coulomb) friction model, each of these non-linear springs would completely prevent motion until the total force exceeds the limiting friction force, at which point free motion is allowed subject to a resistance which is equal to the limiting friction force. The main difficulty with implementing such a friction model in a finite element program such as Flexcom, which solves for deflections, is that the stiffness of the ideal spring is effectively infinite in the region corresponding to zero deflection. This would make it virtually impossible for the program's iterative solution scheme to converge on the correct solution. Instead, this spring characteristic has the region around the zero-deflection point replaced by a section of very high (but not infinite) stiffness. The stiffness of this section of the force-deflection curve is given by the expression:

$$k = \frac{\mu R}{0.05 L_c} \qquad (1)$$

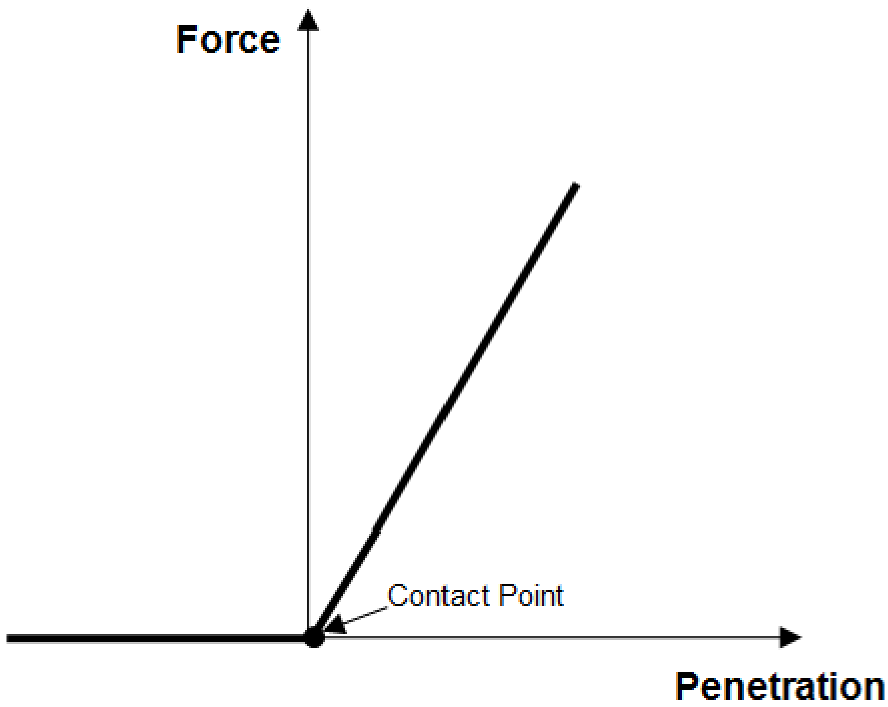
Here L_c is what is known as the 'characteristic length'. Clearly, the value of L_c affects the stiffness of the non-linear spring – the smaller this value, the greater is the stiffness and the closer the friction model is to an ideal model. However, reducing L_c makes it harder for the program to converge on a solution.

The guide surface friction model differs from the seabed friction in terms of characteristic length selection. In the seabed case, the value of L_c is based on the element length, as this gives a degree of scalability to the model. You can also specify maximum characteristic lengths in both the longitudinal and transverse directions (both of which default to 3.047m/10ft if unspecified). Guides are typically used to model contact over relatively short contact lengths – a typical application might be that of a Spar, where intermittent contact is modelled between risers and sections of the Spar hull. As the surfaces are typically short, the characteristic length is unrelated to the element lengths, and simply defaults to 10% of the guide 'size', though you do have the option to override this. This 'size' is the flat surface height or cylindrical surface length. Note also that the same characteristic length is used in both the longitudinal and transverse directions.

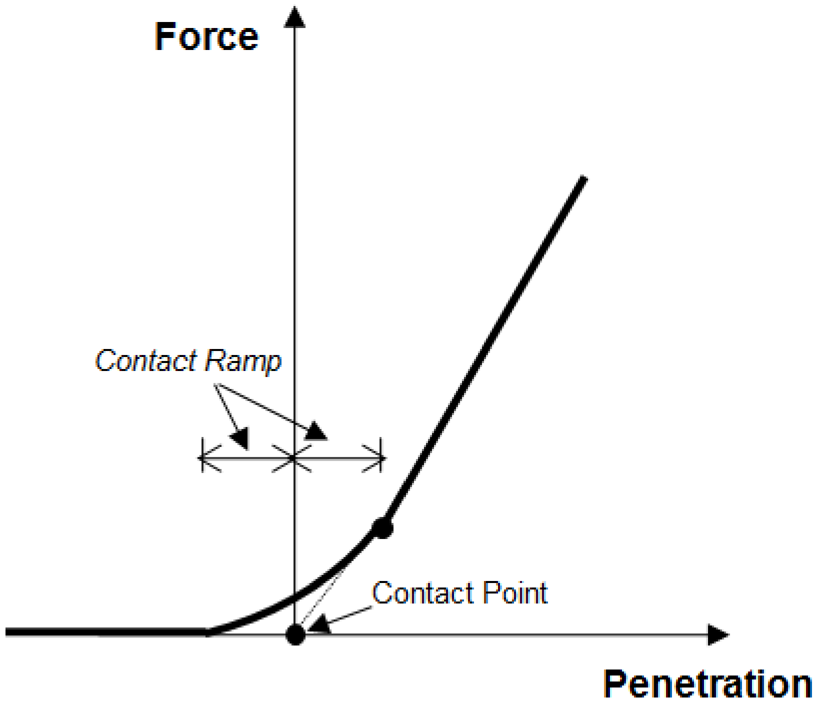
Contact Modelling

THEORY

An important aspect of the contact algorithm is how Flexcom handles the interaction between the structure and the guide surfaces. Each node that is potentially in contact with a guide surface may be considered to be connected to the surface by means of a non-linear spring which has an orientation normal to the surface. Theoretically, each of these springs should have the force-deflection characteristics shown in the first figure below, whereby the force in the spring is zero until the node makes contact with the surface, at which point the linear spring stiffness is activated. While this formulation is theoretically correct, it is not especially robust, so Flexcom can use a modified spring force-deflection curve shown in the second figure below instead. Adopting a non-linear approach tends to lessen the effects of instantaneous impact, reduce high frequency noise, and consequently allows the solution to proceed at larger time step increments.



Theoretical Contact Model



Non-Linear Contact Model

In the second figure above, rather than the contact spring stiffness being activated suddenly when contact occurs, the force in the spring is gradually ‘ramped up’. This means that there is some force in the spring before a node penetrates the guide surface, but the magnitude is relatively small. Overall, the contact algorithm provides a robust and accurate engineering approximation. The distance over which the force in the spring is ramped up is symmetric about the Force axis. In other words, the distance from the start of the ramp to the Force axis is the same as the distance from the Force axis to where the ramp meets the linear surface stiffness curve. In Flexcom, these equal distances define an input called the Contact Ramp. This input assumes a value of zero by default, meaning that the spring force-deflection characteristics are as shown in the first figure above. It is difficult to provide meaningful guidance as to what Contact Ramp might represent a “reasonable” value, but a distance of 0.1m (or equivalent in Imperial units) is tentatively suggested as a starting point if you are unsure.

It is also worth noting that the curve used to gradually ramp up the force in the contact spring is governed by a power law equation and the associated ‘power’ value can be specified through the Exponent input in the contact modelling settings. The default value for this input is 1.5, which probably represents an optimum value. However for high stiffness supports you may wish to reduce the Contact Power to a smaller value (e.g. 1.0). Reducing the power has the effect of reducing curvature in the ramp region.

IDENTIFYING CONTACT NODES

Switching on node numbers in the Model View will allow you to identify Contact Nodes.

RELEVANT KEYWORDS

- `*CONTACT MODELLING` is used to specify specialised parameters relating to guide surface contact modelling.

Zero-Gap Guide Surfaces

THEORY

Each zero-gap guide is an enclosed surface, typically associated with a vessel, so that the guide translates and rotates with the vessel during the course of an analysis. To define a zero-gap guide, it is necessary to specify the following pieces of information at the initial analysis stage:

(i) Whether the zero-gap guide is to be associated with a vessel, or if it is to remain stationary throughout the analysis.

(ii) The name of a set of elements that are to be monitored for contact with the zero-gap guide.

(iii) The location of the zero-gap guide at the start of the analysis. This is specified by inputting the global coordinates of a particular point on the zero-gap guide at the start of the initial analysis.

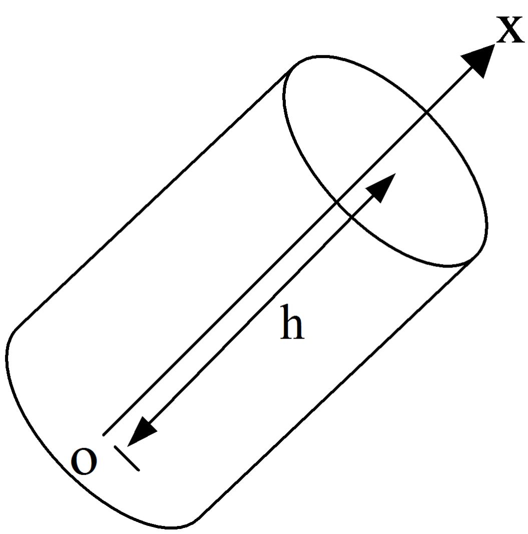
(iv) The orientation of the zero-gap guide at the start of the analysis. In this case this is done by specifying the global components of a vector which is aligned with the local longitudinal axis of the zero-gap guide.

(v) The length of the zero-gap guide.

(vi) The longitudinal friction coefficient that applies to contact between the zero-gap guide and any of the elements in the specified contact element set.

For illustrative purposes, Flexcom automatically generates auxiliary elements at the location of each zero-gap guide, so that guides may be easily identified when models are viewed.

The orientation is input in terms of a single vector defining the local longitudinal axis (see the figure below). The initial location and orientation of the guide is defined by specifying the initial coordinates of the origin of the guide along with this orientation vector, shown as x in the figure below.



Zero-Gap Guide

The origin of the zero-gap guide is defined as the centre point of one end (usually bottom) of the guide. To define the orientation vector, you must specify its global (X, Y & Z) components. The length of this vector is not significant – it is its orientation that is important.

A static preload may be optionally applied to a zero-gap guide. If a preload is specified, the limiting frictional force, F_{lim}, in the longitudinal direction at a given node in contact with the zero-gap guide is defined as:

$$F_{\lim} = \mu(R_n + P_n) \qquad (1)$$

where μ is the longitudinal friction coefficient, R_n is the magnitude of the reaction force at the constrained node, and P_n is the preload associated with the contact node. The user specified preload, P, for the guide is distributed evenly between all nodes which are in contact with the guide at any given time, such that:

$$P_n = \frac{P}{N} \qquad (2)$$

where N is the number of nodes in contact with the guide. If no static preload is specified, the limiting frictional force is computed in the usual way as:

$$F_{\lim} = \mu R_n \qquad (3)$$

RELEVANT KEYWORDS

- `*GUIDE` is used to define guide (contact) surfaces. Specifically, the `TYPE=ZEROGAP` inputs are used to define zero-gap guides.

Contact Element Sets

THEORY

As noted above, an important aspect of guide surface contact modelling is the definition of element sets that are to be monitored for contact. Care should be taken when defining contact element sets that all elements that may come into contact with the guide surface are included in the appropriate set. This is particularly important in situations where, for example, large vessel heave motions mean that relatively large sections of riser may come into contact with a given guide surface.

RELEVANT KEYWORDS

- `*GUIDE` is used to define guide (contact) surfaces.

Finite Element Mesh

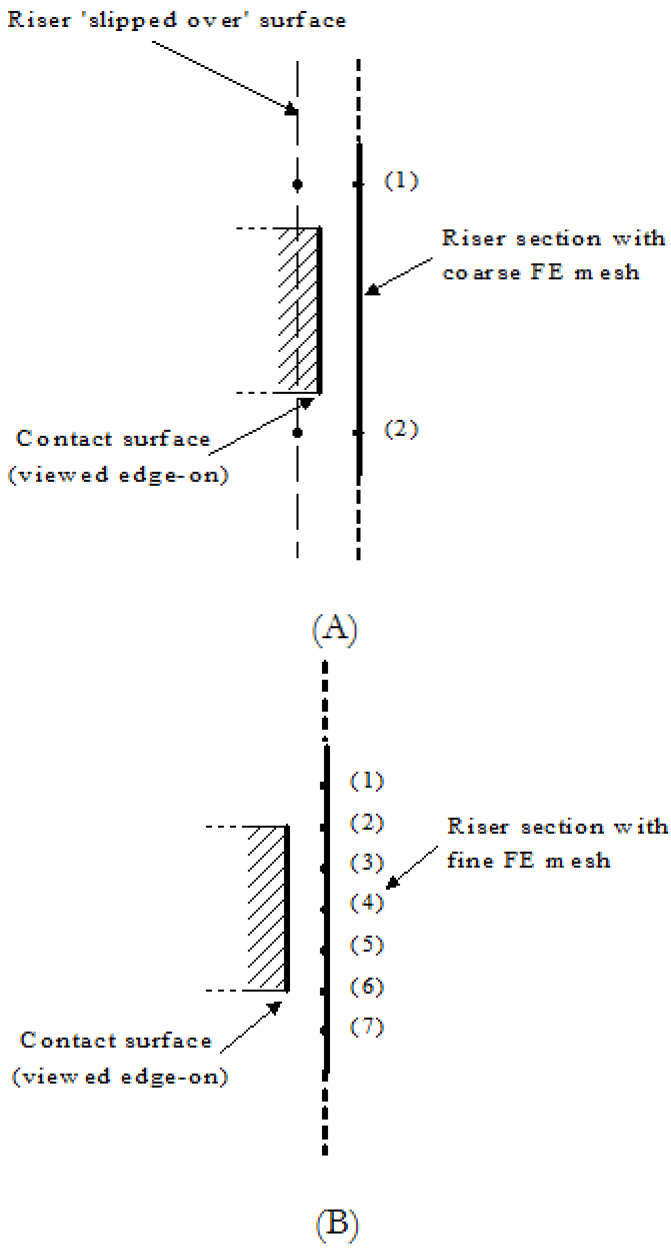
THEORY

The contact algorithm (for both flat and cylindrical guide surfaces) works by checking the position of nodes at each solution step for contact with the guide surfaces. If a node comes into contact with a guide surface, an appropriate stiffness contribution is added to the global stiffness matrix (similar to the insertion of a spring) in a direction normal to the guide surface, thereby resisting penetration of the guide.

A zero-gap guide may be thought of as a sheath positioned around a section of riser, with a contact clearance of zero between the guide and the riser. The zero-gap contact algorithm differs from the corresponding flat/cylindrical guide procedure. If a node comes into contact with a zero-gap guide, appropriate boundary conditions are applied at the node in local axes which are both perpendicular to the longitudinal direction. The node is free to move in the axial direction (subject to frictional restraints) through the zero-gap guide, and whenever the node leaves the guide, the boundary conditions are removed.

Because the contact algorithm is based on monitoring the position of nodes, it is important that the finite element mesh in sections of the riser that may come into contact with guide surfaces is sufficiently fine to properly model contact. For example, the figure below (A) shows a section of riser adjacent to a flat guide contact surface (which is being viewed edge-on and is represented by the heavy black line). Clearly, the finite element mesh in the riser is not sufficiently fine to model contact – with this model, the riser could 'slip-over' the guide surface as illustrated. The figure below (B) shows a more suitable mesh – it is obvious that, in this case, contact between the riser and the guide surface will be modelled correctly.

To properly model contact between a riser and a guide surface, it is desirable that the finite element mesh be such that at least two nodes of the riser are in contact with the guide surface when contact occurs. So the element length in the region of contact should be no more than half the appropriate dimension of the guide surface. This requirement for a fine finite element mesh in the region of contact must also be balanced against the requirement to keep the overall number of elements in the model to a practical level. For this reason, an element length in the range of one half to one third the appropriate dimension of the guide surface appears optimal.



Finite Element Meshes for Surface Contact

RELEVANT KEYWORDS

- *GUIDE is used to define guide (contact) surfaces.
- *LINES is used to define a line (and sections within that line), by specifying relevant set names, lengths, start and end locations, and mesh generation settings.

Contact Element Diameter

THEORY

As described above, you must associate a contact element set with each guide surface that you define. Flexcom then checks each node on every element of the appropriate contact element set for contact with the guide surface. When carrying out this check for flat/cylindrical guide surfaces, the program automatically takes account of the diameter of the structure at the node. For this purpose, you have the option of specifying a contact diameter for each element. The contact diameter at a node is defined as the largest contact diameter of any of the elements connected at that node. In this way, contact between a node on the structure and a guide surface always occurs at the external surface of the structure at the node.

RELEVANT KEYWORDS

- `*GEOMETRIC SETS` is used to assign geometric properties to element sets, including contact diameter.

Dynamic Analyses and Timestep Size

THEORY

By its nature, intermittent contact between risers and flat/cylindrical guide surfaces involves modelling impact of the riser on the guide surface. Impact situations tend to cause structure response at high frequencies. This problem is exacerbated by the fact that, in general, the element length around the area of contact tends to be small, which can lead to increased structure response at these high frequencies. This difficulty may be alleviated in a number of ways:

- A small timestep may be required to successfully model the riser/guide surface impact. This should be borne in mind when selecting the timestep algorithm in dynamic analyses. If you are using a fixed timestep, then it should be sufficiently small to model the riser/guide impact. Similarly, if you are using a variable timestep, then the minimum permitted timestep should also be sufficiently small to model the riser/guide surface impact. For most applications, the variable timestep is recommended, as this allows the program to automatically select a small timestep when required to model impact.

- Even if you wish to model an effectively rigid surface, caution is advised against the specification of excessively large contact stiffness values. Using very high values tends to increase high frequency noise, resulting in the requirement of increasingly small time steps, and computationally intensive analyses.

- Specify a non-zero Contact Ramp, as described in Contact Modelling. Adopting a non-linear approach tends to lessen the effects of instantaneous impact, reduce high frequency noise, and consequently allows the solution to proceed at larger time step increments.

- Zero-gap guides may be used as an alternative to flat guide surfaces – particularly in situations where riser/guide contact is expected to be constant rather than intermittent. Unlike flat guide surfaces, constraints are not removed until sufficient relative motion in the axial direction has occurred. Models which use the zero-gap guide contact algorithm tends to be more dynamically stable as there is little or no high frequency noise introduced by the riser/guide contact.

RELEVANT KEYWORDS

- `*TIME` is used to define time parameters for an analysis.

Exclude Friction Option

THEORY

An option is provided to suppress the effects of friction. Very occasionally, it may be desirable to exclude guide friction, for example, during the initiation of stinger contact in a pipeline installation analysis. By default frictional effects are modelled (provided at least one non-zero friction coefficient is specified), and this is suitable for the majority of analyses.

RELEVANT KEYWORDS

- `*NO FRICTION` is used to suppress the effects of friction.

1.9.2.10 Line Clashing

OVERVIEW

Flexcom incorporates a facility for modelling contact between adjacent lines. Typical applications would include the interference that occurs between flexible risers and vessel mooring lines on a floating production system, or between rigid drilling and production risers on a TLP. You create models of the various structures in the normal fashion, and then indicate regions on both structures over which interference is thought likely to occur. The contact algorithm itself is relatively sophisticated – the following aspects in particular are noteworthy:

- The finite element mesh in the contact region need not be particularly refined. Contact is predicted based on the shortest distance between potential contact elements, regardless of the actual element lengths used. This allows contact regions to be initially identified using a relatively coarse model, while the structural response may be more accurately modelled after subsequent mesh refinement.
- Clashing is a complex and highly non-linear phenomenon, so the use of relatively small time steps is essential in order to accurately capture the impact and subsequent structural response. Flexcom continually monitors the relative velocity of approaching lines, allowing the time step to be gradually reduced in anticipation of contact. Once the lines have separated after impact, the time step begins to increase again. This approach facilitates a robust and accurate contact model, while also ensuring an efficient simulation overall.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Finite Element Mesh
- Contact Regions
- Contact Stiffness and Damping
- Timestep Size
- Contact Ramp
- Axis System

- Solution Outputs

RELEVANT KEYWORDS

- *CLASHING is used to specify regions where clashing may occur, and suitable contact stiffness and damping values.
- *CLASHING SOLUTION is used to specify solution parameters associated with clashing.
- *LINES is used to define a line (and sections within that line), by specifying relevant set names, lengths, start and end locations, and mesh generation settings.
- *TIME is used to define time parameters for an analysis.

If you would like to see an example of how these keywords are used in practice, refer to G01 - Jumper Clashing or J01 - Dropped Object.

Finite Element Mesh

THEORY

The finite element mesh in the contact region need not be particularly refined. The contact algorithm is sufficiently intelligent to predict contact, based on the shortest distance between potential contact elements, and this is effectively independent of the element lengths used in the contact region. However, in order to accurately capture the structural response, it is recommended that relatively short elements are used in the contact region. A suggested modelling approach would be to use a relatively coarse model initially to facilitate identification of contact areas, and then to subsequently refine the finite element mesh in appropriate areas to provide a more precise contact model.

When two adjacent lines are in close proximity, the minimum clearance is computed using some straightforward geometrical relationships, assuming that curvature is not a significant factor, and the elements can be treated as straight line segments. When the lines come into contact, the exact point of contact is identified in terms of distances along each element. Naturally, it is very unlikely that the contact point will coincide exactly with any node of the finite element discretisation, so Flexcom distributes the various finite element terms between the nodes nearest the contact point. Consequently, the contact model should provide a more accurate representation if shorter elements are used.

RELEVANT KEYWORDS

- `*LINES` is used to define a line (and sections within that line), by specifying relevant set names, lengths, start and end locations, and mesh generation settings.

Contact Stiffness and Damping

THEORY

A clashing connection can be considered to operate in a similar fashion to a spring element and (optionally) a damper element in parallel. Specifically, the global stiffness matrix is augmented at appropriate locations by the contact stiffness that you specify, and this tends to oppose the relative penetration of the lines. Once the lines have separated, any stiffness contributions are removed. Similarly, the global load array is augmented at appropriate locations by a damping force term, which is proportional to the relative velocity of the approaching lines. As soon as the lines begin to separate, as the relative velocity changes direction, any damping force terms are switched off.

It may be difficult to quantify what value of contact stiffness represents the physical reality. Generally speaking, the higher the stiffness value chosen, the greater the impact will be, and the smaller the time step that will be required for a robust solution. So you should avoid using excessively high stiffness values, which may lead to convergence difficulties. Conversely, the stiffness value effectively determines the maximum clashing force that may be modelled – the maximum reaction force is equal to the contact stiffness times the average contact radius of the elements which interact with each other. In other words, the maximum relative penetration of the contact elements should not be allowed to exceed the average contact radius. If this occurs, the lines will simply pass though each other. Although in practice this situation would be quite rare, to prevent the situation from occurring, you should ensure that the contact stiffness is sufficiently high. While it is difficult to provide meaningful guidance as to what value might represent a “reasonable” level for all cases, a value of approximately 100 kN/m (or equivalent in Imperial units) is tentatively suggested as a starting point if you are unsure.

Clashing is generally suitable for modelling intermittent contact between adjacent lines. If two lines come into contact, and tend to remain in contact, the Pipe-on-Pipe feature may be more suitable. There is a theoretical limit on the maximum contact force, which is equal to the contact stiffness multiplied by the average contact radius of the contacting elements. Situations where significant contact forces are maintained, for example during a static analysis subject to Current Loading, are better suited to the pipe-on-pipe modelling facility.

Specification of damping is optional, but some level of damping is generally advisable in dynamic analyses to help dissipate high frequency noise.

RELEVANT KEYWORDS

- `*CLASHING` is used to specify regions where clashing may occur, and suitable contact stiffness and damping values.

Timestep Size

THEORY

Clashing is a complex and highly non-linear phenomenon, so it is imperative that you use a temporal discretisation which is sufficiently refined to accurately capture the occurrence of clashing and the subsequent structural response. It is generally advisable to use a variable time step in analyses that include clashing. This allows relatively small time steps to accurately capture the contact and subsequent structural response, while allowing relatively large time steps to be used during periods where no interaction occurs. Various options relating to time stepping are provided (as described in the following paragraphs), but these inputs are meaningless if your analysis uses a fixed time step. Note also that because relative velocities are required as part of the time step optimisation, these are estimated from displacements if you are running a static analysis.

The Maximum Time Step input limits the time step used during clashing. Naturally this must be somewhere between the (overall) maximum and minimum analysis time steps specified for the analysis, and a relatively small value (for example 0.01s) would generally be recommended in order to accurately capture the relative impact between two adjacent lines. Flexcom continually monitors the relative velocity of approaching lines with a view to estimating a time of impact. The time step is gradually reduced in anticipation of contact, remains at a relatively small value while the lines are in contact, and only increases when some finite separation has developed after the lines have separated. The magnitude of this separation is dictated by the Threshold Clearance input. The Threshold Clearance is specified in terms of a percentage of the average contact diameter of the elements which come into contact.

In certain circumstances, it may not be desirable to persist with a very small time step while clashing is taking place, particularly if the interaction between two lines is likely to be constant rather than intermittent. A good example might be a static current analysis where a cross-current loading would tend to induce (and maintain) contact between two adjacent lines. A recommended procedure for performing this type of analysis would be to use a variable time step (whether the analysis is performed statically or quasi-statically), which will allow Flexcom to monitor the relative velocities and reduce the time step in advance of contact. If the lines tend to remain in contact after the initial impact, the time step will continue to be very fine (resulting in a very inefficient analysis). In such situations, it would be advisable to use the Successive Solutions option, allowing the time step to be increased after contact has been firmly established (up to the maximum analysis time step).

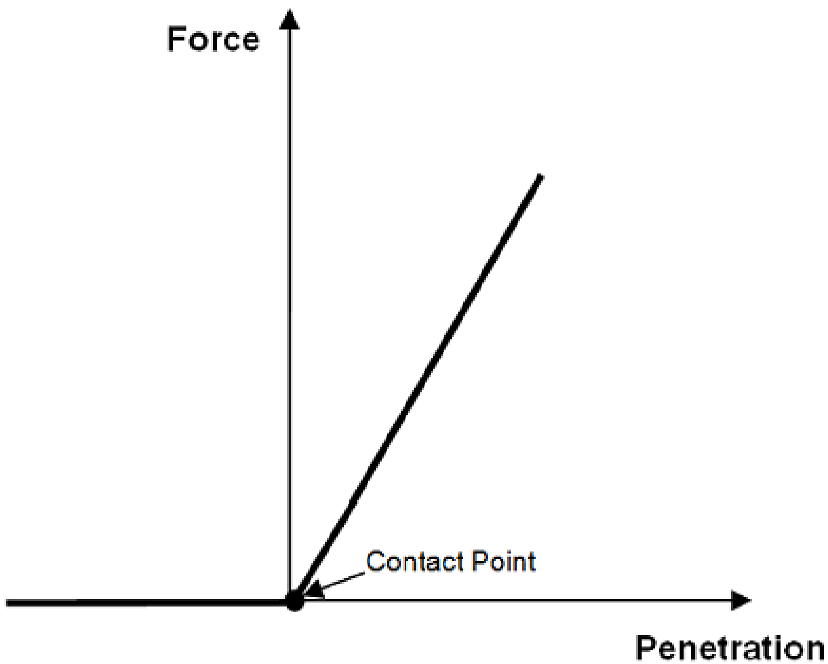
RELEVANT KEYWORDS

- `*TIME` is used to define time parameters for an analysis.
- `*CLASHING SOLUTION` is used to specify solution parameters associated with clashing.

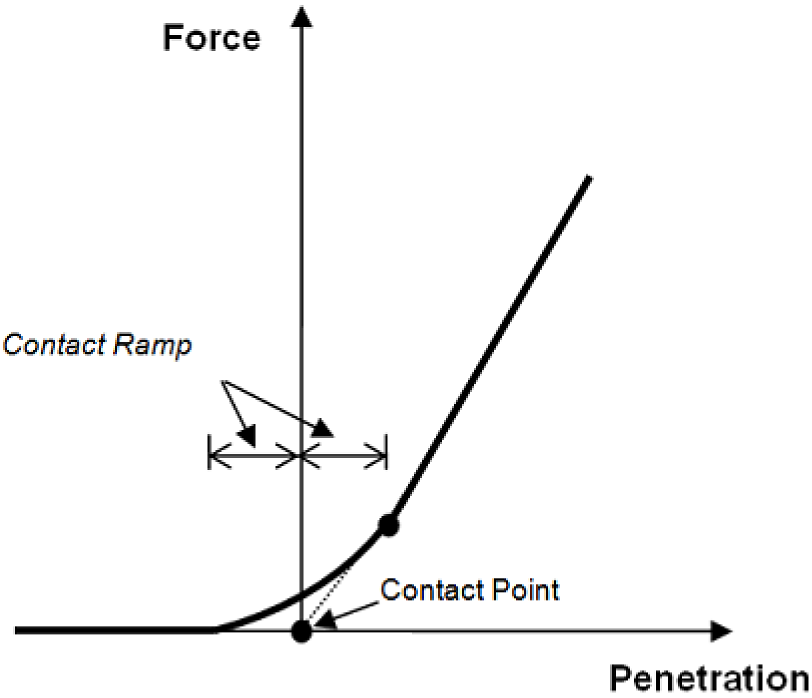
Contact Ramp

THEORY

When two lines come into close proximity, the interaction between the two lines is effectively modelled by the insertion of a non-linear spring between the points at which minimum separation occurs. Theoretically, the spring should have a force-deflection (or penetration) characteristic similar to that shown in the first figure below, whereby the force in the spring is zero until the lines come into contact, at which point a relatively large linear spring stiffness is activated. While this formulation is theoretically correct, you may wish to specify that a contact ramp is to be applied, in which case a force-deflection characteristic similar to that shown in the second figure below is used instead. So rather than the contact stiffness being activated suddenly when contact occurs, the contact force is gradually ‘ramped up’. This means that there is some contact force present before contact actually occurs, but you may feel that the modelling inaccuracy is outweighed by increased solution robustness (for example, you may be able to use larger time steps).



Theoretical Force-Deflection Characteristic



Force-Deflection Characteristic with Contact Ramp

RELEVANT KEYWORDS

- *CLASHING SOLUTION is used to specify solution parameters associated with clashing.

Contact Regions

THEORY

Monitoring of clashing introduces a significant computational overhead, as all elements in the first set are checked for contact with all elements in the second set, for all the clashing regions that you have defined. When defining contact regions, you should be careful that you specify reasonable data to prevent excessive runtimes, by ensuring that the program is not checking for contact between points that will never approach. It is generally recommended that similar spatial discretisations be used for any element sets which may come into contact with each other. The choice of which element set is designated as first or second should be irrelevant – it is merely a convenience for the internal operation of the contact modelling algorithm.

You can optionally specify that active contact elements are be echoed to the output file. If you invoke this option, the output file shows you exactly what elements contact each other at every solution time. This feature is particularly useful as it allows you to refine the contact set element definitions and consequently reduce run times.

RELEVANT KEYWORDS

- *CLASHING is used to specify regions where clashing may occur, and suitable contact stiffness and damping values.

Axis System

THEORY

By default, the axis system for application of contact forces between a pair of contact elements is based on the instantaneous positions of the elements at any given time. In some rare circumstances, you may wish to specify that the line of action of the contact forces remains constant, and be based on the initial positions of the contact elements. An example might be a static analysis for which solution convergence occurs, as an equilibrium configuration is difficult to attain due to the nature of the contact interaction. However, the constant axis system option should rarely be invoked.

RELEVANT KEYWORDS

- *CLASHING SOLUTION is used to specify solution parameters associated with clashing.

Solution Outputs

THEORY

Options are provided to request time history plots of various parameters relating to clashing analyses. These are as follows:

(i) The minimum clearance between element sets for which a clashing region is defined. A clearance value of zero indicates that contact has occurred. A value of less than zero indicates that some degree of relative penetration has occurred between the lines.

(ii) The total reaction force for a clashing region. If there is more than one clashing location the various reaction magnitudes are simply summed together. The reaction at any particular location is based on the contact stiffness times the relative penetration of the lines, plus any damping contribution (proportional to the relative velocity of approaching lines).

(iii)The clashing impulse. This is the integral of the reaction force with respect to time.

RELEVANT KEYWORDS

- `*TIMETRACE` is used to request the creation of timetrace plots. Specifically, the `TYPE=CLASH` inputs are used to request plots of the time history of clearance, reaction or impulse for clashing analyses.

1.9.3 Applied Loading

This section contains information on the following topics.

- Internal Fluid discusses the various aspects of the internal fluid modelling capability.
- Current Loading describes the various program options for modelling current loading.
- Wave Loading discusses the various wave specification options in Flexcom. There are three regular wave options (Regular Airy, Stokes V and Dean's Stream) and five options for defining a random seastate in terms of a wave spectrum (Pierson-Moskowitz, Jonswap, Ochi-Hubble, Torsethaugen and User-defined).
- Additional Loading Options presents some other useful options for applying loads to a model, including Point and Distributed Loads, User-Defined Forces and Temperature Loading.
- Boundary Conditions discusses the range of constraints which may be applied to the finite element model, the most widely used being Constant Boundary Conditions and Vessel Boundary Conditions. Application of Rotational Constraints discusses the significance of rotational degrees of freedom in terms of boundary conditions.
- Wake Interference describes the comprehensive wave interference modelling facility provided by Flexcom. Three wake formulations are available, namely Huses Model, Blevins Model and a User-Defined Wake Model for complete generality.

- Coupled Analysis discusses the Flexcom's coupled analysis capabilities, including the powerful Floating Body feature.

- User Solver Variables describes a powerful feature which provides for complete generality in terms of load application. It is even possible to directly modify the constitutive finite element matrices (i.e. stiffness and mass) should you have very specialised modelling requirements.

- Wind Turbine Modelling discusses the wind turbine modelling feature in Flexcom.

1.9.3.1 Internal Fluid

OVERVIEW

Flexcom provides a comprehensive internal fluid modelling capability. Stationary internal fluids, uniform steady state internal fluid flow and multi-phase slug flow may all be modelled. If slugs are present in an analysis, they are displayed visually in the structural animation.

A stationary internal fluid in Flexcom is characterised by its mass density, pressure above hydrostatic and the height above zero datum to which it extends. The height input is used to determine whether elements are fully filled with fluid, partially filled or empty.

A fluid velocity may be optionally specified, indicating that steady state internal fluid flow is being modelled. Steady state internal fluid flow induces both a centrifugal force (related to pipe curvature) and a Coriolis force (related to pipe rotation). An additional dynamic pressure term which affects pipe wall tension is also modelled. Options are provided to suppress any of these terms in a particular analysis, allowing the relative significance of each term to be assessed.

Note that where a moving internal fluid is being modelled, the direction of flow is assumed to be consistent with the definition of the element set in which the fluid is present – i.e. the fluid flows from the start element of the set towards the end element. If you wish to reverse the direction of fluid flow, you should reorder the relevant element set definition, or specify a negative value for fluid velocity.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Gravitational and Inertial Forces

- Centrifugal Forces
- Coriolis Force
- Hydrostatic Pressure
- Dynamic Pressure
- Slug Flow
- Inertial Effects

RELEVANT KEYWORDS

- `*INTERNAL FLUID` is used to define the properties of an internal fluid.
- `*SLUGS` is used to specify parameters relating to slug flow.
- `*PRINT` is used to request additional printed output to the main output file. Specifically, the `OUTPUT=SLUG FLOW` option is used to request additional data pertaining to fluid/slug flow.

If you would like to see an example of how these keywords are used in practice, refer to G03 - Rigid Spool.

Gravitational and Inertial Forces

The gravitational and inertial loads due to the presence of an internal fluid are superimposed on the main loads caused by the structure itself. The loads due to the fluid are as follows:

$$\underset{\sim}{F}_G = mg \qquad (1)$$

$$\underset{\sim}{F}_I = m\ddot{\underset{\sim}{x}} \qquad (2)$$

where:

- $\underset{\sim}{F}_G$ is the gravity force vector

- $\underset{\sim}{F}_I$ is the inertia force vector
- m is the mass of the entrained fluid
- g is the gravitational constant
- $\ddot{\underset{\sim}{x}}$ is the structural acceleration vector

Centrifugal Force

THEORY

Internal fluid flow through a curved pipe introduces a centrifugal force (Patrikalakis, 1986) defined as follows:

$$\underset{\sim}{F}_{Cent} = mV^2 \underset{\sim}{K} \qquad (1)$$

where:

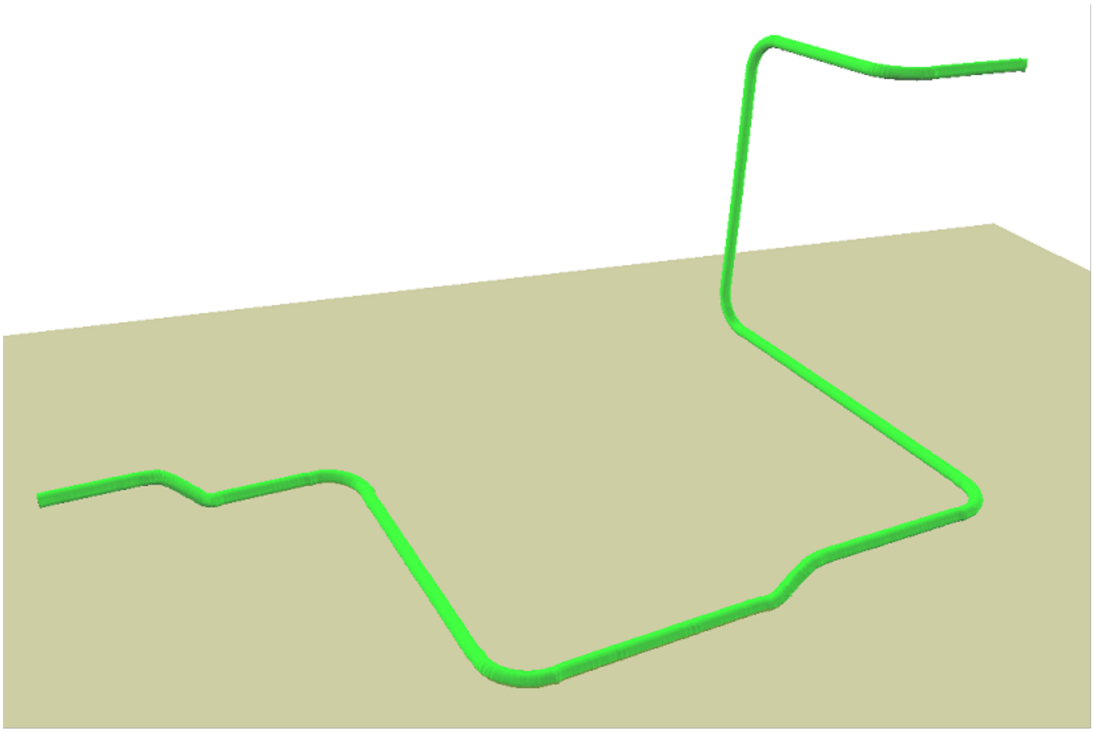
- $\underset{\sim}{F}_{Cent}$ is the centrifugal force vector
- m is the mass of the entrained fluid
- V is the velocity of the entrained fluid
- $\underset{\sim}{K}$ is the curvature vector

The centrifugal force is included in the external virtual work equation and expanded to facilitate incorporation into the Flexcom equations of motion, resulting in two additional terms. The first of these is applied as an additional loading to the element ends on the right hand side of the equations of motion, and this corresponds to Equation (1) above. The second term is conjugate to effective tension and is incorporated into the stiffness matrix on the left hand side of the equations of motion – refer to Dynamic Pressure for further details.

The figure below shows a rigid spool, a type of structure where centrifugal forces play an important role. Earlier versions of the software determined curvature from the nodal rotations, but this approach is unsuitable for structures which are unstressed in a curved configuration (e.g. goosenecks). The Flexcom output for rotational degrees of freedom basically define how the instantaneous or convected axes rotate from their initial, undeformed orientation. This means that for unstressed curved structures such as rigid spools, the curvature as determined from nodal rotations is practically zero, whereas in reality this is obviously not the case.

The centrifugal force is now evaluated on an element-by-element basis, with separate computations for each of its end nodes. The curvature vector at a given node is based on the instantaneous location of the node, and two adjacent nodes, located on either side of it. Given three points in space, it is possible to compute the curvature vector using standard geometrical relationships, and the centrifugal force is assumed to act in a direction consistent with the curvature vector (i.e. normal to the maximum curvature).

As the curvature computation is based on three successive nodes, naturally any element set which is used to define internal fluid loading or slug flow is assumed to be comprised of a continuous line of connected elements. Note also that for nodes located at model extremities, there is only one “adjacent” node, so no centrifugal force is applied.



Rigid Spool

RELEVANT KEYWORDS

- `*INTERNAL FLUID` is used to define the properties of an internal fluid. Centrifugal forces are included by default, but the `CENTRIFUGAL=` option allows you to suppress the effects of centrifugal forces in a particular analysis, allowing the relative significance of these loads to be assessed.
  - `*PRINT` is used to request additional printed output to the main output file. Specifically, the `OUTPUT=CENTRIFUGAL` option is used to request additional data pertaining to centrifugal force application.

Coriolis Force

THEORY

Internal fluid flow through a curved pipe introduces a Coriolis force (Patrikalakis, 1986) defined as follows:

$$\underset{\sim}{F}_{Cor} = -2mV\,\underset{\sim}{\omega} \times \underset{\sim}{e}$$

where:

- $\underset{\sim}{F}_{Cor}$ is the Coriolis force vector
- m is the mass of the entrained fluid
- V is the velocity of the entrained fluid
- $\underset{\sim}{\omega}$ is the vector of nodal rotations
- $\underset{\sim}{e}$ is a unit tangent vector

The tangent vector for a given element is assumed to be parallel to the instantaneous local-x axis of the element, and its orientation is consistent with the direction of fluid flow through the element.

RELEVANT KEYWORDS

- `*INTERNAL FLUID` is used to define the properties of an internal fluid. Coriolis forces are included by default, but the `CORIOLIS=` option allows you to suppress the effects of Coriolis forces in a particular analysis, allowing the relative significance of these loads to be assessed.

Hydrostatic Pressure

INTERNAL FLUID

The hydrostatic pressure due to an internal fluid is computed using the following equation.

$$P = \rho . \underset{\sim}{g} . \Delta h + P_U \qquad (1)$$

where:

- P is the total hydrostatic pressure
- ρ is the mass density of the internal fluid
- g is the acceleration due to gravity

- Δh is the hydrostatic pressure head. This is equal to the difference between the elevation of the point of interest (based on nodal position), and the total elevation of the relevant internal fluid (dictated by the *Level Above Mudline* entry under the *INTERNAL FLUID keyword).

- P_U is an (optional) user-defined constant pressure above hydrostatic. This is defined via the *Internal Pressure* entry under the *INTERNAL FLUID keyword.

For element sets which do not experience slug flow, the hydrostatic pressure within each element is computed independently based on Equation (1) above.

SLUG FLOW

For an element set which experiences slug flow at some time during a simulation (i.e. where the same element set is referenced under both the *INTERNAL FLUID and *SLUGS keywords), the hydrostatic pressure within each element is derived from (i) the hydrostatic pressure at a reference point and (ii) the connectivity of the element set. Specifically:

- The first node of the first element in the set is assumed to act as a reference point for the entire set.

- The hydrostatic pressure at the reference point P_{RP} is computed using Equation (1).

- The hydrostatic pressure at the end of the first element, $P_{1\text{-}END}$, is computed using Equation 2, where ρ_{INT} is the density of the fluid contained within the element, Δh is the difference in elevation between the element end points, and the remaining symbols are as defined previously.

$$P_{1-END} = P_{RP} + \rho_{INT} . \underset{\sim}{g} . \Delta h \qquad (2)$$

- If the element is fully filled with internal fluid, ρ_{INT} is the density of the internal fluid as specified in the *INTERNAL FLUID keyword. If the element is fully filled with slug material, ρ_{INT} is the density of the slug material as specified in the *SLUGS keyword. Where the element contains a mixture of fluid and slug, ρ_{INT} represents an equivalent fluid density which is computed on an integration point-by-point basis, with each integration point assumed to govern a local section of the element surrounding the integration point. Refer to Slug Flow for further information on partial filling.

- The hydrostatic pressure at the start of the second element in the set is equal to the value at the end of the first element i.e. $P_{2\text{-START}} = P_{1\text{-END}}$.

- The hydrostatic pressure at the end of the second element is computed using Equation 2, and so on.

- As the hydrostatic pressure is computed incrementally using the connectivity of the element set, Flexcom mandates that any element set which experiences slug loading must form a continuous line of elements from start to finish.

Note: In earlier versions of Flexcom (up to and including Flexcom 8.10.2), the hydrostatic pressure term was computed solely on the basis of the internal fluid definition, unaffected by presence of any slug flow. This approach was overly simplistic and in some cases could led to incorrect buoyancy forces being modelled, particularly for relatively long slugs in non-horizontal elements.

Dynamic Pressure

THEORY

Steady state internal fluid flow causes an increase in pipe wall tension (Fylling. et al., 1988) as follows:

$$\Delta T_w = m_u V^2 \qquad (1)$$

where:

- ΔT_w is the change in pipe wall tension

- m_u is the mass per unit length of the entrained fluid

- V is the velocity of the entrained fluid

In Flexcom, this effect is achieved by an increase in internal pressure, as follows:

$$P_i = P_i + \rho V^2 \qquad (2)$$

where:

- ρ is fluid density

Note also that the instantaneous fluid density, ρ, can vary depending on whether the local element contains standard internal fluid or a passing slug. Moreover, the density of individual slugs may also vary over time.

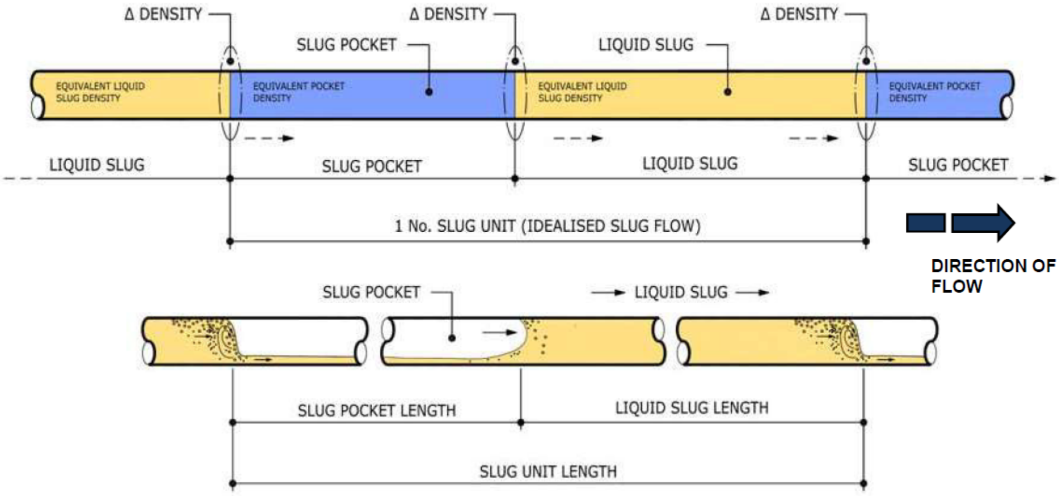
RELEVANT KEYWORDS

- *INTERNAL FLUID is used to define the properties of an internal fluid. Dynamic pressure effects are included by default, but the DYNAMIC PRESSURE= option allows you to suppress the effects of dynamic pressure in a particular analysis, allowing the relative significance of these loads to be assessed.

Slug Flow

INTRODUCTION

Slug flow is very complex in reality, with slugs lengthening and contracting, accelerating and decelerating, overtaking, merging, separating etc. General practice in industry is to adopt to simplified modelling approach which offers a compromise solution balancing accuracy and complexity alongside feasibility and data availability. Hence an 'idealised slug flow' model is often used, which represents slug flow as a moving 'train' of alternating liquid slug and pocket components with each component characterised by an equivalent density and velocity (Kavanagh et al., 2013).



Idealised Slug Flow (Kavanagh et al., 2013)

A ‘train’ in this context means a series of slug units being passed through a piping system. Slug trains may be regular or irregular.

- In regular train analyses, the same slug type is repeatedly passed through the system until a steady state response is achieved.
- In an irregular train analysis, different slug types are passed through the system.

Regardless of slug type (e.g. single unit, regular train or irregular train), either a fixed or variable slug unit can be used.

- Fixed units are assumed to not change properties as they pass through the system.
- Variable slug units have global properties that vary with time as they pass through e.g. the slug units can accelerate, decelerate, elongate, contract.

MODELLING OPTIONS IN FLEXCOM

Suggested Simulation Technique

While all scenarios noted above may be handled by Flexcom, typical engineering practice in industry appears to be the modelling of flow conditions in terms of irregular/fixed slugs. Firstly this involves the use of flow simulation software such as OLGA to produce an arbitrary time history of slug profile. Next the time history can then be transformed into a series of harmonics, based on a statistical representation of the flow regime. Each harmonic represents a certain portion of the slug spectrum, and a large number of slug harmonics may be contained in a single Flexcom simulation. While the raw data from a flow simulation program could be used to define an arbitrary time history of irregular/variable slugs explicitly in Flexcom, this approach is generally not adopted due to statistical uncertainty (the simulation would need to be extremely long, and/or many variations/windows would need to be simulated). Even with the irregular/fixed approach, multiple time domain representations (where the sequence of the various harmonics is different in each) are recommended in order to check sensitivity.

It is important to note that Flexcom is not capable of predicting fluid flow itself – it relies on flow simulation software for this information – rather it models the structural loads induced by the moving slugs.

Data Inputs

All slug related data is specified via the *SLUGS keyword, which is supported by the *INTERNAL FLUID keyword.

The governing properties for a slug are its length, density, velocity, the element set to which it applies, and the start time at which the slug enters the first element in the set. For slugs of constant properties, the slug velocity is an optional input, and by default the slug is assumed to move with the same velocity as the entraining fluid. Time varying slugs facilitate a more realistic slug flow definition, more complex than idealised slugs of constant velocity and density. In this case, the inputs include velocity and density at the slug head and tail respectively as a function of time. Slugs may be single or periodic. If a series of slugs is being considered, the number of slugs must be specified, along with a time delay between the entry time of each slug to the first element of the set. Refer to *SLUGS for more details regarding the specification of inputs.

Slug Induced Loads

Slug flow affects the various load terms induced by internal fluid, including Gravitational and Inertial Forces, Centrifugal Forces, Coriolis Force, Hydrostatic Pressure and Dynamic Pressure. You are invited to read these related articles for further background reading.

Slug Tracking

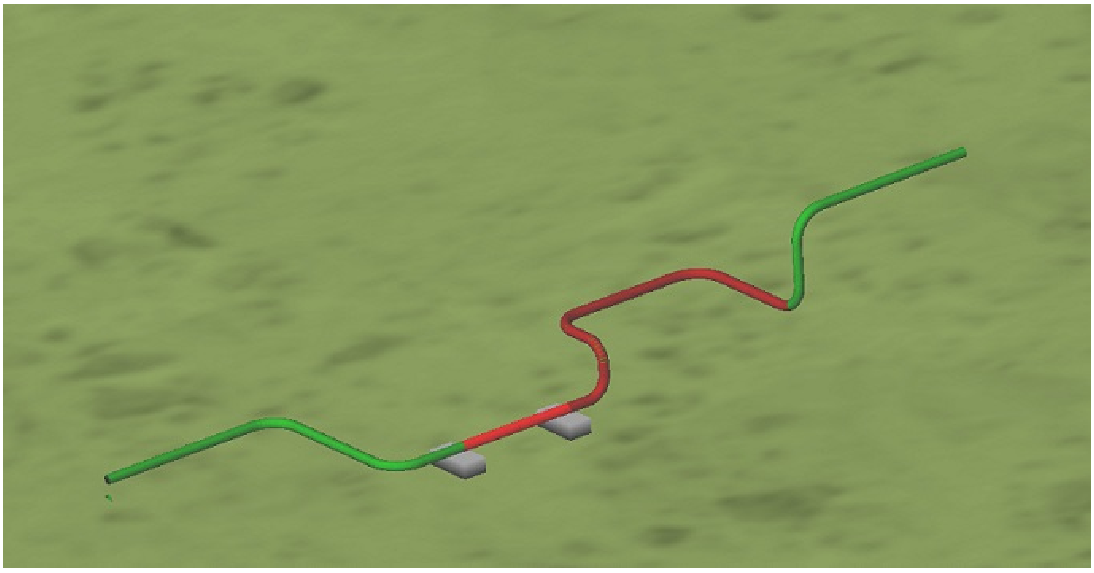
The position of each slug is monitored throughout an analysis and affected elements are identified and loaded accordingly.

Partial filling of an element with a slug may also be captured. For any element which is either partially or wholly filled with a slug, the number of integration points for that particular element is temporarily increased to the maximum number (which is equal to 10, assuming the user is not already using this number). The composition of the internal fluid is then determined on an integration point-by-point basis – each integration point is assumed to govern a section of the element in the region of that point. This ensures that the composition of the entrained fluid is accurately quantified, and means that relatively long elements may still be used without compromising the accuracy of the slug flow model.

It is possible to assign several different slugs to any given element set, and each ‘master’ slug may have a number of related ‘follower’ slugs. There may be several ‘master’ slugs, each of which will have different start times, and possible different velocities also. It is also possible to combine both constant and time-varying slugs in the same analysis. Slugs can potentially elongate and contract over the course of an analysis, given that slug head and tail velocity may be specified as a function of time. In this context it is important to note that any given location in the model can only be assigned a single slug type at any instant in time. If two or more conflicting definitions occur, then the program will simply assume that the latest slug definition takes precedence over earlier slugs. ‘Latest’ in this respect means the last relevant entry in the keyword file. In reality, slug flow is a complex phenomenon, and slugs can conceivably dissolve into, merge with or supersede other slugs. So it is possible that slugs may be defined to deliberately overlap with each other. However, inadvertent overlapping due to erroneous data specification cannot be detected by the program, so the onus is on the user to ensure the integrity of the input data.

Visualising Slug Flow

Flexcom visually displays slugs in the structural animation, which provides a very helpful means of visualising the slug flow post-simulation. The image below shows a portion of a rigid spool (shown in green) with two different slugs flowing through it (the main slug body is shown in dark red, with the tail portion shown in brighter red). Flexcom uses a series of auxiliary elements which track the slugs, highlighting whole or partial elements which contain slug material. If you do not see slugs in your structural animation, you should ensure that the Slug Display option is set to *Yes* (the default setting) before running your simulation, and that auxiliary element sets are enabled in the model view.



Slug Display

Although visualisation is an integral part of understanding slug induced loads, the use of auxiliary elements does result in increased database file sizes, which take up more disk space, and may also have some impact on simulation and post-processing times. For maximum efficiency, some users prefer to set the Slug Display option to *No*, and switch it on occasionally, for example to visually inspect slug paths during critical load cases only.

RELEVANT KEYWORDS

- `*SLUGS` is used to specify parameters relating to slug flow.
- `*INTERNAL FLUID` is used to define the properties of an internal fluid.
- `*PRINT` is used to request additional printed output to the main output file. Specifically, the `OUTPUT=SLUG FLOW` option is used to request additional data pertaining to fluid/slug flow.

If you would like to see an example of how these keywords are used in practice, refer to G03 - Rigid Spool. For a very quick overview, watch our Rigid Spool video online.

Inertial Effects

THEORY

An option is provided to specify that the internal fluid contribution to the structure axial inertia is to be omitted in a dynamic analysis. This modelling capability is specifically intended for the analysis of drilling risers in emergency disconnect mode (riser hangoff analysis). By default, when such a riser is specified as being full of internal fluid (typically seawater), Flexcom automatically includes the contribution of the internal fluid to the riser inertia (mass matrix) in both transverse and axial directions. This assumes that the internal fluid was entrapped in the riser and forced to move with it axially as well as laterally. This may not always be a reasonable assumption; in many cases the disconnected riser is open at its base so that fluid can move in and out of the riser instead of being constrained to move axially with it. The axial inertia omission option is intended for this application, to exclude internal fluid from the riser axial inertia in an emergency disconnect analysis. This option can also be considered for use in rigid riser installation analyses, where the riser is allowed to free flood as it is being run.

In a similar fashion, an option is provided to exclude the internal fluid contribution from the structural lateral inertia. Although the option is of very limited use, it may be desirable in certain circumstances. For instance, in an emergency disconnect analysis, it may be argued that the internal fluid does not contribute to the lateral inertia, and this option allows you to suppress this contribution, and consequently to quantify its effect on the overall solution.

RELEVANT KEYWORDS

- `*INTERNAL FLUID` is used to define the properties of an internal fluid. Fluid inertial effects are included by default, but the `AXIAL INERTIA=` and `LATERAL INERTIA=` options allow you to suppress the effects of fluid inertia in the axial and lateral directions, respectively.

If you would like to see an example of how this keyword is used in practice, refer to the emergency disconnect simulation in A01 - Deepwater Drilling Riser.

1.9.3.2 Current Loading

OVERVIEW

The various program options for modelling current loading are discussed in the following sections:

- Overview presents an overview of the range of current specification options available in Flexcom.
- User-Defined Currents discusses the user-subroutine option which facilitates the definition of time varying currents.
- Effect of Wave on Current discusses the approaches for modelling the variation in current profile with variation in water depth due to waves.
- VIV Drag describes a facility whereby vortex-induced vibration (VIV) drag coefficient amplification factors may be read from the results of a Shear7 analysis.

RELEVANT KEYWORDS

- `*CURRENT` is used to specify current loading.

If you would like to see an example of how this keyword is used in practice, refer to any of the standard Flexcom Examples.

Overview

SPECIFICATION OPTIONS

Flexcom provides several options in relation to current loading follows:

- Uniform. A horizontal current velocity distribution which is uniform in magnitude and direction through the water depth.
- Piecewise Linear. A horizontal current velocity distribution varying in magnitude, direction, or both, with depth.
  - Mudline Upwards. The velocity profile is specified as a function of height above zero datum, X=0.
  - MWL Downwards. The velocity profile is specified as a function of distance below the Mean Water Line (MWL) – this can be especially useful with an arbitrary seabed bathymetry.
- A user-subroutine option to define time varying currents or internal waves.

RELEVANT KEYWORDS

- *CURRENT is used to specify current loading. The UNIFORM, PIECEWISE LINEAR and SUBROUTINE inputs correspond to the specification options outlined above.

User Defined Currents

INTRODUCTION

The user-subroutine option facilitates the definition of time varying currents. This option is available for complete generality, particularly for situations where the standard modelling options do not completely cover user requirements. For example, soliton currents which occur in the South China sea are solitary internal waves which induce a time-varying current distribution.

OPERATION

Any generic compiler may be used to compile a DLL (dynamic link library) which may be linked into Flexcom. When you invoke a user subroutine option, you must have written the relevant source code, and compiled it into a DLL, before you perform the actual Flexcom analysis. Note also that there are standard templates provided in the Flexcom installation folder which illustrate how to create DLLs.

When you specify the name of the DLL file, Flexcom loads the DLL and searches for the load subroutine that was embedded into the DLL when it was compiled. Once Flexcom locates the subroutine, the subroutine is dynamically loaded into the Flexcom analysis module executable. The analysis then proceeds. When running, at each time step the user-subroutine is called once for every integration point on every submerged element. You must program your module to govern the current velocity and accelerations, typically as a function of depth and time. The subroutine arguments provide information regarding the instantaneous location of the integration point under consideration.

Finally, when the analysis is completed the temporary Flexcom executable is deleted. Note that if you use the facility, then the operation is completely automatic and transparent to you, and requires no intervention by you at runtime in compiling or linking the program.

SUBROUTINE FORMAT

A blank or template FORTRAN listing of subroutine currnt, the Flexcom current user-subroutine, is shown below. Experienced FORTRAN users will be aware that the comment lines are optional, and serve no function other than to make the code more comprehensible. Other than the comments, there are some lines that this user-subroutine must contain. These are:

- `the subroutine` statement
- `the implicit none` statement
- `the integer and real(8)` statements
- `the !dec$ attributes` statements to make the variables and subroutine available to Flexcom
- `the end subroutine` statement

The subroutine statement has ten arguments, to be described shortly. The `implicit none` statement requires that all variables used in the routine be declared explicitly. The `integer and real(8)` statements are used to declare integer and double precision variables. It is important to note that all calculations in Flexcom use double precision arithmetic, so you should always use the double precision form of any FORTRAN intrinsic functions you invoke (for example, use dcos rather than cos for the cosine of an angle). The `!dec$ attributes` statements are used to make the variables and subroutine available to Flexcom. Finally, the use of the end subroutine statement signals the end of the subroutine coding.

User-Subroutine Template

```
! Subroutine to define arbitrary current loading
subroutine currnt(time, x, y, z, vcx, vcy, vcz, acx, acy, acz)
!dec$ attributes dllimport, stdcall, reference :: currnt
  implicit none

  real(8), intent(in) :: time
  !dec$ attributes reference :: time
  real(8), intent(in) :: x
  !dec$ attributes reference :: x
  real(8), intent(in) :: y
  !dec$ attributes reference :: y
  real(8), intent(in) :: z
  !dec$ attributes reference :: z
  real(8), intent(inout) :: vcx
  !dec$ attributes reference :: vcx
  real(8), intent(inout) :: vcy
  !dec$ attributes reference :: vcy
  real(8), intent(inout) :: vcz
  !dec$ attributes reference :: vcz
```

```
    real(8), intent(inout) :: acx
    !dec$ attributes reference :: acx
    real(8), intent(inout) :: acy
    !dec$ attributes reference :: acy
    real(8), intent(inout) :: acz
    !dec$ attributes reference :: acz

    !
    ! Variable Names
    !
    !     time    : The instantaneous solution time
    !     x       : The global X position of the integration point under consideration
    !     y       : The global Y position of the integration point under consideration
    !     z       : The global Z position of the integration point under consideration
    !     vcx     : The current velocity in the global X direction
    !     vcy     : The current velocity in the global Y direction
    !     vcz     : The current velocity in the global Z direction
    !     acx     : The current acceleration in the global X direction
    !     acy     : The current acceleration in the global Y direction
    !     acz     : The current acceleration in the global Z direction
    !
    ! Declare local variables.

    ! Insert coding to define user-specified loading below this line.

  end subroutine currnt
```

The ten arguments of subroutine `currnt` are listed in the comments statements above. The first four, comprising solution time and instantaneous location of point of interest, are passed to `currnt` for information only, to be used in calculating the user-defined current velocity and acceleration terms at that point in space and time. The values of these variables should not be altered. The remaining six variables are the ones which you control as the user. The source code to compute and define the current velocity and acceleration in each degree of freedom should be inserted after the 'Insert coding' comment, but before the end subroutine statement.

All applied loads and displacements in Flexcom are optionally ramped or gradually increased up to the full values over a time period (the ramp time) you define. Further information is provided in Load Ramping.

Effect of Wave on Current

THEORY

In early versions of Flexcom (prior to Version 6.1), the current velocity at any point below the MWL was time-invariant (unless the point was above the instantaneous water surface at any stage), and calculated from the user-specified velocity distribution in the usual way.

In more recent versions of Flexcom, if you input a piecewise linear current profile, then that profile is stretched or compressed so that the current at the instantaneous water surface remains constant throughout an analysis, as does the velocity as a proportion of the instantaneous depth. This is done by factoring the profile you specify at each solution time by ($d/d+\eta$), where d is the water depth and η is the instantaneous wave elevation.

In practice this means that Flexcom has to calculate the water surface elevation at each integration point on each element at each iteration at each solution time. In the vast majority of analyses this is not sufficiently onerous to adversely affect runtimes. However in a very small number of random sea analyses, particularly where a multi-directional sea was specified, runtimes can increase substantially.

The procedure of stretching user-defined piecewise linear current distributions remains the default in the latest version of Flexcom. However an option is provided to over-ride this default, and to instruct the program to return to the earlier method of using a time-invariant current variation with depth.

RELEVANT KEYWORDS

- `*CURRENT` is used to specify current loading. Specifically, the `LEVEL=` input is used to define the current stretching algorithm.

VIV Drag

THEORY

This section describes a facility whereby vortex-induced vibration (VIV) drag coefficient amplification factors may be read from the results of a Shear7 analysis.

Flexcom provides an option to allow vortex-induced vibration (VIV) drag coefficient amplification factors to be read from the results of a Shear7 analysis. If this option is invoked, the relevant inputs are the name of the Shear7 analysis from which drag amplification factors are to be read, and the set of elements to which these factors are to be applied.

How this capability operates is as follows. When computing current forces, Flexcom loops over all elements in the nominate element set. For a given integration point, the program finds the drag amplification factor output for the nearest points in the Shear7 data, and then calculates a value for the integration point by linear interpolation. This factor then multiplies the user-specified drag coefficient when the current force for the integration point is being found via Morison's Equation.

RELEVANT KEYWORDS

- `*CURRENT` is used to specify current loading.
- `*VIV DRAG` is used to instruct Flexcom to read vortex-induced vibration (VIV) drag coefficient amplification factors from the results of a Shear7 analysis.

If you would like to see an example of how these keywords are used in practice, refer to A04 - TTR Wake Interference.

1.9.3.3 Wave Loading

WAVE SPECIFICATION OPTIONS

Flexcom presents a wide range of wave specification options. There are three regular wave options as follows:

- Regular Airy describes linear Airy wave theory.
- Stokes V describes Stokes V wave theory, and presents the various options for the calculation of vessel response.
- Dean's Stream describes Dean's Stream wave theory, and presents the various options for the calculation of vessel response.

There are five options for defining a random seastate in terms of a wave spectrum, as follows:

- Pierson-Moskowitz describes the Pierson-Moskowitz wave spectum model, and presents the various formats for input data specification.
- Jonswap describes the Jonswap wave spectum model, and presents the various formats for input data specification.
- Ochi-Hubble outlines the Ochi-Hubble wave spectum model.

- Torsethaugen outlines the Torsethaugen wave spectum model.
- User-defined describes a facility whereby you can input an arbitrary wave spectrum directly in terms of a series of data pairs.

You may also define a random seastate in terms of a time history of water surface elevation; the name of the file containing the time is the actual program input in the case. When this option is invoked, Flexcom first calculates a wave spectrum from the time history. The output from this process is then treated as a series of regular waves, each with its individual period, amplitude and phase.

VALID COMBINATIONS

Waves can be specified in various combinations in a time domain dynamic analysis as follows:

(i) One Stokes V wave.

(ii) One Dean's Stream wave.

(iii) One or multiple regular Airy waves.

(iv) One or multiple random seas (characterised by Pierson-Moskowitz, Jonswap, Ochi-Hubble, Torsethaugen or User-Defined spectra).

(v) One wave elevation time history.

(vi) One or multiple random seas of any type combined with one or more regular Airy waves.

The full range of inputs is available to you to define a spectrum for the above combinations. For example, each spectrum can have a separate dominant direction, and may be uni- or multi-directional, regardless of combination.

A narrower range of combinations is available in the frequency domain:

(i) One regular Airy wave.

(ii) Multiple regular Airy waves. This option consists of a series of regular wave analyses over a range of periods, each wave having the same amplitude. The solution for each wave period is performed independently. The only use of this option is to generate RAOs of the response. These are calculated by dividing the amplitude of the response by the wave amplitude, so it is often the case that the wave is of unit amplitude.

(iii)One random sea, characterised by a Pierson-Moskowitz, Jonswap, Ochi-Hubble, Torsethaugen or user-defined wave spectrum. Flexcom first discretises the spectrum into component harmonics, before assembling and solving the equations of motion at each harmonic. However, the solutions at each harmonic are not independent, but are in fact coupled through the drag linearisation process. Further details are provided in Frequency Domain Analysis, where the frequency domain theory is described in detail.

WAVE PREVIEW OPTIONS

Flexcom provides a helpful Wave Spectrum Plot option in the user interface. If there is random sea data defined in the analysis, it displays a plot of the resultant wave spectrum. There may be more than one wave spectrum defined, in which case the various spectra are combined. Another helpful feature is the Water Elevation Plot option. This affords you an instantaneous preview of results, by producing a time history of the water surface elevation at the origin. This option is particularly useful if you are interested in performing a random sea analysis in the time domain – you can verify the relevant input data is correct before undertaking a full 3-hour simulation.

RELEVANT KEYWORDS

Regular Waves

- `*WAVE-REGULAR` is used to specify regular Airy wave loading.
- `*WAVE-STOKES` is used to specify Stokes V regular wave loading.
- `*WAVE-DEANS` is used to specify Dean’s Stream regular wave loading.

Random Seas

- `*WAVE-PIERSON-MOSKOWITZ` is used to specify a Pierson-Moskowitz random sea wave spectrum or spectra.
- `*WAVE-JONSWAP` is used to specify a JONSWAP random sea wave spectrum or spectra.
- `*WAVE-OCHI-HUBBLE` is used to specify an Ochi-Hubble random sea wave spectrum or spectra.

- *WAVE-TORSETHAUGEN is used to specify a Torsethaugen random sea wave spectrum or spectra.
- *WAVE-TIME-HISTORY is used to specify a random seastate in terms of a time history of water surface elevation.
- *WAVE-USER-DEFINED is used to specify a User-Defined random sea wave spectrum or spectra.

Miscellaneous

- *WAVE-GENERAL is used to specify miscellaneous parameters to wave loading.

Regular Airy Wave

THEORY

A single regular wave travelling in the direction of θ^o anticlockwise relative to the global Y direction, results in a water surface elevation $\eta(y,z,t)$ at a point in the wave field which is represented by the equation

$$\eta(y,z,t)=a\cos(ks-\omega t) \qquad (1)$$

where:

- a is the wave amplitude (MWL to crest or trough)
- k is the wave number $\left(k=\dfrac{2\pi}{\lambda}\right)$
- λ is the wavelength
- s is the horizontal distance from vertical axis (Y=Z=0) to point in question $(S = y\cos\theta+z\sin\theta)$
- y is the horizontal distance from vertical axis to point in question in global Y direction
- z is the horizontal distance from vertical axis to point in question in global Z direction

The wave spacial origin can be specified via the *WAVE-GENERAL keyword such that S becomes $S = (\hat{y} + yCos\theta + \hat{z} + zCos\theta)$ where y & ? are the specified y and z offsets from the origin.

- ω is the wave circular frequency $\left(\omega = \dfrac{2\pi}{T}\right)$
- T is the wave period
- t is the present solution time

Water particle velocities and accelerations at any point through the depth in the horizontal and vertical directions are calculated using Airy wave theory via the following expressions:

$$u_w = \frac{agk}{\omega}\frac{\sinh kx}{\cosh kd}\sin(ks - \omega t) \qquad (2)$$

$$v_w = \frac{agk}{\omega}\frac{\cosh kx}{\cosh kd}\cos(ks - \omega t) \qquad (3)$$

$$\dot{u}_w = -agk\frac{\sinh kx}{\cosh kd}\cos(ks - \omega t \qquad (4)$$

$$\dot{v}_w = agk\frac{\cosh kx}{\cosh kd}\sin(ks - \omega t) \qquad (5)$$

where:

- u_w is the water particle velocity in vertical (global X) direction
- g is the gravitational constant
- x is the vertical distance from mudline to point in question
- d is the water depth
- v_w is the water particle velocity in horizontal direction of wave travel

- $\dot{u}_w$ is the water particle acceleration in vertical (global X) direction
- $\dot{v}_w$ is the water particle acceleration in horizontal direction of wave travel

Where several regular Airy waves are specified, the above five Equations are replaced by the following summations:

$$\eta(y,z,t)=\sum_{n=1}^{N} a_n \cos(k_n s_n - \omega_n t + \phi_n)$$
$$=\sum_{n=1}^{N} \eta_i \qquad (6)$$

$$u_w = \sum_{n=1}^{N} \frac{a_n g k_n}{\omega_n} \frac{\sinh k_n x}{\cosh k_n d} \sin(k_n s_n - \omega_n t + \phi_n) \qquad (7)$$

$$v_w = \sum_{n=1}^{N} \frac{a_n g k_n}{\omega_n} \frac{\cosh k_n x}{\cosh k_n d} \cos(k_n s_n - \omega_n t + \phi_n) \qquad (8)$$

$$\dot{u}_w = -\sum_{n=1}^{N} a_n g k_n \frac{\sinh k_n x}{\cosh k_n d} \cos(k_n s_n - \omega_n t + \phi_n) \qquad (9)$$

$$v_w = \sum_{n=1}^{N} a_n g k_n \frac{\cosh k_n x}{\cosh k_n d} \sin(k_n s_n - \omega_n t + \phi_n) \qquad (10)$$

where:

- a_n is the amplitude of n^th regular wave
- k_n is the wave number of n^th regular wave
- s_n is the horizontal distance from vertical axis to point in question in direction of travel of n^th regular wave $\left(s_n = y\cos\theta_n + z\sin\theta_n\right)$
- θ_n is the angle of approach of n^th regular wave relative to global Y axis (positive anticlockwise

- ω_n is the circular frequency of n[th] regular wave

- ϕ_n is the phase of n[th] regular wave relative to some datum

- N is the number of user-specified regular Airy waves

RELEVANT KEYWORDS

- *WAVE-REGULAR is used to specify regular Airy wave loading.

If you would like to see an example of how this keyword is used in practice, refer to C02 - Multi-Line Flexible System.

Stokes V Wave

OVERVIEW

Flexcom provides a range of options for calculating the response of a vessel to a Stokes V regular wave, when that response is defined in terms of RAOs in the various vessel degrees of freedom. This section first discusses the Stokes V Wave Theory, then presents the various options for vessel motion calculations (denoted Fifth Order,Equivalent Airy Wave and Superposition of Harmonics).

RELEVANT KEYWORDS

- *WAVE-STOKES is used to specify Stokes V regular wave loading.

Stokes V Wave Theory

The following equations define the wave height H and the wave celerity c for a Stokes V regular wave (Chakrabarti, 1987)

$$H = \frac{2}{k}\left[\mu + B_{33}\mu^3 + \left(B_{35} + B_{55}\right)\mu^5\right] \qquad (1)$$

and

$$c^2 = c_0^2\left(1 + \mu^2 C_1 + \mu^4 C_2\right) \qquad (2)$$

where:

- c_0 is the celerity given by linear wave theory
- μ is an unknown along with the wave number k

and the quantities B_{ij} and C_i are functions only of kd and are listed in 'Stokes V Coefficients' table below. Note that in these expressions $S = \sinh(kd)$ and $C = \cosh(kd)$, where d represents water depth.

For a given (user-specified) value of H, the quantities μ and k are determined from Equations (1) and (2) through an iterative Newton-Raphson technique. Once they have been found, the water surface elevation is found from the following:

$$\eta(t) = \frac{1}{k}[\mu\cos\varphi + \left(B_{22}\mu^2 + B_{24}\mu^4\right)\cos 2\varphi + (B_{33}\mu^3 + B_{35}\mu^5)\cos 3\varphi + B_{44}\mu^4\cos 4\varphi + B_{44}\mu^5\cos 5\varphi] \qquad (3)$$

where:

$$\varphi = ks - \omega t$$

Again the relevant B_{ij} quantities are listed in the 'Stokes V Coefficients' table below.

The fifth-order velocity potential Φ is written in a series form as:

$$\Phi = \frac{c}{k}\sum_{n=1}^{5}\mu_n \cosh nks \sin n\varphi \qquad (4)$$

where the non-dimensional coefficients μ_n are given by:

$$\mu_1 = \mu A_{11} + \mu^3 A_{13} + \mu^5 A_{15}$$
$$\mu_2 = \mu^2 A_{22} + \mu^4 A_{24}$$
$$\mu_3 = \mu^3 A_{33} + \mu^5 A_{35}$$
$$\mu_4 = \mu^4 A_{44}$$
$$\mu_5 = \mu^5 A_{55} \qquad (5)$$

Expressions for the water particle velocities and accelerations are obtained in the usual way by differentiating Φ . For example, the vertical water particle velocity is given by:

$$u_x = \frac{\partial \Phi}{\partial x}$$

The complete expressions for u_x , u_y , $\dot{u}_x$, and $\dot{u}_y$ are available in standard texts such as Chakrabarti (1987), and are omitted here.

Stokes V Coefficients

$$A_{11} = \frac{1}{S}$$

$$A_{13} = \frac{-C^2(5C^2+1)}{8S^5}$$

$$A_{15} = \frac{-(1184C^{10} - 1440C^8 - 1992C^6 + 2641C^4 - 249C^2 + 18)}{1536S^{11}}$$

$$A_{22} = \frac{3}{8S^4}$$

$$A_{24} = \frac{(192C^8 + 424C^6 - 312C^4 + 480C^2 - 17)}{768S^{10}}$$

$$A_{33} = \frac{(13 - 4C^2)}{64S^7}$$

$$A_{35}=\frac{(512C^{12}+4224C^{10}-6880C^{8}-12{,}808C^{6}+16{,}704C^{4}-3154C^{2}+107)}{4096S^{13}(6C^{2}-1)}$$

$$A_{44}=\frac{(80C^{6}-816C^{4}+1338C^{2}-197)}{1536S^{10}(6C^{2}-1)}$$

$$A_{55}=\frac{-(2880C^{10}-72{,}480C^{8}+324{,}000C^{6}-432{,}000C^{4}+163{,}470C^{2}-16{,}245)}{61{,}440S^{11}(6C^{2}-1)(8C^{4}-11C^{2}+3)}$$

$$B_{22}=C\frac{(2C^{2}+1)}{4S^{3}}$$

$$B_{24}=\frac{C(272C^{8}-504C^{6}-192C^{4}+322C^{2}+21)}{384S^{9}}$$

$$B_{33}=\frac{3(8C^{6}+1)}{64S^{6}}$$

$$B_{35}=\frac{(88{,}128C-208{,}224C^{12}+70{,}848C^{10}+54{,}000C^{8}-21{,}816C^{6}+6{,}264C^{4}-54C^{2}-81)}{12{,}288S^{12}(6C^{2}-1)}$$

$$B_{44}=\frac{C(768C^{10}-448C^{8}-48C^{6}+48C^{4}+106C^{2}-21)}{384S^{9}(6C^{2}-1)}$$

$$B_{55}=\frac{(192{,}000C^{16}-262{,}720C^{14}+83{,}680C^{12}+20{,}160C^{10}-7280C^{8})}{12{,}288S^{10}(6C^{2}-1)(8C^{4}-11C^{2}+3)}+\frac{(7160C^{6}-1800C^{4}-1050C^{2}+225)}{12{,}288S^{10}(6C^{2}-1)(8C^{4}-11C^{2}+3)}$$

$$C_{1}=\frac{(8C^{4}-8C^{2}+9)}{8S^{4}}$$

$$C_{2}=\frac{(3840C^{12}-4096C^{10}+2592C^{8}-1008C^{6}+5944C^{4}-1830C^{2}+147)}{512S^{10}(6C^{2}-1)}$$

$$C_3 = \frac{1}{4SC} \qquad C_4 = \frac{(12C^8 + 36C^6 - 162C^4 + 141C^2 - 27)}{192CS^9}$$

Vessel Motions - Fifth Order

The high frequency response of the vessel reference point in local degree of freedom i to a Stokes V wave when you nominate the Fifth Order option is defined by:

$$\hat{R}_i = RAO_i(\theta_h, \omega)\ \bar{\eta} \qquad (1)$$

where:

$$\bar{\eta} = \frac{1}{k}[\mu\cos\varphi^* + \left(B_{22}\mu^2 + B_{24}\mu^4\right)\cos 2\varphi^* + (B_{33}\mu^3 + B_{35}\mu^5)\cos 3\varphi^* + B_{44}\mu^4\cos 4\varphi^* + B_{55}\mu^5\cos 5\varphi^*] \qquad (2)$$

and:

- $RAO_i(\theta_h, \omega)$ is the vessel RAO in degree of freedom i, which is a function of wave heading θ_h and wave frequency ω
- φ^* = $ks - \omega t + \Phi_i(\theta_h, \omega)$
- $\Phi_i(\theta_h, \omega)$ is the vessel phase angle in degree of freedom i, which is also a function of θ_h and ω

Remaining symbols are as previously defined. This option gives a fifth order vessel response to the fifth order wave.

Vessel Motions - Equivalent Airy Wave

This option is provided to allow you to specify a sinusoidal vessel response to a fifth order wave. To invoke this facility you specify an equivalent Airy wave amplitude when you are inputting the Stokes V wave data; you can optionally specify an equivalent wave period. What then happens is the following. The Stokes V wave height, period and direction are used in the normal way in calculating water particle kinematics and ultimately wave forces. The equivalent Airy wave parameters on the other hand are combined with the vessel RAOs and phase angles to calculate vessel motions. This will ensure a sinusoidal response.

You can specify an equivalent Airy wave amplitude without inputting a corresponding period, in which case the Stokes V wave period is used by default.

The high frequency response of the vessel reference point in local degree of freedom i to a Stokes V wave when you nominate the Equivalent Airy Wave option is defined by

$$\hat{R}_i = RAO_i(\theta_h, \omega_e)\, A_e \cos\{ks - \omega_e t + \Phi_i(\theta_h, \omega_e)\} \qquad (1)$$

where:

- $RAO_i(\theta_h, \omega_e)$ is the vessel RAO in degree of freedom i, which is a function of wave heading θ_h and equivalent wave frequency ω_e
- ω_e is the equivalent Airy wave circular frequency $\left(\omega_e = \dfrac{2\pi}{T_e}\right)$
- T_e is the period of equivalent Airy wave
- A_e is the amplitude of equivalent Airy wave
- $\Phi_i(\theta_h, \omega_e)$ is the vessel phase angle in degree of freedom i, which is also a function of θ_h and ω_e

Remaining symbols are as previously defined.

Vessel Motions – Superposition of Harmonics

The high frequency response of the vessel reference point in local degree of freedom i to a Stokes V wave when you nominate the Superposition of Harmonics option is defined by:

$$\begin{aligned}\hat{R}_i = \frac{1}{k}[&RAO_i(\theta_h,\omega)\mu\cos\{ks-\omega t+\Phi_i(\theta_h,\omega)\} + \\ &RAO_i(\theta_h,2\omega)\left(B_{22}\mu^2+B_{24}\mu^4\right)\cos 2\{ks-\omega t+\Phi_i(\theta_h,2\omega)\} + \\ &RAO_i(\theta_h,3\omega)(B_{33}\mu^3+B_{35}\mu^5)\cos 3\{ks-\omega t+\Phi_i(\theta_h,3\omega)\} + \\ &RAO_i(\theta_h,4\omega)B_{44}\mu^4\cos 4\{ks-\omega t+\Phi_i(\theta_h,4\omega)\} + \\ &RAO_i(\theta_h,5\omega)B_{55}\mu^5\cos 5\{ks-\omega t+\Phi_i(\theta_h,5\omega)\}]\end{aligned} \quad (1)$$

where:

- $RAO_i(\theta_h,n\omega)$ is the vessel RAO in degree of freedom i, which is a function of wave heading θ_h and wave frequency $n\omega$
- $\Phi_i(\theta_h,n\omega)$ is the vessel phase angle in degree of freedom i, which is also a function of θ_h and $n\omega$

Remaining symbols are as previously defined. With this option, the vessel response is a summation of five components, which correspond to the “components” in the expression for water surface elevation in the Stokes v Wave - Theory (Eq.3). RAO values and phase angles corresponding to each component in the Stokes v Wave - Theory (Eq.3) are extracted from the user-specified data in the usual way, and the vessel response is found by summing over the five terms in a similar manner as would be used in a random sea. Obviously for this exercise to be meaningful RAOs and phase angles must be defined over the full range of circular frequencies from to .

Deans Stream Wave

OVERVIEW

This section describes the implementation of Dean’s stream function in Flexcom. Dean's Stream Theory is first outlined, and then the vessel response to the stream function is described. Each of the three vessel motion options – Multiple Order, Equivalent Airy Wave and Superposition of Harmonics – are discussed in turn.

Stream function theory, a non-linear wave theory related to that of Stokes, was developed by Dean in order to examine fully non-linear waves numerically. Flexcom uses a version of Dean's stream function known as Fenton's Fourier series theory to compute the wave kinematics for a given set of wave parameters. This method eliminates the need to use tabular or graphical presentations of the solutions in order to solve the stream function equation and is an efficient and accurate solution method.

RELEVANT KEYWORDS

- *WAVE-DEANS is used to specify Dean's Stream regular wave loading.

Deans Stream Theory

Fourier series wave theory derives its name from an approximate solution to the governing wave equation using a Fourier cosine series in θ, as follows:

$$\psi(x,\theta) = -\bar{u}(d+x) + \left(\frac{g}{k^3}\right)^{\frac{1}{2}} \sum_{j=1}^{N} B_J \frac{\sinh jk(d+x)}{\cosh(jkd)} \cos(j\theta) \qquad (1)$$

where:

- ψ is the Stream Function
- $\bar{u}$ is the mean value of horizontal fluid velocity for a constant value of x, over one wavelength
- d is the water depth
- x is the height above the mean water line
- g is the acceleration due to gravity
- k is the wave number
- Bj is a set of dimensionless Fourier coefficients
- N is the order of the stream function. This is a measure of how non-linear the wave is. In deep water the order can be relatively low (between 3 and 5), while in very shallow water the order can be as high as 30.

This formulation for the stream function, combined with the dynamic and kinematic free surface boundary conditions and the dispersion relationship are used to set up a system of complex non-linear equations. The unknown variables in the stream function equation can then be calculated for any given set of wave height, wave period, water depth and underlying current velocity. The wave kinematics (surface elevation, water particle velocities and accelerations etc.) are then derived from the solution of the governing equation.

The water surface elevation at any point in the wave field is found using the following equation:

$$\eta(\theta) = \sum_{j=1}^{N-1} f_j \cos(j\theta) + 0.5 f_N \cos(N\theta) \qquad (2)$$

where:

$$f_j = \frac{2}{N}\left\{0.5\eta_0 + \sum_{m=1}^{N-1} \eta_m \cos\frac{jm\pi}{N} + \frac{1}{2}\eta_N \cos j\pi \right\} \qquad (3)$$

and:

- θ = ks - wt
- s is the horizontal distance from vertical axis Y=Z=0 to point in question (s = ycosϕ + zsinϕ)
- y is the horizontal distance from vertical axis to point in global Y direction
- z is the horizontal distance from vertical axis to point in global Z direction

The wave spacial origin can be specified via the *WAVE-GENERAL keyword such that s becomes $S = (\hat{y} + yCos\phi + \hat{z} + zCos\phi)$ where y & ? are the specified y and z offsets from the origin.

- ϕ is the wave direction relative to global Y
- w is the wave circular frequency (w = 2π/T)
- T is the wave period
- t is the present solution time

Vessel Motions - Multiple Order

The high frequency response $\hat{R}_i$ of the vessel reference point in local degree of freedom i to a Dean’s Stream Function wave when you nominate the Multiple Order option is defined by:

$$\hat{R}_i = RAO_i(\theta_h, \omega)\,\bar{\eta} \qquad (1)$$

where:

$$\bar{\eta} = \sum_{j=1}^{N-1} f_j \cos(j\varphi^*) + 0.5 f_N \cos(N\varphi^*) \qquad (2)$$

and:

- $RAO_i(\theta_h,\omega)$ is the vessel RAO in degree of freedom i, which is a function of wave heading θ_h and wave frequency ω
- f is the surface elevation function as defined in the Deans Stream Wave - Theory (Eq.3)
- φ^* = $ks - \omega t + \Phi_i(\theta_h, \omega)$
- $\Phi_i(\theta_h,\omega)$ is the vessel phase angle in degree of freedom i, which is also a function of θ_h and ω .

Remaining symbols are as previously. This option gives a vessel response of order corresponding to the order of expansion of the Fourier Series.

Vessel Motions – Equivalent Airy Wave

This option is provided to allow you to specify a sinusoidal vessel response to the multiple order stream function wave. To invoke this facility you specify an equivalent Airy wave amplitude when you are inputting the Stream Function data; you can optionally specify an equivalent wave period. What then happens is the following. The Dean's Stream Function wave height, period and direction are used in the normal way in calculating water particle kinematics and ultimately wave forces. The equivalent Airy wave parameters on the other hand are combined with the vessel RAOs and phase angles to calculate vessel motions. This ensures a sinusoidal response. You can specify an equivalent Airy wave amplitude without inputting a corresponding period, in which case the Dean's Stream Function wave period is used by default.

The high frequency response $\hat{R}_i$ of the vessel reference point in local degree of freedom i to a Dean's Stream Function wave when you nominate the Equivalent Airy Wave option is defined by:

$$\hat{R}_i = RAO_i(\theta_h, \omega_e)\, A_e \cos\{ks - \omega_e t + \Phi_i(\theta_h, \omega_e)\} \qquad (1)$$

where:

- $RAO_i(\theta_h, \omega_e)$ is the vessel RAO in degree of freedom i, which is a function of wave heading θ_h and equivalent wave frequency ω_e
- ω_e is the equivalent Airy wave circular frequency $\left(\omega_e = \dfrac{2\pi}{T_e}\right)$
- T_e is the period of equivalent Airy wave
- A_e is the amplitude of equivalent Airy wave
- $\Phi_i(\theta_h, \omega_e)$ is the vessel phase angle in degree of freedom i, which is also a function of θ_h and ω_e

Remaining symbols are as previously.

Vessel Motions – Superposition of Harmonics

The high frequency response $\hat{R}_i$ of the vessel reference point in local degree of freedom i to a Dean's Stream Function wave when you nominate the Superposition of Harmonics option is defined by:

$$R_j = \sum_{j=1}^{N-1} f_j \cos j\{ks - \omega t + \Phi_j(\theta_h, j\omega)\} RAO_i(\theta_h, j\omega) + \frac{1}{2} f_N \cos N\{ks - \omega t + \Phi_j(\theta_h, N\omega)\} RAO_i(\theta_h, N\omega) \quad (1)$$

where:

- $RAO_i(\theta_h, j\omega)$ is the vessel RAO in degree of freedom i, which is a function of wave heading θ_h and wave frequency $j\omega$
- $\Phi_i(\theta_h, j\omega)$ is the vessel phase angle in degree of freedom i, which is also a function of θ_h and $j\omega$
- f is the surface elevation function as defined in the Deans Stream Wave - Theory (Eq.3).

Remaining symbols are as previously. With this option, the vessel response is a summation of N components, which correspond to the "components" in the expression for water surface elevation in the Deans Stream Wave - Theory (Eq.2). RAO values and phase angles corresponding to each component in the Deans Stream Wave - Theory (Eq.2) are extracted from the user-specified data in the usual way, and the vessel response is found by summing over the N terms in a similar manner as would be used in a random sea. Obviously for this exercise to be meaningful RAOs and phase angles must be defined over the full range of circular frequencies from ω to $N\omega$.

Pierson-Moskowitz Wave

THEORY

Pierson and Moskowitz (1964) developed a spectral formulation for a fully developed seastate. The fetch and duration are considered infinitely large (measured data from the North Atlantic was used during the development of the spectral formulation), so that the wind must be of relatively constant speed over a relatively large area for the underlying assumptions to remain valid. The Pierson-Moskowitz spectrum is based on the following formulation:

$$S_\eta(f)=\frac{1}{4\pi}\frac{H_s^2}{T_z^4}\frac{1}{f^5}e^{\left[-\frac{1}{\pi T_z^4 f^4}\right]} \qquad (1)$$

where:

- f is the wave frequency in Hertz
- Hs is the significant wave height
- Tz is the mean zero up-crossing period

SPECIFICATION OPTIONS

Flexcom offers a choice of formats for defining a Pierson-Moskowitz wave spectrum. The standard or default option is in terms of significant wave height H_s and mean zero up-crossing period T_z as per the Equation above. Alternatively, you can define the spectrum in terms of H_s and T_p, where T_p is the spectrum peak period. In this case, T_p is converted internally to T_z using the relationship:

$$T_Z = T_p / 1.4$$

RELEVANT KEYWORDS

- `*WAVE-PIERSON-MOSKOWITZ` is used to specify a Pierson-Moskowitz random sea wave spectrum or spectra.

If you would like to see an example of how this keyword is used in practice, refer to B01 - Steel Catenary Riser.

Jonswap Wave

THEORY

Hasselmann et al. (1973) developed a spectral formulation for developing seastates known as Jonswap (it originated during a Joint North Sea Wave Observation Project). It is essentially a modified version of the Pierson-Moskowitz spectrum, with some adjustments to align the spectrum with the measured North Sea data. The Jonswap spectrum is based on the following formulation:

$$S_\eta(f)=\alpha g^2.\frac{1}{(2\pi)^4}\frac{1}{f^5}e^{\left[-\frac{5}{4}\left(\frac{f}{f_p}\right)^{-4}\right]}.\gamma^{FAC} \qquad (1)$$

where:

$$FAC=e^{\left[-\frac{(f-f_p)^2}{2\sigma^2 f_p^2}\right]} \qquad (2)$$

and:

- α is the Phillip's constant
- f_p is the spectral peak frequency $\left(f_p=\frac{1}{T_p}\right)$
- T_p is the peak period
- γ is the peakedness parameter
- σ is the spectral width parameter; σ = 0.07 for $f<f_p$, σ = 0.09 for $f\geq f_p$.

SPECIFICATION OPTIONS

Flexcom offers a choice of formats for defining a Jonswap wave spectrum. The standard or default option is in terms of peak frequency f_p, Phillips constant α, and spectrum peakedness parameter γ parameter, as per Equation 1 above. Alternatively you can define the data in terms of:

- Significant wave height H_s and mean zero up-crossing period T_z
- H_s and peak period T_p
- H_s, T_p and peakedness parameter γ

The first two alternative formats are different from the third in one respect. When you describe a spectrum in terms of H_s and T_z or H_s and T_p, then Flexcom uses special algorithms to select appropriate values for f_p, α and γ. These algorithms are summarised shortly. Thereafter, Flexcom operates in exactly the same way as if the standard or default option had been invoked.

When the third (H_s, T_p and γ) format is invoked, a special algorithm (to be outline shortly) is used to calculate a value for α. The actual spectrum values are calculated using Goda's approximation (Goda, 1979), as follows:

$$S(f) = \alpha\, H_s^2 \frac{f_p^4}{f^5}\, e^{\left[-\frac{5}{4}\left(\frac{f}{f_p}\right)^{-4}\right]}\, \gamma^{FAC} \qquad (3)$$

H_s/T_z Format

For the H_s/T_z combination, the method adopted here is one due to Isherwood (1987), who publishes in his paper of 1987 a revised Jonswap spectrum parameterisation based on empirical data published by Houmb and Overvik (1976). This parameterisation will not be described in detail here; the interested reader is referred instead to the publications listed in the references. However the procedure is outlined here for completeness.

Isherwood first defines an equivalent wave steepness s as:

$$s = \frac{2\pi H_s}{g T_z^2} \qquad (4)$$

where g is the gravitational constant. The value of γ is then found from:

$$\gamma = 10.54 - 1.34 s^{-\frac{1}{2}} - \exp(-19 + 3.775 s^{-\frac{1}{2}}) \quad for\, s \geq 0.037$$
$$\gamma = 0.9 + \exp(18.86 - 3.67 s^{-\frac{1}{2}}) \qquad for\, s < 0.037 \qquad (5)$$

The value of α is next calculated from the relation:

$$\frac{\alpha}{s^2} = 2.964 + 0.4788\gamma^{\frac{1}{2}} - 0.3430\gamma + 0.04225\gamma^{\frac{3}{2}} \qquad (6)$$

Finally the value of the peak frequency f_p is found from:

$$f_p T_z = 0.6063 + 0.1164\gamma^{\frac{1}{2}} - 0.01224\gamma \qquad (7)$$

H_s/T_p Format

For the H_s/T_p combination, Flexcom first categorises the seastate based on the value of the parameter (T_p /√ H_s). Values for γ and α are then calculated according to the table below; f_p is simply $1/T_p$.

Parameter	Regime		
	Windsea Regime (Tp /√ Hs) < 3.6	Jonswap Range 3.6 < (Tp /√ Hs) < 5	Swell Regime (Tp /√ Hs) > 5
α	2.73 Hs^2 / Tp^4	0.036 – 0.0056(Tp /√ Hs)	5.07 Hs^2 / Tp^4

γ	5	exp{5.75 – 1.15(Tp /√ Hs)}	1

Calculation of Jonswap Parameters

The numerical values in the table above are actually appropriate only for H_s in metres. If you specify a Jonswap spectrum in H_s/T_p format, Flexcom checks the value you specify for the gravitational constant g. If g is in the range 32 < g < 33, then the program assumes that your data is in Imperial units, and your value for H_s is converted internally to Metric before the table calculations. If your value for g is outside of this range, H_s is assumed to be in metres, and the table above's formulae are applied without modification.

$H_s/T_p/\gamma$ Format

For the $H_s/T_p/\gamma$ combination, Flexcom uses this formula to calculate α.

$$\alpha = \frac{0.0624}{0.230 + 0.0336\gamma - 0.185(1.9 + \gamma)^{-1}} \qquad (8)$$

Equation 3 above is then used for the wave spectrum values.

RELEVANT KEYWORDS

- `*WAVE-JONSWAP` is used to specify a JONSWAP random sea wave spectrum or spectra.

Ochi Hubble Wave

THEORY

Ochi and Hubble (1976) developed a six parameter wave spectrum model consisting of two parts, one for the lower frequency components of the wave energy and the other covering the higher frequency components. Each component is expressed in terms of three parameters and the total spectrum is written as a linear combination of the two. This means that double peaks present in a wave energy density can be modelled, representing for example a (low frequency) swell along with (high frequency) wind-generated waves. The spectrum represents almost all stages of development of a sea in a storm. The Ochi-Hubble spectrum is based on the following formulation:

$$S(\omega)=\frac{1}{4}\sum_{j=1}^{2}\frac{\left(\frac{4\lambda_j+1}{4}\omega_{0j}^{\ 4}\right)^{\lambda_j}}{\Gamma(\lambda_j)}\frac{H_{sj}^{\ 2}}{\omega^{4\lambda_j+1}}\exp\left[-\left(\frac{4\lambda_j+1}{4}\right)\left(\frac{\omega_{0j}}{\omega}\right)^4\right] \qquad (1)$$

where the modal frequency, w_{0j}, is related to the periods (peak and mean zero up-crossing) by the following equation:

$$\omega_{0j}=\left(\frac{2\pi}{T_{pj}}\right)=\left(\frac{4}{5\pi}\right)^{1/4}\left(\frac{2\pi}{T_{zj}}\right) \qquad (2)$$

and:

- H_{S1} is the significant wave height for the lower frequency components
- T_{P1} is the peak period for the lower frequency components
- T_{Z1} is the mean zero up-crossing period for the lower frequency components
- λ_1 is the shape factor for the lower frequency components
- H_{S2} is the significant wave height for the higher frequency components
- T_{P2} is the peak period for the higher frequency components
- T_{Z2} is the mean zero up-crossing period for the higher frequency components

- λ_2 is the shape factor for the higher frequency components

RELEVANT KEYWORDS

- *WAVE-OCHI-HUBBLE is used to specify an Ochi-Hubble random sea wave spectrum or spectra.

Torsethaugen Wave

THEORY

Torsethaugen and Haver (2004) developed a double peak spectral model which represents wave conditions in open ocean areas where the waves are dominated by local wind sea, but also exposed to swell. For the locally fully developed sea, the spectral peak period is a function of the significant wave height, $T_{pf}=a_f*H_s^{1/3}$, with a_f slightly dependent on fetch. This classifies sea states as wind sea $T_p<=T_{pf}$ and swell $T_p>T_{pf}$ each consisting of two wave systems. Empirical parameters given by significant wave height and spectral peak period define the spectral form parameters and energy distribution between the two wave systems in the model. The adequacy of the model is verified by comparing with measured wave spectra from the Norwegian Continental shelf. The Torsethaugen spectrum is based on the following formulation:

$$S(f_n)=\sum_{j=1}^{2} E_j S_{jn}(f_{jn}) \qquad (1)$$

$$E_1=\left(\frac{1}{16}\right)H_1^2 T_{p1} \quad \text{and} \quad E_2=\left(\frac{1}{16}\right)H_2^2 T_{p2} \qquad (2)$$

$$S_{1n}(f_{1n})=G_0 A_\gamma f_{1n}^{-4} e^{-f_{1n}^{-4}} \gamma^{\left(\exp-\left(\frac{1}{2\sigma^2}\right)(f_{1n}-1)^2\right)} \quad \text{and} \quad S_{2n}(f_{2n})=G_0 f_{2n}^{-4} e^{-f_{2n}^{-4}} \qquad (3)$$

where:

- j = 1 for the primary sea system, j = 2 for the secondary sea system
- $f_{1n}=f\times T_{p1}$
- $f_{2n}=f\times T_{p2}$

- $G_0 = 3.26$
- $A_\gamma = \left(\frac{1 + 1.1(\ln \gamma)^{1.19}}{\gamma} \right)$
- $\sigma = 0.07$ for $f_n < 1$ and $\sigma = 0.09$ for $f_n > 1$

For a wind dominated sea, parameters for the primary and secondary peaks are defined as follows:

Primary Peak:

$$H_{s1} = H_{sw} = r_{pw} H_s$$

$$T_{p1} = T_{pw} = T_p$$

$$\gamma = 35 \left[\frac{2\pi}{g} \frac{H_{sw}}{T_p^2} \right]^{0.857} \qquad (4)$$

Secondary Peak:

$$H_{s2} = H_{ssw} = \sqrt{1 - r_{pw}^2} H_s$$

$$T_{p2} = T_{psw} = T_{pf} + 2$$

$$\gamma = 1 \qquad (5)$$

where:

$$r_{pw} = 0.7 + 0.3 \exp\left(- \left(2 \frac{T_{pf} - T_p}{T_{pf} - 2\sqrt{H_s}} \right)^2 \right) \qquad (6)$$

For a swell dominated sea, parameters for primary and secondary peaks are defined as:

Primary Peak:

$$H_{s1} = H_{ssw} = r_{ps} H_s$$

$$T_{p1} = T_{psw} = T_p$$

$$\gamma = 35 \left[\frac{2\pi}{g} \frac{H_s}{T_{pf}^2} \right]^{0.857} \left(1 + 6 \frac{T_p - T_{pf}}{25 - T_{pf}} \right) \qquad (7)$$

Secondary Peak:

$$H_{s2} = H_{sw} = \sqrt{1 - r_{ps}^2} H_s$$

$$T_{p2} = T_{pw} = 6.6 H_{sw}^{1/3}$$

$$\gamma = 1 \qquad (8)$$

where:

$$r_{ps} = 0.6 + 0.4 \exp\left(- \left(\frac{T_p - T_{pf}}{0.3(25 - T_{pf})} \right)^2 \right) \qquad (9)$$

RELEVANT KEYWORDS

- *WAVE-TORSETHAUGEN is used to specify a Torsethaugen random sea wave spectrum or spectra.

User-Defined Wave Spectrum

THEORY

If you wish to model a wave spectrum which is not well represented by any of the predefined wave spectra, Flexcom provides a facility whereby you can input an arbitrary wave spectrum directly in terms of a series of (frequency, spectral ordinate) data pairs. This feature is intended to provide complete generality in terms of defining a wave spectrum.

RELEVANT KEYWORDS

- *WAVE-USER-DEFINED is used to specify a User-Defined random sea wave spectrum or spectra.

Time History of Water Surface Elevation

THEORY

You may also define a random seastate in terms of a time history of water surface elevation. When this option is invoked, Flexcom first calculates a wave spectrum from the time history. A record with a fixed time step is required for this; if your record has a variable time step, then Flexcom synthesises a record with a fixed step by interpolation. You have the option of specifying the fixed time step to use in this process, or of letting Flexcom choose a suitable value. The output from this process is then treated as a series of regular waves, each with its individual frequency, amplitude and phase.

In order for the realised wave elevation history to be in exact agreement with the input wave history defined in the data file, a very large number of harmonics are required (n/2 in fact, where n is the number of data lines in the wave elevation time history). Naturally the subsequent dynamic analysis can be quite time consuming if it contains thousands of harmonics. A more efficient approach is to divide the input time history into a number of shorter time histories using the Ensembles entry. For example, using ten ensembles rather than one will result in approximately a 90% reduction in the number of harmonics used, and this has obvious benefits in terms of the overall run time. Where the number of ensembles is greater than one, the Fourier Transform process is repeated at regular intervals over the course of the dynamic analysis, with each region having its own unique combination of regular wave harmonics.

RELEVANT KEYWORDS

- *WAVE-TIME-HISTORY is used to specify a random seastate in terms of a time history of water surface elevation.

Advanced Topics

This section contains information on:

- Water Surface Elevation discusses how the water surface elevation is found at any point in the wave field.

- Water Particle Velocities and Accelerations provides details on how these terms are defined for any point in the wave field.
- Spectrum Discretisation discusses the available methods for discretising a wave spectrum into its component harmonics.
- Random Seed discusses the assignment of random phase angles to component harmonics of a wave spectrum.
- Wave Energy Spreading discusses how wave energy is distributed between different directions in a multi-directional random sea.
- Wave Kinematics discusses the modifications to linear Airy wave theory for waves of finite amplitude.
- Equivalent Regular Waves describes a facility to model a random sea as an equivalent regular wave.
- Selected Frequencies describes a frequency domain option to request that specific frequencies be included in the range of solution harmonics.

Water Surface Elevation

This section discusses how the water surface elevation is found at any point in the wave field.

As described in Spectrum Discretisation, a random sea is discretised into component harmonics. The output from this process is essentially a series of regular waves, each with its individual period, amplitude and phase. For a multi-directional random sea, the resulting water surface elevation at a point in the wave field is found as a superposition of these wave components over all wave directions and is given by the equation:

$$\eta(y,z,t)=\sum_{m=1}^{M}\sum_{n=1}^{N}a_{mn}\cos(k_n s_m-\omega_n t+\phi_{mn})$$

where:

- a_{mn} is the wave amplitude in mth direction of nth harmonic
- k_n is the wave number of nth harmonic

- s_m is the horizontal distance in m[th] direction from vertical axis Y=Z=0 to point in question $\left(s_m = y\cos\theta_m + z\sin\theta_m\right)$
- θ_m is the angle of m[th] wave direction measured anticlockwise relative to global Y
- ϕ_{mn} is the random phase for m[th] wave direction and n[th] harmonic
- N is the number of harmonics
- M is the number of wave directions

For a uni-directional random sea, obviously $M=1$, and the above Equation becomes:

$$\eta(y,z,t)=\sum_{n=1}^{N} a_n \cos(k_n s - \omega_n t + \phi_n)$$

For a uni-directional random sea, the coefficients a_n are found from the wave spectrum $S_\eta(\omega)$ using the relation:

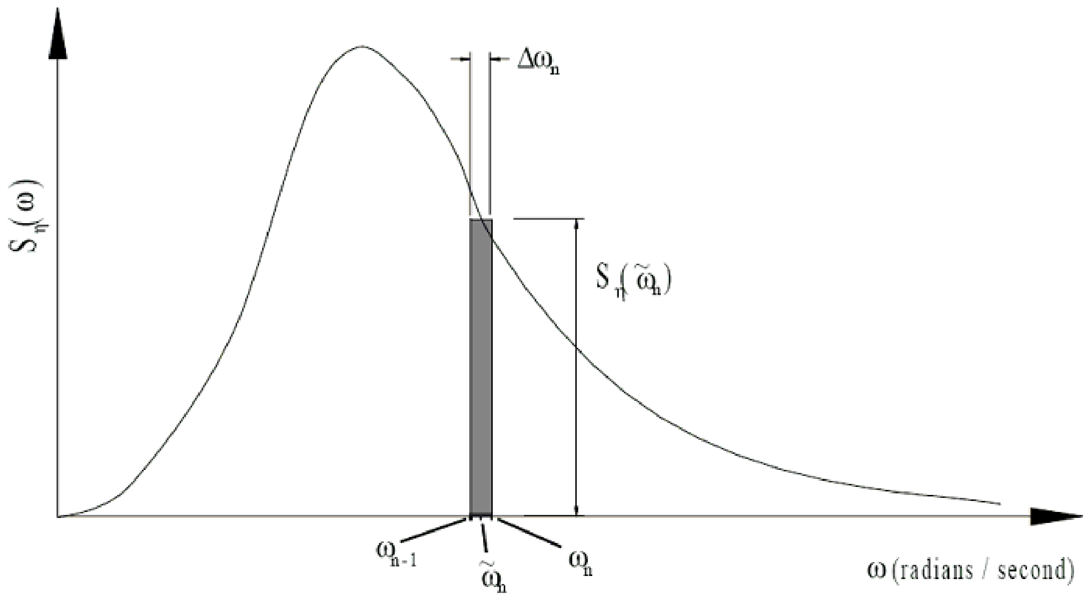
$$a_n = \sqrt{2S_\eta(\varpi_n)\Delta\omega_n}$$

where:

$$\varpi_n = \frac{\omega_n + \omega_{n-1}}{2}$$

$$\Delta\omega_n = \omega_n - \omega_{n-1}$$

The product $S_\eta(\varpi_n)\Delta\omega_n$ is an increment of the area under the spectrum centred on $\omega = \varpi_n$, as shown in the below figure.



Spectrum Area Increment

The wave amplitudes a_{mn} for a multi-directional random sea are related to the uni-directional values a_n as follows:

$$a_{mn} = \sqrt{a_n \int_{\theta_{rm}-\frac{\delta\theta}{2}}^{\theta_{rm}+\frac{\delta\theta}{2}} P(\theta_r)\,d\theta_r}$$

where:

- $P(\theta_r)$ a spreading function used to distribute wave energy about the dominant direction, as described earlier in the ‘Wave Energy Spreading’ section.
- θ_{rm} is the mth wave direction relative to dominant wave direction
- θ_r is the direction relative to dominant wave direction

Water Particle Velocities and Accelerations

THEORY

Fluid velocities and accelerations due to the random wave field are computed at element integration points using similar expressions to Regular Airy Wave (Eq.7 - 10), extended for multi-directional waves as appropriate. The modifications to Airy wave theory represented by the Wave Kinematics (Eq.1), the Wave Kinematics (Eq.2) and the Wave Kinematics (Eq.3) are used for random seas also, with superposition stretching again the default.

A computational economy is also applied here without any significant loss of accuracy. As the water particle velocities and accelerations decay exponentially with depth, for computational efficiency only those wave harmonics which are in the wave zone are considered to contribute towards the generation of water particle velocities and accelerations. The wave zone is considered to extend from the mean water line downwards by a distance of one wavelength times a specified Wavelength Factor. A value of 0.5 is assumed by default, so any harmonic whose half-wavelength is less than the distance from the mean water line to the integration point in question is omitted from water particle velocity and acceleration calculations.

Morison's Equation assumes that the force exerted by unbroken waves on a cylinder can be represented by a linear sum of drag and inertial terms. The formulation is widely established for modelling wave forces on slender offshore structures such as risers and mooring lines. In situations where the body size becomes significant with respect to wavelength, the underlying assumptions become invalid, and the effects of radiation and diffraction must also be considered. So in Flexcom's internal computations, any harmonic whose half-wavelength is less than the element diameter is omitted from the water particle velocity and acceleration calculations.

RELEVANT KEYWORDS

- `*WAVE-GENERAL` is used to specify miscellaneous parameters to wave loading. Specifically, the `WAVEZONE=` input is used to specify the extent of the wave zone, used in the computation of water particle velocities and accelerations.

- `*PRINT` is used to request additional printed output to the main output file. Specifically, the `OUTPUT=WATER PARTICLE HYDRODYNAMICS` option provides detailed output regarding the spatial and temporal distributions of water particle velocity and acceleration.

Spectrum Discretisation

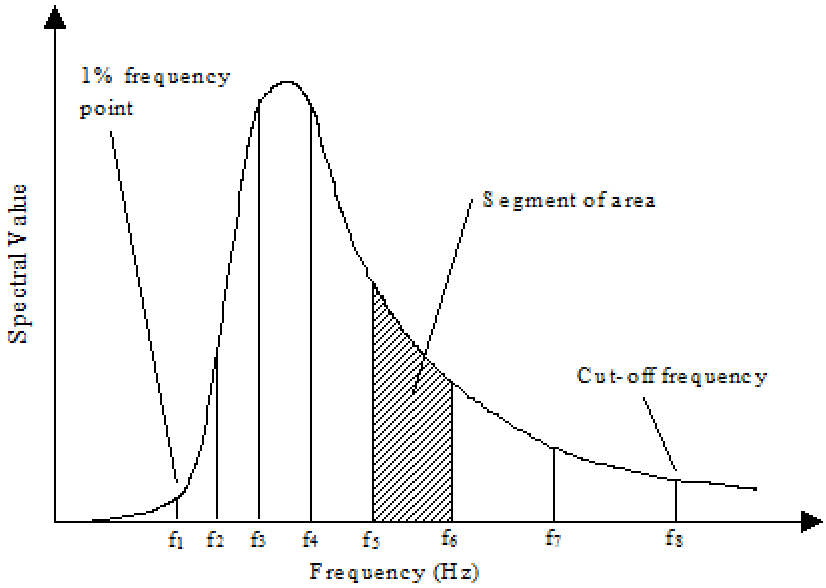
THEORY

In a random sea analysis, Flexcom discretises the spectrum (or spectra) you specify into discrete harmonics, and uses these to generate a time history of random surface elevation using a Monte Carlo simulation technique. This provides the basic dynamic random excitation on the structure. The spectral discretisation procedure may be based on equal area increments of the wave spectrum or on a geometric progression of wave frequencies.

Note also that where an analysis contains two or more wave spectra, each spectrum is discretised and applied independently. In other words, Flexcom does not compile a single equivalent/composite spectrum before the discretisation process begins.

With an equal area discretisation, the random wave spectrum is discretised into component harmonics by dividing the area under the spectrum into segments of equal area. The area of each segment is the total area under the spectrum divided by a user-specified number of harmonics, and each area then defines a single harmonic in the discretisation. This procedure has the disadvantage that it leads to increasingly larger frequency increments at higher frequencies and so to a poor representation of higher harmonics in the wave elevation time series. There is a maximum frequency increment input which may be used to counteract this. If a particular segment requires a frequency increment greater than the specified maximum, the segment area is continuously halved until the required increment is less than the maximum. In general, this leads to a greater number of harmonics being returned by the discretisation than the requested number.

With a geometric progression discretisation, the wave spectrum is discretised into segments based on frequency increments that form a geometric progression. The procedure adopted is as follows. The total area under the wave spectrum up to a user-specified cut-off frequency is found, and the frequency that corresponds to the first 1% of the area under the wave spectrum is identified. Next, this frequency is multiplied by a constant factor to give the next frequency in the series. The area between these two frequencies is taken as the first discrete segment of the wave spectrum. The next frequency in the series is found by again multiplying the last frequency by the constant factor, and the area between these two frequencies is taken as the next discrete segment of the wave spectrum. This procedure is repeated until the frequency corresponding to the first 99% of the area under the wave spectrum is reached. The constant factor which multiplies successive frequencies is governed by a user input, as outlined in the below Equation.



Geometric Progression Wave Spectrum Discretisation

The above figure shows an example of geometric progression wave spectrum discretisation. Each segment of area of the wave spectrum is defined by the frequencies at the start and end of the segment, f_n and f_{n+1} as follows:

$$f_{n+1} = f_n(1+r) \quad (1)$$

where r is the user-specified geometric progression factor. Each segment of area defines a single harmonic in the discretisation, with the frequency of the harmonic for a particular segment given by:

$$f = \tfrac{1}{2}(f_n + f_{n+1}) \quad (2)$$

Flexcom uses a default value of 0.02 for the geometric progression factor, which is sufficient in most cases. The smaller the value used, the greater will be the number of harmonics in the resulting discretisation (and the longer will be the repeat time of the resulting wave elevation time series for time domain analysis). Note that the first figure above is illustrative only, generally a much larger number of harmonics should be produced by the wave spectrum discretisation.

RELEVANT KEYWORDS

Wave spectra may be defined using any of the following keywords:

- *WAVE-PIERSON-MOSKOWITZ is used to specify a Pierson-Moskowitz random sea wave spectrum or spectra.
- *WAVE-JONSWAP is used to specify a JONSWAP random sea wave spectrum or spectra.
- *WAVE-OCHI-HUBBLE is used to specify an Ochi-Hubble random sea wave spectrum or spectra.
- *WAVE-TORSETHAUGEN is used to specify a Torsethaugen random sea wave spectrum or spectra.
- *WAVE-TIME-HISTORY is used to specify a random seastate in terms of a time history of water surface elevation.
- *WAVE-USER-DEFINED is used to specify a User-Defined random sea wave spectrum or spectra.

Taking the Pierson-Moskowitz spectrum as an example, the FREQUENCY=AREA input is used to request equal area discretisation, while the FREQUENCY=GP input is used to request geometric progression discretisation.

- *PRINT is used to request additional printed output to the main output file. Specifically, the OUTPUT=WAVE DISCRETISATION option is used to request the wave discretisation details. When the option is invoked, the printed output might look like the following:

Sample Wave Discretisation Output

```
                    *** ADDITIONAL PRINTED OUTPUT ***
                    ---------------------------------
Wave Discretisation Data:
Harmonic      Omega       Amplitude      Phase       Direction    Wave Number    Wav
             (rad/s)                     (deg)         (deg)
-------------------------------------------------------------------------------------
    1       0.306371     0.176732      357.736       10.000       0.009930        6
    2       0.328848     0.176732      348.678       10.000       0.011265        5
    3       0.340204     0.176732      303.390       10.000       0.011990        5
    4       0.348588     0.176732       76.948       10.000       0.012547        5
etc...
  101       1.872890     0.088366      119.858       10.000       0.357566
  102       2.159313     0.062484      239.289       10.000       0.475294
  103       2.349804     0.062484      116.444       10.000       0.562852
```

Random Seed

THEORY

Flexcom uses a random number generator to assign phase values to harmonics in a discretised wave spectrum or spectra. Unless you specify otherwise, Flexcom always uses the same seed value, so that every time you discretise the same spectrum with the same discretisation parameters, you get the same sequence of random phases and consequently the same time history of water surface elevation. You may want to change this, so that you get a different set of phases and so a different wave elevation time history from the same spectrum and discretisation parameters (other than the seed). This can be done by specifying a different seed value.

Flexcom uses a Multiplicative Congruential Generator to generate a sequence of random phases. For best results, the random number seed for this should be a large (and preferably prime) number. You may enter your own value for the random number seed. Alternatively, you can select a seed value from a prepared list which Flexcom provides.

Specifying a random number seed is optional. If you do not specify a value, Flexcom uses the first value in the list by default.

Naturally, invoking this option is meaningful only in the case of a time domain dynamic analysis with a random wave spectrum or spectra. If your analysis includes two spectra, any seed value you input will be used in discretising both.

RELEVANT KEYWORDS

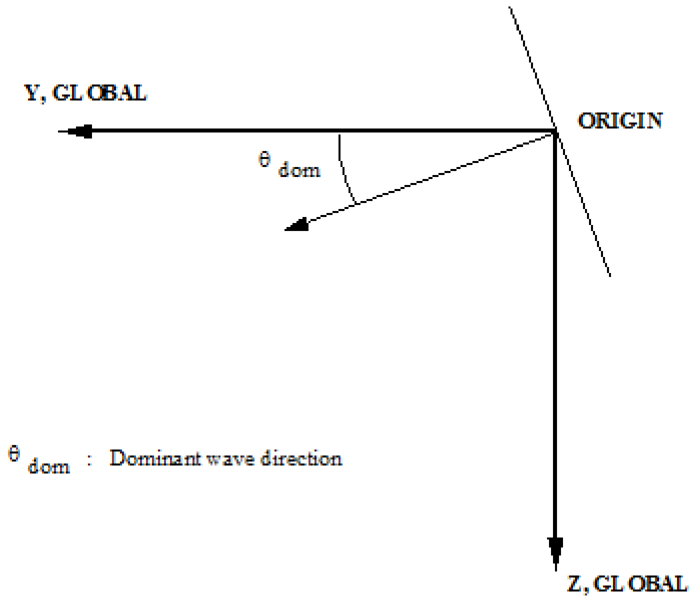
- `*WAVE-GENERAL` is used to specify miscellaneous parameters to wave loading. Specifically, the `SEED=` input is used to specify a seed value for the random number generator that assigns random phases to wave components when discretising a wave spectrum.

Wave Energy Spreading

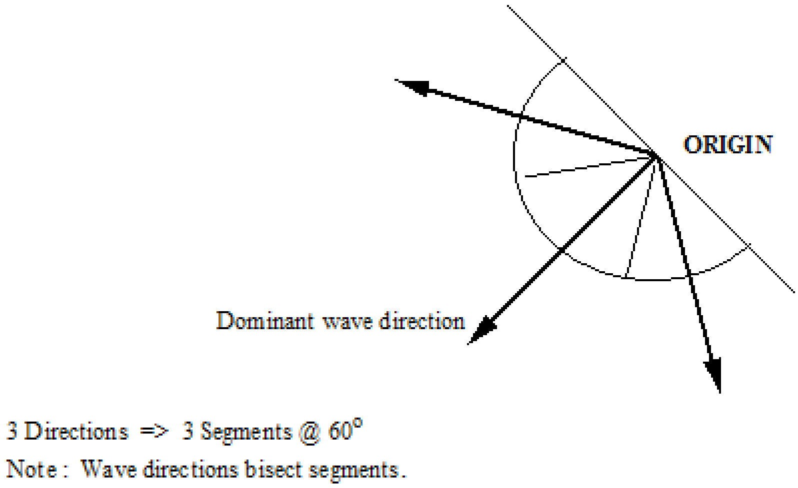
THEORY

A multi-directional random sea is defined in terms of a dominant wave direction and the number of wave directions. The dominant wave direction is the direction at which most of the wave energy is concentrated. A definition of the dominant wave direction, denoted θ_{dom} is shown in the first figure below. If the number of specified directions is denoted n, then Flexcom divides a semi-circular arc between (θ_{dom}- 90°) and (θ_{dom}+ 90°) into n equal segments, as shown for example in the second figure below, where 3 wave directions are specified. The vector centred on the origin and bisecting the segment defines a wave direction corresponding to each segment. Note that if the number of wave directions is even, this process will not result in a wave direction coincident with the dominant, so an odd value is recommended.

Flexcom uses a cosine^n spreading function in distributing wave energy between directions in a multi-directional random sea, where n is a user-defined wave spreading exponent. The spreading exponent must be an even integer and defaults to 2, giving a cosine^2 spreading function.



Dominant Wave Direction



Wave Spreading Directions

A mathematical description of a directional seastate is feasible by assuming that the sea state can be considered as a superposition of a large number of regular sinusoidal wave components with different frequencies and directions. With this assumption, the representation of a spectrum in frequency and direction becomes a direct extension of the frequency spectrum alone, allowing the use of Fast Fourier Transform method. It is often convenient to express the wave spectrumE(f,θ) describing the angular distribution of wave energy at respective frequencies by:

$$E(f,\theta) = E(f)G(f,\theta)$$

where the function G(f,θ) is a dimensionless quantity, and is known as the directional spreading function. Other acronyms for G(f,θ) are the spreading function, angular distribution function and the directional distribution.

The one-dimensional spectra may be obtained by integrating the associated directional spectra over θ as:

$$E(f) = \int_{-\pi}^{\pi} E(f,\theta)d\theta$$

Therefore, from the above last two equations, G(f,θ) must satisfy:

$$\int_{-\pi}^{\pi} G(f,\theta)d\theta = 1$$

The functional form of G(f,θ) has no universal shape and several proposed formulas are available.

In the most convenient simplification of G(f,θ), it is customary to consider G to be independent of frequency f, such that we have:

$$G(\theta) = \frac{2}{\pi} cos^2\theta, \text{ for } |\theta| < \frac{\pi}{2}$$

The above can be generalised to:

$$G(\theta) = C(n)cos^n(\theta - \theta_0) \text{ where}$$

$$C(n) = \frac{1}{\sqrt{\pi}} \frac{\Gamma\left(\frac{n}{2}+1\right)}{\Gamma\left(\frac{n}{2}+\frac{1}{2}\right)} \text{ and}$$

- θ_0 is the dominant (principal) direction for the spectrum
- *n* is the wave spreading exponent that determines the peakedness of the directional spreading
- *C(n)* is a constant satisfying the normalization condition
- *θ* is a counterclockwise measured angle from the principal wave direction
- Γ is the Gamma function

If you plot the G function with respect to the angle, you will look at a bell graph. The peak and mean of this bell graph will be at the dominant direction.

The standard referring the wave spreading exponent is DNV RP-C205-2010 - environmental conditions and loads. This is recommended practice only, the formulation above is purely mathematical.

Variations in wave spreading exponent

If you increase the spreading exponent then you are focusing most of the energy around the dominant direction and less towards the peripheral directions. In Flexcom, the directional spreading function G does not depend on frequency f. As a result, after the discretisation of the specified spectrum, the amplitude of each harmonic is scaled with the G function based on the angle. The scaling is repeated for each wave direction you requested. So, if a spectrum is discretised with 100 harmonics and you requested 3 directions you should get 300 harmonics in your load case.

RELEVANT KEYWORDS

Wave spectra may be defined using any of the following keywords:

- *WAVE-PIERSON-MOSKOWITZ is used to specify a Pierson-Moskowitz random sea wave spectrum or spectra.
- *WAVE-JONSWAP is used to specify a JONSWAP random sea wave spectrum or spectra.
- *WAVE-OCHI-HUBBLE is used to specify an Ochi-Hubble random sea wave spectrum or spectra.
- *WAVE-TORSETHAUGEN is used to specify a Torsethaugen random sea wave spectrum or spectra.
- *WAVE-TIME-HISTORY is used to specify a random seastate in terms of a time history of water surface elevation.
- *WAVE-USER-DEFINED is used to specify a User-Defined random sea wave spectrum or spectra.

Each keyword has an associated Wave Spreading Exponent input, which defines the exponent used in distributing wave energy between directions in a multi-directional random sea.

Wave Kinematics

THEORY

Linear Airy wave theory is strictly valid only for infinitesimal wave heights. A number of modifications have been proposed to modify Airy wave theory for finite amplitude waves, and three of these are incorporated into Flexcom. The wave kinematics algorithms available are:

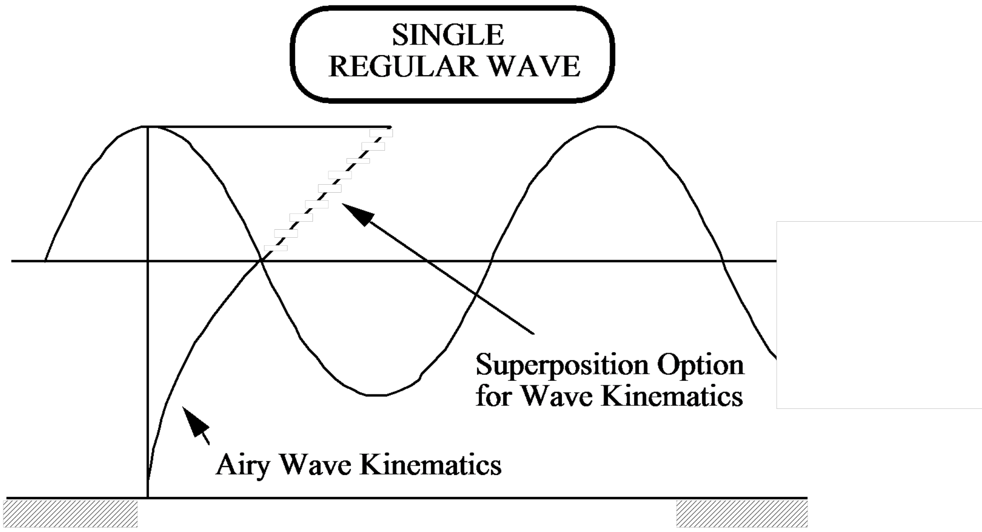
- Superposition Stretching (the default)
- MWL Values Above MWL
- Extend MWL to Wave Surface

The theoretical basis of each for the algorithms is as follows. Consider x to be the vertical coordinate of a point in the fluid. The Airy wave theory equations are modified by replacing x by $x^* = \beta x$ in the appropriate expressions for horizontal and vertical water particle velocities and accelerations. Let the wave surface elevation at any time be denoted by η. Typically η is a superposition of individual harmonics η_i, $\left(\eta = \sum \eta_i\right)$.

For the (default) Superposition Stretching approach, the parameter b is computed as:

$$\beta = \frac{d+\eta_i}{d+\eta}, \quad 0 \le x \le d+\eta \qquad (1)$$

where d is the water depth. Note that for a single regular wave, β = 1. This means that the Airy wave equations are simply extrapolated above the MWL – see the above Equation.



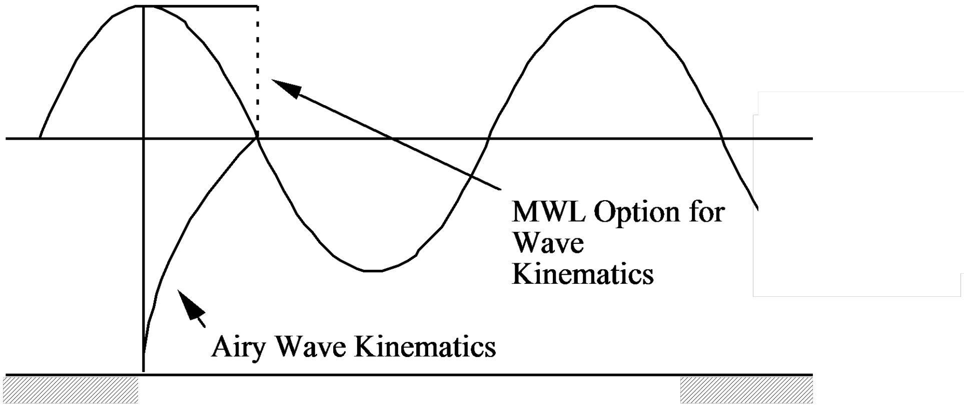
Superposition Stretching

For the MWL Values Above MWL approach, β is computed as:

$$\beta = 1, \quad 0 \le x \le d$$

$$x^* = d, \quad d < x \le \eta \qquad (2)$$

This means that wave kinematics above the MWL are set to the value at the MWL – see the below figure.

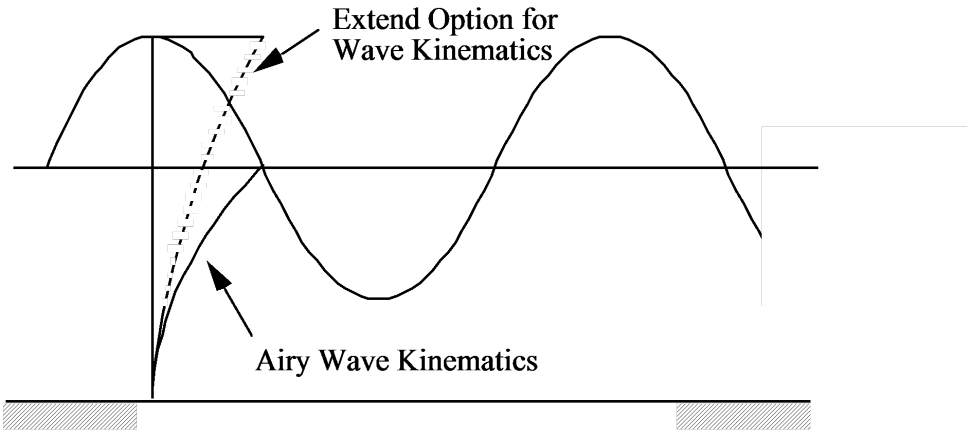


MWL Values Above MWL

For the Extend MWL to Wave Surface approach, the parameter β is computed as:

$$\beta = \frac{d}{d+\eta} \qquad (3)$$

This means that each wave component is extended from the MWL to the free surface – see the figure below.



Extend MWL to Wave Surface

It must be emphasised that there is no compelling theoretical justification for any of the three kinematics options, and there is no sense in which one is more correct than the other. The choice between them is a matter of engineering judgement. In general, the default option results in the largest wave forces, and can be considered the most conservative.

RELEVANT KEYWORDS

- *WAVE-GENERAL is used to specify miscellaneous parameters to wave loading. Specifically, the OPTION= input is used to specify the algorithm to be used by Flexcom in calculating Airy wave kinematics.

Equivalent Regular Waves

THEORY

Flexcom includes a capability to model a random sea as an equivalent regular wave. If you invoke this option, the approach adopted is as follows. The random sea is first discretised into component harmonics in the usual way, and equivalent regular wave properties are calculated from this discretisation. The equivalent regular wave height is got by multiplying the random sea significant wave height H_s (which is calculated from the moments of the discretised spectrum) by a user-specified factor. The equivalent regular wave period is got by multiplying the period at which the maximum value of the spectrum occurs (that is, the spectrum peak period) by a user-specified factor. Both factors (for wave period and wave height) are optional entries and suitable default values are provided (1.86 and 0.95, respectively) if not explicitly specified.

The program then calculates a response spectrum for each attached node in all degrees of freedom, by combining the wave spectrum with the vessel RAOs you input. From each response spectrum, sinusoidal boundary condition amplitudes are calculated in a similar manner to the above, and BC phase angles are estimated from the vessel RAOs. The period of the sinusoidal BCs is the wave period calculated above. These sinusoidal boundary conditions are then applied at the attached nodes in a regular wave dynamic analysis with the equivalent regular wave.

This facility can be invoked if the analysis includes two wave spectra. Each spectrum is handled separately using the procedure outlined above. In this case the dynamic analysis will have two independent regular waves, and sinusoidal BCs from the two spectra will be superposed or added together to define the motions of the attached node(s).

RELEVANT KEYWORDS

- `*REGULAR WAVE EQUIVALENT` is used to specify that Flexcom is to replace a random wave spectrum or spectra by an equivalent regular wave or waves, and to calculate equivalent sinusoidal boundary conditions for attached nodes.

Selected Frequencies

THEORY

The procedure for discretising a random sea into component harmonics (described earlier in Spectrum Discretisation) in the frequency domain is identical to that performed in the time domain. However, there is one additional option available in the frequency domain – you may optionally specify selected (exact) frequencies which Flexcom must include in the range of solution harmonics. This feature can potentially be quite important in terms of ensuring the accuracy of a frequency domain simulation. It might typically be invoked to guarantee that Flexcom includes a solution at any natural frequency of the structure which occurs in the range of harmonics. Unlike time domain simulations, structural excitation at a natural frequency may not be captured in the frequency unless one of the component harmonics corresponds exactly to the natural frequency. In terms of the discretisation itself, Flexcom searches for the harmonic nearest the selected frequency, sets the component frequency equal to the specified frequency, and adjusts the adjacent segment areas appropriately.

RELEVANT KEYWORDS

- `*WAVE-GENERAL` is used to specify miscellaneous parameters to wave loading. Specifically, the `FREQUENCIES=` input is used to specify selected frequencies that Flexcom must include in the range of solution harmonics.

1.9.3.4 Additional Loading Options

OVERVIEW

In addition to the standard loading described in preceding sections (e.g. wave, current etc.), Flexcom also provides a range of other useful options for applying loads to a model. The various options are discussed in the following sections:

- Point and Distributed Loads outlines point and distributed loading options.

- User-Defined Forces describes the user-subroutine load option facility, which provides for complete generality in terms of load application.
- Temperature Loading outlines the temperature loading model.
- Vessel and Harmonic Loads outlines these frequency-dependent load options.

RELEVANT KEYWORDS

- `*LOAD` is used to define arbitrary loading.
- `*TEMPERATURE` is used to apply thermal loading.

Point and Distributed Loads

THEORY

In addition to the standard loading described in preceding sections (e.g. wave, current etc.), Flexcom also provides a range to other useful options for applying loads to a model. This section outlines point and distributed loading options.

Point and distributed loads are largely self-explanatory. Point loads are specified in terms of a single point of application (node), direction and magnitude. Uniformly distributed loads are specified in terms of an area of application (typically a group of consecutive elements, defined in terms of a start element and an end element). The magnitude of the loading may be constant or time-dependent. Both point and distributed loads are input as components in the global coordinate directions – this means that a force oblique to the coordinate axes is specified as three load components.

RELEVANT KEYWORDS

- `*LOAD` is used to define arbitrary loading. Specifically, the `TYPE=CONSTANT` input is used to define point loads, while the `TYPE=DISTRIBUTED` input is used to define distributed loads.

User-Defined Forces

OVERVIEW

The user-subroutine load option facilitates both time varying point loads and uniformly distributed loads. This option is available for complete generality, particularly for situations where the standard loading options do not completely cover user requirements.

Any generic compiler may be used to compile a DLL (dynamic link library) which may be linked into Flexcom. When you invoke a user subroutine option, you must have written the relevant source code, and compiled it into a DLL, before you perform the actual Flexcom analysis. Note also that there are standard templates provided in the Flexcom installation folder which illustrate how to create DLLs.

NOTE

This feature has to a large extent been superseded by the User Solver Variables capability, which provides much greater flexibility and control. So unless you have an existing Flexcom model which utilises this specific feature, you can skip reading this section and proceed directly to User Solver Variables.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Operation
- Subroutine Format
- Sample User-Subroutine Load

RELEVANT KEYWORDS

- `*LOAD` is used to define arbitrary loading. Specifically, the `TYPE=SUBROUTINE` input is used to define arbitrary user-defined loading.

Operation

When you specify the name of the DLL file, Flexcom loads the DLL and searches for the load subroutine that was embedded into the DLL when it was compiled. Once Flexcom locates the subroutine, the subroutine is dynamically loaded into the Flexcom analysis module executable. The analysis then proceeds. When running, at each time step the user-subroutine is called once for each individual point load and each individual uniformly distributed load you specify. The subroutine arguments contain sufficient information for you to be able to determine which combination the subroutine is being called for (presuming there is more than one), and you must program your module accordingly, typically using statements such as “if”, “then”, “else” etc. The sample coding in Sample User-Subroutine Load illustrates this in greater depth. Finally, when the analysis is completed the temporary Flexcom executable is deleted. Note that if you can use the facility, then the operation is completely automatic and transparent to you, and requires no intervention by you at runtime in compiling or linking the program.

Subroutine Format

A blank or template FORTRAN listing of subroutine usrlod, the Flexcom load user-subroutine, is shown below. The format of the user-subroutine file is now detailed with reference to this table. Obviously, experienced FORTRAN users will be aware that the comment lines are optional, and serve no function other than to make the code more comprehensible. Other than the comments, there are some lines that this user-subroutine must contain. These are:

- `the subroutine` statement
- `the implicit none` statement
- `the integer and real(8)` statements
- `the !dec$ attributes` statements to make the variables and subroutine available to Flexcom
- `the end subroutine` statement

The subroutine statement has eight arguments, to be described shortly. The `implicit none` statement requires that all variables used in the routine be declared explicitly. The `integer and real(8)` statements are used to declare integer and double precision variables. It is important to note that all calculations in Flexcom use double precision arithmetic, so you should always use the double precision form of any FORTRAN intrinsic functions you invoke (for example, use dcos rather than cos for the cosine of an angle). The `!dec$ attributes` statements are used to make the variables and subroutine available to Flexcom. Finally, the use of the end subroutine statement signals the end of the subroutine coding.

User-Subroutine Template

```
! Subroutine usrlod -- to prescribe time-dependent loads.
! Do not modify the following lines.
subroutine usrlod(key, node, lsta, lend, ndof, time, ramp, value)
!dec$ attributes dllexport, stdcall, reference :: usrlod
  implicit none

  integer, intent(in) :: key
  !dec$ attributes reference :: key
  integer, intent(in) :: node
  !dec$ attributes reference :: node
  integer, intent(in) :: lsta
  !dec$ attributes reference :: lsta
  integer, intent(in) :: lend
  !dec$ attributes reference :: lend
  integer, intent(in) :: ndof
  !dec$ attributes reference :: ndof
  real(8), intent(in) :: time
  !dec$ attributes reference :: time
  real(8), intent(in) :: ramp
  !dec$ attributes reference :: ramp
  real(8), intent(inout) :: value
  !dec$ attributes reference :: value
!
! Variable Names
!
!     key     : Key to indicate point load (key = 1) or udl (key = 2).
!     node    : The number of the node for which usrlod is being called,
!               for key = 1; node is 0 for key = 2.
!     lsta    : The number of the start element for which usrlod is being
!             : called to prescribe a udl, for key = 2; lsta is 0 for
!             : key = 1.
!     lend    : The number of the endt element for which usrlod is being
!             : called to prescribe a udl, for key = 2; lend is 0 for
!             : key = 1.
!     ndof    : The number of the degree of freedom (dof) for which usrlod
!               is being called.
```

```
!      time     : The present simulation time.
!      ramp     : The current value of the ramp being applied to loads and
!                 displacements.
!      value    : The value of the load to be applied for this time value
!                 (user-specified).
!
! Declare local variables.


! Insert coding to define user-specified loading below this line.



! Do not alter the next line.
end subroutine usrlod
```

The eight arguments of subroutine `usrlod` are listed in the comments statements above. Of these, the first seven - the key, the node number, the start element, the end element, the degree of freedom, the present simulation time and the current value of the ramp - are passed to `usrlod` for information only, to be used in calculating the user-defined load value. The values of these variables should not be altered. The eighth variable is the one to which the user-defined load value is to be assigned. The coding to define the value of value should be inserted after the Insert coding comment, but before the end subroutine statement, as indicated in the table above and illustrated further in the next section.

All applied loads and displacements in Flexcom are optionally ramped or gradually increased up to the full values over a time period (the ramp time) you define. Further information is provided in Load Ramping. However, with regard to user-programmed load values, it is important to note that a ramp is not applied to the value you assign to the variable value after the return from usrlod. If you want this value to be gradually ramped on, then the responsibility for applying the ramp is yours and you should do so in the user-subroutine coding. This generality is necessary, since you have no way of indicating to the program whether the application of a ramp is appropriate or not. Refer to Sample User-Subroutine Load in the next section for more details.

Sample User-Subroutine Load

In a particular analysis, it is required to invoke the user-subroutine option to define point loads at two nodes (say 20 and 30) in degrees of freedom 2 and 3 respectively. It is also necessary to define a uniformly distributed load along a section of the structure (say Elements 36, 37 and 38) in DOF 2. The various loads are all sinusoidal, of period 10 seconds, but the magnitudes and phases relative to the analysis regular wave are different for each load, as defined in the first table below. It must be stressed here of course that these values are entirely notional, and do not represent any reasonable condition, or indeed a situation that is ever likely to occur. A subroutine usrlod to apply these displacements is shown in the second table below. Note the use of `if ... then ...` else statements to vary the load magnitude and phase.

Sample Amplitude and Phase Values

Load	Node	Element	DOF	Amplitude (N)	Phase (deg)
Point	20	---	2	500.	90.
Point	30	---	3	100.	-90.
Distributed	---	36, 37, 38	2	400.	45.

Sample Load User-Subroutine

```
! Subroutine usrlod -- to prescribe time-dependent loads.
! Do not modify the following lines.
subroutine usrlod(key, node, lsta, lend, ndof, time, ramp, value)
!dec$ attributes dllexport, stdcall, reference :: usrlod
  implicit none

  integer, intent(in) :: key
  !dec$ attributes reference :: key
  integer, intent(in) :: node
  !dec$ attributes reference :: node
  integer, intent(in) :: lsta
  !dec$ attributes reference :: lsta
  integer, intent(in) :: lend
  !dec$ attributes reference :: lend
  integer, intent(in) :: ndof
```

```
  !dec$ attributes reference :: ndof
  real(8), intent(in) :: time
  !dec$ attributes reference :: time
  real(8), intent(in) :: ramp
  !dec$ attributes reference :: ramp
  real(8), intent(inout) :: value
  !dec$ attributes reference :: value
!
! Variable Names
!
!     key     : Key to indicate point load (key = 1) or udl (key = 2).
!     node    : The number of the node for which usrlod is being called,
!               for key = 1; node is 0 for key = 2.
!     lsta    : The number of the start element for which usrlod is being
!             : called to prescribe a udl, for key = 2; lsta is 0 for
!             : key = 1.
!     lend    : The number of the endt element for which usrlod is being
!             : called to prescribe a udl, for key = 2; lend is 0 for
!             : key = 1.
!     ndof    : The number of the degree of freedom (dof) for which usrlod
!               is being called.
!     time    : The present simulation time.
!     ramp    : The current value of the ramp being applied to loads and
!               displacements.
!     value   : The value of the load to be applied for this time value
!               (user-specified).
!
! Declare local variables.
  real(8) :: period
  real(8) :: pi
  real(8) :: factor
  real(8) :: omega
  real(8) :: amag
  real(8) :: phase

! Insert coding to define user-specified loading below this line.

  period = 10.d0
  pi     = 4.d0 * datan(1.d0)
  factor = 180.d0 / pi
  omega  = 2.d0 * pi / period
  amag   = 0.d0
  phase  = 0.d0

  if( key == 1 )then
    if( node == 20 )then
      amag  = 500.d0
      phase = 90.d0
    else if( node == 30 )then
      amag  = 100.d0
      phase = -90.d0
    end if
  else if( key == 2 )then
```

```
      amag  = 400.d0
      phase = 45.d0
    end if

    phase = phase / factor

    value = amag * dcos(-omega*time + phase) * ramp

  ! Do not alter the next line.
  end subroutine usrlod
```

Temperature Loading

THEORY

A simple temperature loading model is available based on a specified thermal expansion coefficient (α) and a temperature variation (ΔT). The axial strain, ε_A , of an element subjected to thermal loading is defined as:

$$\varepsilon_A = \alpha.\Delta T \qquad (1)$$

So the applied axial force due to thermal expansion, F_T, is defined as:

$$F_T = EA.\alpha.\Delta T \qquad (2)$$

where EA is the axial stiffness of the element.

RELEVANT KEYWORDS

- `*TEMPERATURE` is used to apply thermal loading.

If you would like to see an example of how this keyword is used in practice, refer to F02 - Upheaval Buckling.

Vessel and Harmonic Loads

THEORY

Vessel and harmonic loads are only applicable to frequency domain analysis.

Vessel loads are used to specify frequency-dependent loads, for example, the variation in top tension of a riser connected to a TLP. They are specified in terms of a point of application (node), direction and a load RAO definition. Vessel loads can be used with constant point loads – in which case the constant load (applied at all frequencies) is added to the vessel load at each particular frequency.

Harmonics loads define sinusoidally varying point loads, and are only applicable in a frequency domain regular wave analysis. They are specified in terms of a point of application (node), direction, amplitude and phase. The harmonic load has the same period as the analysis regular wave, or waves in the case of many regular wave analyses.

RELEVANT KEYWORDS

- `*LOAD` is used to define arbitrary loading. Specifically, the `TYPE=VESSEL` input is used to define vessel loads, while the `TYPE=HARMONIC` input is used to define harmonic loads.

1.9.3.5 Boundary Conditions

OVERVIEW

Flexcom presents a comprehensive range of options to fully describe the constraints which are applied to the finite element model. The various options for the application of boundary conditions are discussed in the following sections:

- Application of Rotational Constraints discusses the significance of rotational degrees of freedom in terms of boundary conditions.
- Constant Boundary Conditions outlines standard time invariant constraints.
- Vessel Boundary Conditions how vessel motions may be applied to attached nodes.
- Sinusoidal Boundary Conditions describes a facility to apply motions which vary sinusoidally with time.
- Harmonic Boundary Conditions describes a facility to apply motions which vary harmonically in a frequency domain regular wave analysis.
- Arbitrary Boundary Conditions describes the user-subroutine boundary condition facility, which provides for complete generality in terms of constraint application.

* Timetrace Boundary Conditions outlines how arbitrary time-varying boundary conditions may be applied via motion time histories.

The specification of boundary conditions is optional, and by default no constraints are applied. This allows free-falling objects, for example, to be analysed. However, most problems will require some boundary conditions to be specified.

RELEVANT KEYWORDS

* `*BOUNDARY` is used to define boundary conditions.

Application of Rotational Constraints

This section discusses the significance of rotational degrees of freedom in terms of boundary conditions.

When defining boundary conditions, a non-zero displacement term may be optionally associated with a constraint. This allows an offset to be modelled in conjunction with a restraint. Such displacements terms have a straightforward definition in a translational sense, but if specifying displacements in degrees of freedom (DOFs) 4-6, is important if specifying displacements in DOFs 4-6 to understand the significance of these DOFs.

You will no doubt be aware that finite (non-infinitesimal) 3D rotations cannot be represented as vectors, since the addition of such vectors is non-commutative, that is, the order in which the rotations are taken affects the result. In Flexcom this problem is handled by means of a consistent 3D kinematics formulation based on the correspondence between a specially defined rotation vector and a transformation matrix. The full details are omitted here but, in summary, a rotation is represented by a vector such that i) the magnitude of the vector represents the magnitude of the rotation, and ii) the direction of the vector represents the axis of rotation. Therefore the inputs in the Displacement column for DOFs 4-6 are the components of this rotation vector. It is particularly important to realise that in general they do not represent individual rotations about the global coordinate X, Y and Z axes. Refer to Global Degrees of Freedom for further information.

Specification of appropriate boundary conditions in DOFs 4, 5 & 6 requires a knowledge of the local undeformed and desired convected axis systems for the constrained element. Armed with this knowledge, it should be possible to compute relevant boundary conditions given a desired hangoff angle. However, in order to expedite this process and to minimise effort on the part of the user, a standalone program has been created to perform the required computations. Refer to our Downloads page if you require access to this program.

Constant Boundary Conditions

THEORY

Constant boundary conditions are used to define time invariant constraints. They are defined in terms of a node number (or series of numbers), degree of freedom and (optionally) displacement. An option is provided to specify whether the boundary condition is absolute or relative. By default (absolute), the node is fixed with respect to its initial position. Alternatively, the relative option means the node is fixed with respect to its position at the end of the preceding analysis, rather than the initial position.

RELEVANT KEYWORDS

- `*BOUNDARY` is used to define boundary conditions. Specifically, the `TYPE=CONSTANT` input is used to define constant or time invariant boundary conditions.

If you would like to see an example of how this keyword is used in practice, refer to any of the standard Flexcom Examples.

Vessel Boundary Conditions

THEORY

Vessel boundary conditions are used to define the motions of nodes which are attached to a vessel (or vessels). They are defined in terms of a vessel name, node number, degree of freedom and (optionally) displacement. A full discussion of the calculation of vessel motions and corresponding structural displacements is given in Vessels and Vessel Motions.

RELEVANT KEYWORDS

- *BOUNDARY is used to define boundary conditions. Specifically, the TYPE=VESSEL input is used to identify those nodes on the structure whose motions are defined from the motions of an attached vessel.

If you would like to see an example of how this keyword is used in practice, refer to any of the standard Flexcom Examples.

Sinusoidal Boundary Conditions

THEORY

Sinusoidal boundary conditions are used to apply motions which vary sinusoidally with time. They are defined in terms of a node number, degree of freedom, amplitude, phase and period. The last three parameters refer to the sinusoidal function defining the motion. The phase angle is defined relative to the wave datum. If a single regular wave is used in the analysis, the sinusoidal period defaults to the period of the regular wave.

Where structure displacements are specified using the sinusoidal boundary condition option, any other boundary condition may also be applied at the same degree of freedom and at the same node. This can be used, for example, to combine vessel drift (specified via sinusoidal boundary conditions), with high frequency vessel motions (specified using vessel boundary conditions).

This sinusoidal boundary condition option was provided originally for modelling the effect of vessel drift. However more comprehensive and accurate drift modelling capabilities are provided in more recent versions of Flexcom, and these have effectively superseded the sinusoidal boundary condition option. However, they are still retained for compatibility with earlier versions, and you are of course perfectly entitled to use them as required.

RELEVANT KEYWORDS

- *BOUNDARY is used to define boundary conditions. Specifically, the TYPE=SINUSOIDAL input is used to identify those nodes on the structure whose boundary conditions vary sinusoidally with time.

Harmonic Boundary Conditions

THEORY

Harmonic boundary conditions are used where the motions of nodes on the structure vary harmonically in a frequency domain regular wave analysis. They are defined in terms of a node number, degree of freedom, (optionally) static displacement, amplitude and phase. This option can be used in a regular wave analysis with one or many waves. The period of the motion is equal to the period of the regular wave. In an analysis with many regular waves, this means the period of the motion varies in the range of the analysis wave periods. The static displacement determines the mean displacement about which the harmonic motion takes place. So for example, if the preceding static analysis has a corresponding constant boundary condition with non-zero displacement, the static displacement should be consistent with this value. The phase angle is relative to the wave at the static offset position. By convention, a positive phase angle implies a phase lag, whereas a negative phase angle implies a phase lead.

RELEVANT KEYWORDS

- `*BOUNDARY` is used to define boundary conditions. Specifically, the `TYPE=HARMONIC` input is used to identify nodes on the structure whose motions vary harmonically in a frequency domain regular wave analysis.

Arbitrary Boundary Conditions

THEORY

The user-subroutine boundary condition option facilitates the definition of boundary conditions which vary with time. This option is available for complete generality, particularly for situations where the standard boundary condition options do not completely cover user requirements.

The operation of user-defined subroutines was restructured internally recently, which means that having an Intel Visual Fortran Compiler installed on your computer is no longer mandatory. Any generic compiler may be used to compile a DLL (dynamic link library) which may be linked into Flexcom. When you invoke a user subroutine option, you must have written the relevant source code, and compiled it into a DLL, before you perform the actual Flexcom analysis. This differs from earlier versions of Flexcom, where the input was in the form of a FORTRAN source code file. Note also that there are standard templates provided in the Flexcom installation folder which illustrate how to create DLLs.

Further information on this topic is contained in the following sections:

- Operation
- Subroutine Format
- Application
- Other Points to Note
- Example

RELEVANT KEYWORDS

- `*BOUNDARY` is used to define boundary conditions. Specifically, the `TYPE=SUBROUTINE` input is used to identify nodes on the structure whose boundary conditions are defined via a user-subroutine.

Operation

When you specify the name of the DLL file, Flexcom loads the DLL and searches for the boundary condition subroutine. Once Flexcom locates the subroutine, the subroutine is dynamically loaded into the Flexcom analysis module executable. The analysis then proceeds. When running, the user-subroutine is called once at each solution time for each node and DOF you specified as having an arbitrary boundary condition applied. The subroutine arguments contain sufficient information for you to be able to determine which combination the subroutine is being called for (presuming there is more than one), and you must program your module accordingly, typically using statements such as “if”, “then”, “else” etc. Finally, when the analysis is completed the temporary Flexcom executable is deleted. Note that if you invoke this user-subroutine facility, then the operation is completely automatic and transparent to you, and requires no intervention by you at runtime in compiling or linking the program.

Subroutine Format

A blank or template FORTRAN listing of subroutine usrdsp, the Flexcom boundary condition user-subroutine, is shown in the user-subroutine template below. The format of the user-subroutine file is now detailed with reference to this table. Obviously, experienced FORTRAN users will be aware that the comment lines are optional, and serve no function other than to make the code more comprehensible. Other than the comments, there are some lines that this user-subroutine must contain. These are:

- `the subroutine statement`
- `the implicit none statement`
- `the integer and real(8) statements`
- `the !dec$ attributes` statements to make the variables and subroutine available to Flexcom
- `the end subroutine statement`

The subroutine statement has eight arguments, to be described shortly. The implicit none statement requires that all variables used in the routine be declared explicitly. The integer and real(8) statements are used to declare integer and double precision variables. It is important to note that all calculations in Flexcom use double precision arithmetic, so you should always use the double precision form of any FORTRAN intrinsic functions you invoke (for example, use dcos rather than cos for the cosine of an angle). The !dec$ attributes statements are used to make the variables and subroutine available to Flexcom. Finally, the use of the end subroutine statement signals the end of the subroutine coding.

User-Subroutine Template

```
! Subroutine usrdsp -- to prescribe time-dependent displacements.
! Do not modify the following lines.
subroutine usrdsp(node, ndof, time, ramp, dispx, dispy, dispz, disp)
!dec$ attributes dllexport, stdcall, reference :: usrdsp
  implicit none

  integer, intent(in) :: node
  !dec$ attributes reference :: node
  integer, intent(in) :: ndof
  !dec$ attributes reference :: ndof
  real(8), intent(in) :: time
  !dec$ attributes reference :: time
  real(8), intent(in) :: ramp
  !dec$ attributes reference :: ramp
  real(8), intent(in) :: dispx
  !dec$ attributes reference :: dispx
  real(8), intent(in) :: dispy
  !dec$ attributes reference :: dispy
  real(8), intent(in) :: dispz
  !dec$ attributes reference :: dispz
  real(8), intent(inout) :: disp
  !dec$ attributes reference :: disp

!
! Variable Names
!
!     node    : The number of the node for which usrdsp is being called.
!     ndof    : The number of the degree of freedom (dof) for which usrdsp
!               is being called.
!     time    : The present simulation time.
!     ramp    : The current value of the ramp being applied to loads and
!               displacements.
!
!     dispx   : The displacement value currently in the array of nodal
!               displacements for the vertical dof at this node.
!               If heave motions have been specified at this node, dispx will
!               contain the heave calculated at this time.
!     dispy   : The displacement value currently in the array of nodal
```

```
!                 displacements for dof 2 at this node.
!      dispz    : The displacement value currently in the array of nodal
!                 displacements for dof 3 at this node.
!
!      disp     : The displacement value to be assigned to this node and dof
!                 for this time value (user-specified).
!
! Declare local variables.


! Insert coding to define user-specified motion below this line.



! Do not alter the next line.
end subroutine usrdsp
```

The eight arguments of subroutine `usrdsp` are described in the comments statements of the above table. Of these, the first seven - the node number, the degree of freedom, the present simulation time, the current value of the ramp and the current position of the node - are passed to `usrdsp` for information only, to be used in calculating the user-defined displacement value. The values of these variables should not be altered. The eighth variable is the one to which the user-defined displacement value is to be assigned. The coding to define the value of `disp` should be inserted after the Insert coding comment, but before the end subroutine statement, as indicated in the above table and illustrated further in the next section.

It is important to note that the variable `disp` is a displacement, that is a motion from or about the user-specified initial coordinate. The actual position in space of the node at any time is the sum of the initial coordinate and the displacement value in `disp`. If the node in question is specified as being a cable interior node, then the initial nodal coordinate is the location calculated by the cable pre-static step. This location is echoed to the output file jobname.out for user scrutiny. Invoking the user-subroutine option to define the subsequent motion of such a node would constitute an unusual step.

The values of `dispx, dispy and dispz` are the displacements of the node from the initial coordinates in the global X, Y and Z directions respectively. For the case of any of these displacements which is not determined by the motions of an attached vessel, the value of the variable which is passed to subroutine `usrdsp` is the value of that displacement at the last solution time. So for example, if the present solution time is 12.1s and the previous solution was at 12s, and if the vertical motion of the node in question is not determined from the motions of an attached vessel; then the value of `dispx` when subroutine usrdsp is called at 12.1s is the value of the vertical displacement of the node at 12s.

What this means is that the values of `dispx, dispy and dispz` can be used in calculating the value of `disp`, which will then in the normal course of events become the next value (the value at the present solution time) of one of these `(dispx, dispy or dispz)`. Of course in very many analyses, the value to be assigned to disp will be independent of `dispx, dispy and dispz`, and the values of these variables will be immaterial and unused in your FORTRAN coding.

Of greater significance is the situation where one or other (or indeed all) of `dispx, dispy and dispz` are defined by the motions of an attached vessel. In this case the value(s) of these variables will be the values calculated from the vessel motions at the current solution time. So for example, if the present solution time is 12.1s, and the motions in the global Y and Z directions of the node for which the subroutine is called are determined from the motions of an attached vessel; then the values of `dispy` and `dispz` when subroutine `usrdsp` is called at 12.1s are the values of the displacements of the node in these degrees of freedom at this solution time.

Application

Displacements calculated in the user-subroutine can be added to or superimposed on displacements calculated from vessel motions. The major application for which this facility is intended is in modelling displacements due to TLP setdown. Consider, for example, a TLP tether subjected to 2D environmental conditions. The vertical motion of the top of the tether is due to two effects, the vertical motion of the TLP itself (heave) and the setdown due to the horizontal TLP motion (surge). In defining the boundary conditions for an analysis of this tether, you would specify that both the vertical and horizontal motion of the topmost tether node were to be calculated from vessel (TLP) motions via RAOs and phase angles. Then you would further specify a user-subroutine to be invoked for the vertical direction at the node. In the user-subroutine itself the setdown due to the instantaneous value of dispy would be calculated and added to dispx to give the total vertical motion of the node. This procedure is illustrated in the TLP Setdown Example to follow.

Other Points to Note

Incidentally, it is worth pointing out that the situation where you specify both a boundary condition from vessel motion and a boundary condition from a user-subroutine is one of only two situations where you can combine two types of boundary condition at the same node and DOF. For example, it is not valid to combine a constant boundary condition and a vessel boundary condition, or a constant boundary condition and a user-subroutine boundary condition. The only valid combinations at the same node and DOF are, as outlined above, combining a vessel boundary condition and a user-subroutine boundary condition, or combining a sinusoidal boundary condition with any other boundary condition.

All applied loads and displacements in Flexcom are optionally ramped or gradually increased up to the full values over a time period (the ramp time) you define. Further information on load ramping is provided in Load Ramping. However, with regard to user-programmed boundary conditions, it is important to note that a ramp is not applied to the value you assign to the variable `disp` after the return from `usrdsp`. If you want the value of `disp` to be gradually ramped on, then the responsibility for applying the ramp is yours and you should do so in the user-subroutine coding. This generality is necessary, since you have no way of indicating to the program whether the application of a ramp is appropriate or not.

Note that the values of `dispx, dispy and dispz` will already have been ramped prior to the call to subroutine `usrdsp`, so any value calculated on the basis of the current values of these variables will automatically include the effect of the ramp. This is illustrated in the example to follow..

TLP Setdown Example

In an analysis of a TLP tether, it is required to invoke the user-subroutine option to superpose the motion due to TLP setdown on the motion due to TLP heave. The user-subroutine facility is required because of course the TLP setdown is very much a time dependent quantity, since it is a function of the instantaneous TLP surge. The actual equation defining the magnitude of the setdown is:

$$S(t)=L-\sqrt{L^2-Y_{TLP}^2(t)} \qquad (1)$$

where:

- S(t) is the setdown value at time t

- L is the tether length
- $Y_{TLP}(t)$ is the TLP surge at time t

The example user subroutine is shown in the table below.

Example User-Subroutine

```
! Subroutine usrdsp -- to prescribe time-dependent displacements.
! Do not modify the following lines.
subroutine usrdsp(node, ndof, time, ramp, dispx, dispy, dispz, disp)
!dec$ attributes dllexport, stdcall, reference :: usrdsp
  implicit none

  integer, intent(in) :: node
  !dec$ attributes reference :: node
  integer, intent(in) :: ndof
  !dec$ attributes reference :: ndof
  real(8), intent(in) :: time
  !dec$ attributes reference :: time
  real(8), intent(in) :: ramp
  !dec$ attributes reference :: ramp
  real(8), intent(in) :: dispx
  !dec$ attributes reference :: dispx
  real(8), intent(in) :: dispy
  !dec$ attributes reference :: dispy
  real(8), intent(in) :: dispz
  !dec$ attributes reference :: dispz
  real(8), intent(inout) :: disp
  !dec$ attributes reference :: disp

!
! Variable Names
!
!     node    : The number of the node for which usrdsp is being called.
!     ndof    : The number of the degree of freedom (dof) for which usrdsp
!               is being called.
!     time    : The present simulation time.
!     ramp    : The current value of the ramp being applied to loads and
!               displacements.
!
!     dispx   : The displacement value currently in the array of nodal
!               displacements for the vertical dof at this node.
!               If heave motions have been specified at this node, dispx will
!               contain the heave calculated at this time.
!     dispy   : The displacement value currently in the array of nodal
!               displacements for dof 2 at this node.
!     dispz   : The displacement value currently in the array of nodal
!               displacements for dof 3 at this node.
!
```

```
!     disp    : The displacement value to be assigned to this node and dof
!               for this time value (user-specified).
!
! Declare local variables.
  real(8) :: setdown

! Insert coding to define user-specified motion below this line.

  setdown = 315.d0 - dsqrt(315.d0**2 - dispy**2)
  disp    = dispx - setdown

  ! Notes:  (i)   Setdown based on tether length of 315m
  !         (ii)  Setdown superposed on motion due to heave
  !         (iii) dispx, dispy and dispz already have ramp applied,
  !               so do not apply ramp to setdown

! Do not alter the next line.
end subroutine usrdsp
```

Timetrace Boundary Conditions

Timetrace boundary conditions fall into two separate categories in Flexcom:

- Displacement
- Reference point

Displacement Boundary Conditions

THEORY

Displacement boundary conditions allow you to specify arbitrary time-varying boundary conditions at specific nodes of the FE model, where the time history of motion is read from an ASCII data file. The position of the node is then calculated by adding this displacement to the initial position of the node. Flexcom uses cubic spline interpolation to find displacement values at time points intermediate to those specified in the data file (for this reason the analysis solution times do not need to match those in the data file). For analysis times before the earliest time specified in the data file, Flexcom uses the values at that earliest time until the analysis time exceeds this value. Similarly, for analysis times after the latest time in the data file, Flexcom uses the values at that latest time.

The format that this data file takes depends on the number of boundary conditions that are specified in the file, but in general the file contains one column of data for the time and an additional column for each boundary condition. Comment lines, denoted by a capital 'C' in the first column, are permitted, while lines that are completely blank are ignored. An example data file is shown below.

```
C
C Time          Node 51          Node 51           Node 101
C               DOF1             DOF2              DOF1
C
  0.0             0.0              0.0               6.0
  0.2             0.0              0.569             5.960
  0.4             0.0              1.132             5.844
  0.6             0.0              1.680             5.651
  0.8             0.0              2.206             5.384
  1.0             0.0              2.703             5.047
  1.2             0.0              3.165             4.644
  1.4             0.0              3.585             4.181
  1.6             0.0              3.959             3.663
  1.8             0.0              4.281             3.098
  2.0             0.0              4.548             2.492
  2.2             0.0              4.755             1.854
  2.4             0.0              4.900             1.191
  2.6             0.0              4.981             0.513
  2.8             0.0              4.997            -0.171
  3.0             0.0              4.949            -0.853
```

The first column of data must always contain time values. Subsequent columns then contain the boundary condition timetrace data. Flexcom uses cubic spline interpolation to find displacement values at time points intermediate to those specified in the data file (for this reason the analysis solution times do not need to match those in the data file).

Note that the order in which the boundary conditions are specified in the *Timetrace Boundary – Displacement* table must correspond to the order in which the relevant data appears in the data file. For example, the three boundary conditions in the file shown above would be specified as shown below.

Displacement Files	Node	DOF
Displacement.dat	51	1
	51	2

	101	1

Columns of data in the ASCII data file should be separated by blank spaces or tabs. You can specify as many boundary conditions as you wish with a single data file – simply leave the first column of the *Timetrace Boundary – Displacement* table blank for the second and subsequent boundary conditions. The only limitation on this is that each line in the ASCII data file should not exceed 200 characters in length.

RELEVANT KEYWORDS

- `*BOUNDARY` is used to define boundary conditions. Specifically, the `TYPE=FILE` input is used to identify the nodes on the structure whose time-varying displacements are to be read from an ASCII data file.

Reference Point Boundary Conditions

THEORY

Reference point boundary conditions are used where the motion of a point or points on the structure is defined by the motions of a point not on the structure known as a reference point. These boundary conditions are analogous to vessel boundary conditions, with the exception that the motion of the reference point is read in this case from an ASCII data file, rather than being determined from RAOs and ambient wave elevation. Effectively what happens during an analysis is as follows. The location and orientation of the reference point is read from the ASCII data file at the start of the analysis. The static displacement (this is an optional entry associated with reference point boundary conditions), if present, is applied at the boundary node(s). Any subsequent change in the location and orientation of the reference point from the position at the start of the analysis is applied to the relevant node and DOF as if a rigid link existed between the reference point and the location of the node after static displacement. Flexcom uses cubic spline interpolation to find the location and orientation of the reference point at time points between those specified in the data file (for this reason the analysis solution times do not need to match those in the data file). Note that the reference point need not be a point on the structure – in general it is more likely to be, say, the centre of gravity of an attached floating structure. For analysis times before the earliest time in the data file, Flexcom uses the location and orientation of the reference point at that earliest time. Similarly, for analysis times after the latest time in the data file, Flexcom uses the location and orientation of the reference point at that latest time.

The ASCII data file contains seven columns of data. The first column contains time data, and the remaining six correspond to the six degrees of freedom defining the position and orientation of the reference point. Comment lines, denoted by a capital 'C' in the first column, are permitted, while lines that are completely blank are ignored. An example data file is shown below.

```
C
C Time    DOF1     DOF2    DOF3    DOF4    DOF5     DOF6
C
  0.0    140.0    150.0    0.0     0.0     0.0     0.0
  0.2    140.0    150.0    0.0     0.0     0.0     3.419
  0.4    140.0    150.0    0.0     0.0     0.0     6.794
  0.6    140.0    150.0    0.0     0.0     0.0    10.081
  0.8    140.0    150.0    0.0     0.0     0.0    13.236
  1.0    140.0    150.0    0.0     0.0     0.0    16.219
  1.2    140.0    150.0    0.0     0.0     0.0    18.990
  1.4    140.0    150.0    0.0     0.0     0.0    21.513
  1.6    140.0    150.0    0.0     0.0     0.0    23.757
  1.8    140.0    150.0    0.0     0.0     0.0    25.690
```

The first column of data contains time values. Columns 2 – 4 contain the global X, Y and Z coordinates of the reference point, while Columns 5 – 7 contain the components of the rotation vector (in degrees) describing the orientation of the reference point. These columns correspond to degrees of freedom 4 – 6 of the Flexcom coordinate system.

Flexcom provides a number of options for specifying vessel motions to be read from timetrace files, which mostly operate in similar fashion to this facility. Indeed the enhanced vessel motion options could be considered to supersede this earlier feature, which is retained though for compatibility with previous program versions. You are of course perfectly entitled to continue using the reference point boundary capability as required.

RELEVANT KEYWORDS

- `*BOUNDARY` is used to define boundary conditions. Specifically, the `TYPE=REFERENCE` input is used to identify the nodes on the structure whose time-varying motions are to be calculated from the motion of a so-called reference point.

1.9.3.6 Wake Interference

OVERVIEW

This section describes the wave interference modelling facility provided by Flexcom. Three wake formulations are available, namely Huse's formulation, Blevins' formulation and a user-defined wake model for complete generality. This section introduces the range of wave interference modelling options.

In wake analysis, the wake of an upstream structure has the potential to affect the current forces experienced by a downstream structure. A typical application would be in the analysis of a pair of top tensioned risers or a pair of jumpers operating in close proximity. You model the upstream and downstream structures in the normal fashion, and also indicate to Flexcom that you want to model wake interference. Three options are provided for calculating the wake of an upstream structure, namely Huse's formulation, Blevins' formulation and an option for a user-defined wake model, which allows you define drag and lift coefficients for the downstream structure in terms of a grid covering the upstream wake region. VIV effects may also be considered, as Flexcom may automatically perform a modal analysis using Modes, and computed enhanced drag coefficients using Shear7, as part of the wake interference analysis process. The output from a wake interference analysis is essentially the clearance between the upstream and downstream structures.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Operation describes the typical sequence of steps in a wake interference analysis.
- Huses Model outlines the wake interference formulation presented by Huse.
- Blevins Model outlines the wake interference formulation proposed by Blevins.
- User-Defined Wake Model describes the general purpose wake model provided by Flexcom.
- VIV Drag explains the option to amplify drag coefficients, based on the results of a Shear7 analysis, which allows you to model the increase in drag that VIV typically causes.

- VIV Effects explains the automatic interface to Shear7, which allows you to iteratively perform a VIV analysis of the downstream structure as part of the overall wake interference computations. This produces enhanced drag coefficients may be more accurately determined based on the reduced current profile.

- Summary of Scenarios presents a sample list of possible wake interference scenarios and analysis sequences.

LIMITATIONS

- Wake interference effects are only considered when constant hydrodynamic coefficients are specified for the downstream structure. If you specify Reynolds number dependent coefficients, then the effects of wake interference are ignored, and the simulation proceeds without considering any shielding effects. The wake interference model in Flexcom is designed for consistent current flow only (i.e. where the current profile does not vary over time). During the iteration process, the current velocities experienced by the downstream structure can vary due to shielding effects. Hence Reynolds number dependency is not considered as this would add another layer of complexity to the solution process, and potentially conflicting scenarios could occur during the iteration process.

- Wake interference effects are considered for static load cases only (although it is possible to perform a simulation quasi-statically if this is necessary to assist solution convergence). Flexcom is unable to simulate wake interference effects for regular wave or random sea simulations as the wake interference theories implemented in the software are only applicable to established flow regimes. From a theoretical perspective, it is much more difficult to quantify the effects of wake interference in a dynamic context so there are no immediate plans for further development of Flexcom in this area.

RELEVANT KEYWORDS

- `*WAKE UPSTREAM` is used to specify the composition of the upstream structure in terms of element sets, for use in wake interference calculations. The keyword also selects the wake interference model to be used, and defines associated data to characterise the wake field.

- *WAKE DOWNSTREAM is used to specify the composition of the downstream structure in terms of element sets for use in wake interference calculations. The keyword also facilitates the (optional) definition of lift coefficients in the case of a *User-Defined* wake interference model.
- *VIV DRAG is used to instruct Flexcom to read vortex-induced vibration (VIV) drag coefficient amplification factors from the results of a Shear7 analysis.
- *VIV EFFECTS is used to instruct Flexcom to continuously run Modes/Shear7 analyses of the downstream structure during the wake interference analysis.
- *CURRENT is used to specify current loading.

If you would like to see an example of how these keywords are used in practice, refer to A04 - TTR Wake Interference or G02 - Jumper WakeInterference.

Note also that the old *WAKE INTERFERENCE and *UPSTREAM STRUCTURE keywords have effectively been superseded by the new wake interference keywords which are more generic and provide additional functionality.

Operation

THEORY

When you want to perform a wake interference analysis with Flexcom, you model the upstream and downstream structures in the normal fashion, and indicate to Flexcom that you want to model wake interference by specifying appropriate wake interference data.

The two structures may be defined in separate models or combined in a single model (note that is not possible to model a situation where one structure is downstream of another and upstream of a third). Using separate models is appropriate where the two structures are independent (other than being hydrodynamically coupled) or stand-alone, for example top-tensioned drilling and production risers operating in close proximity on a drilling vessel or production unit. The use of a combined model is appropriate where the two components are not structurally independent, for example in the case of two jumper hoses in the same hybrid riser system.

If the upstream and downstream structures are being analysed separately, you need to perform a static analysis of the upstream riser subject to current loading before you can proceed to modelling the downstream structure. You inform Flexcom the name of the analysis with current of this structure. If both structures are defined in the same model, you specify the wake interference data when you are performing the static analysis of both structures subject to current loading. Regardless of which modelling strategy you use, the conduct of the wake interference analysis of the downstream structure is the same. At each iteration at each solution time, Flexcom computes a reduced current velocity profile at the downstream structure based on the configuration of both structures and the selected wake model (Huse's, Blevins' or user-defined). Obviously this can change from iteration to iteration depending on the results from the previous iteration, but eventually a converged static solution is achieved. Once this has been completed for all solution times, the analysis terminates in the usual way.

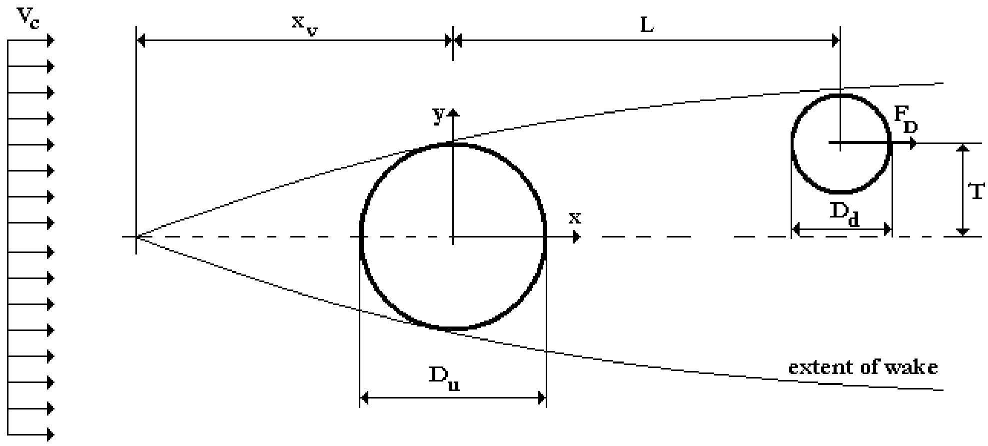
RELEVANT KEYWORDS

- `*WAKE UPSTREAM` is used to specify the composition of the upstream structure in terms of element sets, for use in wake interference calculations. The keyword also selects the wake interference model to be used, and defines associated data to characterise the wake field.
- `*WAKE DOWNSTREAM` is used to specify the composition of the downstream structure in terms of element sets for use in wake interference calculations. The keyword also facilitates the (optional) definition of lift coefficients in the case of a *User-Defined* wake interference model.

Huses Model

THEORY

The basic arrangement of Huse (1993) wake flow model is illustrated in the figure below.



Huse's Model

The current impacts on the upstream structure with velocity Vc, giving rise to the wake region. L and T denote the centre to centre longitudinal and transverse separations between the upstream and downstream structures. Huse's model introduces three coefficients to determine the profile of the reduced current velocity. The default values for these coefficients are k1=0.25, k2=1 and k3=0.693, respectively. The drag load on the downstream structure is evaluated using the reduced current velocity instead of the undisturbed current velocity Vc. In the equations that follow the u subscript refers to the upstream structure and the d subscript to the downstream structure, respectively.

The decrease in the in-line water particle velocity in the wake region (the wake velocity) is represented by *u*.

$$u = U_0 \exp\left(-k_3\left(\frac{y}{b}\right)^2\right) \qquad (1)$$

To examine this velocity in more detail, it is convenient to define several parameters. The peak wake velocity, *U0*, is the wake velocity along the centre line of the wake (y=0).

$$U_0 = k_2 V_c \sqrt{\frac{Cd_u D_u}{x}} \qquad (2)$$

The wake half-width, *b*, is the distance from the centre line of the wake to the point at which the wake velocity is equal to half the peak wake velocity.

$$b = k_1 \sqrt{Cd_u D_u x} \qquad (3)$$

The three Equations above (1) to (3) are strictly valid only some distance downstream of the cylinder. Very close to the structure they give a wake peak which is too high and narrow (Huse, 1993), which will lead to erroneous results in calculating the wake peak. To correct for this, Huse introduces the concept of a “virtual” wake source. In Equations (2) and (3) the term x is now replaced by xs.

$$x_s = x_v + x \qquad (4)$$

At the centre of the upstream cylinder, the half-width of the wake is equal to half of the diameter of the structure. In order to calculate the location of the wake source, Equation (3) is used, with x taken at the centre of the upstream structure. At this location, x is 0 and Equation (4) becomes:

$$x_s = x_v \qquad (5)$$

Therefore, substituting Du/2 for b and xs for x in Equation (3) allows the location of the wake source to be determined as:

$$x_v = \frac{D_u}{4k_1^2 Cd_u} \qquad (6)$$

The reduced current velocity is determined as:

$$V_R = V_c - u \qquad (7)$$

As a result, the drag force on the downstream structure is:

$$F_D = \frac{1}{2}\rho D_d Cd_d V_R^2 \qquad (8)$$

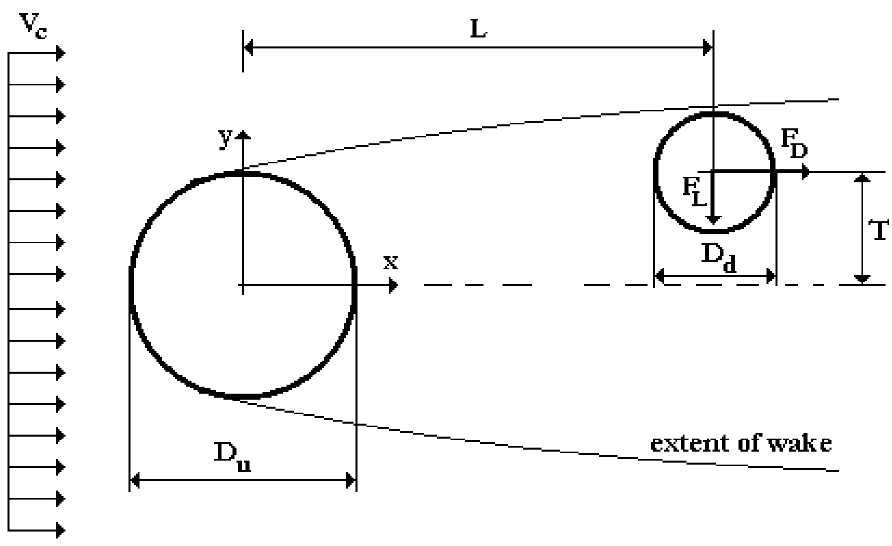
RELEVANT KEYWORDS

- *WAKE UPSTREAM is used to specify the composition of the upstream structure in terms of element sets, for use in wake interference calculations. The keyword also selects the wake interference model to be used, and defines associated data to characterise the wake field. Specifically, the TYPE=HUSE inputs are used to specify parameters relating to Huse's wake interference formulation.

Blevins Model

THEORY

The basic arrangement of Blevins (2005) wake flow model is illustrated in the figure below.



Blevins' Model

The current impacts on the upstream structure with velocity Vc, giving rise to the wake region. L and T denote the centre to centre longitudinal and transverse separations between the upstream and downstream structures. Blevins' model introduces three coefficients to determine the profile of the reduced current velocity and the coefficient of lift. The default values for these coefficients are a1=1, a2=4.5 and a3=-10.6, respectively. The drag load on the downstream structure is evaluated using the reduced current velocity instead of the undisturbed current velocity Vc. The lift force acts towards the centre line of the wake and is evaluated based on the lift coefficient determined from this wake model. In the equations that follow the *u* subscript refers to the upstream structure and the *d* subscript to the downstream structure, respectively.

The decrease in the in-line water particle velocity in the wake region (the wake velocity) is represented by *u*.

$$u = V_c a_1 b N \qquad (1)$$

where N is the decaying factor:

$$N = \exp\left(\frac{-a_2 y^2}{Cd_u D_u x}\right) \qquad (2)$$

and *b* is:

$$b = \sqrt{\frac{Cd_u D_u}{x}} \qquad (3)$$

The reduced current velocity is:

$$V_R = V_c - u \qquad (4)$$

As a result, the drag force on the downstream structure is:

$$F_D = \frac{1}{2}\rho D_d Cd_d V_R^2 \qquad (5)$$

The lift coefficient is:

$$C_L = a_3 \frac{yCd_d D_d}{xCd_u D_u} b(1 - a_1 bN)N \qquad (6)$$

The lift force on the downstream structure is:

$$F_L = \frac{1}{2} \rho D_d C_L V_c^2 \qquad (7)$$

RELEVANT KEYWORDS

- *WAKE UPSTREAM is used to specify the composition of the upstream structure in terms of element sets, for use in wake interference calculations. The keyword also selects the wake interference model to be used, and defines associated data to characterise the wake field. Specifically, the TYPE=BLEVINS inputs are used to specify parameters relating to Blevins wake interference formulation.

User-Defined Wake Model

THEORY

This wake model allows you to specify drag and optionally, lift coefficients as functions of the longitudinal and transverse centre to centre separations between the upstream and downstream structures. The drag and lift coefficients used in the calculation of the drag and lift forces are determined from the specified data using bilinear interpolation. Any coefficient requested outside the specified range is taken as zero. The drag and lift forces on the downstream structure are calculated as follows:

$$F_D = \frac{1}{2} \rho D_d Cd_d V_c^2 \qquad (1)$$

$$F_L = \frac{1}{2} \rho D_d C_L V_c^2 \qquad (2)$$

RELEVANT KEYWORDS

- *WAKE UPSTREAM is used to specify the composition of the upstream structure in terms of element sets, for use in wake interference calculations. The keyword also selects the wake interference model to be used, and defines associated data to characterise the wake field. Specifically, the TYPE=USER inputs are used to specify user defined Drag & Lift coefficients for the upstream structure.

VIV Drag

THEORY

You may optionally specify that the drag coefficients you defined (in term of standard hydrodynamic properties) are to be amplified, based on the results of an analysis with the VIV program Shear7. This enables you to model the increase in drag that VIV typically causes.

There are two inputs, one mandatory, when you want to include so-called VIV Drag in a Flexcom analysis with current. The first (mandatory) input is the name of the Shear7 analysis from the results of which you want Flexcom to read drag amplification factors. The second (optional) input is the name of the set of elements to which you want these amplification factors to be applied; this defaults to all elements. How this capability operates is as follows. When computing current forces, Flexcom loops over all elements in the set you nominate. For a given integration point, the program finds the drag amplification factor output for the nearest points in the Shear7 data, and then calculates a value for the integration point by linear interpolation. This factor then multiplies the user-specified drag coefficient when the current force for the integration point is being found via Morison's Equation.

RELEVANT KEYWORDS

- *VIV DRAG is used to instruct Flexcom to read vortex-induced vibration (VIV) drag coefficient amplification factors from the results of a Shear7 analysis.

VIV Effects

THEORY

This section explains the automatic interface to Shear7, which allows you to iteratively perform a VIV analysis of the downstream structure as part of the overall wake interference computations. This produces enhanced drag coefficients may be more accurately determined based on the reduced current profile.

You can include the effect of VIV-enhanced drag on one or both of the upstream and downstream structures using the VIV Drag facility described earlier (refer to VIV Drag for further details). However, Flexcom provides a further level of sophistication for the downstream structure analysis, as follows.

In performing a Shear7 analysis of either structure, you input a current velocity distribution, which is generally the ‘undisturbed’ current distribution you also input into Flexcom. In addition, the modes shapes generated by Flexcom for input to Shear7 might (optionally) have been calculated on the basis of this current velocity distribution (and the resulting distribution of tension in the structure). You can tell Flexcom to repeatedly recalculate the modal solution for the downstream structure and to rerun Shear7 based on this solution as the wake interference analysis proceeds, to reflect the actual current velocity ‘seen’ by that structure and the resulting tension distribution in the structure. In Flexcom parlance this capability is termed VIV Effects, to distinguish it from the VIV Drag capability described earlier.

To summarise, the procedure is as follows:

1. At each iteration at each solution time, once the static configuration is determined, the program automatically performs a modal analysis of the downstream structure (or of both structures if you are using a combined model).

2. The results of this analysis are used to generate an mds file for input to Shear7.

3. Flexcom automatically executes Shear7, and reads the drag amplification factors from the Shear7 output file.

4. The drag amplification factors are used to compute enhanced drag coefficients for the downstream structure.

5. The program solves again to determine the downstream structure static configuration, and so on until convergence is achieved at all solution times.

For Flexcom to be able to operate in this way, you need to prepare (in advance) a modal analysis input file with the relevant data for both the modal analysis itself and the Shear mds file generation. The name of this file forms part of the VIV Effects specification. If your two structures are in the same model, you may need to give careful consideration in preparing this file, to ensure that Flexcom includes the correct modes in the Shear7 file.

It is worth noting that this process is computationally intensive and greatly increases the run-times for wake interference analyses. Also, in order to include the effect of VIV on your downstream structure analysis in this way, you must have Shear7 installed on your PC. Shear7 is not supplied with Flexcom.

RELEVANT KEYWORDS

- `*VIV EFFECTS` is used to instruct Flexcom to continuously run Modes/Shear7 analyses of the downstream structure during the wake interference analysis.

Summary of Scenarios

All of the above means that there are quite a number of different scenarios or analysis sequences possible when you want to model wake interference effects. In order to demonstrate what is possible and permissible, Structures Analysed Separately and Structures Analysed Together outline some of the possible scenarios. The list of scenarios is not intended to be exhaustive; there are other scenarios possible which are not included here. However, the list of scenarios does illustrate the range of possible specifications from the simplest to the most sophisticated. Scenarios are provided firstly for the situation where the two structures are analysed separately, and then for the situation where the two structures are analysed in the same model. Note that each bullet point represents a separate analysis which must be performed directly by the user.

Structures Analysed Separately

Scenario 1:

- Initial static analysis of upstream structure
- Static analysis with current of upstream structure
- Initial static analysis of downstream structure
- Static analysis with current of downstream structure with wake interference

Scenario 2:

- Initial static analysis of upstream structure
- Modal analysis of upstream structure with Shear7 output

- Shear7 analysis of upstream structure
- Static analysis with current of upstream structure with VIV drag
- Initial static analysis of downstream structure
- Modal analysis of downstream structure with Shear7 output
- Shear7 analysis of downstream structure
- Static analysis with current of downstream structure with wake interference and VIV drag

Scenario 3:

- Initial static analysis of upstream structure
- Modal analysis of upstream structure with Shear7 output
- Shear7 analysis of upstream structure
- Static analysis with current of upstream structure with VIV drag
- Initial static analysis of downstream structure
- Static analysis with current of downstream structure with wake interference and VIV effects – Flexcom automatically performs modal analysis and Shear7 analysis at each iteration.

Structures Analysed Together

Scenario 1:

- Initial static analysis of both structures
- Static analysis with current of both structures with wake interference

Scenario 2:

- Initial static analysis of both structures
- Modal analysis of both structures with Shear7 output for upstream structure
- Shear7 analysis of upstream structure

- Modal analysis of both structures with Shear7 output for downstream structure
- Shear7 analysis of downstream structure
- Static analysis with current of both structures with wake interference and VIV drag

Scenario 3:

- Initial static analysis of both structures
- Modal analysis of both structures with Shear7 output for upstream structure
- Shear7 analysis of upstream structure
- Static analysis with current of both structures with wake interference and a) VIV drag for upstream structure and b) VIV effects for downstream structure – Flexcom automatically performs modal analysis and Shear7 analysis at each iteration.

1.9.3.7 Coupled Analysis

OVERVIEW

A coupled analysis is a specialized type of analysis in which the vessel (or more correctly referred to as the floating body) is explicitly included in the model. It might typically be used in the analysis of relatively small offshore structure such as CALM buoys, where the motions of the body are heavily influenced by the lines which are attached to it. The inertia of vessels such as FPSOs is sufficiently large such that the vessel motions are independent of the presence of the attached risers. In traditional (decoupled) analysis the vessel motions are prescribed via (displacement) RAOs which are known beforehand. In a coupled analysis, various first and second order forces are applied to the vessel, and the solution is essentially the vessel response.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Evolution of Flexcom Capabilities outlines the main options available in Flexcom, beginning with the earlier “CALM Buoy” and “Moored Vessel” facilities, and describes the additional modelling complexity afforded by the more recent “Floating Body” capability.

- CALM Buoy provides a general overview of the CALM buoy analysis capability.
- Moored Vessel provides a general overview of the moored vessel analysis capability.
- Floating Body provides a general overview of the floating body analysis capability.

RELEVANT KEYWORDS

- `*CALM MODEL` is used to specify the properties of calm buoys.
- `*MOORED VESSEL` is used to define a moored vessel and its associated properties.
- `*FLOATING BODY` is used to define a floating body and its associated properties.

Evolution of Flexcom Capabilities

The current version of Flexcom possesses comprehensive coupled analysis capabilities. Earlier versions of Flexcom incorporated two basic options in terms of coupled analysis. The first of these, the CALM Buoy facility, calculates the high frequency response of a floating buoy subject to first order wave forces. The second option, the Moored Vessel capability, is designed to capture the low frequency response of a moored vessel subject to second order drift forces.

More recent versions also provide a more generalised Floating Body capability, developed by Wood (formerly MCS), with the support of Chevron, ConocoPhillips and ExxonMobil, as part of a joint industry project on coupled analysis. The floating body model is a considerably more powerful facility than either of the earlier coupled analysis features:

(i) Floating body analysis is applicable to both the time and frequency domain solutions (both the CALM buoy and moored vessel facilities are restricted to the time domain).

(ii) First and second order forces can be applied to the floating body (loading on the CALM buoy is restricted to first order, while only low frequency drift forces may be applied to the moored vessel).

(iii)The characteristics of the floating body (e.g. added mass) and the applied loading (e.g. via force RAOs) is specified in terms of full three-dimensional 6x6 matrices, whereas the specification for both the CALM buoy and moored vessel facilities is restricted to simplified two-dimensional inputs.

(iv) The added mass and radiation damping coefficients may vary as a function of frequency, whereas the specification for both the CALM buoy and moored vessel facilities is restricted to frequency independent terms. For the floating body model, frequency-dependent added mass and inertia terms are calculated via a convolution integral approach in the time domain.

(v) Hydrodynamic coupling between adjacent floating bodies may be modelled.

(vi) An interface to WAMIT is provided, allowing hydrodynamic data to be imported directly into Flexcom.

CALM Buoy

THEORY

Flexcom can perform a coupled regular wave analysis of an offshore system that includes a CALM buoy, such as for example an oil offloading system. In such an analysis, the buoy motions are not calculated from user-specified RAOs. Rather the buoy is explicitly included in the model (using standard beam-column finite elements), and is subjected to weight, buoyancy and hydrodynamic forces based on specified buoy properties and force terms. These will typically come from a radiation/diffraction analysis of the buoy structure. The advantage of this procedure is that the effect of attached risers and mooring lines on the motions of the buoy is more accurately modelled. If these motions are calculated directly from buoy RAOs, then the influence of the risers and mooring lines may be lost, since these influences would not normally be considered in the derivation of the RAOs.

Flexcom assumes that the CALM buoy is symmetric, so all force, added mass and damping data refers to a 2D plane, with heave, surge and pitch components only. Of course the response of the buoy may not be confined to this 2D plane (because of asymmetry in riser and mooring line connections), so inertia and stiffness terms for yaw and roll are required in the buoy properties. The terms heave, surge, yaw, roll and pitch in this context refer as usual to a local axis system centred on the buoy CoG. The definition of this system is as follows. Heave is vertically upwards, coincident with global X. The buoy local surge axis is assumed to be coincident with the dynamic analysis regular wave direction. The other local directions then follow based on a right-handed axis system.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Model Set-up presents some general guidelines on setting up a CALM buoy model.
- Buoy Properties provides an overview of the relevant CALM buoy properties.
- Buoy Forces outlines how the forces on the CALM buoy are calculated.

RELEVANT KEYWORDS

- `*CALM MODEL` is used to specify the properties of calm buoys.
- `*CALM LOAD` is used to specify data defining the force terms to be applied to a CALM buoy.

If you would like to see an example of how these keywords are used in practice, refer to E01 - CALM Buoy - Simple.

Model Set-up

The following is a summary of how to use this facility in a series of analyses of a system that includes a CALM buoy. Firstly, you include the buoy directly in your model with an assemblage of standard beam-column elements. The geometric properties you assign to the buoy elements should be such as to make the buoy rigid, and one node of the assemblage should be located at the buoy centre of gravity (CoG). You should suppress drag and buoyancy forces on the buoy elements by specifying zero effective drag and buoyancy diameters for these elements. Likewise the elements should have zero mass and zero polar inertia. You specify data for all of these forces on the buoy separately, and all of these effects are applied at the buoy CoG. The reason why you model the buoy as an assemblage of elements rather than just have one CoG node is so that you can have the risers and moorings lines attached to the buoy in their actual, real-life locations. This allows the effect on the buoy motions of the spatial separation between these connection points to be accurately modelled.

Buoy Properties

THEORY

The specified buoy properties include:

(i) The mass of the buoy, and its inertia in yaw, roll and pitch

(ii) The buoy added mass matrix

(iii)The buoy stiffness (due to buoyancy) in heave, roll and pitch

(iv)The buoy radiation damping matrix

(v) Hydrodynamic force terms in heave, surge and pitch, appropriate to the dynamic analysis regular wave period

Properties (i) to (iv) must be supplied in an initial static analysis, along with the geometry and properties of the rest of the model. The load terms in (v) would normally be supplied with the rest of the regular wave dynamic analysis data. All of the inputs would typically be generated using a standard radiation/diffraction program.

There is one final important point to note. The initial draft of the CALM buoy, as reflected by the locations of the nodes and elements of the CALM buoy assemblage, should be at the level where the weight and buoyancy of the buoy are in equilibrium (the buoy is floating). This is implicitly assumed by Flexcom, so that changes in the buoyancy loads due to relative buoy/surface elevation changes are accounted for by the buoyancy terms in the stiffness matrix from (iii) above.

The buoy mass and inertia matrix is defined in the following format:

$$\begin{bmatrix} M_{11} & 0 & 0 & 0 & 0 & 0 \\ 0 & M_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & M_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{11} & 0 & 0 \\ 0 & 0 & 0 & 0 & I_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{33} \end{bmatrix}$$

where:

- $M_{11} = M_{22} = M_{33}$ is the mass of the buoy
- I_{11} is the roll inertia of the buoy
- I_{22} is the pitch inertia of the buoy
- I_{33} is the yaw inertia of the buoy

The added mass matrix is defined in the following format:

$$\begin{bmatrix} A_{11} & 0 & 0 & 0 & A_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ A_{15} & 0 & 0 & 0 & A_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where:

- A_{11} is the surge added mass
- A_{33} is the heave added mass
- A_{55} is the pitch added mass
- A_{15} is the coupled surge pitch added mass

The buoyancy stiffness matrix is defined in the following format:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & K_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & K_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where:

- K_{33} is the heave stiffness
- K_{44} is the roll stiffness
- K_{55} is the pitch stiffness

The radiation damping matrix is defined in the following format:

$$\begin{bmatrix} D_{11} & 0 & 0 & 0 & D_{15} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & D_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ D_{15} & 0 & 0 & 0 & D_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

where:

- D_{11} is the surge damping
- D_{33} is the heave damping
- D_{55} is the pitch damping
- D_{15} is the coupled surge pitch damping

RELEVANT KEYWORDS

- `*CALM MODEL` is used to specify the properties of calm buoys.

If you would like to see an example of how this keyword is used in practice, refer to E01 - CALM Buoy - Simple.

Buoy Forces

THEORY

Sinusoidally varying point loads are applied directly in the local axes (heave, surge and pitch) at the CoG node, calculated from the force RAO data using the following equation:

$$F_i = A_w R_i \cos(kS - \omega t + \phi_i) \qquad (1)$$

where:

- F_i is the force in degree of freedom i

- A_w is the wave amplitude
- R_i is the force RAO for degree of freedom i
- k is the regular wave number (calculated by Flexcom)
- S is the distance of the CoG node from the vertical axis (Y=Z=0) at the start of the dynamic analysis (calculated by Flexcom)
- ω is the regular wave circular period
- t is the current analysis or simulation time
- φ_i is the force phase angle

RELEVANT KEYWORDS

- `*CALM LOAD` is used to specify data defining the force terms to be applied to a CALM buoy.

If you would like to see an example of how this keyword is used in practice, refer to E01 - CALM Buoy - Simple.

Moored Vessel

THEORY

Similar in some respects to the CALM buoy modelling facility, Flexcom can perform static and dynamic analyses of a model which explicitly includes a moored vessel. Again you model the vessel with an assemblage of standard beam-column elements, and locate a node at the vessel CoG. In this case, the additional “vessel” elements connect the vessel CoG to all of those riser nodes whose motions are defined by the motions of the moored vessel. These nodes are referred to as “attached” nodes in the following for convenience. There should be one (and only one) vessel element for each attached node, connecting it to the CoG node. These vessel elements should again be rigid or stiff, and should have zero mass and zero drag and buoyancy diameters.

The objective of a Flexcom dynamic analysis of a model which includes a moored vessel is to calculate the response of the vessel to i) current; ii) wind; iii) thrusters; and iv) drift forces calculated from a wave spectrum and user-input Quadratic Transfer Functions (QTFs). A key output from this type of analysis is the second order drift motion of the vessel (CoG node), which is in fact routed to an ASCII file as the analysis proceeds. These motions can then be subsequently added to first order (high frequency) vessel motions in a dynamic analysis of a detailed model of one or more of the mooring lines or risers.

In the normal way, the model will comprise of nodes and elements representing the mooring lines and risers (if present) which are connected to the vessel. The vessel itself will be modelled with an assemblage of standard beam-column elements, with a node located at the vessel CoG. These “vessel” elements will connect the vessel CoG to all of those mooring line and riser nodes whose motions are defined by the motions of the moored vessel. There should be one (and only one) vessel element for each attached node, connecting it to the CoG node. The vessel elements should be rigid or stiff, and should have zero mass and zero drag and buoyancy diameters. In order to keep the model size to reasonable proportions, and because long simulations of the low frequency vessel response may be required, the risers and mooring lines should be modelled with fewer elements than would typically be used if the wave frequency response of these components were the primary analysis objective.

After the various mooring analysis stages are run, the vessel low frequency or drift response is available as a function of time. Subsequently, more detailed models of one or more of the mooring lines and/or the risers would be developed, to determine the dynamic response of this component or components to wave frequency loading in the normal way - the model(s) in this case would not include the vessel explicitly. Rather the dynamic excitation would now consist of a combination of first order (high frequency) motions determined from RAOs, plus the second order motion time history produced with the earlier model. Flexcom includes an option to specify a vessel motion combination of this nature.

The rationale for this solution procedure is that, in accordance with mooring codes of practice, vessel low and high frequency motions are typically considered independently. In addition, the high frequency vessel response is typically not influenced by the restraint applied by mooring lines and any risers present.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Calculation of Vessel Forces explains in detail how the forces on the moored vessel are calculated.

- Analysis Procedure presents some general guidelines on the steps involved in performing a full dynamic analysis with a moored vessel.

- Input Formats describes the relevant input formats for the definition of current coefficients, wind coefficients and QTFs.

- Vessel Heading discusses vessel heading in the context of moored vessel analysis.

RELEVANT KEYWORDS

- `*MOORED VESSEL` is used to define a moored vessel and its associated properties.

- `*ANALYSIS TYPE` is used to specify the analysis type. Specifically, `TYPE=MOORING` is used in conjunction with `*MOORED VESSEL`.

- `*WIND` is used to specify wind loading.

- `*WIND COEFF` is used to specify wind coefficients used to determine the wind loading on a floating body or moored vessel.

- `*CURRENT COEFF` is used to specify current coefficients used to determine the current loading on a floating body or moored vessel.

- `*QTF` is used to specify Quadratic Transfer Functions (QTFs) that allow the slow drift loads on a floating body or moored vessel to be determined.

- `*MOMENTS` is used to specify the value of Molin’s yaw coefficient for a moored vessel, and also to specify the fractions of Molin’s Moment and Munk’s Moment that are applied to the moored vessel.

- `*THRUSTER` is used to specify thruster loads on a moored vessel.

If you would like to see an example of how these keywords are used in practice, refer to D01 - Moored Vessel.

Calculation of Vessel Forces

The following vessel forces are considered in determining the low-frequency vessel response:

(i) Hydrodynamic loads

(ii) Damping loads

(iii)Wave drift loads

(iv)Wind loads

(v) Current loads

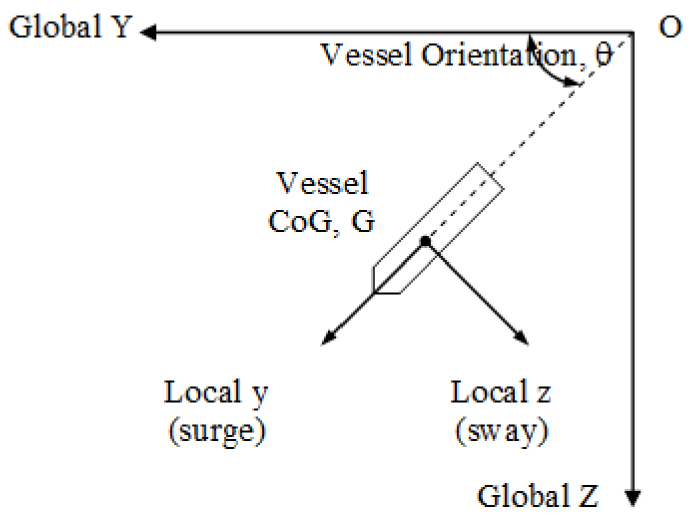
(vi)Thruster loads

(vii)Mooring line loads

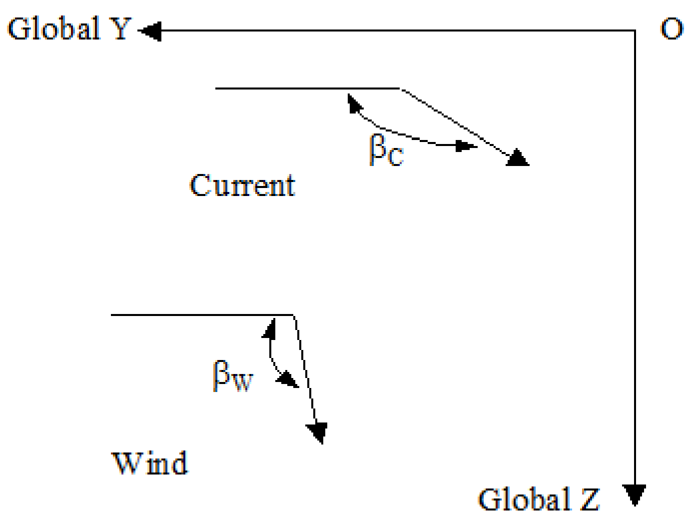
(viii)Riser loads

The final two of these loads (the riser and mooring line loads) are automatically applied to the vessel by virtue of the fact that these are included (coupled) with the vessel in the model. The other loads are calculated individually and are applied to the vessel CoG node.

The loads are calculated in a vessel axis system defined with its origin at the vessel centre of gravity, G, as shown in the figure below. The vessel y axis defines the local surge axis, and the z axis the local sway axis. The instantaneous vessel orientation, θ, is measured positive anticlockwise from the global Y axis. Only motions of the vessel in the horizontal plane (that is, surge, sway and yaw) are considered, so there are only three components to the vessel force vector: F_y and F_z, the forces along the local surge and sway axes respectively, and $M_{\theta/G}$, the yaw moment about the vessel centre of gravity G.



Local Vessel Axis System



Definition of Wind and Current Directions

THEORY

The hydrodynamic loads on the vessel, denoted F_{Hy} , F_{Hz} and $M_{H\theta/G}$, are determined according to the theory of manoeuvrability as follows:

$$F_{Hy} = -Ma_{yy}\dot{u}' + Ma_{zz}v'\dot{\theta} + Ma_{z\theta}\dot{\theta}^2$$
$$F_{Hz} = -Ma_{zz}\dot{v}' - Ma_{yy}u'\dot{\theta} - Ma_{z\theta}\ddot{\theta}$$
$$M_{H\theta/G} = -Ma_{\theta\theta}\ddot{\theta} - Ma_{z\theta}\dot{v}' - Ma_{z\theta}u'\dot{\theta} - (Ma_{zz} - Ma_{yy})u'v' \qquad (1)$$

where:

- Ma_{yy} is the vessel added mass in surge
- Ma_{zz} is the vessel added mass in sway
- $Ma_{\theta\theta}$ is the vessel added mass in yaw
- $Ma_{z\theta}$ is the vessel added mass in sway-yaw coupling
- θ is the instantaneous vessel orientation as defined in Local Vessel Axis System figure
- u' and v' are the components, in the vessel axis system, of the relative vessel/current velocity.

u' and v' are given by the equations:

$$u' = u - v_c \cos(\beta_c - \theta)$$
$$v' = v - v_c \sin(\beta_c - \theta) \qquad (2)$$

where u and v are the components in the vessel axes of the vessel CoG velocity; v_c is the current velocity magnitude (which is a user input); and β_c is the current direction as defined in the above figure. Differentiating u' and v' with respect to time (assuming a constant current velocity) gives:

$$\dot{u}' = \dot{u} - v_c\dot{\theta}\sin(\beta_c - \theta)$$
$$\dot{v}' = \dot{v} + v_c\dot{\theta}\cos(\beta_c - \theta) \qquad (3)$$

Substituting the above expressions into the equations for the hydrodynamic loads gives the following:

$$F_{Hy} = -Ma_{yy}\dot{u} + Ma_{yy}v_c\dot{\theta}\sin(\beta_c - \theta) + Ma_{zz}v'\dot{\theta} + Ma_{z\theta}\dot{\theta}^2$$
$$F_{Hz} = -Ma_{zz}\dot{v} - Ma_{zz}v_c\dot{\theta}\cos(\beta_c - \theta) - Ma_{yy}u'\dot{\theta} - Ma_{z\theta}\ddot{\theta}$$
$$M_{H\theta/G} = -Ma_{\theta\theta}\ddot{\theta} - Ma_{z\theta}\dot{v} - Ma_{z\theta}v_c\dot{\theta}\cos(\beta_c - \theta) - Ma_{z\theta}u'\dot{\theta} - (Ma_{zz} - Ma_{yy})u'v'$$

(4)

For numerical reasons it is desirable to move any terms consisting of the product of a mass by a structure acceleration term to the left-hand side of the overall matrix equations of motion solved by Flexcom. So Term (1) in F_{Hy} , Terms (1) & (4) in F_{Hz} , and Terms (1) & (2) in $M_{H\theta/G}$ are moved in this way. This is achieved by adding the relevant vessel added mass terms to the mass matrix at the location corresponding to the vessel centre of gravity. The following expressions are then used to calculate the hydrodynamic forces which are applied at the vessel CoG:

$$F_{Hy} = Ma_{yy}v_c\dot{\theta}\sin(\beta_c - \theta) + Ma_{zz}v'\dot{\theta} + Ma_{z\theta}\dot{\theta}^2$$
$$F_{Hz} = -Ma_{zz}v_c\dot{\theta}\cos(\beta_c - \theta) - Ma_{yy}u'\dot{\theta}$$
$$M_{H\theta/G} = -Ma_{z\theta}v_c\dot{\theta}\cos(\beta_c - \theta) - Ma_{z\theta}u'\dot{\theta} - (Ma_{zz} - Ma_{yy})u'v' \qquad (5)$$

RELEVANT KEYWORDS

- `*MOORED VESSEL` is used to define a moored vessel and its associated properties. Specifically, the `TYPE=MASS` inputs are used to specify mass and added mass data.

If you would like to see an example of how this keyword is used in practice, refer to D01 - Moored Vessel.

THEORY

The damping loads are given by the following equations:

$$F_{By} = -B_{yy}u$$
$$F_{Bz} = -B_{zz}v$$
$$M_{B\theta/G} = -B_{\theta\theta}\dot{\theta} \qquad (1)$$

where:

- B_{yy} is the linear damping coefficient in surge referred to the vessel CoG
- B_{zz} is the linear damping coefficient in sway referred to the vessel CoG
- $B_{\theta\theta}$ is the linear damping coefficient in yaw referred to the CoG

RELEVANT KEYWORDS

- `*MOORED VESSEL` is used to define a moored vessel and its associated properties. Specifically, the `TYPE=DAMP` inputs are used to specify damping coefficient data.

If you would like to see an example of how this keyword is used in practice, refer to D01 - Moored Vessel.

THEORY

The wave drift loads are calculated from user-specified Quadratic Transfer Function (QTF) values according to the following equation:

$$F_{DG}(t)=\left[\sum_{k=1}^{n} a_k\sqrt{\left|QTF_G(\alpha_k,\omega_k)\right|}\cos(\omega_k t+\phi_k)\right]\bullet$$
$$\left[\sum_{k=1}^{n} a_k\sqrt{\left|QTF_G(\alpha_k,\omega_k)\right|}\cos(\omega_k t+\phi_k)\mathrm{sign}\left(QTF_G(\alpha_k,\omega_k)\right)\right]+$$
$$\left[\sum_{k=1}^{n} a_k\sqrt{\left|QTF_G(\alpha_k,\omega_k)\right|}\sin(\omega_k t+\phi_k)\right]\bullet$$
$$\left[\sum_{k=1}^{n} a_k\sqrt{\left|QTF_G(\alpha_k,\omega_k)\right|}\sin(\omega_k t+\phi_k)\mathrm{sign}\left(QTF_G(\alpha_k,\omega_k)\right)\right] \quad (1)$$

where:

- $F_{DG}(t)$ is one of the three wave drift loads applied to the vessel CoG (i.e. F_{Dy} , F_{Dz} , or $M_{D\theta/G}$)

- $QTF_G(\alpha_k, \omega_k)$ is the QTF value for relative wave incidence α_k (wave direction relative to the instantaneous vessel orientation) and frequency ω_k for the load direction in question
- a_k, ω_k and ϕ_k are the amplitude, frequency and phase respectively of the k^{th} wave harmonic
- sign(p) equals -1 if p < 0, 0 if p = 0, and 1 if p >0

The wave drift loads are calculated for every wave frequency present in the (discretised) wave spectrum and for every wave direction. A mean drift load is calculated in Static Mooring analyses to determine the mean vessel position, using the equation:

$$F_{DGmean} = 2\int_{\omega} QTF_G(\alpha_k, \omega_k) S_{\eta\eta}(\omega_k) d\omega \qquad (2)$$

where:

- F_{DGMean} is one of three mean wave drift loads applied to the vessel CoG (i.e. F_{DyMean}, F_{DzMean}, or $M_{D\theta Mean/G}$)
- $S_{\eta\eta}(\omega_k)$ is the wave spectrum

and the remaining symbols have the same meanings as above.

RELEVANT KEYWORDS

- `*QTF` is used to specify Quadratic Transfer Functions (QTFs) that allow the slow drift loads on a floating body or moored vessel to be determined.

If you would like to see an example of how this keyword is used in practice, refer to D01 - Moored Vessel.

THEORY

Current loads, denoted F_{Cy}, F_{Cz} and $M_{C\theta/G}$, are calculated using:

$$F_{Cy} = \frac{1}{2} C_{Cy}(\alpha_c)\rho_w TBu_c^2$$
$$F_{Cz} = \frac{1}{2} C_{Cz}(\alpha_c)\rho_w TL_{bp}u_c^2$$
$$M_{C\theta/G} = \frac{1}{2} C_{C\theta}(\alpha_c)\rho_w TL_{bp}^2 u_c^2 \quad (1)$$

where:

- $C_{Cy}(\alpha_c)$, $C_{Cz}(\alpha_c)$ and $C_{C\theta}(\alpha_c)$ are the current coefficients in surge, sway and yaw respectively for an equivalent current incidence of α_c
- ρ_w is the water density
- T is the vessel draft
- B is the vessel beam
- L_{bp} is the length between perpendiculars which is the sum of y_{Fore} and y_{Aft}
- y_{Fore} is the distance from the vessel centre of gravity to the forward perpendicular
- y_{Aft} is the distance from the vessel centre of gravity to the aft perpendicular
- u_c is the equivalent current velocity

The equivalent current incidence α_c is obtained using:

$$\alpha_c = \tan^{-1}\left[\frac{v'}{u'}\right] \quad (1)$$

and the equivalent current velocity u_c is given by:

$$u_c = \sqrt{u'^2 + v'^2} \qquad (2)$$

where u' and v' are as previously defined.

You can specify that a certain fraction of Munk's yaw moment be subtracted from the current loads applied to the vessel. Munk's moment is calculated using:

$$M_{C\theta Munk/G} = -\left(Ma_{zz} - Ma_{yy}\right) v_c^2 \sin\left(\beta_c - \theta\right) \cos\left(\beta_c - \theta\right) \qquad (3)$$

where the various symbols are as previously defined. Additionally, you can specify that a certain fraction of Molin's yaw moment is added to the current loads applied to the vessel. Molin's moment is calculated using the following expression:

$$M_{C\theta Molin/G} = \frac{1}{2}\rho_w C_{Molin} T \int_{y_{aft}}^{y_{fore}} \left(V_T(y,\dot{\theta}) V(y,\dot{\theta}) - V_T(y,0) V(v,0)\right) y\,dy$$

(4)

where:

- C_{Molin} is Molin's yaw coefficient
- $V_T(y,\dot{\theta})$ is the transverse component of the relative fluid velocity at a distance y from the vessel centre of gravity, given by $V_T(y,\dot{\theta}) = -\left(v' + y\dot{\theta}\right)$
- $V(y,\dot{\theta})$ is the total relative fluid velocity at a distance y from the vessel centre of gravity, given by $V(y,\dot{\theta}) = \sqrt{u'^2 + (v' + y\dot{\theta})^2}$

RELEVANT KEYWORDS

- *MOORED VESSEL is used to define a moored vessel and its associated properties. Specifically, the TYPE=GEOM inputs are used to specify vessel geometrical data such as draft, beam, length between perpendiculars etc.

- *CURRENT COEFF is used to specify current coefficients used to determine the current loading on a floating body or moored vessel.
- *MOMENTS is used to specify the value of Molin's yaw coefficient for a moored vessel, and also to specify the fractions of Molin's Moment and Munk's Moment that are applied to the moored vessel.

If you would like to see an example of how these keywords are used in practice, refer to D01 - Moored Vessel.

THEORY

Vessel wind loads are calculated using:

$$F_{Wy} = \frac{1}{2} C_{Wy}(\alpha_w) \rho_a A_T u_w^2$$
$$F_{Wz} = \frac{1}{2} C_{Wz}(\alpha_w) \rho_a A_L u_w^2$$
$$M_{W\theta/G} = \frac{1}{2} C_{W\theta}(\alpha_w) \rho_a A_L L_{bp} u_w^2 \qquad (1)$$

where:

- $C_{Wy}(\alpha_c)$, $C_{Wz}(\alpha_c)$ and $C_{W\theta}(\alpha_c)$ are the wind coefficients in surge, sway and yaw respectively for an equivalent wind incidence of α_w
- ρ_a is the air density
- A_T is the transverse area of the vessel which is exposed to wind action
- A_L is the longitudinal area of the vessel which is exposed to wind action
- u_w is the equivalent wind velocity

The equivalent wind incidence α_w is obtained using:

$$\alpha_w = \tan^{-1}\left[\frac{v - v_w \sin(\beta_w - \theta)}{u - v_w \cos(\beta_w - \theta)}\right] \tag{2}$$

where:

- v_w is the (global) wind velocity
- β_w is the wind direction as defined in the Wind and Current Directions figure.

The equivalent wind velocity u_w is given by:

$$u_w = \sqrt{(u - v_w \cos(\beta_w - \theta))^2 + (v - v_w \sin(\beta_w - \theta))^2} \tag{3}$$

RELEVANT KEYWORDS

- `*MOORED VESSEL` is used to define a moored vessel and its associated properties. Specifically, the `TYPE=GEOM` inputs are used to specify vessel geometrical data such as draft, beam, length between perpendiculars etc.
- `*WIND` is used to specify wind loading.
- `*WIND COEFF` is used to specify wind coefficients used to determine the wind loading on a floating body or moored vessel.

If you would like to see an example of how these keywords are used in practice, refer to D01 - Moored Vessel

THEORY

Thruster loads are directly specified by the user and may be defined with reference to the local vessel axis system or the global axes. You define the locations of the various thrusters (that is, the point of application of the thruster loads) relative to the local vessel axis system. Flexcom automatically transforms thruster loads to a force and moment applied at the vessel centre of gravity. Thruster loads can be constant (time-invariant) or switched on and off during an analysis.

RELEVANT KEYWORDS

- *THRUSTER is used to specify thruster loads on a moored vessel.

Analysis Procedure

THEORY

Introduction

As noted earlier, the moored vessel is explicitly defined by a node positioned at the vessel CoG and rigid massless elements connecting this CoG node to all “attached” nodes on the mooring lines and risers. Boundary conditions should be applied at the mooring line or riser nodes on the seabed only. BCs should not be specified at either the CoG node or any attached node(s). Flexcom automatically restrains these nodes as appropriate to the various mooring analysis types or stages. These analysis stages and the BCs that are automatically applied are as follows.

Fixed Mooring Analysis Stage

The first mooring analysis type is denoted a Static Fixed – this corresponds to an initial static analysis of a standard riser model, and should be the first analysis performed on the combined mooring lines/risers/vessel model. So this option will be invoked in the run in which the geometry and properties of your FE model are defined. In this analysis, the CoG node and all attached nodes are restrained in all DOFs. This is because the principal objective in this case is to develop tension distributions in the mooring lines and risers prior to the application of any displacements or loads (except for buoyancy and gravity of course).

Static Mooring Analysis Stage

The second mooring analysis type is Static Mooring. This would correspond to a static restart of a standard riser model in which offset and/or current are applied, the objective being to find a mean static configuration prior to the application of dynamic loading. In a Static Mooring analysis, static vessel forces due to i) wind, ii) current, iii) thrusters loads, and iv) a mean drift force calculated from a wave spectrum and user-input Quadratic Transfer Functions (QTFs), are applied at the CoG node. In this analysis the vessel is restrained to translate and rotate in a vertical plane; so only DOF 2, 3 and 4 motions are free at the CoG (the DOF 4 motions correspond to yaw of the vessel). DOFs 1, 5 and 6 are restrained at the CoG. The objective here is again to find a stable mean static configuration prior to the application of dynamic low frequency or drift forces. After Static Mooring analyses, Flexcom calculates an estimate of the mooring system linear stiffness at the mean vessel position. This estimate is calculated using force and deflection data from the final mean vessel position and from the most recently available position of the vessel prior to the final mean vessel position. For example, if the static vessel loads are ramped on in ten steps during the Static Mooring analysis, Flexcom will calculate the mooring system stiffness using the results from Step 9 and Step 10. So it is recommended that in a Static Mooring analysis a reasonably large number of steps should be used. It should, however, be noted that the mooring system stiffness is only an estimate which takes no account of direction or the non-linear nature of the parameter.

Dynamic Mooring Analysis Stage

The dynamic drift forces are applied in the final mooring analysis type, Dynamic Mooring. The loading in this analysis is similar to the Static Mooring case, except that now the QTFs and the wave spectrum define a dynamically-varying (rather than a mean) drift force. The same restraints or boundary conditions at the CoG and attached nodes are (automatically) applied by Flexcom. In addition to the normal output files produced in a Flexcom dynamic analysis, in this case the program also generates (automatically) an ASCII file containing time histories of CoG motion in all 6 DOFs (3 of which are naturally 0). This file is named jobname.mor, where as usual jobname is the generic analysis file name.

After the Dynamic Mooring analysis is completed, this file would typically be used in a separate dynamic analysis of a detailed model or models of one or more of the mooring lines and/or the risers. The objective here would be to determine the dynamic response of this component or components to wave frequency loading in the usual way; the model(s) in this case would not include the vessel explicitly. Rather the dynamic excitation would now consist of a combination of first order (high frequency) motions determined from RAOs, plus the second order motion time history produced with the earlier model. Flexcom includes an option to specify a vessel motion combination of this nature. The regular output file from the Dynamic Mooring analysis also contains statistics of offsets. These statistics are calculated using the mooring system stiffness determined by the preceding Static Mooring analysis. The mean, significant, statistical maximum and observed maximum offset are determined for the low frequency motions. A warning is given if the analysis is of insufficient length to provide meaningful statistics.

RELEVANT KEYWORDS

- `*MOORED VESSEL` is used to define a moored vessel and its associated properties.
- `*ANALYSIS TYPE` is used to specify the analysis type. Specifically, `TYPE=MOORING` is used in conjunction with `*MOORED VESSEL`. Further options are provided by the `MOORING=` input, which can be `FIXED`, `STATIC` or `DYNAMIC`, corresponding to the respective sections above.

If you would like to see an example of how these keywords are used in practice, refer to D01 - Moored Vessel.

Input Formats

THEORY

Overview

QTF values, current coefficients and wind coefficients are input using ASCII files. Each of these files comprise two types of lines, a data line and a comment line. The letter C (uppercase essential) in Column 1 identifies a comment line. Comment lines (which can be included at any point in the file) are completely ignored by Flexcom, and are intended to allow you to include comments for your own benefit or for other users. Any line that is not a comment line is a data line. The actual data values expected on a data line are a function of position within the file. Numerical inputs on a data line are in free format, and can be specified as floating point numbers with or without exponent. Use an uppercase E when specifying an exponent. Either commas or blanks may separate a number of values on a particular data line; do not use TABs.

QTF Data

The QTF data is divided into groups, with each group providing data for a particular wave frequency. Each group consists of the following lines:

1st Line: One value:

FREQ= Wave frequency (in Hz)

2nd and subsequent lines: Four values as follows:

Relative vessel/wave heading, QTFy, QTFz, QTF_{θ}

Line 2 is repeated until QTF values are specified for all headings for this wave frequency. Then a second FREQ= line signals the start of the next group of values, and so on until values for all wave frequencies have been input. Relative vessel/wave heading in this context is defined as the angle of incidence of the wave relative to the vessel axes; refer to Vessel Heading for a more detailed definition. QTFy, QTFz and QTF_{θ} are respectively the QTF values in surge, sway and yaw at a particular frequency.

Current Coefficients

Each line of data in a current coefficient file has the following format:

All data lines: Four values as follows:

Relative vessel/current heading, Cc_y, Cc_z, Cc_{θ}

This line is repeated until current coefficient values are specified for all headings. Relative vessel/current heading in this context is defined as the angle of incidence of the current relative to the vessel axes; refer to Vessel Heading for a more detailed definition. Cc_y, Cc_z, and Cc_θ are respectively the current coefficients in surge, sway and yaw at a particular heading.

Wind Coefficients

Each line of data in a wind coefficient file has the following format:

All data lines: Four values as follows:

Relative vessel/wind heading, Cw_y, Cw_z, Cw_θ

This line is repeated until wind coefficient values are specified for all headings. Relative vessel/wind heading in this context is defined as the angle of incidence of the wind relative to the vessel axes; refer to Vessel Heading for a more detailed definition. Cw_y, Cw_z, and Cw_θ are respectively the wind coefficients in surge, sway and yaw at a particular heading.

QTFs and current and wind coefficients at frequencies or headings intermediate to the values you input here are found by linear interpolation. Outside of the range of frequency and/or heading values you use, a value of zero is assumed for the parameter in question.

RELEVANT KEYWORDS

- `*QTF` is used to specify Quadratic Transfer Functions (QTFs) that allow the slow drift loads on a floating body or moored vessel to be determined.
- `*CURRENT COEFF` is used to specify current coefficients used to determine the current loading on a floating body or moored vessel.
- `*WIND COEFF` is used to specify wind coefficients used to determine the wind loading on a floating body or moored vessel.

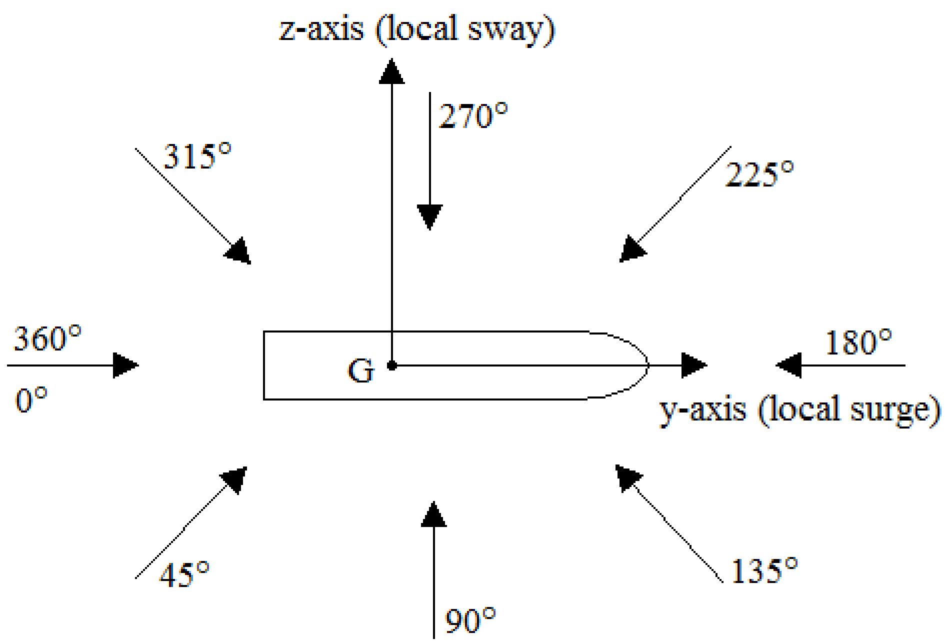
If you would like to see an example of how these keywords are used in practice, refer to D01 - Moored Vessel.

Vessel Heading

Current coefficient, wind coefficient, and QTF data, is specified as a function of the relative “heading” between the vessel and, respectively, current, wind and waves (QTFs are of course also defined as a function of wave frequency). It is important to be clear on the definition of “heading” for these inputs which is slightly different from the definition of “vessel heading” which applies to the input of first order vessel motion RAOs.

Dealing first with current, the coefficients are described as being defined as a function of Relative vessel/current heading, or of the angle of incidence of the current relative to the vessel axes. This “heading” is in fact the equivalent current incidence α_c defined previously in Current Loads, earlier in this note. For the case of a Static Mooring analysis, the vessel CoG velocity components u and v are both 0, so α_c is simply (β_c - θ) that is, the global current direction minus the instantaneous vessel orientation. This is simply the current direction relative to the vessel surge axis, measured positive anti-clockwise from the positive axis direction. The figure below illustrates this definition schematically. Note that in strict mathematical terms the Relative vessel/current heading is the angle, positive anti-clockwise, from the positive direction of the local surge axis to the current direction drawn at the CoG.

The current coefficient sign at a particular heading should naturally reflect the direction of the current force for that heading. For example, a current with a heading of 180° would be expected to apply a force to the vessel along the negative direction of the local surge axis. So the current coefficient in surge for this heading would typically be negative.



Definition of Relative Vessel/Current Heading

Likewise wind coefficients are described as being a function of Relative vessel/wind heading, or of the angle of incidence of the wind relative to the vessel axes. This "heading" is likewise the equivalent wind incidence α_w defined previously in Wind Loads. For the case of a Static Mooring analysis, α_w is simply (β_w - θ), that is, the global wind direction minus the instantaneous vessel orientation, which is again simply the wind direction relative to the vessel surge axis, positive anti-clockwise from the positive direction of the axis.

Finally, QTFs are described as being defined as a function of Relative vessel/wave heading, or of the angle of incidence of the wave relative to the vessel axes. This is the quantity α_k used in Wave Drift Loads. Again this is simply the wave harmonic direction relative to the vessel surge axis, measured positive anti-clockwise from the positive axis direction. In strict mathematical terms the Relative vessel/wave heading is the angle from the positive direction of the local surge axis to the wave harmonic direction drawn at the CoG. In this the definition is slightly different to that used in defining RAOs, which is detailed in Vessels and Vessel Motions. There the incident wave heading for the definition of RAOs is defined as the angle between the wave direction drawn at the vessel reference point and the negative direction of the local surge axis.

Finally, it is important to define current and wind coefficients and QTFs over the full range of headings likely to be experienced in an analysis, as a value of zero is assumed for each of these parameters outside of the range of headings you specify. Headings values should ideally cover the full range from 0° to 360°. Likewise QTFs should be defined over a comprehensive range of frequencies, as a value of 0 will again be assigned to a QTF at a wave frequency outside of the range of values used in the data definition.

Floating Body

OVERVIEW

The floating body coupled analysis capability was developed by Wood (formerly MCS), with the support of Chevron, ConocoPhillips and ExxonMobil, as part of a joint industry project on coupled analysis. The background to the initiative stemmed from remote deepwater developments, particularly West of Africa, that have demonstrated the limitations of traditional, uncoupled, design methodologies for the design of production and offloading systems.

The main features of the capability are:

- Nonlinear time domain and linearised frequency domain coupled analysis capabilities
- First and second order forces on multiple floating bodies
- Viscous damping modelling
- Hydrodynamic coupling between adjacent floating bodies
- Interface to the WaMIT hydrodynamic analysis program

More recently Flexcom has been successfully utilised in the optimisation of wave energy devices. The numerical modelling capabilities have been validated via a benchmarking study with published data from the U.S. Department of Energy (Connolly et al., 2018), and comparisons with empirical data derived from model-scale tank test facilities (Connolly & Brewster, 2017).

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Applied Loading explains how each of the forces on the floating body are computed.

- Time Domain Analysis outlines the coupled analysis procedure in the time domain.
- Frequency Domain Analysis outlines the coupled analysis procedure in the frequency domain.
- Potential Flow Theory & Morison's Equation discusses the main differences between these hydrodynamic modelling approaches.
- Floating Body Modelling Detail discusses relatively simple (concentrated loads) and more complex approaches (distributed loads) for physically modelling the floating structure.
- Analysis Sequence presents a recommended sequence of steps for performing a floating body coupled analysis.
- WAMIT Interface describes the interface to the hydrodynamic radiation/diffraction analysis program WAMIT. Note that the WAMIT Interface is a legacy feature which has effectively been superseded by the newer Hydrodynamic Data Importer. This allows allows you to automatically import characteristic data relating to a Floating Body from a range of well-known hydrodynamic simulation packages.
- Input Formats describes the relevant input formats for the definition of force RAOs, the various force coefficients, added mass, radiation damping, viscous damping and hydrodynamic coupling coefficients.

RELEVANT KEYWORDS

- `*FLOATING BODY` is used to define a floating body and its associated properties.
- `*ADDED MASS` is used to define added mass for a floating body.
- `*RADIATION DAMPING` is used to define radiation damping for a floating body.
- `*FORCE RAO` is used to specify force RAOs for a floating body.
- `*WIND` is used to specify wind loading.
- `*WIND COEFF` is used to specify wind coefficients used to determine the wind loading on a floating body or moored vessel.
- `*CURRENT COEFF` is used to specify current coefficients used to determine the current loading on a floating body or moored vessel.

- `*QTF` is used to specify Quadratic Transfer Functions (QTFs) that allow the slow drift loads on a floating body or moored vessel to be determined.
- `*QTF CALIBRATION FB` is used to specify calibration coefficients used to scale the QTF coefficients for a floating body.
- `*VISCOUS DRAG` is used to define viscous drag for a floating body.
- `*HYDRODYNAMIC COUPLING` is used to define hydrodynamic coupling between adjacent floating bodies.
- `*WAMIT` is used to specify that Flexcom is to read floating body data from WAMIT output.
- `*WAVE-GENERAL` is used to specify miscellaneous parameters to wave loading. Specifically, the `FLOAT RAO FORCES`, `FLOAT QTF FORCES` and `FLOAT CONVOLUTION` options are used to provide greater user control over the application of first order forces, second order forces, and the convolution integral procedure itself.

If you would like to see an example of how these keywords are used in practice, refer to E02 - CALM Buoy - Complex.

Applied Loading

OVERVIEW

Hydrodynamic loading on the floating platform includes the various items listed below. The loads are computed for the floating body as a whole, and then applied at an appropriate location in the global force vector (e.g. at a node which corresponds to a centralised location such as the centre of gravity).

- First-Order Wave Loads (high frequency) derived from Force RAOs

- Wave Radiation Loads. An important issue arises with respect to the Added Mass and Radiation Damping terms associated with the floating body, and how these are modelled in the time domain. The frequency-dependent nature of these terms is accounted for using the established impulse response approach developed by Cummins (1962), and its implementation in Flexcom is described in detail by Connaire et al. (2003) and Lang et al. (2005). Specifically, the frequency-dependent damping term is replaced by a convolution integral of retardation functions and velocity time histories in the time domain.

- Viscous Damping Loads. These may be derived from centralised Viscous Damping Coefficients at some point on the floating body, or simulated in a distributed manner via Morison's Equation.

- Second-Order Wave Drift Loads (low frequency) derived from Quadratic Transfer Functions (QTFs)

- Current Loads computed via Current Coefficients

- Wind Loads computed via Wind Coefficients

- Hydrodynamic Loads determined according to the theory of manoeuvrability

FURTHER INFORMATION

- Potential Flow Theory & Morison's Equation discusses the main differences between these hydrodynamic modelling approaches.

- Floating Body Modelling Detail discusses relatively simple (concentrated loads) and more complex approaches (distributed loads) for physically modelling the floating structure.

THEORY

First-order wave forces due to the presence of Regular Airy Waves are computed as follows:

$$F_i(t) = \sum_{n=1}^{N} RAO_i(\theta,\omega)\ a_n\ \cos(-\omega_n t + k_n s_n + \varphi_n + \Phi_i(\theta,\omega)) \qquad (1)$$

where:

- $F_i(t)$ is the instantaneous force or moment in a particular degree of freedom, i (heave force, surge force, sway force, yaw moment, roll moment or pitch moment)
- t is the instantaneous solution time in seconds
- N is the number of regular Airy waves
- $RAO_i(\theta,\omega)$ is the RAO magnitude in degree of freedom i, which is a function of Incident Wave Heading θ and wave frequency ω
- a_n is the amplitude of the n^{th} regular wave (MWL to crest or trough)
- k_n is the wave number of the n^{th} regular wave ($k = 2\pi/\lambda$, where λ is wavelength)
- s_n is the horizontal distance from the global X axis to the vessel reference point in the direction of wave propagation for the n^{th} regular wave ($s_n = y \cos\theta_n + z \sin\theta_n$, where y and z are the coordinates of the vessel reference point in the global Y and Z axes respectively). These coordinates may be original, instantaneous, or computed using an average over a number of preceding time steps. Refer to the Reference Position input for further details.
- ω_n is the circular frequency of the n^{th} regular wave
- φ_n is the phase of the n^{th} regular wave relative to zero datum
- $\Phi_i(\theta,\omega)$ is the RAO phase angle in degree of freedom i, which is a function of wave heading θ and wave frequency ω. Note that a positive phase angle denotes a phase lag relative to the incident wave harmonic at the vessel reference point.

If there is only a single regular Airy wave present, then the summation from 1 to N in Equation 1 is obviously not required.

All random seas (including Pierson-Moskowitz, Jonswap, Ochi-Hubble, Torsethaugen and User-defined) are simulated using a series of individual component harmonics which are generated using a Spectrum Discretisation technique. Each harmonic is basically a regular Airy wave, so Equation 1 above remains valid.

RELEVANT KEYWORDS

- *FORCE RAO is used to specify force RAOs for a floating body.

If you would like to see an example of how this keyword is used in practice, refer to E02 - CALM Buoy - Complex.

RADIATION-DIFFRACTION

Morison's Equation represents a semi-empirical approach to hydrodynamics which is commonly used in marine engineering. It assumes that the force exerted by unbroken waves on a cylinder can be represented by a linear sum of drag and inertial terms. Morison's equation is widely established for modelling wave forces on slender offshore structures such as mooring lines. In situations where the body size becomes significant with respect to wavelength, the underlying assumptions become invalid, and the effects of radiation and diffraction must also be considered. Hence it is generally unsuitable for modelling arbitrary floating bodies. Refer to Potential Flow Theory & Morison's Equation for a discussion about the main differences between these hydrodynamic modelling approaches.

The velocity potential is classically decomposed into incident, diffraction and radiated potentials. Incident potential represents the wave excitation, diffraction represents disturbance of the wave induced fluid motion due to the presence of a fixed body, while the radiated potential represents the fluid motion caused by a moving body in still water. Radiation-diffraction programs solve the radiation and diffraction potentials, and provide coefficient terms, such as Added Mass, Radiation Damping and Force RAOs, at a range of discrete frequencies which may be readily inserted into Flexcom.

EQUATION OF MOTION

The general equation of motion for a floating structure in six degrees of freedom may be stated as follows:

$$\sum_{J-1}^{6}(\mathrm{M}_{\mathrm{kj}}+\mathrm{a}_{\mathrm{kj}})\,\ddot{\mathrm{x}}_{\mathrm{j}}+C_{\mathrm{kj}}\,\dot{\mathrm{x}}(\tau)+K_{\mathrm{kj}}\mathrm{x}_{\mathrm{j}}=\mathrm{F}_{\mathrm{k}}(\mathrm{t}) \qquad \mathrm{k}\ =1,\ 2\ \ldots\ldots \tag{1}$$

where:

- x_j is the displacement in the j^{th} degree of freedom

- Fk(t) is the dynamic external force in the k^{th} degree of freedom

- M is the inertia matrix
- a is the added inertia matrix (frequency dependent)
- C is the system damping matrix (frequency dependent)
- K is the hydrostatic stiffness matrix

CONVOLUTION INTEGRAL

A particular consideration arises in relation to the added mass and damping terms, and how these are treated in the time domain. In general these are not constant and vary as a function of frequency of response of the floating body. To deal with this problem Flexcom uses the technique put forward by Cummins (1962). In particular, the frequency dependant damping term is replaced by a convolution integral in the time domain, and the frequency dependent added mass is replaced with a constant value. Thus Equation (1) above is transformed to:

$$\sum_{J-1}^{6}(M_{kj} + m_{kj})\,\ddot{x}_j + \int_{-\infty}^{t} R_{kj}(t-\tau)\,\dot{x}(\tau)\,d\tau + K_{kj}x_j = F_k(t) \qquad k = 1,\ 2\ ....6$$

(2)

Here the frequency independent added inertia coefficients (m_{kj}) and the retardation functions (R_{kj}) can be computed from:

$$R_{kj}(t) = \frac{2}{\pi}\int_{0}^{\infty} C_{kj}(\omega)\cos\omega t\,d\omega \qquad (3)$$

$$m_{kj} = a_{kj}(\omega') + \frac{1}{\omega'}\int_{0}^{\infty} R_{kj}(\tau)\sin\omega' t\,d\tau \qquad (4)$$

The retardation functions are derived from the user-specified frequency dependant damping values and the coefficients of added inertia are derived from the user-specified added mass at a particular frequency (i.e. the reference frequency, ω').

Once the system of coupled differential equations is obtained, arbitrarily time-varying loads such as wave induced loads, current forces, non-potential fluid reaction forces and non-linear mooring forces may be incorporated as external force contributions. The final equation of motion to be solved is:

$$\sum_{J=1}^{6}(M_{kj} + m_{kj})\,\ddot{x}_j + K_{kj}x_j = F_k(t) - \sum_{j=1}^{6}\int_0^t R_{kj}(t-\tau)\,\dot{x}(\tau)\,d\tau \qquad k = 1,\ 2\ \ldots\ldots$$

(5)

Regular Wave Simulations

If your simulation contains a single regular wave only, the wave excitation occurs at a single frequency, so Flexcom does not typically compute the damping forces using a convolution integral of velocity time history and retardation functions. Instead it uses a constant 6x6 radiation damping matrix, either by selecting one directly from the frequency-dependent radiation damping data, or via interpolation if no data is available at that particular frequency. Should you prefer to use the convolution integral for some reason, perhaps to further investigate software behaviour, you can explicitly request that the convolution be adopted. Refer to *WAVE-GENERAL for further details. Note however that use of the convolution integral in such circumstances is unnecessary and less computationally efficient. Both approaches will produce very similar results, and while there may be some discrepancies in the early stages due to initial transience, results should be consistent once the simulation has reached steady-state.

RETARDATION FUNCTIONS

A number of issues are immediately apparent with respect to the evaluation of R(t). The user can supply a low frequency cut off value, below which C is assumed to be zero everywhere. The user also needs to supply an upper limit on time, above which R(t) should be very small in comparison to R(t=0). If a time limit is not specified, it defaults to 100s (or the total simulation time, if this is less than 100s). A time step also needs to be specified, (typically this will be in the range 0.05 to 0.1) so as the convolution integral in Equation (2) can be evaluated using the trapezoidal rule at each time step.

This leads directly to a critical issue. Equation (2) will be solved at each time step over the total simulation time period. As the convolution integral will have to be re-evaluated at each time step this will mean the solution time will increase in proportion to the square of the simulation time length. After the solution time has passed the maximum time used to define R(t), the total time needed to solve Equation (2) will increase at a constant rate.

With these issues in mind it is advisable that the time step used to define the complete curve for R(t) (in Equation (3) is an integer multiple of the time step used in the solution of Equation (2). If this is not the case it will be necessary at each time increment in the solution to interpolate the curve of R(t) so as the convolution integral can be evaluated. Further, it is therefore not recommended that variable time stepping be used when complex models of floating bodies are analysed.

Flexcom provides a facility whereby the computed retardation functions may be examined and verified by the user. Specifically, the computed functions are echoed to an ACSII file, entitled 'Ret_Fn_I.dat', where I is an integer value indicating the number of the relevant floating body. The layout of the output file is as follows:

Time, R_{11}, …, R_{22}, …, R_{33}, …, R_{44}, …, R_{55}, …, R_{66}

where:

- Time is the time value in seconds
- R_{11} is the heave retardation function value
- R_{22} is the surge retardation function value
- R_{33} is the sway retardation function value
- R_{44} is the yaw retardation function value
- R_{55} is the roll retardation function value
- R_{66} is the pitch retardation function value

and the intermediate terms (e.g. R12, R13, R23 etc.) represent the coupled retardation function terms between various degrees of freedom at the relevant time.

RELEVANT KEYWORDS

- *FLOATING BODY is used to define a floating body and its associated properties.
- *ADDED MASS is used to define added mass for a floating body.
- *RADIATION DAMPING is used to define radiation damping for a floating body.
- *FORCE RAO is used to specify force RAOs for a floating body.
- *WIND is used to specify wind loading.
- *WIND COEFF is used to specify wind coefficients used to determine the wind loading on a floating body or moored vessel.
- *CURRENT COEFF is used to specify current coefficients used to determine the current loading on a floating body or moored vessel.
- *QTF is used to specify Quadratic Transfer Functions (QTFs) that allow the slow drift loads on a floating body or moored vessel to be determined.
- *QTF CALIBRATION FB is used to specify calibration coefficients used to scale the QTF coefficients for a floating body.
- *VISCOUS DRAG is used to define viscous drag for a floating body.
- *HYDRODYNAMIC COUPLING is used to define hydrodynamic coupling between adjacent floating bodies.
- *WAVE-GENERAL is used to specify miscellaneous parameters to wave loading.

If you would like to see an example of how these keywords are used in practice, refer to E02 - CALM Buoy - Complex.

THEORY

At each solution step, viscous damping loads are computed from a matrix of damping coefficients and the relative fluid-structure velocity:

$$\vec{F}_d = \tfrac{1}{2}\,\rho\,(\underline{CA})\,\vec{v}\,\left|\vec{v}\right| \qquad (3)$$

where F_d is the vector of viscous damping forces and moments for all degrees of freedom, CA is the matrix of specified Viscous Damping Coefficients, ρ is the density of seawater, and v is the vector of relative fluid-structure velocity.

For the purpose of the damping force computation, the structure velocity is taken at the location of the floating body CoG, while the fluid velocity is computed at a horizontal location corresponding to the location of the floating body CoG and at an elevation corresponding to the Mean Water Line (MWL). An option is also provided to use either the relative fluid/structure velocity or the absolute structure velocity in the computations.

The damping force also includes coupled terms corresponding to the off-diagonal entries of the 6x6 drag coefficient matrix. In this way, it is possible to fully account for the effect of coupling between viscous damping loading and structure response in different degrees of freedom. The resulting forces and moments are applied to the floating body at the centre of drag.

RELEVANT KEYWORDS

- `*VISCOUS DRAG` is used to define viscous drag for a floating body.

THEORY

The wave drift loads are calculated from user-specified Quadratic Transfer Function (QTF) coefficients according to the following equation:

$$F_{DG}(t) = \left[\sum_{k=1}^{n} a_k \sqrt{\left|QTF_G(\alpha_k,\omega_k)\right|}\cos(\omega_k t + \phi_k)\right] \bullet \left[\sum_{k=1}^{n} a_k \sqrt{\left|QTF_G(\alpha_k,\omega_k)\right|}\cos(\omega_k t + \phi_k)\,\mathrm{sign}\left(QTF_G(\alpha_k,\omega_k)\right)\right] + \left[\sum_{k=1}^{n} a_k \sqrt{\left|QTF_G(\alpha_k,\omega_k)\right|}\sin(\omega_k t + \phi_k)\right] \bullet \left[\sum_{k=1}^{n} a_k \sqrt{\left|QTF_G(\alpha_k,\omega_k)\right|}\sin(\omega_k t + \phi_k)\,\mathrm{sign}\left(QTF_G(\alpha_k,\omega_k)\right)\right] \quad (1)$$

where:

- F_{DG} is one of the three wave drift loads applied to the body CoG, corresponding to F_{Dy} (surge), F_{Dz} (sway) or $M_{D\theta}$ (yaw).
- $QTF_G(\alpha_k,\omega_k)$ is the QTF value for relative wave incidence α_k (wave direction relative to the instantaneous body orientation) and frequency ω_k for the load direction in question
- α_k, ω_k and ϕ_k are the amplitude, frequency and phase respectively of the k^{th} wave harmonic
- sign(p) equals -1 if p < 0, 0 if p = 0, and 1 if p >0

The wave drift loads are calculated for every wave frequency present in the (discretised) wave spectrum and for every wave direction. A mean drift load is calculated in Static Mooring analyses to determine the mean body position, using the equation:

$$F_{DGmean} = 2\int_{\omega} QTF_G(\alpha_k,\omega_k)S_{\eta\eta}(\omega_k)d\omega \qquad (2)$$

where:

- F_{DGMean} is one of the three mean wave drift loads applied to the body CoG, corresponding to F_{DyMean} (surge), F_{DzMean} (sway) or $M_{D\theta Mean}$ (yaw).
- $S_{\eta\eta}(\omega_k)$ is the wave spectrum

and the remaining symbols have the same meanings as above.

RELEVANT KEYWORDS

- `*QTF` is used to specify Quadratic Transfer Functions (QTFs) that allow the second-order wave drift loads on a floating body to be determined.

If you would like to see an example of how this keyword is used in practice, refer to E02 - CALM Buoy - Complex.

THEORY

Current loads in surge (F_{Cy}), sway (F_{Cz}) and yaw ($M_{C\theta}$) are calculated as follows:

$$F_{Cy} = \frac{1}{2} C_{Cy}(\alpha_c) \rho_w T B u_c^2$$
$$F_{Cz} = \frac{1}{2} C_{Cz}(\alpha_c) \rho_w T L_{bp} u_c^2$$
$$M_{C\theta/G} = \frac{1}{2} C_{C\theta}(\alpha_c) \rho_w T L_{bp}^2 u_c^2 \qquad (1)$$

where:

- $C_{Cy}(\alpha_c)$, $C_{Cz}(\alpha_c)$ and $C_{C\theta}(\alpha_c)$ are the Current Coefficients in surge, sway and yaw respectively for an equivalent current incidence of α_c
- ρ_w is the water density
- T is the body draft
- B is the body beam
- L_{bp} is the length between perpendiculars which is the sum of y_{fore} and y_{aft}
- y_{fore} is the distance from the body centre of gravity to the forward perpendicular
- y_{aft} is the distance from the body centre of gravity to the aft perpendicular
- u_c is the equivalent current velocity

The equivalent current incidence α_c is obtained using:

$$\alpha_c = \tan^{-1}\left[\frac{v'}{u'}\right] \qquad (2)$$

and the equivalent current velocity u_c is given by:

$$u_c = \sqrt{u'^2 + v'^2} \qquad (3)$$

where u' and ν' are the components, in the body axis system, of the relative body/current velocity.

u' and ν' are given by the equations:

$$u' = u - v_c \cos(\beta_c - \theta)$$
$$v' = v - v_c \sin(\beta_c - \theta) \qquad (4)$$

where u and v are the components in the body axes of the body CoG velocity, v_c is the current velocity magnitude (which is a user input); and β_c is the current direction as defined in Definition of Wind and Current Directions.

RELEVANT KEYWORDS

- `*FLOATING BODY` is used to define a floating body and its associated properties.Specifically, inputs such as draft, beam and length are used in the computation of current loads.
- `*CURRENT` is used to specify current loading.
- `*CURRENT COEFF` is used to specify current coefficients used to determine the current loading on a floating body.

THEORY

Wind loads are calculated using:

$$F_{Wy} = \frac{1}{2} C_{Wy}(\alpha_w) \rho_a A_T u_w^2$$
$$F_{Wz} = \frac{1}{2} C_{Wz}(\alpha_w) \rho_a A_L u_w^2$$
$$M_{W\theta/G} = \frac{1}{2} C_{W\theta}(\alpha_w) \rho_a A_L L_{bp} u_w^2 \qquad (1)$$

where:

- $C_{Wy}(\alpha_w)$, $C_{Wz}(\alpha_w)$ and $C_{W\theta}(\alpha_w)$ are the Wind Coefficients in surge, sway and yaw respectively for an equivalent wind incidence of α_w
- ρ_a is the air density

- A_T is the transverse area of the body which is exposed to wind action
- A_L is the longitudinal area of the body which is exposed to wind action
- u_w is the equivalent wind velocity

The equivalent wind incidence α_w is obtained using:

$$\alpha_w = \tan^{-1}\left[\frac{v - v_w \sin(\beta_w - \theta)}{u - v_w \cos(\beta_w - \theta)}\right] \qquad (2)$$

where:

- v_w is the (global) wind velocity
- β_w is the wind direction as defined in Wind and Current Directions.

The equivalent wind velocity u_w is given by:

$$u_w = \sqrt{(u - v_w \cos(\beta_w - \theta))^2 + (v - v_w \sin(\beta_w - \theta))^2} \qquad (3)$$

RELEVANT KEYWORDS

- `*FLOATING BODY` is used to define a floating body and its associated properties. Specifically, inputs such as longitudinal and transverse areas are used in the computation of wind loads.
- `*WIND` is used to specify wind loading.
- `*WIND COEFF` is used to specify wind coefficients used to determine the wind loading on a floating body.

THEORY

The hydrodynamic loads, denoted F_{Hy} (surge), F_{Hz} (sway) and $M_{H\theta}$ (yaw), are determined according to the theory of manoeuvrability as follows:

$$
\begin{aligned}
F_{Hy} &= -Ma_{yy}\dot{u}' + Ma_{zz}v'\dot{\theta} + Ma_{z\theta}\dot{\theta}^2 \\
F_{Hz} &= -Ma_{zz}\dot{v}' - Ma_{yy}u'\dot{\theta} - Ma_{z\theta}\ddot{\theta} \\
M_{H\theta/G} &= -Ma_{\theta\theta}\ddot{\theta} - Ma_{z\theta}\dot{v}' - Ma_{z\theta}u'\dot{\theta} - \left(Ma_{zz} - Ma_{yy}\right)u'v'
\end{aligned}
\qquad (1)
$$

where:

- Ma_{yy} is the body added mass in surge
- Ma_{zz} is the body added mass in sway
- $Ma_{\theta\theta}$ is the body added mass in yaw
- $Ma_{z\theta}$ is the body added mass in sway-yaw coupling
- θ is the instantaneous body orientation as defined in Local Vessel Axis System
- u' and ν' are the components, in the body axis system, of the relative body/current velocity.

u' and ν' are given by the equations:

$$
\begin{aligned}
u' &= u - v_c \cos(\beta_c - \theta) \\
v' &= v - v_c \sin(\beta_c - \theta)
\end{aligned}
\qquad (2)
$$

where u and ν are the components in the body axes of the body CoG velocity; ν_c is the current velocity magnitude (which is a user input); and β_c is the current direction as defined in Definition of Wind and Current Directions. Differentiating u' and ν' with respect to time (assuming a constant current velocity) gives:

$$
\begin{aligned}
\dot{u}' &= \dot{u} - v_c\dot{\theta}\sin(\beta_c - \theta) \\
\dot{v}' &= \dot{v} + v_c\dot{\theta}\cos(\beta_c - \theta)
\end{aligned}
\qquad (3)
$$

Substituting the above expressions into the equations for the hydrodynamic loads gives the following:

$$
\begin{aligned}
F_{Hy} &= -Ma_{yy}\dot{u} + Ma_{yy}v_c\dot{\theta}\sin(\beta_c - \theta) + Ma_{zz}v'\dot{\theta} + Ma_{z\theta}\dot{\theta}^2 \\
F_{Hz} &= -Ma_{zz}\dot{v} - Ma_{zz}v_c\dot{\theta}\cos(\beta_c - \theta) - Ma_{yy}u'\dot{\theta} - Ma_{z\theta}\ddot{\theta} \\
M_{H\theta/G} &= -Ma_{\theta\theta}\ddot{\theta} - Ma_{z\theta}\dot{v} - Ma_{z\theta}v_c\dot{\theta}\cos(\beta_c - \theta) - Ma_{z\theta}u'\dot{\theta} - (Ma_{zz} - Ma_{yy})u'v'
\end{aligned}
$$

(4)

For numerical reasons it is desirable to move any terms consisting of the product of a mass by a structure acceleration term to the left-hand side of the overall matrix equations of motion solved by Flexcom. So Term (1) in F_{Hy}, Terms (1) & (4) in F_{Hz}, and Terms (1) & (2) in $M_{H\theta}$ are moved in this way. This is achieved by adding the relevant body added mass terms to the mass matrix at the location corresponding to the body centre of gravity. The following expressions are then used to calculate the hydrodynamic forces which are applied at the body CoG:

$$
\begin{aligned}
F_{Hy} &= Ma_{yy}v_c\dot{\theta}\sin(\beta_c - \theta) + Ma_{zz}v'\dot{\theta} + Ma_{z\theta}\dot{\theta}^2 \\
F_{Hz} &= -Ma_{zz}v_c\dot{\theta}\cos(\beta_c - \theta) - Ma_{yy}u'\dot{\theta} \\
M_{H\theta/G} &= -Ma_{z\theta}v_c\dot{\theta}\cos(\beta_c - \theta) - Ma_{z\theta}u'\dot{\theta} - (Ma_{zz} - Ma_{yy})u'v'
\end{aligned}
\qquad (5)
$$

RELEVANT KEYWORDS

- `*ADDED MASS` is used to define added mass for a floating body.
- `*CURRENT` is used to specify current loading.

If you would like to see an example of how these keywords are used in practice, refer to E02 - CALM Buoy - Complex.

Time Domain Analysis

THEORY

Each floating body is modelled as a rigid body with six degrees of freedom that are included amongst the solution variables. The hydrostatic stiffness contribution associated with the floating body is included in the global stiffness matrix at the location of the floating body Centre of Buoyancy (CoB). The (frequency independent) mass and inertia of the floating body is included in the global mass matrix at the location corresponding to the floating body Centre of Gravity (CoG). Mechanical loads from connected structures (such as mooring lines, risers, transfer lines etc.) are automatically transmitted to the floating body by means of its inclusion in the finite element model.

During time domain dynamic analyses, static, low frequency and wave frequency loads on the floating body are computed and accumulated at the appropriate location in the global force vector. Further details are provided in Applied Loading.

The position of the floating body is solved for at each timestep, subject to the full range of environmental loading and mechanical loads from connected structures.

Hydrodynamic coupling between pairs of floating bodies is specified by means of defining co-influence added mass and radiation damping matrices for the two floating bodies. In the case of time domain dynamic analyses, the off-diagonal terms of the co-influence matrices give rise to additional off-diagonal retardation functions in the matrix of retardation functions, and additional off-diagonal terms in the frequency-independent added mass matrix.

As discussed in Wave Radiation Loads, frequency-dependent radiation damping forces are computed via a convolution integral of velocity time history and retardation functions. This would be the standard modelling procedure for random sea analyses in the time domain. However, there are certain circumstances where a different approach is adopted. For example, Flexcom does not generally compute the damping forces in this manner if the wave loading consists of a single regular wave only, even if frequency-dependent radiation damping data has been specified. Typically the program finds a (constant) radiation damping matrix (by interpolation if necessary) corresponding to the single regular wave frequency, and adds this matrix at an appropriate location on the left-hand side of the equations of motion. However, you have the option to override this default behaviour, and compel Flexcom to perform the convolution calculation, in which case the radiation damping becomes a force on the right-hand side of the equations of motion as before.

Frequency Domain Analysis

THEORY

The underlying principles of analysis in the frequency domain are essentially the same for both traditional de-coupled analyses and coupled analyses. These principles are described in Frequency Domain Analysis, which summarises the theory underpinning frequency domain analysis. In this section, the important differences with respect to coupled analyses (as opposed to decoupled analyses) are described.

One of the major differences between the coupled analysis procedure in the time and frequency domains concerns the treatment of the frequency dependent added mass and radiation damping terms. In frequency domain analyses, the system response is solved for at each component harmonic in the spectrum of excitation forces. Hence, when solving for the response at a particular frequency, the appropriate frequency-dependent added mass and radiation damping terms for the floating body are simply incorporated in the global mass and damping matrices at the appropriate locations. Thus the solution fully accounts for the frequency-dependent nature of the added mass and radiation damping terms. Note that at frequencies below the user-specified low frequency damping cut-off frequency, the floating body added mass and damping terms correspond to the user-specified low frequency damping values.

First order (wave-frequency) loads are applied to each floating body by means of force RAOs, while mean (static) current and wind loads are applied to each floating body in the static phases of the frequency domain analysis.

If appropriate, hydrodynamic coupling between floating bodies is included in the frequency domain analysis by incorporation of the relevant terms from the added mass and damping co-influence matrices in the global mass and damping matrices respectively.

An important consideration is the treatment of low frequency second-order loads in frequency domain irregular sea coupled analyses. As second-order loads are inherently non-linear, a linearisation technique must be employed before these loads can be incorporated in frequency domain dynamic analyses. The procedure employed by Flexcom is as follows. First, the wave spectrum is discretised in the usual manner (using either the Equal Area or Geometric Progression discretisation procedures, as discussed in Spectrum Discretisation). Next, Flexcom identifies each frequency pair in the discretised seastate, and from them computes all difference frequencies. Note that, in general, there will be a significantly larger number of difference frequencies present than there are component harmonics in the discretised wave spectrum; in a wave spectrum with n component harmonics, there will be ½ n (n-1) frequency pairs. Each difference frequency is then added to the list of frequencies at which the equations of motion of the coupled system are to be solved in the irregular sea analysis. The magnitude of the harmonic excitation at each difference frequency is computed from the appropriate QTF values and component wave amplitudes. These are the QTFs and component wave amplitudes corresponding to the two frequencies in the frequency pair from which the difference frequency is derived. The magnitude of the low-frequency harmonic excitation is multiplied by the user-specified wave drift force calibration coefficients and the resulting low frequency harmonic excitation load applied to each floating body at each difference frequency for which the equations of motion are being solved.

Due to the non-linear nature of the floating body viscous damping loads, it is necessary to linearise these loads before they can be included in frequency domain dynamic analyses. The linearisation technique employed for the floating body viscous damping loads is based on the proven techniques currently used in Flexcom to linearise the Morison's Equation drag force. These techniques are described in detail in Frequency Domain Analysis.

Potential Flow Theory and Morison's Equation

POTENTIAL FLOW THEORY

Broadly speaking, two different modelling techniques are generally used to model hydrodynamic loads on a floating structure, Potential Flow Theory and Morison's Equation. Potential flow theory is based on the evaluation of pressure integrals around the body surface, and is applicable in situations where the floating body is relatively large compared to the wave length of the incident seastate, as the wave field is disturbed by the presence of the submerged structure. This type of model is quite accurate in the sense that it will capture wave excitation forces (including diffraction) and radiation loads (including added mass and damping). However it does have some limitations:

- It does not account for viscous drag loading on the structure resulting from flow separation.
- Radiation and diffraction potentials are often solved in the frequency domain assuming a linearised boundary for reasons of computational efficiency. This technique assumes that displacements of the free surface and the floating body away from their mean positions remain relatively small, which simplifies the wave-structure interactions significantly, but the downside is that this simplification results in forces which vary linearly with wave amplitude.
- When used in conjunction with structural codes like Flexcom, global hydrodynamic loads are typically concentrated at a single location such as the centre of gravity, rather than being distributed over the wetted surface area of the body. Radiation-diffraction codes readily provide the total loads in terms of coefficient matrices so this is the most convenient means of transferring loads to the structural model. In theory it is possible to apply distributed loads, but this requires more detailed information regarding the load distribution, and the subsequent mapping of these forces to relevant locations on the structural model.

Potential flow theory is widely accepted industry for marine renewables, particularly for operational sea states where wave heights are moderate.

MORISON'S EQUATION

Morison’s equation is normally used for smaller structures. For example, for cylindrical structures in regular waves, it is generally deemed acceptable if the diameter is less than one-fifth of the wave length. Morison’s equation is an empirically derived hydrodynamic loading model which includes the wave excitation force, and added mass term, and viscous drag forces. Although relatively simplistic in that it completely ignores wave field disturbance, it has some advantages in that:

- Viscous drag forces may be modelled.
- Applied loads may be computed based on the instantaneous location and orientation of the structure, and wave forces may readily be integrated up to the instantaneous free surface elevation using a wave stretching algorithm.
- Applied loading is automatically distributed to various locations around the floating body, given that it is modelled using a mesh of slender beam elements, and this allows the spatial nature of applied loads to be modelled.

These factors result in the application of higher order non-linear loads on a floating body.

Note also that one of the main challenges associated with the Morison approach is the selection of appropriate hydrodynamic coefficients. Guidelines are available based on Reynolds number and surface roughness, but these relate to simple cylinders only. If you are using Morison to model a more complex shape, or indeed a collection of cylinders (e.g. a semi-submersible platform is effectively comprised of a series of columns, pontoons and cross-braces), then the choice of appropriate hydrodynamic coefficients is far from trivial. It may be possible to fine-tune selected values by benchmarking a Morison simulation with a corresponding potential flow model or experimental data.

WHICH APPROACH SHOULD YOU USE?

The applicability of potential flow theory or Morison’s equation is generally assessed via dimensionless parameters, such as the Keulegan-Carpenter number, the Reynolds number and the diameter-to-wavelength ratio. These parameters define the relative importance of inertia, diffraction, and drag for different flow regions. There is a large body of research material available in this area, and a quick internet search for “regions of validity of potential theory and morison’s equation” will reveal some further helpful background reading.

Many popular simulation tools such as Flexcom allow users to adopt a combined approach, whereby the radiation and diffraction components are modelled using potential flow theory, which is then supplemented by the inclusion of viscous drag terms derived from Morison's equation.

The next article on Floating Body Modelling Detail is related to this one. It discusses relatively simple (concentrated loads) and more complex approaches (distributed loads) for physically modelling the floating structure.

Floating Body Modelling Detail

OVERVIEW

There are varying levels of detail with which you can physically model the floating structure. The simplest possible option is to place finite element nodes only at key points of interest, such as the centres of gravity and buoyancy, and to model all the applied loads as concentrated forces at these points. Conversely, it is also possible to model the floating body in explicit detail, including finite elements to model the various pontoons, decks, braces etc. which comprise the real-world structure. Typically the simple approach is adopted by most users, but there are some advantages associated with creating a more detailed model. Both methods are now briefly summarised and contrasted.

The preceding article on Potential Flow Theory & Morison's Equation is related to this one. It discusses the main differences between these hydrodynamic modelling approaches.

SIMPLE MODEL

Finite element nodes are placed at key locations only, typically corresponding to...

- Platform centre of gravity (CoG)
- Platform centre of buoyancy (CoB)
- Platform RAO reference point (CoF)
- Platform centre of drag (CoD)
- Fairlead positions of attached mooring lines.

This effectively serves as a framework upon which the various constituents may be applied. All of the Applied Loads are concentrated at these key locations. The elements are defined using the Flexible Format as this affords greatest control over the assigned properties.

- The floating structure is assumed to act as a perfectly rigid body, so the nodes are connected together via elements of relatively high stiffness (e.g. 1.0E+10 N.m^2).
- The inertia of the floating body is concentrated at the CoG node, so the finite elements are assigned zero Mass per Unit Length. The inertial terms are specified via the Floating Body Inertia Inputs.
- The buoyancy force on the floating body is concentrated at the CoB node, so the finite elements are assigned zero Buoyancy Diameter. The effects of buoyancy are modelled using the Floating Body Hydrodynamic Stiffness Inputs.
- Regarding the previous two points, it is important to note that gravitational and mean buoyancy forces, both static, are not explicitly simulated. It is assumed that the initial position and orientation of floating body represents the static equilibrium configuration. Assuming equilibrium conditions, the static forces cancel each other out. The dynamic inertia of the floating body is correctly modelled in a dynamic simulation via the inclusion of the inertial terms in the global mass matrix. Similarly, the dynamically changing buoyancy force is modelled correctly via the hydrostatic stiffness terms which simulate changes in buoyancy force due to displacements and rotations of the floating body away from its mean position.
- Generally speaking, the wave excitation forces on the floating structure are determined using a diffraction solver. These forces are produced using a frequency domain solution technique and are provided in a concentrated form at a single point on the body. Diffraction codes readily provide the total loads so this is the most convenient means of transferring loads to the structural model. This location is identified as the RAO Reference Node in the Flexcom model.
- The overall drag force on the floating body is concentrated at the CoD node, so the finite elements are assigned zero Drag Diameter. The effects of viscous drag are modelled using the Floating Body Viscous Drag Inputs.
- As the element assembly looks rather skeleton-like, and does not remotely resemble the real world structure, the addition of an auxiliary Vessel Profile is advisable. While this does not have any structural function, it greatly enhances the visual appeal of the model, and assists in the understanding of floating body motions post-simulation.

COMPLEX MODEL

The floating body is modelled in explicit detail using an assemblage of finite element nodes and elements which approximate the physical structural profile. The coordinates of the nodes, and the connectivity of the structural model, would have a similar level of detail to that used to create the optional auxiliary profile for the simple model described above. Having this detailed framework in place facilitates the application of distributed rather than concentrated loads, for some or all of the constituent terms. Again it seems logical to define the elements using the Flexible Format as this affords greatest control over the assigned properties. The following are some advantages of this approach.

- Elasticity/flexibility of the floating body may be captured. So the elements should be assigned appropriate structural stiffness inputs rather than arbitrarily high stiffness values (particularly in bending). In practice however, most floating bodies are effectively rigid so the choice of inputs is unlikely to have any major impact on the overall solution. Note also that hydrodynamic loading would need to be applied in a distributed manner. Assuming potential flow theory is being used, the wave excitation forces come from a diffraction solver. Wave pressures would need to be integrated over the hydrodynamic panels and the derived forces mapped to relevant locations in Flexcom model. An alternative approach would be to use Morison’s Equation rather than the diffraction approach. This is much easier to do, but the theory is not well suited to large structures like FOWTs where wave field disturbance is a feature. Refer to Potential Flow Theory and Morison's Equation for some background information.

- Structural mass can be modelled in a distributed manner. In this case, the elements should be assigned appropriate Mass per Unit Length terms, and the inertia of the floating body itself (specified via the Floating Body Inertia Inputs) should be set to zero. In a similar vein to the previous point, the floating body is effectively rigid, so there will be little or no difference between concentrated or distributed mass modelling. Note that if a distributed mass approach is adopted, then a distributed buoyancy approach (discussed in the next point) should also be adopted for consistency. This will ensure that both aspects inherently include static/mean and dynamic terms and avoid any complications arising from force imbalances.

- Buoyancy forces can be modelled in a distributed manner. This is a more accurate modelling technique which allows non-linear variations in buoyancy (e.g. due to significant pitch motions) to be simulated. In this case, the elements should be assigned appropriate Buoyancy Diameter terms, and the buoyancy associated with the floating body itself (specified via the Floating Body Hydrodynamic Stiffness Inputs) should be set to zero. Note also that if a concentrated mass approach is being used in conjunction with a distributed buoyancy approach, then the gravitational (static) force due to the floating body must be applied separately as a Point Load at the CoG node. This is necessary in order to ensure a balance of forces and static equilibrium.

- Viscous drag forces may be applied in a distributed manner. This is a more accurate modelling technique which allows non-linear variations in drag (e.g. due to intermittent submergence) to be simulated. In this case, the elements should be assigned appropriate Drag Diameter terms (with loading to computed via Morison's Equation), and the drag associated with the floating body itself (specified via the Floating Body Viscous Drag Inputs) should be set to zero.

Analysis Sequence

The following sequence of steps would be recommended for analysing a complex multi-component model for a particular combination of waves, current, wind and other user-defined loading:

- Step 1: In an initial static analysis, apply buoyancy and gravity loads, any forces critical to the stability of the system (such as a riser top tension), zero constant displacements (such as at riser and mooring line seabed connections), while restraining the vessel CoG node in all translational degrees of freedom except the vertical.

- Step 2: In a static restart from Step 1, release all restraints at the vessel CoG node.

- Step 3: Complete any offset and/or decay test analyses being performed to verify the system response in a restart or series of restarts from Step 2.

- Step 4: Apply all remaining static loads such as those due to wind and current, again in a restart from Step 2. If the dynamic response to a random wave is to be simulated in a subsequent step, include the wave spectrum definition in the input data for this step; Flexcom will then apply the static mean drift force in this step.

- Step 5: Simulate the dynamic response of the system to the wave environment in a dynamic restart from Step 4 in either the time or the frequency domain.

Some key points in the rationale for performing analyses in the above sequence of steps are as follows:

- Step 1 allows the build up of tension in the various lines of the model (mooring lines, risers, TLP tendons if present, etc.), while enabling the vessel to find its vertical equilibrium position. Building up tensions in this way is generally mandatory for the types of model Flexcom is routinely used to analyse, because so many components of such systems depend largely on tension to resist applied loads.

- Step 2 then allows the vessel to find its horizontal equilibrium position. In some cases it may be possible to combine Steps 1 and 2 in a single static step. On the other hand, for some sensitive systems it may be necessary to perform the Step 2 analysis quasi-statically.

- A quasi-static Flexcom analysis is one in which a static configuration is obtained from a highly damped dynamic analysis. In such analyses, high mass and stiffness damping coefficients are specified, and these have the effect of dissipating structural dynamics resulting from the release of the CoG restraints. The analysis continues until the structure settles into a static configuration, at which point all loads and displacements are constant.

- The Step 3 analysis or analyses, if performed, are mainly verification or validation exercises. However they do have the advantage that they increase the engineer’s understanding of the behaviour of the overall system, and offer the opportunity to correct, refine and/or improve the Flexcom as appropriate, prior to the later static and dynamic stages.

- The Step 4 analysis allows all of the remaining static loads to be added to the model.

- Preceding a frequency domain dynamic analysis by some or all of Steps 1-4 is mandatory (a frequency domain analysis in Flexcom must restart from a converged static solution).

- The advantage of the multi-step procedure for time domain dynamic analysis is that starting a dynamic run from a stable static configuration generally minimises the influence of initial transients associated with the application of the dynamic loads, with a consequent saving in CPU effort. In addition, there are of course no transients generated by the addition of static loads, since these are already present at the start of the dynamic phase.

- Finally, using the sequence of steps allows for the possibility of restarting multiple dynamic runs from the same static configuration; you might want to look at a range of wave periods, amplitudes, directions and even wave specification types (regular or random). There is an obvious saving in restarting all of these from the same static configuration, rather than repeating all static stages in every dynamic run.

WAMIT Interface

PREFACE

Note that the WAMIT Interface is a legacy feature which has effectively been superseded by the newer Hydrodynamic Data Importer. This allows allows you to automatically import characteristic data relating to a Floating Body from a range of well-known hydrodynamic simulation packages.

THEORY

Flexcom provides an interface to the hydrodynamic radiation/diffraction analysis program WAMIT. This allows you to specify that certain floating body properties (such as wave frequency force RAOs, added mass terms etc.) are read from the output file of a WAMIT analysis, thereby eliminating the scope for error in manually converting these parameters to the format required for input to Flexcom. To invoke this feature, you simply specify the name of each floating body (which is used to uniquely identify the floating body) and the path and filename of the WAMIT output file corresponding to each floating body. Note that all analysis input data, including floating body properties read and converted from a WAMIT output file, are echoed to the Flexcom output file, allowing you to visually inspect the converted data.

The data that Flexcom retrieves for the case of one floating body only is as follows:

- Hydrostatic stiffnesses

- Added mass coefficients

- Radiation damping coefficients
- Force RAOs
- QTFs

The data that Flexcom retrieves for the case of two floating bodies is as follows:

- Hydrostatic stiffnesses for both bodies
- Added mass coefficients, including co-influence terms
- Radiation damping coefficients, including co-influence terms
- Force RAOs for both bodies
- QTFs for both bodies

WAMIT calculates force RAOs (or "exciting force" in the program terminology) using two different methods termed "Exciting forces from Haskind's relations" and "Exciting forces from direct integration of dynamic pressures". You have the choice of nominating which values to use as force RAOs. Choosing *Radiation* as the *RAO Computation* option via the `*WAMIT` keyword corresponds to the first of these methods, choosing *Diffraction* corresponds to the second.

If the same data is available to Flexcom from two sources, namely your Flexcom input data and a WAMIT output file, then Flexcom gives precedence to the WAMIT data. For example, your Flexcom input data might include hydrostatic stiffness terms, and might also include WAMIT interface data, meaning that Flexcom had two sources of (possibly different) hydrostatic stiffness values. In such an event, Flexcom uses the values retrieved from the WAMIT output file. A warning to this effect will be contained in the Flexcom output data.

RELEVANT KEYWORDS

- `*WAMIT` is used to specify that Flexcom is to read floating body data from WAMIT output.

Input Formats

Each of the following are input using ASCII files.

- Force RAOs

- Wind force coefficients
- Current force coefficients
- QTF coefficients
- QTF calibration coefficients
- Added mass coefficients
- Radiation damping coefficients
- Viscous damping coefficients
- Hydrodynamic coupling coefficients

All comprise of two types of lines, data lines and comment lines. The letter C (uppercase essential) in Column 1 identifies a comment line. Comment lines (which can be included at any point in the file) are completely ignored by Flexcom, and are intended to allow you to include comments for your own benefit or for other users. Any line that is not a comment line is a data line. You can input data in free format, as floating point numbers with or without exponent. Use an uppercase E when specifying an exponent. Either commas or blanks may separate a number of values on a particular data line; do not use TABs.

Regarding the input conventions in Flexcom:

- Direction and Heading Conventions describes the conventions used in the definition of current, wind and wave loading with respect to the local body axis system.

THEORY

Flexcom supports two different schemes for laying out the data in the RAO file, namely the MCS layout and the Line layout. This former is the standard layout traditionally used by Flexcom. The Line layout is a more recent addition to the software, and it represents a more general layout that, for example, simplifies copying and pasting RAO data from spreadsheet programs.

RAO data comprises two types of lines, a data line and a comment line. The letter C (uppercase essential) in Column 1 identifies a comment line. Comment lines are completely ignored by Flexcom, and are intended to allow you to include comments in an RAO file for other users or to assist in later scrutiny. Comment lines can be included at any point in the RAO file.

Any line that is not a comment line is a data line. The actual data values expected on a data line are, of course, a function of position within the file. Numerical inputs on a data line are in free format, and can be specified as floating point numbers with or without exponent. Use an uppercase E when specifying an exponent. Either commas or blanks may separate a number of values on a particular data line; do not use TABs.

The MCS layout begins with a single line defining the incident wave heading at which the RAOs are being defined. This is followed by a block of three lines specifying the incident wave frequency and the relevant RAO amplitudes and phases for this heading and wave frequency, as shown below. Note that entries shown below in italics should be replaced with their actual numeric values in the RAO file.

```
HEADING=Wave Heading
Wave Frequency
Heave RAO, Surge RAO, Sway RAO, Yaw RAO, Roll RAO, Pitch RAO
Heave Phase, Surge Phase, Sway Phase, Yaw Phase, Roll Phase, Pitch Phase
```

The block of three lines specifying the incident wave frequency, RAOs and phases are repeated as often as necessary until the RAOs are defined over the required range of wave frequencies. To define RAOs at more incident wave headings, you simply repeat the HEADING= line for the new wave heading, and the RAO data for this wave heading is specified as before. If RAOs are being defined for more than one incident wave heading, then they must also be defined for more than one frequency at each wave heading, and the number of frequencies at which the RAOs are defined is the same for each heading. If the RAOs are independent of the incident wave heading, then the HEADING= line should be omitted.

The Line layout is very similar to the standard MCS layout, with the exception that the RAO and phase data for a particular heading and frequency appear on a single line, the format of which is shown below. Note that, for clarity, the data shown below is split over a number of lines, but is in reality specified on a single line of the RAO file.

```
Wave Heading, Wave Frequency, Heave RAO, Heave Phase, Surge RAO, Surge Phase,
Yaw RAO, Yaw Phase, Roll RAO, Roll Phase, Pitch RAO, Pitch Phase
```

This line is repeated for every wave frequency and every wave heading for which RAOs are being defined. As with the MCS layout, if RAOs are being defined at more than one incident wave heading, then they must also be defined at more than one frequency at each wave heading, and the number of frequencies at which the RAOs are defined is the same for each heading. So, for example, if RAOs were to be defined at three wave headings and ten frequencies per heading, then 30 lines of data would be required to specify the RAOs using the Line layout. If the same wave heading is specified on all lines, then Flexcom assumes that the RAOs are independent of wave heading. Note also that the order in which the lines of RAO data appear is not important. Flexcom automatically sorts the RAO data by heading and frequency.

RELEVANT KEYWORDS

- `*FORCE RAO` is used to specify force RAOs for a floating body.

If you would like to see an example of how this keyword is used in practice, refer to E02 - CALM Buoy - Complex.

THEORY

Each line of data in a current coefficient file has the following format:

Relative floating body/current heading, Cc_y, Cc_z, Cc_θ

This line is repeated until current coefficient values are specified for all headings. Relative floating body/current heading in this context is defined as the angle of incidence of the current relative to the floating body axes; refer to Definition of Wind and Current Directions for a more detailed definition. Cc_y, Cc_z, and Cc_θ are respectively the current coefficients in surge, sway and yaw at a particular heading.

RELEVANT KEYWORDS

- `*CURRENT COEFF` is used to specify current coefficients used to determine the current loading on a floating body or moored vessel.

If you would like to see an example of how this keyword is used in practice, refer to D01 - Moored Vessel.

THEORY

Each line of data in a wind coefficient file has the following format:

Relative floating body/wind heading, Cw_y, Cw_z, Cw_θ

This line is repeated until wind coefficient values are specified for all headings. Relative floating body/wind heading in this context is defined as the angle of incidence of the wind relative to the floating body axes; refer to Definition of Wind and Current Directions for a more detailed definition. Cw_y, Cw_z, and Cw_θ are respectively the wind coefficients in surge, sway and yaw at a particular heading.

RELEVANT KEYWORDS

- `*WIND COEFF` is used to specify wind coefficients used to determine the wind loading on a floating body or moored vessel.

If you would like to see an example of how this keyword is used in practice, refer to D01 - Moored Vessel.

THEORY

The QTF data is divided into groups, with each group providing data for a particular wave frequency. Each group consists of the following lines:

FREQ=Wave frequency (in Hz)

Relative floating body/wave heading, QTF_y, QTF_z, QTF_θ

Line 2 is repeated until QTF values are specified for all headings for this wave frequency. Then a second FREQ= line signals the start of the next group of values, and so on until values for all wave frequencies have been input. Relative floating body/wave heading in this context is defined as the angle of incidence of the wave relative to the floating body axes; refer to Definition of Wave Heading for QTF Data for a more detailed definition. QTF_y, QTF_z and QTF_θ are respectively the QTF values in surge, sway and yaw at a particular frequency.

RELEVANT KEYWORDS

- `*QTF` is used to specify Quadratic Transfer Functions (QTFs) that allow the slow drift loads on a floating body or moored vessel to be determined.

If you would like to see an example of how this keyword is used in practice, refer to D01 - Moored Vessel.

THEORY

Each line of data in a QTF calibration coefficient file has the following format:

Relative floating body/wave heading, QTFCALy, QTFCALz, $QTFCAL_{\theta}$

This line is repeated until QTF calibration coefficient values are specified for all headings. Relative floating body/wave heading in this context is defined as the angle of incidence of the wave relative to the floating body axes; refer to Definition of Wave Heading for QTF Data for a more detailed definition. QTFCALy, QTFCALz and $QTFCAL_{\theta}$ are respectively the QTF calibration values in surge, sway and yaw at a particular heading.

RELEVANT KEYWORDS

- `*QTF CALIBRATION FB` is used to specify calibration coefficients used to scale the QTF coefficients for a floating body.

THEORY

The layout of the added mass terms is shown below and this can be repeated as often as necessary to define the added mass over a sufficiently large range of frequencies:

FREQ= Frequency

$$
\begin{array}{l}
A11, A12, A13, A14, A15, A16 \\
A21, A22, A23, A24, A25, A26 \\
A31, A32, A33, A34, A35, A36 \\
A41, A42, A43, A44, A45, A46 \\
A51, A52, A53, A54, A55, A56 \\
A61, A62, A63, A64, A65, A66
\end{array}
$$

where:

- Frequency is the frequency value in Hz, to which the coefficients relate
- A_{11} is the surge added mass at the relevant frequency
- A_{22} is the sway added mass at the relevant frequency
- A_{33} is the heave added mass at the relevant frequency
- A_{44} is the roll added mass at the relevant frequency
- A_{55} is the pitch added mass at the relevant frequency
- A_{66} is the yaw added mass at the relevant frequency

and the remaining terms represent the added mass terms between various degrees of freedom at the relevant frequency.

RELEVANT KEYWORDS

- `*ADDED MASS` is used to define added mass for a floating body.

If you would like to see an example of how this keyword is used in practice, refer to E02 - CALM Buoy - Complex.

THEORY

The layout of the radiation damping terms is shown below and this can be repeated as often as necessary to define the radiation damping over a sufficiently large range of frequencies:

FREQ= Frequency

$$
\begin{array}{l}
C11, C12, C13, C14, C15, C16 \\
C21, C22, C23, C24, C25, C26 \\
C31, C32, C33, C34, C35, C36 \\
C41, C42, C43, C44, C45, C46 \\
C51, C52, C53, C54, C55, C56 \\
C61, C62, C63, C64, C65, C66
\end{array}
$$

where:

- Frequency is the frequency value in Hz, to which the coefficients relate
- C_{11} is the surge radiation damping at the relevant frequency
- C_{22} is the sway radiation damping at the relevant frequency
- C_{33} is the heave radiation damping at the relevant frequency
- C_{44} is the roll radiation damping at the relevant frequency
- C_{55} is the pitch radiation damping at the relevant frequency
- C_{66} is the yaw radiation damping at the relevant frequency

and the remaining terms represent the radiation damping terms between various degrees of freedom at the relevant frequency.

RELEVANT KEYWORDS

- `*RADIATION DAMPING` is used to define radiation damping for a floating body.

If you would like to see an example of how this keyword is used in practice, refer to E02 - CALM Buoy - Complex.

THEORY

The layout of the viscous damping terms is shown below. Each input corresponds to a viscous damping coefficient times an appropriate area term related to the dimensions of the floating body.

$$\begin{matrix} V11, V12, V13, V14, V15, V16 \\ V21, V22, V23, V24, V25, V26 \\ V31, V32, V33, V34, V35, V36 \\ V41, V42, V43, V44, V45, V46 \\ V51, V52, V53, V54, V55, V56 \\ V61, V62, V63, V64, V65, V66 \end{matrix}$$

For example, V_{11} is the viscous damping coefficient in surge times the projected floating body area in the local surge direction.

RELEVANT KEYWORDS

- *VISCOUS DRAG is used to define viscous drag for a floating body.

THEORY

The layout of the added mass coupling terms is shown below and this can be repeated as often as necessary to define the coupling over a sufficiently large range of frequencies:

FREQ= Frequency

$$\begin{bmatrix} A11, A12, A13, A14, A15, A16 \\ A21, A22, A23, A24, A25, A26 \\ A31, A32, A33, A34, A35, A36 \\ A41, A42, A43, A44, A45, A46 \\ A51, A52, A53, A54, A55, A56 \\ A61, A62, A63, A64, A65, A66 \end{bmatrix}_{1 \rightarrow 2}$$

$$\begin{bmatrix} A11, A12, A13, A14, A15, A16 \\ A21, A22, A23, A24, A25, A26 \\ A31, A32, A33, A34, A35, A36 \\ A41, A42, A43, A44, A45, A46 \\ A51, A52, A53, A54, A55, A56 \\ A61, A62, A63, A64, A65, A66 \end{bmatrix}_{2 \rightarrow 1}$$

where:

- Frequency is the frequency value in Hz, to which the coefficients relate
- A_{11} is the surge coupling between the floating bodies at the relevant frequency
- A_{22} is the sway coupling between the floating bodies at the relevant frequency
- A_{33} is the heave coupling between the floating bodies at the relevant frequency
- A_{44} is the roll coupling between the floating bodies at the relevant frequency
- A_{55} is the pitch coupling between the floating bodies at the relevant frequency
- A_{66} is the yaw coupling between the floating bodies at the relevant frequency

and the remaining terms represent the added mass coupling terms between various degrees of freedom at the relevant frequency. The first subscript denotes that the coupling influence is of the first floating body on the second, and vice versa for the second subscript.

The layout of the radiation damping terms is shown below and this can be repeated as often as necessary to define the coupling over a sufficiently large range of frequencies:

FREQ= *Frequency*

$$\begin{bmatrix} \mathrm{C11, C12, C13, C14, C15, C16} \\ \mathrm{C21, C22, C23, C24, C25, C26} \\ \mathrm{C31, C32, C33, C34, C35, C36} \\ \mathrm{C41, C42, C43, C44, C45, C46} \\ \mathrm{C51, C52, C53, C54, C55, C56} \\ \mathrm{C61, C62, C63, C64, C65, C66} \end{bmatrix}_{1 \to 2}$$

$$\begin{bmatrix} \mathrm{C11, C12, C13, C14, C15, C16} \\ \mathrm{C21, C22, C23, C24, C25, C26} \\ \mathrm{C31, C32, C33, C34, C35, C36} \\ \mathrm{C41, C42, C43, C44, C45, C46} \\ \mathrm{C51, C52, C53, C54, C55, C56} \\ \mathrm{C61, C62, C63, C64, C65, C66} \end{bmatrix}_{2 \to 1}$$

where:

- Frequency is the frequency value in Hz, to which the coefficients relate
- C_{11} is the surge coupling between the floating bodies at the relevant frequency
- C_{22} is the sway coupling between the floating bodies at the relevant frequency
- C_{33} is the heave coupling between the floating bodies at the relevant frequency
- C_{44} is the roll coupling between the floating bodies at the relevant frequency
- C_{55} is the pitch coupling between the floating bodies at the relevant frequency
- C_{66} is the yaw coupling between the floating bodies at the relevant frequency

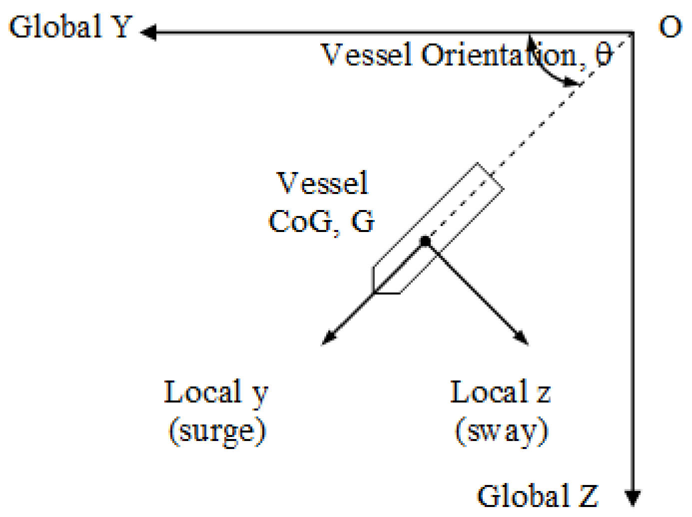
and the remaining terms represent the added mass coupling terms between various degrees of freedom at the relevant frequency. The first subscript denotes that the coupling influence is of the first floating body on the second, and vice versa for the second subscript.

RELEVANT KEYWORDS

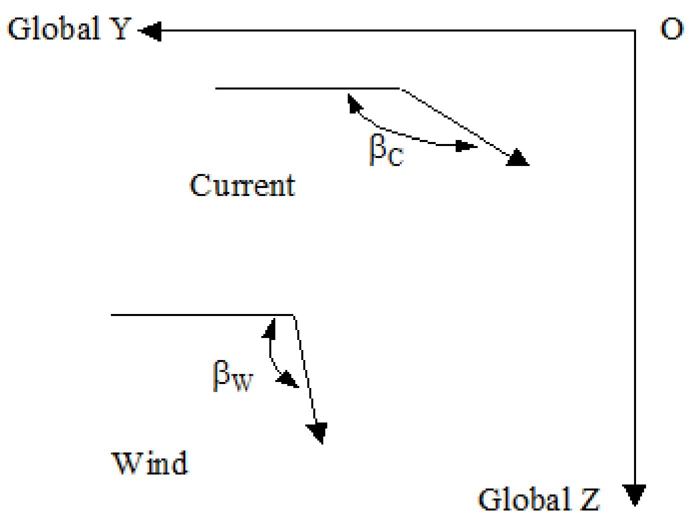
- `*HYDRODYNAMIC COUPLING` is used to define hydrodynamic coupling between adjacent floating bodies.

CURRENT AND WIND DIRECTIONS

The loads are calculated in a vessel axis system defined with its origin at the vessel centre of gravity, G, as shown in the figure below. The vessel y axis defines the local surge axis, and the z axis the local sway axis. The instantaneous vessel orientation, θ, is measured positive anticlockwise from the global Y axis.



Local Vessel Axis System

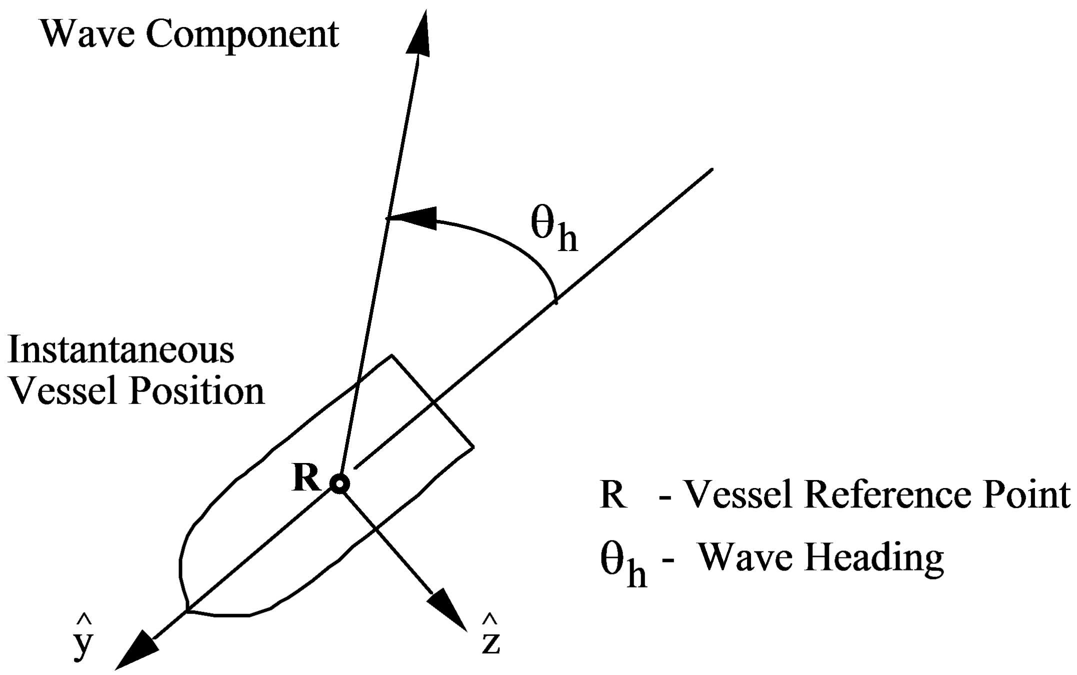


Definition of Wind and Current Directions

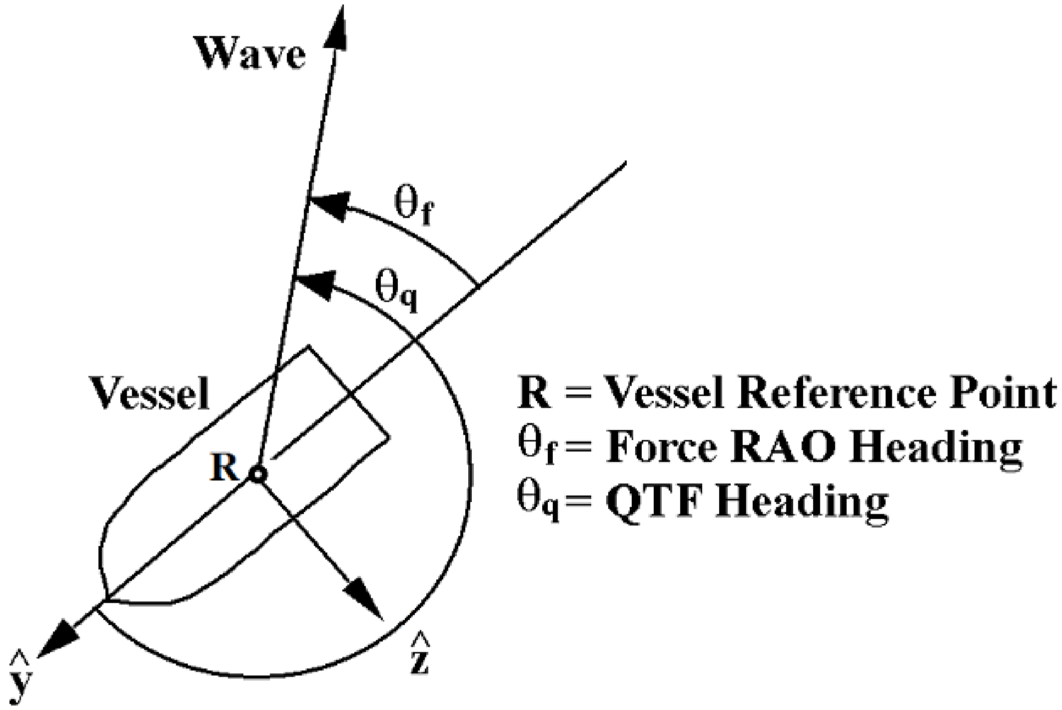
WAVE HEADING

Vessels and Vessel Motions provided the Flexcom definition of wave heading appropriate to the input of first order vessel motion (or displacement) RAOs. As outlined in Calculating Vessel RAO Response, heading in this context is defined as the angle (positive anticlockwise) from the negative direction of the vessel local surge axis to the wave direction drawn at the vessel reference point. This is shown in the first figure below. In defining floating body vessel data, this definition of wave heading also applies to the input of first order force RAOs.

The definition of wave heading (or strictly speaking the relative floating body/wave heading) used in conjunction with QTFs is slightly different. In this case, heading is measured anti-clockwise from the positive direction of the local surge axis to the wave. This definition is contrasted with the definition for first order motions and forces in the second figure below.



Definition of Wave Heading for RAO Data



Definition of Wave Heading for QTF Data

The definition of heading for the input of coefficients for current (Relative floating body/current heading) and wind (Relative floating body/wind heading) calculations is consistent with the definition used for QTFs, as it was for moored vessel data. See the Vessel Heading section for full details.

1.9.3.8 User Subroutines

OVERVIEW

Flexcom now provides the following advanced user subroutines:

- User Defined Element (this option allows you, for example, to vary an element's structural properties while a simulation is in progress)
- User Solver Variables (this option allows you, for example, to directly augment the global force vector to simulate an arbitrary time-varying load)

These user subroutines provide much greater flexibility and control, as distinguished from legacy features.

BACKGROUND

Flexcom has traditionally provided a number of basic user subroutine options to accommodate specialised loading such as unusual current profiles (e.g. solitons), arbitrary loading (e.g. load time histories) and boundary conditions (e.g. load time histories).

- User-Defined Currents
- User-Defined Forces
- User-Defined Boundary Conditions

These options are somewhat limited in the sense that the arbitrary loading is largely prescribed beforehand, with little or no dependence on instantaneous solution parameters which evolve as the simulation progresses.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- DLL Compilation
- Subroutine Format

RELEVANT KEYWORDS

- `*USER DEFINED ELEMENT` provides advanced Flexcom users with the ability to define custom code for altering element properties.
- `*USER SOLVER VARIABLES` provides advanced Flexcom users with the ability to define custom code for increased modelling flexibility.

User Defined Element

OVERVIEW

Using this feature, you may redefine element properties (such as stiffness, mass, diameter and drag coefficients) by creating your own custom subroutine. You have access to a range of solution variables including nodal kinematic variables (displacements, velocities, accelerations) and elemental restoring forces (effective tension, bending moment, curvature, torque etc.). Equipped with this information it is then possible to directly change elemental properties while a simulation is in progress. The modified element parameters are passed back to Flexcom from this subroutine at the beginning of every solution step. Use of this feature requires some programming expertise, but it provides complete generality for power users to change elemental properties at any instant during the simulation. Some illustrative examples are provided to help upskill users and demonstrate the potential of this powerful modelling feature.

RELEVANT KEYWORDS

- `*USER DEFINED ELEMENT` provides advanced Flexcom users with the ability to define custom code for altering element properties.

The DLL subroutine interface is defined in the Subroutine Format section. If you would like to see an example of how this keyword is used in practice, refer to J01 - Dropped Object and Recovery, in particular, the recovery of the dropped object which uses a User Defined Element subroutine to redefine the buoyancy diameter of the lift balloon.

User Solver Variables

OVERVIEW

Flexcom now provides advanced user subroutines, denoted as ‘user solver variables’ to distinguish from legacy features, which provide much greater flexibility and control. The user is empowered by the ability to access a range of solution variables including nodal kinematic variables (displacements, velocities, accelerations) and elemental restoring forces (effective tension, bending moment, curvature, torque etc.). Equipped with this information it is then possible to directly augment the global force vector to simulate an arbitrary time-varying load. It is even possible to directly modify the constitutive finite element matrices (i.e. stiffness and mass) should you have very specialised modelling requirements. Naturally this requires some programming expertise, but it provides complete generality for power users. Some illustrative examples are provided to help upskill users and demonstrate the potential of this powerful modelling feature.

RELEVANT KEYWORDS

- `*USER SOLVER VARIABLES` provides advanced Flexcom users with the ability to define custom code for increased modelling flexibility.

The DLL subroutine interface is defined in the Subroutine Format section. If you would like to see an example of how this keyword is used in practice, refer to J04 - User Solver Variables.

DLL Compilation

Any generic compiler may be used to compile a DLL (dynamic link library) which may be linked into Flexcom. When you invoke a user subroutine option, you must have written the relevant source code, and compiled it into a DLL, before you perform the actual Flexcom analysis. Note also that there are standard templates provided in the Flexcom installation folder which illustrate how to create DLLs.

When you specify the name of the DLL file, Flexcom loads the DLL and searches for the subroutine that was embedded into the DLL when it was compiled. Once Flexcom locates the subroutine, the subroutine is dynamically loaded into the Flexcom analysis module executable. The analysis then proceeds. At run-time the user-subroutine is called once for every solution iteration. The call is located just prior to the solution of the finite element equations of motion, where you have access to all the assembled matrices and constituent variables.

Finally, when the analysis is completed the temporary Flexcom executable is deleted. Note that if you use the facility, then the operation is completely automatic and transparent to you, and requires no intervention by you at runtime in compiling or linking the program.

Subroutine Format

Blank or template FORTRAN listings of the `user_defined_element` and `user_solver_variables` subroutines are shown below. The structure of these subroutines will be familiar to experienced FORTRAN programmers, but is quite straightforward even for beginners.

- The `subroutine` statement at the top opens the routine. This contains a list of arguments which are used to share information between the custom subroutine and the main program source code.
- The `implicit none` statement ensures that all variables are explicitly defined.
- The variable declaration section lists the program arguments and their data types. Array variables are also sized appropriately.
- Comments, starting with an exclamation (`!`) symbol, are used to improve code clarity.
- The `end subroutine` statement closes the routine.

You should insert your own custom code towards the end of the template, just after the "`Insert user code here`" statement.

User Defined Element Template

```
subroutine user_defined_element(iter,time,ramp,depth,rhow,nnode,nncon,ncord,nndof,
     mxpipnod,n0kpipnod,maxre,num_int,elmcon,nodcon,intoun,ietoue,etype,edamp,corc
     vel_prev,acc_prev,cdnn,cdtt,cann,catt,cmnn,reno,nore,pip_reaction,axial_force
     z_curvature,length,eiyy,eizz,gj,ea,mass,polar,dint,ddrag,dbuoy,dout,dcont,flu
```

```
!dec$ attributes dllexport, stdcall, reference :: user_defined_element
implicit none

! Input variables - cannot be modified within this subroutine


integer(4), intent(in) :: iter        !< Current iteration
!dec$ attributes value :: iter
real(8), intent(in)    :: time        !< Current timestep
!dec$ attributes value :: time
real(8), intent(in)    :: ramp        !< Ramp
!dec$ attributes value :: ramp
real(8), intent(in)    :: depth       !< Water depth
!dec$ attributes value :: depth
real(8), intent(in)    :: rhow        !< Water density
!dec$ attributes value :: rhow
integer(4), intent(in) :: nnode       !< Number of nodes in the model.
!dec$ attributes value :: nnode
integer(4), intent(in) :: nncon       !< Number of nodes with connected elemen
!dec$ attributes value :: nncon
integer(4), intent(in) :: ncord       !< Number of coordinates
!dec$ attributes value :: ncord
integer(4), intent(in) :: nndof       !< Number of degrees of freedom per node
!dec$ attributes value :: nndof
integer(4), intent(in) :: nelmn       !< Number of elements in the model.
!dec$ attributes value :: nelmn
integer(4), intent(in) :: necon       !< Number of elements connected
!dec$ attributes value :: necon
intege      r(4), intent(in) :: nedof       !< Number of degrees of freedom pe
!dec$ attributes value :: nedof
integer(4), intent(in) :: ndamp       !< Number of damper elements
!dec$ attributes value :: ndamp
integer(4), intent(in) :: npipcurv    !< No. of pipe-in-pipe curves.
!dec$ attributes value :: npipcurv
integer(4), intent(in) :: npippts     !< No. of pipe-in-pipe curve points.
!dec$ attributes value :: npippts
integer(4), intent(in) :: mxpipnod    !< Max. no. of pipe-in-pipe connecting n
!dec$ attributes value :: mxpipnod
integer(4), intent(in) :: n0kpipnod   !< No. of zero-stiffness sliding pipe-in
!dec$ attributes value :: n0kpipnod
integer(4), intent(in) :: maxre       !< Maximum Reynolds' number array size.
!dec$ attributes value :: maxre
integer(4), intent(in) :: num_int     !< Number of integration points per elem
!dec$ attributes value :: num_int

integer(4), intent(in), dimension(nncon, nelmn) :: elmcon        !< Element c
!dec$ attributes reference :: elmcon
integer(4), intent(in), dimension(necon,nnode)  :: nodcon        !< Node conn
!dec$ attributes reference :: nodcon
integer(4), intent(in), dimension(nnode)        :: intoun        !< Internal
!dec$ attributes reference :: intoun
integer(4), intent(in), dimension(nelmn)        :: ietoue        !< Internal
```

```
!dec$ attributes reference :: ietoue
integer(4),  intent(in), dimension(nelmn)       :: etype        !< Element t
!dec$ attributes reference :: etype
integer(4),  intent(in), dimension(nelmn)       :: edamp        !< Damper el
!dec$ attributes reference :: edamp
real(8),   intent(in), dimension(ncord,nnode)   :: cord         !< Initial n
!dec$ attributes reference :: cord
real(8),   intent(in), dimension(nndof,nnode)   :: displacement !< Nodal dis
!dec$ attributes reference :: displacement
real(8),   intent(in), dimension(nndof,nnode)   :: velocity     !< Nodal vel
!dec$ attributes reference :: velocity
real(8),   intent(in), dimension(nndof,nnode)   :: acceleration !< Nodal acc
!dec$ attributes reference :: acceleration
real(8),   intent(in), dimension(3,3,nelmn)     :: trgb         !< Rigid bod
!dec$ attributes reference :: trgb
real(8),   intent(in), dimension(3,3,nelmn)     :: tglu         !< Global to
!dec$ attributes reference :: tglu
real(8),   intent(in), dimension(nndof,nnode)   :: disp_prev    !< Nodal dis
!dec$ attributes reference :: disp_prev
real(8),   intent(in), dimension(nndof,nnode)   :: pos_prev     !< Nodal pos
!dec$ attributes reference :: pos_prev
real(8),   intent(in), dimension(nndof,nnode)   :: vel_prev     !< Nodal vel
!dec$ attributes reference :: vel_prev
real(8),   intent(in), dimension(nndof,nnode)   :: acc_prev     !< Nodal acc
!dec$ attributes reference :: acc_prev
real(8),   intent(in), dimension(nelmn,3)       :: axial_force  !< Axial for
!dec$ attributes reference :: axial_force
real(8),   intent(in), dimension(nelmn,3)       :: y_shear      !< Y Shear f
!dec$ attributes reference :: y_shear
real(8),   intent(in), dimension(nelmn,3)       :: z_shear      !< Z Shear f
!dec$ attributes reference :: z_shear
real(8),   intent(in), dimension(nelmn,3)       :: torque       !< Torque in
!dec$ attributes reference :: torque
real(8),   intent(in), dimension(nelmn,3)       :: y_bending    !< Y bending
!dec$ attributes reference :: y_bending
real(8),   intent(in), dimension(nelmn,3)       :: z_bending    !< Z bending
!dec$ attributes reference :: z_bending
real(8),   intent(in), dimension(nelmn,3)       :: eff_tension  !< Effective
!dec$ attributes reference :: eff_tension
real(8),   intent(in), dimension(nelmn,3)       :: y_curvature  !< Y curvatu
!dec$ attributes reference :: y_curvature
real(8),   intent(in), dimension(nelmn,3)       :: z_curvature  !< Z curvatu
!dec$ attributes reference :: z_curvature

! Output variables - can be modified within this subroutine if required

real(8),   intent(inout), dimension(nelmn)      :: length       !< Element r
!dec$ attributes reference :: length
real(8),   intent(inout), dimension(nelmn)      :: eiyy         !< Element I
!dec$ attributes reference :: eiyy
real(8),   intent(inout), dimension(nelmn)      :: eizz         !< Element I
!dec$ attributes reference :: eizz
real(8),   intent(inout), dimension(nelmn)      :: gj           !< Element I
```

```
!dec$ attributes reference :: gj
real(8),  intent(inout), dimension(nelmn)       :: ea           !< Element l
!dec$ attributes reference :: ea
real(8),  intent(inout), dimension(nelmn)       :: mass         !< Element m
!dec$ attributes reference :: mass
real(8),  intent(inout), dimension(nelmn)       :: polar        !< Element p
!dec$ attributes reference :: polar
real(8),  intent(inout), dimension(nelmn)       :: dint         !< Element i
!dec$ attributes reference :: dint
real(8),  intent(inout), dimension(nelmn)       :: ddrag        !< Element c
!dec$ attributes reference :: ddrag
real(8),  intent(inout), dimension(nelmn)       :: dbuoy        !< Element b
!dec$ attributes reference :: dbuoy
real(8),  intent(inout), dimension(nelmn)       :: dout         !< Element c
!dec$ attributes reference :: dout
real(8),  intent(inout), dimension(nelmn)       :: dcont        !< Element c
!dec$ attributes reference :: dcont
real(8),  intent(inout), dimension(nelmn,4)     :: fluid        !< Element f
!dec$ attributes reference :: fluid
real(8),  intent(inout), dimension(ndamp,4)     :: damper       !< Damper el
!dec$ attributes reference :: damper

real(8),  intent(inout), dimension(maxre,num_int,nelmn)  :: cdnn !< Drag coef
!dec$ attributes reference :: cdnn
real(8),  intent(inout), dimension(maxre,nelmn)          :: cdtt !< Drag coef
!dec$ attributes reference :: cdtt
real(8),  intent(inout), dimension(maxre,nelmn)          :: cann !< Added mas
!dec$ attributes reference :: cann
real(8),  intent(inout), dimension(maxre,nelmn)          :: catt !< Added mas
!dec$ attributes reference :: catt
real(8),  intent(inout), dimension(maxre,nelmn)          :: cmnn !< Inertia c
!dec$ attributes reference :: cmnn
real(8),  intent(inout), dimension(maxre,nelmn)          :: reno !< Reynolds'
!dec$ attributes reference :: reno
integer(4),  intent(inout), dimension(maxre,nelmn)       :: nore !< Number of
!dec$ attributes reference :: nore

real(8),  intent(in), dimension(mxpipnod-n0kpipnod,3)    :: pip_reaction
!dec$ attributes reference :: pip_reaction

real(8),  intent(inout), dimension(mxpipnod-n0kpipnod,3) :: pip_stiffness_dat
!dec$ attributes reference :: pip_stiffness_data

real(8),  intent(inout), dimension(npipcurv,npippts,2)   :: pip_curve_defn
!dec$ attributes reference :: pip_curve_defn

integer(4),  intent(inout), dimension(npipcurv)          :: pip_curve_npts
!dec$ attributes reference :: pip_curve_npts

! Variable names
!     iter               : Current iteration
!     time               : Current timestep
!     ramp               : Ramp
```

```
!     depth                 : Water depth
!     rhow                  : Water density
!     nnode                 : Number of nodes in the model
!     nncon                 : Number of nodes with connected elements
!     ncord                 : Number of coordinates
!     nndof                 : Number of degrees of freedom per node (6)
!     nelmn                 : Number of elements in the model
!     necon                 : Number of elements connected
!     nedof                 : Number of degrees of freedom per element (14)
!     ndamp                 : Number of damper elements
!     npipcurv              : Number of pipe-in-pipe curves
!     npippts               : Number of points in each pipe-in-pipe curve
!     mxpipnod              : Number of pipe-in-pipe nodes
!     n0kpipnod             : Number of pipe-in-pipe zero stiffness nodes
!     elmcon                : Element connectivity array
!     nodcon                : Node connectivity array
!     intoun                : Internal node to user node numbering array
!     ietoue                : Internal element to user element numbering array
!     etype                 : Element type - (1) Beam, (2) Spring, (3) Hinge, (4
!     edamp                 : Damper element number array
!     cord                  : Initial nodal co-ordinates
!     displacement          : Nodal displacements at previous iteration
!     velocity              : Nodal velocities at previous iteration
!     acceleration          : Nodal accelerations at previous iteration
!     trgb                  : Rigid body rotation (local undeformed -> convected
!     tglu                  : Global to local undeformed transformation matrix
!     disp_prev             : Nodal displacements at previous timestep
!     pos_prev              : Nodal positions at previous timestep
!     vel_prev              : Nodal velocities at previous timestep
!     acc_prev              : Nodal accelerations at previous timestep
!     axial_force           : Axial force in elements at previous timestep
!     y_shear               : Y Shear forces in elements at previous timestep
!     z_shear               : Z Shear forces in elements at previous timestep
!     torque                : Torque in elements at previous timestep
!     y_bending             : Y bending moments in elements at previous timestep
!     z_bending             : Z bending moments in elements at previous timestep
!     eff_tension           : Effective Tension in elements at previous timestep
!     y_curvature           : Y curvatures in elements at previous timestep
!     z_curvature           : Z curvatures in elements at previous timestep
!     length                : Element natural length
!     eiyy                  : Element linear EIyy
!     eizz                  : Element linear EIzz
!     gj                    : Element linear GJ
!     ea                    : Element linear EA
!     mass                  : Element mass per unit length
!     polar                 : Element polar inertia per unit length
!     dint                  : Element internal diameters
!     ddrag                 : Element drag diameters
!     dbuoy                 : Element buoyancy diameters
!     dout                  : Element outer diameters
!     dcont                 : Element contact diameters
!     fluid                 : Element fluid contents -
!                               (1) Top elevation (fluid head)
```

```
!                             (2) Density
!                             (3) Internal pressure
!                             (4) Velocity
!     damper              : Element contact diameters
!                             (1) C0
!                             (2) C1
!                             (3) C2
!                             (4) C0_Threshold
!     pip_reaction            (1) - Axial reaction.
!                             (2) - Normal Reaction.
!                             (3) - Transverse Reaction.
!     pip_stiffness_data      (1) - Axial Stiffness.
!                             (2) - Normal Stiffness.
!                             (3) - Transverse Stiffness.
!     pip_curve_defn          Each point on the a data pairs curve is describe
!                             (1) - Displacement.
!                             (2) - Force.
!                             In the case of a Power Law curve, there is only
!                             (1) - Exponent.
!                             (2) - Contact force.


! Declare local variables...


! Insert user code below this line...

  end subroutine user_defined_element
```

User Solver Variables Template

```
subroutine user_solver_variables(iter,time,ramp,depth,rhow,nnode,nncon,ncord,nndof,nelmn,r
     size_force,size_mass,size_stiff,mxpipnod,n0kpipnod,elmcon,nodcon,intoun,ietoue,cord,
     velocity,acceleration,trgb,tglu,disp_prev,pos_prev,vel_prev,acc_prev,pip_reaction,ax
     z_shear,torque,y_bending,z_bending,eff_tension,y_curvature,z_curvature,force,mass,st
   !dec$ attributes dllexport, stdcall, reference :: user_solver_variables
     implicit none

     integer, intent(in)     :: iter          !< Current iteration
     !dec$ attributes value :: iter
     real(8), intent(in)     :: time          !< Current timestep
     !dec$ attributes value :: time
     real(8), intent(in)     :: ramp          !< Ramp
     !dec$ attributes value :: ramp
     real(8), intent(inout) :: depth          !< Water depth
     !dec$ attributes reference :: depth
     real(8), intent(inout) :: rhow           !< Water density
     !dec$ attributes reference :: rhow
     integer(4), intent(in) :: nnode          !< Number of nodes in the model.
     !dec$ attributes value :: nnode
     integer(4), intent(in) :: nncon          !< Number of nodes with connected elements
     !dec$ attributes value :: nncon
     integer(4), intent(in) :: ncord          !< Number of coordinates
     !dec$ attributes value :: ncord
     integer(4), intent(in) :: nndof          !< Number of degrees of freedom per node (6)
     !dec$ attributes value :: nndof
     integer(4), intent(in) :: nelmn          !< Number of elements in the model.
     !dec$ attributes value :: nelmn
     integer(4), intent(in) :: necon          !< Number of elements connected
     !dec$ attributes value :: necon
     integer(4), intent(in) :: nedof          !< Number of degrees of freedom per element (1
     !dec$ attributes value :: nedof
     integer(4), intent(in) :: size_force     !< Dimension of the global force vector
     !dec$ attributes value :: size_force
     integer(4), intent(in) :: size_mass      !< Dimension of the global mass matrix
     !dec$ attributes value :: size_mass
     integer(4), intent(in) :: size_stiff     !< Dimension of the global stiffness matrix
     !dec$ attributes value :: size_stiff
     integer(4), intent(in) :: mxpipnod       !< Max. no. of pipe-in-pipe connecting node pa
     !dec$ attributes value :: mxpipnod
     integer(4), intent(in) :: n0kpipnod      !< No. of zero-stiffness sliding pipe-in-pipe
     !dec$ attributes value :: n0kpipnod

     integer(4), intent(in), dimension(nncon, nelmn)           :: elmcon        !< Ele
     !dec$ attributes reference :: elmcon
     integer(4), intent(in), dimension(necon,nnode)            :: nodcon        !< Nod
     !dec$ attributes reference :: nodcon
     integer(4), intent(in), dimension(nnode)                  :: intoun        !< Int
     !dec$ attributes reference :: intoun
     integer(4), intent(in), dimension(nelmn)                  :: ietoue        !< Int
     !dec$ attributes reference :: ietoue
     real(8),   intent(in), dimension(ncord,nnode)             :: cord          !< Ini
     !dec$ attributes reference :: cord
     real(8),   intent(in), dimension(nndof,nnode)             :: displacement  !< Nod
```

```
!dec$ attributes reference :: displacement
real(8),  intent(in), dimension(nndof,nnode)             :: velocity     !< Nod
!dec$ attributes reference :: velocity
real(8),  intent(in), dimension(nndof,nnode)             :: acceleration !< Nod
!dec$ attributes reference :: acceleration
real(8),  intent(in), dimension(3,3,nelmn)               :: trgb         !< Rig
!dec$ attributes reference :: trgb
real(8),  intent(in), dimension(3,3,nelmn)               :: tglu         !< Glo
!dec$ attributes reference :: tglu
real(8),  intent(in), dimension(nndof,nnode)             :: disp_prev    !< Nod
!dec$ attributes reference :: disp_prev
real(8),  intent(in), dimension(nndof,nnode)             :: pos_prev     !< Nod
!dec$ attributes reference :: pos_prev
real(8),  intent(in), dimension(nndof,nnode)             :: vel_prev     !< Nod
!dec$ attributes reference :: vel_prev
real(8),  intent(in), dimension(nndof,nnode)             :: acc_prev     !< Nod
!dec$ attributes reference :: acc_prev
real(8),  intent(in), dimension(nelmn,3)                 :: axial_force  !< Axi
!dec$ attributes reference :: axial_force
real(8),  intent(in), dimension(nelmn,3)                 :: y_shear      !< Y S
!dec$ attributes reference :: y_shear
real(8),  intent(in), dimension(nelmn,3)                 :: z_shear      !< Z S
!dec$ attributes reference :: z_shear
real(8),  intent(in), dimension(nelmn,3)                 :: torque       !< Tor
!dec$ attributes reference :: torque
real(8),  intent(in), dimension(nelmn,3)                 :: y_bending    !< Y b
!dec$ attributes reference :: y_bending
real(8),  intent(in), dimension(nelmn,3)                 :: z_bending    !< Z b
!dec$ attributes reference :: z_bending
real(8),  intent(in), dimension(nelmn,3)                 :: eff_tension  !< Eff
!dec$ attributes reference :: eff_tension
real(8),  intent(in), dimension(nelmn,3)                 :: y_curvature  !< Y c
!dec$ attributes reference :: y_curvature
real(8),  intent(in), dimension(nelmn,3)                 :: z_curvature  !< Z c
!dec$ attributes reference :: z_curvature

real(8),  intent(inout), dimension(size_force,nedof)     :: force        !< Glo
!dec$ attributes reference :: force
real(8),  intent(inout), dimension(nedof,nedof,size_mass) :: mass        !< Glo
!dec$ attributes reference :: mass
real(8),  intent(inout), dimension(nedof,nedof,size_stiff) :: stiff      !< Glo
!dec$ attributes reference :: stiff

real(8),  intent(in), dimension(mxpipnod-n0kpipnod,3)    :: pip_reaction !< The
!dec$ attributes reference :: pip_reaction                               !!
                                                                         !< 1 -
                                                                         !< 2 -
                                                                         !< 3 -

  ! Variable names
!     iter                 : Current iteration
!     time                 : Current timestep
!     ramp                 : Ramp
```

```
!     nnode               : Number of nodes in the model
!     nncon               : Number of nodes with connected elements
!     ncord               : Number of coordinates
!     nndof               : Number of degrees of freedom per node (6)
!     nelmn               : Number of elements in the model
!     necon               : Number of elements connected
!     nedof               : Number of degrees of freedom per element (14)
!     size_force          : Dimension of the global force vector
!     size_mass           : Dimension of the global mass matrix
!     size_stiff          : Dimension of the global stiffness matrix
!     elmcon              : Element connectivity array
!     nodcon              : Node connectivity array
!     intoun              : Internal node to user node numbering array
!     ietoue              : Internal element to user element numbering array
!     cord                : Initial nodal co-ordinates
!     displacement        : Nodal displacements at previous iteration
!     velocity            : Nodal velocities at previous iteration
!     acceleration        : Nodal accelerations at previous iteration
!     trgb                : Rigid body rotation (local undeformed -> convected) transfor
!     tglu                : Global to local undeformed transformation matrix
!     disp_prev           : Nodal displacements at previous timestep
!     pos_prev            : Nodal positions at previous timestep
!     vel_prev            : Nodal velocities at previous timestep
!     acc_prev            : Nodal accelerations at previous timestep
!     axial_force         : Axial force in elements at previous timestep
!     y_shear             : Y Shear forces in elements at previous timestep
!     z_shear             : Z Shear forces in elements at previous timestep
!     torque              : Torque in elements at previous timestep
!     y_bending           : Y bending moments in elements at previous timestep
!     z_bending           : Z bending moments in elements at previous timestep
!     eff_tension         : Effective Tension in elements at previous timestep
!     y_curvature         : Y curvatures in elements at previous timestep
!     z_curvature         : Z curvatures in elements at previous timestep
!     force               : Global force vector at previous iteration
!     mass                : Global mass matrix.
!     stiff               : Global stiffness matrix.

! Declare local variables...

! Insert user code below this line...

 end subroutine user_solver_variables
```

1.9.3.9 Wind Turbine Modelling

INTRODUCTION

Flexcom Wind is Wood's dedicated simulator package for floating offshore wind turbines.

Based on our existing marine engineering and software development skills, Wood identified an opportunity to develop and deliver a new simulator to help support the floating wind industry. An independent technical advisory group was established in 2016, with a view to garnering real-world feedback from key players in the floating wind industry. Guidance from the steering group has helped to shape a new product offering in this area, known as Flexcom Wind. First launched in June 2017, further, more advanced versions of the software will be developed over time.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Development Strategy outlines the rationale behind the decision to couple Flexcom with an established, third-party aerodynamic solver (OpenFAST).
- Model Building outlines a typical series of steps to follow when building a model of floating wind turbine.
- Dynamic Simulations provides guidelines regarding how to perform dynamic simulations.
- Extraction of Results explains how to extract relevant results via post-processing.
- Computational Methodology describes the computation of the primary load components in a wind turbine simulation, namely aerodynamic loads on the turbine blades and supporting tower, hydrodynamic loads on the floating platform, and structural and hydrodynamic loads imparted by the mooring lines. It also discusses how coupling between the structural and aerodynamic models is achieved, and contains several helpful sub-articles:
  - InflowWind Overview
  - AeroDyn Overview
  - TurbSim Overview
  - Turbine Geometry & Aerodynamic Coordinate Systems
- Software Modelling Limitations briefly outlines a number of limitations associated the modelling capability as it currently stands.
- Validation briefly outlines some test cases which were used to validate the software.

Development Strategy

INTRODUCTION

Based on our existing marine engineering and software development skills, Wood identified an opportunity to develop and deliver a new simulator to help support the floating wind industry. An independent technical advisory group was established in 2016, with a view to garnering real-world feedback from key players in the floating wind industry. Guidance from the steering group has helped to shape a new product offering in this area, known as Flexcom Wind. First launched in June 2017, further, more advanced versions of the software will be developed over time.

DEVELOPMENT APPROACH

Rather than independently developing a brand new aerodynamic modelling capability, the Flexcom development team decided instead to team up with the National Renewable Energy Laboratory of the United States Department of Energy. Since circa the year 2000, NREL have been progressively developing a software product called OpenFAST (an acronym for Fatigue, Aerodynamics, Structures, and Turbulence), a tool for simulating the coupled dynamic response of wind turbines. The software is highly regarded from a technical perspective by leading experts in the wind industry, and the research goals of NREL are well reflected by the state-of-the-art analytical solver which they have produced. Given that development efforts have focused exclusively on technical aspects of the solution, OpenFAST does not have any conventional user interface associated with it (all user inputs are defined via a series of plain text input files), so it is not perceived as being overly user-friendly, particularly for first time users. This is not surprising as OpenFAST was not designed to be a commercial software product. Instead NREL have generously made the software freely available as an open-source product, with the aim of aiding and accelerating the global development of the floating wind industry. On a practical level, OpenFAST is effectively a modularised framework of several component software modules (Jonkman, 2013). From a Flexcom perspective, the modules of primary interest are the wind loading processor, InflowWind, its aerodynamic solver, AeroDyn and control module ServoDyn. The coupling with these modules serves as an ideal complement to Flexcom's long established structural and hydrodynamic modelling capabilities.

Flexcom 2022.1.1 (August 2022) is coupled with OpenFAST V2.6.0 (May 2021).

Model Building

Building a floating wind turbine model in Flexcom is very straightforward. You simply follow a logical sequence of steps, adding new components in sequential fashion, as you would do for any offshore structure in Flexcom. Some of the following information may be very familiar to existing Flexcom users, but a coherent summary could serve as a helpful reference point.

- Set up an arrangement of Nodes and Elements to model the floating platform. It is not necessary to model the platform in explicit detail, but ensure that at a minimum, finite element nodes are placed at key locations (e.g. platform centre of gravity (CoG), platform centre of buoyancy (CoB), tower base, and fairlead positions of any attached mooring lines). This effectively serves as a framework upon which the various constituents may be applied. The connecting elements between these key locations are typically rigid (so the floater moves as a single rigid body), massless (the floater's inertia is typically concentrated at the CoG node), have zero Drag Diameter (to suppress the application of Morison drag loads, unless you are modelling viscous drag effects in distributed manner), and have zero Buoyancy Diameter (the floater's buoyancy is typically concentrated at the CoB node).

- At this point the assembly of elements will look rather skeleton-like, and not remotely resemble a floating platform, so the addition of an auxiliary Vessel Profile is advisable. While this will have no structural function, it will greatly enhance the visual appeal of the model, and will assist in the understanding of floater motions post-simulation.

- Define a Floating Body to model the physical characteristics of the floater. Specify the relevant Inertia terms at the centre of gravity. Use the Hydrostatic Stiffness terms to simulate restoring forces and moments due to buoyancy. Define Added Mass, Radiation Damping and Force RAO coefficients for the floating body over a range of discrete frequencies - these terms enable the computation of incident, diffracted and radiated (linear) wave potentials to be simulated. Define QTF Coefficients if second order drift loads are to be modelled also. All of these inputs must be derived separately from a radiation-diffraction analysis. Some of the more common commercial codes include WAMIT and ANSYS Aqwa, while NEMOH is a popular open-source code.

- Construct the tower using a single Line. Attach the lower end of the tower to the relevant node on the floating platform using the Equivalent Nodes facility. As the tower is normally tapered from a wide base to a more slender top, it is normally constructed in Flexcom using several Line Sections of different diameter. The mesh density for the structural model is governed by the Line Mesh Generation settings for the line and its sub-sections, while the mesh density for the aerodynamic model is controlled via the `*TOWER INFLUENCE` keyword. It is not necessary to use the same mesh density for both models, but as a minimum, structural nodes should be placed at elevations which correspond exactly to equivalent nodes in the aerodynamic model. In practice, the structural tower is typically modelled using a certain number of sub-sections of equal length, and the intersection points between these sections serve as the aerodynamic nodes also.

- The rotor is typically modelled using a short Line with an appropriate Mass per Unit Length. If one of the rotor nodes is coincident with the tower top, the two points may be connected together as Equivalent Nodes facility. Otherwise a short rigid massless Element may be used to form the required connection. Note that the Line Locations feature may be used if required to position nodes exactly at specific lengths along the rotor line.

- The nacelle is typically modelled by ensuring that a Point Mass is applied to a node which is located at the Nacelle center of mass. This node is typically connected to the rotor element using a rigid massless element. Similarly, the inertial effects of the hub and blades may be included by application of a point mass (or masses) at the end of the rotor.

- Note that the blade rotation cannot be modelled in Flexcom so the addition of Auxiliary Profiles to visually represent the blades is strongly recommended.

- Specify all the wind turbine inputs which are required by AeroDyn to compute the aerodynamic loading on the blades and tower. This category of inputs (logically grouped together under `$AERODYN`) will be intuitively familiar to engineers with some wind turbine modelling experience. Fundamental inputs in this category include Blade Geometries, Aerofoil Coefficients, miscellaneous Turbine Inputs (such as hub height, hub radius, overhang, shaft tilt, blade precone etc.) and Tower Influence (i.e. tower drag). There are also various other advanced options affording user control e.g. Blade Element Momentum Theory.

- Create the mooring lines also using the Lines feature. Attach the upper end of each mooring line to the relevant fairlead node on the floating platform using the Equivalent Nodes facility. Constrain the lower (seabed) end of each mooring line using Fixed Boundary Conditions.

- Define Environmental Parameters, such as ocean depth and water density, and include a Seabed definition.

- Perform a static analysis in order to determine the static equilibrium configuration of the entire system subject to gravity and buoyancy loads only. In order to aid solution convergence, this simulation is normally performed in two stages, with the CoG node being temporarily restrained in an Initial Static Analysis using additional Fixed Boundary Conditions, which are subsequently removed in a Restart Static Analysis. Floating systems can be sensitive to minor changes in displacement, and if static convergence proves difficult to achieve, the second stage may be performed as a Quasi-Static Analysis.

- One further restart analysis may also be performed if static Current Loads on the floater are being modelled.

Regarding the floating platform itself, there are varying levels of detail with which you can model it. The above suggests that finite element nodes are placed only at key points of interest, such as the centres of gravity and buoyancy, and that all applied loads are concentrated at these points. It is also possible to model the floating body in explicit detail, including finite elements to model the various pontoons, decks, braces etc. which comprise the real-world structure. There are some advantages associated with creating a more detailed model - refer to Floating Body Modelling Detail if you are interested in further details.

Once the static equilibrium configuration of the entire system has been achieved, the ambient environmental loads due to wind and waves may be defined, and a Dynamic Simulation may be performed.

Dynamic Simulations

Once the static equilibrium configuration of the entire system has been achieved following the Model Building process, the ambient environmental loads due to wind and waves may be defined, and dynamic simulations may be performed. The following guidelines are helpful in this regard. Typically a single dynamic analysis is examined initially, to ensure that the dynamic response to a sample environmental simulation is consistent with expectations.

* Set up a new Time Domain Analysis input file, restarting from the preceding static solution, and select appropriate Time Variables. In standard offshore applications, appropriate time variables are typically dependent on the ambient seastate. For wind turbine modelling, NREL recommend that the solution time step for aerodynamic calculations be set such that there are at least 200 time steps per rotor revolution.

* Create a link between the structural (Flexcom) and aerodynamic (AeroDyn) models, via the `*AERODYN DRIVER` inputs. This identifies key locations in the model, such as the hub node and tower elements, which act as central points for information exchange between the two solvers. Load case specific inputs such as wind Loading and blade rotational speed are also defined at this stage.

* Define the wind loading. Steady wind is the simplest option, and is invoked in Flexcom by specifying a constant *Wind Speed* and *Shear Exponent* via the `*AERODYN DRIVER` keyword (or the Wind Component if you are using the Flexcom Wind module). For any of the more advanced wind field definitions, you specify the name of the InflowWind file (via the *INFLOWWIND keyword or the Wind Component), which you will need to have created in advance. Refer to InflowWind Overview for further details. You will also need to generate a turbulent wind field using TurbSim - refer to TurbSim Overview for further information.

* Define the wind turbine control system via the `*SERVODYN` keyword. You can vary generator torque to control rotor speed and/or vary blade pitch via a Bladed-style dynamic link library (dll). Refer to ServoDyn Overview for further details.

* Choose from the range of options for applying Wave Loading.

After a successful dynamic simulation, the next logical step is to proceed to examining a matrix of simulations.

* Set up some Parameters to represent key inputs. For example, you may wish to create seastate parameters such as wave period and wave direction, or wind parameters such as wind speed or shear exponent.

* Use Keyword Based Variations (ideal for parameters which vary in fixed increments) or Spreadsheet Based Variations (for arbitrary variations) to examine the effect of varying key parameters. One master template input file automatically generates all the required input files to simulate each unique combination of different parameters.

If you would like to see an example of how these inputs are used in practice, refer to J03 - Summary Postprocessing Collation.

Extraction of Results

Retrieval of results from a Flexcom simulation is achieved by making a series of postprocessing requests. For a seamless experience, these requests are normally stated in advance, but you can easily extract results post-simulation if required. There are two main postprocessing channels in Flexcom, namely Database Postprocessing which presents detailed information in the form of graphs/plots, and Summary Postprocessing which lists only critical statistical values (e.g. maximum, minimum, mean, standard deviation) in succinct tabular format. Summary postprocessing also has a related feature for the Collation of critical results across a range of different simulation into a single Summary Collation Spreadsheet or a single 3D Summary Collation Plot.

If one single load case is currently under scrutiny, the database postprocessor might be used to create 2D plots of some of the following variables (each hyperlink takes you to the relevant keyword command).

- Time histories of floater motions in 6 degrees of freedom
- RAOs of floater motions in 6 degrees of freedom
- Time histories of mooring line tension at fairlead locations
- Time histories of aerodynamic parameters such as wind speed, lift force, drag force, pitching moment, normal force, tangential force, tip speed ratio, aerodynamic power etc. Interested readers may browse a full list of all possible AeroDyn Output Parameters.

If a matrix of load cases is being considered, the summary postprocessor might be used to create tabular summaries of some of the following variables (each hyperlink takes you to the relevant keyword command). A single Summary Collation run may then be used to gather all the key information into a single spreadsheet or 3D plot.

- Envelopes of floater motions in 6 degrees of freedom
- Envelopes of mooring line tensions

- Unfortunately it is not possible to extract aerodynamic parameters via summary postprocessing in the current version of Flexcom. This facility will be added in a future software upgrade.

Computational Methodology

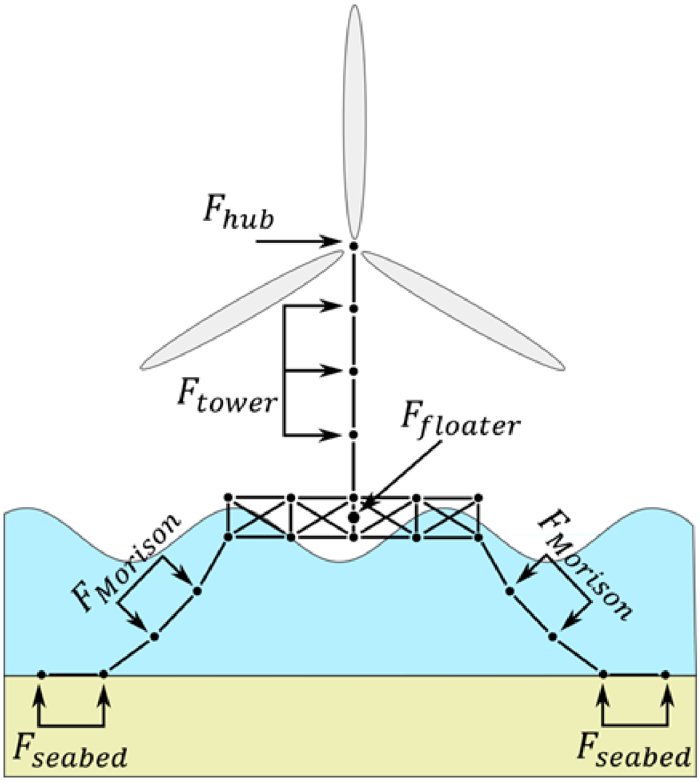
OVERVIEW

This section outlines the computation methodology for floating offshore wind turbines. Further information is provided in Software Architecture and Software Couplings.

There are three primary load components in a Flexcom wind turbine simulation:

- Aerodynamic loads on the turbine blades and supporting tower
- Hydrodynamic loads on the floating platform
- Structural and hydrodynamic loads imparted by the mooring lines

The composition of each term is now outlined, preceded by a description of how Flexcom is coupled to the InflowWind module to provide complex wind loading options to the system. This is followed by a description of the coupling between Flexcom's structural model and AeroDyn's aerodynamic model. This is then followed by a description of Flexcom Wind control systems via coupling with the ServoDyn module.



Primary Load Components

WIND LOADING

Flexcom facilitates the use of the OpenFAST module InflowWind, via the **INFLOWWIND* keyword, so dynamic and turbulent wind loading can be specified on the wind turbine. InflowWind processes wind inflow and supports several wind file formats including uniform, binary TurbSim full-field, binary Bladed-style full-field and HAWC formatted binary full-field wind files. It also has it's own internal calculated steady wind and supports arbitrary wind directions. Platt et al. (2016). The turbulent wind field definition is typically generated in advance using TurbSim - refer to TurbSim Overview for further information.

At each timestep, InflowWind is given the blade and tower nodal locations and then calculates the undisturbed wind-inflow velocities at these positions. As described in Platt et al. (2016) - "There are no states in the module: each wind velocity component is calculated as a function of the input coordinate positions and internal time-varying parameters, undisturbed from interaction with the wind turbine. For full-field wind data types, InflowWind uses Taylor's frozen turbulence hypothesis—valid only for stationary conditions—to translate wind defined in two dimensional planes into three spatial dimensions, using the mean wind speed as the advection speed".

Aerodynamic loads on the blades and tower are computed directly by AeroDyn. The aerodynamic calculations are based on the principles of actuator lines, where the three-dimensional (3D) flow around a body is approximated by local two-dimensional (2D) flow at cross sections, and the distributed pressure and shear stresses are approximated by lift forces, drag forces, and pitching moments lumped at a node in a 2D cross section. Analysis nodes in AeroDyn are distributed along the length of each blade, the 2D forces and moment at each node are computed as distributed loads per unit length, and the total 3D aerodynamic loads are found by integrating the 2D distributed loads along the length. The total loads on the rotating blades are then assembled at the hub, and passed to the Flexcom solver for input into the global force vector on the right hand side of the equations of motion.

The wind load on the tower is based directly on the tower diameter and drag coefficient and the local relative wind velocity between the freestream (undisturbed) wind and structure at each tower analysis node in AeroDyn. These loads are then passed to Flexcom, where they are applied to the corresponding tower node in the structural model, and the global force vector is augmented accordingly.

Refer to AeroDyn Overview for further information regarding the computation of aerodynamic loads on the blades and tower. Refer also to the section on AeroDyn Turbine Geometry, which provides a series of schematics to graphically illustrate the various sub-components of a wind turbine.

FLOATING PLATFORM

Hydrodynamic loading on the floating platform includes the various items listed below. The loads are computed for the floating body as a whole, and then applied at an appropriate location in the global force vector (e.g. at a node which corresponds to a centralised location such as the centre of gravity).

- First-Order Wave Loads (high frequency) derived from Force RAOs

- Wave Radiation Loads. An important issue arises with respect to the Added Mass and Radiation Damping terms associated with the floating body, and how these are modelled in the time domain. The frequency-dependent nature of these terms is accounted for using the established impulse response approach developed by Cummins (1962), and its implementation in Flexcom is described in detail by Connaire et al. (2003) and Lang et al. (2005). Specifically, the frequency-dependent damping term is replaced by a convolution integral of retardation functions and velocity time histories in the time domain.

- Viscous Damping Loads. These may be derived from centralised Viscous Damping Coefficients at some point on the floating body, or simulated in a distributed manner via Morison's Equation.

- Second-Order Wave Drift Loads (low frequency) derived from Quadratic Transfer Functions (QTFs)

- Current Loads computed via Current Coefficients

- Wind Loads computed via Wind Coefficients

- Hydrodynamic Loads determined according to the theory of manoeuvrability

Further information is also provided in these related articles...

- Potential Flow Theory & Morison's Equation discusses the main differences between these hydrodynamic modelling approaches.

- Floating Body Modelling Detail discusses relatively simple (concentrated loads) and more complex approaches (distributed loads) for physically modelling the floating structure.

MOORING LINES

Mooring lines are modelled in the standard fashion using beam elements. These elements form a natural part of the overall finite element solution, so their effect on the floater is handled automatically. Specifically, effective tensions in the mooring lines induce point loads on the floater at the fairlead connection points. In addition to the axial loads, mooring lines are subject to self-weight and Buoyancy Forces. Hydrodynamic Loading is based on Morison's Equation, including the fundamental components of drag, added mass and hydrodynamic inertia. Contact algorithms are used to simulate Seabed Interaction with the local seabed topography.

InflowWind Overview

This section presents a very brief synopsis of the operation of InflowWind, using information sourced directly from the InflowWind program documentation, reproduced with the permission of NREL. Refer to the NREL InflowWind documentation for further information.

InflowWind is a module for processing wind-inflow that has been coupled into OpenFAST. It supports several wind file formats including the following (corresponding to the *WindType* switch in the input file):

i. Steady wind (calculated internally using the Steady Wind Conditions section)

ii. Uniform wind (read externally from a Uniform Wind File)

iii. Binary TurbSim full-field wind file

iv. Binary Bladed-style FF wind file

v. Binary HAWC-style FF wind files

vi. User defined using the UserWind subroutine (to be supplied by and compiled by the end user; template IfW_UserWind.f90 file is provided)

The parameters in InflowWind's primary input file are divided into several sections. The text of an example InflowWind input file is included in Example L01 - OC4 Semi-Submersible. Please note that there are multiple inputs with the same name in different sections of this input file (for example, RefHt, the reference height for the wind file). The parameters in each section are specific to the section in which they are defined. For example, the RefHt used in Steady Wind Conditions section is used only for the steady wind calculations; the RefHt in the HAWC-format section is used only for HAWC-style files.

At each time step, InflowWind receives from the driver code (either the standalone driver or the OpenFAST glue code) the coordinate position of various points and InflowWind returns the undisturbed wind-inflow velocities at these positions. There are no states in the module: each wind velocity component is calculated as a function of the input coordinate positions and internal time-varying parameters, undisturbed from interaction with the wind turbine. For full-field wind data types, InflowWind uses Taylor's frozen turbulence hypothesis - valid only for stationary conditions - to translate wind defined in two-dimensional planes into three spatial dimensions, using the mean wind speed as the advection speed.

AeroDyn Overview

INTRODUCTION

This section presents a very brief synopsis of AeroDyn, using information sourced directly from the program documentation, reproduced with the permission of NREL. Refer to the NREL AeroDyn documentation for further information.

AeroDyn performs aerodynamic calculations for simulations of horizontal axis wind turbines. It calculates the aerodynamic lift, drag, and pitching moment of aerofoil sections along the wind turbine blades, by dividing each blade into a number of segments along the span. Equipped with information about the turbine geometry, operating condition, blade-element velocity and location, and wind inflow from InflowWind, it calculates the various forces for each segment, which may then be used to compute distributed forces on the turbine blades.

Aerodynamic calculations within AeroDyn are based on the principles of actuator lines, where the three-dimensional (3D) flow around a body is approximated by local two-dimensional (2D) flow at cross sections, and the distributed pressure and shear stresses are approximated by lift forces, drag forces, and pitching moments lumped at a node in a 2D cross section. Analysis nodes are distributed along the length of each blade and tower, the 2D forces and moment at each node are computed as distributed loads per unit length, and the total 3D aerodynamic loads are found by integrating the 2D distributed loads along the length. The actuator line approximations restrict the validity of the model to slender structures and 3D behaviour is either neglected, captured through corrections inherent in the model (e.g., tip-loss, hub-loss, or skewed-wake corrections), or captured in the input data (e.g., rotational augmentation corrections applied to aerofoil data).

AeroDyn consists of four submodels: (1) rotor wake/induction, (2) blade aerofoil aerodynamics, (3) tower influence on the wind local to the blade nodes, and (4) tower drag. Nacelle, hub, and tail-vane wind influence and loading, aeroacoustics, and wake and array effects between multiple turbines in a wind plant, are not yet available (AeroDyn v15).

ROTOR WAKE/INDUCTION

For operating wind turbine rotors, AeroDyn calculates the influence of the wake via induction factors based on the quasi-steady Blade-Element/Momentum (BEM) theory, which requires an iterative nonlinear solver (implemented via Brent's method). By quasi-steady, it is meant that the induction reacts instantaneously to loading changes. The induction calculation, and resulting inflow velocities and angles, are based on flow local to each analysis node of each blade, based on the relative velocity between the wind and structure (including the effects of local inflow skew, shear, turbulence, tower wind disturbances, and structural motion, depending on features enabled). The Glauert's empirical correction (with Buhl's modification) replaces the linear momentum balance at high axial induction factors. In the BEM solution, Prandtl tip-loss, Prandtl hub-loss, and Pitt and Peters skewed-wake are all 3D corrections that can optionally be applied. When the skewed-wake correction is enabled, it is applied after the BEM iteration. Additionally, the calculation of tangential induction (from the angular momentum balance), the use of drag in the axial-induction calculation, and the use of drag in the tangential-induction calculation are all terms that can optionally be included in the BEM iteration (even when drag is not used in the BEM iteration, drag is still used to calculate the nodal loads once the induction has been found). The wake/induction calculation can be bypassed altogether for the purposes of modeling rotors that are parked or idling, in which case the inflow velocity and angle are determined purely geometrically. Dynamic wake that accounts for induction dynamics as a result of transient conditions are not yet available (AeroDyn v15).

BLADE AEROFOIL AERODYNAMICS

The blade aerofoil aerodynamics can be steady or unsteady. In the steady model, the supplied static aerofoil data—including the lift-force, drag-force, and optional pitching-moment coefficients versus angle of attack (AoA)—are used directly to calculate nodal loads. The unsteady aerofoil aerodynamic (UA) models account for flow hysteresis, including unsteady attached flow, trailing-edge flow separation, dynamic stall, and flow reattachment. The UA models can be considered as 2D dynamic corrections to the static aerofoil response as a result of time-varying inflow velocities and angles. Three semi-empirical UA models are available: the original theoretical developments of Beddoes-Leishman (B-L), extensions to the B-L developed by González, and extensions to the B-L model developed by Minnema/Pierce. While all of the UA models are documented in the AeroDyn manual, the original B-L model is not yet functional (AeroDyn v15). The aerofoil-, Reynold's-, and Mach-dependent parameters of the UA models may be derived from the static aerofoil data. These UA models are valid for small to moderate AoA under normal rotor operation; the steady model is more appropriate under parked or idling conditions. The static aerofoil data is always used in the BEM iteration; when UA is enabled, it is applied after the BEM iteration and after the skewed-wake correction. The interpolation of aerofoil data based on Reynolds number or aerodynamic-control setting (e.g., flaps) is not yet available (AeroDyn v15).

TOWER INFLUENCE

The influence of the tower on the wind local to the blade is based on a potential-flow and/or a tower shadow model. The potential-flow model uses the analytical potential-flow solution for flow around a cylinder to model the tower dam effect on upwind rotors. In this model, the freestream (undisturbed) wind at each blade node is disturbed based on the location of the blade node relative to the tower and the tower diameter, including lower velocities upstream and downstream of the tower, higher velocities to the left and right of the tower, and cross-stream flow. The Bak correction can optionally be included in the potential-flow model, which augments the tower upstream disturbance and improves the tower wake for downwind rotors based on the tower drag coefficient. The tower shadow model can also be enabled to account for the tower wake deficit on downwind rotors. This model includes an axial wind deficit on the freestream wind at each blade node dependent on the location of the blade node relative to the tower and the tower diameter and drag coefficient, based on the work of Powles. Both tower-influence models are quasi-steady models, in that the disturbance is applied directly to the freestream wind at the blade nodes without dynamics, and are applied within the BEM iteration.

TOWER DRAG

The wind load on the tower is based directly on the tower diameter and drag coefficient and the local relative wind velocity between the freestream (undisturbed) wind and structure at each tower analysis node (including the effects of local shear, turbulence, and structural motion, depending on features enabled). The tower drag load calculation is quasi-steady and independent from the tower influence on wind models.

TurbSim Overview

INTRODUCTION

This section presents a very brief synopsis of TurbSim, using information sourced directly from the program documentation, reproduced with the permission of NREL. Refer to the NREL TurbSim documentation for further information.

TurbSim is a stochastic, full-field, turbulence simulator primarily for use with InflowWind-based simulation tools. The TurbSim stochastic inflow turbulence tool has been developed to provide a numerical simulation of a full-field flow that contains coherent turbulence structures that reflect the proper spatiotemporal turbulent velocity field relationships seen in instabilities associated with nocturnal boundary layer flows and which are not represented well by the IEC Normal Turbulence Models. Its purpose is to provide the wind turbine designer with the ability to drive design code simulations of advanced turbine designs with simulated inflow turbulence environments that incorporate many of the important fluid dynamic features known to adversely affect turbine aeroelastic response and loading. TurbSim is more efficient than its predecessors in terms of both CPU and memory usage. This software was developed by Neil Kelley and Bonnie Jonkman of NREL.

INSTALLING TURBSIM

1. Download the TurbSim program from the TurbSim website or OpenFAST GitHub site. The downloads are supplied as self-extracting ZIP files.

2. Source code is available should you wish to compile the software, but it is more straightforward to use a pre-compiled executable program. You will be seeking a file called TurbSim_x64.exe (or TurbSim_Win32.exe if you are running a 32-bit operating system).

3. Note that the TurbSim website also provides a helpful user guide. This contains useful technical information and ideally you should familiarise yourself with the various wind parameter inputs.

4. Save TurbSim_x64.exe to a convenient location on your hard drive, for example ‘C:\Program Files\NREL\TurbSim\TurbSim_x64.exe’.

RUNNING TURBSIM

Wind Field Generator

The easiest way to run TurbSim is via the Wind Field Generator app, which effectively incorporates TurbSim into the Flexcom environment. This is a helpful tool which acts a user-friendly interface to the TurbSim software which does not have a conventional Window-based GUI of its own. It allows you to run batches of TurbSim wind field simulations to generate all the wind data files which you require to support your design load cases. Several of Flexcom's wind turbine examples, such as Example L04 - UMaine VolturnUS-S IEA15MW, utilise TurbSim binary wind-field definitions files. With the file extension BTS (denoting Binary TurbSim), these files tend to be very large so they are not supplied with Flexcom as it is not practical to include them in the installation package. Instead you can readily generate the BTS files yourself with the Wind Field Generator app.

Command Prompt

Alternatively you can run TurbSim from a command prompt window if you prefer. TurbSim is a DOS-based program and many younger software users may not be familiar with this system which predates Windows. But it is quite straightforward, and you can follow these steps even if you are not experienced with DOS.

1. Open a command prompt window. You can do this by typing CMD into the Windows search box.

2. Note that you can use the command 'cd foldername' to change directory in the command prompt window. The command 'cd..' allows you to move up one directory level. If the file paths contain spaces, you must enclose the entries in quotation marks.

3. Navigate to a location on your computer where some TurbSim input files are located. For Example L04, the command will be something like cd “C:\Users\Public\Documents\Wood\Flexcom\Version 2022.1.1\Example Projects\L - Wind Energy\04 - UMaineVolturnUS-IEA15MW\Data\Wind”. Or you can move in a series of smaller steps, for example (i) cd C:\Users, (ii) cd Public, and so on.

4. If you examine this wind data folder in Explorer, you will notice that it contains a sample IN file, called IECKAI-1NTM-4-60362647-0.in. It uses the following naming convention:

   a. IECKAI is the turbulence model, a Kaimal spectrum in this case

   b. 1NTM is the IEC turbulence type, in this case 1 is the wind turbine class and NTM denotes normal turbulence

   c. 4 is the wind speed, 4m/s in this case

   d. 60362647 is a random seed associated with the generated wind profile

   e. 0 is the wind direction, 0 degrees in this case

5. To generate a binary (BTS) file from an input (IN) file, you type the name of the TurbSim executable, followed by the name of the wind input file. For example, the command might look as follows: “C:\Program Files\NREL\TurbSim\TurbSim_x64.exe” “IECKAI-1NTM-4-60362647-0.in”. This approach will allow you to generate all the required BTS files one by one.

6. If you would like to automatically generate several files simultaneously, you can examine the batch file which is contained in the wind folder (RunTurbsimNew.bat). This can be viewed in any standard text editor and can be saved in text format after you have made your changes.

   a. Identify the location of TurbSim on your computer, via the ‘turbsim_exe’ entry

   b. Set the number of concurrent runs, via the ‘maxProc’ entry, this should be less than the number of CPUs available on your computer

   c. Edit the list of IN files which you wish to process, these are listed under the ‘list of commands’ comment

d. Save the file and close once you are finished

e. Run the batch by typing the name of the batch file into the command window, i.e. type ‘RunTurbsimNew.bat’

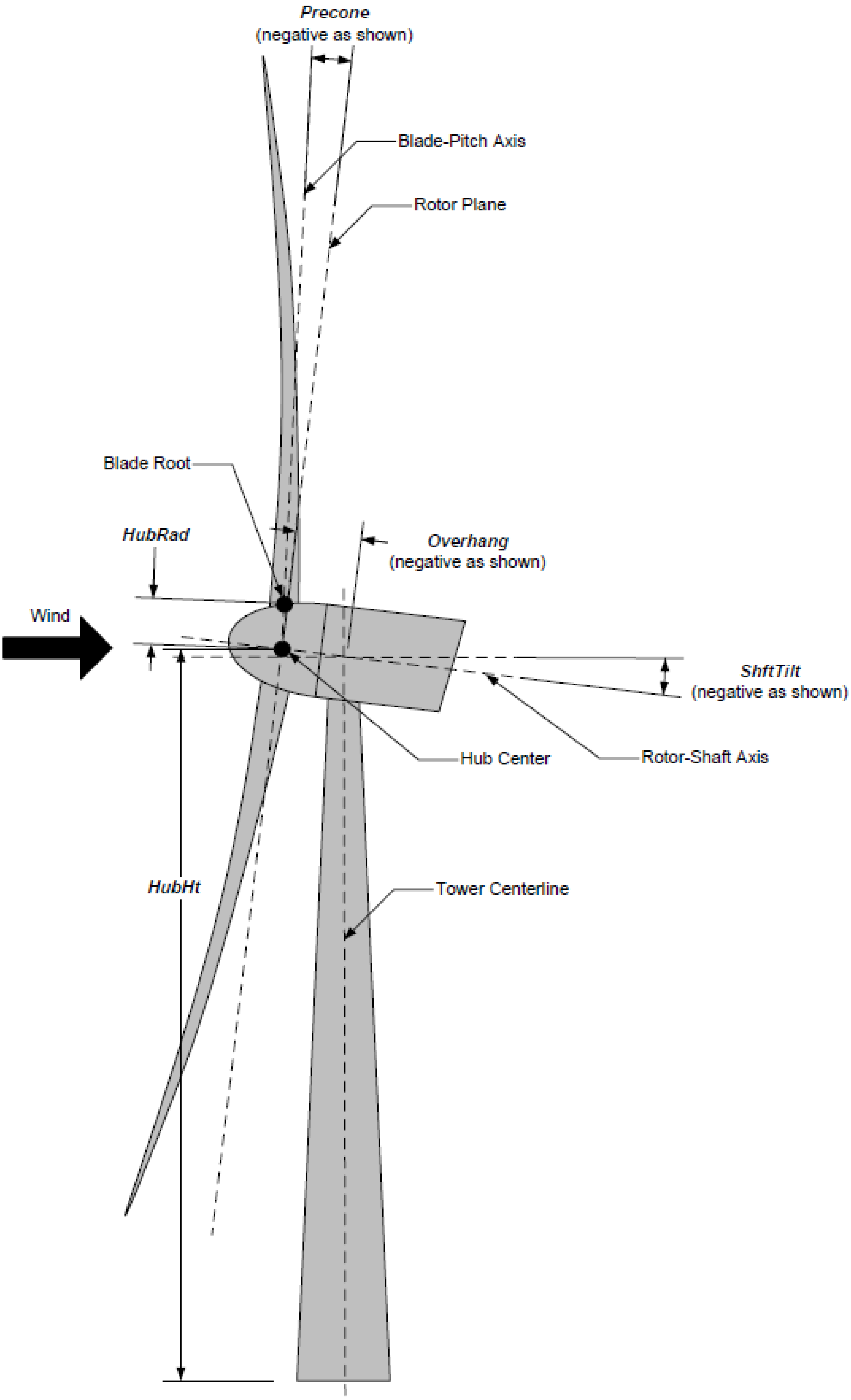
ServoDyn Overview

Documentation on ServoDyn is incomplete as yet. Further details will soon become available on the ServoDyn website.

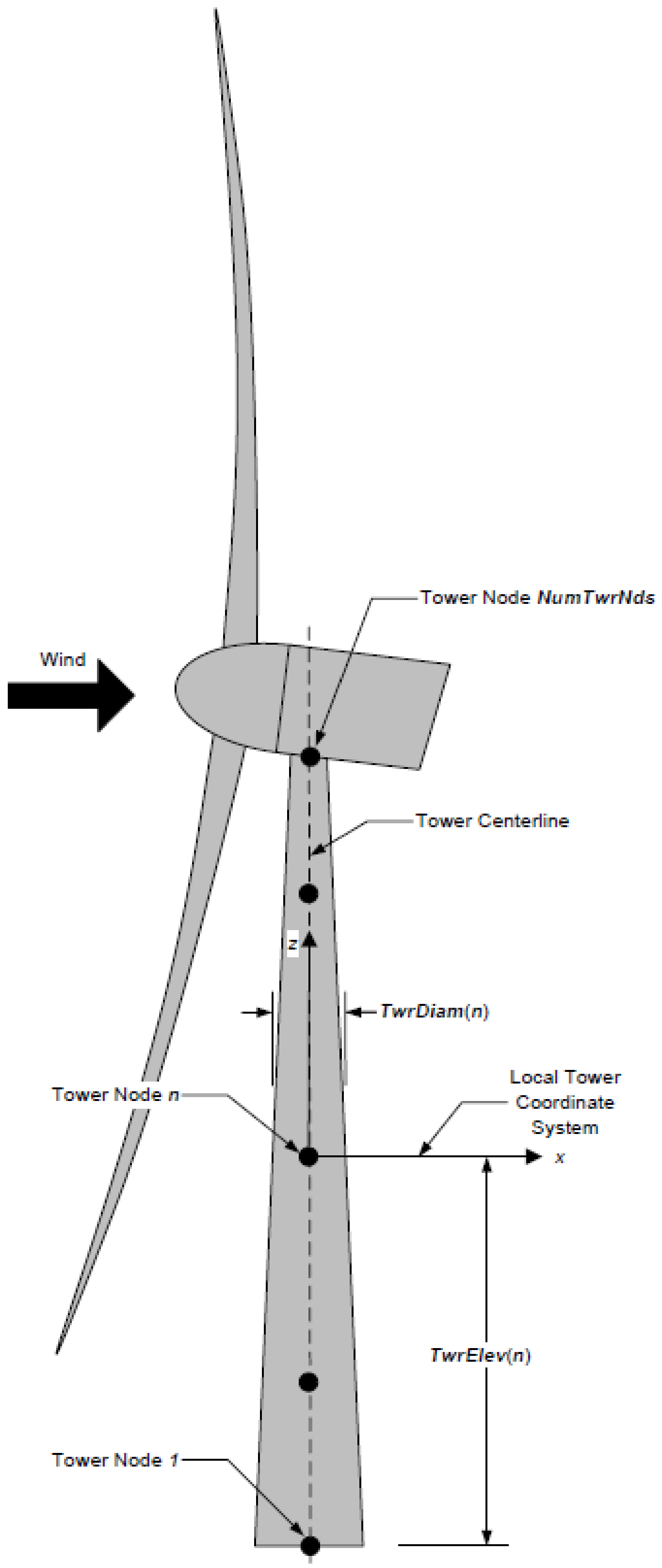
Turbine Geometry

This section provides a series of schematics which graphically illustrate the various sub-components of a wind turbine. It serves as a useful reference point for all information relating to turbine geometry. Refer also to Aerodynamic Coordinate Systems for additional information.

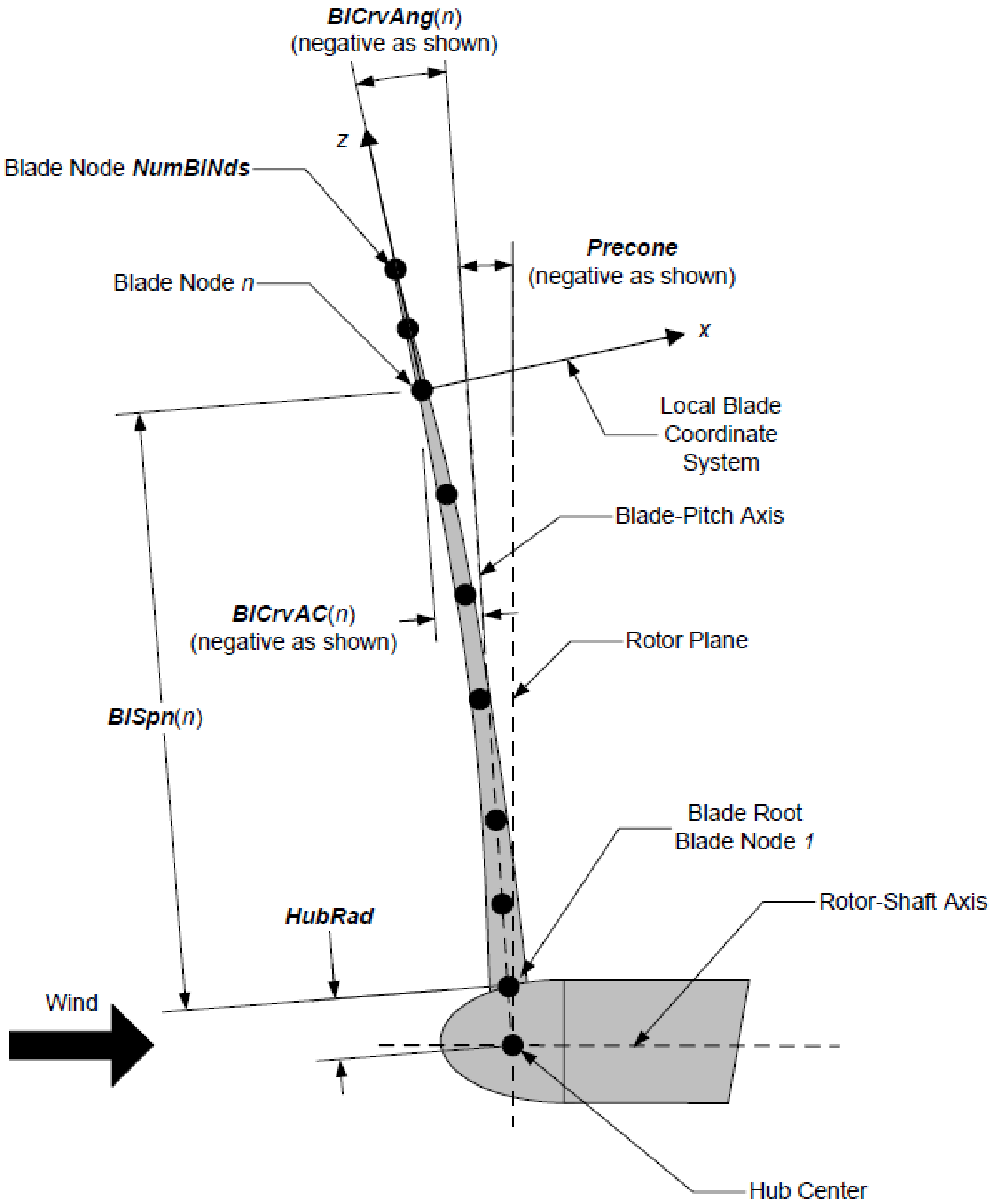
All of the images are reproduced from NREL's AeroDyn program documentation, with the kind permission of NREL.



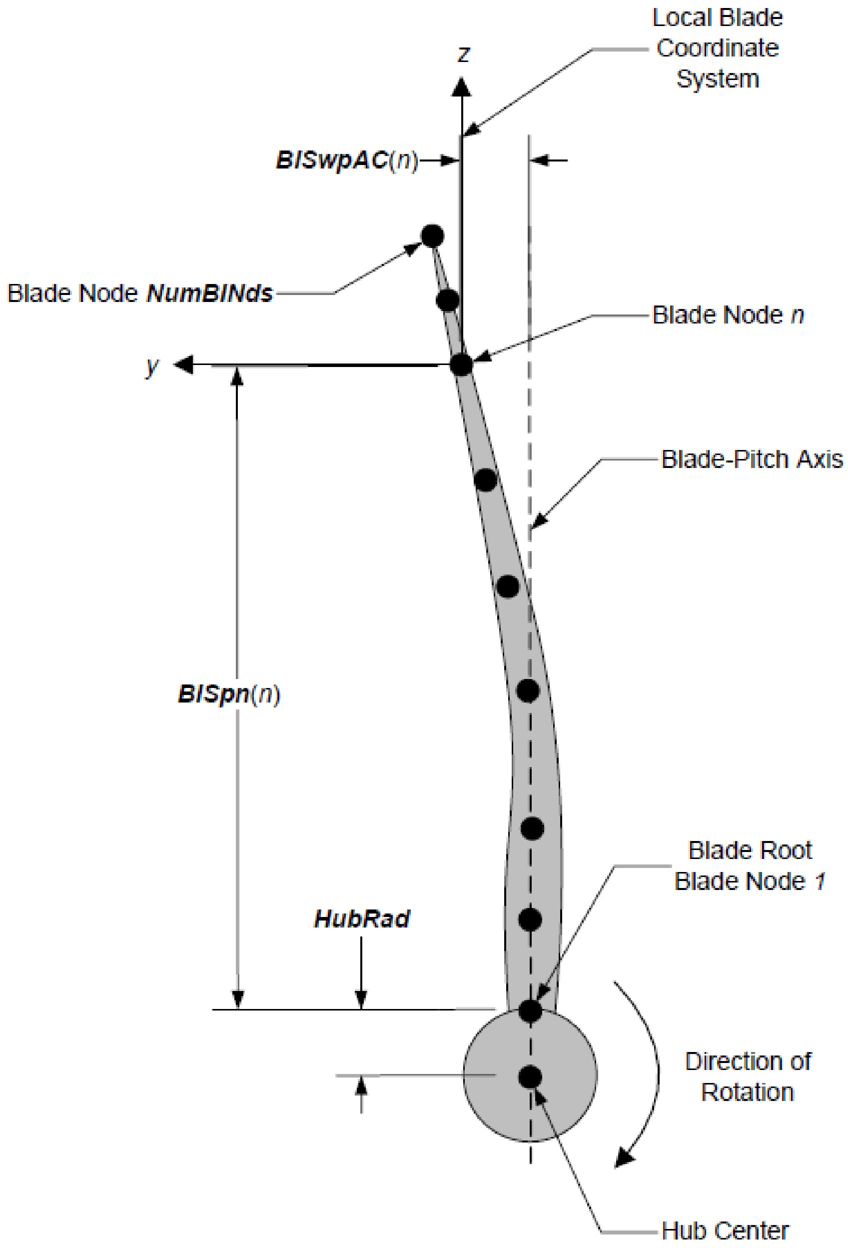
AeroDyn Driver Turbine Geometry
(image courtesy of NREL)



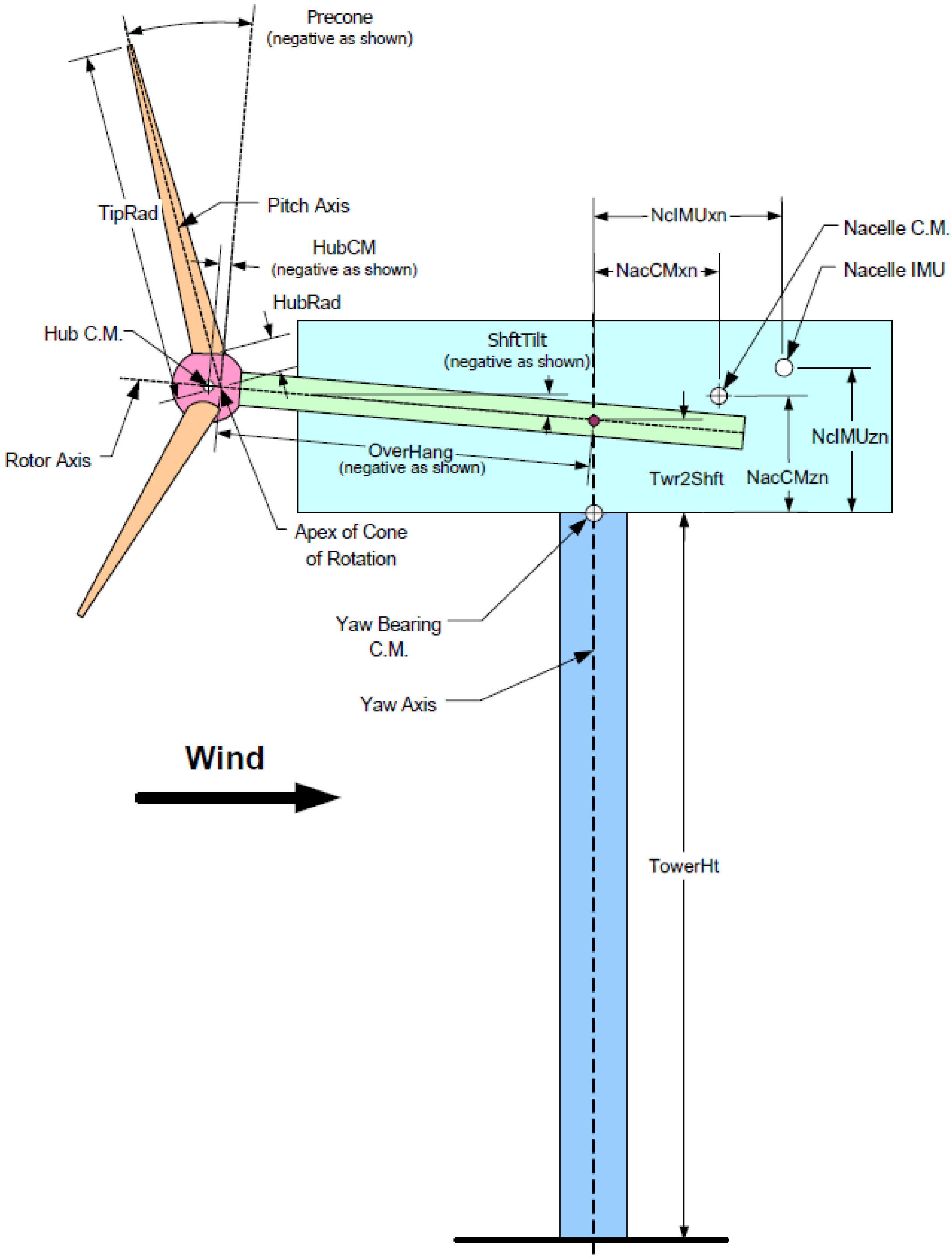
AeroDyn Tower Geometry
(image courtesy of NREL)



AeroDyn Blade Geometry - Side View
(image courtesy of NREL)



AeroDyn Blade Geometry - Front View (Looking Downwind)
(image courtesy of NREL)



Layout of a conventional, upwind, 3-bladed turbine
(image courtesy of NREL)

Aerodynamic Coordinate Systems

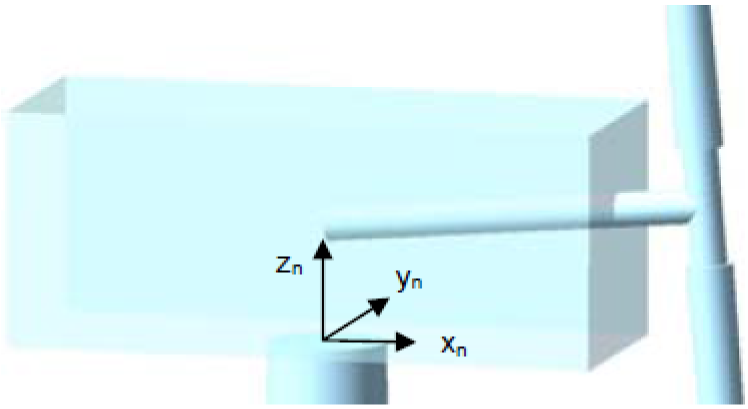
This section provides a series of schematics which graphically illustrate the various coordinate systems used for wind turbine modelling. Refer also to Turbine Geometry for additional information.

All of the images are reproduced from NREL's AeroDyn program documentation, with the kind permission of NREL.

NACELLE/YAW COORDINATE SYSTEM

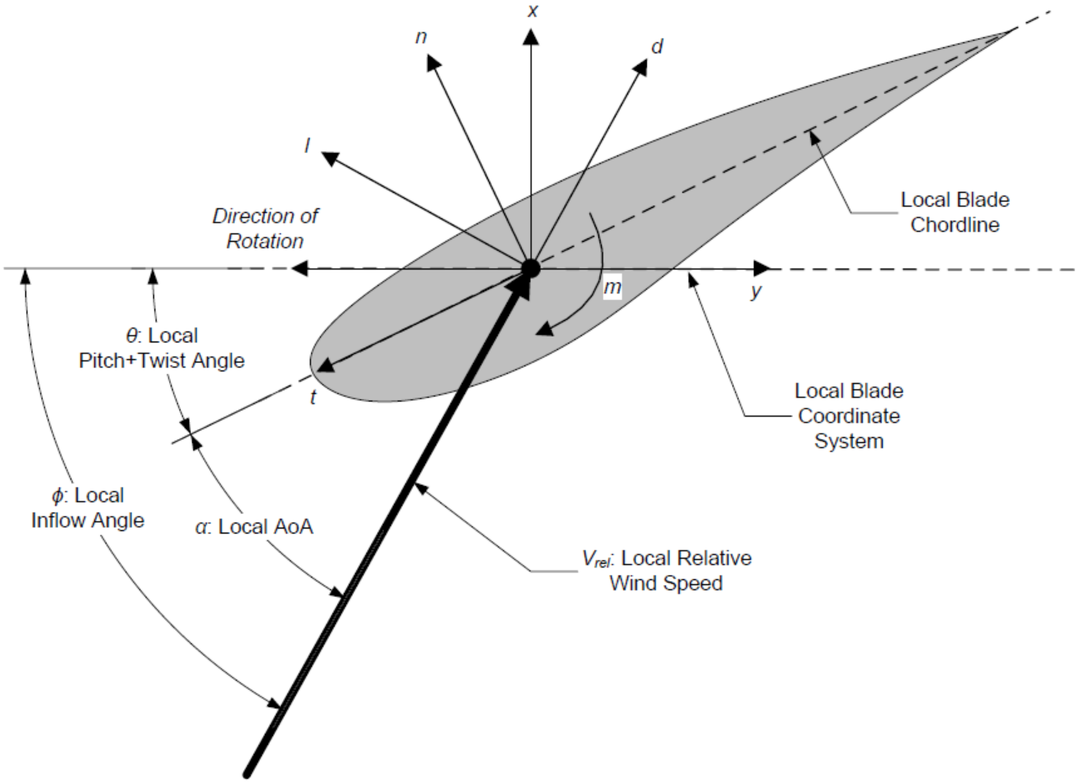
This coordinate system translates and rotates with the top of the tower, plus it yaws with the nacelle.

- Origin A point on the yaw axis at a height of 'TowerHt' above mean sea level. Refer to Conventional Upwind Turbine Layout for an illustration of TowerHt.
- x_n axis Pointing horizontally toward the nominally downwind end of the nacelle
- y_n axis Pointing to the left when looking toward the nominally downwind end of the nacelle
- z_n axis Coaxial with the tower/yaw axis and pointing up



Nacelle/Yaw Coordinate System

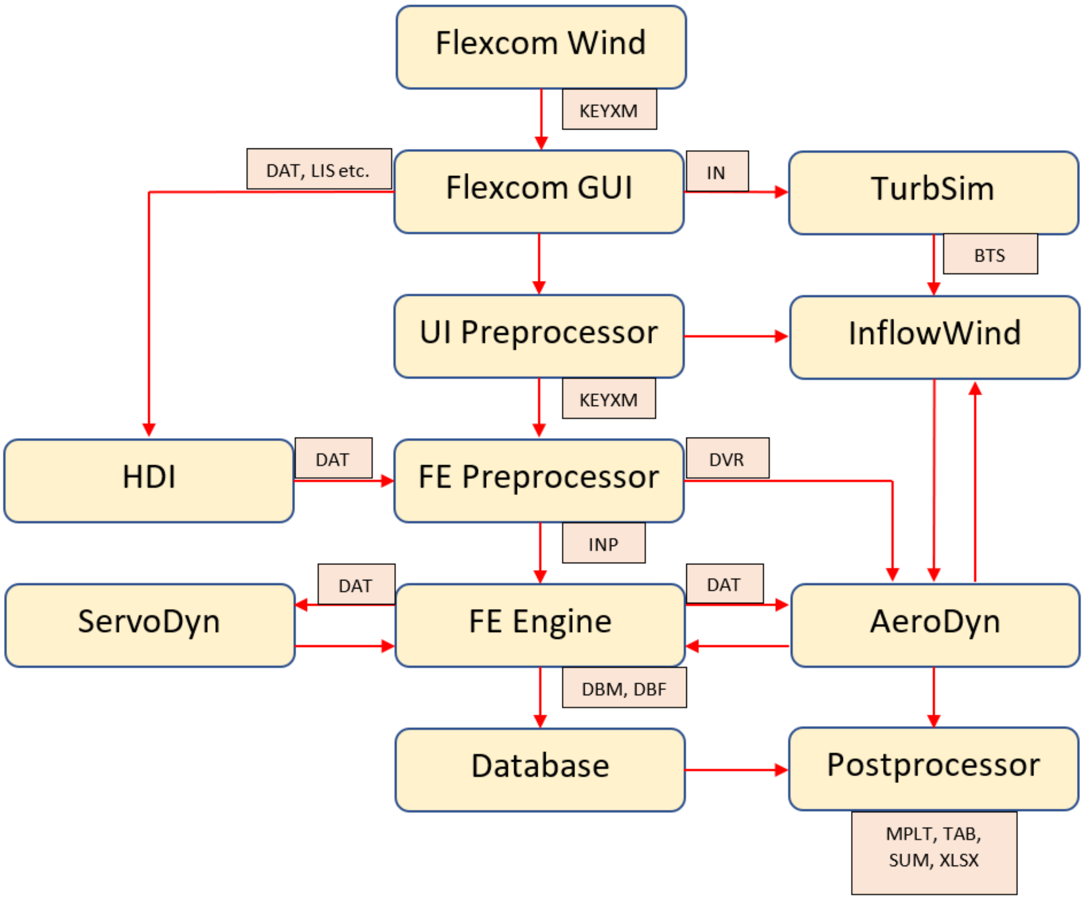
BLADE LOCAL COORDINATE SYSTEM



AeroDyn Local Blade Coordinate System (Looking Toward the Tip, from the Root) – l: Lift, d: Drag, m: Pitching, x: Normal (to Plane), y: Tangential (to Plane), n: Normal (to Chord), and t: Tangential (to Chord) (image courtesy of NREL)

Software Architecture

The following schematic illustrates the software architecture, supported by the additional notes underneath.



- Flexcom Wind is the dedicated user interface for model building of offshore wind turbines. It creates a well structured, heavily parameterised model in standard Keyword File format (KEYXM file) which can be subsequently edited within the main Flexcom GUI to meet individual requirements.

- Flexcom GUI is the main starting point for more experienced Flexcom users, who may already have template models built up from previous projects. It provides complete flexibility and control in terms of model building and running simulations.

- TurbSim, a full-field turbulent wind simulator, may be run from the Flexcom GUI. Flexcom creates input files (IN files) for TurbSim to model various wind conditions, and it creates a set of binary TurbSim files (BTS files) which contain wind field data as a function of space and time.

- UI Pre-processor, known as PreKey, is responsible for model generation within the Flexcom GUI. Its most important tasks are:

    - Finite element mesh generation for lines

  - Preliminary model solver based on a Runge-Kutta solution scheme
  - Preparing a simplified keyword file (KEYXM file) to the FE Preprocessing module
  - Displaying the model preview in the Model View

- FE Pre-processor, known as ACM (analysis control module), is responsible for reading of keyword files (KEYXM) provided by the Flexcom GUI, processing and storing this data, and passing it on to the FE Engine (via the INP file, a file which the user is largely unaware of). It also creates the driver file (DVR file) which is used to initiate AeroDyn.

- FE Engine is the core computational module of Flexcom. It applies the finite element formulation to solve for the structural motions and forces. The FE engine interacts with AeroDyn (DVR file), InflowWind (DAT file) and ServoDyn (DAT file) at every solution time step, to ensure that the aerodynamic and structural models are fully coupled. Further details are provided in Software Couplings.

- Hydrodynamic Data Importer allows data to be imported from hydrodynamic simulation packages such as WAMIT and AQWA. It accepts input files such as OUT (WAMIT) and LIS (AQWA) etc. and creates a set of DAT files which are passed to the FE Preprocessor.

- Database Files are created by the FE Engine to store simulation results. DBM and DBF files are used to store motion and force data respectively.

- Postprocessor modules are used to extract information from the results database and display it to the user. The Database Postprocessor creates plot files (MPLT), tabular output files (TAB), and spreadsheets (XLSX). The Summary Postprocessor creates summary output files (SUM). Summary Collation creates summary collation spreadsheets (XLSX) and 3D plots (MPLT).

Software Couplings

COUPLING BETWEEN FLEXCOM AND AERODYN

The `*AERODYN DRIVER` command provides the key link between Flexcom and AeroDyn. Here the user indicates to Flexcom some fundamental pieces of information, including:

- The location (node number/label) in the finite element model which corresponds to the hub location in the aerodynamic model.
- The portion (set of elements) of the finite element model which represents the tower.

The hub location is particularly crucial. It is the central point for information exchange between the two solvers for all data pertaining to the aerodynamic loading on the rotating blades. Similarly, the tower nodes facilitate the inclusion of wind loads on the tower in the structural model.

At the beginning of a dynamic simulation, Flexcom passes all information pertaining to the blades and tower to AeroDyn - such as the initial hub location in terms of global XYZ coordinates, the initial shaft tilt angle, the initial nacelle yaw etc. AeroDyn then computes the aerodynamic loads using blade element momentum theory. The total loads on the rotating blades are assembled at the hub, and passed back to the Flexcom solver where they are added to the global force vector on the right hand side of the equations of motion. Similarly, wind loads on each tower node in the AeroDyn model are passed to the corresponding structural node in the Flexcom solver, and the global force vector is augmented accordingly.

Once all the constituent terms have been assembled, Flexcom solves the Finite Element Equations of Motion, and the global solution vector (predominately consisting of displacement terms) is populated. The updated locations for the hub and tower nodes are then passed back to AeroDyn, and the aerodynamic computations are performed again, before the updated wind loads and passed back to the structural model. This solution progresses in an iterative manner until Solution Convergence has been achieved in the structural solver. The solution time is then advanced by one Fixed Time Step and the whole process recommences. The iterative nature of this solution scheme ensures that full coupling between the structural and aerodynamic models is achieved.

Flexcom 2022.1.1 (August 2022) is coupled with OpenFAST V2.6.0 (May 2021).

COUPLING BETWEEN FLEXCOM AND INFLOWWIND

In addition to a coupling with AeroDyn, Flexcom also interfaces to the OpenFAST module InflowWind, via the *INFLOWWIND keyword, so dynamic and turbulent wind loading can be specified. InflowWind processes wind inflow data and supports several wind file formats including uniform, binary TurbSim full-field, binary Bladed-style full-field and HAWC formatted binary full-field wind files. It also has its own internal calculated steady wind and supports arbitrary wind directions. The turbulent wind field definition is typically generated in advance using TurbSim - refer to TurbSim Overview for further information.

At each timestep, InflowWind is given the blade and tower nodal locations and then calculates the undisturbed wind-inflow velocities at these positions. These are then used by AeroDyn to compute the aerodynamic loads at the relevant locations in the Flexcom model. Refer to InflowWind Overview for further information on this module.

COUPLING BETWEEN FLEXCOM AND SERVODYN

The *SERVODYN keyword provides a link between Flexcom and the OpenFAST module ServoDyn. ServoDyn provides an interface between a user generated control dynamic link library (DLL) and solution variables provided by AeroDyn and Flexcom.

Flexcom Wind implements variable rotor speed control by allowing the user to modify generator torque. The turbine speed is varied by changing the generator or control torque via the DLL which is linked to Flexcom. Generator torque is computed at each iteration and turbine speed is calculated based on the relationship between angular acceleration and net torque:

$$\dot{\omega} = \frac{1}{J}\left(\tau_{Aero} - \tau_{Generator}\right)$$

where ω = rotor speed,

τ_{Aero} = Aerodynamic torque,

$\tau_{Generator}$ = Generator torque,

J = Rotor Inertia.

The rotor speed is then updated at each time step, typically to maximise energy capture.

When operating above the rated wind speed Flexcom provides blade pitch control, which is provided via the *SERVODYN keyword, so as to shed additional power. Blade pitch is updated at each time step in the same manner as rotor control.

Yaw control is also provide via the *SERVODYN keyword where the yaw rate as calculated via the user defined dynamic link library. The finite elements defining the turbine structure (as indicated by the *TURBINE SET=* definition) are then yawed as requested.

Software Modelling Limitations

INTRODUCTION

There are a number of limitations associated the modelling capability as it currently stands. The ability to model floating wind turbines is a relatively recent addition to the software, and the feature will grow and develop over time. In the meantime, users should be familiar with the underlying assumptions which the current model is based on. Limitations are sub-divided into aerodynamic and hydrodynamic for ease of reference.

AERODYNAMIC MODEL

- A simplified RNA model is used in Flexcom 8.13, as an intermediate step in the software development process. It is planned to address these limitations using a more advanced RNA model in the next version.
    - Blade flexures under applied loading are assumed to be negligible, hence the blade geometries are approximated as rigid profiles. This means that variations in aerodynamic loading due to dynamic blade deformations are not captured.
    - Due to the rigidity simplification, the blades are not explicitly modelled using finite elements. Hence blade rotational inertia effects are not included in the simulation. Only the static weight of the blades is taken into account.
    - The shaft is not modelled explicitly, and is instead represented by a rigid element which connects the hub location to the top of the tower. This element is non-rotational and its sole purpose is to transfer the aerodynamics loads from the hub to the tower. As the low-speed shaft torque is unknown in the current Flexcom model, it is derived from the generator torque and the gearbox ratio, and this leads to a smoother signal than would be observed in reality.

- AeroDyn is capable of modelling one-, two-, or three-bladed rotors, but from a Flexcom perspective, a three-bladed configuration is implicitly assumed. Given that three-bladed turbines are the most common type, there are no immediate plans to support one- or two-bladed rotors.

- AeroDyn is designed to model horizontal-axis wind turbines only. It is not possible to model vertical-axis wind turbines, so there are no immediate plans to make this functionality available in Flexcom.

HYDRODYNAMIC MODEL

- As noted in the section on Floating Platform Loads, hydrodynamic loading on the platform is based on information which is derived from a radiation-diffraction analysis. These simulations compute pressure integrals based on Bernoulli's equation around the surface of a floating body, allowing forces and moments at the body's centre of gravity to be derived. Flexcom is not capable of performing a radiation-diffraction simulation, so you will need access to a suitable hydrodynamic solver. There are several established commercial codes available such as WAMIT and ANSYS Aqwa, while NEMOH is a popular open-source tool.

- A linear solution technique is generally adopted, which assumes that displacements of the free surface and the floating body away from their mean positions remain relatively small, such that the boundaries may be linearised, and this simplifies the wave-structure interactions significantly. A fully non-linear solution, which considers the instantaneous free surface and body surface as a function of time, is inherently more accurate but also very computationally expensive. There are no immediate plans to develop a fully-coupled non-linear potential solver approach with Flexcom. In any case, for bodies with relatively simple geometry (standard platforms are effectively composed of one or more submerged cylinders), the linear approach is assumed to be adequate, particularly for operational seastates where wave heights are limited.

- Wave induced loads are typically concentrated at a single location, such as the centre of gravity, rather than being distributed over the wetted surface area of the platform. Radiation-diffraction codes readily provide the total loads at the CoG so this is the most convenient means of transferring loads to the structural model. It is possible to apply the loads in a distributed manner, but this would require knowledge of the pressure forces acting over each of the hydrodynamic panels, and mapping the derived forces to relevant locations on the structural model. In summary, it would require significant additional effort on the part of the software user.

- As noted in the section on Floating Platform Loads, second order wave drift loads are computed using QTF inputs. Second order drift loads stem from two different sources, at difference frequency pairs between two different incident wave harmonics (ω_1-ω_2) and sum frequency pairs between harmonics (ω_1+ω_2) . Sum frequencies tend to be very high frequency and are not typically of interest, but difference frequencies are very important as they induce slow drift effects at frequencies much lower that that of the incident wave harmonics. Strictly speaking, a full QTF matrix is required for complete accuracy, whereby QTF coefficients are specified for all possible difference frequency pairs, and this may be further complicated if multi-directional wave loading is experienced. Furthermore, the calculation procedure for evaluating the second order loads with this level of precision is extremely computationally expensive. For efficiency, Flexcom uses a well-established approximation (Newman, 1974), which only requires the diagonal terms of the full QTF matrix to be specified. The off-diagonal terms are then approximated using an average of the relevant diagonal entries. This is generally regarded as acceptable given that the difference frequencies of most interest relate to those of similar frequency - i.e. adjacent harmonics, which have similar diagonal QTF terms anyway. Despite its advantages, the accuracy of Newman's approximation has been challenged, particularly for shallow water environments. It is not currently possible to model second order wave drift loads in Flexcom using the full QTF method, but it is hoped to add this functionality in the near future.

Validation

The numerical modelling capabilities have been validated via several studies, including:

OC4

OC4 (Offshore Code Comparison Collaboration Continuation) is a code-to-code verification project operated under the IEA (International Energy Agency) Wind Task 30 and coordinated by NREL (National Renewable Energy Laboratory). In Phase II of OC4, participants used an assortment of simulation codes to model the coupled dynamic response of a 5-MW wind turbine installed on a floating semi-submersible in 200 m of water. Code predictions were compared from load case simulations selected to test different model features. Although Flexcom was not officially represented in OC4, the software has been retrospectively benchmarked against OC4 results, as NREL and the IEA have kindly made all data from the project publicly available. The software validation is primarily focused on the OC4 semi-submersible platform (Connolly & O'Mahony, 2021), but comparisons are available for the OC4 jacket structure (Connolly & O'Mahony, 2021) also.

OC6

OC6 (Offshore Code Comparison Collaboration, Continued with Correlation and unCertainty) is a further extension of the OCx research series. Flexcom is officially represented for the first time via Queen's University Belfast who are using the software in OC6 Phase I (Robertson et al., 2020). This phase of the project is focused on examining why offshore wind design tools under-predict the response of the OC5 semi-submersible at its surge and pitch natural frequencies, by means of new validation campaigns designed to separately examine the different hydrodynamic load components.

EOLINK FWT

Flexcom was used by Eolink to simulate their innovative floating wind turbine concept. Results from the numerical simulations were validated with empirical data derived from model-scale tank tests (Connolly et al., 2018) and sea trials (Guyot et al., 2019).

1.9.4 Analysis

OVERVIEW

The various analysis types in Flexcom are described in detail in various sections of this manual, but a brief overview of the different types is appropriate at this juncture in order to place time domain analysis into perspective.

- A Static Analysis considers time invariant loading and structure response, and is specified for most analyses that do not contain wave loading.
- The presence of waves, whether regular or random, with or without current and/or vessel motions, requires you to specify a dynamic analysis. As Flexcom has traditionally been a time domain analysis tool, existing users may instinctively associate "'dynamic analysis" with Time Domain Analysis.
- Frequency Domain Analysis is far more efficient than time domain, but its use is restricted to systems whose behaviour is (largely) linear. Where the technique can be applied, the computational effort is only a fraction of that required for a corresponding time domain simulation, resulting in huge savings in terms of run time.

- A Quasi-Static Analysis is a (typically highly damped) time domain dynamic analysis in which the applied loads and displacements are constant after an initial ramping on period. The damping increasingly dissipates inertia effects and the final solution achieved approaches that of a static one. A quasi-static analysis is required in a small number of highly sensitive cases when a genuine static analysis cannot be successfully performed.

FURTHER INFORMATION

Further information is contained in the following sections:

- Static Analysis discusses time invariant loading and associated structure response.

- Restart Analyses describes the program's restart facility, used for example to starting a dynamic analysis from a preceding static configuration.

- Time Domain Analysis discusses loading and associated structure response, and includes sections on Time Variables and Time Integration Algorithms.

- Frequency Domain Analysis summarises the theory underpinning the operation of the frequency domain analysis capability.

- Modal Analysis summarises the theory underpinning the operation of the modal domain analysis capability.

- Time Domain Fatigue Analysis summarises the operation of the time domain fatigue analysis capability in Flexcom.

- Frequency Domain Fatigue Analysis summarises the operation of the frequency domain fatigue analysis capability in Flexcom.

- Specialised Solution Topics discusses some advanced topics in relation to finite element analysis, and includes sections on Solution Convergence, Bandwidth Optimisation and Gaussian Quadrature.

- Troubleshooting Simulation Failures presents some general guidelines regarding the use of Flexcom, and also gives some advice on what to do if an analysis fails to complete successfully.

1.9.4.1 Static Analysis

INTRODUCTION

A static analysis considers time invariant loading and structure response, and is specified for most analyses that do not contain wave loading.

The restart facility is described in detail in Restart Analyses, but a brief introduction to the topic is appropriate at this juncture to provide a background to static analysis.

RESTART ANALYSES

A given design condition will typically require a structure to withstand a combination of static and dynamic loads and displacements. In performing a Flexcom analysis for these inputs, it should be possible to apply all static and dynamic loads and displacements in a single dynamic run. However, this is frequently not possible, and is rarely desirable. A better option is to invoke the program Restart facility, whereby you can specify that a particular analysis is to be restarted from a previous run, to build up to the full dynamic solution in stages. In a restart, the structure configuration at the end of the preceding analysis becomes the starting configuration for the restart. New loads or displacements are combined with those in the preceding run, and the structure response to the combined loading is found in a static or dynamic solution.

RELATED TOPICS

Further information on static analysis is contained in the following sections:

- Time Variables in Static Analysis describes the significance of time variables in a Flexcom static analysis and explains the rationale behind their inclusion.

- Solution Criteria Automation describes the solution criteria automation feature, which allows you to specify desired values of critical solution parameters. The program automatically adjusts the model configuration to satisfy the specified criteria.

RELEVANT KEYWORDS

- `*ANALYSIS TYPE` is used to specify the analysis type.

- `*TIME` is used to define time parameters for an analysis.

- *CRITERIA is used to specify certain criteria that need to be satisfied in a static analysis, and to define how the model is to be adjusted to satisfy the desired criteria.

Time Variables in Static Analysis

RATIONALE

Naturally, all time domain analyses require the specification of time variables. Since a static analysis is one which considers time invariant loading and structure response, time variables are effectively meaningless in this context. On that basis, it may seem curious that you are also required to specify time variables for a static analysis in Flexcom. This is purely a consequence of the historical evolution of the software. Although Flexcom now incorporates frequency domain analysis capabilities, the program has traditionally been primarily a time domain analysis tool. Given the importance of static analysis in terms of providing a platform for subsequent dynamic analysis, versions of Flexcom right from the beginning have provided a static analysis capability in addition to the (effectively core) time domain dynamic analysis capability. In an effort to standardise the solution inputs across all analysis types, time variables became mandatory for static analyses also, even if the specified values are somewhat notional. This has remained the case ever since, for various reasons including the following:

(i) Maintaining keyword compatibility with previous versions

(ii) Consistency of restart information storage and retrieval for static and time domain analysis

(iii) Consistency of database output storage and retrieval for static and time domain postprocessing

LOAD RAMPING

While the situation regarding time variables in static analysis is somewhat artificial and essentially a legacy stemming from early versions of Flexcom, the time variables do nonetheless serve a useful function in providing control over the rate of build-up of the applied static loading

As the time variables are largely notional in a static analysis, an initial static analysis is typically run from t=0 to t=1 second, while a static restart analysis (for example, to introduce a vessel offset) is typically run from t=1 to t=2 seconds. Any further static restarts (for example, to introduce current loading etc.) would normally be run from t=2 to t=3 seconds, from t=3 to t=4 seconds, and so on.

In many cases static loads and displacements can be applied in a single step, and if so, you are not required to specify any value of Time Step – this defaults to the difference between the analysis start and end times. If a time step is specified, then the static loads and displacements are built up to their full values at the end of the analysis over the specified number of steps. So for example in a static analysis running from t=1 to t=2 seconds with a fixed time step of 0.1 seconds, the loads increase linearly to their full value over 10 steps.

Note also that the Ramp Time entry is intended specifically for dynamic analysis, to allow control over the build-up of dynamic loads and displacements in an analysis with waves and/or vessel motions. For example, wave loads in a regular wave analysis are typically ramped on over 1 wave period, with the solution proceeding for a further 3-4 wave periods to achieve a steady state solution. Any Ramp Time specified in a static analysis is simply ignored, as the static loads and displacements are built up to their full values over the entire simulation time.

Two options are provided in terms of Ramp Type, both Linear and Nonlinear. With the former, the loads increase linearly to their full values between the analysis start to end times. When a Nonlinear ramp is specified (and this is the default), a half cosine ramp function is used (as opposed to a linear function).

TIME STEPPING

Flexcom has two time stepping algorithms, fixed and variable. When you specify a fixed time step, the simulation proceeds from the start time to the end time using the fixed time step specified. In a variable step analysis, the choice of time step magnitude is made by the program based on a number of criteria. The time step is continuously monitored and varied as appropriate by the program within user-specified limits, to ensure a stable and convergent solution.

The vast majority of static analyses use a fixed time step, although a variable time step may be necessary in some extreme cases. The relevant inputs in this context are a Suggested Time Step, a Minimum Time Step, a Maximum Time Step, and optionally a Step Length parameter. The first three parameters are self-explanatory. The Step Length parameter pertains specifically to dynamic analyses, but is mentioned briefly here for completeness. If you are using a variable time step, the program chooses an optimum time step based on two main criteria, namely (i) the number of iterations required for the last three convergent solutions, and (ii) the instantaneous current response period, which is a measure of the dominant period in the response at any particular instant. The optimum time step, when this is based on the dominant response period, is obtained by multiplying the instantaneous current period by the Step Length parameter. As the notion of a dominant response period is meaningless for static analysis, any value you specify for Step Length is simply ignored. The variable step algorithm is most typically employed in dynamic analyses, but is also available also in static runs. For static analyses, the size of the step is selected only on the basis of the number of iterations required for convergence. The larger the number of iterations, the smaller the size of the time-step, and vice versa.

RELEVANT KEYWORDS

- `*TIME` is used to define time parameters for an analysis.

If you would like to see an example of how this keyword is used in practice, refer to any of the standard Flexcom Examples.

Solution Criteria Automation

THEORY

Flexcom allows you to specify desired values of critical solution parameters, and the program automatically adjusts the model configuration to satisfy the specified criteria. For example, you could specify the maximum allowable bending moment in the touchdown zone for an SCR, or the desired hang-off angle at the vessel connection. You also have control over how the model is to be adjusted, for example using a vessel offset or a change in line length. The increased automation provided by this new feature will expedite model set-up and improve engineering productivity.

Note that criteria apply to static analyses only. Required criteria may be specified in an initial static analysis, or in any subsequent static restart analysis. Automation on solution parameters is not possible in a dynamic simulation.

The inputs are defined in terms of a criterion type (i.e. a solution parameter), a criterion monitor (i.e. a particular region of interest), a target value and, optionally an allowable tolerance on the target.

Most of the criteria types represent standard program outputs, but a small number require further elaboration. The full list of accepted criteria is as follows.

1. Axial Stress.
2. Axial Strain.
3. Bending Stress.
4. Bending Strain.
5. Bending Moment.
6. Hangoff Angle. This is the angle in degrees that the specified element is required to make with the global x-axis. Note that unlike many of the other criteria types, a single element (rather than a set of elements) is used to monitor the hang-off angle.
7. Support Reaction. This is the contact reaction force at a guide surface. Note that unlike many of the other criteria types, the name of a guide surface (as opposed to a set of elements) is used to monitor the contact reaction.
8. Seabed Contact. This is the reaction at a node or nodes due to contact with the seabed. Note that unlike many of the other criteria types, a single node, or set of nodes (as opposed to a set of elements) is used to monitor the seabed contact reaction.
9. Tension.
10. Von Mises Stress.

You have control over how the requested criteria are to be achieved. For example, the adjustment variable may be a node, a vessel, or an element set. In the case of a node or vessel adjustment, Flexcom moves the selected node or vessel along a line parallel to the designated vector as it attempts to satisfy the specified criteria. Regarding the selected vector, Flexcom initially applies motion in a positive sense with respect to the vector, so the vector should ideally be set up such that it tends to aid criteria convergence if possible. In any case, Flexcom will immediately switch to motion in a negative sense if criteria divergence is exhibited in the first iteration. In the case of an element set length adjustment, Flexcom will adjust the length of the selected element set as it attempts to satisfy the specified criteria. The initial change in length may be positive or negative – Flexcom automatically estimates the direction most likely to aid criteria convergence.

Multiple criteria may be specified in a single analysis. If this is the case, the order in which the criteria are specified determines the priority. Specifically, Flexcom will attempt to satisfy the first criteria, and only proceed to the second and subsequent criteria after the first has been satisfied. While multiple criteria may be specified, only one parameter can be iterated upon in order to satisfy the specified criteria.

The solution parameters should only be varied if Flexcom is finding it difficult to attain a configuration which satisfies the specified criteria. The default values are adequate for the majority of analyses. For example, you may wish to reduce the Min Adjustment to ensure convergence if you have specified a very small deviation tolerance for your criterion. The minimum adjustment defaults to 0.1m or 0.328ft (or 0.1 degrees for angular adjustments), and the maximum adjustment defaults to 50m or 164.042ft (or 10 degrees for angular adjustments), depending on the unit system being used. Note also that there is no need to specify the time step under the `*TIME` keyword because the number of solution steps is automatically governed by the Max Iterations input.

RELEVANT KEYWORDS

- `*CRITERIA` is used to specify certain criteria that need to be satisfied in a static analysis, and to define how the model is to be adjusted to satisfy the desired criteria.
    - `*PRINT` is used to request additional printed output to the main output file. Specifically, the `OUTPUT=CRITERIA` option is used to request additional information regarding solution convergence towards specified criteria.

If you would like to see an example of how these keywords are used in practice, refer to B01 - Steel Catenary Riser.

1.9.4.2 Restart Analyses

THEORY

A given design condition will typically require a structure to withstand a combination of static and dynamic loads and displacements. In performing a Flexcom analysis for these inputs, it should be possible to apply all static and dynamic loads and displacements in a single dynamic run. However, this is frequently not possible, and is rarely desirable. A better option is to invoke the program Restart facility, whereby you can specify that a particular analysis is to be restarted from a previous run, to build up to the full dynamic solution in stages. In a restart, the structure configuration at the end of the preceding analysis becomes the starting configuration for the restart. New loads or displacements are combined with those in the preceding run, and the structure response to the combined loading is found in a static or dynamic solution.

The major application of this facility is in starting a dynamic analysis from a static configuration due to gravity, buoyancy, current and applied constant loads and displacements. The advantage of this procedure in the time domain is that it minimises the influence of initial transients associated with the application of the dynamic loads, with a consequent saving in CPU effort in the dynamic run. In terms of the frequency domain, a necessary prerequisite is that the dynamic analysis restarts from a converged static solution.

The use of restarts is also frequently mandatory in the analysis of flexible risers and mooring systems. Because of their low bending stiffnesses these systems depend largely on tension to resist applied loads. When static loads are applied in an initial step or steps, the system develops a resistance to lateral loads such as those due to waves and current, which are then applied in restart runs.

Further information on this topic is contained in the following sections:

- Restart Types outlines the two basic restart types, determined by changes in loads and boundary conditions.
- Data Specification in a Restart categorises the various program inputs in terms of which parameters can and cannot be altered in a restart.

RELEVANT KEYWORDS

- *RESTART is used to indicate that an analysis is to be restarted from a previous run.

If you would like to see an example of how this keyword is used in practice, refer to any of the standard Flexcom Examples.

Restart Types

Flexcom has in fact two restart types. The first involves the application of new loads or boundary conditions in the restart. The case discussed above of a static run followed by a dynamic run falls into this category. When you choose this option, Flexcom takes care to ramp on or gradually build up to their full values only the new loads or boundary conditions, while those loads which were at their full values at the end of the preceding run remain so. This restart type is the one most frequently used.

The second restart type is a simple continuation where the restarted run takes up where the preceding run finished with no new loads or boundary conditions applied. Applications of this option include the staggering of a long simulation over several shorter runs or the continuation of a dynamic regular wave analysis when steady state conditions have not been achieved.

In order to facilitate restart analyses, Flexcom automatically outputs to a binary restart file, at the end of each analysis, sufficient information to enable a subsequent restart to be performed. Earlier versions of Flexcom actually provided an option to request that the restart file also be optionally written periodically while an analysis proceeds, so that in theory if an analysis terminated abruptly before completion (perhaps due to a power failure or a machine crashing), then it would not be necessary to perform the whole analysis again from the very start. In practice however, it was found that the facility was very rarely invoked, and the associated reduction in run-time performance was not justifiable (writing all the restart data to the hard drive at regular intervals can be quite time consuming), so the feature was removed altogether.

Note that the preceding discussion of restart types related specifically to time domain analysis. As already noted, frequency domain dynamic analyses must restart from a converged static solution. So there is essentially only one restart category in this case, corresponding to the first type described above.

Data Specification in a Restart

As you can appreciate, the data required in a restart analysis is substantially the same as that in the analysis from which it is restarted. In Flexcom (with a few minor exceptions) you are required to specify only new or altered inputs in setting up the restart run. In fact, a large amount of data (in particular the finite element mesh) cannot be altered in a restart, and the program will object if you attempt to invoke any of the options appropriate to specifying this data. This section categorises the Flexcom input data in terms of those inputs that can and cannot be altered in a restart. The categories represent a logical division of the data and the reason for a particular input being in a particular category is normally self-evident.

Flexcom in fact groups the input data into three categories for this purpose. The first category consists of data that must be specified in the very first of a series of consecutive analyses. This data is carried through to all subsequent restarts and may not be changed. This category naturally includes the finite element discretisation, structural and hydrodynamic properties, plus any other inputs which characterise the initial model configuration (e.g. initial vessel position, seabed properties, ocean depth etc.) – basically any information which cannot logically change from run to run are contained in Category 1.

The second category consists of parameters that may be entered in any run of a series of runs. This data is automatically carried through to all subsequent runs, but may of course be subsequently altered again. This category naturally includes environmental parameters such as current and waves, boundary conditions of various kinds, internal fluid loading, and the analysis and solution type.

Strictly speaking, the majority of parameters which relate specifically to frequency domain dynamic analysis (e.g. harmonic loads), or those which also relate to the time domain but are being discussed in the context of the frequency domain (e.g. wave loading) do not fall into this category. The reason being that all such analyses must restart from converged static solutions, so it is not entirely true to say that the parameters that may be entered in any run of a series of runs. Furthermore, it is not possible to perform another restart analysis subsequent to a frequency domain dynamic analysis. However, such parameters cannot be accurately classified by categories one or three either, so it is logical to include them in this category, along with inputs of similar purpose in the time domain.

The final category consists of parameters which may be specified in any run of a series of runs but which do not carry through to subsequent runs. Rather they are specific to that particular run. The most obvious of these are the solution variables (such as time, damping and convergence parameters for example) – these must be specifically input for each analysis stage. Other parameters would include data storage requests (i.e. database and timetrace), stress properties etc.

1.9.4.3 Time Domain Analysis

THEORY

This section discusses the time domain analysis capability in Flexcom. The presence of waves, whether regular or random, with or without current and/or vessel motions, requires you to specify a dynamic analysis. As Flexcom has traditionally been a time domain analysis tool, existing users may instinctively associate "'dynamic analysis" with "time domain".

- Time Variables describes the significance of time variables in a Flexcom dynamic analysis, and includes sub-sections discussing 'Fixed Time Stepping', 'Variable Time Stepping', 'Choice of Time Step', 'Simulation Length' and 'Load Ramping'.
- Time Integration Algorithms describes the algorithms available in Flexcom for the discretisation of finite element equations of motion in time.
- Quasi-Static Analysis discusses the use of quasi-static analysis in circumstances where a converged static solution is difficult to achieve.
- Damping presents the range structural damping options available in Flexcom.

It is worth mentioning that the solution procedure in a time domain analysis may be specified as either Linear or Nonlinear. A linear analysis is one in which the structure stiffness, mass and damping matrices are assembled once only, at the simulation start time. The use of this option is valid where the structure displacements subsequently are such that it is reasonable not to reassemble these matrices. If this is not the case, a non-linear analysis is appropriate. The vast majority of time domain analyses are non-linear, and this option is selected by default.

RELEVANT KEYWORDS

- `*ANALYSIS TYPE` is used to specify the analysis type.

- *TIME is used to define time parameters for an analysis.
- *TIME STEPPING is used to select the time stepping algorithm and to define associated numerical damping coefficient.

If you would like to see an example of how these keywords are used in practice, refer to any of the standard Flexcom Examples.

Time Variables

Flexcom has two time stepping algorithms, Fixed and Variable. Whether you opt for a fixed or variable time step, it is important to choose a Time-Step which picks up necessary detail in excitation and response. Selection of an appropriate Simulation Length depends on whether you are running a regular wave or random sea dynamic analysis. Load Ramping is provided to allow control over the build-up of dynamic loads and displacements in an analysis with waves and/or vessel motions.

Fixed Time Stepping

THEORY

When you specify a fixed time step, the simulation proceeds from the start time to the end time using the fixed time step specified. Generally speaking, a fixed time step is used in analyses where the nature of structural response is broadly similar throughout. For example, in the case of a rigid riser, and/or benign rather than extreme loading conditions. It is well suited to lengthy simulations for a stable model, and is the preferred option provided the required time step is reasonable.

RELEVANT KEYWORDS

- *TIME is used to define time parameters for an analysis. Specifically, the STEP=FIXED inputs are used to specify fixed time step data.

If you would like to see an example of how this keyword is used in practice, refer to A01 - Deepwater Drilling Riser.

Variable Time Stepping

THEORY

In a variable step analysis, the choice of time step magnitude is made by the program based on a number of criteria. The time step is continuously monitored and varied as appropriate by the program within user-specified limits, to ensure a stable and convergent solution. A variable step is typically used in analyses where the structural response varies significantly during the course of the simulation. A good example might be that of a steel catenary riser with intermittent seabed contact. However, a variable time step may also be useful during the development phase for any particular model, regardless of the nature of structural response. It is useful in this context to determine if a fixed time step is potentially suitable or not, and if so, to provide an indication regarding a suitable value of time step. Some other points worth noting in relation to variable time step analyses are as follows:

- A variable time step introduces a computational overhead, due to constant monitoring of various parameters to facilitate the selection of an optimum time step (these parameters are discussed in more detail shortly).
- Changes of time-step can introduce numerical noise into the solution.
- Caution is advised against the specification of an excessively large maximum time-step value in an effort to speed up the analysis. The variable time stepping algorithm will try to achieve the maximum value at every possible opportunity, and this can actually make the overall solution less efficient, or even affect solution robustness in certain circumstances.

The relevant inputs in the context of a variable time step analysis are Suggested Time Step, Minimum Time Step, Maximum Time Step, and optionally a Step Length parameter. The first three parameters are self-explanatory. If you are using a variable time step, the program chooses an optimum time step based on two main criteria, namely (i) the number of iterations required for the last three convergent solutions, and (ii) the instantaneous current response period, which is a measure of the dominant period in the response at any particular instant.

The mathematical definition of the current period parameter T_p was initially developed by Bergan et al., (1985), and applied to the analysis of impact loading on rigid beam structures.

In a Flexcom time domain dynamic analysis, T_p is computed at the nth solution time using the equations:

$$\omega_p^2 = \frac{\sum_e (\delta \underset{\sim}{d}_e^T \underline{K}_e \, \delta \underset{\sim}{d}_e)}{\sum_e (\delta \underset{\sim}{d}_e^T \underline{M}_e \, \delta \underset{\sim}{d}_e)} \qquad (1)$$

and:

$$T_p = \frac{2\pi}{\omega_p} \qquad (2)$$

where:

- $\delta \underset{\sim}{d}_e$ is the incremental nodal displacement vector. The increment is the difference in $\underset{\sim}{d}$ for the element between the (n-1)th and nth solution times, that is, $\delta \underset{\sim}{d}_e = (\underset{\sim}{d}_n - \underset{\sim}{d}_{n-1})_e$

- $\underline{K}_e$ is the element stiffness matrix

- $\underline{M}_e$ is the element mass matrix

and the summation is performed over all the elements of the structure.

The optimum time step, when this is based on the dominant response period, is obtained by multiplying the instantaneous current period by the Step Length parameter. So for example the default Step Length value of 0.055 means the time step is approximately 1/18th of the dominant period in the dynamic response at any time. The default value is adequate in almost all cases, but may occasionally be increased (say to 0.1 or 1/10th) by experienced users who feel the resulting time step is too short.

It is shown by Bergan. & Mollestad, (1985) that ω_p is a weighted average of the dominant frequencies comprising the instantaneous response of the structure. So if the structure is responding in a high frequency mode, ω_p will be large, and consequently T_p and by definition the optimum step size will be small. The opposite is true for low frequency compliant response.

The sole use of the current period parameter as a means of determining the time-step size would result in frequent and possibly large changes in the size of the step during a dynamic analysis. This in turn would introduce undesirable numerical oscillations into the solution. To counteract this, an efficient scheme has been implemented such that the size and frequency of time-step changes are moderated, and an optimum stable step is reached, if possible. This scheme is based on two main principles, namely (i) not changing the step size if the difference between old and new values is below a certain tolerance, and (ii) basing decisions on whether to change the time-step (though not decisions on the actual time-step value to use) on the variation in the current period over the last number of solution times rather than just the last one.

RELEVANT KEYWORDS

- `*TIME` is used to define time parameters for an analysis. Specifically, the `STEP=VARIABLE` inputs are used to specify fixed time step data.

If you would like to see an example of how this keyword is used in practice, refer to C02 - Multi-Line Flexible System.

Choice of Time Step

Whether you opt for a fixed or variable time step, it is important to choose a time-step which picks up necessary detail in excitation and response. This applies to all time domain dynamic analyses, but it is particularly relevant to those which include significant non-linearities. As a general rule of thumb, it is recommended to use a time step of between 1/12th and 1/20th of the applied wave period in a regular wave analysis.

However, much smaller time steps may be required if the analysis includes any significant non-linearity, such as intermittent contact between a risers and a guide surface for example. Impact situations tend to cause structure response at high frequencies. This problem is exacerbated by the fact that, in general, the element length around the area of contact tends to be small, which can lead to increased structure response at these high frequencies.

If you are particularly concerned about the time step you have chosen, a useful exercise might be to run some sensitivity analyses, to examine sensitivity of results to the time step size. You would typically expect to see the solution converge towards a consistent set of results for all time steps below a reasonable threshold level.

Another general principle of finite element analysis is that in theory you should always attempt to match the temporal and spatial discretisations. This is a slightly more difficult concept to embody, and requires a considerable level of finite element experience. Basically it means that if you are using a relatively fine finite element mesh, then you should be using a relatively fine time step. Conversely, if you are using a relatively coarse mesh, then a relatively coarse time step will be suitable.

Simulation Length

Selection of an appropriate simulation length depends on whether you are running a regular wave or random sea dynamic analysis. In regular wave analysis, the simulation length needs only to be long enough for the effects of initial transience to dissipate fully and for steady state response to be established. Generally speaking, a simulation length of 4-6 times the period of the regular wave is sufficient. However, there may be some exceptional cases when it takes considerably longer for steady state conditions to be fully established. For example, a catenary riser subjected to cross current loading may experience “riser walking” – a phenomenon which involves lateral displacement of the touchdown zone increasing over successive wave periods. For this reason, it generally advisable to check some sample time histories of response to ensure that steady state conditions have been attained, rather than simply assuming this is the case, and proceeding directly to envelopes of response over the last couple of wave periods.

In the case of random sea dynamic analysis, the simulation time needs to be much longer than that of a regular wave analysis. The rationale behind this is as follows. The input to a random sea analysis is in the form or a wave spectrum (or spectra). Flexcom discretises the spectrum into discrete component harmonics, and uses these to generate a time history of random surface elevation. Each harmonic is essentially a regular wave, with its own individual amplitude, period, direction and random phase angle. As the assignment of phase angles is a completely random process, the resulting time history of water surface elevation represents only one possible realisation of a frequency domain excitation in the time domain. For this reason, the results of interest are really the statistics of response (as opposed to the actual time histories themselves). So the analysis must be run for long enough to provide confidence in the statistical measures – in order words until the response statistics reach steady state. A simulation time of 1800s (0.5 hours) would be considered a lower bound, but random sea analyses are often run for 10,800s (3 hours). Such lengthy simulation times obviously have implications in terms of run time and storage capacity.

Load Ramping

A ramp time entry is provided to allow control over the build-up of dynamic loads and displacements in an analysis with waves and/or vessel motions. For example, wave loads in a regular wave analysis are typically ramped on over one wave period, with the solution proceeding for a further 3-4 wave periods to achieve a steady state solution.

Two options are provided in terms of ramp type, both linear and nonlinear. With the former, the loads increase linearly to their full values between the analysis start to end times. When a Nonlinear ramp is specified (and this is the default), a half cosine ramp function is used (as opposed to a linear function).

Time Integration Algorithms

For time domain solutions, the equilibrium equations of motions are integrated in time using a step-by-step procedure. So rather than solving the equations at any time t, the solution is obtained at discrete time steps, Δt apart. In order to facilitate this approach, assumptions must be made about the variation of displacement, velocity and acceleration within each time interval. Implicit integration is used exclusively by Flexcom, whereby the solution at time $t+\Delta t$ is based on the conditions at time $t+\Delta t$. This means the exact equations of motion are solved for at each solution time, ensuring that precise equilibrium is achieved. This approach contrasts with that of explicit integration, where the solution at time $t+\Delta t$ is based on the conditions at time t. This means the equilibrium equations are never exactly solved for at any solution time, so very fine time steps are generally required to maintain the accuracy and stability of the solution.

Flexcom provides two algorithms for the discretisation of finite element equations of motion in time, namely the Hilber et al. (1977) integration method and the Generalised-α method (Chung. and Hulbert, 1993). Both approaches are similar in many regards – the Hilber-Hughes-Taylor operator is in fact a special case of the Generalised-α method. In early versions of Flexcom, Hilber-Hughes-Taylor was the only option available, but the options have been extended to include Generalised-α as the program evolved.

The Generalised-α method achieves an optimal combination of high-frequency and low-frequency dissipation, so that for any given value of high frequency dissipation, the low frequency dissipation is minimised. This is particularly beneficial for analyses with high frequency noise, such as for example those involving intermittent contact, but should also in general allow most dynamic analyses to run with larger time steps than the Hilber-Hughes-Taylor operator. Further details are provided in Theory.

Earlier versions of Flexcom (up to and including Flexcom 8.10) used the Hilber-Hughes-Taylor operator as the default. Generalised-Alpha has since been shown to provide more effective numerical damping, particularly for sensitive models, so it is now the default method. If you are re-running some old simulations which previously used Hilber-Hughes, it is possible that you may notice some very slight differences in results.

RELEVANT KEYWORDS

- `*TIME STEPPING` is used to select the time stepping algorithm and to define associated numerical damping coefficients.

Theory

The basic finite element equation of motion is:

$$\underset{-}{M}\,\underset{\sim}{\ddot{d}}+\underset{-}{C}\,\underset{\sim}{\dot{d}}+\underset{-}{K}\,\underset{\sim}{d}=\underset{\sim}{F} \qquad (1)$$

where $\underset{-}{M}$, $\underset{-}{C}$, and $\underset{-}{K}$ are the mass, damping, and stiffness matrices, respectively, $\underset{\sim}{F}$ is the vector of applied loads, and $\underset{\sim}{d}$ is the vector of displacement unknowns.

Time integration algorithms such as Generalised-α and Hilber-Hughes-Taylor have the following common form. The vectors $\underset{\sim}{d}_n$, $\underset{\sim}{\dot{d}}_n$, and $\underset{\sim}{\ddot{d}}_n$ are given approximations to $\underset{\sim}{F}(t_n)$, $\underset{\sim}{\dot{F}}(t_n)$ and $\underset{\sim}{\ddot{F}}(t_n)$, respectively. Expressions for $\underset{\sim}{d}_{n+1}$ and $\underset{\sim}{\dot{d}}_{n+1}$ are specified as linear combinations of $\underset{\sim}{d}_n$, $\underset{\sim}{\dot{d}}_n$, $\underset{\sim}{\ddot{d}}_n$ and $\underset{\sim}{\ddot{d}}_{n+1}$. Algorithms having this form may be classified as one-step, three-stage (or three-level) time integration methods. The algorithms are one-step methods because the solution at time t_{n+1} depends only on the solution history at time t_n. The three-stage designation refers to the solution being described by the three solution vectors: $\underset{\sim}{d}_n$, $\underset{\sim}{\dot{d}}_n$, and $\underset{\sim}{\ddot{d}}_n$.

The basic form of the generalized-a method is given by:

$$\underset{\sim}{d}_{n+1}=\underset{\sim}{d}_n+\Delta t\,\underset{\sim}{\dot{d}}_n+\Delta t^2\left(\left(\frac{1}{2}-\beta\right)\underset{\sim}{\ddot{d}}_n+\beta\underset{\sim}{\ddot{d}}_{n+1}\right) \qquad (2)$$

$$\underset{\sim}{\dot{d}}_{n+1}=\underset{\sim}{\dot{d}}_n+\Delta t\left((1-\gamma)\underset{\sim}{\ddot{d}}_n+\gamma\underset{\sim}{\ddot{d}}_{n+1}\right) \qquad (3)$$

$$\underset{-}{M}\,\underset{\sim}{\ddot{d}}_{n+1-\alpha_m}+\underset{-}{C}\,\underset{\sim}{\dot{d}}_{n+1-\alpha_f}+\underset{-}{K}\,\underset{\sim}{d}_{n+1-\alpha_f}=\underset{\sim}{F}(t_{n+1-\alpha_f}) \qquad (4)$$

$$\underset{\sim}{d}_{n+1-\alpha_f}=(1-\alpha_f)\underset{\sim}{d}_{n+1}+\alpha_f\underset{\sim}{d}_n \qquad (5)$$

$$\dot{d}_{n+1-\alpha_f}=(1-\alpha_f)\underset{\sim}{\dot{d}}_{n+1}+\alpha_f\underset{\sim}{\dot{d}}_n \qquad (6)$$

$$\dot{d}_{n+1-\alpha_f} = (1-\alpha_f)\dot{\underset{\sim}{d}}_{n+1} + \alpha_f \dot{\underset{\sim}{d}}_{n} \quad (7)$$

$$\ddot{d}_{n+1-\alpha_m} = (1-\alpha_m)\ddot{\underset{\sim}{d}}_{n+1} + \alpha_m \ddot{\underset{\sim}{d}} \quad (8)$$

$$t_{n+1-\alpha_f} = (1-\alpha_f)t_{n+1} + \alpha_f t \quad (9)$$

The structure of the displacement (2) and velocity (3) update equations above is obtained by restricting the sum of the coefficients of their acceleration terms to equal the coefficient of the acceleration term in a Taylor series expansion of $\underset{\sim}{d}(t_{n+1})$ and $\dot{d}(t_{n+1})$ about t_n . Simple numerical experiments have shown that this update equation structure results in a monotone increase per period in the peak displacement and velocity errors. The modified balance equation, (4), is effectively a combination of the Hilber-Hughes-Taylor and Wood et al.(1981) balance equations. With appropriate expressions for γ and β, if α_m = 0, the algorithm reduces to the Hilber-Hughes-Taylor method. The generalized-α method is second-order accurate, provided:

$$\gamma = \frac{1}{2} - \alpha_m + \alpha_f \quad (10)$$

High-frequency dissipation is maximized when:

$$\beta = \frac{1}{4}\left(1 - \alpha_m + \alpha_f\right)^2 \quad (11)$$

You are referred to publications on the Generalised-α method by Chung, and Hulbert (1993), and the Hilber-Hughes-Taylor method by Hilber et al. (1977), for further discussion of these step-by-step time integration algorithms.

Quasi-Static Analysis

THEORY

Very occasionally the response of a structure to static loading cannot be successfully found in a static analysis. This is normally due to the fact the structure is highly sensitive to changes in configuration from iteration to iteration, and a static analysis may tend to diverge rather than converge to a stable configuration. One of the most commonly occurring cases of this sensitivity is in systems which include significant water surface piercing, such as for example a buoyancy tank used during a riser tow-out. If an initial approximation to the equilibrium position of the structure actually has an excess of, say, buoyancy over gravity, in the next iteration the structure may be largely or wholly out of the water, and subsequent iterations may find alternately increasing portions of the structure exposed or submerged, and the solution rapidly diverges. In such circumstances, a quasi-static approach is more suitable, and the model will tend settle in a dynamic fashion, typically under the influence of damping, towards its static equilibrium configuration.

A quasi-static analysis is in reality a dynamic analysis in which the applied loads and displacements are constant after an initial ramping on period. Typically, a large mass damping coefficients is specified, and this damping increasingly dissipates inertia effects and the final solution achieved approaches a static one. The reason why a quasi-static analysis is successful where a static run is not is because the inertia of the system prevents the oscillations between unrealistic configurations described previously. In a quasi-static analysis a variable step is typically used, and the time-step tends to increase quickly as the transients dissipate. In choosing the optimum time-step, Flexcom bases the choice on the ambient current period value (as described in Variable Time Stepping earlier in this section) in the early part of the simulation when loads and displacements are being ramped on and the dynamic response may be significant. Beyond the ramp time however, the dynamics begin to dissipate and the use of the current period as a step size indicator is no longer suitable. As in the case of a static analysis, the choice of time-step during this phase is based only on the number of iterations required for convergence. In fact this dual strategy for computing the optimum time-step during the two phases of an analysis is the only way in which the conduct of a quasi-static analysis differs internally in Flexcom from a dynamic run.

In summary, the following general guidelines in relation to quasi-static analysis are useful:

(i) A variable time step should be used, with a relatively long run time to allow the structure to settle towards static equilibrium. In order to check if the run time you have chosen is sufficiently long, you should examine time histories of nodal displacements and/or velocities at critical locations. Note also that in some circumstances, it may be possible to successfully perform a subsequent static restart analysis.

(ii) A relatively small minimum time step should be specified in order to avoid any potential convergence issues.

(iii) A relatively large maximum step should be specified for efficiency. Caution is advised against the specification of an excessively large maximum time-step value in an effort to speed up the analysis. The variable time stepping algorithm will try to achieve the maximum value at every possible opportunity, and this can actually make the overall solution less efficient, or even affect solution robustness in certain circumstances.

(iv) The static loads should be ramped on relatively quickly, again for efficiency, by specifying a relatively short ramp time.

(v) Inclusion of a significant level mass-proportional damping is usually quite beneficial in dissipating transients. For example, a coefficient of 1.0 would not be unusual. Obviously this would be excessively high in the context of a normal dynamic analysis (e.g. regular wave or random sea), but a quasi-static analysis merely serves as a means to attaining a static equilibrium configuration.

RELEVANT KEYWORDS

- `*ANALYSIS TYPE` is used to specify the analysis type.
- `*TIME` is used to define time parameters for an analysis. Specifically, the `STEP=VARIABLE` inputs are used to specify fixed time step data.
- `*DAMPING` is used to incorporate damping into a dynamic analysis.

If you would like to see an example of how these keywords are used in practice, refer to E03 - Floating Hose.

Damping

THEORY

Flexcom presents a range of options to apply structural damping to the finite element model. This section briefly outlines the background theory for structural damping.

The structure damping matrix $\underline{C}$, which multiplies the vector of structure velocity on the left hand side of the equations of motion, is defined by:

$$\underline{C} = \lambda \underline{K} + \mu \underline{M} \qquad (1)$$

where:

- $\underline{K}$ is the structure stiffness matrix (this includes contributions from both structural stiffness and geometric stiffness)
- $\underline{M}$ is the structure mass matrix
- λ is the stiffness proportional damping coefficient
- μ is the mass proportional damping coefficient

A small level of damping is beneficial in many dynamic analyses in dissipating the effects of transients and high frequency noise. However, what particular values of l and m represent a small level of damping is very much dependent on the structure under consideration. For this reason, and because it is obviously important to quantify the effect particular values are having on the response in a particular run, it would in general be recommended that you perform a dynamic analysis with no damping, either initially or as a check.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Damping Coefficients describes stiffness and mass proportional damping coefficients.
- Damping Ratio describes coefficients defined as a function of a damping ratio and period.
- Deformation Mode Damping describes deformation mode specific damping.

- Damping Formulation explains the formulation options, updated and constant.

RELEVANT KEYWORDS

- `*DAMPING` is used to incorporate damping into a dynamic analysis.
- `*DAMPING FORMULATION` is used to specify the damping formulation to be used in a time domain dynamic analysis.
- `*DAMPING RATIO` is used to specify stiffness damping coefficients as a function of a damping ratio and a damping period.

Damping Coefficients

THEORY

Structural damping is generally specified directly in terms of stiffness and mass proportional damping coefficients as described in Damping (Eq.1). By default, a single value of each coefficient is assigned to all elements in the model. However, it is also possible to assign different damping coefficients to various element sets within the model, in which case individual entries of $\underline{K}$ and $\underline{M}$ are scaled by appropriate values of λ and μ, and $\underline{C}$ is assembled by a matrix addition of the relevant component terms. Note that this option is only available in time domain dynamic analyses, and any damping specified in a frequency domain analysis is applied to all elements in the model.

RELEVANT KEYWORDS

- `*DAMPING` is used to incorporate damping into a dynamic analysis.

Damping Ratio

THEORY

Rather than explicitly defining a coefficient (or coefficients), stiffness proportional damping may alternatively be defined as a function of a damping ratio and a damping period in a time domain dynamic analysis. The damping matrix is assembled in exactly the same manner as described in the Damping (Eq.1), however in this case λ and μ are computed from user-specified values of ξ and T as follows:

$$\lambda = \frac{\xi T}{\pi}; \ \mu = 0 \qquad (1)$$

where:

- ξ is the damping ratio
- T is the damping period

The damping ratio ξ must be between 0 and 1. If the excitation in your dynamic analysis is a single regular wave, then the specification of damping period T is optional. If omitted, T defaults to the regular wave period. If the excitation is of any other type, then the period input is required.

As with explicit values of λ and μ, it is also possible to assign different values of ξ and T to various element sets within the model. Similarly, individual entries of $\underline{K}$ and $\underline{M}$ are scaled by appropriate values of λ and μ, and $\underline{C}$ is assembled by a matrix addition of the relevant component terms. Again this option is only available in time domain dynamic analyses, and any values of ξ and T specified in a frequency domain analysis are applied to all elements in the model. Note also that the two specification options (i.e. damping ratio/period and stiffness/mass coefficients) are mutually exclusive for any given element set.

RELEVANT KEYWORDS

- `*DAMPING RATIO` is used to specify stiffness damping coefficients as a function of a damping ratio and a damping period.

Deformation Mode Damping

THEORY

Specification of deformation mode damping allows you to define individual stiffness damping coefficients for each of the structure deformation modes, namely, axial, bending and torsion. When this option is invoked the damping matrix $\underline{C}$ is based on the linear stiffness matrix $\underline{K}_{Lin}$, the geometric stiffness matrix $\underline{K}_{Geom}$ and the mass matrix $\underline{M}$, using the relation:

$$\underline{C} = \lambda_{Axial}\underline{K}_{Lin}^{Axial} + \lambda_{Bending}\left(\underline{K}_{Lin}^{Bending} + \underline{K}_{Geom}^{Bending}\right) + \lambda_{Torsion}\underline{K}_{Lin}^{Torsion} + \mu\underline{M} \qquad (1)$$

where:

- λ_{Axial} , $\lambda_{Bending}$ and $\lambda_{Torsion}$ are the stiffness proportional damping coefficients for the axial, bending and torsional modes, respectively
- μ is the mass proportional damping coefficient
- the superscripts Axial, Bending and Torsion as applied to the stiffness matrix terms indicate that these terms represent sub-matrices of the overall stiffness matrix comprising only axial, bending and torsion terms, respectively

Note that deformation mode damping is not available for frequency domain analysis.

RELEVANT KEYWORDS

- `*DAMPING` is used to incorporate damping into a dynamic analysis.

Damping Formulation

THEORY

The structure damping matrix is by default computed at each solution step based on the instantaneous mass and stiffness matrices, and this known as the Updated damping formulation in program terminology. Alternatively, a constant formulation may be invoked, in which case damping matrix remains constant throughout the analysis, based on the mass and stiffness matrices at the static equilibrium position (that is, the position at the start of the dynamic analysis).

RELEVANT KEYWORDS

- `*DAMPING FORMULATION` is used to specify the damping formulation to be used in a time domain dynamic analysis.

1.9.4.4 Frequency Domain Analysis

OVERVIEW

This section summarises the theory underpinning the operation of the frequency domain analysis capability in Flexcom. The aim of the note is not to provide a mathematically exhaustive description of the solution procedure – it is intended to provide an adequate introductory background, and to explain the different steps in a frequency domain analysis.

Frequency domain analysis is far more efficient than time domain, but its use is restricted to systems whose behaviour is (largely) linear. Where the technique can be applied, the computational effort is only a fraction of that required for a corresponding time domain simulation, resulting in huge savings in terms of run time. A frequency domain dynamic analysis restarts from a converged static solution (e.g. an initial static or current static analysis), in much the same way as a time domain dynamic analysis would. Flexcom performs the frequency domain dynamic analysis (by expressing the applied loading and structural response in terms of sinusoidal components), and then automatically performs a restart static analysis (executed internally by the software so no additional user action is required). This step is necessary because the linearised drag forces due to current, in an analysis with waves and current, are a function of the dynamic response due to wave action (further details are provided later in this section). Appropriate use of the frequency domain is the responsibility of the user, but it should be noted that any non-linearities associated with the applied loading or structural response should be relatively small for the underlying assumptions of frequency domain theory to remain valid.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Evolution of Flexcom Software outlines the rationale for merging the frequency domain analytical capabilities of Freecom (formerly a standalone frequency domain package) into Flexcom.
- Mathematical Background describes the theory underpinning the frequency domain analysis procedure.
- Static Solution discusses the method used to find the mean or static structure response.

- Dynamic Solution explores how the dynamic response about this mean solution is computed.

- Drag Linearisation discusses the techniques used to linearise the drag component of Morison's equation.

- Summary of Solution Procedures presents a summary of the solution procedures for completeness.

RELEVANT KEYWORDS

- `*ANALYSIS TYPE` is used to specify the analysis type.

- `*HYDRODYNAMIC SETS` is used to assign hydrodynamic coefficients to element sets.

If you would like to see an example of how these keywords are used in practice, refer to B01 - Steel Catenary Riser.

Evolution of Flexcom Software

Originally, Flexcom had a sister package called Freecom, which was a frequency domain finite element offshore analysis package. With the advent of Flexcom 7.7, a major milestone was reached – the frequency domain analytical capabilities of Freecom were merged into Flexcom, the net effect being that Freecom ceased to exist as a standalone entity. The decision to merge the two programs was taken for a variety of reasons, including:

- Freecom had two basic modes of operation, the stand-alone mode in which it performed all stages of the analysis (static and dynamic) itself, and the restart mode in which it restarted from a Flexcom static analysis. The latter mode was more powerful, as Freecom on its own was not capable of performing static analysis of a wide range of structures (such as SCRs and flexible risers), and so required Flexcom to evaluate the structure stiffness matrix. So since Freecom relied on Flexcom to a large extent, it was logical to merge the two programs.

- The dependency of Freecom on Flexcom described above also meant that compatibility had to continuously maintained between the two programs. Generally speaking, this meant that every time a new version of Flexcom was released, an updated version of Freecom was also required to maintain compatibility, even if no technical enhancements had been made to Freecom itself. Naturally, this scenario was not ideal, and inefficient in many cases.

Mathematical Background

This section provides some of the mathematical background to the frequency domain analysis procedure. Matrix algebra is used extensively in the derivations that follow. A matrix quantity is denoted by **bold** typeface, for example $\boldsymbol{w}$. An uppercase matrix variable, for example $\boldsymbol{K}$, represents a rectangular array or tensor, that is an array comprised of m rows and n columns, where m and n are both greater than 1. A lowercase matrix quantity, for example $\boldsymbol{f}$, represents a vector quantity, that is a matrix comprising m rows and one column only.

The dynamic equations of motion for a multi-degree of freedom offshore structure subjected to regular wave and current can be expressed in standard finite element fashion as:

$$\boldsymbol{M}\ddot{\boldsymbol{w}} + \boldsymbol{C}\dot{\boldsymbol{w}} + \boldsymbol{K}\boldsymbol{w} = \boldsymbol{f} \qquad (1)$$

where $\boldsymbol{K}$ is the total structure stiffness matrix; $\boldsymbol{C}$ is the total structure damping matrix; $\boldsymbol{M}$ is the total structure mass matrix; $\ddot{\boldsymbol{w}}$, $\dot{\boldsymbol{w}}$ and $\boldsymbol{w}$ are respectively the vectors of nodal accelerations, velocities and displacements; and $\boldsymbol{f}$ is the vector of applied nodal loads. Note that the damping matrix $\boldsymbol{C}$ is given by:

$$\boldsymbol{C} = \lambda\boldsymbol{K} + \mu\boldsymbol{M} \qquad (2)$$

where λ and μ are user-specified stiffness and mass damping coefficients.

The development of the frequency domain equations is based on a decomposition of the load vector $\boldsymbol{f}$ into two parts, time-dependent and time-independent loads. The former are the hydrodynamic loads experienced by the structure due to waves and vessel motions, while the latter include buoyancy and gravity loads and fluid loading due to current. User-specified mechanical loads can fall into either category. In the same way the nodal displacements are decomposed into time-dependent and time-independent components, where the constant component is the mean displacement and the time-dependent displacements represent the periodic motion of the structure about this mean position. This division of loads and displacements into constant and time-varying can be represented by the equations:

$$\boldsymbol{f} = \boldsymbol{f_s} + \boldsymbol{f_d} \tag{3}$$

and

$$\boldsymbol{w} = \boldsymbol{w_s} + \boldsymbol{w_d} \tag{4}$$

where the subscript s denotes static (time-independent) and d dynamic (time-dependent). Of course:

$$\dot{\boldsymbol{w}}_{\boldsymbol{s}} = \ddot{\boldsymbol{w}}_{\boldsymbol{s}} = \boldsymbol{0} \tag{5}$$

so that:

$$\dot{\boldsymbol{w}} = \dot{\boldsymbol{w}}_{\boldsymbol{d}} \quad \text{and} \quad \ddot{\boldsymbol{w}} = \ddot{\boldsymbol{w}}_{\boldsymbol{d}} \tag{6}$$

Based on this, the first Equation above can be written as:

$$\boldsymbol{M}\ddot{\boldsymbol{w}}_{\boldsymbol{d}} + \boldsymbol{C}\dot{\boldsymbol{w}}_{\boldsymbol{d}} + \boldsymbol{K}\boldsymbol{w_d} + \boldsymbol{K}\boldsymbol{w_s} = \boldsymbol{f_d} + \boldsymbol{f_s} \tag{7}$$

This equation is also decomposed into two parts, representing time-dependent and time-independent response as follows:

$$\boldsymbol{M}\ddot{\boldsymbol{w}}_{\boldsymbol{d}} + \boldsymbol{C}\dot{\boldsymbol{w}}_{\boldsymbol{d}} + \boldsymbol{K}\boldsymbol{w_d} = \boldsymbol{f_d} \tag{8}$$

and:

$$\boldsymbol{K}\boldsymbol{w_s} = \boldsymbol{f_s} \tag{9}$$

The solution procedure entails the assembly and solution of the two Equations (8) and (9) above to find $\boldsymbol{w_d}$ and $\boldsymbol{w_s}$. The total solution $\boldsymbol{w}$ is then found from Equation (4). The force vectors $\boldsymbol{f_d}$ and $\boldsymbol{f_s}$ both contain non-linear hydrodynamic drag components, and a linearisation procedure must be adopted if the equations are to be solved directly in closed form. The technique adopted here has the effect of coupling the two equations through the linearisation procedure, that is the two linearised equations are not independent, and the solutions $\boldsymbol{w_d}$ and $\boldsymbol{w_s}$ must be found iteratively. Further details are provided later in the section.

The development to this point presumed a regular wave analysis, with or without current. In the case of a random sea analysis, where the seastate is characterised by a wave energy spectrum, the analysis begins by decomposing the input wave spectrum into a series of harmonic components. A pair of linearised equations, similar to Equations (8) and (9), is then assembled and solved at each component frequency, with the iteration loop in this case extending over all frequencies in the seastate discretisation. The drag force linearisation has the effect in this case of coupling the linearised equations at all of the component harmonics. This point is elaborated later.

Static Solution

INITIAL STATIC SOLUTION

This section discusses the solution of Mathematical Background (Eq.9), the equation for the mean or static displacement component $\boldsymbol{w_s}$ of the total solution vector $\boldsymbol{w}$. The total stiffness matrix on the left-hand side of Mathematical Background (Eq.9) (which also appears on the left-hand side of Mathematical Background (Eq.8) is the sum of two components, the ordinary linear stiffness matrix $\boldsymbol{K^L}$ and the geometric stiffness matrix $\boldsymbol{K^G}$. This latter term includes the effect of the distribution of effective tension in the structure on the structure bending stiffness. This effective tension distribution is a function of the loading on the structure due to, for example, gravity, buoyancy and any user-specified point or distributed loads (such as a riser top tension). Applied constant displacements (for example a mean vessel surge) also influence the effective tension distribution. It is a basic premise of the frequency domain development that the dynamic variation in the geometric stiffness matrix is negligible and can be ignored.

So the first step in a frequency domain analysis is to evaluate the total stiffness matrix $\boldsymbol{K}$. As mentioned previously, the frequency domain analysis is always preceded by a separate initial static analysis (this is logically termed the initial static solution), from which $\boldsymbol{K}$ is determined without any further computation.

FULL STATIC SOLUTION

Current forces in the initial static solution may not be entirely accurate. The reason why this is the case is related to the drag linearisation algorithm, which was mentioned previously in Mathematical Background and is further discussed later in the section. For now, it is only necessary to note that linearised drag forces due to current, in an analysis with waves and current, are a function of the dynamic response due to wave action, whether a regular wave or random sea. So the current forces cannot be evaluated with complete accuracy until the dynamic response has been found.

What this means is that an analysis with waves and current is a three step sequence with first an initial static analysis (to evaluate $\boldsymbol{K}$); then the dynamic analysis, to find the response to a single harmonic (regular wave) or a discretised spectrum (random sea); followed finally by a so-called full static solution, which involves the assembly and solution of Mathematical Background (Eq.9) with accurate current forces included in the (total) static force vector at this stage. Like the initial static solution this last analysis phase is iterative, because large displacements can be considered, and so the full static solution is nonlinear. In this context it is worth noting that nonlinear effects can include sections of riser moving onto or off the seabed. This point is elaborated in Seabed Interaction.

Ideally you should include current in the initial static analysis, even though the current loading can be based only on the current velocity distribution as the dynamic response is as yet unknown. The rationale for this is that it provides the dynamic phase with an accurate an estimate as possible of the mean or static riser configuration. A good example is the fatigue analysis of SCRs, where an accurate assessment of the effect of current on the riser touchdown point can be important.

Dynamic Solution

This section discusses the solution of Mathematical Background (Eq.8), the equation for the dynamic response about the mean structure position. In the development that follows it is convenient to represent sinusoidal components in terms of complex variables, so it is understood that it is the real part of all terms containing $e^{i\omega t}$ which is actually being used in any equation.

The first step in deriving the frequency domain dynamic equation is to express the hydrodynamic forces on the structure and the resulting structure motions in terms of sinusoidal components. Consider a structure subjected to a regular (cosine) wave or a harmonic component of a discretised spectrum, whose circular frequency is w radians/second. The frequency domain development assumes the vector of nodal time-varying displacements $\boldsymbol{w_d}$ can be expressed as:

$$\boldsymbol{w_d} = \boldsymbol{w^o} e^{i\omega t} \qquad (1)$$

where $\boldsymbol{w^o}$ is a complex vector of nodal displacements (the use of the subscript ^{o} denotes a complex quantity). The complex entries of $\boldsymbol{w^o}$ contain information about both the amplitudes of the nodal displacements and their phasing relative to the incident wave harmonic. The vectors of nodal velocities and accelerations are found by differentiating the Equation (1) above once to get:

$$\boldsymbol{\dot{w}_d} = i\,\omega \boldsymbol{w^o} e^{i\omega t} \qquad (2)$$

and twice to give:

$$\boldsymbol{\ddot{w}_d} = -\omega^{2} e^{i\omega t} \qquad (3)$$

The left-hand side of Mathematical Background (Eq.8) can now be rewritten as follows:

$$\mathrm{M}\boldsymbol{\ddot{w}_d} + \boldsymbol{C\dot{w}_d} + \boldsymbol{Kw_d} = \boldsymbol{(}-\omega^{2}\boldsymbol{M} + i\omega\boldsymbol{C} + \boldsymbol{K)}\boldsymbol{w^o} e^{i\omega t} \qquad (4)$$

It is further assumed that the vector $\boldsymbol{f_d}$ can be expressed in a similar manner to the first Equation above as:

$$\boldsymbol{f_d} = \boldsymbol{f^o} e^{i\omega t} \qquad (5)$$

where again $\boldsymbol{f^o}$ is a vector of complex quantities combining amplitude and phase information. The final frequency domain form of Mathematical Background (Eq.8) can be written as:

$$(-\omega^2\boldsymbol{M} + i\omega\boldsymbol{C} + \boldsymbol{K})\boldsymbol{w^o} e^{i\omega t} = \boldsymbol{f^o}\, e^{i\omega t} \qquad (6)$$

Once the $e^{i\omega t}$ term is cancelled from both sides, Equation (6) above can be solved for the complex vector $\boldsymbol{w^o}$, from which the time-varying displacements can be found using Equation (1) above. Finally the total displacement vector $\boldsymbol{w}$ is found by combining $\boldsymbol{w_s}$ and $\boldsymbol{w_d}$ using Equation (4).

The major feature underpinning the success of any frequency domain method is the derivation of an accurate formulation for the complex force vector $\boldsymbol{f^o}$ to justify the assumption of Equation (5). This topic of Drag Linearisation is now discussed.

Drag Linearisation

The hydrodynamic forces on an offshore structure are calculated using the standard Morison's Equation formulation. The forces predicted using this approach include drag and inertia components. The inertia component is linear and does not present any difficulty with respect to the application of Dynamic Solution (Eq.5). However the drag component is quadratic, and this component must be replaced by an equivalent linear term chosen on the basis of a best fit between linear and non-linear forms, if the above frequency domain development is to remain valid. The development of this best fit linearisation is summarised here. It is not however proposed to present detailed mathematics - interested readers are referred to relevant publications for detailed presentations. A more detailed version of the material presented in this section is described by McNamara. & Lane (1991). The program regular wave drag linearisation is based on the procedure described by Hamilton (1980). The program random sea drag linearisation follows the method of Langley (1984).

The formulation of the hydrodynamic drag via Morison's equation is based on the relative fluid/structure velocity vector $\boldsymbol{v_r}$ at any point on the structure, which is given by:

$$\boldsymbol{v_r} = \boldsymbol{v_w} - \boldsymbol{v_s} + \boldsymbol{v_c} \qquad (1)$$

Here $\boldsymbol{v_w}$ is water particle velocity due to waves, $\boldsymbol{v_s}$ is structure velocity and $\boldsymbol{v_c}$ is current velocity. All of these vectors are expressed in global coordinate axes. Morison's equation forces are more meaningfully calculated in a local axis system with components normal and tangential to an arbitrarily inclined element. We transform $\boldsymbol{v_r}$ into $\boldsymbol{v_r}$ in this local system (the subscript ^ denoting local axes), with components:

$$\hat{\boldsymbol{v}}_{\boldsymbol{r}} = \begin{bmatrix} \hat{v}_{rt} \\ \hat{\boldsymbol{v}}_{rn} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{v_r}.\boldsymbol{i})\boldsymbol{i} \\ \boldsymbol{v_r} - (\boldsymbol{v_r}.\boldsymbol{i})\boldsymbol{i} \end{bmatrix} \qquad (2)$$

where $\boldsymbol{t}$ and $\boldsymbol{n}$ indicate vectors tangential and normal to an element, and $\boldsymbol{i}$ is a unit vector in the tangential direction.

Tangential and normal pressure vectors due to fluid drag are now defined using Morison's equation as follows:

$$\hat{\boldsymbol{p}} = \begin{bmatrix} \hat{p}_t \\ \hat{\boldsymbol{p}}_n \end{bmatrix} = \frac{1}{2}\rho_w D_d \begin{bmatrix} C_{dt}\, \hat{v}_{rt} \left| \hat{v}_{rt} \right| \\ C_{dn}\, \hat{\boldsymbol{v}}_{rn} \left| \hat{\boldsymbol{v}}_{rn} \right| \end{bmatrix} \qquad (3)$$

where ρ_w is water density, D_d is drag diameter, and C_{dt} and C_{dn} are respectively tangential and normal drag coefficients.

The development of the form of the full non-linear drag involves writing expressions for $\boldsymbol{v_w}$ and $\boldsymbol{v_s}$ similar to Dynamic Soluation (Eq.1) and Dynamic Solution (Eq.5). From these, expressions for $\hat{v}_{rt}$, $\hat{\boldsymbol{v}}_{rn}$ and ultimately $\hat{\boldsymbol{p}}$ can be found. However the form of $\hat{\boldsymbol{p}}$ in the Equation above makes it clear that Dynamic Solution (Eq.5) will not be valid.

Drag linearisation involves replacing the quadratic forms in Equation (3) above by equivalent linearised terms chosen to minimise the error between linear and nonlinear forms. The linearised pressure vector can be written as:

$$\hat{\boldsymbol{p}} = \begin{bmatrix} \hat{p}_t \\ \hat{\boldsymbol{p}}_n \end{bmatrix} = \frac{1}{2}\rho_w D_d \begin{bmatrix} C_{dt}\{B_{1t}(\hat{v}^o_{wt} - \hat{v}^o_{st})e^{i\omega\omega} + B_{2t}\hat{v}_{ct}\} \\ C_{dn}\{\boldsymbol{B}_{1n}(\hat{\boldsymbol{v}}^o_{wn} - \hat{\boldsymbol{v}}^o_{sn})e^{i\omega\omega} + \boldsymbol{B}_{2n}\hat{\boldsymbol{v}}_{cn}\} \end{bmatrix} \quad (4)$$

Here the velocity terms due to waves and current ($\boldsymbol{v_w}$ and $\boldsymbol{v_c}$ respectively) and the structure velocity $\boldsymbol{v_s}$ have been rotated to local axes and resolved into tangential and normal components. The $\boldsymbol{B}$ terms are the linearisation coefficients, which take various forms for wave loading with and without current, and for regular waves or random seas. Note that the tangential linearisation coefficients and and are scalars, since the tangential drag is independent of the normal components and is effectively one-dimensional. For the normal direction on the other hand the linearisation coefficients B_{1t} and B_{2t} are matrices.

There are two important points to note about Equation (4) above. Firstly, it does permit the decomposition of drag forces into time-varying and time-independent components as required by Mathematical Background (Eq.3). Secondly, it does also allow the time-varying drag force to be expressed in the form of Dynamic Soluation (Eq.5).

The following are some general comments on the form of the linearisation coefficients, beginning with the dynamic force terms B_{1t} and $\boldsymbol{B_{1n}}$. How these are calculated for a point on the structure depends on whether the seastate is modelled as a regular wave or random sea. In a regular wave analysis the coefficients at a particular location are a function of the amplitude of the relative fluid/structure velocity at that point (Hamilton, 1980). Since this is a function of the (unknown) structure response, an iterative solution is required. For the first iteration the relative velocity is assumed equal to the water particle velocity, and a first estimate of the linearisation coefficients is calculated on this basis. Mathematical Background (Eq.9) is assembled and solved, and relative velocity and linearisation coefficients are updated based on this solution. The analysis proceeds in this way until the change in dynamic response from iteration to iteration is below a certain tolerance.

In a random sea analysis the coefficients are a function of the standard deviation of relative velocity (Langley, 1984), which is calculated by summing over all of the solution harmonic components. Again an iterative procedure is required, with the iteration loop in this case requiring a “sweep” over all the frequencies. The standard deviation of relative velocity and the dynamic drag linearisation coefficients are updated at the end of each sweep, and iteration continues until the change in relative velocity statistics from successive sweeps is less than a certain tolerance.

Turning now to the static force terms B_{2t} and $\boldsymbol{B_{2n}}$, it was mentioned earlier that these terms are functions of the dynamic response, and this is true in the case of both regular wave with current and random sea with current. So in both cases the so-called full static solution takes place after the completion of the (iterative) dynamic phase. The full static solution is again iterative, though not because the linearisation coefficients vary with the static solution (they do not), but because the static phase is nonlinear, as previously discussed.

Summary of Solution Procedures

REGULAR WAVE WITH CURRENT

The first step in the solution is to evaluate the total stiffness matrix $\boldsymbol{K}$ and the total mass matrix $\boldsymbol{M}$. As the frequency domain analysis is a restart from an initial static analysis, the initial static solution provides the required data.

An initial estimate is now made of the fluid/structure relative velocity distribution, and this is used to evaluate the drag linearisation coefficients B_{1t} and $\boldsymbol{B_{1n}}$. This in turn leads to the evaluation of the hydrodynamic loading distribution. The Dynamic Solution (Eq.6) is solved to give a first approximation to the dynamic displacements $\boldsymbol{W_d}$, which are then used to update B_{1t} and $\boldsymbol{B_{1n}}$. Dynamic Solution (Eq.6) is again assembled and solved. This continues until convergence is achieved, typically in relatively small number of iterations. The current drag force coefficients are now calculated, and Mathematical Background (Eq.9) is solved iteratively for the static displacements. Internal restoring forces and reactions corresponding to the total solution are finally evaluated, and the analysis concludes with the output of all solution variables.

RANDOM SEA WITH CURRENT

After discretising the input wave spectrum a random sea analysis proceeds in a manner analogous to that for the regular wave case. The total structure stiffness and mass matrices, $\boldsymbol{K}$ and $\boldsymbol{M}$, are first assembled as before. A first estimate for the distribution of standard deviation of relative velocity is now evaluated by looping over all the seastate harmonic components, using the wave particle velocity at each frequency as a first approximation to the unknown relative velocity at that frequency. Values of B_{1t} and $\boldsymbol{B_{1n}}$ at each integration point on the riser are then found. Using these values the dynamic motion equation, Dynamic Solution (Eq.6), is now assembled and solved at each of the component frequencies, and the results of each solution stored. When this sweep is concluded, the stored results are used to again evaluate the distribution of standard deviation of relative velocity, and this leads to new values of the linearisation coefficients. The sweep over all component frequencies is now repeated, and the process continues until relative velocity standard deviations from successive sweeps are almost identical. At this point a final sweep is performed, during which the final static solution is also calculated. Then at each harmonic the total solution for that harmonic is obtained, the corresponding internal restoring forces and reactions are calculated, and all of the solution variables for that harmonic are output as for the regular wave case. This completes the random sea analysis. The frequency domain postprocessor can be used at this point to generate required statistical measures from the results at the individual harmonics.

1.9.4.5 Modal Analysis

OVERVIEW

This section summarises the theory underpinning the operation of the modal analysis capability in Flexcom. Its aim is not to provide a mathematically exhaustive description of the solution procedure, rather it is intended to provide an adequate introductory background. This section briefly outlines the analytical capabilities of Modes, an ancillary module of Flexcom, which performs modal analysis.

The objective of a modal analysis is to calculate the natural frequencies and mode shapes of a structure. Flexcom possesses an ancillary module, Modes, which computes the natural frequencies and corresponding three-dimensional mode shapes of offshore structures. A large variety of offshore systems can be analysed using Modes, including flexible and rigid risers, mooring lines, TLP tethers and complex multi-articulated towers. Particular features of the software are as follows:

- Modes can find the natural frequencies of any structure that can be analysed using Flexcom, including structures whose geometry definition uses cables.
- Natural frequencies are calculated using the subspace iteration technique, which is fast, accurate and efficient.
- Program output is in both tabular and graphical form. The graphical output consists of full, on-screen, 3D representations and plots of the structure mode shapes.
- Modes is fully integrated into the Flexcom, and the program inputs are specified in a similar fashion.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Mathematical Background describes the theory underpinning the modal analysis procedure.
- Seabed Modelling introduces the topic of riser-seabed interaction in modal analysis.
- Subspace Iteration outlines the subspace iteration technique, used by Modes to converge to the required eigensolution.
- Output File outlines the options for controlling the contents of the modal analysis output file.
- Solution Convergence notes some helpful tips for overcoming convergence issues in a modal analysis.
- SHEAR7 Interface describes the capability of Modes to interface with SHEAR7, a program which estimates fatigue damage caused by vortex induced vibrations. Modes presents information regarding natural frequencies, mode shapes etc. in a format ready for direct input to SHEAR7.

RELEVANT KEYWORDS

Modal Analysis

- `*ANALYSIS TYPE` is used to specify the analysis type.

- `*EIGENPAIRS` is used to specify the required number of natural frequencies, and also parameters relating to the subspace iteration algorithm.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser or B01 - Steel Catenary Riser.

SHEAR7 Interface

- `*RISER TYPE` (`$MODES` section) is used to specify the riser type and other parameters relating to the Modes facility for generating output for subsequent input to SHEAR7.
- Numerous keywords are contained in the `$SHEAR7` section, a relatively recent addition to Flexcom. This input section provides a streamlined interface with SHEAR7, allowing you to define all of the required SHEAR7 inputs directly within Flexcom itself, resulting in a seamless integration of the modal and VIV fatigue analysis stages.

For full details of how Modes interfaces with SHEAR7, refer to SHEAR7 Interfacing Operation.

Mathematical Background

This section provides some of the mathematical background to the operation of Modes. Matrix algebra is used in the derivations that follow. A matrix quantity is denoted by bold typeface, for example ***w***. An uppercase matrix variable, for example ***K***, represents a rectangular array or tensor, that is an array comprised of m rows and n columns, where both m and n are greater than 1. A lowercase matrix quantity, for example ***v***, represents a vector quantity that is a matrix comprising m rows and one column only.

The basic equation solved by Modes is:

$$K\,v = \omega^2\,M\,v \qquad (1)$$

where ***K*** is the total structure stiffness matrix; ***M*** is the total structure mass matrix; ω^2 is the structure eigenvalue; and v is the associated eigenvector or mode shape. The corresponding structure natural frequency f is given by:

$$f = \frac{\omega}{2\pi} \qquad (2)$$

Seabed Modelling

If you are analysing an SCR or a flexible riser (e.g. a free hanging, lazy S or lazy wave configuration) in a modal analysis, an important consideration is how the program models the interaction between with the riser and the seabed. There are two aspects to this issue, namely:

(i) Riser motion in the direction normal to the seabed.

(ii) Riser motion tangential to the seabed.

Both aspects are discussed in detail in Seabed Modelling in Modal Analysis and you are referred there for further information.

Subspace Iteration

The subspace iteration algorithm used in Modes to solve Mathematical (Eq. 1) is described at length in Bathe et al. (1976) and will be only briefly discussed here. The method is particularly suited to structural systems since in general only a small number of solution eigenpairs is required in these cases. Subspace iteration is basically a simultaneous inverse iteration procedure. A small base of vectors, whose number depends on the required number of eigenpairs, is first created; this defines the eigenproblem “subspace”, and this “subspace” is progressively transformed, by iteration, into the space comprising the lowest few eigenvectors of the overall system.

This procedure requires the complete solution of a reduced eigenproblem in each iteration, and this is done in Modes using the Householder and Q-R methods, which are very efficient methods for the complete solution of small eigensystems. The great advantage of subspace iteration is that the eigenpair extraction takes place in a reduced space, giving rapid convergence to the required eigensolution (Bathe et al., 1976). The method is stable, efficient and fast, with runtimes typically less than a couple of minutes.

When specifying the required number of natural frequencies it is recommended that you specify twice the actual required number you are interested in. For example, if the first 5 natural frequencies are of interest, 10 eigenpairs should be requested. This ensures that the actual required values are estimated to an acceptable accuracy (Bathe et al., 1976).

Output File

THEORY

In early program versions this file contained, in addition to a data echo and a summary list of modes, a table of modal displacements and rotations for each node of the finite element model, for each eigenpair of the solution. This output was certainly comprehensive, but its volume made it difficult to examine in detail. In more recent versions, the data echo section and the summary list of modes are still present, but you can now control directly what else the file contains.

You can control the file content in three ways. Firstly, you can specify directly what modes are to be included in the output; unless you specify otherwise, the output file does not contain details of any individual eigenpair. Secondly, you can nominate that output for each eigenpair actually included in the file is for a selected set of elements only, rather than all elements (which is the default). And thirdly, you can optionally include details of the static solution (used to determine the stiffness matrix K).

In addition to modal displacements and rotations, total and incremental curvatures and stresses are included in the output. Also, displacements are output in both global and local axes for a riser you have identified as an SCR. So for example for an SCR in-plane mode, displacements in axial and transverse directions are added to the more usual global X, Y and Z displacements.

RELEVANT KEYWORDS

- `*OUTPUT OPTIONS` is used to specify the eigensolution details to be routed to the output file.

Solution Convergence

THEORY

Eigenproblem Shift

The basic equation that Modes solves is K v = λ M v (as per Mathematical (Eq.1) presented earlier), where K and M are stiffness and mass matrices respectively, v is an eigenvector, and λ is the corresponding eigenvalue. Shifting by a value μ means transforming this equation into (K + μ M) v = η M v. The eigenvalues of the original and transformed equations are related by $\eta_i = \lambda_i + \mu$ for all i eigenvalues. Eigenvectors are identical. Bathe et al., (1976) claim that applying a shift to an eigenproblem can speed convergence and prevent problems when the stiffness matrix is positive semidefinite. By default Modes applies a shift of 1. In a very small number of analyses increasing this value can guarantee convergence which might otherwise not be achieved, but this option should be very rarely used.

Effective Compression and Local Buckling

Occasionally Flexcom can report the following error message during a modal analysis…

> "*Error: Solution stopped during subspace iteration. Stiffness matrix is not positive definite. Non-positive pivot for equation N, pivot=X.Y*"

Any error message relating to negative pivot terms in the global stiffness matrix generally stems from effective compression in the initial static solution. Where compression is present, this can lead to instability in the subsequent modal analysis, as the riser is susceptible to buckling and the modal solution becomes indeterminate. If this message appears, you should examine the effective tension distribution in the initial static analysis, and check to see if there are regions where the effective tension is negative.

Furthermore, even a successful modal analysis can occasionally present negative eigenvalues in the solution. These are unusual, and again may be caused by effective compression in the preceding static analysis stage.

If either of these issues are a cause of concern, there are a couple of options available to you…

- It may be possible to alter the loading on the model in order to avoid effective compression. For example, additional buoyancy material may be used to reduce the apparent weight of the structure under consideration.
- It may be possible to alter the structural properties, in order to augment the bending stiffness. The critical buckling load is related to bending stiffness, and if the resistance to bending can be increased, this may prevent local buckling.

RELEVANT KEYWORDS

- `*EIGENPAIRS` is used to specify the required number of natural frequencies, and also parameters relating to the subspace iteration algorithm, including an *Eigenproblem Shift* parameter.

SHEAR7 Interface

OVERVIEW

In fluid dynamics, vortex-induced vibrations (VIV) are motions induced by the periodic irregularities in the external flow around a cylinder due to separation of the boundary layer due to excessive curvature. Low pressure eddies, or vortices, are formed changing the pressure distribution along the surface. These vortices, when not symmetric around the body with respect to mid-plane lead to differences in lift forces on each side of the body. This imbalance in forces leads to a motion transverse to the direction of the flow. When the shedding frequency of the vortices is close to the natural period of the riser, lock-in can occur, causing periodic oscillations of the structure.

Because of these periodic oscillations, fatigue damage can be incurred by the structure. VIV can be an important consideration for an engineer in the design and analysis of offshores structures; especially in the design of long slender structures that we are often concerned about in Flexcom.

Most users will be familiar with the SHEAR7 program which estimates fatigue damage caused by vortex induced vibrations (VIV). SHEAR7 is a program developed by a team at MIT lead by Professor J. Kim Vandiver and is recognized as the primary software in the prediction of vortex-induced vibrations (VIV) for risers. The program allows users to estimate structural response, drag force amplification factors, and predict fatigue life. SHEAR7 evaluates which modes are likely to be excited by vortex shedding and estimates VIV response. In calculating riser VIV response, one of the SHEAR7 requirements is a knowledge of the riser natural frequencies, mode shapes, and modal curvature distributions. SHEAR7 does itself include an analytical module which can calculate this data for a large class of mainly simplified structural models. However if your riser does not correspond to one of these structural models, then you must supply the eigensolution data to SHEAR7 as part of your VIV analysis input data. Modes includes a facility to produce this data as part of the eigensolution output, and in a format ready for direct input to SHEAR7.The Flexcom-SHEAR7 interface has the capacity to aid the end user in the calculation of VIV fatigue life through the conversion of Flexcom model and mode shapes into SHEAR7 input and running SHEAR7.

This section describes the capability of Flexcom to interface with SHEAR7, a program which estimates fatigue damage caused by vortex induced vibrations. SHEAR7 must be licensed separately through AMOG.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Modal Data Specification
- SHEAR7 Data Specification
- Modes Operation
- Identifying Mixed Modes
- Additional Output and Repeat Runs
- SHEAR7 Interfacing Operation
- MDS File Format

RELEVANT KEYWORDS

- *RISER TYPE ($MODES section) is used to specify the riser type and other parameters relating to the Modes facility for generating output for subsequent input to SHEAR7.

- Numerous keywords are contained in the $SHEAR7 section, a relatively recent addition to Flexcom. This input section provides a streamlined interface with SHEAR7, allowing you to define all of the required SHEAR7 inputs directly within Flexcom itself, resulting in a seamless integration of the modal and VIV fatigue analysis stages.

For full details of how Modes interfaces with SHEAR7, refer to SHEAR7 Interfacing Operation.

Modal Data Specification

THEORY

Firstly, you must identify your riser as being a certain type. The are two main categories in this context; (i) top-tensioned risers (TTR) and (ii) catenary risers (SCR). Modes bases some important decisions on what modes to include in the SHEAR7 output on this classification. This is discussed in more detail shortly. If your riser does not fall into either of these categories, you may classify it a User defined riser type.

In terms of VIV data, one mandatory input is the number of segments of equal length to be used in generating the SHEAR7 data. The VIV program requires eigensolution details at equally spaced points along the riser, but your finite element model will rarely be based on elements of equal length. However Modes allows you complete flexibility in defining your FE model as you require via the Flexcom main analysis module, and then produces SHEAR7 input for an equal spacing you define in Modes, by interpolation between the values for the elements of your discretisation.

There are several other optional inputs. You may specify the number of modes to consider for output to the SHEAR7 input file(s). This defaults to all modes of the eigensolution, but you might want to exclude higher modes, as these are less accurately determined than the lower modes. If you invoke this option, only the lowest number of eigenpairs are considered for output. You may also specify a particular element set for output, in which case the SHEAR7 output is restricted to the elements of this set only. If you do not specify an element set, the composition of the default set depends on the riser type under consideration. If the riser type is TTR or User, then the default is all elements of the finite element mesh. If on the other hand the riser type is SCR, then the default is all elements that are not wholly on the seabed, starting at that element whose first node is the SCR touchdown point. A further optional input is a damping ratio for use with a subsequent SHEAR7 analysis. Note that this input is only relevant to wake interference analyses, where Flexcom can automatically run a SHEAR7 analysis to compute enhanced drag and lift coefficients.

RELEVANT KEYWORDS

- *RISER TYPE ($MODES section) is used to specify the riser type and other parameters relating to the Modes facility for generating output for subsequent input to SHEAR7.

SHEAR7 Data Specification

INTRODUCTION

Flexcom provides a number of keywords to allow you the flexibility to change any model input, calculation option, and output option as you see fit. While it may seem like a large number of options to control the SHEAR7 analysis, there are only a few required keywords: *RESTART, *CURRENT,*S-N CURVE and *FATIGUE DATA.

While not required, the keywords *VERSION, *FOLDER OPTION, and *ZONES are commonly used to facilitate more control over the analysis.

Additionally, there are a variety of other keywords to control other inputs; these include *SECTION COEFFICIENTS, *SECTION PARAMETERS, *FATIGUE OPTIONS, *OUTPUT FILES, *POWER RATIO EXPONENT, *CUTOFF MODES, *DAMPING RATIO, *RESPONSE, *HIGHER HARMONICS, *REFERENCE DIAMETER and *NON-ORTHOGONAL DAMPING. The various inputs and their significance are outlined below.

For further information on all Shear7 inputs, refer to the Shear7 User Guide which is available for download from the Shear7 Support web page.

DATA INPUTS

Section Properties

- Set Name. The name of the section (i.e. the element set name) whose properties are being defined. Note that any Set Name referenced under the Environmental category must have section properties defined here.

Restart

- Modes File Name. The location of the modal analysis that has been performed prior to the SHEAR7 analysis. This can be referenced as the path\filename, or the path\filename with the keyword extension.

Current

- Probability of Occurrence. The current's annual probability of occurrence. This defines the current profile's probability in one year. In SHEAR7 this value is multiplied by the computed damage.

- Profile ID. This value is the number reported by SHEAR7 in the output file to help you keep track of which current profile is being used.

- Current Option. The current option allows you to specify which format you wish to define the current. The options are ascending, descending or x/L. Current values are converted to x/L format before being writing to the SHEAR7 input file. The Flexcom output file presents the original and converted current profiles to facilitate inspection of the conversion process. If you not fully satisfied with the in-built conversion process in Flexcom, you can choose the x/L current definition option instead. This transfers the user specified data directly to Shear7, with no conversion process applied.

- Elevation and Velocity. The current elevation and current velocities.

    - Elevation is converted to SHEAR7 x/L format. A minimum of two points are required to define a current profile.

- VIV is only considered to be caused by flow that is normal to the riser. While the consideration of this is not generally an issue for TTRs because they can be seen to be vertical or mostly vertical and there is no specific vertical plane flow, this consideration has a heavy impact for in-plane current flow which excites out-of-plane modes for SCR and SLWR configurations. SHEAR7 only provides functionality whereby you can specify the normal flow velocity not taking into account the directionality. This may pose a problem in the modelling of SLWRs as the flow is actually reversed in the hog-bend. While it is not completely accurate to consider the flow all being in the same direction, this is the only way the SHEAR7 currently allows it, therefore it is generally an acceptable approach as it is typically more conservative. If you wish to use your own methodology for the application of current, an option has been reserved for explicit definition in terms of x/L as mentioned above.

- For TTRs and SCRs (or user-defined structures) analysed for in-plane VIV, the current conversion process simply uses linear interpolation based on nodal elevation to find the relevant current velocity at each x/L location.

- For SCRs (or user-defined structures) analysed for out-of-plane VIV, the current conversion process also uses linear interpolation in the same manner, but includes an additional adjustment to compute the current velocity normal to the structure. Specifically, it computes a local orientation based on the Euler angles in DOFs 5 & 6, combines these into a (resultant) local inclination angle, and then factors the current velocity by the cosine of this angle at each x/L location.

S-N Curve

- Curve Name. The name of the S-N curve.

- Endurance Limit. The stress range at which the number of cycles to failure is unlimited. SHEAR7 refers to this value as the cutoff stress range.

- Stress Range. The stress value that corresponds to the number of cycles to failure. In SHEAR7 as many as ten segments or 11 data pairs can be used to define an S-N curve. On a log-log plot the S-N curve is assumed to be linear between these points. The stress range for the different points must be in increasing order.

- Cycles to Failure. The number of cycles to failure at the reference stress range.

Fatigue Data

- Global SCF. The stress concentration factor to be applied to the entire structure. The stress computed by SHEAR7 for the entire structure is multiplied by this value before the damage rate is computed.

- Fatigue Set. The set which will correspond to defined S-N curves. If a set is not assigned an S-N curve, the first S-N curve defined will be assigned to the set.

- S-N Curve. The curve name as defined in the S-N Curve keyword. This curve will be assigned to the element set defined.

Version

- SHEAR7 Installation Path. The location of the SHEAR7 executable path.

- SHEAR7 Version. The version of SHEAR7 to be used. Flexcom currently supports SHEAR7 4.6, 4.7 and 4.8.

- CL Path. SHEAR7 provides lift tables to be used in the SHEAR7 analysis. The file is generally located in the SHEAR7 installation directory.

Folder Options

- Folder. The folder where the SHEAR7 analysis will be performed. The common.mds, and common.CL files will be moved to this location and the SHEAR7 input file will be generated.

Zones

- Element Set. This defines a set of elements that will be assigned consistent structural properties for the purpose of the SHEAR7 model. SHEAR7 recommends not trying to model too much detail in zones. Conversely, Flexcom will capture small geometric changes in the model to provide the most accurate model possible. The can prove to be problematic in areas such as tapered sections or detailed casing models where small geometric changes may lead to infinitesimal zone size generation. Zones allows you to have more control over the zone which is generated.

Section Coefficients

- Element Set. The element set which the coefficients should correspond to.

- Added Mass Coefficient. The normal added mass coefficient for the element set. If this value is left blank these values will default to the added mass coefficients specified in the static file.

- Damping Coefficient 0. The Reynolds still water damping coefficient assigned to the element set. This is only used in SHEAR7 4.8 onwards.

- Damping Coefficient 1. The A/D still water damping coefficient assigned to the element set.

- Damping Coefficient 2. The low relative velocity region damping coefficient assigned to the element set.

- Damping Coefficient 3. The high relative velocity region damping coefficient assigned to the element set.

- Damping Coefficient 4. The axial flow region damping coefficient assigned to the element set. This is only used in SHEAR7 4.8 onwards.

Section Parameters

- Element Set. The element set which the parameters should correspond to.

- Reduced Velocity Bandwidth. The reduced velocity bandwidth for the set. This parameter is the width of the band expressed as a fraction of the critical relative velocity that will be used to define which possible modes could lock-in. This value will likely change depending on whether the riser is bare, straked, or faired.

- Strouhal Number. The strouhal number for this set. The strouhal number is a dimensionless parameters that describes the relationship between vortex shedding frequency, characteristic length and the flow velocity. This value will likely change depending on whether the riser is bare, straked, or faired.

- CL Reduction Factor. The lift coefficient reduction factor for the set. The life coefficient (CL) reduction factor allows you to modify the lift coefficient iteration scheme. The lift coefficient reduction factor is multiplied by the CL value in the CL tables.

- Zone CL Type. The lift coefficient table for the set. The lift coefficient (CL) table specifies which CL table from the common.CL table will be used. This value will likely change depending on whether the riser is bare, straked, or faired.

Output Files

- Animation Flag. The option to output a MATLAB animation.

- Debugging Flag. The option to output additional debug information.

- Damages Flag. The option to output Rayleigh fatigue damage rate per year for all resonant modes at each node.

- Fatigue Flag. The option to output the total response amplitude and phase angles at every node for every resonant frequency.

- RMS Stress Flag. The option to output the RMS stress for each participating mode. This option is not available for SHEAR7 4.6.

- Additional Output Flags. The option to output additional information. There is no option to suppress the basic output which contains input echoes and results summaries. Additional output files include lift coefficients for each mode and resonant modes for all nodes of the structure.

Power Ratio Exponent

- Power Ratio Exponent Type. The option for determining time sharing probabilities when multiple modes are predicted to be excited.

Cutoff Modes

- Power Cutoff. The power cut-off level controls how many modal power-in regions there are. The lower the number, the more possible excited modes are included. The closer the number is to one, the higher the likelihood of a single frequency lock-in.

- Primary Zone Amplitude Limit. The response definition is the resolution by which the solution is summarized in the SHEAR7 output file. This output option has no bearing on the results.

Damping Ratio

- Structural Damping Ratio. The structural damping ratio in terms of critical damping.

Response

- Response Definition. The response definition is the resolution by which the solution is summarised in the SHEAR7 output file. This output option has no bearing on the results.

Higher Harmonics

- Amplification Factor. Higher Harmonics amplification factor.
- Threshold. Higher Harmonics threshold, above which a Higher Harmonics amplification factor gets applied.

Reference Diameter

- Reference Diameter. Reference diameter for A_f* calculation. The diameter provided is used in calculating the non-dimensional parameters A_f* and c_f*.
- A_f* is defined as the flexible cylinder equivalent dimensionless response amplitude. The goal of A_f* is to provide a single number that characterizes the response for the entire flexible cylinder by using the average of the RMS response in the power-in region.
- c_f* is defined as the flexible cylinder equivalent damping parameter and improves on the previous c* parameter by capturing the role of damping in the determination of the VIV of flexible cylinders in sheared flow.
- If the diameter is specified appropriately, the product of A_f* and c_f* represents the lift coefficient of the riser.

Non-orthogonal Damping

- Flag to account for non-orthogonal damping when calculating the VIV response. Using this flag activates a significant improvement to the modelling of spatially varying damping, resulting in a more accurate response prediction, particularly in regions outside of the excitation zone. When using non-orthogonal damping the beta control number should be set to a non-zero value.

RELEVANT KEYWORDS

- *CURRENT is used to specify the current to be considered in the SHEAR7 analysis.
- *CUTTOFF MODES is used to define the SHEAR7 parameters “Power Cutoff” and “Primary Zone Amplitude Limit”. These parameters are applicable to SHEAR7 4.5 and later. The SHEAR7 User Manual provides a detailed description of these parameters.
- *DAMPING RATIO is used to define the SHEAR7 structural damping ratio.
- *FATIGUE DATA is used to assign fatigue data to SHEAR7 sets.
- *FATIGUE OPTIONS is used to specify fatigue calculation options for SHEAR7 in Block 5.
- *FOLDER OPTIONS is used to indicate where the SHEAR7 input files will be generated and run.
- *NAME is used to specify a title for SHEAR7 analysis run.
- *OUTPUT FILES is used to request output file types from SHEAR7. The available output file extensions are *.anm, *.scr, *.dmg, *.fat, *.str, *.out, *.out1, and *.out2. Refer to the SHEAR7 User Manual for further information on the output file generation options. The output file options can vary between versions of SHEAR7.
- *POWER RATIO EXPONENT is used to define the type of the Power Ratio Exponent for SHEAR7.
- *RESPONSE is used to define the SHEAR7 response definition/resolution to be used in the output file.
- *RESTART is used to indicate which modal analysis should be used in order to generate the SHEAR7 model.
- *SECTION COEFFICIENTS is used to assign SHEAR7 coefficients to element sets.
- *SECTION PARAMETERS is used to assign SHEAR7 parameters to element sets.
- *S-N CURVE is used to define S-N curves to be used in SHEAR7 analysis.
- *VERSION is used to provide Flexcom with basic SHEAR7 interface information.

* *ZONES is used to define element sets that will be used as one zone in SHEAR7.

Modes Operation

EIGENSOLVER

What actually happens when you request data for input to SHEAR7 is the following. The actual eigensolution proceeds in exactly the same way as when the SHEAR7 interface is not invoked – the generation of SHEAR7 input is purely a postprocessing operation. Once the eigensolution is completed, Modes highlights the mode shapes of interest for SHEAR7 analysis as pure Bending modes, and in the case of SCRs, these modes are further categorised into In-Plane Bending and Out of Plane Bending modes. Although not relevant to SHEAR7, the remaining modes are classified as pure Axial modes, pure Torsional modes, and Unknown (typically mixed) modes.

This is an important step in the process. The reasoning is this. Modes is naturally a full 3D eigensolver, and all possible types of modes including bending, axial and torsional motions, and combinations of these, can be predicted. However of these, only pure bending modes are of consequence for the SHEAR7 calculations, so Modes does what it can to identify these. The first step in this process (for TTR and SCR types only) is to immediately eliminate any mode for which the maximum displacement occurs in either DOF 1 (likely to be a pure axial mode) or DOF 4 (likely to be a pure torsional mode). Any such modes are classified as Axial and Torsional respectively. Thereafter the program applies different criteria depending on the riser classification.

BENDING MODES IN A TTR

For a TTR, pure bending modes are assumed to occur in identical or nearly identical pairs. So Modes searches for these pairs and categorises one of them as Bending and the other as Unknown. Any singly-occurring mode is likewise deemed Unknown and excluded from the SHEAR7 data. The application of this rule assumes your TTR model is symmetric (or almost symmetric) with respect to horizontal displacements; for unambiguous results you should take care to ensure this is the case in setting up your model.

Modes outputs the SHEAR7 data for a TTR to a single file entitled common.mds – this is the name SHEAR7 requires you to use for this input. The common.mds data is in exactly the required format for SHEAR7, and comprises details of modal distributions of displacement and curvature for each mode in the data. A “header block” at the top of the file lists the actual natural frequencies. For further details, refer to MDS File Format.

BENDING MODES IN AN SCR

For an SCR, the division of Bending modes into In-Plane and Out of Plane modes is a relatively straightforward operation. For In-Plane modes, displacements occur only in the plane of the SCR. Note that this plane can be oblique to the global co-ordinate axes. There is no requirement to set up your model in either of the XY or XZ planes, although this would be usual. Likewise Out of Plane modes have displacements normal to the initial plane of the SCR only. Any mode combining motions in all three directions is deemed Unknown.

Modes routes SCR SHEAR7 data to two files, named common.inp (for in-plane modes) and common.out (for the out of plane modes). The reason is that the two sets of modes will be excited by different current distributions and so will typically be considered in separate SHEAR7 runs. When doing your SHEAR7 analysis, you will need to rename one or other common.mds as appropriate. You can also indicated to Modes in advance whether you are interested in in-plane or out of plane modes, in which case the relevant common file will be renamed automatically for you.

USER RISER TYPE

Earlier versions of the software did not possess the User riser type option. This capability was added subsequently to cater for models for which the SHEAR7 interface was not producing meaningful output (for example, 3D spool pieces). Specifically, for the TTR and SCR options, the modal displacements relate to a riser plane, which is computed internally as a vertical plane which includes the start and end points of the structure. Obviously this is valid for standard models which lie in a single plane, but is not very useful arbitrary structures which do not lie in any particular plane. If you invoke the User option, both global and local displacements are presented as part of the standard program output. This allows for extraction of the normalised local element displacements and curvatures for a given mode shape which can then be translated for use in a subsequent SHEAR7 analysis. Note that Modes does not make any assumptions about what constitutes a bending mode for a user-defined riser (the “rules” mentioned for TTR and SCR risers are irrelevant in this context), so the onus is on you to interpret the mode shapes manually, and to explicitly specify which modes you wish to include in the SHEAR7 output. Any modes which you decide to include are classified as Bending, and any ones you exclude are deemed Unknown.

Modes outputs the SHEAR7 data for a User riser to a single file entitled common.mds – this is the name SHEAR7 requires you to use for this input. The common.mds data is in exactly the required format for SHEAR7, and comprises details of modal distributions of displacement and curvature for each mode in the data. A “header block” at the top of the file lists the actual natural frequencies.

RELEVANT KEYWORDS

- *RISER TYPE ($MODES section) is used to specify the riser type and other parameters relating to the Modes facility for generating output for subsequent input to SHEAR7.

Identifying Mixed Modes

THEORY

The “rules” described in the last section enable Modes to exclude the majority of non-bending modes from SHEAR7 output. However it is difficult to ensure categorically that all modes routed to the common files are bending modes. This is particularly true when you request a large number of solution eigenpairs, and specify that most or all should be considered for SHEAR7 output, which is often the case as VIV typically excites the higher modes of vibration. The modes which Modes cannot easily identify are called mixed modes, because they are combinations of, typically, bending and axial or bending and torsion. For example, in-plane SCR modes may be pure bending or may combine bending and axial displacements; while out of plane modes may combine displacements and rotations due to torque.

Modes does produce output to help you decide if SHEAR7 output includes mixed modes. This is based on the fact that a plot of mode number v maximum modal curvature will be monotonically increasing if only bending modes are considered, but that mixed modes will cause local maxima or “spikes” in such a plot. So when you request SHEAR7 output, Modes automatically produces one (TTR and User options) or two (SCR option) plot files graphing maximum curvature as a function of mode number.

It is generally advisable to also visually inspect the modal response manually. The procedure is relatively straightforward – you simply open two views of the structure simultaneously in the Model View. One should be a front elevation (i.e. a view normal to the plane of the SCR) and the other an end elevation (i.e. a view parallel to the plane of the SCR). Pure in-plane bending modes will be visible in the front elevation view, and pure out of plane bending modes will be visible in the end elevation view. Additionally, if there are any combined modes present, these will contain some combination of bending, axial and torsional deformation. In a pure bending mode, all of the nodes of the mode (as opposed to the nodes of the finite element model) will be stationary, so a standing wave effect will be exhibited. In a coupled mode, there will be modal nodes which are not stationary, as there will be an axial or torsional response component present also.

If on examining the modal results you conclude your SHEAR7 data does include mixed modes, then an obvious question is what to do about them. One option of course would be to manually edit the SHEAR7 file or files to remove them. A better solution though is to use the options provided by Modes for excluding specific modes from SHEAR7 output. In the case of a TTR, you simply list the modes you want to exclude. For the case of an SCR, you nominate in-plane and out of plane modes to exclude separately.

Modes includes two further options in relation to manipulating SHEAR7 output. The first of these is an option to “include” modes that Modes would otherwise have excluded (in the case of an SCR, you nominate in-plane and out of plane modes to include separately). This facility is provided because very occasionally the rules described above which Modes applies to identify mixed modes lead it to excluding a genuine bending mode or modes. The second is an option to “replace” modes. Normally when a mode is excluded from SHEAR7 output, both the natural frequency and the modal displacements and curvatures are eliminated. However you have the option of replacing rather than excluding a mode; the natural frequency of a replaced mode is retained in the SHEAR7 output, but its mode shape and curvature distribution are replaced by those of another mode you specify.

RELEVANT KEYWORDS

- `*EXCLUDE MODES` (`$MODES` section) is used to specify modes to be excluded from SHEAR7 output.
- `*INCLUDE MODES` (`$MODES` section) is used to specify modes to be included in SHEAR7 output.
- `*REPLACE MODES` (`$MODES` section) is used to specify modes to be replaced in SHEAR7 output.

Additional Output and Repeat Runs

THEORY

Modes produces one further category of output to help assess SHEAR7 data. This is output at the end of the main output file, and consists of a table of the maximum curvature in each mode of the eigensolution. Two values are output for each mode. One is calculated by looping over each integration point on each element, and the other by looping over the end of each segment of equal length – recall these segments are the basis for the actual SHEAR7 output. The intention here is to allow you to decide if you have specified enough segments to accurately capture the actual element-by-element modal curvature distribution.

If an examination of this output leads you to conclude you actually need more segments, then one option is obviously to rerun your Modes analysis in full but with a larger number of segments specified. Likewise if the plots described in the last section show your SHEAR7 output included mixed modes, once again you may want to rerun your modal analysis to exclude these modes. Now a modal analysis in which you request a very large number of modes (say, 150 or more) can take some time to complete. And anyway the calculation of SHEAR7 output, as noted earlier, is purely a postprocessing operation, so repeating the eigensolution in full could be avoided. This is the rationale for the so-called Repeat Run capability.

This is how it works. Modes automatically creates a “restart” file at the end of each program run, which contains sufficient information for repeat run analyses. If you nominate that an analysis as a repeat run, Modes reads the eigensolution details from the restart file, and proceeds directly to generating the SHEAR7 data. In addition to changing the number of riser segments or excluding specific modes, you can also change the element set for SHEAR7 output, or the lowest number of modes that Modes can consider for this output. You can also use this option to change the contents of the program output file. Note though that only parameters relating to SHEAR7 file generation can be altered in a repeat run. You cannot for example change the required number of eigenpairs, and indeed Modes will terminate with an error if you attempt to do so. The program also terminates with an error in a repeat run if a relevant restart file does not exist.

RELEVANT KEYWORDS

- `*REPEAT` (`$MODES` section) is used to specify that the present Modes analysis is a Repeat Run.
- `*EXCLUDE MODES` (`$MODES` section) is used to specify modes to be excluded from SHEAR7 output.
- `*INCLUDE MODES` (`$MODES` section) is used to specify modes to be included in SHEAR7 output.
- `*REPLACE MODES` (`$MODES` section) is used to specify modes to be replaced in SHEAR7 output.
- `*OUTPUT OPTIONS` (`$MODES` section) is used to specify the eigensolution details to be routed to the output file.

- `*RISER TYPE` (`$MODES` section) is used to specify the riser type and other parameters relating to the Modes facility for generating output for subsequent input to SHEAR7.

SHEAR7 Interfacing Operation

OVERVIEW

The main purpose of this module is to quickly and efficiently run a full SHEAR7 analysis with very little effort from the Flexcom user. This section outlines the theory behind the modal operation for SHEAR7 input, current transformation, input file creation, and finally running a SHEAR7 analysis.

$MODES OPERATION

What actually happens when you request data for input to SHEAR7 is the following. The actual eigensolution proceeds in exactly the same way as when the SHEAR7 interface is not invoked – the generation of SHEAR7 input is purely a postprocessing operation. Once the eigensolution is completed, Modes highlights the mode shapes of interest for SHEAR7 analysis as pure Bending modes, and in the case of SCRs, these modes are further categorised into In-Plane Bending and Out of Plane Bending modes. Although not relevant to SHEAR7, the remaining modes are classified as pure Axial modes, pure Torsional modes, and Unknown (typically mixed) modes.

This is an important step in the process. The reasoning is this. Modes is naturally a full 3D eigensolver, and all possible types of modes including bending, axial and torsional motions, and combinations of these, can be predicted. However of these, only pure bending modes are of consequence for the SHEAR7 calculations, so Modes does what it can to identify these. The first step in this process (for TTR and SCR types only) is to immediately eliminate any mode for which the maximum displacement occurs in either DOF 1 (likely to be a pure axial mode) or DOF 4 (likely to be a pure torsional mode). Any such modes are classified as Axial and Torsional respectively. Thereafter the program applies different criteria depending on the riser classification.

For a TTR, pure bending modes are assumed to occur in identical or nearly identical pairs. So Modes searches for these pairs and categorises one of them as Bending and the other as Unknown. Any singly-occurring mode is likewise deemed Unknown and excluded from the SHEAR7 data. The application of this rule assumes your TTR model is symmetric (or almost symmetric) with respect to horizontal displacements; for unambiguous results you should take care to ensure this is the case in setting up your model.

Modes outputs the SHEAR7 data for a TTR to a single file entitled common.mds – this is the name SHEAR7 requires you to use for this input. The common.mds data is in exactly the required format for SHEAR7, and comprises details of modal distributions of displacement and curvature for each mode in the data. A “header block” at the top of the file lists the actual natural frequencies. For further details, refer to MDS File Format.

For an SCR, the division of Bending modes into In-Plane and Out of Plane modes is a relatively straightforward operation. For In-Plane modes, displacements occur only in the plane of the SCR. Note that this plane can be oblique to the global co-ordinate axes. There is no requirement to set up your model in either of the XY or XZ planes, although this would be usual. Likewise Out of Plane modes have displacements normal to the initial plane of the SCR only. Any mode combining motions in all three directions is deemed Unknown.

Modes routes SCR SHEAR7 data to two files, named common.inp (for in-plane modes) and common.out (for the out of plane modes). The reason is that the two sets of modes will be excited by different current distributions and so will typically be considered in separate SHEAR7 runs. When doing your SHEAR7 analysis, you will need to rename one or other common.mds as appropriate. You can also indicated to Modes in advance whether you are interested in in-plane or out of plane modes, in which case the relevant common file will be renamed automatically for you.

Earlier versions of the software did not possess the User riser type option. This capability was added subsequently to cater for models for which the SHEAR7 interface was not producing meaningful output (for example, 3D spool pieces). Specifically, for the TTR and SCR options, the modal displacements relate to a riser plane, which is computed internally as a vertical plane which includes the start and end points of the structure. Obviously this is valid for standard models which lie in a single plane, but is not very useful arbitrary structures which do not lie in any particular plane. If you invoke the User option, both global and local displacements are presented as part of the standard program output. This allows for extraction of the normalised local element displacements and curvatures for a given mode shape which can then be translated for use in a subsequent SHEAR7 analysis. Note that Modes does not make any assumptions about what constitutes a bending mode for a user-defined riser (the “rules” mentioned for TTR and SCR risers are irrelevant in this context), so the onus is on you to interpret the mode shapes manually, and to explicitly specify which modes you wish to include in the SHEAR7 output. Any modes which you decide to include are classified as Bending, and any ones you exclude are deemed Unknown.

Modes outputs the SHEAR7 data for a User riser to a single file entitled common.mds – this is the name SHEAR7 requires you to use for this input. The common.mds data is in exactly the required format for SHEAR7, and comprises details of modal distributions of displacement and curvature for each mode in the data. A “header block” at the top of the file lists the actual natural frequencies.

$SHEAR7 OPERATION

After the modal frequencies and shapes have been satisfactorily obtained from `$MODES`, the `$SHEAR7` interfacing module allows you to convert the Flexcom model into SHEAR7 format, transform current profiles to x/L format, and run SHEAR7.

The model is constructed quickly by assuming the element set included in the *common.mds* file, or mds set, is the desired set to be converted to the SHEAR7 format. From this set, a number of properties/values are obtained from the modal analysis elements including the unit flag, drag diameters, outer diameters, inner diameters, unit masses, internal fluid masses, added masses, internal fluid mass, element lengths, number of segments, riser type and axial stiffness. Some of these values are used directly in the creation of the SHEAR7, while others are used in the determination of other SHEAR7 parameters. Young's modulus, for example, is back-calculated from the axial stiffness and area; other values such as diameters and added mass coefficients are used explicitly. All values used in the SHEAR7 analysis are presented in the Flexcom output file, SHEAR7 input file, and additionally in the SHEAR7 output file once the analysis has run. It it is recommended to check both outputs to ensure that the desired element set and properties have been used, as very occasional default values may not be desirable.

After the model has been converted, and the current transformed, SHEAR7 is employed. Flexcom will notify you if there is an issue with SHEAR7 licensing, or if you have specified an invalid path to the SHEAR7 executable. Unfortunately, due to time it takes for SHEAR7 to search for a dongle, generating *.dat files without a SHEAR7 dongle will take significantly longer. Currently, the onus is on the user to post-process SHEAR7 results from here, but this feature will likely be supplemented in a future version.

SHEAR7 VERSIONS

Supported Versions

Backwards compatibility with previous versions of SHEAR7 has been maintained. The versions supported by the `$SHEAR7` module include:

- SHEAR7 Version 4.6 : February, 2011
- SHEAR7 Version 4.7 : September, 2012
- SHEAR7 Version 4.8 : October, 2014
- SHEAR7 Version 4.9 : December, 2015
- SHEAR7 Version 4.10 : February, 2018
- SHEAR7 Version 4.11 : July, 2021

As SHEAR7 has progressed over time the format of the input options has also changed. The differences are not generally drastic, but important to note if transitioning between versions. The following will out outline the differences between the versions.

Transition between SHEAR7 4.6 and 4.7

- BLOCK 4 Changes
  - Addition of bending stress curvature flag and load factor.
- BLOCK 5 Changes
  - Addition of flag for calculating fatigue with zero-crossing method
  - Addition of flag for calculating first mode in-line fatigue damage
  - Addition of *.STR file generation flag

Transition between SHEAR7 4.7 and 4.8

- The first string of the first line must be SHEAR7 4.8.
- BLOCK 2 Changes
  - Addition of Reynolds Still Water Damping Coefficient (DampCoeff0)
  - Addition of Axial Flow Regions Damping Coefficient (DampCoeff4)
- BLOCK 5 Changes
  - Addition of Beta Control Number
  - Addition of flag for selecting riser ID or OD for fatigue calculation

Transition between SHEAR7 4.8 and 4.9

- TBC

Transition between SHEAR7 4.9 and 4.10

- TBC

Transition between SHEAR7 4.10 and 4.11

- General
    - User may now specify a unique name for the CL file (<unique-name>.s7CL) for improved quality control purposes or maintain as common.s7cL
    - Upper mode limit of 1000 removed
    - Changing all the SHEAR7 file extensions to avoid conflicts with other programs. All files now use ".s7xxx" format. e.g. .dat -> .s7dat. Applies to the following:
        - Input files: .dat, .CL, .mds
- BLOCK 5 Changes
    - Addition of flag for unique named lift table file, (0=n,1=y, 2=y(unique name))
    - Addition of flag for stick-slip hysteresis (0=n,1=y)
    - Addition of flag for generating *.curv file, (0=n,1=y)
    - Addition of flag for generating *.z eta-hyst file, (0=n,1=y)
- BLOCK 7 (New)
    - Flag for stress time history output (.s7sth), (1=y;0=n)
    - TOTAL=total time, (optional, default 3600.0)
    - SAMPLE=sample period, (optional, default 1.0)
    - NODES=output node ranges, (optional)
    - SEED=random number seed, (optional)

SHEAR7 MODEL STRUCTURE

This section explains the layout of the SHEAR7 model and the mechanism by which the SHEAR7 input file is created.

SHEAR7 input data is provided in the form of a text-based input file with a *.dat file extension. The data in this file is structured in sections of data referred to in SHEAR7 as BLOCKS. The six blocks can be summarised as follows:

1. BLOCK 1 - Unit System
2. BLOCK 2 - Structural & Hydrodynamic Data
3. BLOCK 3 - Current Data
4. BLOCK 4 - S-N and SCF Data
5. BLOCK 5 - Computation/Output Options
6. BLOCK 6 - Supplementary Data

Once provided the SHEAR7 input file (*.dat), there are 4 calculation options supported by SHEAR7. Option 2 is used exclusively by Flexcom.

1. Option 0 – This option calculates the natural frequencies and modes shapes being computed. This option does not compute the VIV response or damage rate.
2. Option 1 – This option calculates the natural frequencies and mode shapes, then additionally perform a VIV analysis.
3. Option 2 – This is the sole option employed by Flexcom's uncoupled mode. This option uses externally (Flexcom provided) structural natural frequencies and mode shapes located in a file called "common.mds". The VIV response is calculated based on these predicted mode shapes.
4. Option 3 – This option is similar to option 2, except this option allows the user to specify the name of the *.mds file.

Since $SHEAR7 employs calculation Option 2, in which the mode shapes and frequencies are already available to SHEAR7 in the "*common.mds*" file, there is some information that is not required by SHEAR7 to perform the VIV analysis. This is as follows:

1. Flag for Structural Model
2. Effective Tension at Origin
3. Inertia
4. Submerged Weight/Length

In the writing of SHEAR7 input file, $SHEAR7 will use assign nominal values for these unnecessary parameters.

1. Flag for Structural Model: “999”. Any number could be used when calculation Options 2 and 3 are employed as the modal analysis has already been performed and the *.mds file generated. However for clarity that an external file is being used, a token value of 999 is written.

2. Effective Tension at Origin: “0.100000E+01”. It is beneficial to use a non-zero value as SHEAR7 will issue an warning if the value is set to zero. A value of 1.0 should potentially prompt users to review the Flexcom documentation, in which case they will discover this value is not necessary.

3. Inertia: Calculated. Although this value is unnecessary, a value is calculated anyway.

4. Submerged Weight/Length: "0.00000E+00”.

SHEAR7 CURRENT

VIV is only considered to be caused by flow that is normal to the riser. This is not generally an issue for TTRs, because they can be seen as vertical or almost vertical, and there is no specific vertical plane flow. However, it has a significant impact for in-plane current flow which excites out-of-plane modes for SCR and SLWR configurations. The normal current used in this case is determining by the inclination of the element. Another limitation is that SHEAR7 only provides functionality whereby the user can specify the normal flow velocity not taking into account the directionality. This often poses a problem in the modelling of SLWRs as the flow is actually reversed in the hogbend. While is not completely accurate to consider the flow as being all in the same direction, this is the only way that SHEAR7 currently allows, and it is generally an acceptable solution as it is typically more conservative. If you wish to use your own methodology for the application of current, an option has been reserved for explicit definition in terms of x/L. Elevation is converted to SHEAR7 x/L format by using each nodal locations resulting in a largely detailed current profile. A minimum of two points are required to define a current profile. When the current is defined in terms of x/L, that explicit current value is used directly in the SHEAR7 input file.

DEFAULT VALUES

A number of default values are automatically assumed which allows you to quickly generate the SHEAR7 input file, with very little mandatory effort. While for most cases, the default values will be sufficient, there are a number of supporting keywords should you decide to change them. Most of the default values correspond to bare riser parameter as recommended by SHEAR7. These default values are presented in the table below:

Default Parameters and Coefficients

Property		Default Value
Added Mass		1.0
Strouhal Number		0.18
Bandwidth		0.40
CL Table		1
Bet Control Number		4
Damping Coefficients	Reynolds Number Still Water (DampCoeff0)	1.00
	A/D Still Water (DampCoeff1)	0.20
	Low VR Regions (DampCoeff2)	0.18
	High VR Regions (DampCoeff3)	0.20
	Axial Flow Regions (DampCoeff4)	0.00
Power Cutoff Ratio		0.05
Primary Zone Amplitude Limit		0.3
Lift Coefficient Reduction Factor		1.0
Power Ratio Exponent		1

Other values that are specific to the model and zones are extracted, evaluated, or calculated from the modal restart or database files. The unit flag is derived from the gravitational constant. Hydrodynamic diameter, outer diameter, inner diameter, added mass, are extracted from the modal database. Modulus of elasticity is calculated from axial stiffness, and area calculated from the outer diameter.

While it might appear that a lot of input data is required to build the model, if the default values are used, then realistically there are only a small number of keywords that need to be explicitly defined. These include `*RESTART`, `*CURRENT`, `*FATIGUE DATA` and `*S-N CURVE`.

ZONES

SHEAR7 recommends not attempting to model too much detail in zones. The zone capability in SHEAR7 should not be used to include small structural details. On the other hand, Flexcom will seek to capture even slight changes in geometry when it builds the SHEAR7 model. This may be problematic in areas like tapered joints with changing geometry may cause the generation of multiple zones. In order to allow you to best capture these geometric changes as you see fit, `*ZONES` can be used to generate sections with equivalent properties.

In the generation of these equivalent sections, the maximum drag diameter is used while a weighted average of other parameters is used. It's worth noting that when internal fluid is present, the total weight of the internal fluid will be calculated and smeared over the section. Additionally, the user should be careful as to which set is used as a zone; the set specified for zone generation is not required by Flexcom to be continuous.

MDS File Format

The MDS file contains the modal response of the structure, including the natural frequency, mode shape, mode slope and mode curvature, for each natural bending mode. The data is presented at the ends of the segments which are used by Shear7 to discretise the structure. Typically the segments used in the Shear7 simulation are of uniform length, so these locations will rarely correspond to the nodes of the Flexcom finite element model.

The MDS file is sub-divided into 3 distinct three blocks.

- Block 1 – this block consists of just one line with 2 numbers, the number of modes (Nm) presented in the MDS file, and the number of Shear7 nodes (Nn) used to describe the mode shape. The number of Shear7 nodes is always one greater than the number of Shear7 segments (Ns).
- Block 2 – this block consists of Nm lines, where Nm is the number of modes defined in Block 1. Each line has 2 numbers, the first is the mode number and the second is the corresponding natural frequency, omega (in radians/second).
- Block 3 – the rest of the file presents the mode shape, slope and curvature as a function of Shear7 node number, repeated for each modal frequency. So the block consists of Nn (number of Shear7 nodes) times Nm (number of modes) lines. Block 3 normally has 5 columns. Optionally a 6th column may be included, if the Shear7 segments are not of uniform length.
    - Column 1 – Mode number

- Column 2 – Shear7 node number
- Column 3 – Mode shape. This is the normalised displacement computed by Modes, with the maximum amplitude of Column 3 being 1.0 for a given modal frequency
- Column 4 – Mode slope, corresponding to normalised mode shape
- Column 5 – Mode curvature, corresponding to normalised mode shape
- Column 6 (optional) – Shear7 node positions, defined as a fraction of length along the structure (i.e. ranging from 0.0 to 1.0). This data is only necessary if the Shear7 segments are not of uniform length, and in this case presented for the first modal frequency only (as it is frequency independent)

A sample MDS file is shown below. Block 3 is truncated for illustrative purposes. Note that the MDS file contains no blank lines.

common.mds

```
   5        165
      1             0.2688          23.3720
      2             0.9169           6.8523
      3             2.1681           2.8980
      4             2.4446           2.5702
      5             2.4862           2.5272
   1        1  0.000000E+00   0.160575E-02   0.371656E-05
   1        2  0.403291E-01   0.157677E-02  -0.603492E-05
   1        3  0.774264E-01   0.121366E-02  -0.354650E-04
   1        4  0.102142E+00   0.917934E-03   0.691976E-05
   1        5  0.125919E+00   0.921225E-03  -0.720360E-05
   1        6  0.148624E+00   0.883391E-03  -0.272258E-05
   1        7  0.170980E+00   0.879462E-03   0.297516E-05
   1        8  0.193276E+00   0.876901E-03   0.288730E-05
   1        9  0.215456E+00   0.869246E-03   0.242865E-05
   1        10  0.237393E+00   0.861154E-03   0.279006E-05
   1        11  0.259167E+00   0.853267E-03   0.240219E-05
   1        12  0.280700E+00   0.845205E-03   0.273300E-05
   1        13  0.302063E+00   0.836607E-03   0.230361E-05
   1        14  0.323163E+00   0.827810E-03   0.267090E-05
   1        15  0.344078E+00   0.818624E-03   0.221590E-05
   1        16  0.364713E+00   0.809237E-03   0.261197E-05
   1        17  0.385152E+00   0.799334E-03   0.214267E-05
   1        18  0.405292E+00   0.789357E-03   0.257261E-05
```

Block 1: Number of modes and number of Shear7 nodes

Block 2: List of mode numbers and natural frequencies

Block 3: Mode number, Shear7 node number, mode shape, slope & curvature (repeated for each natural frequency)

1.9.4.6 Time Domain Fatigue Analysis

THEORY

Flexcom possesses an ancillary module, LifeTime, which is a time domain fatigue postprocessor and general cycle counting tool. LifeTime can operate in two ways - in the program terminology, LifeTime has two Modes of Operation, which are identified as Mode 1 and Mode 2.

Mode 1 is the more usual method of running the program, and is used to compute fatigue life estimates from the results of Flexcom simulations. The different methods used by the program to accomplish this task are detailed later in this section.

In the second mode of operation, LifeTime can be used as a general cycle counting tool. The input in this case is a random response time history or histories, typically the output from a Flexcom random sea analysis stored in Timetrace format. However you can use LifeTime to cycle count any time history data organised in a format which mimics the timetrace output format. The LifeTime Mode 2 output is simply histograms of response.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Fatigue Analysis (Mode1) describes the sequence of steps involved in performing a time domain fatigue analysis.
- Cycle Counting (Mode2) describes the sequence of steps involved in performing a general cycle counting analysis.
- Input Data describes various pertinent program inputs which merit further elaboration.
- Stress Histogram discusses the relevance of stress histograms and their associated parameters.

RELEVANT KEYWORDS

- Numerous keywords are contained in the `$LIFETIME FATIGUE` section, which corresponds to LifeTime Mode 1.
- Numerous keywords are contained in the `$LIFETIME CYCLE` section, which corresponds to LifeTime Mode 2.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser.

Fatigue Analysis (Mode 1)

FOREWORD

Fatigue analysis in Flexcom is typically based on a series of random sea dynamic analyses, representing the loading experienced by an offshore structure over the course of its lifetime. Random variations in axial force and bending moment are translated into stress cycles which are then used to estimate fatigue damage. This approach is adopted in the vast majority of cases to estimate design life of offshore structures.

An alternative modelling approach is adopted to estimate fatigue induced by vortex induced vibration (VIV), where a structure is excited at a natural frequency by incident current loading. Fatigue of the riser is normally estimated by Shear7 following a modal analysis, and this is not related to LifeTime in any way. A more specialised case of VIV induced fatigue is that of pipe-in-pipe systems, where the VIV of the outer pipe forces the inner pipe to move accordingly. Given that the displacement and stresses in the inner pipe are effectively governed by the VIV response of the outer pipe, it is not possible to estimate fatigue damage in the usual manner via Shear7. Instead Flexcom adopts a novel approach based on a solution methodology proposed by Williams & Kenny (2017). The computational procedure involves the construction of regular/periodic time histories of bending moment, derived from the results of a static analysis of a riser system deformed into a specific mode shape, which are then post-processed by LifeTime in a manner similar to a random sea fatigue simulation. Refer to VIV Induced Fatigue of Pipe-in-Pipe Systems for further details.

SIMULATION STAGES

There are three basic steps in performing a fatigue analysis with LifeTime, as outlined in the following sections.

Step 1

If you are performing a standard fatigue analysis based on random sea loading experienced by an offshore structure over the course of its lifetime:

- Perform a Flexcom random sea analysis for each fatigue seastate. In setting up each analysis, you must store time histories of axial force, Y bending moment and Z bending moment at each location (hot spot) of interest. Earlier versions of Flexcom required these parameters be stored via timetrace output, whereas more recent versions also allow you to store this data using the more widely used database output. You can also optionally store any other time history of interest.

If you are performing a specialised fatigue analysis estimating fatigue damage of an inner pipe in a pipe-in-pipe configuration:

- Once the fatigue simulation commences, Flexcom automatically constructs the time histories of axial force, Y bending moment and Z bending moment at each hot spot for you. In preparation, you must have performed a series of static simulations in advance, each of which deforms the structure into a specific mode shape. Refer to VIV Induced Fatigue of Pipe-in-Pipe Systems for further details.

Step 2

Specify the fatigue analysis input data. This would typically include the following information:

- The names of the Flexcom simulations and their corresponding percentage occurrences. For a standard fatigue analysis, these will be the names of the random sea dynamic analysis. For the specialised case of VIV fatigue of an inner pipe of a pipe-in-pipe configuration, these will be the names of the static simulations which represent the various mode shapes.
- Stress concentration factor
- S-N curve data
- Probability density function, whether Rayleigh or Dirlik
- Whether to use bending stresses only or combined bending and axial stresses
- Whether to include thickness effects, and if so, what threshold thickness to employ
- A list of the fatigue hot spots of interest

Step 3

- Run LifeTime. When running, the program loops over all Flexcom simulations and all of the hot spots, and calculates fatigue damage using three different methods, each of which is described in Fatigue Analysis Methods. Further details are provided in Program Outputs.

RELEVANT KEYWORDS

Step 1

- Numerous keywords are contained in the $LOAD CASE section, including data such as environmental parameters (e.g. current and waves).
    - *DATABASE is used to request the storage of database output.
    - *DATABASE CONTENT is used to customise the contents of the database output files.
    - *TIMETRACE is used to request the storage of timetrace output (this option is not relevant for *VIV Induced Fatigue of Pipe-in-Pipe Systems* as you must use static database output in this case).

Step 2

- Numerous keywords are contained in the $LIFETIME FATIGUE section, which corresponds to LifeTime Mode 1.
    - *FATIGUE DATA is used to assign fatigue data to hot spot sets.
    - *HOT SPOT SETS is used to define the fatigue analysis hot spots - these are the locations on the structure for which fatigue life estimates are required.
    - *PDF is used to specify the probability density function to be used in calculating fatigue life estimates from stress spectra (this option is not relevant for *VIV Induced Fatigue of Pipe-in-Pipe Systems* as fatigue computations are based on regular/periodic time histories).
    - *PROPERTIES is used to assign effective structural properties to hot spot sets for use in calculating stresses.
    - *SEASTATE FILES is used to specify the names of the Flexcom simulations and their corresponding percentage occurrences. For a standard fatigue analysis, these will be the names of the random sea dynamic analysis. For the specialised case of *VIV Induced Fatigue of Pipe-in-Pipe Systems*, these will be the names of the static simulations which represent the various mode shapes.
    - *S-N CURVE is used to define fatigue analysis S-N curves.
    - *SOURCE TYPE is used to indicate the type of data storage file you wish to use as input to the fatigue analysis.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser.

Fatigue Analysis Methods

THEORY

Three methods are used when you run a LifeTime Mode 1 analysis to calculate fatigue damage and fatigue life, termed, Statistics, Spectrum and Rainflow, respectively.

- Statistics. Calculating the standard deviation (σ) and mean zero up-crossing period (T_z) of combined axial and bending stress directly from Flexcom time histories, and then calculating fatigue damage from these based on an assumption that stress peaks are distributed according to the Rayleigh probability density function (pdf).
- Spectrum. Calculating the spectrum of combined stress from the Flexcom time histories, and then evaluating the moments of this spectrum. These are then used to compute fatigue based on an assumption that stress peaks are distributed according to either the Rayleigh or Dirlik's probability density function.
- Rainflow. Calculating damage directly from the Flexcom stress histories using the rainflow cycle counting technique (ASTM Standard E1049-85, 1985 (2005)).

Because combined stress varies throughout the cross-section, LifeTime actually applies the above three methods at 8 points, at 45° intervals, around the hot spot outer surface. So for each hot spot you nominate, a fatigue life estimate is actually calculated for 8 points around the circumference.

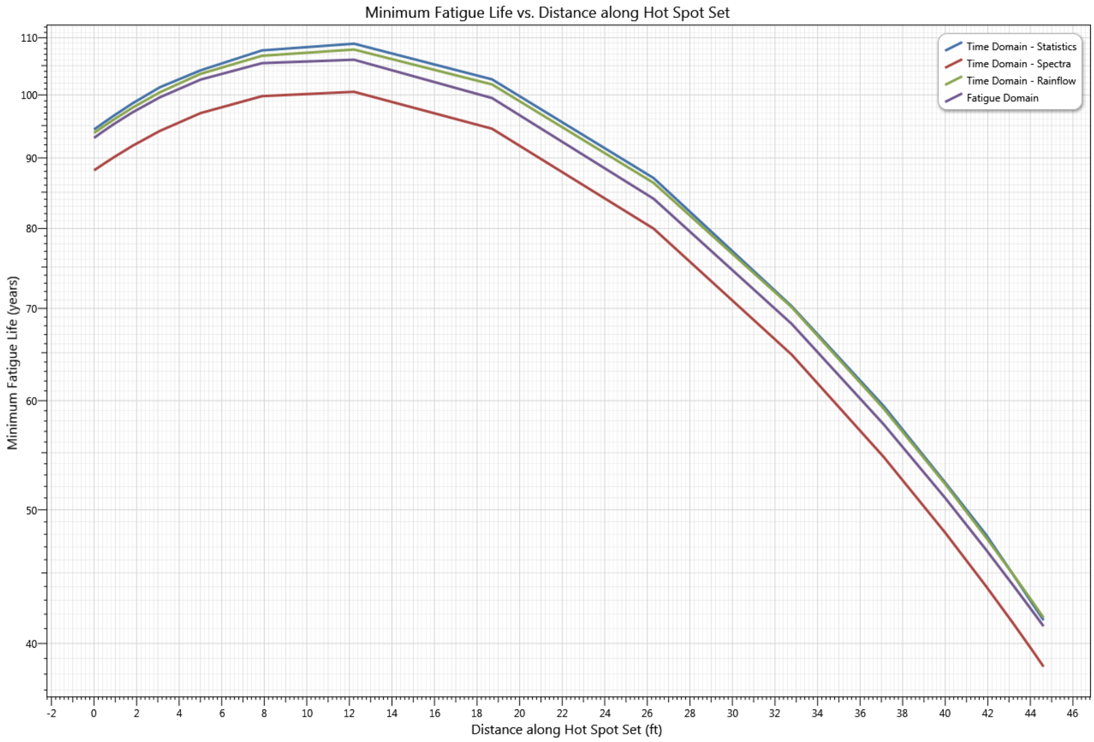
Note that of the three methods mentioned above, only the Rainflow method is relevant to VIV Induced Fatigue of Pipe-in-Pipe Systems as fatigue computations are based on regular/periodic time histories for this specialised case.

RELEVANT KEYWORDS

- `*TD OPTIONS` is used to specify a number of miscellaneous parameters, many of which relate to the calculation of stress spectra.

Program Outputs

LifeTime creates tabular and graphical output for fatigue analysis runs. The main analysis results (i.e. in terms of fatigue life, damage etc.) are presented in the main output file named jobname.out, where jobname is the name of your LifeTime analysis. This file also contains “debug” output, which presents results from the program intermediate calculations for each individual seastate (i.e. each Flexcom analysis), for checking and interpreting the main analysis results. A plot file (entitled jobname.mplt) is also created which graphically illustrates the minimum fatigue life at each hot spot from each of the three LifeTime methods.



Sample Fatigue Life Plot

VIV Induced Fatigue of Pipe-in-Pipe Systems

INTRODUCTION

Fatigue analysis in Flexcom is typically based on a series of random sea dynamic analyses, representing the loading experienced by an offshore structure over the course of its lifetime. Random variations in axial force and bending moment are translated into stress cycles which are then used to estimate fatigue damage. This approach is adopted in the vast majority of cases to estimate design life of offshore structures.

An alternative modelling approach is adopted to estimate fatigue induced by vortex induced vibration (VIV), where a structure is excited at a natural frequency by incident current loading. Fatigue of the riser is normally estimated by Shear7 following a modal analysis, and this is not related to LifeTime in any way. A more specialised case of VIV induced fatigue is that of pipe-in-pipe systems, where the VIV of the outer pipe forces the inner pipe to move accordingly. Given that the displacement and stresses in the inner pipe are effectively governed by the VIV response of the outer pipe, it is not possible to estimate fatigue damage in the usual manner via Shear7. Instead Flexcom adopts a novel approach based on a solution methodology proposed by Williams & Kenny (2017). The computational procedure involves the construction of regular/periodic time histories of bending moment, derived from the results of a static analysis of a riser system deformed into a specific mode shape, which are then post-processed by LifeTime in a manner similar to a random sea fatigue simulation. Full details are provided in the referenced paper but the main points are summarised here.

METHODOLOGY

The following methodology is used to capture the response on the inner pipe as a result of VIV excitation of the outer pipe in a pipe-in-pipe system.

Step 1

Perform a modal analysis of the system in order to determine the natural frequencies and associated mode shapes. Perform a single mode VIV analysis of the proposed riser system using Shear7 to determine the dominant mode of response and its associated frequency.

Assuming that each exciting mode acts independently, take the normalised mode outputted from Modes corresponding to the discrete mode excited in the Shear 7 analysis and scale the normalised mode by the maximum displacement amplitude predicted by Shear7 for a given current load.

The resulting deflected mode shape is applied using boundary conditions along the riser and solved statically, so that force databases are available with axial force, local y and z bending moment at riser hotspots.

Step 2

Enable SOURCE=TIMETRACE in the *SOURCE TYPE keyword and refer to the static solution database files via the *SEASTATE FILES keyword, with the *FREQUENCY=Modal frequency* defining the frequency of the exciting mode. The LifeTime fatigue analysis proceeds in the usual manner but rather than reading time histories of axial force, Y bending moment and Z bending moment at each hot spot from dynamic database files, they will be constructed from the static database files using the following method.

Time histories of Y and Z bending moment are generated from the following formula:

$$M(t)_{(y,z)} = M_{s(y,z)} \operatorname{Cos}(2\pi ft)$$

where:

- $M(t)_{(y,z)}$ is the Y or Z bending moment time history at time *t*.
- $M_{s(y,z)}$ is the local Y or Z bending moment from the static analysis.
- f is the modal frequency of the excited mode.
- t is time.

Axial force is considered to be constant over the time history.

Once all the time histories are generated by Flexcom, the LifeTime fatigue analysis will proceed in the normal manner.

For an illustration of this methodology in practice, refer to A03 - Pipe-in-Pipe Production Riser.

RELEVANT KEYWORDS

- *SEASTATE FILES is used to specify the names of the Flexcom simulations and their corresponding percentage occurrences. For a standard fatigue analysis, these will be the names of the random sea dynamic analysis. For the specialised case of *VIV Induced Fatigue of Pipe-in-Pipe Systems*, these will be the names of the static simulations which represent the various mode shapes.
- *SOURCE TYPE is used to indicate the type of data storage file you wish to use as input to the fatigue analysis.

If you would like to see an example of how these keywords are used in practice, refer to A03 - Pipe-in-Pipe Production Riser.

Cycle Counting Analysis (Mode 2)

OVERVIEW

The three steps in performing a general cycle counting analysis with LifeTime are as follows:

Step 1

Perform a Flexcom random sea analysis for each fatigue seastate. In setting up each analysis, you must store time histories of the variable(s) of interest. Unlike full fatigue analysis runs, cycle counting requires that these parameters be stored via timetrace output – database output is not sufficient. Obviously for meaningful cycle counting the same variables should be nominated in all of the Flexcom runs.

Step 2

Specify the fatigue analysis input data. This would typically include the following information:

- The names of the Flexcom random sea analyses, and tabulate the percentage occurrences of the corresponding seastates
- The variables (or “channels”) of the Flexcom timetrace output which you want cycle counted. Channel in this context refers to an individual variable in a Flexcom timetrace output file.
- The divisions (or “bins”) for the histogram output

Step 3

Run LifeTime. When running, the program loops over all seastates and all of the relevant channels, and performs the cycle counting. Further details are provided in Program Outputs.

RELEVANT KEYWORDS

Step 1

- Numerous keywords are contained in the `$LOAD CASE` section, including data such as environmental parameters (e.g. current and waves).

- o *TIMETRACE is used to request the storage of results for timetrace postprocessing (this is mainly used in the area of time domain fatigue analysis).

Step 2

- Numerous keywords are contained in the $LIFETIME CYCLE section, which corresponds to LifeTime Mode 2.
  - o *BINS is used to specify bins or divisions for histogram output.
  - o *CHANNELS is used to specify channels for cycle counting.
  - o *SEASTATE FILES is used to specify all of the Flexcom random sea analyses on which the LifeTime calculations are to be based, and to tabulate the percentage occurrences of the corresponding seastates.

Step 3

- *HISTOGRAM OUTPUT is used to specify data relating to histogram output.

Program Outputs

LifeTime creates tabular output only for cycle counting runs. The main analysis results, histograms in this case, are presented in the main output file named jobname.out, where jobname is the name of your LifeTime analysis. This file also contains “debug” output, which presents results from the program intermediate calculations for each individual seastate (i.e. each Flexcom analysis), for checking and interpreting the main analysis results.

Input Data

The following sections describe various pertinent program inputs which merit further elaboration.

- S-N Curve Data outlines the equations used for calculating S-N Curves.
- Scale Factor for Stress Computations describes how scale factors can be used to transform stress units so that they are consistent with S-N Curves.
- Threshold Thickness discusses the effect of threshold thickness on materials.

- Stress Properties describes how to apply effective geometric properties to element sets for use in calculating stresses.

S-N Curve Data

THEORY

S-N curves are generally defined in the form $NS^m = K$ where S denotes stress range, N the number of cycles to failure at this range, and m and K are constants. Taking logarithms of both sides and rearranging gives:

$$\log S = -\frac{1}{m}\log N + \frac{1}{m}\log K \qquad (1)$$

which is the equation of a straight line when log S is plotted against log N. In this case m is the inverse slope and K is a function of the line intercept. These are the parameters input above. The specification of an Endurance Limit is optional, and by default there is no endurance limit.

A piecewise-linear S-N curve is one which plots as a series of straight line segments when log S is plotted against log N. In this case m and K vary between line segments; the N1 and N2 values above specify the segment of the S-N curve X (number of cycles to failure) axis where each m and K combination apply.

It is also possible to define an arbitrary S-N curve, in terms of pairs of S and N values.

RELEVANT KEYWORDS

- `*S-N CURVE` is used to define fatigue analysis S-N curves.
- `*FATIGUE DATA` is used to assign fatigue data to hot spot sets.

Scale Factor for Stress Computations

THEORY

You may optionally specify a scale factor to be used to transform stresses in the Flexcom units to units consistent with the S-N curve data. For Imperial units, a value of 6.9444E-06 would be typical to transform lb/ft^2 to ksi. For SI units, a value of 1.E-06 would be typical to transform N/m^2 to MPa.

In some cases LifeTime is in a position to decide the appropriate value to use, in which case it is not necessary to explicitly specify a scale factor. Specifically, LifeTime retrieves the gravitational constant and uses it to determine if you used Imperial or SI units in your data. Specifically, if $9 \le g \le 10$, then LifeTime decides you employed SI units; if $32 \le g \le 33$, then LifeTime decides you used Imperial units. In either case, LifeTime automatically determines the scale factor required to transform stresses to MPa or ksi as appropriate, so you do not need to explicitly specify a scale factor – LifeTime will determine the appropriate scale.

RELEVANT KEYWORDS

- *FATIGUE DATA is used to assign fatigue data to hot spot sets.

Threshold Thickness

THEORY

The specification of a threshold thickness allows you to take account of the fact that the fatigue strength of some structural members can be dependent on material thickness, fatigue strength decreasing with increasing thickness. If you specify a threshold thickness, the stresses calculated by LifeTime are further multiplied by a factor f given by:

$$f = \left(\frac{t}{t_b}\right)^{\frac{1}{n}} \qquad (1)$$

where t_b is the specified threshold thickness; and t is the greater of t_b and the actual thickness of the particular location under consideration (this ensures that f is always greater than or equal to 1). Note that although a single value of t_b is input, f is naturally computed individually for each hot spot, since the structure thickness may vary from location to location. The exponent value *n* can also be specified and defaults to 4.

RELEVANT KEYWORDS

- `*FATIGUE DATA` is used to assign fatigue data to hot spot sets.

Stress Properties

THEORY

You have the option of defining effective geometric properties to element sets for use in calculating stresses. These geometric properties include element external and internal diameters, cross-sectional area, and moments of area about the local element axes. Naturally this option is strictly relevant to fatigue analyses only, as general cycle counting analyses do not involve any stress computations. Even for fatigue analyses, it may seem curious that you might want to specify geometric properties, given that these values have already been input to Flexcom. The reason is because Flexcom does not echo this data to timetrace output files, and LifeTime fatigue analyses may be performed solely on the basis Flexcom timetrace output. In fact, earlier versions of Flexcom required that time histories of axial force, Y bending moment and Z bending moment at each hot spot be stored via timetrace output, whereas more recent versions also allow you to store/retrieve this data using the more widely used database output. So it is conceivable that, if you are using timetrace output as the source of axial force and bending moment data, and have not requested any database output at all, you would need to specify relevant stress properties in LifeTime itself.

If the fatigue analysis is based on timetrace output, and if database output (even a minimal content is sufficient, as explained below) is also available, then LifeTime can read the required structural data from that source, thus eliminating the need for a repeat specification of this data. In this case, LifeTime reads the name of each Flexcom analysis and opens the appropriate timetrace output files for that analysis. The program also checks for the existence of a database output file from the analysis. If a database exists, the program retrieves the external and internal diameter values for each hot spot from that file, and then uses these values to calculate cross-section areas and moments of inertia. There are a number of important points to note with regard to this facility:

- Only one database needs to exist, and it can correspond to any analysis in your list.
- LifeTime reads database output once only, regardless of how many database files exist. Once it has retrieved the required structural properties from a database file, the program does not check for the existence of database output from any other referenced dynamic analysis.

- The database file need not contain very much actual results output. The inputs required by LifeTime are written by default to a header block at the start of the database file at the time of the first output to the database. So in fact database output at only one time is sufficient, and the actual amount of output can be minimised accordingly.
- If database output exists but you specify structural properties for some or all hot spot sets, the values you input in LifeTime take precedence over the values in the database.

RELEVANT KEYWORDS

There are several keywords which may be used to assign effective structural properties to element sets for use in calculating stresses.

- `*PROPERTIES` in the `$MODEL` section.
- `*PROPERTIES` in the `$DATABASE POSTPROCESSING` section.
- `*PROPERTIES` in the `$LIFETIME FATIGUE` section.

Stress Histograms

THEORY

When you run a fatigue analysis (Mode 1), the main output file jobname.out contains, by default, a histogram of stress at the hot spot with the lowest fatigue life, for the location on the cross-section (the “stress point”) at which this minimum occurs. You have the option, when defining hot spots, of requesting histograms at other locations as well. A number of data inputs relate to the format of this output, and the significance of these is now illustrated with reference to an actual LifeTime stress histogram.

The figure below shows a sample stress histogram from a LifeTime Mode 1 analysis (for a notional Element 205 and Stress Point 5). Note that stress point in this context refers to an integer value between 1 and 8, corresponding to an angle measured in degrees anti-clockwise from the local element cross-section y-axis where 1 = 0°, 2 = 45°, 3 = 90°, etc. As you can see, the histogram is in 5 columns. The first is simply the bin number. The second is the associated stress range (that is, peak to trough). Note that these are nominal longitudinal stresses, that is, not including the effect of SCF. The third column is used for output of the number of cycles in this stress range identified during rainflow cycle counting.

Columns 4 and 5 show details of fatigue life calculations based directly on the histogram data. A total damage is calculated for each bin based on a representative or equivalent stress for the bin, the user-specified SCF, and the user-specified S-N curve. By default, this equivalent stress is simply the average of the stresses enclosing or defining the bin. However you can change from this default using the Stress Histogram Range Index input. The index value you specify, which we will denote here as n, is used to calculate an equivalent stress range Seq using the following relation:

$$S_{eq} = \left[\frac{S_2^{n+1} - S_1^{n+1}}{(n+1)(S_2 - S_1)} \right]^{\frac{1}{n}} \qquad (1)$$

where S_1 and S_2 are the stresses defining the bin ($S_2 > S_1$). This equivalent stress can be considered a “power average” of the stress bin. Note that for the default n value of 1, Seq is simply ½(S_1 + S_2).

Stress Histogram for Element 205 Stress Point 5

Bin #	Stress Range (ksi)	No. of Cycles	Equivalent Range (ksi)	Damage
1	0.000- 0.010	722262	0.005	.70610E-09
2	0.010- 0.025	734964	0.018	.23979E-07
3	0.025- 0.050	695252	0.038	.19164E-06
4	0.050- 0.100	688244	0.075	.13212E-05
5	0.100- 0.200	699632	0.150	.93537E-05
6	0.200- 0.300	871474	0.250	.48702E-04
7	0.300- 0.400	814534	0.350	.11678E-03
8	0.400- 0.500	568232	0.450	.16466E-03
9	0.500- 0.600	459608	0.550	.23359E-03
10	0.600- 0.700	303826	0.650	.24651E-03
11	0.700- 0.800	281780	0.750	.34130E-03
12	0.800- 0.900	159724	0.850	.27466E-03
13	0.900- 1.000	86578	0.950	.20328E-03
14	1.000- 1.100	55480	1.050	.17239E-03
15	1.100- 1.200	31390	1.150	.12584E-03
16	1.200- 1.300	12556	1.250	.63571E-04
17	1.300- 1.400	1314	1.350	.82525E-05
18	1.400- 1.500	1168	1.450	.89605E-05
19	1.500- 1.600	0	1.550	.00000E+00
20	1.600- 1.700	0	1.650	.00000E+00
21	1.700- 1.800	0	1.750	.00000E+00
22	1.800- 1.900	0	1.850	.00000E+00
23	1.900- 2.000	0	1.950	.00000E+00
Totals		7188018		.20194E-02

Fatigue damage directly from histogram .20194E-02
Fatigue life directly from histogram 495.20 years

Associated weighted mean stress value 37.116

Sample Stress Histogram

This additional fatigue output is provided as a means of checking the adequacy of the definition of stress bins. If the definition is reasonable, the fatigue life output as calculated directly from the histogram will agree reasonably well with the value calculated by the respective method (rainflow cycle counting, spectral or statistical). Note that the histogram output also reports a “weighted mean stress”. This is simply the sum of the mean stresses at this location from each of the fatigue seastates, weighted by the percentage occurrence of that seastate.

If you do not explicitly define any histogram bins, LifeTime behaves as follows. If the program has deduced from that stresses are in either ksi or MPa, then a default bin specification with 21 bins is used. The stress ranges defining these bins are shown in the table below. If on the other hand the program is unable to determine what are the units of stress, then no stress histograms are produced, regardless of whether such histograms have been requested in the specification of hot spots.

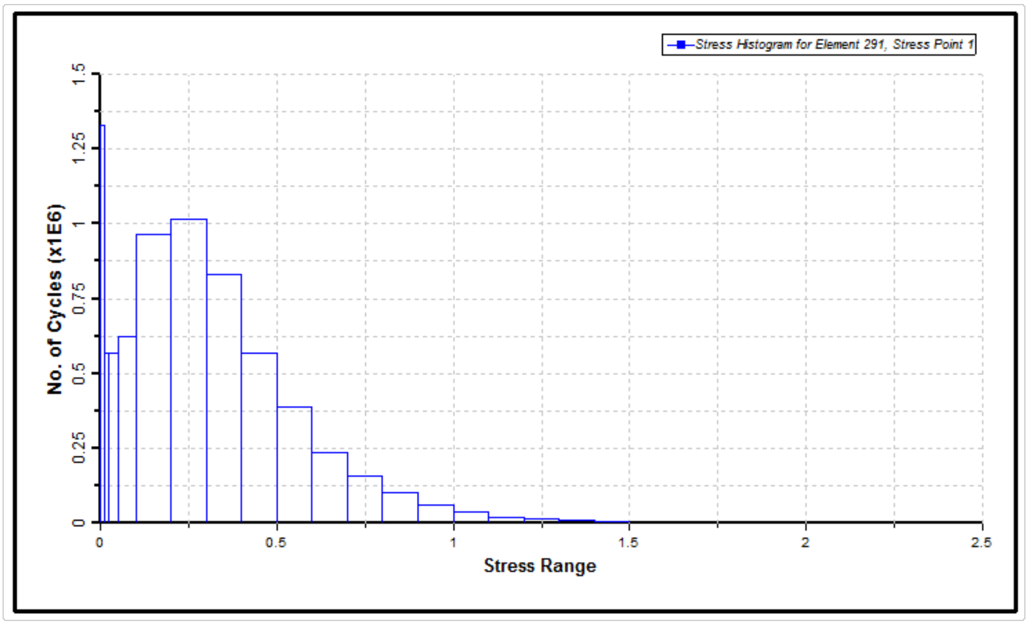
Default Stress Histogram Ranges

Bin	Stress Range	
	ksi	MPa
1	0.00-0.10	0.00-0.50
2	0.10-0.20	0.50-1.00
3	0.20-0.30	1.00-1.50
4	0.30-0.40	1.50-2.00
5	0.40-0.50	2.00-2.50
6	0.50-0.75	2.50-4.00

7	0.75-1.00	4.00-6.00
8	1.00-1.20	6.00-8.00
9	1.20-1.40	8.00-10.0
10	1.40-1.60	10.0-12.5
11	1.60-1.80	12.5-15.0
12	1.80-2.00	15.0-20.0
13	2.00-5.00	20.0-30.0
14	5.00-7.50	30.0-45.0
15	7.50-10.0	45.0-60.0
16	10.0-12.5	60.0-80.0
17	12.5-15.0	80.0-100.
18	15.0-20.0	100.-150.
19	20.0-25.0	150.-300.
20	25.0-50.0	300.-500.
21	> 50.0	> 500.

If you do specify a stress bins definition, LifeTime actually adds one bin to the end of your specification, and uses it for all stresses above the maximum value in your data. This is provided of course that the associated number of such cycles is non-zero. In the sample histogram shown in the figure below, this final bin is absent, so clearly there are no stress ranges counted above 2 ksi in this case.

Note that LifeTime now also creates a plot of each histogram you request. A sample plot is presented in the figure below. Histogram plots are arbitrarily assigned the same file extension as snapshot plots.



Sample Stress Histogram

RELEVANT KEYWORDS

- *HISTOGRAM DATA is used to specify histogram options for hot spots.
- *BINS is used to specify bins or divisions for histogram output.

1.9.4.7 Frequency Domain Fatigue Analysis

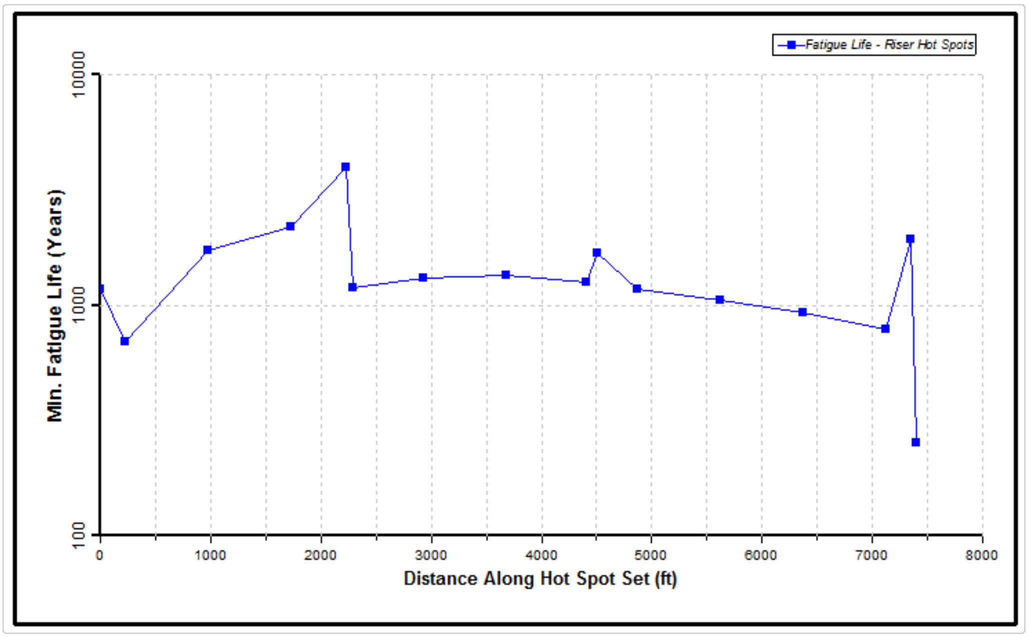
THEORY

Flexcom possesses an ancillary module, LifeFrequency, which is a frequency domain fatigue postprocessor. The LifeFrequency procedure for calculating the fatigue life at a point on a riser is based on generating a spectrum of combined bending and axial stress at that point, for each combination of wave height, wave period and wave direction in the long term environmental data for the location in question.

LifeFrequency has two modes of operation as follows, differing in how these stress spectra are produced:

- Stress Spectra Mode. In this mode, your fatigue analysis is preceded by a series of Flexcom frequency domain random sea analysis runs that you perform directly in Flexcom, to find the dynamic response for each combination of wave period, wave height and wave direction in the scatter diagram. The input to LifeFrequency is then a list of Flexcom output files, from which LifeFrequency reads in turn the stress spectra required to complete the fatigue life estimation.
- RAOs Mode. This mode is similar to the Postprocessor with Stress Spectra mode, in that your fatigue analysis is preceded by a range of Flexcom frequency domain random sea analyses. The difference is that you run analyses for selected combinations of wave height, period and direction only, and you postprocess these to generate RAOs of effective tension and bending moment (or stress). These RAOs then become the LifeFrequency inputs, and the program uses these to transform wave spectra in the scatter diagram for which you did not do a dynamic analysis, into stress spectra.

Each LifeFrequency analysis generates a plot of the minimum fatigue life at each hot spot plotted against distance along the hot spot set. The figure below shows an example of such a fatigue life plot. The distance on the X or horizontal axis is measured from the first element of each hot spot set. The Y or vertical axis actually displays the fatigue life (note the logarithmic scale). If there are two or more the user-defined hot spot sets in the fatigue run, a curve is generated for each, and LifeFrequency automatically superimposes these curves.



Sample Fatigue Life Plot

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Input Data
- Analysis Procedure
- Histogram Overview

RELEVANT KEYWORDS

- Numerous keywords are contained in the `$LIFEFREQUENCY` section, which relates to frequency domain fatigue postprocessing.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser or B01 - Steel Catenary Riser.

Input Data

The input data required for the fatigue analysis using LifeFrequency can be grouped under the following categories:

- Category (i). Structure finite element model and general environmental data (water depth, density etc.).
- Category (ii). Wave scatter diagram and long-term directionality data.
- Category (iii). A list of pre-run Flexcom analyses which form the basis of the fatigue analysis using LifeFrequency.
- Category (iv). Fatigue-specific data such as hot spot locations, stress concentration factors and material S-N curves.

Structural and Environmental Data

THEORY

All the required structural and environmental data is specified in standard Flexcom analyses. You would typically perform one or more static analyses (e.g. initial static, current static etc.) in advance of several random sea analyses, corresponding to all or selected seastates in the wave scatter diagram for each direction of wave approach. Analysing all seastates is consistent with the Stress Spectra mode of operation, while modelling selected seastates typically corresponds to the RAOs mode – the specific operation of both modes is expanded further throughout this section.

RELEVANT KEYWORDS

- Numerous keywords are contained in the `$MODEL` section, including data such as the finite element discretisation, structural and hydrodynamic properties, plus any other inputs which characterise the initial model configuration (e.g. initial vessel position, seabed properties, ocean depth etc.).
- Numerous keywords are contained in the `$LOAD CASE` section, including data such as environmental parameters (e.g. current and waves), boundary conditions of various kinds, internal fluid loading, and the analysis type and solution parameters.

Long Term Environmental Conditions

THEORY

Long-term environmental conditions at the location in question are defined in terms of (i) the wave scatter diagram, and (ii) long-term directionality data. Although these inputs are strictly only required for the RAOs mode of operation in LifeFrequency, they are central to the overall operation of the software, and even though they are not explicitly required inputs for the Stress Spectra mode, they effectively determine the choice of seastates for this mode also.

The format for inputting the scatter diagram very much reflects how the actual wave scatter diagram is usually presented. Each cell represents a particular combination of H_s and T_z, and you input into a cell the number of occurrences (typically the number of “three-hour intervals”) of that particular combination during, say, a 10 or 20-year period. You can alternatively specify the scatter diagram in terms of H_s and T_p, the wave spectrum peak period.

In principle, a fatigue analysis should involve performing a Flexcom random sea analysis for each seastate in the scatter diagram for each direction of wave approach, and then using the results to calculate fatigue damage. This is the approach adopted by the Stress Spectra mode of operation, however this might be computationally expensive. Alternatively, when you are inputting the scatter diagram in the RAOs mode of operation, as well as inputting individual numbers of occurrences, you can also divide the scatter diagram into a small number of “blocks”, and nominate a seastate (a particular combination of H_s and T_z) within each block as the representative or “reference” seastate for that block. An example of a wave scatter diagram input in this way is shown in the table below. Here the scatter diagram is divided into 12 blocks, each with a reference seastate shown highlighted. The values input and the blocking scheme are largely arbitrary.

Sample Seastate Scatter Diagram

						Tz					
		4	5	6	7	8	9	10	11	12	13

0.5	136076	252100	255434	235832			
2.0		321287	334477	298020	74558		
3.5		272100	362849	383204	148557	167188	
5.0			174127	296125	308157	275432	110168
6.5				175352	264871	260697	126533

	8.0	117157	233065	231621	110175		
Hs	9.5		111420	143196	102307	93240	
	11.0			83161	90355	80073	
	12.5				67966	52943	
	14.0				32669	36253	
	15.5					11986	20976

	17.0		13739	
	18.5		10200	
	20.0		1520	279

Long-term directionality effects are accounted for in LifeFrequency by considering storm directions from eight compass headings. For each direction, a percentage occurrence value is specified. Obviously, the sum of all occurrences must total 100%.

RELEVANT KEYWORDS

- *BLOCK is used to input the fatigue analysis wave scatter diagram, including the definition of blocks and/or reference seastates as appropriate.
- *DIRECTION is used to specify long-term directionality data.
- *SPECTRUM is used to specify the wave spectrum type to use for fatigue life calculations.

Pre-Run Analyses

THEORY

In the case of the Stress Spectra mode, you provide a list of Pre-Run Analyses, and LifeFrequency reads the stress spectra from each of these Flexcom runs in order to perform the fatigue life estimation.

In the case of the RAOs mode, you provide a list of Pre-Run Analyses, consisting of the Flexcom analyses corresponding to the scatter diagram reference seastates. There are a couple of important points to note regarding this data.

(i) You first identify a combination of H_s and T_z or T_p, which must correspond to a reference seastate in the wave scatter diagram. Then you identify an associated direction, which must correspond to one of the directions with non-zero percentage occurrence you specified as part of long-term directionality data. You also specify the name of the Flexcom analysis you ran for this combination of seastate and direction.

(ii) Note that you must specify a data combination (i.e. a combination of H_s and T_z or T_p, an associated direction, and a Flexcom analysis) for every reference seastate for every direction with non-zero percentage occurrence. So, for example, if you have five reference seastates and four directions, you must have 20 data combinations in the Pre-run Analyses specification. When you come to actually run the LifeFrequency analysis, the program does check this for you, and if there is a combination in the environmental data without a Flexcom file name, the program prints an error message to that effect and the fatigue analysis does not run.

RELEVANT KEYWORDS

Stress Spectra Mode

- `*SEASTATE FILES` is used to specify all of the Flexcom random sea analyses on which the LifeFrequency calculations are to be based, and to tabulate the percentage occurrences of the corresponding seastates.

RAOs Mode

- `*BLOCK` is used to input the fatigue analysis wave scatter diagram, including the definition of blocks and/or reference seastates as appropriate.
- `*DIRECTION` is used to specify long-term directionality data.
- `*SPECTRUM` is used to specify the wave spectrum type to use for fatigue life calculations.

Fatigue Data

THEORY

You are required to identify the fatigue analysis hot spots, these being simply the points on the structure for which fatigue life estimates are required. Hot spots are defined as belonging to hot spot sets. For each set you input (i) a stress concentration factor (SCF), (ii) an S-N curve, and (iii) optionally, a threshold thickness. You also specify the required stress type, whether bending, axial or combined bending and axial.

SCF specification is standard, and the specified SCF values multiply the stresses calculated by Flexcom/LifeFrequency to account for various stress raisers.

For S-N curve specification, a range of options is provided to allow a wide degree of generality. A particular curve may be defined by two parameters m and K such that the curve is given by $NS^m = K$, where S represents stress range and N number of cycles to failure. Such a curve plots as a straight line on log-log scales. An endurance limit (a stress range below which no fatigue damage results regardless of the number of cycles) may be optionally specified. Alternatively, a series of m and K values may define the curve over particular regions, representing a piecewise-linear log-log plot. In the most general case, a particular curve may be specified as a series of (S, N) data pairs.

The specification of a threshold thickness allows you to take account of the fact that the fatigue strength of some structural members can be dependent on material thickness, with fatigue strength decreasing with increasing thickness. If you specify a threshold thickness for a particular hot spot set, the stresses calculated by Flexcom/LifeFrequency are further multiplied by a factor f given by:

$$f = \left(\frac{t}{t_b} \right)^{\frac{1}{n}}$$

Here t_b is the threshold thickness you specify; and t is the greater of t_b and the actual thickness of the particular location under consideration (this ensures that f is always greater than or equal to 1). Note that although t_b is specified for a hot spot set, f is computed individually for each location in the set, since the structure thickness may vary within a hot spot set. The exponent value n can also be specified and defaults to 4. The specification of t_b is optional for each set; by default thickness effects are omitted.

Finally, a particular hot spot can belong to a number of hot spot sets, each with a different SCF and/or S-N curve and/or threshold thickness. In this way the effect of variations in these parameters can be evaluated in a single LifeFrequency run.

RELEVANT KEYWORDS

- `*FATIGUE DATA` is used to assign fatigue data to hot spot sets.
- `*HOT SPOT SETS` is used to define the fatigue analysis hot spots - these are the locations on the structure for which fatigue life estimates are required.
- `*PDF` is used to specify the probability density function to be used in calculating fatigue life estimates from stress spectra.
- `*PROPERTIES` is used to assign effective structural properties to hot spot sets for use in calculating stresses.
- `*S-N CURVE` is used to define fatigue analysis S-N curves.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser or B01 - Steel Catenary Riser.

Analysis Procedure

This section summarises the operation of the frequency domain fatigue analysis capability in Flexcom. It is divided into the following sections:

- Flexcom Analyses
- Comuptation of Stress Spectra
- Stress Ranges

- Fatigue Damage
- Special Cases

Flexcom Analyses

THEORY

If you intend to use LifeFrequency in the RAOs mode of operation, you must perform a Flexcom random sea analysis for every combination of reference seastate and wave direction with non-zero percentage occurrence. Additionally, you must also postprocess the Flexcom analyses to generate RAOs of axial force and bending moment at the locations of interest (i.e. the hot spots). LifeFrequency subsequently postprocesses the results to obtain stress RAOs (transfer functions). The stress RAOs for a particular reference seastate and wave direction are then used to produce stress spectra for all of the remaining seastates within that block for that direction. In this way stress spectra are produced for every combination of seastate and wave direction in the long-term environmental data. Note that the assumption implicit in this procedure is that stress RAOs are invariant with respect to seastate over short “distances” within the scatter diagram.

If you intend to use LifeFrequency in the Stress Spectra mode of operation, you must perform a Flexcom random sea analysis for every combination of all seastates and wave direction with non-zero percentage occurrence. Obviously the result of doing a Flexcom random sea analysis for every seastate for every wave direction is that you have immediately the axial force and bending moment spectra that are required for the fatigue analysis. In this context, the next section covering Computation of Stress Spectra is not relevant for the Stress Spectra mode.

After the Flexcom analyses (including postprocessing) for each reference seastate/direction combination have been completed, LifeFrequency proceeds to the fatigue life prediction process proper. The fatigue life at a particular location is found by looping over all of the seastates in the scatter diagram and all of the storm approach directions, and computing and accumulating the fatigue damage due to each combination.

RELEVANT KEYWORDS

Stress Spectra Mode

- *SEASTATE FILES is used to specify all of the Flexcom random sea analyses on which the LifeFrequency calculations are to be based, and to tabulate the percentage occurrences of the corresponding seastates.

RAOs Mode

- *BLOCK is used to input the fatigue analysis wave scatter diagram, including the definition of blocks and/or reference seastates as appropriate.
- *DIRECTION is used to specify long-term directionality data.
- *SPECTRUM is used to specify the wave spectrum type to use for fatigue life calculations.
- *RAO is used to request the creation of RAO plots.

Computation of Stress Spectra

THEORY

In the RAOs mode of operation, the first step in the fatigue analysis is to extract the required hot spot RAO or RAOs from the Flexcom output (for each combination of seastate and direction) and to transform and/or combine these as required. The appropriate RAO file will depend on what seastate is being analysed and what block this corresponds to in the scatter diagram.

If you specify that combined stresses are to be considered at a particular location, then LifeFrequency scans the Flexcom RAO output for axial force RAOs and RAOs of bending about both local axes. The axial force RAOs are transformed to axial stress and the two bending RAOs are transformed to bending stress. The three RAOs are then combined to produce combined stress RAOs. The combining of the RAOs uses complex arithmetic to include the effect of relative phasing between the stress components.

Note that LifeFrequency produces fatigue life estimates for eight points (denoted ‘stress points’) around the section circumference, and so bending stress RAOs are factored as appropriate depending on the actual ‘stress point’ under consideration. Each one of the eight points corresponds to an integer multiple of 45°, measured in degrees anti-clockwise from the local element cross-section y-axis. This methodology is consistent with the Location Parameter Input which is accepted convention in Flexcom post-processing.

Once LifeFrequency has read and evaluated the RAO data it requires, the next step in the fatigue analysis is to use this data to produce a hot spot stress spectrum according to the formula:

$$S_\sigma(f) = |H(f)|^2 \; S_\eta(f)$$

where $S_\sigma(f)$ is the output stress spectrum, $H(f)$ is the stress RAO or transfer function, $S_\eta(f)$ is the particular seastate spectrum, f denotes frequency, $|x|$ and denotes the magnitude of the complex quantity x . Note that the seastate spectrum $S_\eta(f)$ may be either Pierson-Moskowitz or Jonswap; you specify which when inputting the seastate scatter data.

The availability of this option requires that LifeFrequency incorporates an algorithm to select values for the three parameters usually used to define a Jonswap spectrum (peak frequency, γ and α) for each seastate, since these are defined only in terms of H_s and T_z or T_p. The procedure is not described in detail here – instead the interested reader is referred to Jonswap Wave.

RELEVANT KEYWORDS

- `*FATIGUE DATA` is used to assign fatigue data to hot spot sets, including an option to nominate the type of stress to be used in the fatigue calculations. You can choose between bending stresses only, axial stresses only or combined bending and axial stresses, which is the default.

Stress Ranges

THEORY

The various quantities required to complete the fatigue analysis can be evaluated from the calculated hot spot stress spectrum, or more correctly from the moments of the stress spectrum about the origin, which are defined as:

$$m_n = \int_0^{\infty} f^n S_{\sigma}(f)df \qquad (1)$$

where m_n denotes the n^{th} spectral moment and the remaining symbols are as described previously.

In order to complete the analysis using these moments, certain assumptions are made regarding the distribution of stress peaks and ranges. The first assumption concerns the probability distribution function (pdf) that can be used to determine the probability of occurrence of various stress peaks. You can choose between either the Rayleigh distribution or Dirlik's rainflow range distribution. The Rayleigh distribution is completely defined by m_0, the zeroth moment or the area under the stress spectrum curve. This distribution is suitable for stress spectra that are narrow banded. The Dirlik distribution is defined by m_0, m_1, m_2, and m_4, the zeroth, first, second, and fourth moments of the stress spectrum. This distribution is more appropriate when stress spectra are broad banded. Since either distribution refers to stress peaks and in fatigue analysis stress ranges are of interest, the further assumption must also be made that each peak magnitude is half the magnitude of the corresponding stress range. The probability of occurrence of various stress ranges in the response to a particular seastate can therefore be calculated by dividing the area under the corresponding probability distribution curve into a finite number of areas.

In the case of the RAOs mode of operation, the total number of all stress peaks (and hence stress ranges) in one year for a particular seastate i, for a particular direction j, denoted M_{ij}, can be calculated from m_0, m_2 and m_4, the zeroth, second and fourth moments of the stress response spectrum respectively, as follows:

$$M_{ij} = \frac{1\,\text{year (in s)} * (\%\,\text{occurrence of seastate}\,i) * (\%\,\text{occurrence of direction}\,j)}{T_{z\sigma}(\text{in s})} \qquad (2)$$

In the case of the Stress Spectra mode of operation, LifeFrequency is dealing directly with the stress spectra produced by these runs, so there is no need for the fatigue program to know the actual combinations of environmental conditions corresponding to each run. The only environmental data required is the percentage annual occurrence of this combination. The percentage value you input in this case is used in a slightly amended form of the Equation above, where it replaces the product "(% occurrence of seastate i) * (% occurrence of direction j)". The amended form of the equation is:

$$M_{ij} = \frac{1\,year\,(in\,s) * (\%\,occurrence\,of\,this\,combination)}{T_{z\sigma}(in\,s)} \quad (3)$$

T_{zs} is the mean stress up-crossing period, is given by:

$$T_{z\sigma} = \sqrt{\frac{m_0}{m_2}} \quad (4)$$

The probability of occurrence of a particular stress range S_k in the response to seastate i and direction j, denoted $p_{ij}(S_k)$, is evaluated by integrating the area under the distribution curve between appropriate ordinates, thus:

$$p_{ij}(S_k) = \int_{S-\Delta S}^{S+\Delta S} p(x)\,dx \quad (5)$$

where p(x) is the probability distribution, and ΔS is chosen on the basis of a suitable subdivision of the area under the curve into a finite number of areas. The Rayleigh distribution is given by:

$$p(x) = \frac{x}{m_0}\exp(-x^2/2m_0) \quad (6)$$

The Dirlik distribution is given by:

$$p(x) = \frac{\frac{D1}{Q}e^{-Z/Q} + \frac{D2Z}{R^2}e^{-Z^2/2R^2} + D_3 Ze^{-Z^2/2}}{2(m_0)^{1/2}} \quad (7)$$

where:

$$D_1 = \frac{2(x_m - \beta^2)}{1+\beta^2} \quad (8)$$

$$x_m = \frac{T_c}{T_m} = \frac{m_1}{m_0}\left(\frac{m_2}{m_4}\right)^{1/2} \quad (9)$$

$$\beta = \frac{T_c}{T_{z\sigma}} = \left(\frac{m_2{}^2}{m_0 m_4}\right)^{1/2} \quad (10)$$

$$Z = \frac{x}{2\sqrt{m_0}} \quad (11)$$

$$Q = \frac{1.25(\beta - D_3 - (D_2 R))}{D_1} \quad (12)$$

$$D_2 = \frac{(1-\beta-D_1+D_1{}^2)}{1-R} \quad (13)$$

$$D_3 = 1 - D_1 - D_2 \quad (14)$$

$$R = \frac{\beta - x_m - D_1{}^2}{1-\beta-D_1+D_1{}^2} \quad (15)$$

$$T_c = \sqrt{\frac{m_2}{m_4}} \quad (16)$$

$$T_m = \frac{m_0}{m_1} \quad (17)$$

The actual number of occurrences in one year of stress range S_k in response to seastate i, direction j, denoted n_{ij} (S_k), or simply n_{ijk}, is given by:

$$n_{ijk} = p_{ij}(S_k)M_{ij} \quad (18)$$

RELEVANT KEYWORDS

- *PDF is used to specify the probability density function to be used in calculating fatigue life estimates from stress spectra.

Fatigue Damage

THEORY

The damage due to stress range k in seastate i, direction j, as defined by the Palmgren-Miner Rule, is found by dividing the actual number of occurrences of stress range S_k, that is n_{ijk}, by the number of cycles of this stress range required to cause failure. This latter quantity is denoted $N(S_k)$ or N_k, and is found from the appropriate S-N curve. Denoting the damage due to stress range k in seastate i, direction j, as d_{ijk} we write:

$$d_{ijk} = \frac{n_{ijk}}{N_k} \qquad (1)$$

and the accumulated damage in the response to seastate i due to all stress ranges and directions, denoted di, is given by:

$$d_i = \sum_j \sum_k d_{ijk} = \sum_j \sum_k \frac{n_{ijk}}{N_k} \qquad (2)$$

The accumulated damage in one year due to all seastates, which is denoted d1, is given by:

$$d_1 = \sum_i d_i \qquad (3)$$

According to the Palmgren-Miner Rule the fatigue life at a particular hot spot is 1/d1 years. This is the procedure used to predict fatigue life in LifeFrequency.

RELEVANT KEYWORDS

- *S-N CURVE is used to define fatigue analysis S-N curves.
- *FATIGUE DATA is used to assign fatigue data to hot spot sets.

Special Cases

THEORY

There are two special cases of the above general procedure for dividing the scatter diagram into blocks and nominating a reference seastate within each block.

(i) In the case of the RAOs mode, if you have only a single block (encompassing the full scatter diagram) with a single reference seastate, you perform one random sea analysis only for each wave direction with non-zero percentage occurrence. The RAOs from these analyses are used for all of the seastates in the scatter diagram, but otherwise the fatigue analysis proceeds exactly as per the Equations found in Computation of Stress Spectra, Stress Ranges and Fatigue Damage.

(ii) You want Flexcom to do a random sea analysis for every seastate in the scatter diagram, for every wave direction with non-zero percentage occurrence. In the case of the RAOs mode, this is equivalent to making every cell of the scatter diagram into both a seastate block and the reference seastate for that block. The fatigue life calculations are slightly different in this case to the general procedure outlined above – but only slightly. The result of doing a random sea analysis for every seastate for every wave direction is that you have immediately the axial force and bending moment spectra that are required to complete the fatigue analysis. There is no requirement in this case to postprocess the random sea results to produce response RAOs. So in effect the fatigue calculations begin at Stress Ranges (Eq.1), the calculation of the moments of the combined stress spectrum. Otherwise the procedure is exactly the same as in the general case. Note that the use of the RAOs mode in this way is essentially the same as running LifeFrequency in the Stress Spectra mode.

RELEVANT KEYWORDS

- `*BLOCK` is used to input the fatigue analysis wave scatter diagram, including the definition of blocks and/or reference seastates as appropriate.

Histogram Overview

OVERVIEW

Histogram is a frequency domain postprocessor to Flexcom and is used as a general cycle counting tool. The input is taken directly from the results of Flexcom random sea analyses, and the results are output in the form of response histograms. The program can combine results from a number of analyses taking percentage occurrence of the corresponding seastates into account. The module produces tabular and graphical output.

One important point to note is that response histograms in the frequency domain are presented with respect to amplitude. This is inconsistent with the corresponding time domain cycle counting tool (LifeTime), which presents response histograms in terms of parameter range.

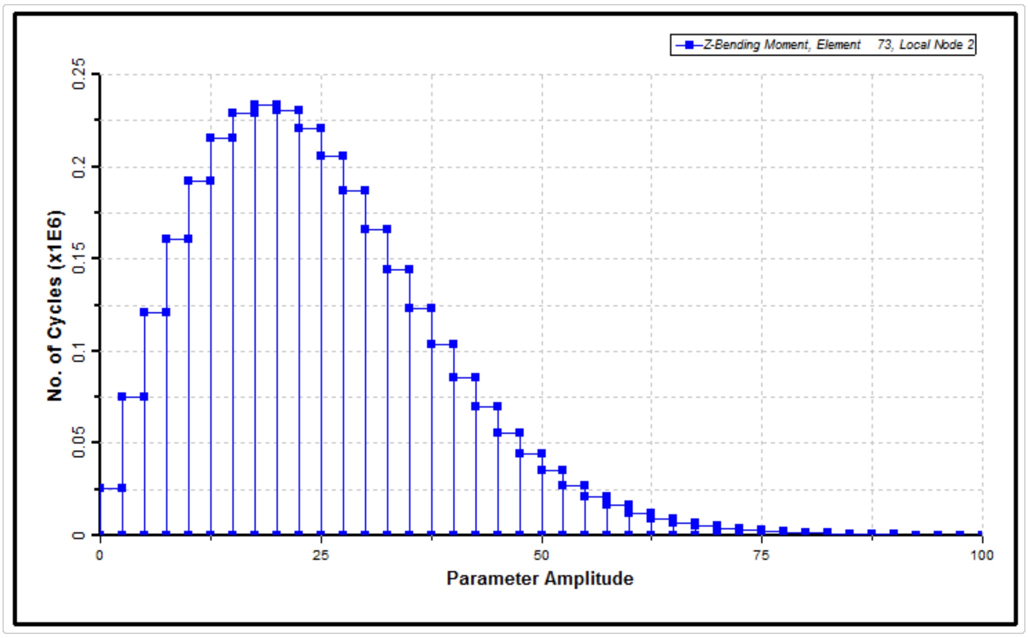
OPERATION

The steps involved in performing a general cycle counting analysis with Histogram are as follows:

- Step 1. Perform a Flexcom random sea analysis for each fatigue seastate.
- Step 2. Specify the relevant input data into Histogram. Fatigue seastate data is specified in terms of a list of Flexcom random sea analyses, along with percentage occurrences of the corresponding seastates. General data includes the parameters from the Flexcom output which you want cycle counted, plus the “bins” or divisions for the Histogram output.
- Step 3. Run Histogram. When running, the program loops over all seastates and opens the Flexcom postprocessor file for each. It reads the parameter values from these files and then calculates the response histograms for each parameter requested

The main analysis results, histograms in this case, are then output to a file named jobname.out. This file also contains “debug” output, which presents results of the program’s intermediate calculations for each individual seastate (Flexcom analysis). This “debug” output may be used for checking and interpreting the analysis results.

For each histogram you request, Histogram also creates a separate plot file. An example is shown in the figure below. As you can see, on the X or horizontal axis is Parameter Amplitude, while the Y or vertical axis has No. of Cycles.



RELEVANT KEYWORDS

Step 1

- Numerous keywords are contained in the `$LOAD CASE` section, including data such as environmental parameters (e.g. current and waves), boundary conditions of various kinds, internal fluid loading, and the analysis type and solution parameters.

Step 2

- Numerous keywords are contained in the `$HISTOGRAM` section, which relates to general cycle counting in the frequency domain.
    - `*BINS` is used to specify bins or divisions for histogram output.
    - `*PARA` is used to request response histograms of restoring force, stress, reaction, or relative rotation.
    - `*PDF` is used to specify the probability density function to be used in the calculation of histograms.
    - `*SEASTATE FILES` is used to specify all of the Flexcom random sea analyses on which the Histogram calculations are to be based, and to tabulate the percentage occurrences of the corresponding seastates.

1.9.4.8 Specialised Solution Topics

This section discusses some advanced topics in relation to finite element analysis, with regards to the following sections:

- Solution Convergence - discusses the criteria by which a solution is deemed to have converged or not.
- Bandwidth Optimisation - outlines this methodology for improved solution efficiency.
- Gaussian Quadrature - introduces the topic of numerical integration.

Solution Convergence

INTRODUCTION

This article provides some background theory regarding the evaluation of solution convergence within Flexcom.

If you experience convergence issues with a numerical simulation, you may find the Troubleshooting section helpful. It presents some general advice regarding model building, provides recommendations regarding the selection of appropriate solution parameters, and includes some practical advice on what to do if an analysis fails to complete successfully.

STANDARD CONVERGENCE MEASURE

The standard convergence measure is based on evaluating the change in solution (considering all the degrees of freedom of the hybrid beam column element) from iteration to iteration, and this is the default method for static and time domain dynamic analyses. Specifically, Flexcom determines if convergence has been achieved at each analysis time step by computing a convergence measure t as follows:

t = maximum (t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8) = maximum (t_i)

where:

$$t_i = \sqrt{\frac{\sum_{j=1}^{N}(d_{ij}^{(k)} - d_{ij}^{(k-1)})^2}{\sum_{j=1}^{N}(d_{ij}^{(k)})^2}} \qquad (1)$$

and:

- i is the global degree of freedom. i varies from 1 to 8, where values 1 to 6 correspond to the six spatial DOFs (motions and rotations), and values 7 and 8 correspond to axial force and torque, respectively
- N is the number of nodes for i = 1 to 6, and the number of elements for i = 7 and 8
- $d_{ij}^{(k)}$ is the solution variable at node or element j, degree of freedom i, iteration k (present iteration)
- $d_{ij}^{(k-1)}$ is the solution variable at node or element j, degree of freedom i, iteration $k-1$ (previous iteration)

Note that the axial force and torque terms are included in the convergence calculations because they are solution variables in the Flexcom hybrid beam element formulation.

If the computed value of t at a particular time step is less than the analysis tolerance measure, then convergence has been achieved and the solution advances to the next time step. If not, a further iteration begins. A maximum number of such iterations is set for each analysis, to prevent indefinite looping. The convergence tolerance measure and the maximum number of iterations are optional inputs, with suitable defaults provided.

SMALL TORQUE VALUE

The inclusion of torque in the convergence calculations can in some analyses increase the number of iterations required for convergence. This is particularly true of flexible riser systems subjected to waves and current approaching at an angle oblique to the initial plane of the riser. In reality, the actual torque values may be small, and the influence of torque on the solution (as opposed to the rate of achieving this solution) minimal. The *Small Torque Value* input can be used in such cases to effectively instruct Flexcom to ignore torque in determining convergence.

Specifically, when Flexcom computes t_8 it ignores those elements where the torque is below the user-specified or default Small Torque Value. If convergence is slow in a 3D riser analysis, you may wish to increase from the default value to 100, 500 or possibly 1000. It is strongly recommended that if you exercise this option for a series of analyses that you do at least one verification analysis to ensure the assumption about the relative unimportance of torque is valid.

The small torque value is not dimensionless and so the default value is typically 10 Nm when using the SI system of units and 10 ft.lb for the Imperial system.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Energy Residual Convergence
- Frequency Domain Convergence

RELEVANT KEYWORDS

- `*TOLERANCE` is used to define the analysis convergence tolerance measure and related data.

Energy Residual Convergence

THEORY

An alternative convergence criteria based on energy residuals is also provided, similar to that proposed by Bathe (1982). This approach may be useful for certain applications – for example in modelling post-buckling responses. In such circumstances, the standard convergence measure described above may not be sufficiently stringent, as it is largely displacement driven. Specifically, while the overall displacement measure may have been satisfied for an entire model at a given iteration, significant variations may still be occurring in localised areas. The energy residual convergence is checked for each degree of freedom using the following formula:

$$R = \frac{\left\| \Delta u^i \left(F^i_{ext} - F^i_{\text{int}} \right) \right\|}{\left\| u^i F^i \right\|} < \varepsilon \qquad (1)$$

where:

- Δu^i is the increment in displacement between two successive iterations, i and i-1
- u^i is the displacement at iteration i
- F^i_{ext} and F^i_{int} are the external and internal force terms, respectively, at the current iteration i
- ε is the energy tolerance
- $\|A\|$ is the Euclidian norm of A, defined as $\|A\| = \sqrt{\sum A_i^2}$

Note also that:

- $F^i = F^i_{ext}$ if the respective node does NOT have a boundary condition imposed for that DOF
- $F^i = F^i_{\text{int}}$ if the respective node has a boundary condition imposed for that DOF

If $\|u^i F^i\|$ is negligible, then $R = \left\| \Delta u^i \left(F^i_{ext} - F^i_{\text{int}} \right) \right\|$.

This additional criterion works alongside the standard convergence criterion. If you do not specify an energy residual tolerance level, the criterion is not checked. A converged solution is achieved when both criteria are satisfied for a particular iteration.

RELEVANT KEYWORDS

- `*TOLERANCE` is used to define the analysis convergence tolerance measure and related data.

Frequency Domain Convergence

THEORY

Flexcom uses a standard technique from numerical analysis, which is termed relaxation, to improve the rate of convergence of frequency domain dynamic analyses. Using this technique, the solution of the equations of motion from a particular iteration is replaced by a weighted average of the solutions for that iteration and the previous one.

For example, consider where the value of a typical displacement as predicted by the n^{th} iteration of Flexcom is denoted $\boldsymbol{X_n}$, while the corresponding term from the previous iteration is denoted $\boldsymbol{X_n}$-1. Both $\boldsymbol{X_n}$ and $\boldsymbol{X_{n-1}}$ are of course complex. The relaxation procedure may be described by the assignment statement:

$$rx_n + (1 - r)\, x_{n-1} \Rightarrow x_n$$

Here r is termed the relaxation parameter and is in the range 0 < r < 1. Note that the above is not an equation. The use of the arrow indicates that the expression on the left hand side is first evaluated, and that this then becomes (is assigned to) the value of the variable on the right hand side, replacing its existing value.

Relaxation as noted is a common numerical procedure and has the effect of improving the rate of convergence of iterative techniques, particularly in cases where results from successive iterations oscillate about the correct solution. As for the convergence parameters, the default value of 0.8 for r is adequate in the vast majority of cases. The value of r should be changed only when:

(i) a dynamic analysis repeatedly fails to converge, even though the tolerance measure and/or the maximum number of iterations have been altered (within reason)

(ii) you are certain the finite element model is indeed physically reasonable and capable of withstanding the specified environmental conditions

RELEVANT KEYWORDS

- `*TOLERANCE` is used to define the analysis convergence tolerance measure and related data.

Bandwidth Optimisation

THEORY

Finite element matrices are typically symmetric and sparsely populated. The symmetric nature of these matrices means that the entire matrix need not be stored. Computational efficiency may be increased significantly if the matrix is “banded”, meaning that non-zero terms are located close to the matrix diagonal, as the number of equations to be solved can be significantly reduced. Bandwidth optimisation essentially involves an internal renumbering of the finite element model, such that the original sparse matrix is transformed into an equivalent one which is heavily banded.

Flexcom provides an option to specify whether or not bandwidth optimisation is to be performed. This option is only relevant for non-restart analyses, for which bandwidth optimisation is never performed, as the solution profile (whether optimised by bandwidth optimisation or not) is taken from the preceding analysis. For initial (non-restart) analyses, bandwidth optimisation is performed by default, and this is recommended for the vast majority of models. Very occasionally, particularly if your model is quite complex, the actual optimisation can take a long time. If you want to quickly see results from, say, an initial static analysis, you can use the bandwidth disabling option to suppress the optimisation, but that would be an unusual move.

RELEVANT KEYWORDS

- `*NO OPTIMISE` is used to specify whether or not bandwidth optimisation is to be performed.

Gaussian Quadrature

Exact integration of expressions for element matrices can be difficult, so numerical integration (also referred to as quadrature) is an essential part of finite element analysis. Quadrature is used in numerical analysis, to approximate the definite integral of a function, usually stated as a weighted sum of function values at specified points within the domain of integration. Gaussian quadrature is constructed to yield an exact result for polynomials of degree (2n − 1). The domain of integration is [−1, 1], so the integral is written as:

$$\int_{-1}^{+1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \qquad (1)$$

where x_i is a designated evaluation point, and w_i is the weight of that point in the sum. For the purposes of finite element analysis, the calculations involved in determining the values of f, the function to be integrated, may be complex. So the Gaussian processes are ideally suited, requiring the least number of such evaluations. The table below summarises the positions and weighting coefficients for Gaussian quadrature.

Gaussian Abscissae and Weight Coefficients

Number of Integration Points	Location (x_i)	Weight (w_i)
2	±0.5773502691	1.0
3	0.0	0.888888889
	±0.7745966692	0.555555555
5	0.0	0.568888889
	±0.5384693101	0.478628670
	±0.9061798459	0.236926885
10	±0.14887434	0.29552422
	±0.43339539	0.26926672
	±0.67940957	0.21908636
	±0.86506337	0.14945135
	±0.97390653	0.06667134

1.9.4.9 Troubleshooting Simulation Failures

This section presents some general guidelines regarding the use of Flexcom, and it also gives some advice on what to do if an analysis fails to complete successfully. It is divided into the following sections:

- Main Output File discusses the usefulness of the main text based output file.
- Model Building Guidelines presents some general advice regarding model building, and includes subsections on ‘Complex Models’, ‘Finite Element Discretisation’, ‘Structural Properties’, ‘Nonlinear Properties’, ‘Model Preview Facility’ and ‘Cable Solution’.
- Solution Recommendations presents some recommendations regarding the selection of appropriate solution parameters, including ‘Convergence Ratios’, ‘Indeterminate Solutions’, ‘Converged Solutions’, ‘Time Stepping’, ‘Solution Parameters’, ‘Frequency Domain Analysis’ and ‘Quasi-Static Analysis’.
- Contact Wood if you require further assistance from the Flexcom Software Support Team.

Main Output File

INPUT DATA ECHO

The main output file generated by a Flexcom analysis (with the file extension OUT) contains an echo of all the specified input data. It is strongly recommended (particularly for an initial static analysis) that this file be used to check the accuracy of the input data, namely nodal co-ordinates, element connectivity, properties, loads, boundary conditions and time-step parameters. This should highlight any elementary errors in the input data specification before you embark on a series of dynamic analysis for a substantial load case matrix.

CONVERGENCE RATIOS

If an analysis fails to converge successfully, the program outputs the convergence ratio for each unsuccessful iteration – this is presented at the end of the input data echo section. There are typically 15 iterations (depending on the analysis type, and the time-stepping procedure if applicable), and the table will show whether or not the analysis was approaching the convergence tolerance. It will also allow you to identify the critical degrees of freedom. Refer to Convergence Ratios for further information.

ADDITIONAL OUTPUT

It is also possible to augment the contents of the main output file with greater details in specific areas of interest. This affords greater transparency regarding some of the internal computations which are performed by Flexcom at run-time. For example, it is possible to request further information relating to...

- Instantaneous Convected Element Axes in the Finite Element Solution
- Solution Criteria Automation in Static Analysis
- Wave Spectrum Discretisation in Time Domain Analysis
- Internal Fluid Centrifugal Forces & Slug Flow

... any many more parameters besides.

RELEVANT KEYWORDS

- `*PRINT` is used to request additional printed output to the main output file.

Model Building Guidelines

Because the range of structures that can be analysed by Flexcom is so extensive, it is difficult to make any hard and fast rules about how you should approach a given problem. Nevertheless, certain general guidelines can be presented, and these are described in the following sections:

- Complex Model
- Finite Element Discretisation
- Structural Properties
- Nonlinear Properties
- Model Preview Facility
- Cable Solution

Complex Model

When creating a model of a particular structure for the first time, you should resist the temptation to being with a model which incorporates full complexity. Better practice would be to create a relatively simplistic model initially, and to gradually introduce complexity once you are entirely satisfied with the basic model. It is generally advisable to examine some key outputs from the initial static analysis first – it may be possible to verify these using some simple hand calculations. For example, the maximum effective tension at the top of a riser should be roughly consistent with the (wet) weight of riser suspended below it. As you examine various load cases (e.g. vessel offset, current etc.), make sure you understand the structural response and you are satisfied it makes sense intuitively.

Finite Element Discretisation

THEORY

Ideally, the aim is to create a mesh which is sufficiently dense to accurately capture structural behaviour, while not being unduly complex and resulting in lengthy simulation times. For example, if defining a steel catenary riser, you would typically require a relatively fine mesh in the touchdown region, while a more coarse mesh would suffice for the portion of riser lying flat on the seabed, or the portion extending upwards through the water column.

Intersections between different sections within the model, model boundaries (e.g. vessel connection) and the wave zone are generally regions of interest also and therefore should have a more refined mesh. In order to avoid over-meshing, you should use longer elements in the middle of long sections of continuous properties. You should also attempt to prevent large changes in relative element length across the finite element mesh by gradually stepping up and down element lengths along the structure. A suggested maximum aspect ratio between the lengths of adjacent elements would be approximately 1.5. These guidelines are particularly important if you are building a model manually using Nodes and Elements and Cables, but this approach has been largely superseded by the new Lines modelling feature. Since lines provide automatic mesh generation, meshing at intersections and boundaries, and meshing aspect ratios, are automatically adhered to when you use lines.

If you are modelling contact, it is important that the finite element mesh in sections of the model where contact is likely to occur is sufficiently fine to properly model contact. For example, in order to properly model contact between a riser and a guide surface, it is desirable that at least two nodes of the riser are in contact with the guide surface when contact occurs.

If your model is likely to experience significant compressive loads, you will need to use a relatively fine mesh in order to ensure that the critical Euler buckling load is not exceeded - refer to Compression and Buckling for further details.

It is generally recommended that you should endeavour to match the spatial and temporal discretisations. Specifically, if you are running a time domain dynamic analysis, the select time step (or range of steps in the case of a variable step analysis) should be broadly consistent with the finite element mesh density. In order words, if you have a relatively refined model, then you should be using relatively fine time steps. Conversely, if you have a relatively coarse model, then larger time steps will be more suitable.

Avoid the specification of zero length beam elements. The presence of any such elements will immediately cause a static indeterminacy, and the source of the indeterminacy may be difficult for you diagnose.

RELEVANT KEYWORDS

- `*LINES` is used to define a line (and sections within that line), by specifying relevant set names, lengths, start and end locations, and mesh generation settings.
- `*GUIDE` is used to define guide (contact) surfaces.
- `*TIME` is used to define time parameters for an analysis.

Structural Properties

THEORY

Caution is recommended when specifying properties for rigid or very flexible structures. Experience suggests that using elements of arbitrarily large or small stiffness values is not good practice for finite element analysis, and can lead to solution instability in certain circumstances. For example, if you are modelling a perfectly rigid structure, then it would be advisable to specify a bending stiffness of perhaps 1.0E+10Nm^2, rather than an excessively large value of perhaps 1.0E+15Nm^2. Conversely, if you are modelling a chain which has effective zero structural bending stiffness in reality, then it might be advisable to specify a relatively low (but non-zero) bending stiffness, such as 10Nm^2. For example, if you were to specify as zero bending stiffness for a chain, a solution indeterminacy may occur at some point, if the effective tension distribution in the chain does not provide a sufficient geometric contribution to the overall stiffness.

RELEVANT KEYWORDS

- `*GEOMETRIC SETS` is used to assign geometric properties to element sets.

Nonlinear Properties

THEORY

It is important to remember that Flexcom adopts a tangent approach for modelling most non-linear relationships. One exception is non-linear soil structure interaction, where the P-y relationships are modelled using a secant stiffness approach. If your model includes any non-linear properties (e.g. non-linear materials, non-linear flex joints, non-linear spring elements etc.), the following simple guidelines should generally be adhered to:

- Avoid abrupt changes in slope. The non-linear characteristic relationship should appear as a smooth curve, without any sharp changes or “corners”.
- Increasing the number of data points may help to provide smooth curves.
- If do wish to model an instantaneous change in slope in reality (for example, to simulate contact between an inner and outer pipe in a pipe-in-pipe configuration), then it may be advisable to gradually increase the slope over a finite range of displacement. This approach should ensure a reasonable compromise between solution accuracy and robustness.
- The resistance provided by a non-linear curve should either monotonically increase or monotonically decrease with increasing displacement. If this is not the case, there may be more than one location along the curve which results in the same restoring force, and this may contribute to solution instability.

RELEVANT KEYWORDS

- `*MOMENT-CURVATURE` is used to define moment-curvature curves for non-linear materials (Flexible Riser format).
- `*FORCE-STRAIN` is used to define force-strain curves for non-linear materials (Flexible Riser format).
- `*TORQUE-TWIST` is used to define torque-twist curves for non-linear materials (Flexible Riser format).

  - *STRESS/STRAIN DIRECT is used to define stress-strain curves for non-linear materials (Rigid Riser format).

- *PIP STIFFNESS is used to define force-deflection curves for non-linear pipe-in-pipe connection stiffnesses. Specifically, the TYPE=POWER LAW input is used to create a non-linear contact relationship based on a power law approach.

Model Preview Facility

The latest version of Flexcom can be truly considered as an integrated engineering environment, with all the necessary tools available “in one box”. One tool which is certainly useful is the Model View – it provides a “live” structure preview facility during model building. It is continually updated based on the information you specify. For example, if you add a new node/element to the model, then it will automatically appear in the structure preview facility. The preview allows you to visually inspect all the nodal coordinates in your model, and the finite element connectivity. Any fundamental problems should be immediately apparent, for example if you have defined a node which has not yet been assigned to an element, or if there are cable sections which are too long or too short. The Model View allows you to rotate and pan the viewpoint, to zoom in and out, and has many other useful display features (feel free to experiment) such node and element numbering, nodal coordinates, seabed topography, water surface profile etc.

Cable Solution

Another useful facility during model development is the ability to view the cable solution in the Model View. This allows you to view the model configuration after the Cable Pre-Static Step has been performed. In the vast of models which include cables, the initial static analysis runs without difficulty, and convergence is rapidly achieved. Occasionally though, the initial static fails to converge, and one reason can be that the cable solution is in fact a poor approximation to the required static configuration. In such cases, the *Source Indicator* in the Display Toolbar will state 'Cable'. If the cable solution is indeed the problem, then it invariably points to some shortcoming in the specified input data.

Solution Recommendations

This section presents some recommendations regarding the selection of appropriate solution parameters, including:

- Convergence Ratios
- Indeterminate Solutions
- Converged Solutions
- Time Stepping
- Solution Parameters
- Frequency Domain Analysis
- Quasi-Static Analysis
- Modal Analysis

Convergence Ratios

THEORY

If an analysis fails to converge successfully, the program outputs the convergence ratio for each unsuccessful iteration – this is presented at the end of the input data echo section of the main output file (with the file extension OUT). Refer to Solution Convergence for further information on convergence ratios. There are typically 15 iterations (depending on the analysis type, and the time-stepping procedure if applicable), and the table will show whether or not the analysis was approaching the convergence tolerance. It will also allow you to identify the critical degrees of freedom. If the analysis was approaching convergence, then you could possibly try increasing the maximum number of iterations permitted. In the case of static analyses, you could try reducing the load increment. In the case of dynamic analyses, you could try varying the time stepping or solution parameters, as discussed in the following sections. If the convergence ratios suggest that the analysis is highly unlikely to converge, for example if the critical ratios are alternating between two or three identical values at successive iterations, this is generally indicative of shortcomings in the model specification, and you may want to review the input specification (the advice presented in Nonlinear Properties may be important in this context).

RELEVANT KEYWORDS

- *TOLERANCE is used to define the analysis convergence tolerance measure and related data.

Indeterminate Solutions

THEORY

Occasionally, the program terminates because the structure is indeterminate. If this happens, it is most likely that the model, or a portion of it, is not sufficiently restrained. In the case of static analyses, you should consider applying additional boundary conditions to the structure. In the case of dynamic analyses, you could either apply additional boundary conditions to the structure or try varying the time stepping or solution parameters, as discussed in the following sections.

RELEVANT KEYWORDS

- *BOUNDARY is used to define boundary conditions.
- *TIME is used to define time parameters for an analysis.
- *TOLERANCE is used to define the analysis convergence tolerance measure and related data.

Converged Solutions

THEORY

If a dynamic analysis that has been running for some time terminates unsuccessfully, it may be helpful to view the structural response in the Model View for solution times up to the point at which the solution failed. There may have been a progressive deterioration of the solution prior to the actual termination, and a visual inspection might give you an indication of the source of the problem. For example, you may notice your model is susceptible to buckling, indicating it is undergoing effective compression which may be unexpected or undesired. You could also investigate results from the converged solutions up to that point. For example, time histories of effective tension, bending moment etc. can be plotted to see if they are reasonable.

RELEVANT KEYWORDS

- `*TIMETRACE` is used to request the creation of timetrace plots.

Time Stepping

THEORY

Flexcom has two time stepping algorithms, Fixed Time Stepping and Variable Time Stepping. Generally speaking, a fixed time step is used in analyses where the nature of structural response is broadly similar throughout. For example, in the case of a rigid riser, and/or benign rather than extreme loading conditions. It is well suited to lengthy simulations for a stable model, and is the preferred option provided the required time step is reasonable. A variable step is typically used in analyses where the structural response varies significantly during the course of the simulation. A good example might be that of a steel catenary riser with intermittent seabed contact.

Whether you opt for a fixed or variable time step, the Choice of Time Step is important. It is necessary to choose a time-step which picks up necessary detail in excitation and response. This applies to all time domain dynamic analyses, but it is particularly relevant to those which include significant non-linearities. As a general rule of thumb, it is recommended to use a time step of between 1/12th and 1/20th of the applied wave period in a regular wave analysis.

However, much smaller time steps may be required if the analysis includes any significant non-linearity, such as intermittent contact between a risers and a guide surface for example. Impact situations tend to cause structure response at high frequencies. This problem is exacerbated by the fact that, in general, the element length around the area of contact tends to be small, which can lead to increased structure response at these high frequencies.

Finally, as a general comment, it is important not to ignore any perceived difficulties with your analysis. For example, if you notice that the solution is taking a relatively large number of iterations at each solution step before Solution Convergence is achieved, this may suggest the program is struggling to cope with the time stepping parameters you have chosen. Similarly, if you are running a variable time step analysis, and you notice that the actual time step used never seems to approach the maximum value, then this may also indicate the specified time values are inappropriate for the analysis being performed. Caution is advised against the specification of an excessively large maximum time-step value in an effort to speed up the analysis. The variable time stepping algorithm will try to achieve the maximum value at every possible opportunity, and this can actually make the overall solution less efficient, or even affect solution robustness in certain circumstances.

RELEVANT KEYWORDS

- *TIME is used to define time parameters for an analysis.

Solution Parameters

SOLUTION PARAMETERS

If convergence problems are experienced in a time domain dynamic analysis, you could try varying the solution parameters. Before resorting to this course of action, you should try to fully understand the model behaviour, and the reasons why the analysis is failing to complete with the existing solution parameters. Possible alterations you might consider might include some of the following:

- Increasing the ramp time.
- Adding, or increasing, damping.
- Relaxing the tolerance measures.
- Utilisation of the Small Torque Value option.

DAMPING

A small level of damping is beneficial in many dynamic analyses in dissipating the effects of transients and high frequency noise. However, what particular values of l (stiffness proportional damping coefficient) and m (mass proportional damping coefficient) represent a small level of damping is very much dependent on the structure under consideration. For this reason, and because it is obviously important to quantify the effect particular values are having on the response in a particular run, it would in general be recommended that you perform a dynamic analysis with no damping, either initially or as a check. Refer to Damping for further information on structural damping.

TOLERANCE MEASURES

Note that the default tolerance levels are adequate in the vast majority of cases, and the adjustment options will be rarely invoked. You should be extremely cautious of relaxing the tolerance measures beyond recommended default values, as this could result in cumulative errors in the solution. Refer to Solution Convergence for further information on convergence ratios.

SMALL TORQUE VALUE

The inclusion of torque in the convergence calculations can in some analyses increase the number of iterations required for convergence. This is particularly true of flexible riser systems subjected to waves and current approaching at an angle oblique to the initial plane of the riser. In reality, the actual torque values may be small, and the influence of torque on the solution (as opposed to the rate of achieving this solution) minimal. The *Small Torque Value* input can be used in such cases to effectively instruct Flexcom to ignore torque in determining convergence. Refer to Solution Convergence for further information on the small torque value.

RELEVANT KEYWORDS

- `*TOLERANCE` is used to define the analysis convergence tolerance measure and related data.
- `*DAMPING` is used to incorporate damping into a dynamic analysis.

Frequency Domain Analysis

THEORY

Flexcom uses a standard technique from numerical analysis, which is termed relaxation, to improve the rate of convergence of frequency domain dynamic analyses. Using this technique, the solution of the equations of motion from a particular iteration is replaced by a weighted average of the solutions for that iteration and the previous one. Refer to Frequency Domain Convergence for further details.

Relaxation is a common numerical procedure and has the effect of improving the rate of convergence of iterative techniques, particularly in cases where results from successive iterations oscillate about the correct solution. As for the convergence parameter, the default value of 0.8 is adequate in the vast majority of cases. The value should be changed only when:

- A dynamic analysis repeatedly fails to converge, even though the tolerance measure and/or the maximum number of iterations have been altered (within reason).
- You are certain the finite element model is indeed physically reasonable and capable of withstanding the specified environmental conditions.

RELEVANT KEYWORDS

- `*TOLERANCE` is used to define the analysis convergence tolerance measure and related data.

Quasi-Static Analysis

Very occasionally the response of a structure to static loading cannot be successfully found in a static analysis. This is normally due to the fact the structure is highly sensitive to changes in configuration from iteration to iteration, and a static analysis may tend to diverge rather than converge to a stable configuration. One of the most commonly occurring cases of this sensitivity is in systems which include significant water surface piercing, such as for example a buoyancy tank used during a riser tow-out. If an initial approximation to the equilibrium position of the structure actually has an excess of, say, buoyancy over gravity, in the next iteration the structure may be largely or wholly out of the water, and subsequent iterations may find alternately increasing portions of the structure exposed or submerged, and the solution rapidly diverges. In such circumstances, a quasi-static approach is more suitable, and the model will tend settle in a dynamic fashion, typically under the influence of damping, towards its static equilibrium configuration. Refer to Quasi-Static Analysis for further discussion on quasi-static analysis, including some helpful guidelines.

Modal Analysis

THEORY

Eigenproblem Shift

The basic equation that Modes solves is K v = λ M v (as per Mathematical (Eq.1) presented earlier), where K and M are stiffness and mass matrices respectively, v is an eigenvector, and λ is the corresponding eigenvalue. Shifting by a value μ means transforming this equation into (K + μ M) v = η M v. The eigenvalues of the original and transformed equations are related by $\eta_i = \lambda_i + \mu$ for all i eigenvalues. Eigenvectors are identical. Bathe et al., (1976) claim that applying a shift to an eigenproblem can speed convergence and prevent problems when the stiffness matrix is positive semidefinite. By default Modes applies a shift of 1. In a very small number of analyses increasing this value can guarantee convergence which might otherwise not be achieved, but this option should be very rarely used.

Effective Compression and Local Buckling

Occasionally Flexcom can report the following error message during a modal analysis…

> "*Error: Solution stopped during subspace iteration. Stiffness matrix is not positive definite. Non-positive pivot for equation N, pivot=X.Y*"

Any error message relating to negative pivot terms in the global stiffness matrix generally stems from effective compression in the initial static solution. Where compression is present, this can lead to instability in the subsequent modal analysis, as the riser is susceptible to buckling and the modal solution becomes indeterminate. If this message appears, you should examine the effective tension distribution in the initial static analysis, and check to see if there are regions where the effective tension is negative.

Furthermore, even a successful modal analysis can occasionally present negative eigenvalues in the solution. These are unusual, and again may be caused by effective compression in the preceding static analysis stage.

If either of these issues are a cause of concern, there are a couple of options available to you…

- It may be possible to alter the loading on the model in order to avoid effective compression. For example, additional buoyancy material may be used to reduce the apparent weight of the structure under consideration.
- It may be possible to alter the structural properties, in order to augment the bending stiffness. The critical buckling load is related to bending stiffness, and if the resistance to bending can be increased, this may prevent local buckling.

RELEVANT KEYWORDS

- `*EIGENPAIRS` is used to specify the required number of natural frequencies, and also parameters relating to the subspace iteration algorithm, including an *Eigenproblem Shift* parameter.

Model View Troubleshooting

The following troubleshooting guide should be useful if you experience problems with the Model View. The facility works perfectly for the vast majority of users, but some users may experience a blank Model View which is incapable of displaying anything.

If at any stage you attempt a suggested corrective action and it does not help, repeat the guide from Step 1.

1. Does this problem occur with all models, including standard examples? If yes, go to Step 6, otherwise proceed to Step 2.

2. Can this model be opened in the Flexcom Database Viewer? This package may be launched from within Flexcom itself (via the Tools menu) or from the Windows Start Menu (under the Flexcom sub-section). If yes, try running Flexcom with default settings from the Windows Start Menu. If no, proceed to Step 3.

3. Was this analysis run with a later version of Flexcom than the one which you are currently using? There is a possibility that the database format may not be compatible. If yes, try re-running the analysis in this version of Flexcom. If no, proceed to Step 4.

4. Is this a Flexcom 8.2.3 analysis which also contains Point Masses? Flexcom 8.2.3 experiences problems when the number of point masses is an integer multiple of eight. If yes, either add some token point masses (with a mass of zero), or re-run the analysis in a later version of the software. If no, proceed to Step 5.

5. Is this a large model with tens of thousands of nodes/elements? If yes, and you have a 32-bit version of Flexcom installed, try installing the 64-bit version of Flexcom to see if that helps. If no, or you already have a 64-bit version of Flexcom installed, please Contact Wood with a copy of your model file.

6. Are you accessing Flexcom via Remote Desktop or Terminal Services? If yes, Flexcom uses DirectX which is not generally compatible with Remote Desktop or Terminal Services. To view the Model View, you must open your models on a computer to which you are directly logged-in. If no, proceed to Step 7.

7. Can you open any models in the Flexcom Database Viewer? If yes, try running Flexcom with default settings from the Windows Start Menu. If no, proceed to Step 8.

8. Try running the DirectX Diagnostic Tool. Search for the program “DxDiag.exe” on your machine and run it. This program is typically located in your Windows system folder or one of its sub-folders. The easiest way to find it is to use the “Search Programs” option which is located immediately above the Windows Start Button after pressing it. The DirectX Diagnostic Tool checks your DirectX and display driver installation. Once it’s running, click on the tab titled “Display 1”. The entry “Direct3D Acceleration” should read “Enabled”. If it does not, upgrade your display drivers (search your computer manufacturer's website for appropriate downloads for your system) and then reinstall DirectX from the Flexcom installation. If it does, proceed to Step 9.

9. Are you running on a laptop and does the Model View work when on the battery but not when charging, or vice versa? If yes, some laptops have dual graphics adapters (one which is used in a high-performance, high-power, setting and one which is used in a low-performance, low-power, setting) and you may be experiencing a problem with one of these adapters. It is assumed that you have already ensured that you are running the most up-to-date display drivers in Step 8. It may be possible to alter settings that will configure Flexcom to use one adapter exclusively. Consult your computer vendor for instructions. If no, Contact Wood.

NetHASP License Troubleshooting

The following steps are useful for troubleshooting any issues with network licensing which uses the NetHASP system (physical red dongle).

PART1: ON THE NOMINATED SERVER MACHINE

1. Is the red network dongle connected to the main server machine? The red dongle must be plugged into the nominated server machine – with an IP address matching that provided to Wood.

2. Is the dongle illuminated? If not, this could indicate a problem with the USB port (see Server Point 3) or incorrect drivers (see Server Point 4).

3. Is the dongle inserted into a standard USB port? If the dongle is inserted into a SuperSpeed USB port (typically labelled 'SS' followed by the normal USB ‘branch’ symbol), it may not be recognised. If so, try using a standard USB port instead. If the dongle becomes illuminated, proceed directly to Step 5. If not, continue to Step 4.

4. If the drivers have not been installed correctly, or if they have somehow become corrupted, they may need to be reinstalled. To reinstall the drivers, follow these steps:

   a. Unplug the dongle from the machine

   b. Open the Sentinel Downloads website

   c. Download "Sentinel HASP/LDK - Command Line Run-time Installer" and unzip the files to the local hard drive

   d. Open a command prompt window as an administrator (search for "cmd" on the Windows Start menu, right-click on the executable and select "Run as administrator")

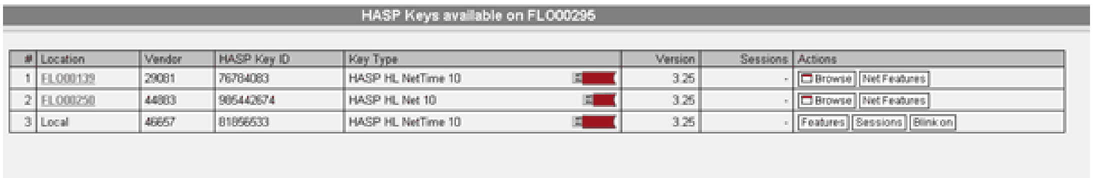
   e. Navigate to the location where you unzipped the file (you can use the command 'cd foldername' to change directory in a command prompt window; the command 'cd..' allows you to move up one directory level)

   f. Type the command 'haspdinst.exe –i' and press Enter

   g. Once the process completes successfully, the device drivers should now be fully installed

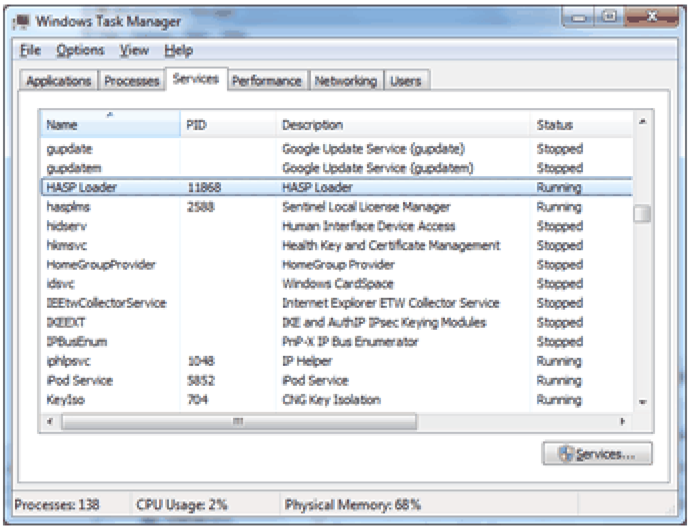
   h. Plug the dongle back into the machine again. At this point the dongle should become illuminated. If not, please Contact Us for further instructions.

5. Is the Network License Manager running on the server machine? To check this, navigate to http://localhost:1947 within an internet browser on the server machine. Click ‘Sentinel Keys’ from the menu on the left of the page. As per the screenshot below, you should see a HASP key attached with Vendor ID 46657 and Location ‘Local’.



6. Can you not see the HASP key? If the HASP key is not present or you cannot see the page, the Network License Manager may not be installed correctly.

   a. Reopen the Network License Manager Installer software, and use the Wizard to reinstall the *NetHASP License Manager* as per the instructions in the help article. Note in particular that the software should be installed as a service (rather than an application) and the setup should be run with administrator permissions.



7. Does the nominated server IP address match the MCSCode file on the client machine?

   a. Open the client MCSCode file in a text editor like Notepad++. You should see text similar to the text below. Take note of the IP address specified in the file (example IP address [123.456.78.90] highlighted in yellow).

   b. Obtain the actual IP address for the server machine. Instructions to find your IP address can be found on the Microsoft site: http://windows.microsoft.com/en-ie/windows/find-computers-ip-address#1TC=windows-7. Compare this with the IP address stored in the license file - both should be identical.

```
[Flexcom]
01/04/2014
01/04/2015
01-02-03-04-05-06
Company Name
12345
4
2014
Company Name
Company Address 1
Company Address 2

Company Name
1 2 1
FREECOM
AdfdfgPq?[Y1WK=~?!8fgjn?>$|EYN-1MGyU\f)Lh)pd%0$<$^£%=t*\$p)p(7
[Network]
1
12345, ServerName, 123.456.78.90
=I)!u8o+DUNkdjidgn9£%DY£$YS£*FHW^DGE&dfEz1zOogLF|RS7K)Rh6W}b@W
```

8. Is port 475 available for communication? Check that port 475 on the server machine is available and is not blocked by Firewall. This is the port which the license manager uses to communicate. If the port is blocked, can it be opened? If not, please use the following steps to change the port number.

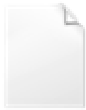
    a. The port can be set in the license manager configuration file 'nethasp.ini'. To configure the nethasp.ini file, add the following to the [NH_TCPIP] section: NH_PORT_NUMBER = and set the TCP/IP port number. The default number is 475. Replace it with a port number that is open (do not use any brackets around the number).

    b. Then on the license manager server, in the license manager installation directory should be a 'nhsrv.ini' file. Open it using a standard text editor and change the following line under the [NHS_IP] section: NHS_IP_portnum = 475. Change the number from 475 to match the number you put in the 'nethasp.ini' file, then re-start the license manager. Make sure that any firewall installed on your NetHASP License Manager host allows incoming connections on that port.

9. Does the server machine have more than one red/network NetHASP dongle attached to it? For example, it is possible (although unusual) to have two separate network licenses of Flexcom under your software contract, or you may have other third-party software which also requires a NetHASP dongle on the server machine. If this is the case, you will need to move the Flexcom dongle to another machine and inform Wood of the new IP address.

PART 2: ON THE CLIENT MACHINE

1. Is the MCSCode file placed in the Bin folder? The MCSCode file that you received from Wood should be placed in the Bin folder of the Flexcom installation directory on the client machine. For example, C:\Program Files\Wood\Flexcom\Version 2022.1.1\bin.

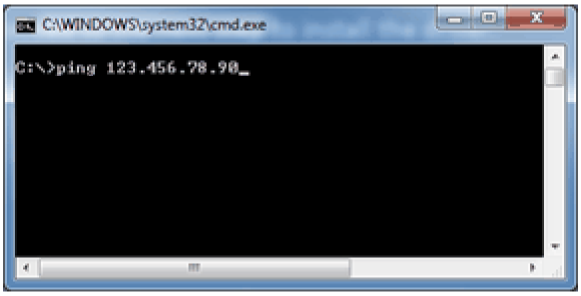


MCSCode

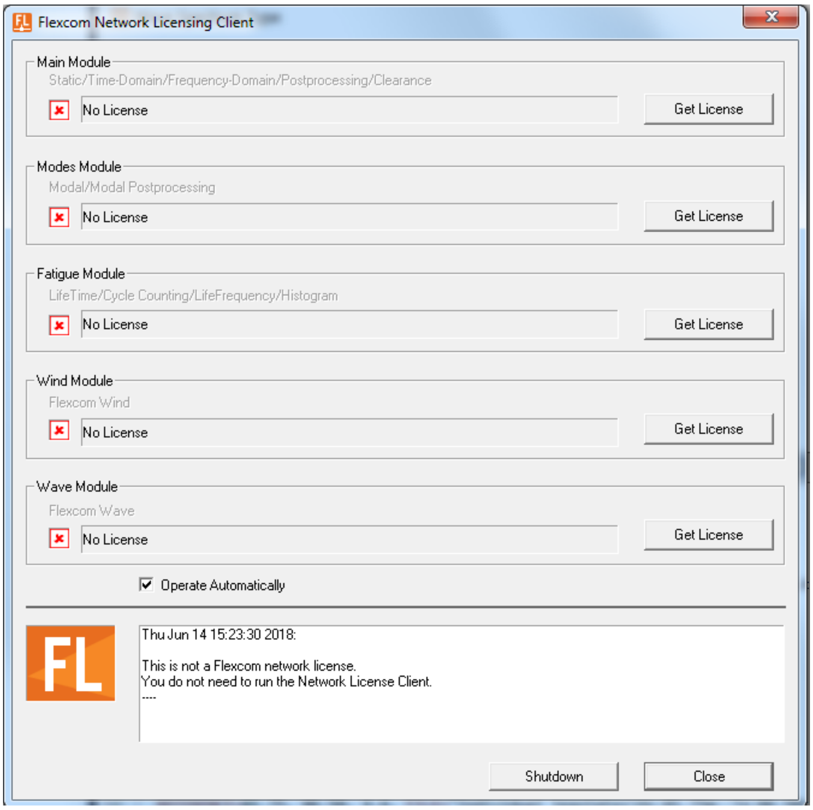
2. Can you see the license server machine on the network? Let's assume for illustrative purposes that the server IP address is 123.456.78.90. You can check that you can connect to the server machine by either:

    a. Navigating to http://123.456.78.90:1947 in your internet browser or

    b. Pinging the server by opening a command prompt in Windows, and typing 'ping 123.456.78.90' at the prompt, where '123.456.78.90' is the actual IP address of your server machine, as provided to Wood.



3. Can you get a license manually? Open the Flexcom Network Licensing Client app by selecting *Flexcom Main Menu* > *Licensing* > *Show Network License Client*. Click the first *Get License* button for Flexcom. Do you get error message(s) in the text box or do you successfully retrieve a license from the server? Any error message would be useful to help us diagnose the problem. The entire contents of the text box should be pasted into your email to us - note that the box can have scroll bars.



4. Could there still be some problem with the communication across the network? To eliminate this, let's try moving the network dongle to the client machine. In this way, the same machine will act as both the client/user and the server/host from a licensing perspective.

   a. On the user machine, follow the steps listed in Network License Manager Installer.

   b. Unplug the network dongle from the server machine, and plug it into the client machine. The dongle should illuminate. If not, this could indicate a problem with the USB port (see Server Point 3) or incorrect drivers (see Server Point 4).

   c. Obtain the IP address for the user machine. Instructions to find your IP address can be found on the Microsoft site: http://windows.microsoft.com/en-ie/windows/find-computers-ip-address#1TC=windows-7. Email this IP address to Wood. Our technical support team will create a temporary license file which will use the IP address of the user machine (rather than the server machine).

   d. When you receive the new license file, save it into the local Flexcom installation folder on the client machine.

   e. Try to get a licence with the Network Licensing Client app on the user machine

IS YOUR PROBLEM RESOLVED NOW?

The troubleshooting tips above should have helped to resolve the problem. If you are still experiencing issues, please Contact Us and provide us with as much information as possible i.e. let us know what happened at each of the various steps, including screenshots or text output as appropriate.

Contact Us

If you have problems using Flexcom, and none of the suggestions in the documentation help to resolve those problems, we are happy to provide you with technical support. Generally speaking, e-mail is the best medium by which to send support queries, as keyword files are typically required by our support team to reproduce problematic behaviour. Our support email address is sw.support@woodplc.com. We recommend that you direct all queries to this general address, rather than to any particular individual within the company. The rationale being that this address is continually monitored by several members of the support team, and so should ensure a prompt reply. Any queries addressed directly to individuals may be inadvertently overlooked, for example if that person is out of the office.

You can also contact the software business development team for more information on Flexcom and other industry leading solutions in the Wood advanced software range. Again the most reliable method of communication is via email, to our sales address sw.support@woodplc.com. We also have a web site with information about our range of products and services, http://www.woodplc.com.

Finally, feel free to contact us by phone (+353-91-481210) if you prefer.

1.9.5 Postprocessing

OVERVIEW

Flexcom provides a variety of postprocessing options, and the primary channels are as follows.

- Database Postprocessing generally represents the most comprehensive postprocessing resource. It is capable of generating Graphical Output, Tabular Output and Spreadsheet Output.

- Summary Postprocessing allows you to extract pertinent results (e.g. maximum/minimum values) from time domain analyses in succinct tabular format in the form of a Summary Output File. Although summary output is predominately used to create succinct tabular output, it is also capable of generating Graphical Output in the form of plot files.

- Summary Postprocessing Collation allows you to collate the summary postprocessing results across a range of different time domain analyses into a single Summary Collation Spreadsheet. Enhanced data visualisation is also provided by the ability to produce 3-dimensional Summary Collation Plots.

- Custom Postprocessing provides additional options for advanced users, facilitating the development of specialised postprocessing tools tailored to meet specific requirements. This section covers areas such as the Excel Add-In, Database Access Routines, and VBA code.

The second and third options, in particular, facilitate the development of specialised postprocessing tools tailored to meet the specific requirements of individual customers.

FURTHER INFORMATION

This section contains information on the following topics.

- Database Storage Files describes the options for storing output produced by a Flexcom analysis run for subsequent postprocessing.

- Database Postprocessing describes the database postprocessing facility, which generally represents the most comprehensive postprocessing resource.

- Summary Postprocessing describes the summary postprocessing facility, which allows you to extract pertinent results (e.g. maximum/minimum values) from time domain analyses in succinct tabular format.

- Summary Postprocessing Collation describes the summary collation postprocessing module, which allows you to collate the summary postprocessing results across a range of different time domain analyses.

- Timetrace describes the timetrace postprocessing facility, which is mainly used in the area of time domain fatigue analysis.

- Advanced Topics discusses advanced postprocessing options such as angles output, extreme values, spectra and ensembles, and the computation of RAOs.

- Modal Analysis Postprocessing describes the options for postprocessing modal analyses.
- Clearance & Interference Postprocessing briefly outlines the analytical capabilities of Clear, an ancillary module of Flexcom, which performs clearance/interference postprocessing calculations.
- Code Checking discusses the code checking facility.
- Force and Stress Outputs describes the calculation of various force and stress outputs provided by Flexcom.

RELEVANT KEYWORDS

- `$DATABASE POSTPROCESSING` corresponds to the database postprocessing facility, which generally represents the most comprehensive postprocessing resource. Refer to Database Postprocessing for further details.
- `$SUMMARY POSTPROCESSING` corresponds to the summary postprocessing facility, which allows you to extract pertinent results (e.g. maximum/minimum values) from time domain analyses in succinct tabular format. Refer to Summary Postprocessing for further details.
- `$SUMMARY COLLATE` corresponds to the summary collation postprocessing module, which allows you to collate the summary postprocessing results across a range of different time domain analyses. Refer to Summary Collate for further details.
- `$TIMETRACE POSTPROCESSING` corresponds to the timetrace postprocessing facility, which is mainly used in the area of time domain fatigue analysis. Refer to Timetrace Postprocessing for further details.
- `$MODES POSTPROCESSING` - This section corresponds to the postprocessing of modal analyses. Refer to Modal Analysis Postprocessing for further details.
- `$CLEAR` corresponds to Clear, an ancillary module of Flexcom, which performs clearance/interference postprocessing calculations. Refer to Clearance & Interference Postprocessing for further details.
- `$CODE CHECKING` corresponds to the code checking facility, which allows you to check analysis results against specific design codes/procedures. Refer to Code Checking for further details.

1.9.5.1 Data Storage Files

TIME DOMAIN SIMULATIONS

Various options are provided to specify the type and level of output produced by a time domain analysis run for subsequent postprocessing. The relevant data falls into two main categories, reflecting the fact that two (time domain) postprocessing modules are available in Flexcom, namely the Database Postprocessor and Timetrace Postprocessor, respectively. The differences between two output file types, and their relative advantages and disadvantages, are described in the following sections on Database Files and Timetrace Files.

The vast majority of Flexcom users tend to use database files to store program outputs. Timetrace output is essentially a legacy feature which continues to be supported for backward compatibility reasons. It's traditional advantages have since been counteracted by the ever-increasing functionality available for database files...

- Small size and quick access times have been replicated by customisable database files.
- Ability to produce a text-based output (rather than binary) has been replicated by the ability to produce spreadsheet based output.

In summary, the importance of timetrace output in the overall context of Flexcom has greatly diminished.

FREQUENCY DOMAIN SIMULATIONS

No options are provided regarding control of output produced by a frequency domain analysis run for subsequent postprocessing. The reason for this is that a frequency domain analysis is relatively concise by comparison with that of a corresponding time domain simulation, and all of the relevant output data is stored automatically in a frequency domain database file.

RELEVANT KEYWORDS

- `$DATABASE POSTPROCESSING` corresponds to the database postprocessing facility, which generally represents the most comprehensive postprocessing resource. Refer to Database Postprocessing for further details.

- `$TIMETRACE POSTPROCESSING` corresponds to the timetrace postprocessing facility, which is mainly used in the area of time domain fatigue analysis. Refer to Timetrace Postprocessing for further details.

Database Files

THEORY

Flexcom can produce two database files, referred to as the motion database file and the force database file. Together, these can contain a detailed description of the analysis solution at all or selected solution times. These files are unformatted random access files, so you cannot type, print or edit them. The motion database file can contain information on the position, velocity and acceleration of all or selected nodes in the model (including runtime generated statistics of motion). The force database file can contain information on:

(i) Restoring forces in all or selected elements

(ii) Reactions at all restraints

(iii)Statistics of restoring forces generated during runtime

Restoring forces in this context may include axial forces, effective tensions, shears, bending moments relative to both local $\hat{y}$ and $\hat{z}$ axes, and curvatures relative to these same local axes. A number of further outputs may optionally be added to these parameters.

The database files provide a very detailed picture of the structure response. However, this is often obtained at a cost. For large models, or in long simulations with a large number of database outputs, the database files can become quite large. This is frequently an unavoidable overhead, particularly in the early stages of a job, when you are trying to develop a stable, efficient model or examine the effect of a number of design options. The size of the database files may be reduced, however, by customising the contents to suit your own requirements.

Another point of note is that the Summary Postprocessing facility (which allows you to extract pertinent results in succinct tabular format) is also based on database output, so if wish to avail of this facility, the presence of database files is a fundamental prerequisite.

You may also use customised Database Access Routines to access the data stored in the database files. This feature is intended for advanced users of the software, and facilitates the development of specialised postprocessing tools tailored to meet the specific requirements of individual clients. This feature is discussed in detail in Database Access Routines.

RELEVANT KEYWORDS

- `*DATABASE` is used to specify the frequency of database output.
- `*DATABASE CONTENT` is used to customise the contents of the database output files.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser or B01 - Steel Catenary Riser.

Storage Options

OVERVIEW

Database output is generally produced by Flexcom, but you have option to disable it completely, though this would be quite unusual in practice. If you are interested in database output, a range of options is provided to specify the frequency and level of data stored.

By default, Flexcom creates a default database based on the analysis type...

- Static and quasi-static analyses – database output produced at the final solution time only (equivalent to the *End of Analysis* option under the `*DATABASE` keyword)
- Dynamic analyses – database output produced at certain time steps only (equivalent to the *Selected time steps* option under the `*DATABASE` keyword). The *Recording Interval* is equal to the solution time step in a fixed time step analysis, and the maximum time step in a variable time step analysis. The *Start Time* is equal to the solution start time.

STANDARD DATABASE CONTENTS

The standard level of output comprises the following:

(i) Nodal motions for all nodes in the model (including auxiliary nodes)

(ii) Axial forces and effective tensions for all elements

(iii) Shear forces for all elements

(iv) Bending moments and curvatures for all elements

(v) Torque moments for all elements

(vi) Reactions at all restrained nodes

You may also specify the frequency of database output. The options include All time steps (the default), which produces database output at all solution times, Selected time steps, which produces output at a recording interval you specify, and End of Analysis, which produces output at the final solution time only.

CUSTOMISED DATABASE CONTENTS

You may also customise the database, to control the contents of the motion and force database files that are produced by Flexcom. By reducing the number of parameters that are output to the database files, and/or the number of elements for which those parameters are output, and/or the number of data storage points per element, it is possible to substantially reduce the size of the database files produced. This can be particularly useful for large models and/or long simulation runtimes, especially if only a particular section of the model or certain parameters are of interest.

A further advantage of customisation is that you may request that runtime generated statistics of nodal motions and element restoring forces are included in the database files. If requested, Flexcom automatically calculates statistical parameters (minimum, maximum, mean and standard deviation) during dynamic analyses for nodal positions and certain element restoring forces (effective tensions, local-y and local-z-shear forces, torque moments, and local-y and local-z bending moments) while the analysis is running. These results are shown in the main analysis output file (**jobname.out**) at the end of the analysis. Additionally, these statistical parameters are written to the database files. This may be useful for analyses with long simulation runtimes where subsequently generating statistics using the database postprocessor can take a long time. If runtime-generated statistics have been included in the database files, and you subsequently request a plot of statistics using the database postprocessor, the postprocessor checks if the relevant data is included in the runtime statistics. If it is, the relevant data is simply read from the database files; if it is not, the database postprocessor must scan the database files to calculate the relevant statistical parameters. Including runtime-generated statistics in the database files can save a considerable amount of time during subsequent postprocessing, although this is at the expense of increased database file size.

RELEVANT KEYWORDS

- `*DATABASE` is used to specify the frequency of database output.
- `*DATABASE CONTENT` is used to customise the contents of the database output files.

If you would like to see an example of how these keywords are used in practice, refer to A01 - Deepwater Drilling Riser or B01 - Steel Catenary Riser.

Timetrace Files

THEORY

The timetrace file stores the values of selected variables at all or selected solution times, and may be in ASCII or binary format. You select the variables to be output to the timetrace file before the solution begins. Normally a relatively small number of outputs is requested, and so the file produced is significantly smaller than a standard (non-customised) database file. This is the one advantage of using the timetrace file option. However, it should be noted that modern versions of Flexcom also support customised database files for storage efficiency, so the case for using timetrace files has greatly reduced.

One disadvantage is that you must know in advance what variables are of interest, and also that your analysis is going to be completed successfully. If you perform a simulation with only timetrace output, and discover subsequently that you omitted some variables of interest, you have no remedy except to repeat the run with the timetrace requests expanded as required. In addition, if an analysis terminates unsuccessfully, as occasionally happens, the timetrace output is of limited use in determining why. For this reason, you mighty typically use the timetrace output option in the later stages of analysing a stable model, typically in long regular wave or random sea analyses.

Another application of timetrace output is the area of time domain fatigue analysis. When performing a fatigue analysis in Flexcom, the first step would be to perform a random sea analysis for each fatigue seastate. In setting up each analysis, you must store time histories of axial force, Y bending moment and Z bending moment at each location (hot spot) of interest. Earlier versions of Flexcom required these parameters be stored via timetrace output, so there is in a sense a historical link between timetrace output and fatigue analysis in Flexcom. However, more recent versions also allow you to store the axial force and bending moment data using the more widely used database output, so the importance of timetrace output in the overall context of Flexcom has greatly diminished.

One final point to note is that you can request the storage of any time history of interest. The most common application is probably the storage of axial force and bending moment as mentioned above, but it is not confined to those two. In this respect, LifeTime (which is described in Time Domain Fatigue Analysis) may also be used as a general cycle counting tool. The input in this case is typically a random time history generated from random sea analysis timetrace output. However you can use LifeTime to cycle count any time history data organised in a format which mimics the Flexcom timetrace output format. The cycle counting output is presented in terms of response histograms.

RELEVANT KEYWORDS

- `*TIMETRACE` is used to request the storage of results for timetrace postprocessing (this is mainly used in the area of time domain fatigue analysis).

1.9.5.2 Database Postprocessing

OVERVIEW

Retrieval of results from an analysis database, in Flexcom terminology, is done by making a series of postprocessing requests. Firstly, you indicate to the program which analysis variables you want to examine. Then you perform a postprocessing “run”, instructing Flexcom to extract the required results from the analysis database, and to present them in various output files for subsequent examination. Each postprocessing request typically leads to the creation of:

(i) graphical output, in the form of a plot file (with the file extension MPLT)

(ii) text based output, in the form of a tabular output file (with the file extension TAB)

(iii) spreadsheet based output, in the form of an Excel file (with the file extension XLSX)

(iv) comma separated values file (with the file extension CSV)

The first three types of output are produced by default and the last is optional. You can suppress any of these types depending on your preference for examining program results.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Output Categories

- Graphical Output
- Tabular Output
- Spreadsheet Output

RELEVANT KEYWORDS

- Numerous keywords are contained in the `$DATABASE POSTPROCESSING` section, each of which is capable of generating Graphical Output, Tabular Output and Spreadsheet Output.
  - `*STANDARD OUTPUT` is used to quickly request a selection of commonly used outputs
  - `*TIMETRACE` is used to request the creation of timetrace plots.
  - `*SNAPSHOT` is used to request the creation of snapshot plots.
  - `*STATISTICS` is used to request the creation of statistics plots.
  - `*SPECTRUM` is used to request the creation of spectrum plots.
  - `*RAO` is used to request the creation of RAO plots.
- `*OUTPUT FILES` is used to specify the types of output file required from database postprocessing.

If you would like to see an example of how these keywords are used in practice, refer to any of the standard Flexcom Examples.

Output Categories

OVERVIEW

This section provides a summary of the postprocessing request options available in terms of database postprocessing. Results can be requested in five basic categories – Timetrace, Snapshot, Statistics, Spectrum and RAO. Additional facilities are available to define Element Sets, Axes & Vectors and parameters relating to the calculation of Extreme Values. The following is a summary of each category.

STANDARD OUTPUT

The aim of the Standard Output option is to allow you to quickly request a summary of pertinent information, without the inconvenience of explicitly requesting specific outputs. This facility automates many of the (admittedly minor) tasks which you would otherwise undertake manually. For every element set referenced, Flexcom produces outputs of effective tension, resultant bending moment and von Mises stress for that set. The element set entry is also optional, and if no element sets are listed, it is assumed that output is required for all elements (i.e. element set All). For static and quasi-static analyses, snapshot plots are presented at the last solution time step. For dynamic analyses, statistical plots are created – envelopes in the case of the regular wave analyses, and extreme value plots for random sea analyses. Statistics are measured over the last two wave periods for regular wave analyses, and initial transients are automatically excluded from random sea analysis postprocessing. Given that these output parameters are rarely examined in base units (e.g. effective tension is normally presented in kN or kips, rather than N or lbf), appropriate scale factors are automatically applied to the outputs

TIMETRACE

A timetrace is a time history of the variation of a particular parameter throughout an analysis. You can request timetraces under the following headings:

Timetrace Headings

Heading	Description
Kinematic	The response of a node in a particular degree of freedom
Force	The restoring force or stress at a point on an element
Reaction	The reaction at a restrained node in a particular degree of freedom
Wave Elevation	The water surface elevation (calculated at Y=Z=0)
Time Step	The time step throughout the analysis

Element Angles	The angle between two elements
Vector Angles	The angle between an element and a vector
Axis System Angles	The angle between an element and an axis system
Guide Surface Reaction	The contact reaction at a flat guide surface in a particular degree of freedom
Zero-Gap Reaction	The contact reaction at a zero-gap guide in a particular degree of freedom
Pipe-in-Pipe Reaction	The contact reaction at a pipe-in-pipe connection in a particular degree of freedom
Clashing	The minimum clearance, contact reaction force or clashing impulse between two adjacent lines

Most of this represents standard finite element output, and the use of the Flexcom facilities is straightforward and usually self-explanatory. Flexcom provides a range of outputs under the general heading of Angles. These outputs are provided both here, in Database Postprocessing, and in Summary Postprocessing. Rather than describe the range of outputs in two sections of the manual, a detailed description is confined to one section only. You are referred to Angles Output for a more detailed discussion of Flexcom angles output.

SNAPSHOT

A snapshot is a plot of the distribution of a particular parameter throughout the structure (or a part of it you specify) at a particular time during the analysis. Flexcom has two snapshot options, namely Motion and Force. Motion requests the distribution of the motion in a global degree of freedom, while Force is available for plotting the distribution of a restoring force or stress.

STATISTICS

Options are provided for plotting statistics of motions and forces for specified element sets or the whole structure. Outputs include maximum/minimum envelopes, mean values, standard deviations and extreme values. Extreme values are discussed in further detail in the Extreme Values Section.

SPECTRUM

The results of a random sea dynamic analysis are usually presented in terms of response spectra, for which Flexcom has four options. These are Motion, Force, Reaction and Elevation. Motion, Force and Reaction have the same meaning as before, while Elevation refers to the water surface elevation at Y=Z=0. This last option is useful for comparing the realised and target wave spectra.

RAOS

Flexcom provides an option to generate RAO amplitudes and phases for selected output parameters. Specifically, it is possible to obtain RAOs for nodal motions, element forces and stresses, and floating body motions. The procedure followed by the software for the computation of RAOs depends on whether the relevant analysis involved regular or irregular wave loading. Further details are provided in Computation of RAOs.

RELEVANT KEYWORDS

- Numerous keywords are contained in the `$DATABASE POSTPROCESSING` section, each of which is capable of generating Graphical Output, Tabular Output and Spreadsheet Output.
  - `*STANDARD OUTPUT` is used to quickly request a selection of commonly used outputs
  - `*TIMETRACE` is used to request the creation of timetrace plots.
  - `*SNAPSHOT` is used to request the creation of snapshot plots.

- o *STATISTICS is used to request the creation of statistics plots.
- o *SPECTRUM is used to request the creation of spectrum plots.
- o *RAO is used to request the creation of RAO plots.

- *OUTPUT FILES is used to specify the types of output file required from database postprocessing.

If you would like to see an example of how these keywords are used in practice, refer to any of the standard Flexcom Examples.

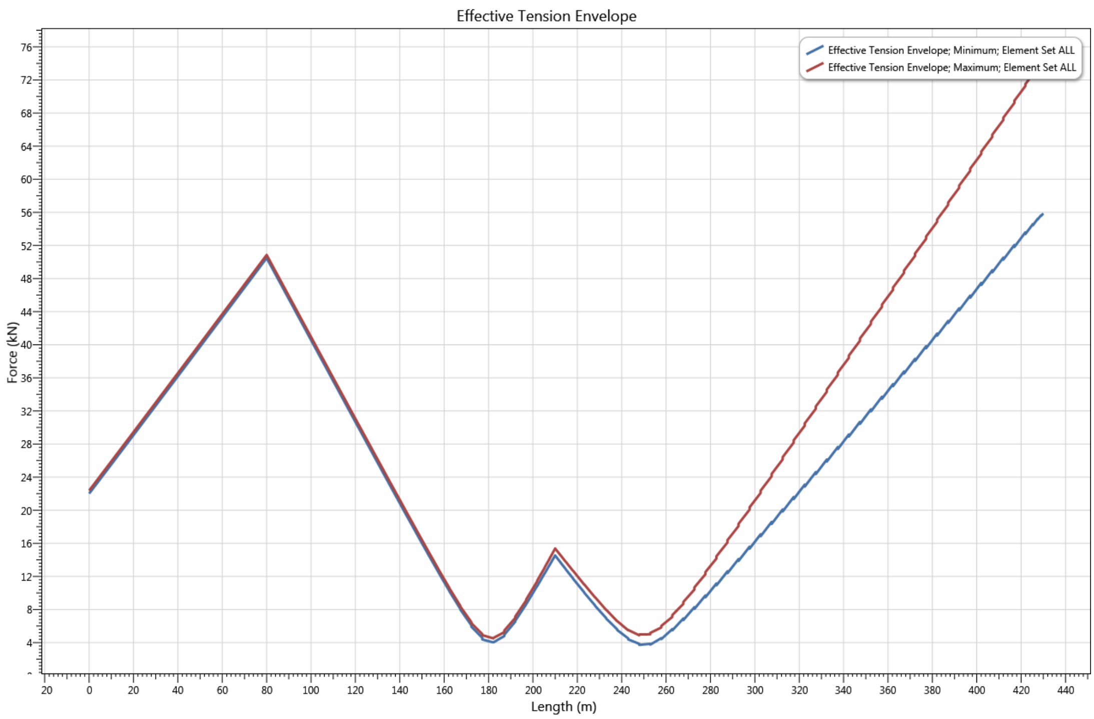
Graphical Output

OVERVIEW

It was noted earlier that a plot file is created for each database postprocessor request, and these may be examined using the Plotting facility.

A standard plot file extension (.mplt) is now used for all Flexcom plots. Earlier versions of the software actually used different extensions for the various plot types – .mpt for timetrace plots, .mpf for spectral plots, .mps for snapshot plots and .mpd for statistical plots. The plot file numbers, indicated above by n, increase from 1 up to possible maximum of 9999. The format of each plot file number depends on the maximum number of files requested, such that each plot file name has the same length. So for example, if you request 9 plots, n will vary from 1 to 9. If 99 plots are requested, n will vary from 01 to 99, and so on.

Optionally, a further CSV (comma separated values) file is produced to accompany a .mplt plot file. The extension of these files is .csv. These contain similar information to the plot file but the advantage is that they are easily opened in any spreadsheet application such as Excel, for further processing.



Sample Graphical Output

Tabular Output

A text based output file (with the file extension TAB) is also produced when you perform database postprocessing. In this file, a separate table is written for each postprocessing request. This table is basically a summary of the Graphical Output in a report ready format.

The table below shows a sample table corresponding to a spectrum plot. This consists of a section where the actual spectrum values are tabulated, followed by a statistical summary which includes the zeroth to fifth moments of the spectrum.

Sample Tabular Output

```
--------------------------------------------------------------------------------
!  Frequency   Spectrum   !  Frequency   Spectrum   !  Frequency   Spectrum   !
!    (Hz)                 !    (Hz)                 !    (Hz)                 !
--------------------------------------------------------------------------------
! 0.00000E+00 0.44770E-01 ! 0.16830E+00 0.65933E-02 ! 0.33659E+00 0.31624E-05 !
! 0.39139E-02 0.21326E+00 ! 0.17221E+00 0.65054E-02 ! 0.34051E+00 0.11010E-04 !
! 0.78278E-02 0.16746E+00 ! 0.17613E+00 0.99972E-02 ! 0.34442E+00 0.10945E-04 !
! 0.11742E-01 0.11122E+00 ! 0.18004E+00 0.67062E-02 ! 0.34834E+00 0.18436E-05 !
!                                                                              !
!  etc. etc.                                                                   !
!                                                                              !
! 0.15264E+00 0.27675E-02 ! 0.32094E+00 0.73455E-05 ! 0.48924E+00 0.10707E-06 !
! 0.15656E+00 0.12318E-01 ! 0.32485E+00 0.11777E-04 ! 0.49315E+00 0.51445E-06 !
! 0.16047E+00 0.24879E-01 ! 0.32877E+00 0.94371E-05 ! 0.49706E+00 0.20823E-06 !
! 0.16438E+00 0.16842E-01 ! 0.33268E+00 0.12654E-04 ! 0.50098E+00 0.24877E-06 !
--------------------------------------------------------------------------------


                         Response Statistics
                         ===================

            Mean Value:                         0.30991E+01
            Spectrum Zeroth Moment, M0:         0.17787E+00
            Standard Deviation (Root M0):       0.42174E+00
            Spectrum First Moment,  M1:         0.11235E-01
            Spectrum Second Moment, M2:         0.78061E-03
            Spectrum Third Moment,  M3:         0.59568E-04
            Spectrum Fourth Moment, M4:         0.50587E-05
```

Spreadsheet Output

Spreadsheet based output, in the form of an Excel file, basically contains the same data as the Tabular Output, but it is conveniently formatted in a spreadsheet. For every postprocessing request, relevant output data is presented in an individual worksheet of the Excel file.

The figure below shows a sample spreadsheet output corresponding to a timetrace plot. When you request a timetrace of a variable, the output file contains a statistical summary including maximum, minimum, mean and standard deviation values.

	A	B	C	D	E
1	Flexcom				
2					
3	*Version 8.3.2*				
4					
5	**Reaction Timetrace; Node 87; DOF 1**				
6					
7	**Reaction Timetrace**				
8	**Node No.**	87			
9	**DOF**	1			
10					
11	**Maximum:**	75.838	**kN at time:**	10.5	seconds
12					
13	**Minimum:**	55.75	**kN at time:**	35.5	seconds
14					
15	**Mean Value:**	64.873	**Standard Deviation:**	6.5131	

Sample Spreadsheet Output

The Plotting feature has a facility to export data from individual plots to Excel or CSV (comma seperated value) files. This is very useful if you wish to process the plot data further or present it differently. When working with a large number of plot files, perhaps through a script, this export procedure can get tedious. A more efficient approach is to request an additional CSV file to accompany all plot files (via the `*OUTPUT FILES` keyword). It should then be straightforward to extract the data from these CSV files into another application or through a script of your own.

1.9.5.3 Summary Postprocessing

THEORY

Flexcom provides a powerful facility for generating a Summary Output File. Here the maximum value, the minimum value, the range of values and the standard deviation of values is tabulated for a group of parameters you specify. The output is in a succinct format and is suitable for inserting directly into a study report. The response statistics are calculated by default over the full simulation length, but options are provided to specify start and end times for the computations.

If you are performing a series of analyses (for example to examine a large number of different load cases), the Summary Postprocessing Collation facility provides a useful means of assembling all the pertinent summary data across a range of load cases into a single Summary Collation Spreadsheet. Enhanced data visualisation is also provided by the ability to produce 3-dimensional Summary Collation Plots.

The parameters for inclusion in the summary output file can be specified after an analysis completes, and the summary output file itself subsequently generated. More importantly, the parameters for the summary file can be specified before an analysis begins, at the same time perhaps as the analysis data is input. In this latter case the summary postprocessing runs automatically after the analysis completes, without further user intervention. In this way report-ready information is immediately available, without the need for further postprocessing. This offers the facility for increased automation of riser design, for example in a situation where a large number of load cases needs to be run but only a small number of outputs must actually be checked in each.

In addition to the tabular summary file, you can optionally request that Flexcom generate a timetrace plot file for some or all of the parameters you nominate for inclusion in the summary output. In this way all of the tabular and plot data required for inclusion in a report can be generated at the same time.

Note that the capability to generate a summary output file is not available for frequency domain analyses. This is mainly because the effort involved in performing and postprocessing analyses in the frequency domain is less onerous than in the time domain. For example, the multiple regular wave facility allows you to run a series of regular wave analyses over a range of periods, and response RAOs may be readily produced via Database Postprocessing. So all of the options pertaining to summary postprocessing, and consequently the remainder of this section, relate specifically to time domain analysis.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Output Parameters
- Other Points to Note
- Summary Output File

The section on Summary Postprocessing Collation is highly relevant also.

RELEVANT KEYWORDS

- Numerous keywords are contained in the `$SUMMARY POSTPROCESSING` section, allowing you to extract pertinent results (e.g. maximum/minimum values) from time domain analyses in succinct tabular format. For example, you could use the `*PARA FORCE ENVELOPE` keyword to request summary output of statistics of element restoring forces. Or you could use the `*STANDARD OUTPUT` keyword to quickly request a selection of commonly used outputs.

If you would like to see an example of how these keywords are used in practice, refer to J03 - Summary Collation.

Output Parameters

OVERVIEW

The parameters for inclusion in the summary output file are grouped under 7 headings, namely:

(i) Nodal motions

(ii) Element restoring forces

(iii) Reactions

(iv) Restoring force envelopes

(v) Angle between two elements

(vi) Angle between an element and a vector or axis system (vii)Seabed parameters

(viii)Pseudo-curvature and associated angle and tension

Very many of these parameters are straightforward. For example, in the case of nodal motions or reactions, you simply identify a node and a global degree of freedom. Likewise in the case of element restoring forces, you simply identify an element, a location on the element (start node, midpoint or end node), and the restoring force of interest, whether effective tension, bending moment, von Mises stress or any of the standard outputs. Force envelopes are similar except that you identify a range or set of elements, and again the restoring force of interest. Flexcom then loops over all the elements of the set and all three locations on each element in computing the response statistics. The remaining categories require further elaboration. This is provided in the following sections.

STANDARD OUTPUT

The aim of the Standard Output option is to allow you to quickly request a summary of pertinent information, without the inconvenience of explicitly requesting specific outputs. This facility automates many of the (admittedly minor) tasks which you would otherwise undertake manually. For every element set referenced, Flexcom produces outputs of effective tension, resultant bending moment and von Mises stress for that set. The element set entry is also optional, and if no element sets are listed, it is assumed that output is required for all elements (i.e. element set All). Statistical data is presented for dynamic analyses (i.e. maximum, minimum, mean and standard deviation). Note that the option is not relevant for static or quasi-static analyses. Statistics are measured over the last two wave periods for regular wave analyses, and initial transients are automatically excluded from random sea analysis postprocessing. Given that these output parameters are rarely examined in base units (e.g. effective tension is normally presented in kN or kips, rather than N or lbf), appropriate scale factors are automatically applied to the outputs.

ANGLES OUTPUT

Flexcom provides a range of outputs under the general heading of Angles. Specifically, Flexcom can calculate and output the angle between an element and i) another element, ii) a vector, and/or iii) an axis system, which can be either the global axes or a local axis system you define. These outputs are provided both here, in Summary Postprocessing, and in Database Postprocessing. Rather than describe the range of outputs in two sections of the manual, a detailed description is confined to one section only. You are referred to Angles Output for a more detailed discussion of Flexcom angles output.

SEABED PARAMETERS

Flexcom offers three options under the heading of Seabed Parameters as follows:

- Clearance. The first summary output option is the statistics of clearance (minimum and maximum distance) between this range of elements and the seabed, whether rigid or elastic, horizontal or of arbitrary bathymetry.
- Length on Seabed. The second option is to calculate the statistics of the length of riser in the element range lying on the seabed during the analysis.
- Maximum Span and Average Gap. The third option is to request outputs of maximum span and average gap. Maximum Span is defined as the maximum continuous length of pipeline that is not supported by the seabed. Average Gap is defined as the average clearance between the pipeline and the seabed within the region of the maximum span. The Average Gap is also presented over the central third of the maximum span.

In all cases, you identify a range or set of elements. All options are largely self-explanatory. Note however that in calculating the first option, Clearance, what actually happens is that for the element set you specify the program first attempts to identify “sagging” and “hogging” sections within the set. The minimum clearance calculations are then based on the sagging sections and the maximum clearance calculations on hogging sections. Where no such sections can be identified, the calculations cannot be performed, and a zero value is returned by default. Consider for example the case of a free-hanging catenary partly lying on the seabed, where seabed clearance is requested for a set of elements well above the touchdown zone. Flexcom will report zero clearance in this case, because no sagging and hogging regions are present. Although this might appear counter-intuitive the program is operating as intended.

PSEUDO-CURVATURE

Pseudo-curvature is a parameter used by some manufacturers and contractors in the design of bend stiffeners. The parameter is defined as follows:

$$\kappa_p = 2T\sin^2\frac{\theta}{2} \qquad (1)$$

where:

- κ_p is pseudo-curvature

- T is axial force or effective tension
- θ is rotation angle

Pseudo-curvature is calculated for an element and location (start, middle or end) you identify. You also specify whether you want Flexcom to use axial force or effective tension in the calculations. Finally, the rotation angle θ is the angle between the element you nominate and either another element, a vector or an axis system that you also identify.

When you add pseudo-curvature to the list of summary output parameters, the actual statistics output to the summary file are slightly different to those output for the other parameters. Specifically, Flexcom calculates and outputs:

- The maximum value of k_p, and the associated values of T and θ
- The maximum value of θ, and the associated values of k_p and T
- The maximum value of T, and the associated values of k_p and θ

Minima and ranges are not parameters of interest with respect to pseudo-curvature. An example of pseudo-curvature output is provided later in the Summary Output File section.

RELEVANT KEYWORDS

Standard Output

- `*STANDARD OUTPUT` is used to quickly request a selection of commonly used outputs.

Motions

- `*PARA KINEMATIC` is used to request summary output of nodal motions.

Forces

- `*PARA FORCE` is used to request summary output of element restoring forces.
- `*PARA FORCE ENVELOPE` is used to request summary output of statistics of element restoring forces.

Reactions

- `*PARA REACTION` is used to request summary output of nodal reactions.

Angles Output

- `*PARA ANGLE AXIS` is used to request summary output of the angle between an element and either a vector or an axis system.
- `*PARA ANGLE ELEMENT` is used to request summary output of the angle between two elements.
- `*PARA ANGLE TENSION` is used to request summary output of angle/tension/pseudo-curvature.
- `*AXIS/VECTOR` is used to define axis systems and vectors for use in postprocessing.

Seabed Parameters

- `*PARA SEABED` is used to request summary output of statistics of parameters related to seabed contact.

Miscellaneous Control Variables

- `*COLLATE PLOT AXES` is used to define a number of key parameters which uniquely identify a particular simulation within a load case matrix. This is a prerequisite for generating 3-dimensional Summary Collation Plots.
- `*ELEMENT SETS` is used to group individual elements into element sets.
- `*OPTIONS` is used to select a number of options relating to summary postprocessing.
- `*RESTART` is used to indicate that a postprocessing run is to be restarted from an analysis file of different stub name.
- `*TIME` is used to specify the time interval over which the summary output statistics are to be calculated.

If you would like to see an example of how some of these keywords are used in practice, refer to E01 - CALM Buoy - Simple.

Other Points to Note

THEORY

Flexcom issues a warning in the summary output file if the maximum or minimum of a particular parameter is more than two standard deviations from the mean value. The warning consists of a '*' character beside the parameter in the summary output table, together with a legend indicating the significance of the character. An example is given in Summary Output File. This facility is intended to alert users to the possibility that a maximum or minimum value in the summary output is perhaps a numerical "spike" rather than a genuine extremum. To confirm whether or not this is the case, you would look at the timetrace of the response, presuming one was generated. Flexcom generates warnings by default, but you can invoke an option to suppress this.

You may also specify whether or not the results database is to be deleted when the summary output postprocessing is completed. By default results databases are not deleted. This option might be useful when a large number of analyses is being performed. The results database could automatically be used to generate the summary output file for each analysis and each database could then be deleted, freeing disk space for subsequent analyses.

RELEVANT KEYWORDS

- `*OPTIONS` is used to select a number of options relating to summary postprocessing.

Summary Output File

A sample summary output file is shown in the table below. Note that 5 rows end in the character *. This means, as explained at the bottom of the file, that the minimum or maximum values contained in the file (or both), represents a value which is more than two standard deviations from the mean value. You would be recommended in such a case to examine the timetrace plots for these parameters, to ensure that the predicted extrema are not spurious numerical "spikes".

```
Example 2 - Free Hanging Catenary - Dynamic

Summary of results from analysis:  Example2-dynamic
--------------------------------------------------------------------------------
Variable                                  Minimum    Maximum    Range     Standard
                                                                          Deviation
--------------------------------------------------------------------------------
(1) Motions
-----------
Riser Top Motions - DOF 1 (m)            138.216    141.783     3.568      1.266
- DOF 2 (m)                              157.001    162.998     5.997      2.119

(2) Angles
----------
True Angle between Riser & Vessel        152.042    167.566    15.523      5.147

(3) Reactions
-------------
Reaction @ PLEM - DOF 1 (kN)               0.878      0.878     0.000      0.000 *
- DOF 2 (kN)                             -20.191      2.249    22.440      5.876 *
- DOF 3 (kN)                              -0.006     -0.003     0.002      0.001 *

(4) Forces
----------
Effective Tension @ Riser Top (kN)        62.156     91.966    29.811      9.365

(5) Force Envelopes
-------------------
Effective Tension Distribution (kN)        1.303     91.966    90.664     12.622 *
Curvature Distribution (kN)                0.000      0.101     0.101      0.000 *

(6) Seabed parameters
---------------------
Length on Seabed                          82.333     99.667    17.333      5.333
--------------------------------------------------------------------------------
Notes:
------
(1) Parameters calculated over time interval      13.301 to     57.301

* Warning: Extreme values for this parameter are greater than two standard deviations
           from mean.
```

Sample Summary Output File

1.9.5.4 Summary Wave Scatter

OVERVIEW

The Summary Wave Scatter feature allows you to generate Summary Database Files for seastate combinations which you have not actually simulated. Based on some selected 'reference seastates' in your scatter diagram, Flexcom estimates simulation results for adjacent seastates based on an extrapolation technique. You may then use the Summary Postprocessing Collation to collate information from all seastates.

The simulation of an entire scatter diagram in the time domain can be quite time consuming, so this is a highly efficient solution technique which can save you considerable computational time. However given the inherent approximation involved, caution is strongly advised regarding the application of this feature. The Summary Wave Scatter feature should never be used to support detailed engineering design. It can certainly be used during preliminary feasibility studies, and may be used with caution as part of front end engineering design (FEED) studies.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- User Operation provides a series of steps for you to follow in sequential fashion. It should serve as a helpful reference guide when using the Summary Wave Scatter feature.
- Extrapolation Technique provides some background theory on the computational methodology.

The section on Summary Postprocessing Collation is highly relevant also.

RELEVANT KEYWORDS

- `*SCATTER DIAGRAM` is used to specify the wave scatter diagram, including the grouping of similar seastates into blocks and the nomination of appropriate reference seastates.

If you would like to see an example of how these keywords are used in practice, refer to K01 - Floating Dual-Body Point Absorber.

User Operation

Using the Summary Wave Scatter feature is quite straightforward, you simply follow a logical sequence of steps. The following instructions should serve as a helpful reference guide.

- Obtain metocean data for your geographical location of interest. Depending on your data source, the data may be readily transferred into a spreadsheet for ease of inspection.

- Create a new Flexcom keyword file and insert a new `$SUMMARY WAVE SCATTER` section. The Summary Wave Scatter feature requires access to many input and output files, so this keyword file is typically placed in the top/root level of your project folder.

- Define the wave scatter diagram in Flexcom. When transferring the metocean data from Excel into Flexcom, the quickest way is to use the Scatter Diagram table - this will allow you to copy and paste directly from Excel, and is more efficient than working with `*SCATTER DIAGRAM` in the keyword editor.

- Note that the wave spectrum used in the definition of the scatter diagram must be either Pierson-Moskowitz or Jonswap. Use the `*WAVE SPECTRUM` keyword to inform Flexcom which spectrum type you are using, and also whether your wave periods are defined in terms of T_z or T_p or T_e.

- Use the Scatter Diagram table to group similar seastates into blocks and to nominate a reference seastate to represent each block. Some caution is advised regarding the number and selection of reference seastates which are chosen to approximately represent the full scatter diagram. This will require engineering judgment on your part, but the following general advice may be helpful.

  - Generally speaking, an inaccuracies associated with the extrapolation process are likely to be more pronounced across different wave periods rather than wave amplitudes. Hence it is not advisable to have blocks which span more than two different values of $T_z/T_p/T_e$.

  - Structural responses at smaller wave amplitudes tend to be more linear than responses at higher amplitudes, so the extrapolation technique is likely to be more accurate.

- Create Flexcom keyword files to perform time domain simulations for each reference seastate. It may be helpful to create a single master template input file which will automatically generate all the required input files to simulate each seastate. Use the `*PARAMETERS` keyword to create variables to represent H_s and $T_z/T_p/T_e$. Then you can use either `*VARIATION` or `*EXCEL VARIATIONS` to define the various combinations of H_s and $T_z/T_p/T_e$ which you wish to consider (i.e. the reference seastates).

- Add a `$DATABASE POSTPROCESSING` section to the master template file and use the `*TIMETRACE` keyword to request the creation of time history plots of pertinent output variables. Nomination of these parameters is essential as the Summary Wave Scatter feature will use the plot files created by Flexcom for the reference simulations in order to generate Summary Database Files for all the non-reference seastates in the wave scatter diagram. Refer to Extrapolation Technique for further information.

- Run the time domain simulations and associated postprocessing for the reference seastates.

- Run the Summary Wave Scatter feature. This will generate Summary Database Files for all the seastates in the wave scatter diagram.

- Create a new Flexcom keyword file and insert a new `$SUMMARY COLLATE` section. The Summary Collation feature requires access to many input and output files, so this keyword file is typically placed in the top/root level of your project folder. Use the `*IDENTIFY` keyword to nominate the sub-folders of interest within your project workspace. Use the `*PLOT` keyword to request the creation of 3D plots as a function of H_s and $T_z/T_p/T_e$. Ensure that the *Figure Titles* that you specified in the `*TIMETRACE` keyword of the `$DATABASE POSTPROCESSING` section are consistent with the *Output Title* that you specify in the `*PLOT` keyword.

- Run the Summary Postprocessing Collation and examine the newly created Summary Collation Plots. Note that the Summary Collation Spreadsheet provides a tabular representation of the same data also.

Extrapolation Technique

Following successful completion of the time domain simulations for the reference seastates, you are required to create time history plots of variables of interest via database postprocessing.

- Flexcom users may request these plots via the `*TIMETRACE` keyword, which leads to the creation of 2D Plots with an MPLT file extension.

- Users of Flexcom Wave need not worry about this preliminary step as all tasks are handled automatically by the software for user convenience.

As discussed previously, you will already have grouped similar seastates in the wave scatter diagram into blocks, and nominated a reference seastate to represent each block. For a given block, the Summary Wave Scatter feature estimates a time history of each variable of interest for the remaining cells within the block, $VAR_{Cell}(t)$, using the following relationship:

$$VAR_{Cell}(t) = \overline{VAR_{Ref}} + \left(VAR_{Ref}(t) - \overline{VAR_{Ref}}\right) \times \left(\frac{\int_0^{\infty} S_{Cell}(\omega)\, d\omega}{\int_0^{\infty} S_{Ref}(\omega)\, d\omega} \right)$$

(1)

where $VAR_{Ref}(t)$ is the time history of the variable of interest for the reference seastate,

$\overline{VAR_{Ref}}$ is the mean value of the reference time history, S_{Cell} is the wave elevation spectrum for the non-reference cell, and S_{Rref} is the wave elevation spectrum for the reference seastate.

Note that the time histories of resultants/magnitudes are always positive by definition, so the mean value may be meaningless. The following types of variables represent resultants/magnitudes:

- `TYPE=KINEMATIC` - DOF 7 (Magnitude of Rotation) and DOF 8 (Magnitude of Translation)
- `TYPE=REACTION` - DOF 7 (Magnitude of Rotation) and DOF 8 (Magnitude of Translation)
- `TYPE=FORCE` - Force variables 8 (Resultant Bending Moment), 9 (Resultant Curvature), 10 (Resultant Shear), 13 (von Mises Stress) and 18 (von Mises Stress (API-2RD))

Note also that for the special case of Damper Power (force variable 23) the mean value is set to zero.

Flexcom subsequently computes maximum, minimum, range and standard deviation values from the estimated time history in the standard fashion. This information is then used to generate a Summary Database File for each non-reference seastate in the block.

Wave Energy Period

INTRODUCTION

Flexcom has traditionally modelled wave spectra in terms of either T_z or T_p. If you specify a scatter diagram in terms of T_e, Flexcom converts T_e into T_z. Cahill and Lewis (2014) state that energy period is related to mean zero up-crossing period as follows:

$$T_e = \beta T_z \tag{1}$$

$$\beta = \left(\frac{4.2+\gamma}{5+\gamma}\right)\left(\frac{11+\gamma}{5+\gamma}\right)^{1/2} \tag{2}$$

Separate conversion procedures are used by Flexcom depending on the wave spectrum type.

PIERSON-MOSKOWITZ

For Pierson-Moskowitz spectra, γ is equal to 1, so the computation of β and T_z is trivial.

JONSWAP

As discussed in Jonswap Wave, Isherwood (1987) defines wave steepness s as:

$$s = \frac{2\pi H_s}{g T_z^2} \tag{3}$$

where g is the gravitational constant. The value of γ is then found from:

$$\begin{aligned} \gamma &= 10.54 - 1.34 s^{-\frac{1}{2}} - \exp(-19 + 3.775 s^{-\frac{1}{2}}) \quad for\, s \geq 0.037 \\ \gamma &= 0.9 + \exp(18.86 - 3.67 s^{-\frac{1}{2}}) \quad for\, s < 0.037 \end{aligned} \tag{4}$$

Flexcom estimates an initial value for T_z, calculates s as per Eq.(3), γ as per Eq.(4), β as per Eq.(2), and finally Tz from Eq.(1). This computation procedure is performed iteratively, until the estimated and computed values of T_z converge.

1.9.5.5 Summary Postprocessing Collation

OVERVIEW

As discussed in the preceding section, Flexcom provides a powerful facility for generating a Summary Output File. For any given analysis, the maximum value, the minimum value, the range of values and the standard deviation of values is tabulated for a group of parameters you specify.

If you are performing a series of analyses (for example to examine a large number of different load cases), the Summary Postprocessing Collation facility provides a useful means of assembling all the pertinent summary data across a range of load cases into a single Summary Collation Spreadsheet. Enhanced data visualisation is also provided by the ability to produce 3-dimensional Summary Collation Plots. You can plot the variation of any summary postprocessing output against any key driving parameters. For example, you can plot maximum effective tension as a function of both wave period and incident wave heading in a 3-dimensional space.

OPERATION

Summary collation is an extremely useful feature which is very easy to use. There are a number of logical steps involved in the process.

- Firstly, you need to explicitly request the creation of summary output from the individual simulations. Numerous keywords are contained in the `$SUMMARY POSTPROCESSING` section, allowing you to extract pertinent results. For example, you could use the `*PARA FORCE ENVELOPE` keyword to request summary output of statistics of element restoring forces. Or you could use the `*STANDARD OUTPUT` keyword to quickly request a selection of commonly used outputs. After a simulation has completed, Flexcom produces a text-based Summary Output File for visual inspection, but more importantly it also creates a Summary Database File, which is effectively a binary version of the same data, for subsequent collation.

- Secondly, you also need to define one or more key parameters which uniquely identify each individual simulation within a load case matrix. For example, wave period and wave direction would be defining parameters in a regular wave load case matrix, and serve as ideal parameters for identification purposes. Identifiers must be explicitly defined via the `*COLLATE PLOT AXES` keyword.

- Thirdly, you request the collation of parameters of interest across the entire load matrix (or a subset of it). The `$SUMMARY COLLATE` section contains a number of relevant keywords, such as `*IDENTIFY` which nominates the folders of interest on your hard drive. After the collation process, Flexcom produces a very helpful Summary Collation Spreadsheet, which assembles all the important data into a single location, providing you with a dashboard view of critical design parameters.

- Finally, it also possible to request the creation of Summary Collation Plots for enhanced data visualisation. Specifically, you can plot the variation of any summary postprocessing output against any key driving parameters. For example, you can plot maximum effective tension as a function of both wave period and incident wave heading in a 3-dimensional space.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Summary Database File
- Data Collation
- Summary Collation Spreadsheet
- Summary Collation Plots

RELEVANT KEYWORDS

- Numerous keywords are contained in the `$SUMMARY COLLATE` section, which allows you to collate the summary postprocessing results across a range of different time domain analyses.

If you would like to see an example of how these keywords are used in practice, refer to J03 - Summary Collation.

Summary Database File

THEORY

Every time a Summary Output File is created for an analysis, a corresponding Summary Database File (with the extension DBS) is created automatically also. While the former is stored in plain text format to facilitate visual inspection, the latter is stored in binary format for subsequent access during the summary collation process. Each DBS file contains general descriptive information relating to the analysis, such as the analysis title, the start and end times used to generate the summary data, relevant element set definitions etc. It also contains the actual results for each parameter summarised, including:

- Descriptive title
- Parameter type (e.g. motion, force, angle etc.)
- Node/element number
- Degree of freedom
- Scale factor
- Maximum value, and corresponding time of occurrence
- Minimum value, and corresponding time of occurrence
- Mean value
- Range
- Standard deviation

RELEVANT KEYWORDS

- Numerous keywords are contained in the `$SUMMARY POSTPROCESSING` section, allowing you to extract pertinent results (e.g. maximum/minimum values) from time domain analyses in succinct tabular format. For example, you could use the `*PARA FORCE ENVELOPE` keyword to request summary output of statistics of element restoring forces. Or you could use the `*STANDARD OUTPUT` keyword to quickly request a selection of commonly used outputs.

If you would like to see an example of how these keywords are used in practice, refer to J03 - Summary Collation.

Data Collation

THEORY

You typically indicate to Flexcom which DBS files are to be included in the summary collation process, by specifying a master or “root level” directory in which Flexcom is to search for relevant files. Specification of the master directory is optional, and if omitted, it defaults to the current working directory (i.e. the folder in which the Summary Postprocessing Collation keyword file is located). You may also optionally invoke a Subfolders option, to indicate that you wish to include all subdirectories of the master folder (with the exception of hidden directories, system directories and symbolic links) in the search process also.

You may also a specific “tag” (or list of such tags) to be identified in the DBS file name, in order to refine the search. If you invoke this option, Flexcom searches through the DBS files, retaining only those which contain the specified tag in their file names. In order for a DBS file to be considered, its file name must contain all the specified tags. For example, this would allow you to easily sort “near” current load cases from “far” cases.

Assuming that at least two suitable DBS files are successfully identified, the collation process begins. You have control over what parameters you wish to collate, based on the descriptive titles assigned to various outputs during the creation of the Summary Output File for each individual analysis. This entry is optional however, and if you do not specify any descriptive titles for collation purposes, Flexcom will attempt to collate all available data. Only parameters which share the same descriptive title are compared (any outputs with blank titles are ignored).

You may also specify exclusion criteria to be applied, in terms of threshold levels for minima, maxima or standard deviation. If you choose to specify a threshold minimum, then any analysis where the parameter under consideration is less than the threshold is omitted. Likewise, if you opt to specify a threshold maximum or standard deviation, then any individual analysis results where the relevant parameter exceeds the threshold level is omitted. An output must pass all specified filters before it is included in the collated results.

RELEVANT KEYWORDS

- `*IDENTIFY` is used to identify output files for inclusion in summary postprocessing collation.

- *COLLATE is used to specify the summary postprocessing data to be collated and any exclusion criteria. This keyword is optional, and if you do not explicitly designate certain parameters for collation purposes, Flexcom will attempt to collate all available data.

If you would like to see an example of how these keywords are used in practice, refer to J03 - Summary Collation.

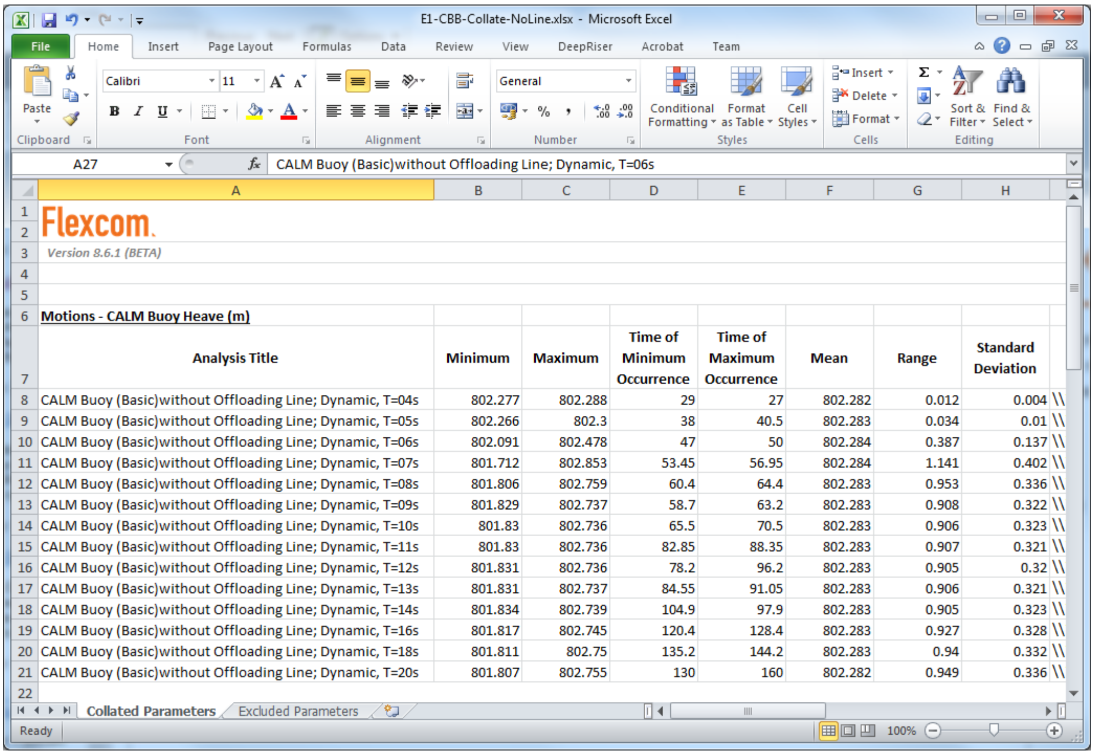
Summary Collation Spreadsheet

LIST OF COLLATED PARAMETERS

For every parameter which is collated (i.e. based on the descriptive titles assigned to various outputs during the creation of the Summary Output File for each individual analysis), a summary table is presented with the following headings:

- Analysis Title
- Maximum value, and corresponding time of occurrence
- Minimum value, and corresponding time of occurrence
- Mean value
- Range
- Standard deviation
- Filename and path of Summary Database File

The table below shows a sample extract from a summary collation file. Based on a coupled analysis from the standard Flexcom examples set, it summaries the heave motions of a CALM buoy for a range of applied regular wave loading.

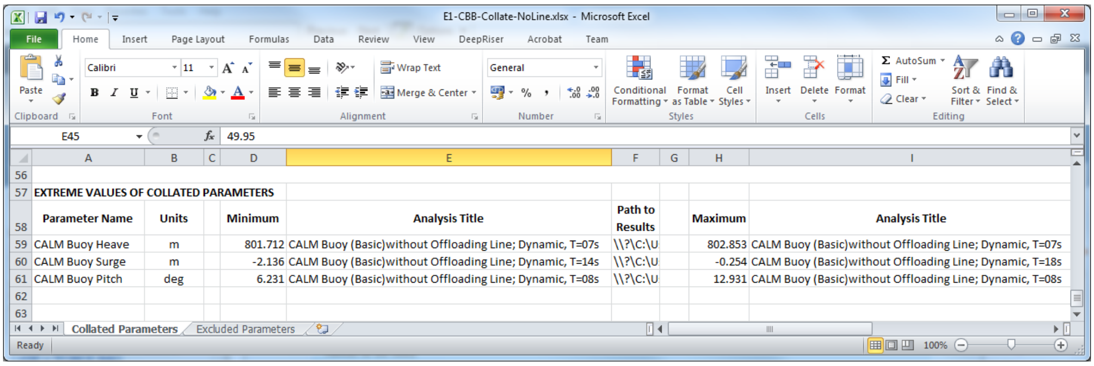


	Analysis Title	Minimum	Maximum	Time of Minimum Occurrence	Time of Maximum Occurrence	Mean	Range	Standard Deviation
8	CALM Buoy (Basic)without Offloading Line; Dynamic, T=04s	802.277	802.288	29	27	802.282	0.012	0.004
9	CALM Buoy (Basic)without Offloading Line; Dynamic, T=05s	802.266	802.3	38	40.5	802.283	0.034	0.01
10	CALM Buoy (Basic)without Offloading Line; Dynamic, T=06s	802.091	802.478	47	50	802.284	0.387	0.137
11	CALM Buoy (Basic)without Offloading Line; Dynamic, T=07s	801.712	802.853	53.45	56.95	802.284	1.141	0.402
12	CALM Buoy (Basic)without Offloading Line; Dynamic, T=08s	801.806	802.759	60.4	64.4	802.283	0.953	0.336
13	CALM Buoy (Basic)without Offloading Line; Dynamic, T=09s	801.829	802.737	58.7	63.2	802.283	0.908	0.322
14	CALM Buoy (Basic)without Offloading Line; Dynamic, T=10s	801.83	802.736	65.5	70.5	802.283	0.906	0.323
15	CALM Buoy (Basic)without Offloading Line; Dynamic, T=11s	801.83	802.736	82.85	88.35	802.283	0.907	0.321
16	CALM Buoy (Basic)without Offloading Line; Dynamic, T=12s	801.831	802.736	78.2	96.2	802.283	0.905	0.32
17	CALM Buoy (Basic)without Offloading Line; Dynamic, T=13s	801.831	802.737	84.55	91.05	802.283	0.906	0.321
18	CALM Buoy (Basic)without Offloading Line; Dynamic, T=14s	801.834	802.739	104.9	97.9	802.283	0.905	0.323
19	CALM Buoy (Basic)without Offloading Line; Dynamic, T=16s	801.817	802.745	120.4	128.4	802.283	0.927	0.328
20	CALM Buoy (Basic)without Offloading Line; Dynamic, T=18s	801.811	802.75	135.2	144.2	802.283	0.94	0.332
21	CALM Buoy (Basic)without Offloading Line; Dynamic, T=20s	801.807	802.755	130	160	802.282	0.949	0.336

Sample List of Collation Parameters

CRITICAL LOAD CASES

Towards the end of the spreadsheet, a useful summary of extreme values is also presented. For every parameter which is collated, this section highlights the most critical analyses in the load case matrix, both in terms of greatest maxima and greatest minima attained. For example, the largest heave motions of the CALM buoy occur for a wave period of 7s, and this is readily evident from the summary data.



Parameter Name	Units	Minimum	Analysis Title	Path to Results	Maximum	Analysis Title
CALM Buoy Heave	m	801.712	CALM Buoy (Basic)without Offloading Line; Dynamic, T=07s	\\?\C:\U	802.853	CALM Buoy (Basic)without Offloading Line; Dynamic, T=07s
CALM Buoy Surge	m	-2.136	CALM Buoy (Basic)without Offloading Line; Dynamic, T=14s	\\?\C:\U	-0.254	CALM Buoy (Basic)without Offloading Line; Dynamic, T=18s
CALM Buoy Pitch	deg	6.231	CALM Buoy (Basic)without Offloading Line; Dynamic, T=08s	\\?\C:\U	12.931	CALM Buoy (Basic)without Offloading Line; Dynamic, T=08s

Extreme Values of Collation Parameters

RELEVANT KEYWORDS

- *IDENTIFY is used to identify output files for inclusion in summary postprocessing collation.
- *COLLATE is used to specify the summary postprocessing data to be collated and any exclusion criteria. This keyword is optional, and if you do not explicitly designate certain parameters for collation purposes, Flexcom will attempt to collate all available data.

Summary Collation Plot

OVERVIEW

The overall operation of the summary collation feature was outlined already, but a brief synopsis is useful at this point.

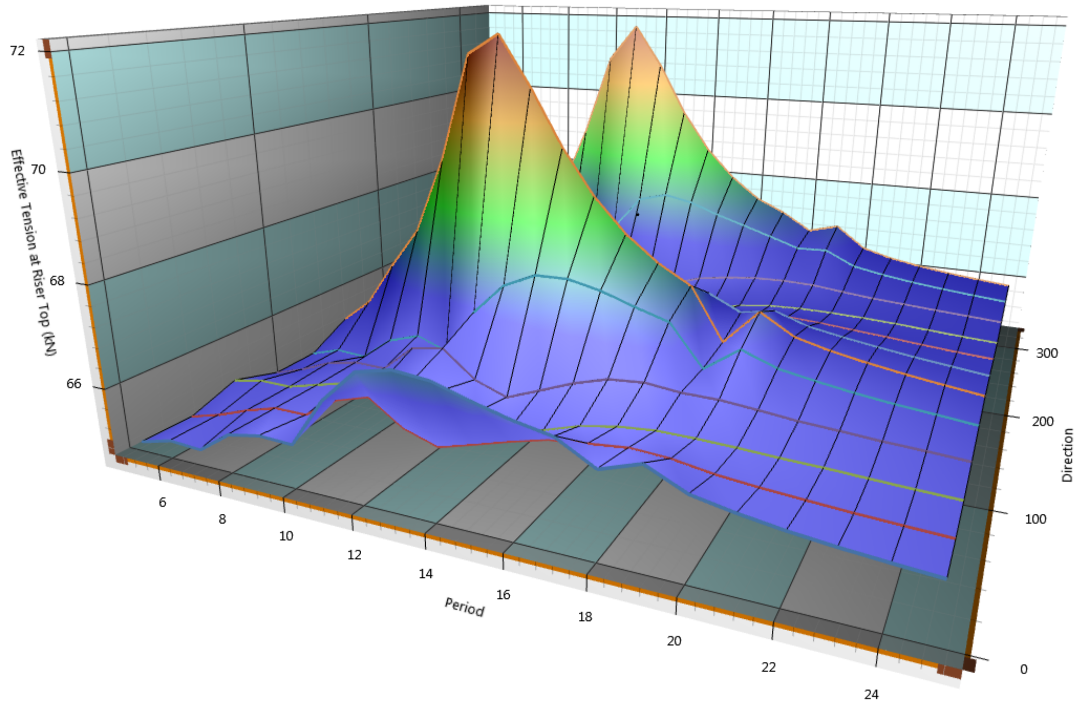
- Firstly, you explicitly request the creation of summary output from the individual simulations, achieved via the inclusion of the relevant $SUMMARY POSTPROCESSING keywords. After a simulation has completed, Flexcom stores the relevant data in Summary Database Files.
- Secondly, you define one or more key parameters which uniquely identify each individual simulation within a load case matrix. Identifiers are explicitly defined via the *COLLATE PLOT AXES keyword.
- Thirdly, you request the collation of parameters of interest across the entire load matrix (or a subset of it). The $SUMMARY COLLATE section contains a number of relevant keywords, such as *IDENTIFY which nominates the folders of interest on your hard drive.
- In addition to a Summary Collation Spreadsheet, it also possible to request the creation of Summary Collation Plots, via the *PLOT keyword, for enhanced data visualisation. Specifically, you can plot the variation of any summary postprocessing output against any key driving parameters. For example, you can plot maximum effective tension as a function of both wave period and incident wave heading in a 3-dimensional space.

PLOT DATA

Summary collation plots are typically 3-dimensional, based on the following axes.

- The relevant output variable is plotted on the vertical axis. Depending on the required statistical measurement, the meaning/scale on the vertical axis can vary. Typically the maximum value attained is presented as this is usually of greatest interest. However, it is possible to examine minimum, mean, range or deviation also.
- The first key driving parameter is plotted on the primary horizontal axis.
- The second key driving parameter is plotted on the secondary horizontal axis. This parameter is optional, and a 2D Plot will be created if it is omitted.

The following sample plot presents maximum effective tension (kN) as a function of both wave period (s) and incident wave heading (degrees).



Sample Summary Collation Plot

RELEVANT KEYWORDS

- `*IDENTIFY` is used to identify output files for inclusion in summary postprocessing collation.

- *PLOT is used to request the creation of a summary collation plot which graphically presents the variation of any summary postprocessing output against key driving parameters.
- *COLLATE is used to specify the summary postprocessing data to be collated and any exclusion criteria. *COLLATE is used to specify the summary postprocessing data to be collated and any exclusion criteria. This keyword is optional, and if you do not explicitly designate certain parameters for collation purposes, Flexcom will attempt to collate all available data.

1.9.5.6 Timetrace Postprocessing

Postprocessing of timetrace results is slightly different to database postprocessing.

Database postprocessing requests can be specified after an analysis completes, and the plot files generated subsequently. Alternatively, the postprocessing requests can be specified before an analysis begins, and the postprocessing runs automatically after the analysis completes, without further user intervention.

In the case of timetrace postprocessing, the variables of interest must be nominated before the analysis run. This is because the timetrace file only stores selected variables (typically a relatively small number of outputs is requested), so the file produced is significantly smaller than the corresponding database file. Postprocessing consists in this instance of (i) creating a file where statistical parameters for each output variable are tabulated, and (ii) optionally generating timetrace and spectra plot files for each output variable in the timetrace output file.

Further details are provided in Storage Format and Output Files.

Storage Format

OVERVIEW

In earlier versions of Flexcom, timetrace output was stored in ASCII format by default. In this case, values at each solution time are output on a series of lines – the actual solution time (only) is on the first line of the series, and the requested outputs are then on subsequent lines, typically four values to a line. If the Import format is requested, the file format is similar to ASCII, but all values at a particular solution time, including the actual solution time itself, are output on a single line. This allows the data to be readily imported into Excel.

Text based files allow the data to be readily examined and interpreted, but have the disadvantage that for a long simulation with many outputs requested, the file size can be quite large. The recommended storage method is now Binary – the main advantages being that storage/retrieval of data tends to be quicker, and file sizes are smaller, but means that you cannot edit the file manually.

FILE FORMAT

If you are interested in creating your own customised timetrace files, for example to perform cycle counting on some arbitrary time series, the Import format is the preferred option. The GRD file (this is the file extension used by Flexcom to denote timetrace output files) is comprised of a header block containing some general settings and the names of the stored variables, with the main body of the file containing the actual time histories. The layout of a text based file (i.e. either Import or ASCII format) is now described.

Header Block

- Line 1: Storage format (e.g. Import)
- Line 2: Program version (e.g. Flexcom 8.2.1)
- Line 3: Analysis title
- Line 4: Integer value indicating if a random sea is present in the analysis (a value of 1 denotes Yes, a value of 0 denotes No)
- Line 5: Integer value indicating if a fixed time step is used in the analysis (a value of 1 denotes Fixed, a value of 0 denotes Variable)
- Line 6: Number of data series stored in the main body of the file (including Wave Elevation, which is stored by default by Flexcom)
- Line 7: Number of outputs per line in the main body of the file (relates specifically to the ASCII format only, and is typically assigned a value of 4)
- Line 8: Maximum cut-off frequency of all random seastates in the analysis (this variable is assigned a value of 0.0 if the analysis does not contain any random seas)
- Line 9: Dominant direction of random sea in the analysis (this variable is assigned a value of 0.0 if the analysis does not contain a random sea, or if more than one random sea is present)

- Line 10: Acceleration due to gravity, Base units (see note (a)), Unit system flag (see note (b))
- Line 11: Analysis fixed time step (this variable is assigned a value of 0.0 if the analysis uses a variable time step)
- Line 12: Name of the first time series stored in the main body of the file (typically the first time series corresponds to Wave Elevation, which is stored by default by Flexcom)
- Line 13: Scale factor for the first time series stored in the main body of the file

The remainder of the header block is comprised of pairs of lines similar to Lines 12 and 13, listing the names and scale factors of any subsequent time series stored in the main body of the file. The total number of time series is defined in Line 6, and the number of lines in this section must be consistent with this definition.

In relation to unit systems, the following points are noteworthy.

(a) . Base units depend on the extension of the keyword files, and has the value = 1 (if .keyxm), = 2 (if .keyxi) = 3 (if value of "Acceleration due to gravity" is not recognized as metric or imperial), and = 4 (if .keyx, and "Acceleration due to gravity" is recognized as metric or imperial).

(b) . Unit system flag refers to the unit system actually used in the analysis, and = 1 (if metric), = 2 (if imperial) = 3 (user defined but value of "Acceleration due to gravity" is not recognized as metric or imperial, and = 4 (if user defined and "Acceleration due to gravity" is recognized as metric or imperial).

Main Body

In the case of an Import format file, the remaining lines contain N values, where N is the total number of time series as defined in Line 6, plus one for the actual time values themselves. The first value represents time, and the remaining values represent the various parameters (as listed in Lines 12, 14 etc.) corresponding to that particular time.

In the case of an ASCII format file, the layout is a little more cumbersome. The main body of the file is organised into blocks of lines. The first line of the block contains the time value on its own. The next line contains a maximum of N values, where N is the number of outputs per line as defined in Line 7. These values represent the various parameters (as listed in Lines 12, 14 etc.) corresponding to the time value. If the total number of time series (as defined in Line 6) is less than or equal to the number of outputs per line (as defined in Line 7), this line concludes the block of data for this time step. Otherwise subsequent lines (containing a maximum of N values) are used to present values for the remaining data series at the relevant time value. Once the block of data is complete, the next block begins with the time value for the next time, and so on.

RELEVANT KEYWORDS

- `*TIMETRACE` is used to request the storage of results for timetrace postprocessing (this is mainly used in the area of time domain fatigue analysis).

Output Files

THEORY

When postprocessing timetrace data, a combination of plot files and a statistical summary file is produced. By default no plot files are produced. When the data comes from a regular wave analysis, you have the option of requesting a timetrace plot file for each variable in the data file. In the case of random sea data, you have the option of requesting (i) a timetrace plot, (ii) a plot of the spectrum of the output variable, or (iii) both plots.

Statistical output is always generated whenever the timetrace postprocessor is run. An example statistics extract is shown in the table below.

Sample Timetrace Output

```
                ***********************************************************
                *                                                         *
                *   F L E X C O M         Timetrace Post-processing       *
                *                                                         *
                ***********************************************************

     Generic file name:         exam3
     Analysis title:
        EXAMPLE ... 3D STEEP WAVE RISER ... DYNAMIC ANALYSIS
     Time history length:         30.00 seconds
     Processing start time:       22.05 seconds

     ***********************************************************************************
     No. Mean       Standard   Predicted  Predicted  Root        Extreme    Extreme
         Value      Deviation  Minimum    Maximum    Mo          Minimum    Maximum

     ***********************************************************************************
     1 0.285E-02  0.246E+01 -0.325E+01  0.375E+01  0.246E+01 -0.325E+01  0.375E+01
     2 0.120E+03  0.138E+01  0.118E+03  0.122E+03  0.138E+01  0.118E+03  0.122E+03
     **********************************************************************************


                               Channel Descriptions
                               ====================

        Channel No. 1 is WAVE ELEVATION, m
        Channel No. 2 is MOTION NODE      71, DOF 2, m

              ***********************************************************
              *                         Legend                          *
              *                         ======                          *
              *                                                         *
              *   No.                 - Channel no.                     *
              *   Mean Value          - Calculated from time history    *
              *   Standard Deviation  - Calculated from time history    *
              *   Predicted Minimum   - Minimum value in time history   *
              *   Predicted Maximum   - Maximum value in time history   *
              *   Root Mo             - Square root of zeroth moment    *
              *                         of response spectrum            *
              *   Extreme Minimum     - Calculated from spectrum        *
              *   Extreme Maximum     - Calculated from spectrum        *
              *                                                         *
              ***********************************************************
```

As you can see, the file contains a table of statistics, a description of each channel and a legend section. There is one line in the statistics table for each of the variables output to the timetrace file. The statistical parameters in the table are as follows:

- Mean value
- Standard deviation
- Predicted minimum value
- Predicted maximum value
- Square root of the zeroth moment of the spectrum of the variable
- The extreme minimum value
- The extreme maximum value

The first four parameters in the statistical table are calculated directly from the timetrace data using standard statistical procedures. The next parameter requires that the response timetrace is transformed into a response spectrum. The extreme values may be calculated using either the Rayleigh or Weibull distributions, as described in detail in Extreme Values.

If the timetrace data is created by a regular wave analysis then no response spectra are calculated and outputs (v), (vi) and (vii) default to outputs (ii), (iii) and (iv) respectively.

When the timetrace data is the result of a random sea analysis it is useful to compare outputs (ii) and (v). If they are not approximately equal then it is possible that the time history is not long enough for unambiguous statistics calculations, and that a longer simulation time may be required.

The data in the table above is an example from postprocessing results from a regular wave dynamic analysis. The outputs shown in the table above are augmented in the case of a random sea analysis by a table showing the zeroth to fourth moments of the response spectrum for each variable in the data.

RELEVANT KEYWORDS

- `*PLOT` is used to request the creation of plots using the timetrace postprocessing.
- `*PARAMETERS` is used to define miscellaneous parameters for use in timetrace postprocessing.

1.9.5.7 Advanced Topics

This section contains information on advanced postprocessing options such as:

- Angles Output describes the Timetrace outputs Flexcom can produce in regards to angles.
- Extreme Values introduces Flexcoms capabilities for estimating extreme values of the structural response.
- Spectra and Ensembles discusses how spectra and ensembles are calculated
- Computation of RAOs outlines the generation of RAO amplitudes and phases for selected output parameters

Angles Output

OVERVIEW

Flexcom provides a range of timetrace outputs under the general heading of Angles. Specifically, you can request a time history...

1. Angle between Two Elements
2. Angle between an Element and a Vector
3. Angle between an Element and an Axis System
4. Angle between a Vector and a Vector

RELEVANT KEYWORDS

Database Postprocessing

- `*TIMETRACE` is used to request the creation of timetrace plots. Specifically,
  - `TYPE=ELEMENT_ELEMENT_ANGLE` is used to request a timetrace of the angle between two elements.
  - `TYPE=ELEMENT_VECTOR_ANGLE` is used to request a timetrace of the angle between an element and a vector.
  - `TYPE=ELEMENT_AXIS_ANGLE` is used to request a timetrace of the angle between an element and an axis system.
  - `TYPE=VECTOR_VECTOR_ANGLE` is used to request a timetrace of the angle between two vectors.

Summary Postprocessing

- `*PARA_ANGLE_ELEMENT` is used to request summary output of the angle between two elements.
- `*PARA_ANGLE_AXIS` is used to request summary output of the angle between an element and either a vector or an axis system.

Angle between Two Elements

THEORY

The simplest output under this heading of Angles is the angle between two elements. When you invoke this option you must naturally input the numbers of the two elements in question. Note that one of the elements can be a so-called auxiliary element. This allows you to find the angle between, say, the last element on a riser coming into a vessel connection and an auxiliary element representing a flange or some part of the vessel. What Flexcom actually outputs when you invoke this facility is the true or actual angle between the two elements, in the plane formed by the elements. Mathematically, the angle is the arc-cosine of the dot product of two unit vectors directed along the elements (in their instantaneous positions), from first-specified to second-specified nodes. This angle will be in the range 0° to 180°.

RELEVANT KEYWORDS

Database Postprocessing

- `*TIMETRACE` is used to request the creation of timetrace plots. Specifically, `TYPE=ELEMENT_ELEMENT_ANGLE` is used to request a timetrace of the angle between two elements.

Summary Postprocessing

- `*PARA_ANGLE_ELEMENT` is used to request summary output of the angle between two elements.

Angle between an Element and a Vector

THEORY

The second type of Angles output is the angle between an element you identify and a vector that you also define. In order to define a vector, you associate a name or label with the vector, input the vector components in the global axes, and identify the node at which the vector is located. This node can be either a node of the structure finite element discretisation or a so-called auxiliary node. The significance of this last input is as follows. The vector components you input give the vector orientation at the undisplaced or initial position of the node. As the node displaces subsequently (either statically or dynamically), the rotations at the node are also applied to the vector so that the vector orientation changes with the motion of the node. In this way you can find the angle between an element and, say, a vector at an auxiliary node which is located on a floating vessel, and which is translating and rotating with that vessel. This allows you, for example, to find the true angle between the last element on a riser coming into a vessel connection and the static hang-off angle of that riser. The output in this case is again the true or actual angle between the element and the vector, in the plane formed by both of these. Mathematically, the angle is again the arc-cosine of the dot product of (i) a unit vector directed from first-specified to second-specified node of the element, and (ii) a unit vector in the instantaneous direction of the vector you specified. Again, this angle will be in the range 0° to 180°.

RELEVANT KEYWORDS

Database Postprocessing

- `*TIMETRACE` is used to request the creation of timetrace plots. Specifically, `TYPE=ELEMENT VECTOR ANGLE` is used to request a timetrace of the angle between an element and a vector.

Summary Postprocessing

- `*PARA ANGLE AXIS` is used to request summary output of the angle between an element and either a vector or an axis system.

Angle between an Element and an Axis System

THEORY

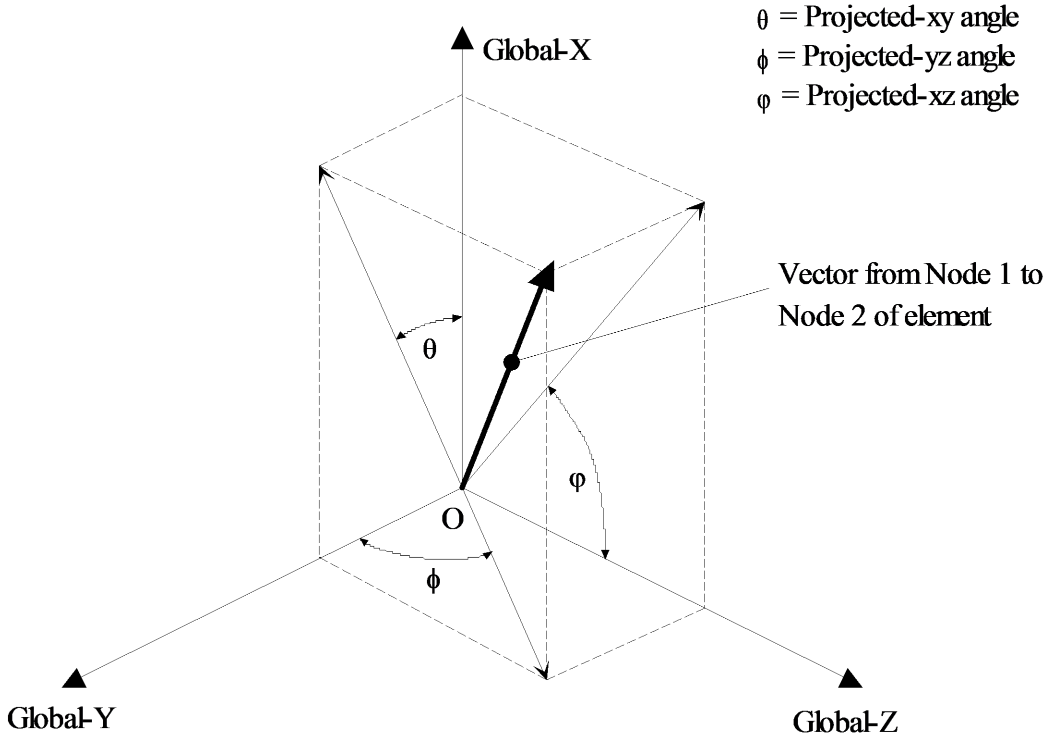
The third category of Angles output provided by Flexcom is the angle between an element you identify and an axis system you specify. This axis system can be either (i) the global axes, or (ii) a local axis system which you define. In this latter case you again associate a name or label with the axes, and input the vector components, in the global axes, of the local x- and y-axes of the system you are defining. The local z-axis is automatically found by Flexcom using the right-hand rule. When inputting a local axis system, you must again identify the node at which the axes are located, which can again be either a node of the structure finite element discretisation or a so-called auxiliary node. The significance of this input is as follows. The local x- and y-axes you specify give the orientation of the axis system at the undisplaced or initial position of the node. As the node displaces subsequently (either statically or dynamically), the rotations at the node are also applied to the local axes, so that the orientation changes with the motion of the node. This is similar to the corresponding facility available when requesting the angle between an element and a vector. The global axes of course are fixed throughout all analyses.

ANGLE PROJECTIONS

Four output options are available in this case of the angle between an element and an axis system. These are denoted (i) the Actual angle, (ii) the Projected-xy angle, (iii) the Projected-xz angle and (iv) the Projected-yz angle. The significance of these is now discussed, with reference to the two figures below.

Global Axis System

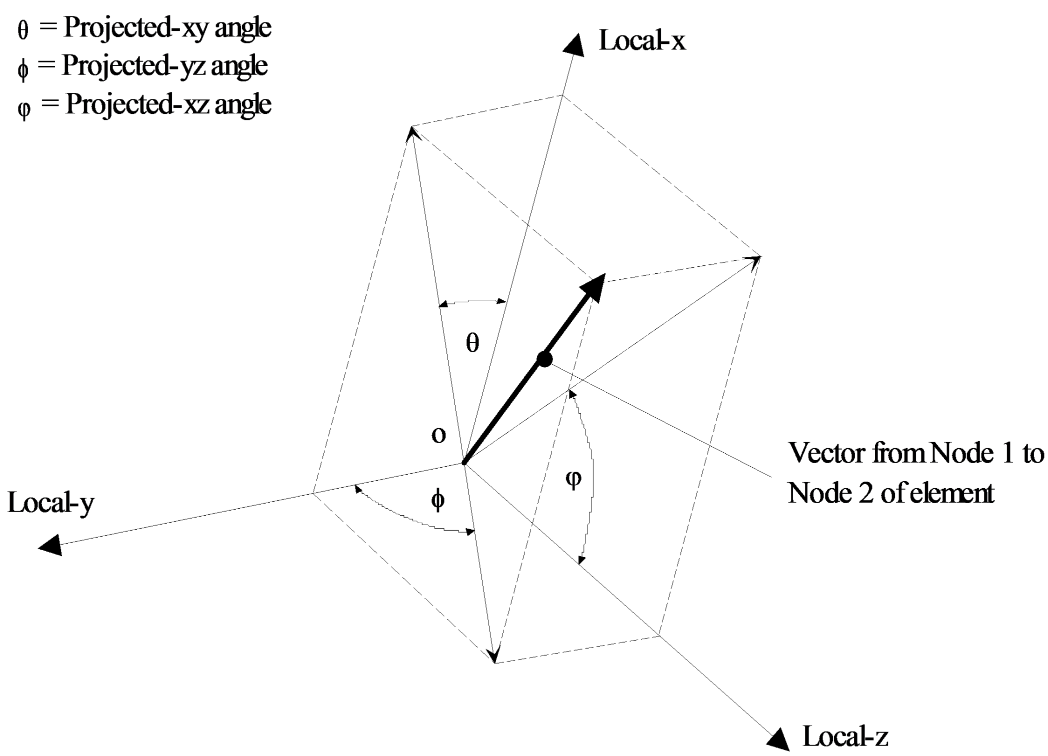
Dealing first with the case of the angle between an element and the global axes, as shown in the figure below, then Actual angle is the true or actual angle (using the same definition as previously) between the element in its instantaneous position and the global X-axis. This angle will be in the range 0° to 180°. The Projected-xy angle is the angle between the global X-axis and the projection of the element onto the global XY-plane. The Projected-yz angle is the angle between the global Y-axis and the projection of the element onto the global YZ-plane. Finally, the Projected-xz angle is the angle between the global Z-axis and the projection of the element onto the global XZ-plane. These projected angles will all be in the range –180° to 180°.



Angles between an Element and the Global Axes

Local Axis System

For the case of the angle between an element and a local axis system, as shown in the figure below, then Actual angle is the true or actual angle (using the same definition as previously) between the element in its instantaneous position and the local x-axis, in its instantaneous position. The Projected-xy angle is the angle between the local x-axis and the projection of the element onto the local xy-plane. The Projected-yz angle is the angle between the local y-axis and the projection of the element onto the local yz-plane. Finally, the Projected-xz angle is the angle between the local z-axis and the projection of the element onto the local xz-plane. As before, the Actual angle will be in the range 0° to 180°, while the Projected angles will all be in the range –180° to 180°.



Angles between an Element and a Local Axis System

Finding angles between an element and a local axis system allows you, for example, to find the true angle between the last element on a riser connected to a subsea buoy and the buoy structure itself. Alternatively, using Projected angles allows you to find the azimuth and declination angle of the riser relative to the principal axes of the buoy.

RELEVANT KEYWORDS

Database Postprocessing

- `*TIMETRACE` is used to request the creation of timetrace plots. Specifically, `TYPE=ELEMENT AXIS ANGLE` is used to request a timetrace of the angle between an element and an axis system.

Summary Postprocessing

- `*PARA ANGLE AXIS` is used to request summary output of the angle between an element and either a vector or an axis system.

Angle between a Vector and a Vector

THEORY

The last type of Angles output is the angle between two vectors that you define. In order to define a vector, you associate a name or label with the vector, input the vector components in the global axes, and identify the node at which the vector is located. This node can be either a node of the structure finite element discretisation or a so-called auxiliary node. The significance of this last input is as follows. The vector components you input give the vector orientation at the undisplaced or initial position of the node. As the node displaces subsequently (either statically or dynamically), the rotations at the node are also applied to the vector so that the vector orientation changes with the motion of the node. The output in this case is again the true or actual angle between the two vectors, in the plane formed by both of these. Mathematically, the angle is again the arc-cosine of the dot product of unit vectors in the instantaneous directions of the vectors you specified. Again, this angle will be in the range 0° to 180°. This type of output might be used, for example, to monitor the rotation across a flex joint.

RELEVANT KEYWORDS

Database Postprocessing

- `*TIMETRACE` is used to request the creation of timetrace plots. Specifically, `TYPE=VECTOR_VECTOR_ANGLE` is used to request a timetrace of the angle between two vectors.

Extreme Values

OVERVIEW

Flexcom provides a number of important capabilities for estimating extreme values of the structural response. The extreme values can be determined for any response variable at one or more elements. The estimates are based on standard procedures in the statistics of extreme values (Ochi, 1998), (Ang et al, 1984). The estimates depend on the statistical distribution of the individual (or local) maxima/minima in the timetrace of the structural response. Flexcom computes estimates of the extreme values based on Rayleigh or Weibull distributions of the individual maxima/minima in the timetrace. The following section provides a brief review the main Extreme Values Principles to familiarise the user with this type of statistical analysis, which is usually described in only specialist publications.

RELEVANT KEYWORDS

- *STATISTICS is used to request the creation of statistics plots.
- *EXTREMA is used to specify the procedure and input parameters to be used in the computation of extreme values.

Extreme Value Principles

THEORY

An estimate of the extreme maximum/minimum value is determined from the probability distribution of the largest/smallest value in a timetrace that contains N individual maxima/minima. The theory recognises that several simulations of a stochastic timetrace (with different seeds) will produce a sequence of largest/smallest values. The extreme value is derived from the probability distribution of these largest/smallest values.

Individual maxima with a Rayleigh or Weibull probability distribution of produce an extreme value that has a Gumbel distribution. Note, however, the extreme value can be determined directly from the parameters of the Rayleigh or Weibull distribution. It is not necessary to explicitly compute parameters of the Gumbel distribution to determine the extreme values. The following describes the extreme value prediction of the maximum value. The same approach is used for the extreme value prediction of the minimum value, although in this case, the timetrace is negated beforehand so the individual minima are converted to maxima. This conversion is only explicitly required for the Weibull option. The Rayleigh distribution assumes complete symmetry between the maximum and minimum values.

The exceedance distribution of the extreme value is determined from the relationship:

$$P(X > x_{extreme}) = 1 - \left(F(x_{extreme})\right)^N \qquad (1)$$

where xextreme denotes the extreme value, F the cumulative distribution of the individual maxima in the timetrace and N the number of maxima (e.g. 1000). The extreme value is determined from the above equation in a two-stage process. Let p denote a specified exceedance probability (e.g. 0.01); then the relationship for F, p and N simplifies to:

$$F(x) = (1 - p)^{1/N} \approx 1 - \frac{p}{N} \qquad (2)$$

The approximation on the right holds for small p and large N. Solving for x in Equation (2) depends on the cumulative distribution F of the individual maxima in a timetrace.

RELEVANT KEYWORDS

- *STATISTICS is used to request the creation of statistics plots.
- *EXTREMA is used to specify the procedure and input parameters to be used in the computation of extreme values.

Rayleigh

THEORY

The standard case is a timetrace in which all of its points follow a Gaussian (normal) distribution. The individual maxima in the timetrace are Rayleigh distributed. The Rayleigh cumulative distribution is defined by:

$$F(x) = 1 - e^{-x^2/2m_0}, \quad x \geq 0 \qquad (1)$$

where m0 denotes the variance (zeroth spectral moment) of the timetrace and the maxima are measured as amplitudes above the mean. The estimate of the extreme maximum and minimum values is determined from Extreme Value Principles (Eq.1) and (Eq.2) and Equation (1) above:

$$x_{extreme} = \pm\sigma\left(2\ln\frac{N}{p}\right)^{1/2} \qquad (2)$$

where σ denotes the standard deviation of the timetrace (s = $m0^{0.5}$). In the case that the structural response has a non-zero mean μ, the extreme value becomes:

$$v_{extreme} = \mu \pm x_{extreme} \qquad (3)$$

The number of maxima N is computed from:

$$N = n_{count}\frac{\text{Storm duration}}{\text{Duration of the timetrace}} \qquad (4)$$

where ncount denotes the number of maxima counted in the timetrace. The storm duration and the duration of the timetrace are for example 3 and 0.5 hours, respectively.

RELEVANT KEYWORDS

- *STATISTICS is used to request the creation of statistics plots.
- *EXTREMA is used to specify the procedure and input parameters to be used in the computation of extreme values.

Weibull

THEORY

The timetrace of a structural response does not always follow a Gaussian distribution. In these cases the individual maxima in the timetrace can be Weibull distributed. The Weibull cumulative distribution is defined as:

$$F(x) = 1 - \exp\left(-\left(\frac{x-a}{b}\right)^{c} \right), \quad x \geq a \qquad (1)$$

where a, b and c are known as the location, scale and shape parameters, respectively. The maxima are measured as amplitudes above the mean. The method of determining these parameters is described further below. The Rayleigh distribution is a special case of Weibull with a = 0, b = $(2m0)^{0.5}$ and c = 2. The values of a, b and c will often be close to the Rayleigh values.

The estimate of the extreme value is determined from the previous Equations and Equation (1) above as:

$$x_{extreme} = b\left(\ln\frac{N}{p} \right)^{1/c} + a \qquad (2)$$

The number of individual maxima in a timetrace N is determined as per Rayleigh (Eq.4). Note, however, that the extreme minima are obtained from first negating the timetrace as previously described above and performing a separate Weibull fit of the data.

The parameters of the Weibull distribution are determined by a standard procedure of transforming Equation (1) to a straight-line relationship and performing a least-squares fit of the slope and intercept. The straight line is defined by:

$$\begin{aligned} &Y = M\,X + D \\ &Y = \ln\left(-\ln\left(1 - F(x)\right)\right), \quad X = \ln\left(x - a\right) \\ &M = c, \quad D = -c\ln b \end{aligned} \qquad (3)$$

where M and D denote the slope and Y-axis intercept. The data values for X and Y are defined:

$$Y_k = \ln\left(-\ln(1 - F_{(k)})\right) \quad X_k = \ln\left(x_{(k)} - a\right) \qquad (4)$$

where k = 1, …, n_count, x(k) denotes the individual maxima in the timetrace sorted in ascending order and F(k) is estimated using the plotting position:

$$F_{(k)} = \frac{k}{n\_count + 1} \qquad (5)$$

A threshold option is also available for the least-squares fit of the Weibull distribution. The threshold removes (censors) smaller maxima from the least-squares fit so that better accuracy can be obtained in fitting the tail of the Weibull distribution. The above procedure is modified by restricting k = kT , …, n_count where kT denotes the threshold value of k. The threshold parameter is specified as a value greater than 0 and less than or equal to 1. A threshold of 1 accepts all the maxima in the least-squares fit. A threshold of 1/3 accepts the upper 1/3 of largest maxima. A direct comparison of the Rayleigh and Weibull fits requires the threshold is equal to 1. The alternative 1/3 value is analogous to the significant wave, which is based on the 1/3 highest waves. Other threshold values can also be chosen.

RELEVANT KEYWORDS

- `*STATISTICS` is used to request the creation of statistics plots.
- `*EXTREMA` is used to specify the procedure and input parameters to be used in the computation of extreme values.

Spectra and Ensembles

THEORY

The procedure used by Flexcom in calculating spectra is as follows. Firstly, the output timetrace is divided equally into a number of smaller timetraces or ensembles. A spectrum for each ensemble is then calculated using the Fast Fourier Transform (FFT) algorithm. Finally, the actual spectrum to be output is found as an average of the spectra calculated for each ensemble. This standard procedure minimises the statistical error associated with the FFT process. You specify the number of ensembles to be used in this process, and the specified value should always be greater than 1.

Note also that the FFT algorithm requires a record with a fixed time step. When you perform a variable step analysis, obviously such a record is not available, and so Flexcom must synthesise one by interpolating from the variable step record. So you must inform Flexcom what time step to use in the synthesised record.

It is difficult to provide specific guidance on a suitable number of ensembles, as an appropriate selection is case specific. Generally speaking, if you have performed a lengthy simulation (e.g. 3 hour), you should use a larger number of ensembles than if you have performed a relatively short one (e.g. 0.5 hours). Using a relatively small number of ensembles means that the FFT algorithm uses relatively long individual records, and this tends to ensure good frequency resolution between the realised and target spectra. Conversely, using a larger number of ensembles means that the FFT algorithm uses shorter individual records, and this tends to ensure better amplitude resolution. A useful exercise is to compare the realized wave spectrum against the theoretical or target wave spectrum - Flexcom provides a specific option under the `*SPECTRUM` keyword in the Database Postprocessor to readily achieve this. If you plot the realized wave spectrum for different numbers of ensembles, the one which produces the closest agreement with the target spectrum should be most suitable.

RELEVANT KEYWORDS

- `*SPECTRUM` is used to request the creation of spectrum plots.
- Note also that the `TYPE=ELEVATION` option under the `*SPECTRUM` keyword allows you to request a spectrum of water surface elevation. It also allows you to plot the target wave spectrum, in addition to the realised spectrum, which is useful for comparison purposes.

Computation of RAOs

OVERVIEW

The Flexcom time domain postprocessing facilities facilitate generation of RAO amplitudes and phases for selected output parameters. Specifically, it is possible to obtain RAOs for nodal motions, element forces and stresses, and floating body motions. The procedure followed by the software for the computation of RAOs depends on whether the relevant analysis involved regular or irregular wave loading. The computational procedure for each of these cases is described in the following sections.

RELEVANT KEYWORDS

- `*RAO` is used to request the creation of RAO plots.

If you would like to see an example of how this keyword is used in practice, refer to A01 - Deepwater Drilling Riser.

Regular Wave RAO Computation

THEORY

In the case of time domain regular wave analyses, the procedure for the computation of RAOs is as follows. Firstly a time history of the parameter of interest is generated, and the portion of the time history corresponding to the last five wave periods of the analysis (or the last whole number of wave periods if the analysis duration is less than five wave periods) is then examined and the mean of the parameter is determined.

Next, the maximum and minimum dynamic variation of the parameter from its mean value are determined for each of the five wave periods, and hence the RAO amplitude for each of the five wave periods is computed by comparison with the height of the applied regular wave. The mean and standard deviation of the five RAO amplitude values is then computed and output by the program. This approach has the advantage that the RAO amplitude is computed based on a number of regular wave periods, and the standard deviation of the RAO amplitude permits the user to determine whether the system has achieved steady-state response (and therefore, whether the computed RAO is reliable).

The computation of RAO phase angles for regular wave analyses uses a similar procedure. The mean up-crossing point of the parameter of interest is identified for each of the last five wave periods, and the RAO phase angle is computed by comparison with the mean up-crossing point of the applied regular wave for each of the five wave periods. The mean and standard deviation of the five RAO phase values are then computed and output by the program.

RELEVANT KEYWORDS

- *RAO is used to request the creation of RAO plots.

Irregular Wave RAO Computation

THEORY

In the case of time domain irregular sea analyses, the procedure for computation of RAOs is as follows. Firstly a time history of the parameter of interest is generated. Then the parameter response spectrum is computed using the Fast Fourier Transform (FFT) algorithm. Similarly, the spectral density function for the wave elevation time history is generated. RAO amplitudes are then computed using the relation:

$$RAO(f) = \sqrt{\frac{S_P(f)}{S_\eta(f)}} \qquad (1)$$

where RAO(f) is the RAO amplitude (expressed as a function of frequency), S_n(f) is the wave elevation spectrum and S_p(f) is the spectrum of the parameter of interest.

The RAO phases are computed by algebraic subtraction of the phase angles determined from the FFT of the spectrum of the response parameter, and the phase angles determined from the FFT of the wave elevation. It should be noted that, in general, there is significant statistical error associated with this procedure for computation of phase angles, and the resulting phase angles are likely to be less accurate than those derived from regular sea analyses.

RELEVANT KEYWORDS

- *RAO is used to request the creation of RAO plots.

1.9.5.8 Modal Analysis Postprocessing

OVERVIEW

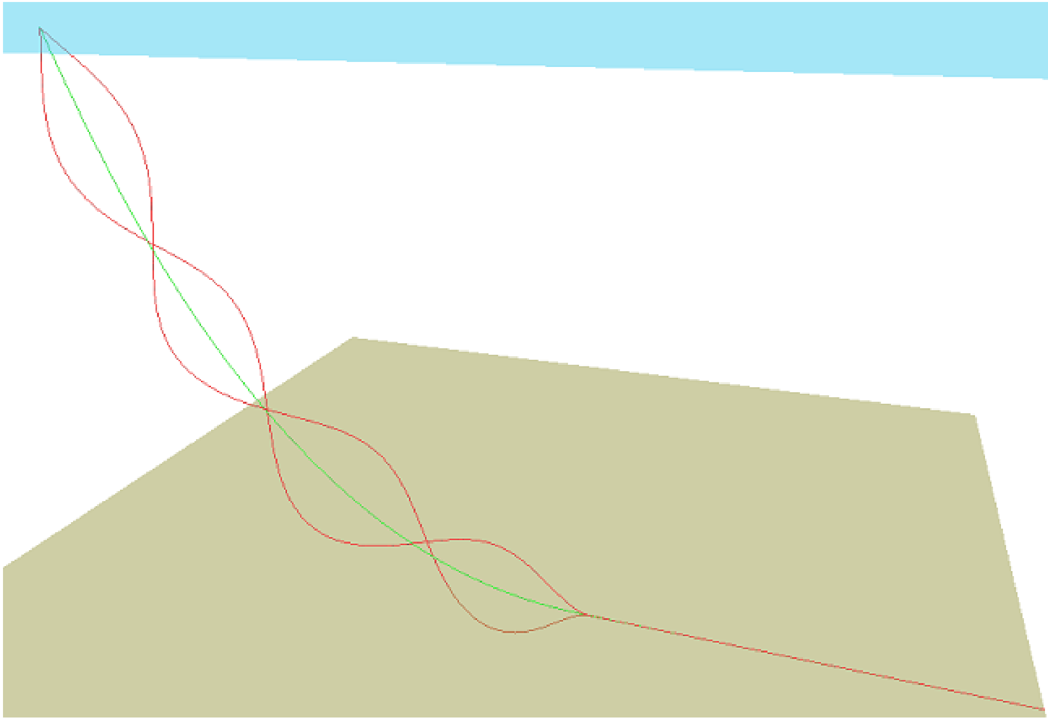
Modes Postprocessing provides a range of options for postprocessing the output from Modal Analysis. When a Modes analysis is successfully completed, details of the solution are contained in the program output file, which has the file extension out. In this file you can find a full data echo, and (optionally) a listing of some or all of the calculated natural frequencies and mode shapes. There is extensive control over the contents of this file, so the output can be as comprehensive as you want it to be. However it is not always easy to visualise the predicted mode shapes when they appear simply as tables of data values. For this reason Modes also supports graphical viewing of modal analysis results, via an on-screen 3D display of Mode Shapes. The postprocessing module can also produce plots of Modal Displacement and Curvature.

RELEVANT KEYWORDS

- `*PLOT` is used to specify selected modes for plotting, and the type of plot required.

Mode Shapes

As mentioned already, a Modal Analysis automatically produces graphical output for viewing, in a similar manner to a standard Time Domain Analysis. Unlike a standard analysis however, the modal analysis output does not represent an animation of the dynamic response of a structure. Rather, it contains a set of three-dimensional snapshots of the structure mode shapes. When you actually view the data using the display facility, the mode shapes are displayed in turn. Each display is held on screen for approximately 10 seconds, and then the next mode shape is shown. In this way the data plays like a three-dimensional slide show. The figure below shows a sample mode shape for a catenary riser. The Model View allows you full freedom to examine the model and its mode shapes in three dimensions by zooming, rotating and panning in the usual way.



Sample Mode Shape Display

Modal Displacement and Curvature

THEORY

In addition to the 3D display of mode shapes, Modes also provides a facility to produce plots of modal displacement and curvature. When you postprocess Modes results, you have the option of generating plot files for all or selected modes calculated by the program, and you can choose between modal displacements or curvatures. Both are plotted against distance along the model. The plots, regardless of the type selected, are produced by default for all elements, but you also have the option to nominate a plotting element set. Note that the option to plot modal displacements and curvatures is only appropriate when SHEAR7 output is requested.

RELEVANT KEYWORDS

- `*PLOT` is used to specify selected modes for plotting, and the type of plot required.
- `*RISER TYPE` is used to specify the riser type and other parameters relating to the Modes facility for generating output for subsequent input to SHEAR7.

1.9.5.9 Clearance & Interference Postprocessing

OVERVIEW

Flexcom possesses an ancillary module, Clear, which is a clearance/interference postprocessor. It produces tabular and graphical output regarding the interference or clearance between adjacent structures, from the results of Flexcom analyses of the structures under consideration. Typical applications would include the prediction of interference occurring between flexible risers and vessel mooring lines on a floating production system, or between rigid drilling and production risers on a TLP.

The use of Clear can be briefly summarised as follows. As a first step, you create Flexcom models of the structures under consideration. These models can be combined in a single Flexcom input file, or you can analyse the structures separately, in a series of Flexcom runs, but with the structures located in their actual positions relative to one another.

Once the Flexcom analysis or analyses are completed, you then run Clear. The major input parameters are (i) the regions on both structures over which interference is thought likely to occur, (ii) the required output format (you can request snapshot, timetrace or statistics output), and (iii) miscellaneous parameters and titles (most of which are optional). Clear performs its calculations, as described in the next section, and creates two output files. These are (i) a tabular output file summarising the analysis results, and (ii) a plot file that can subsequently be viewed or printed using the plotting facility.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Operation
- Analysis Type
- Minimum Distance
- External Diameter
- Database Name

RELEVANT KEYWORDS

- Numerous keywords are contained in the $CLEAR section, which allows you to perform clearance/interference postprocessing calculations.

If you would like to see an example of how these keywords are used in practice, refer to G02 - Jumper Wake Interference.

Operation

THEORY

The following is a summary of how Clear calculates minimum clearance values between two structures. The process begins with you, the user, identifying the region over which interference can occur by defining two element sets, one on each of the structures under consideration. In addition, you also specify the value of a parameter named Number of Points per Element. This determines the number of points at which the program actually performs interference calculations, as follows.

At a particular analysis time, Clear begins at the first point on the first element of the first element set, and calculates the distance between this point and every point of the second set. The minimum predicted distance is stored for subsequent processing, and Clear moves to the second point on the first element of the first element set. This procedure is repeated for all points on all elements of the first element set.

It is important to realise from this that Clear considers that interference can occur between any two points in the two element sets that you identify. You should be careful in defining interference regions to specify reasonable data to prevent excessive runtimes, by ensuring the program is not checking interference between points which will never approach. It is better to perform a series of Clear runs for a succession of smaller interference regions, rather than try to do everything in one run.

RELEVANT KEYWORDS

- *ELEMENT SETS is used to specify data relating to the element sets on both structures between which Clear is to perform clearance calculations.

Analysis Type

THEORY

Clear can perform three types of interference analysis, all of course using the same basic procedure just outlined. In the first of these, the program produces a snapshot of minimum distance between structures, in the second a timetrace of minimum distance is produced, while the third analysis type calculates statistics of minimum distance. Each of these is now further described.

Snapshot

A Clear snapshot is just that, a snapshot of the minimum distance between two structures at a particular time you specify. Specifically, in a snapshot analysis Clear calculates for each point on the first element set the minimum distance between it and all of the points on the second element set. This is then plotted against distance along the first element set. A Clear snapshot analysis is normally used when postprocessing a Flexcom static run.

Timetrace

A Clear timetrace analysis on the other hand is normally used to postprocess the results of a Flexcom dynamic run. The output in this case is a time history of the minimum predicted distance between the two element sets. Specifically, at each analysis time Clear calculates the distance between each pair of points on the two element sets. The minimum of all of these distances is then the Clear output at that time.

Statistics

A Clear timetrace analysis can be used with both regular wave and random sea Flexcom dynamic runs. However, for the random sea case the actual minimum distances may be of less interest than the statistics of minimum distance, and that is the rationale for the third Clear analysis type, the statistics analysis.

The procedure in a statistics analysis is as follows. For each point on the first element set, Clear calculates time histories of distance between it and all of the points on the second element set. From each of these time histories the program then calculates the mean value and the standard deviation of distance between the points.

The mean value and standard deviation are conveniently combined in a single quantity referred to here as the non-dimensional clearance. This is simply the ratio of the two parameters, or

$$\text{Non-dimensional clearance} = \frac{\text{Mean distance}}{\text{Standard deviation of distance}}$$

The non-dimensional clearance can be used to determine the likelihood or probability of interference occurring between two points. The smaller the value of non-dimensional clearance, the more likely is interference between them, because the mean distance between the two points is small in comparison to their relative motions. So in a Clear statistics analysis, the program calculates for each point on the first element set the minimum non-dimensional clearance between it and all of the points on the second element set. This is then plotted against distance along the first element set (the same format as used in a snapshot plot). The output file in this case lists the minimum non-dimensional clearances together with the corresponding mean values and standard deviations.

RELEVANT KEYWORDS

- `*ANALYSIS TYPE` is used to specify the Clear analysis type and related parameters.

Minimum Distance

THEORY

You may optionally specify a Minimum Distance input for a Clear timetrace analysis. This is a separation value below which you want Clear to consider that interference has occurred when counting the number of times the two structures interfere. This defaults to a value of zero, in which case interference is deemed to occur only when the two structures are actually predicted by Clear as being in contact. The rationale for this entry is as follows. Clear calculates the separation or clearance between structures at discrete times. If two structures do indeed come intermittently into contact, the likelihood of this occurring at a Clear solution time is small. For this reason you specify the Minimum Distance value. The assumption is that if the distance between the two structures at a particular solution time is less than this minimum, then it is likely that they do come into contact between Clear solution times. In this case the program should increment its running total of contacts or interferences, and if possible calculate impact velocities and accelerations.

RELEVANT KEYWORDS

- *ANALYSIS TYPE is used to specify the Clear analysis type and related parameters, including the Minimum Distance input.

External Diameter

THEORY

Obviously Clear takes into account the diameter of the two structures when calculating the distance between them at any pair of points. You may optionally specify an effective External Diameter for the elements of each element set for use in these calculations. If you do not input a value for an element set, Clear uses for each individual element the effective external diameter for the element stored in the Flexcom database. This will be either the drag diameter if geometric properties were specified in the Flexible Riser format, or the external diameter if the Rigid Riser format was used. If, however, you specified a diameter to be used in Stress Computations for timetrace output, this will be used as the effective external diameter. Put simply, the diameter which Clear uses is the diameter the main Flexcom analysis module would use if you requested timetrace output of bending, hoop or von Mises stresses.

RELEVANT KEYWORDS

- *ELEMENT SETS ($CLEAR) is used to specify data relating to the element sets on both structures between which Clear is to perform clearance calculations, including external diameter.
- *GEOMETRIC SETS ($MODEL) is used to assign geometric properties to element sets, including external diameter.
- *PROPERTIES ($MODEL) is used to assign effective structural properties to element sets for use in calculating stresses, including external diameter.

Database Name

THEORY

You may optionally use the Database Name input to select the Flexcom database output file(s) from which Clear is to retrieve the positions or motions of the elements of the set. Different file names can be specified for each element set, for the case where the two structures under consideration are analysed in separate Flexcom models. When the two structures are analysed in a single Flexcom model, you input the same name for both sets. This is in fact the default — when you input a file name for the first element set, the database name for the second set defaults to this name. So there is no need to input the database name twice if the two structures are in a single Flexcom model. Where this is not the case, you simply input a separate name for each database. You can also leave the Database Name inputs blank. When you do this, Clear assumes that the two structures are in a single Flexcom model, and that the database file name is the same as the generic name of the Clear run. There is one final point to note about the case where Clear is reading from two databases rather than one. When two structures are analysed individually, the likelihood if one or both analysis used a variable timestep is that the results are at different solution times. In this case Clear must interpolate between results at different solution times in order to perform meaningful calculations. The program does this automatically and without user intervention, but it should be borne in mind that this can occasionally increase the Clear runtimes significantly compared to the case where the two sets of results are in a single database. This is because of the increased number of times Clear must read database output.

RELEVANT KEYWORDS

- `*DATABASE` is used to specify the names of the Flexcom database files on which the Clear analysis is to be based.

1.9.5.10 Code Checking

OVERVIEW

Flexcom allows you to check analysis results against specific design codes/procedures. The codes currently supported are as follows:

- DNV-OS-F101 (Submarine Pipeline Systems), October 2013
- DNV-OS-F201 (Dynamic Risers), October 2010
- API-STD-2RD (Dynamic Risers for Floating Production Systems), September 2013

- ISO-13628-7 (Completion/workover riser systems), November 2005

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

- Inputs
- Operation
- Output

RELEVANT KEYWORDS

- Numerous keywords are contained in the `$CODE CHECKING` section, which allows you to check analysis results against specific design codes.

If you would like to see an example of how these keywords are used in practice, refer to K02 - Worked Example - Complex.

Inputs

THEORY

Inputs are grouped together under similar categories, such as Environmental, Section Properties, Material Properties, Load Factors, Resistance Factors and General. The various inputs and their significance are now outlined.

Code

The codes currently supported are as follows:

- DNV-OS-F101 (Submarine Pipeline Systems), October 2013
- DNV-OS-F201 (Dynamic Risers), October 2010
- API-STD-2RD (Dynamic Risers for Floating Production Systems), September 2013
- ISO-13628-7 (Completion/workover riser systems), November 2005

Environmental

- Environmental File Name. An environmental load is considered to be a load imposed directly or indirectly by the environment. This allows you to specify a list of Flexcom analyses to be included in the code checking calculations. Note that you use this functionality to nominate dynamic analysis keyword files, which are treated as the Environmental load case in terms of the DNV and API code checks.

- Functional File Name. A functional load is a load that is a consequence of the system's existence and use without consideration of environmental effects. A dynamic analysis is typically preceded by one or more static runs (initial static, offset, current etc.). This input may be used to specify which static run to treat as the Functional load case. If the functional file is not specified it defaults to the analysis immediately preceding the environmental analysis in the restart chain.

- Start Time. This input may be used to exclude the effects of initial solution transients from the code checking calculations.

- End Time. This input, in conjunction with the Start Time, allows you to focus on a particular time segment (e.g. for efficiency reasons) if so desired.

- Set Name. This allows you to specify a list of element sets to be considered for code checking. By default all elements in the designated Flexcom analysis are considered.

- Design Factor. This allows you to specify a numerical factor to be considered for API or ISO code checking. For API code checking, the design format sets limits on loads and load effects to a fraction of the capacity of the component to resist the load or load effect with this factor. The design factor depends on the type of load and can be different for SLS, ULS, and ALS categories. For ISO code checking, the design factor depends on the failure mode and design conditions. Please refer to the ISO code document to select the most suitable value. An appropriate value should be specified per environmental. If not specified, this value defaults to 0.6. This value is not used if specified for the DNV code check, or Method 3 of the API code check.

- Collate. This option allows you to make a summary file of your code check and collates utilizations for the environmental(s) in your output file. By default, this value is set to YES.

Section Properties

- Set Name. The name of the section (i.e. the element set name) whose properties are being defined. Note that any Set Name referenced under the Environmental category must have section properties defined here.
- Material Name. The name of the material associated with the section. The actual material properties themselves are defined under the Material category.
- Alpha C. The flow stress parameter accounting for strain hardening (α_c) as per the relevant DNV standard. This value is not used if specified for the ISO code check.
- Diameter and Thickness. The section diameter and wall thickness. If unspecified, these parameters default to the corresponding values used in the Flexcom analysis.
- T_{corr}. Corrosion allowance (T_{corr}) as per the relevant DNV standard. This value defaults to 0.0 if not specified.
- Alpha Fab. Manufacturing process reduction factor (α_{fab}) as per the relevant DNV standard. This value defaults to 0.85 if not specified. This value is not used if specified for the ISO code check.
- Ovality. The section ovality (f_o) as per the relevant DNV standard. This value defaults to 0.005 if not specified.
- Scale Factors. Moment and tension scale factors. If unspecified, these parameters default to 1.0. This value is not used if specified for the ISO code check.

Standard Material Properties

The standard material properties are used for DNV and API code checks.

- Material Name. The name of the material whose properties are being defined. Note that any Material Name referenced under the Section Properties category must have material properties defined here.
- SMYS and SMTS. The Specified Minimum Yield Strength and Specified Minimum Tensile Strength respectively for the material.
- Young’s Modulus and Poisson’s Ratio. Self-explanatory.

- Fy Temp and Fu Temp. The temperature derating factors for the yield stress ($F_{y,temp}$) and tensile stress ($F_{u,temp}$) respectively, as per the relevant DNV standard.
- Alpha U. The material strength factor (α_u) as per the relevant DNV standard. This value defaults to 0.96 if not specified.

ISO Material Properties

The ISO material properties are used for ISO code checks.

- Material Name. The name of the material whose properties are being defined. Note that any Material Name referenced under the Section Properties category must have material properties defined here.
- Rt05. Specified minimum yield strength for 0.5% total elongation at room temperature.
- Rm. Specified minimum ultimate tensile strength at room temperature.
- Elevated Temperature. Used to determine the reduction factor due to temperature.
- Young’s modulus and Poisson’s ratio. Self-explanatory.
- A5. Percentage minimum elongation after fracture.
- Tc. Single load ultimate tension capacity.
- Mc. Single load ultimate bending capacity.
- Pec. Single load ultimate pressure capacity.

Tc, Mc and Pec are used to calculate connector resistance at the start of every section.

Load Factors

The following factors may be applied to API and DNV code checking and are not applied to ISO code checking.

- Functional. The functional factor (γ_f) as per the relevant DNV standard. This value defaults to 1.1 if not specified.
- Environmental. The environmental factor (γ_e) as per the relevant DNV standard. This value defaults to 1.3 if not specified.

- Optimise Factors. An option to optimise the load factors to maximise the load effects. The default value of Yes means that the environmental factor is taken as 1/γe if the environmental loading reduces the combined load effects.

Resistance Factors

The following factors may be applied to API and DNV code checking and are not applied to ISO code checking.

- Material. The material resistance factor (γ_m) as per the relevant DNV standard. This value defaults to 1.15 if not specified.
- Safety. The safety class resistance factor (γ_{sc}) as per the relevant DNV standard. This value defaults to 1.26 if not specified.

General

The following factor may be applied to API and DNV code checking and is not applied to ISO code checking.

- Usage Factor. The Working Stress Design usage (WSD) as per the relevant DNV standard. This value defaults to 0.75 if not specified.

Reference Pressure

The reference pressure may be applied to ISO code checking and is not applied to API or DNV code checking.

- Element. The element at which the local internal pressure is to be defined. The reference point is the midpoint of this element.
- Pressure. The local internal pressure from which the internal pressure at the reference point is calculated as per the *"Design pressure and design temperature"* subsection in the ISO-13628-7 code document.

RELEVANT KEYWORDS

Code

- *CODE is used to specify the relevant code/procedure to be used in code checking.

Environmental

- *ENVIRONMENTAL is used to specify a list of Flexcom analyses to be included in the code checking calculations.

Section Properties

- *SECTION PROPERTIES is used to specify section properties to be used in code checking.
- *ELEMENT SETS is used to group individual elements into element sets.

Material Properties

- *MATERIAL is used to specify standard material properties to be used in DNV and API code checking.
- *MATERIAL-ISO is used to specify ISO material properties to be used in ISO code checking.

Load Factors

- *LOAD FACTOR is used to specify load effect factors to be used in code checking.

Resistance Factors

- *RESISTANCE FACTOR is used to specify resistance factors to be used in code checking.

General

- *GENERAL is used to specify a general usage factor and an incidental to design pressure ratio to be used in code checking.

Reference Pressure

- *REFERENCE PRESSURE is used for ISO code checking only to specify the local internal pressure at a particular location.

If you would like to see an example of how these keywords are used in practice, refer to K02 - Worked Example - Complex.

Operation

Checking analysis results against specific design codes/procedures in Flexcom is performed in the following manner:

1. First, select the code type that you wish to check your analyses against by defining it in the `*CODE` keyword.

2. Choose the environmental file(s) (i.e. the dynamic analyses) to be checked by defining them in the `*ENVIRONMENTAL` keyword.

3. If an environmental file is related to more than one static analysis (e.g. initial static, vessel offset, current etc.), you may wish to explicitly specify which static analysis represents the functional analysis. Unless specified otherwise, the analysis immediately preceding the environmental analysis is assumed to be the functional analysis.

4. If there are element sets of interest that have not been defined in any of the dependent analysis files, you may define additional sets in the `*ELEMENT SETS` keyword.

5. Create material properties that will be applied to the sections in the `*MATERIAL` keyword. If you are running an ISO code check you must create a material using the `*MATERIAL-ISO` keyword.

6. Create the sections that the code checking will be performed on via the `*SECTION PROPERTIES` keyword. These are characterised in terms of element sets and material properties defined in previous steps.

7. Optionally specify additional parameters in the `*LOAD FACTOR`, `*RESISTANCE FACTOR` and `*GENERAL` keywords.

8. Timetrace plots of the following can be requested in `*TIMETRACE` keyword.

   - LRFD Load and Resistance Factor Design
   - WSD Working Stress Design
   - VMS Von Mises Stress
   - METHOD1 API-STD-2RD: Utilization of combined membrane load
   - METHOD2 API-STD-2RD: Utilization of axial load based on yield tension

- METHOD3 DNV-OS-F201: LRFD Utilization
- METHOD4a API-STD-2RD: Utilization of combined pressure and axial load
- METHOD4b API-STD-2RD: Utilization of bending strain
- CLD ISO: Combined Load Design
- CR ISO: Connector Resistance

9. You are now ready to run the `$CODE CHECKING` postprocessing analysis.

Output

THEORY

For each of the analyses listed under the Environmental category, Flexcom creates some standard text and graphical outputs to present the results of the code checking analysis.

Two standard text files are generated automatically, which have the file extensions .sud and .suf, respectively. The .sud file is the main output file, and it is comprised of three main sections.

- Firstly a header block presents general QA information
- The main body of data consists of data output for each location (i.e. element, local node) considered in the code checking analysis. There is a range of outputs provided. For DNV the most important ones are the maximum LRFD, WSD usage and VMS values calculated and the times at which each of these occur. Similarly, the utilizations for each method in API are presented when the API code check is used. For ISO, the CLD, CR and VMS are output.
- A summary of pertinent results is provided towards the end of the file

The data in the .suf file is provided mostly for verification or debug purposes, and consists of the actual tensions and moments (functional and environmental) used in calculating the maximum utilziations. For DNV, the values used to calculate LRFD values at each location are outputted. For API, the values used to calculate Method 1, 2, 3 and 4 utilizations are outputted separately.

In terms of graphical output, for the DNV code checks, three standard plot files are generated automatically, presenting the maximum LRFD, WSD and VMS for each every element set considered for code checking (i.e. the sets listed under the Environmental category). The API code check is similar, however a plot is generated for each of the four methods and a summary plot with overlaid utilizations is also presented. For the ISO code check a single plot that contains both the CLD and the CR at the section start location is generated.

If the collation option is turned on a table of summary of relevant maximum values will be included in the output file.

You may optionally request additional timetrace plots if you wish, in order to examine the actual variation of LRFD, WSD, VMS, METHOD1, METHOD2, METHOD3, METHOD4a, METHOD4b, CLR or CR at a particular location as a function of time.

RELEVANT KEYWORDS

- `*ENVIRONMENTAL` is used to specify a list of Flexcom analyses to be included in the code checking calculations. The collation option is also specified under this heading.
- `*TIMETRACE` is used to request the creation of timetrace plots.

1.9.5.11 Force and Stress Outputs

OVERVIEW

This section describes the calculation of various force and stress outputs provided by Flexcom. Firstly it summarises the significance of some commonly used Flexcom keyword inputs. Sign conventions used in Flexcom for bending moment, shear force and torque are then outlined. The computation of axial force and effective tension are then discussed, followed by a summary of the procedures for calculating the various stress and strain outputs produced by Flexcom.

FURTHER INFORMATION

Further information on this topic is contained in the following sections:

Significance of Flexcom Keyword Inputs

- Force Variable Input

- Location Parameter Input

Sign Conventions

- Bending Moment Sign Convention
- Shear Force Sign Convention
- Torque Sign Convention
- Axial Force an Effective Tension

Stress and Strain Computations

- Bending Stress
- Bending Strain
- Planar Curvature
- Hoop Stress
- Axial Stress
- Von Mises Stress (Standard Method)
- Von Mises Stresses (API-2RD Method)
- Axial Strain
- Longitudinal Strain
- Longitudinal Stress
- Plastic Axial Strain
- Equivalent Plastic Axial Strain
- Plastic Local-Y & Local-Z Curvatures
- Equivalent Plastic Curvature

RELEVANT KEYWORDS

Flexcom provides a variety of postprocessing options, and the primary channels are as follows.

- Database Postprocessing generally represents the most comprehensive postprocessing resource. Numerous keywords are contained in the `$DATABASE POSTPROCESSING` section, each of which is capable of generating Graphical Output, Tabular Output and Spreadsheet Output.
- Summary Postprocessing allows you to extract pertinent results (e.g. maximum/minimum values) from time domain analyses in succinct tabular format. Numerous keywords are contained in the `$SUMMARY POSTPROCESSING` section, which allow you to control the contents of the Summary Output File. Although summary output is predominately used to create succinct tabular output, it is also capable of generating Graphical Output in the form of plot files.
- Summary Postprocessing Collation allows you to collate the summary postprocessing results across a range of different time domain analyses. Numerous keywords are contained in the `$SUMMARY COLLATE` section, which allow you to control the contents of the Summary Collation Spreadsheet.

Local Node Input

INTRODUCTION

Many of the postprocessing keywords accept an integer value (typically denoted as *Local Node*) which allows you to specify the location of interest along an element. The relevant keywords are are follows:

`$DATABASE POSTPROCESSING`

- `*RAO`
- `*SPECTRUM`
- `*TIMETRACE`

`$LOAD CASE`

- *TIMETRACE

$SUMMARY POSTPROCESSING

- *PARA FORCE
- *PARA ANGLE TENSION

LOCAL NODE

The significance of the *Local Node* input depends on whether element based outputs are stored on a node or integration point basis.

- Flexcom has traditionally provided output based on nodes. This operation remains the default and is still used in the vast majority of cases.
- Integration point output is a relatively new addition to the program, and although it is only possible to present output for elements with exactly 3 integration points, this functionality will be extended in a future version to accommodate elements with up to 10 integration points, providing higher resolution of element based outputs.

Local Node	Node Based Output	Integration Point Based Output
1	First Node	First Integration Point
2	Element Midpoint	Second Integration Point
3	Second Node	Third Integration Point

Significance of Local Node Input

Force Variable Input

INTRODUCTION

Many of the postprocessing keywords accept an integer value (typically denoted as *Variable*) which allows you to specify the parameter of interest, such as effective tension or bending moment. The relevant keywords are are follows:

`$DATABASE POSTPROCESSING`

- `*RAO`
- `*SNAPSHOT`
- `*SPECTRUM`
- `*STATISTICS`
- `*TIMETRACE`

`$LOAD CASE`

- `*TIMETRACE`

`$SUMMARY POSTPROCESSING`

- `*PARA FORCE`
- `*PARA FORCE ENVELOPE`

FORCE VARIABLE

The input parameter is known as a force variable, and the following *Variable* values are valid.

1. Axial Force
2. Local-y Shear
3. Local-z Shear
4. Torque

5. Local-y Moment
6. Local-z Moment
7. Effective Tension
8. Resultant Bending Moment
9. Resultant Curvature
10. Resultant Shear
11. Axial Stress
12. Bending Stress
13. von Mises Stress
14. Hoop Stress
15. Bend Radius
16. Local-y Curvature
17. Local-z Curvature
18. von Mises Stress (API-2RD)
19. Bending Strain
20. Axial Strain
21. Temperature
22. Pressure
23. Damper Power. Damper Elements are sometimes used to simulate power extraction. In this context, Flexcom provides post-processing options for damper power.
24. Plastic Axial Strain
25. Plastic Local-Y Curvature

26. Plastic Local-Z Curvature
27. Equivalent Plastic Axial Strain
28. Equivalent Plastic Curvature
29. Plastic Resultant Curvature
30. Plastic Bending Radius
31. Plastic Bending Strain
32. Planar Curvature
33. Plastic Longitudinal Strain
34. Longitudinal Strain
35. Longitudinal Stress

Location Parameter Input

The location parameter input is appropriate when you request an output of bending stress, bending strain, longitudinal stress, longitudinal strain, planar curvature, von Mises stress, or pressure during postprocessing. What the location parameter value represents depends on the respective force type as described in the following subsections.

PRESSURE

If you are postprocessing for *Pressure*, the *Location Parameter* can have a value of 1 or 2, where 1 indicates internal pressure and 2 indicates external pressure. If you do not explicitly specify a location, a value of 1 is assumed.

PLANAR CURVATURE

If you are postprocessing for *Planar Curvature*, the *Location Parameter* can have an integer value between 0 and 360 which represents the angle in degrees at which to calculate the planar curvature. If you do not explicitly specify a location, a value of 0 is assumed.

BENDING STRESS, BENDING STRAIN, LONGITUDINAL STRESS, LONGITUDINAL STRAIN AND VON MISES STRESS

If you are postprocessing for *Bending Stress, Bending Strain, Longitudinal Stress, Longitudinal Strain* or *Von Mises Stress*, the *Location Parameter* is an integer value, n, between 1 and 16, which defines the location on the cross section at which stresses/strains are to be calculated at all solution times. For statistical outputs, the output corresponds to the maximum value occurring on the cross-section at each solution time, although in general the location where this occurs can differ from solution time to solution time. This is the default computation method if you do not explicitly specify a *Location Parameter*.

The *Location Parameter*, n, corresponds to an angle θ, measured in degrees anti-clockwise from the local element cross-section y-axis. The table below indicates the variation of parameter n with θ.

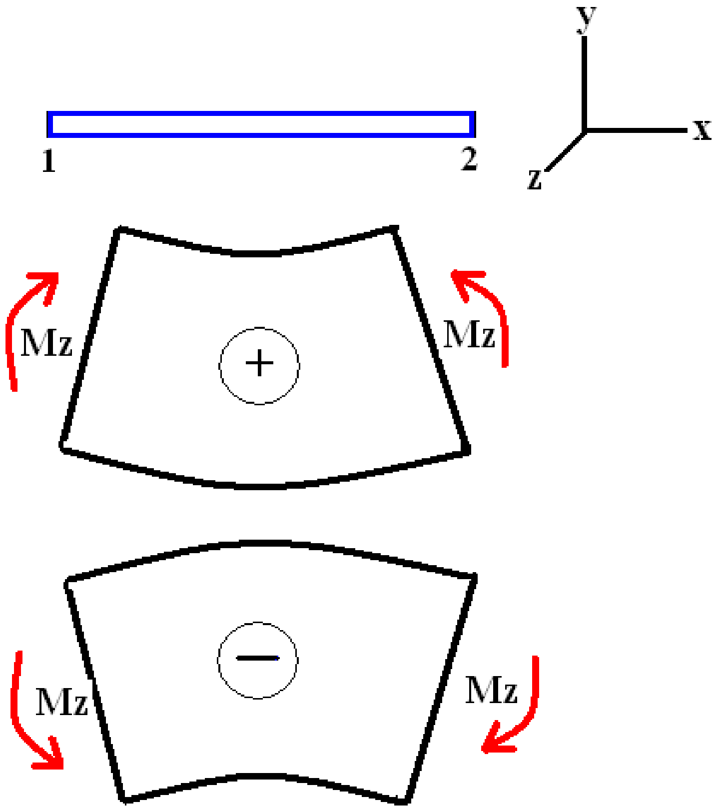
θ°	N		
	Bending Stress/Strain **Longitudinal Stress/Strain**	**Von Mises Stress (Inner Surface)**	**Von Mises Stress (Outer Surface)**
0	1	1	9
45	2	2	10
90	3	3	11
135	4	4	12
180	5	5	13

225	6	6	14
270	7	7	15
315	8	8	16

Since maximum bending stress/strain or longitudinal stress/strain must occur on the riser outer surface, only values for n between 1 and 8 are required (or valid). Longitudinal Strain is available at eight locations around the circumference, just like the bending strain. The longitudinal stress is available at eight locations around the circumference, but not as a maximum value. For von Mises stress, values between 1 and 16 are valid.

Bending Moment Sign Convention

In general, a bending moment is positive if it tends to bend a beam element section concave facing upward with respect to the local axis. This is shown in the figure below. A positive bending moment about the local z axis will put the upper surface of the beam (positive y) into compression and the lower surface into tension. Note that the local x axis for the element points from the first node to the second. In other words, a positive bending moment is one which causes sagging with respect to the local axis system.



Bending Moment Sign Convention

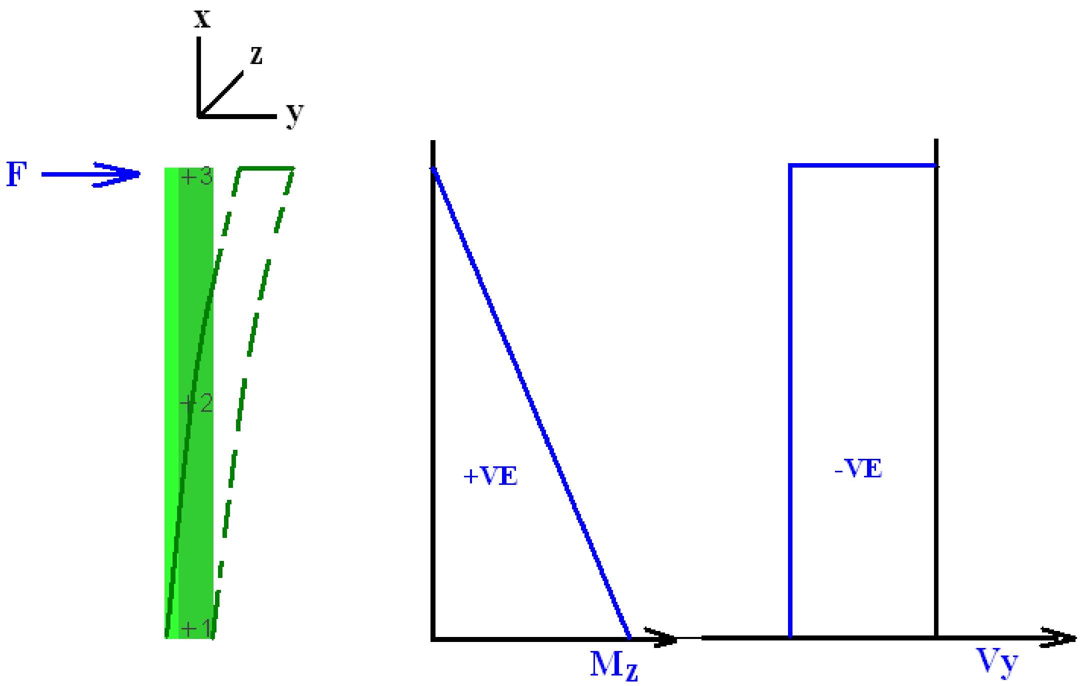
This convention may also be illustrated with a simple example. A model consisting of two elements is shown in the Cantilever Beam with Bending Moment and Shear Force Distributions figure, along with the local axis system. The system modelled (a simple cantilever structure) is restrained in all degrees of freedom at the bottom node (Node 1), and a force F is applied in the positive sense of the y axis at the top. Such a force will cause compression in the right hand side of the elements (positive y) and tension at the left hand side. This defines a positive bending moment about the local z axis.

Shear Force Sign Convention

The sign for the shear force (V) is found using the standard definition:

$$V = \frac{dM}{dx}$$

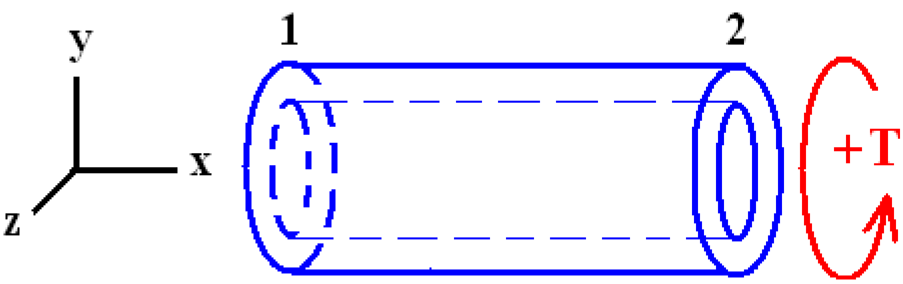
So the shear force distribution for the simple cantilever is a constant negative value as the bending moment distribution is linear with a negative slope.



Cantilever Beam with Bending Moment and Shear Force Distributions

Torque Sign Convention

The sign convention for element torque forces follows the right hand rule, and is illustrated in the below figure. A torque force is positive if it is in the clockwise direction when viewed from the origin towards the positive sense of the local x axis. Again, the local x axis points from the first node to the second.



Torque Sign Convention

Axial Force and Effective Tension

THEORY

The relationship between axial force and effective tension is expressed as follows:

$$T_e = N - \left(P_i A_i - P_o A_o\right) \quad (1)$$

where:

- T_e is effective tension
- N is axial force
- P_i is internal pressure
- A_i is the corresponding internal area
- P_o is external pressure
- A_o is the corresponding external area

In Flexcom, the axial force in an element is a solution variable, and the effective tension T_e is calculated from N using the above relationship. Note that A_o is calculated from the buoyancy diameter Db you specified as part of the Geometric Properties definition.

In some analyses the choice of D_b for input to Flexcom means that the axial force calculated by the program is not a true reflection of the force in the actual wall structure. Obviously, the choice of buoyancy diameter will be governed by the need to ensure that effective tensions in the model are everywhere correct.

If you define an effective external diameter D_o to be used in Stress Computations, and if this value is not equal to the buoyancy diameter D_b you specified as part of the geometric properties definition, then Flexcom recalculates the axial force in the section on the basis of the effective diameter D_o. Specifically, the axial force is calculated using:

$$N = T_e + \left(P_i A_i - P_o A_o^*\right) \quad (2)$$

where P_i , P_o and A_i are as previously, but A_o^* is now calculated from the effective external diameter using:

$$A_o^* = \frac{\pi D_o^2}{4} \qquad (3)$$

Note that this recalculation of axial force takes place automatically whenever you specify a D_o that is not equal to D_b. Note also that if you request a plot of axial stress, this will naturally be computed using the recalculated axial force.

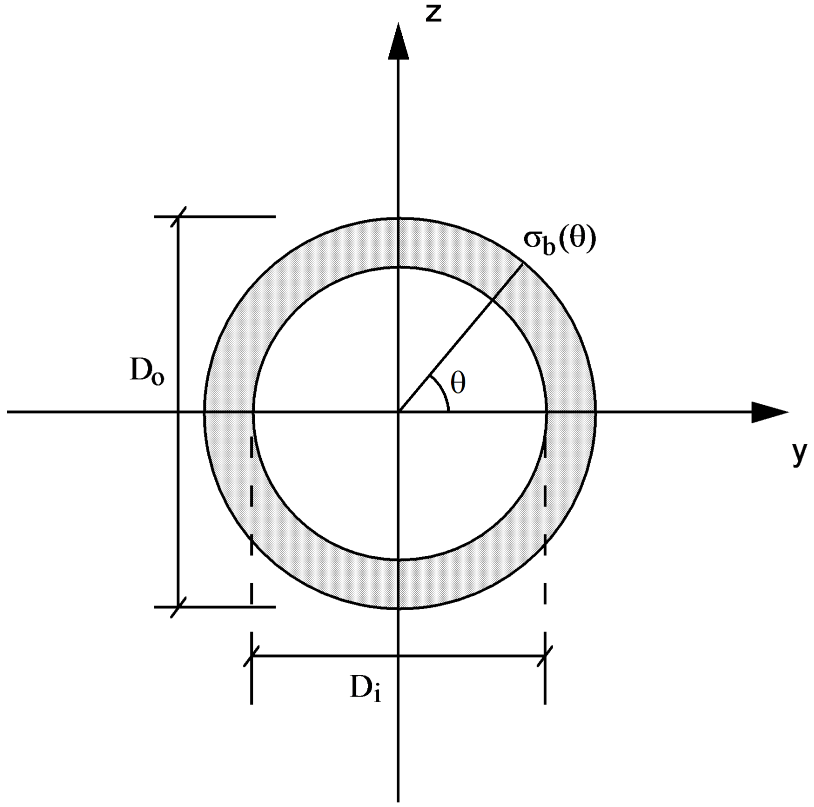
RELEVANT KEYWORDS

- *GEOMETRIC SETS ($MODEL) is used to assign geometric properties to element sets, including buoyancy diameter Db
- *PROPERTIES ($MODEL) and *PROPERTIES ($DATABASE POSTPROCESSING) are both used to assign effective structural properties to element sets for use in calculating stresses, including external diameter Do.

Bending Stress

Bending stress in an element cross section is defined by the Equation below, with reference to the Bending Stress Calculation figure below. Note that the maximum bending stress occurs at the outer surface of the cross section.

$$\sigma_b(\theta) = \frac{M_y D_o}{2 I_{yy}} \sin\theta + \frac{M_z D_o}{2 I_{zz}} \cos\theta \qquad (1)$$



Bending Stress Calculation

where:

- $\sigma_b(\theta)$ is the bending stress for an angle θ as defined in the figure above
- M_y and M_z are the bending moments about the local y- and z-axes, respectively
- I_{yy} and I_{zz} are the second moments of area about the local y- and z-axes, respectively
- D_o is the effective outer diameter

By default, Flexcom outputs the maximum bending stress occurring on the cross section. However, when you request a plot of bending stress, you can optionally specify a location around the element circumference where bending stress is to be computed. The location is specified in terms of an integer value (N) which corresponds to an angle θ, measured in degrees anti-clockwise from the local element cross-section y-axis. The table below indicates the variation of parameter N with θ for bending stress calculations. In this table, σ_B represents bending stress.

Location Parameter for Bending Stress Calculations

θ°	N
0	1
45	2
90	3
135	4
180	5
225	6
270	7
315	8

Since maximum bending stress must occur on the riser outer surface, only values for n between 1 and 8 are required (or valid).

If you do not explicitly specify a location N, Flexcom outputs the maximum bending stress occurring on the cross section. The maximum value of σ_b can be found from the Equation above by setting $\frac{d\sigma_b}{d\theta}=0$. The value of θ corresponding to the maximum bending stress is given by:

$$\theta=\tan^{-1}\left(\frac{M_y I_{zz}}{M_z I_{yy}}\right) \qquad (2)$$

When $I_{yy} = I_{zz}$, as is normally the case, the Equation directly above reduces to:

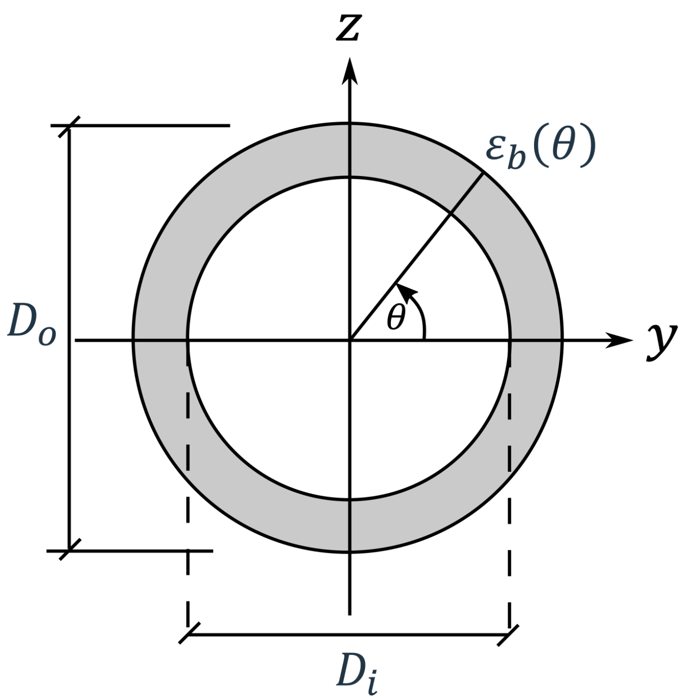
$$\theta=\tan^{-1}\left(\frac{M_y}{M_z}\right) \qquad (3)$$

Flexcom calculates maximum bending stress using the Equations above.

Bending Strain

Bending strain in an element cross section is defined by the Equation below, with reference to the Bending Strain Calculation figure below. Note that the maximum bending strain occurs at the outer surface of the cross section.

$$\varepsilon_b(\theta)=\frac{\kappa_y D_o}{2}\sin\theta+\frac{\kappa_z D_o}{2}\cos\theta \qquad (1)$$



Bending Strain Calculation

where:

- $\varepsilon_b(\theta)$ is the bending stress for an angle θ as defined in the figure above
- κ_y and κ_z are the curvatures about the local y- and z-axes, respectively
- D_o is the effective outer diameter

By default, Flexcom outputs the maximum bending strain occurring on the cross section. However, when you request a plot of bending strain, you can optionally specify a location around the element circumference where bending strain is to be computed. The location is specified in terms of an integer value (N) which corresponds to an angle θ, measured in degrees anti-clockwise from the local element cross-section y-axis. The table below indicates the variation of parameter N with θ for bending strain calculations.

Location Parameter for Bending Strain Calculations

θ°	N

0	1
45	2
90	3
135	4
180	5
225	6
270	7
315	8

Since maximum bending strain must occur on the riser outer surface, only values for n between 1 and 8 are required (or valid).

If you do not explicitly specify a location N, Flexcom outputs the maximum bending strain occurring on the cross section. The maximum value of ε_b can be found from the Equation above by setting $\frac{d\varepsilon_b}{d\theta}=0$. The value of θ corresponding to the maximum bending strain is given by:

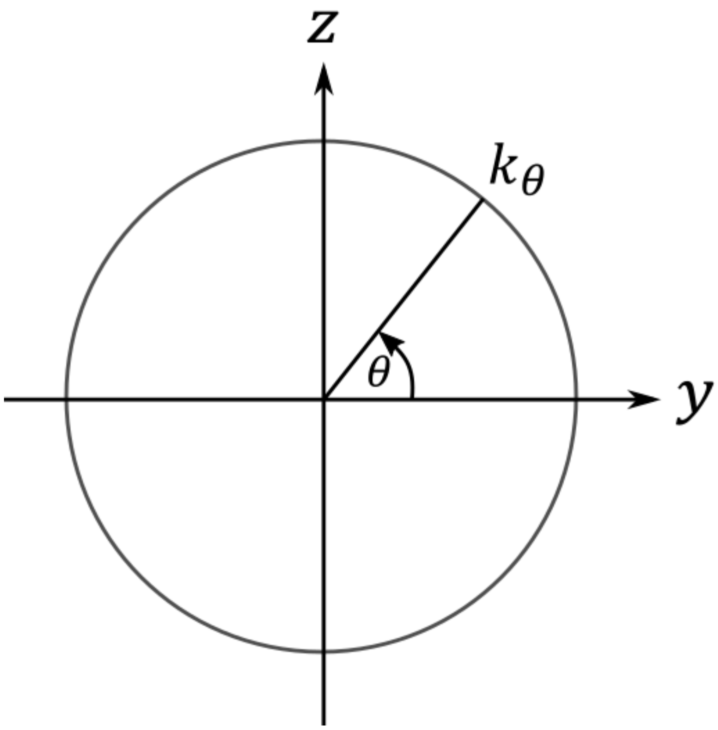
$$\theta = \tan^{-1}\left(\frac{\kappa_y}{\kappa_z}\right) \qquad (2)$$

Flexcom calculates maximum bending strain using the Equation above.

Planar Curvature

Planar curvature in an element cross section is defined by the Equation below, with reference to the Angle Convention for Planar Curvature figure below.

$$\kappa_\theta = \kappa_y \sin\theta + \kappa_z \cos\theta \quad (1)$$



Angle Convention for Planar Curvature

where:

- κ_θ is the bending stress for an angle θ as defined in the figure above
- κ_y and κ_z are the curvatures about the local y- and z-axes, respectively

By default, Flexcom outputs the planar curvature at an angle of 0 degrees. However, when you request a plot of planar curvature, you can optionally specify a location around the element circumference where planar curvature is to be computed. The location is specified as an angle θ, measured in degrees anti-clockwise from the local element cross-section y-axis.

Hoop Stress

Flexcom calculates and outputs hoop stress on the inside surface of the cross section using the relation:

$$\sigma_h = \frac{p_i\left(D_o^2 + D_i^2\right) - 2 p_o D_o^2}{D_o^2 - D_i^2} \qquad (1)$$

Here σ_h is hoop stress; D_i and D_o are the effective internal and outer diameters respectively; and p_i and p_o are respectively the internal and external pressures.

Axial Stress

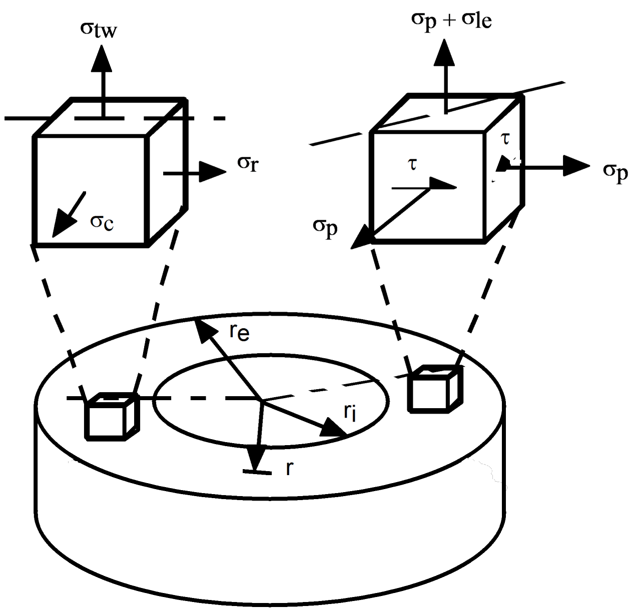
Axial stress, which is constant over the cross section, is calculated from:

$$\sigma_{tw} = \frac{N}{A} \qquad (1)$$

where σ_{tw} is the axial or "true wall" stress; N is axial force or "true wall tension"; and A is effective cross section area.

Von Mises Stress (Standard Method)

The standard or default formulation used by Flexcom in calculating von Mises stress is derived with reference to the figure below, which shows two equivalent representatives of the in-wall stresses induced by tension and internal and external pressures in a general thick walled riser section.



Riser In-Wall Stress System

In the representation on the left, σ_{tw} represents the "true wall" axial stress as defined in Axial Stress (Eq.1). σ_c is the circumferential or hoop stress, and σ_r is the radial stress. σ_{tw} is as noted constant over the section, but σ_c and σr are functions of r, the radius. However, the mean of the two stresses is everywhere constant, and alternative expressions for σc and σr may be written using Lamé's equations as follows:

$$\sigma_c = \sigma_p + \tau \quad \text{and} \quad \sigma_r = \sigma_p - \tau \qquad (1)$$

where:

$$\sigma_p = \frac{p_i D_i^2 - p_o D_o^2}{D_o^2 - D_i^2} \qquad (2)$$

and:

$$\tau = \frac{(p_i - p_o) D_i^2 D_o^2}{4r^2 (D_o^2 - D_i^2)} \qquad (3)$$

Here, p_i and p_o again represent internal and external pressure, and D_i and D_o are effective internal and outer diameters. σ_p is a constant hydrostatic stress and τ is a distortional shear stress, as shown in the representation on the right in the above Riser In-Wall Stress System figure. The effective stress σ_{le} is defined by:

$$\sigma_{le} = \frac{T_{eff}}{A} = \frac{N}{A} + \frac{D_o^2 p_o - D_i^2 p_i}{(D_o^2 - D_i^2)} = \sigma_{tw} - \sigma_p \qquad (4)$$

where T_{eff} is the effective tension. Equation (4) above can be rewritten as:

$$\sigma_{tw} = \sigma_p + \sigma_{le} \qquad (5)$$

as shown in the above Riser In-Wall Stress System figure. The effective stress is the amount by which the axial wall stress σ_{tw} exceeds the in-wall hydrostatic stress σ_p, and is analogous to the deviatoric term used in solid and soil mechanics.

The von Mises stress, σ_{vm}, for the triaxial system of the above Riser In-Wall Stress System figure is given by:

$$2\,\sigma_{vm}^2 = (\sigma_{tw} - \sigma_c)^2 + (\sigma_c - \sigma_r)^2 + (\sigma_r - \sigma_{tw})^2 \qquad (6)$$

Substituting from the Equations above gives an equivalent formulation:

$$\sigma_{vm}^2 = \sigma_{le}^2 + 3\tau^2 \qquad (7)$$

The von Mises stress is therefore closely related to the effective stress, rather than the "true wall" stress, as would be expected from the above Equation, since it is obvious that the hydrostatic stress σ_p can have no effect on the von Mises stress.

Of course, when the pipe of the above Riser In-Wall Stress System figure is subjected to bending in addition to tensile and pressure forces, the resultant bending stress (σ_b say) must be added to σ_{le} when calculating σ_{vm}. Therefore the Equation directly above becomes:

$$\sigma_{vm}^2 = (\sigma_{le} + \sigma_b)^2 + 3\tau^2 \qquad (8)$$

In this case, the von Mises stress must be checked on the pipe inner surface, where τ is a maximum, and on the outer surface, where σ_b is a maximum. For this reason Flexcom calculates σvm on both the inside and the outside surface to find the maximum value.

By default, Flexcom outputs the maximum von Mises stress occurring on the cross section. However, when you request a plot of von Mises stress, you can optionally specify a location around the element circumference where von Mises stress is to be computed. The location is specified in terms of an integer value (N) which corresponds to an angle θ, measured in degrees anti-clockwise from the local element cross-section y-axis. The table below indicates the variation of parameter N with θ for von Mises stress calculations. In this table, σ_{VMI} represents von Mises stress on the element inner surface, while σ_{VMO} represents von Mises stress on the element outer surface.

Location Parameter for von Mises Stress Calculations

θ°	N	
	σ_{VMI}	σ_{VMO}
0	1	9
45	2	10
90	3	11
135	4	12
180	5	13
225	6	14
270	7	15

315	8	16

Von Mises Stress (API-2RD Method)

The API-2RD procedure states that for plain round pipe, where transverse shear and torsion are negligible, the three principal stress components are σ_{pr} , $\sigma_{p\theta}$ and σ_{pz} , where r, θ, and z refer to radial, hoop, and axial stress. So σ_{vm} is defined as follows:

$$\sigma_{vm} = \frac{1}{\sqrt{2}}\sqrt{(\sigma_{pr} - \sigma_{p\theta})^2 + (\sigma_{p\theta} - \sigma_{pz})^2 + (\sigma_{pz} - \sigma_{pr})^2} \qquad (1)$$

Moreover, for a thick walled pipe, the radial, hoop, and axial stress may be defined as follows:

$$\sigma_{pr} = -\frac{(p_0 D_0 + p_i D_i)}{D_0 + D_i} \qquad (2)$$

$$\sigma_{p\theta} = (p_i - p_0)\left(\frac{D_o}{2t_{\min}}\right) - p \qquad (3)$$

$$\sigma_{pz} = \frac{N}{A} \pm \frac{M}{2I}(D_o - t) \qquad (4)$$

Here, t and t_{min} are the nominal and minimum wall thicknesses, respectively. t is defined as:

$$t = \frac{(D_0 - D_i)}{2} \qquad (5)$$

while t_{min} is the user-specified minimum wall thickness which defaults to t if omitted. M and I are the resultant bending moments and average second moments of area, respectively. M and I are defined as follows:

$$M = \sqrt{M_y^{\,2} + M_Z^{\,2}} \qquad (6)$$

$$I = \frac{I_{yy} + I_{ZZ}}{2} \qquad (7)$$

Axial Strain

The axial strain is calculated using the equation:

$$\varepsilon_a = \frac{1}{E}\left[\sigma_{tw} - \nu\left(\sigma_r + \sigma_c\right)\right] \qquad (1)$$

where E and ν are Young's modulus and Poisson's ratio, respectively. The stresses are defined in Axial Stress (Eq.1) and Von Mises Stress (Standard Method) (Eq.1). The sum of the radial and the hoop stresses is defined as:

$$\sigma_r + \sigma_c = 2\sigma_p \qquad (2)$$

Substituting Equation (2) into Equation (1), the axial strain equation becomes:

$$\varepsilon_a = \frac{1}{E}\left(\sigma_{tw} - 2\nu\sigma_p\right) \qquad (3)$$

The definition of σ_p from Von Mises Stress (Standard Method) can be written in terms of areas as:

$$\sigma_p = \frac{p_i A_i - p_o A_o^*}{A} \qquad (4)$$

Substituting Axial Stress (Eq.1) and Equation (4) into Equation (3) expression for the axial strain then becomes:

$$\varepsilon_a = \frac{1}{EA}\left[N - 2\nu\left(p_i A_i - p_o A_o^*\right)\right] \qquad (5)$$

Longitudinal Strain

The longitudinal strain is defined as the sum of the axial and bending strain in the cross-section. This output is available at eight locations around the circumference, just like the bending strain.

Longitudinal Stress

The longitudinal stress is defined as the stress caused by both axial and bending deformations. For a linear elastic, material this is simply the sum of the axial and bending stress. The longitudinal stress is an available output for all nonlinear materials where the stress is defined as function of strain (nonlinear rigid materials). The calculation takes into account the longitudinal strain and the selected material model. The longitudinal stress is available at eight locations around the circumference, but not as a maximum value.

Plasticity Related Ouputs

The primary outputs consist of axial strains and curvatures, such as:

1. Plastic Axial Strain and Equivalent Plastic Axial Strain

2. Plastic Local-Y and Local-Z Curvatures and Equivalent Plastic Curvature.

The primary outputs are computed during the analysis and stored into the force database file, as required.

The secondary outputs consist of:

1. Plastic Resultant Curvature - defined as the magnitude of the plastic local-y and local-z curvatures.

2. Plastic Bending Radius - defined as the inverse of the plastic resultant curvature.

3. Plastic Bending Strain - defined similarly to the bending strain, substituting plastic local-y and local-z curvatures for local-y and local-z curvatures.

4. Plastic Longitudinal Strain - defined as the difference between longitudinal strain and the ratio of longitudinal stress to Young's modulus (1)

$$\varepsilon^{pl} = \varepsilon - \frac{\sigma}{E} \tag{1}$$

1.9.5.12 Custom Postprocessing

Customised postprocessing is supported by Flexcom via the following methods.

- Flexcom is accompanied by a dedicated Excel Add-In which allows you to extract results from Flexcom database files directly into Excel.
- Specialised Database Access Routines are provided for advanced users of the software, who wish to directly access the contents of Flexcom's database files. The database access routines are a collection of procedures that facilitate retrieval of specific results directly from source. Typically code is written in FORTRAN or Visual C++, but any language can be used, as the code is ultimately compiled into a DLL (dynamic link library).
- The Flexcom Excel Add-In may also be used with VBA (Visual Basic Applications) code and the same functions are available as in an Excel worksheet. This is effectively a hybrid of the first two options, in that it requires some programming expertise, but the language used must be VBA, and the range of functions is limited to those provided as standard with the Excel Add-In.

The second and third options, in particular, facilitate the development of specialised postprocessing tools tailored to meet the specific requirements of individual customers.

Excel Add-in

The Excel Add-in Overview gives a general introduction to this tool. Information on using the Excel Add-in can be obtained through VBA. There is also important information contained in Formula Recalculation - Important Information.

Various Excel functions are logically grouped into the following similar catogories:

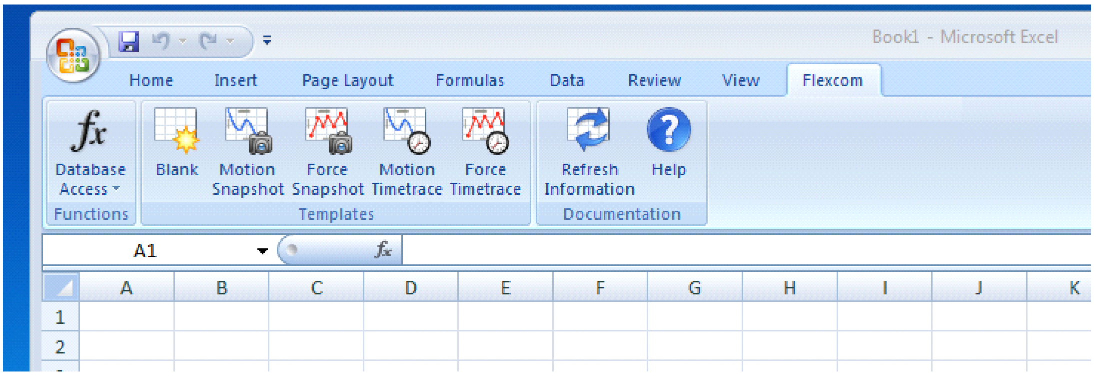
- General Functions
- Element Functions
- Label Functions
- Node Functions
- Set Functions

- Time Functions
- Force Functions
- Kinematic Functions

The Excel Add-in is compatible with both 32-bit and 64-bit editions of Microsoft Excel.

Overview

Flexcom is accompanied by a powerful Excel Add-in, which allows you to extract results from Flexcom database files directly in Excel. A range of standard functions and templates are provided and these are detailed in the following sections. When you install the Flexcom add-in for Excel, a tab is added to the Ribbon in Excel.



To familiarise yourself with the tool, it would be useful to load one of the four supplied templates from the ribbon and examine the formulas used throughout the sheet.

VBA

OVERVIEW

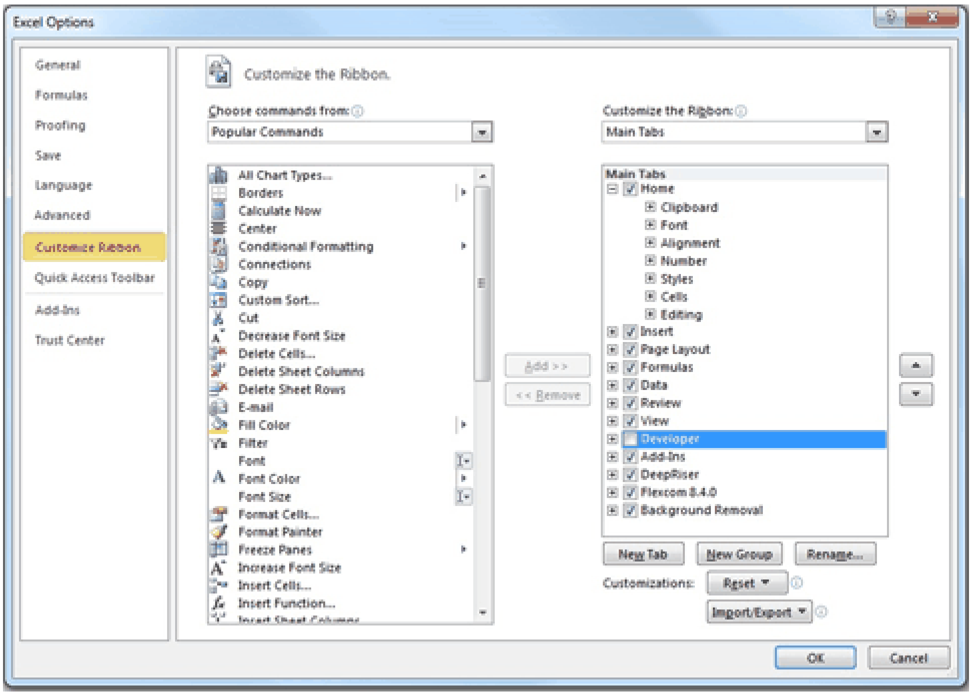
The Flexcom Excel add-in may also be used with VBA and the same functions are available as in an Excel worksheet. The following steps will help you to set up your VBA application to access a Flexcom database in Microsoft Excel 2010. The procedure should be similar for other versions of Excel.

EXCEL DEVELOPER TAB

1. The first step is to enable the Developer tab in Excel.

2. In Excel, click the File tab. On the Microsoft Office Backstage view that appears, click the Options button and then click the Customize Ribbon tab.

3. On the Main Tabs list on the right, select the Developer option.

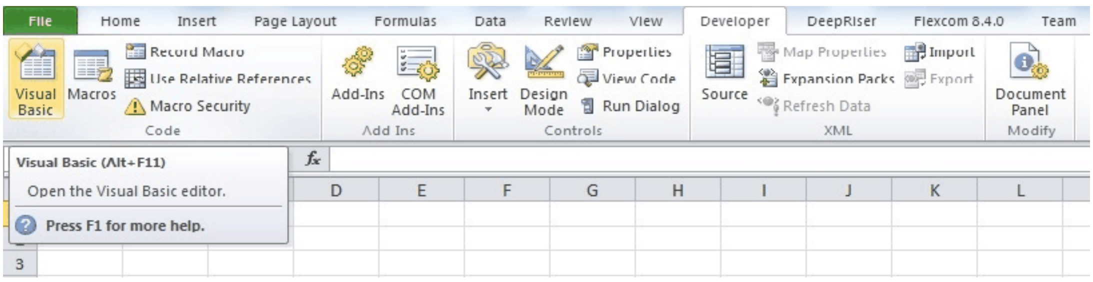


Enable the Developer tab in the Customize Ribbon options

ADDING A REFERENCE

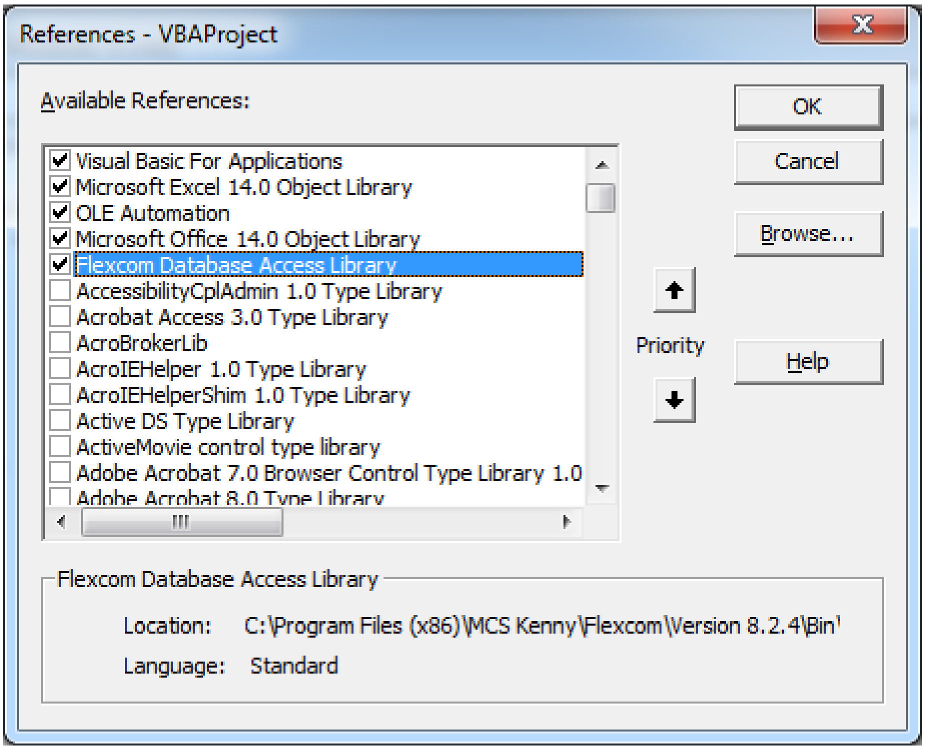
1. Next you to add a reference to the Flexcom Database Access Library to the project.

2. Go to the Developer Tab and click on the Visual Basic button on the left of the ribbon. This will open the VBA Integrated Development Environment (IDE) in a new window.



Visual Basic button

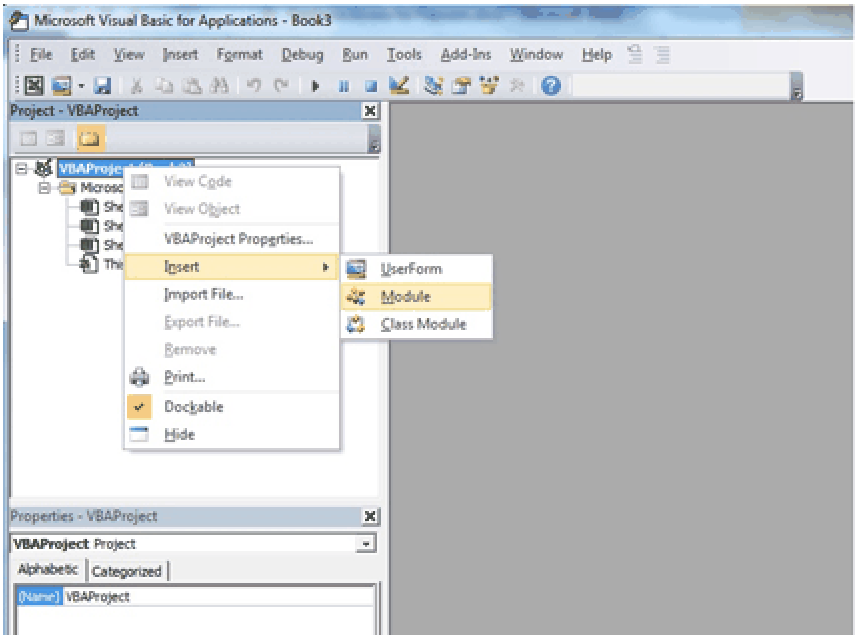
3. On the Tools menu of the IDE, click References, then select Flexcom Database Access Library and click OK.



Adding a reference to the Flexcom Database Access Library in the VBA IDE

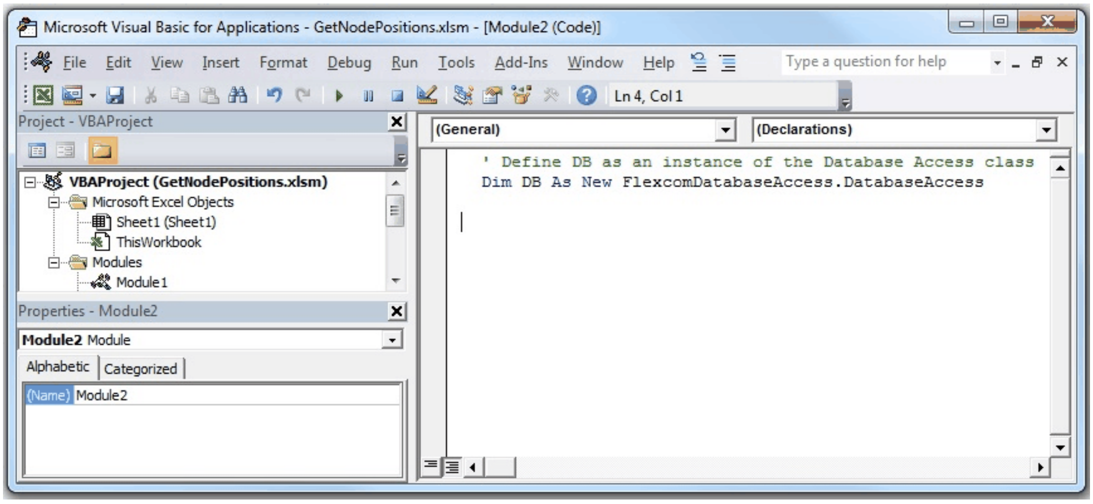
ADDING VBA CODE

1. Next, create a new module by right clicking on the VBA project icon, then click on Insert and select the Module option, as shown in the figure below.



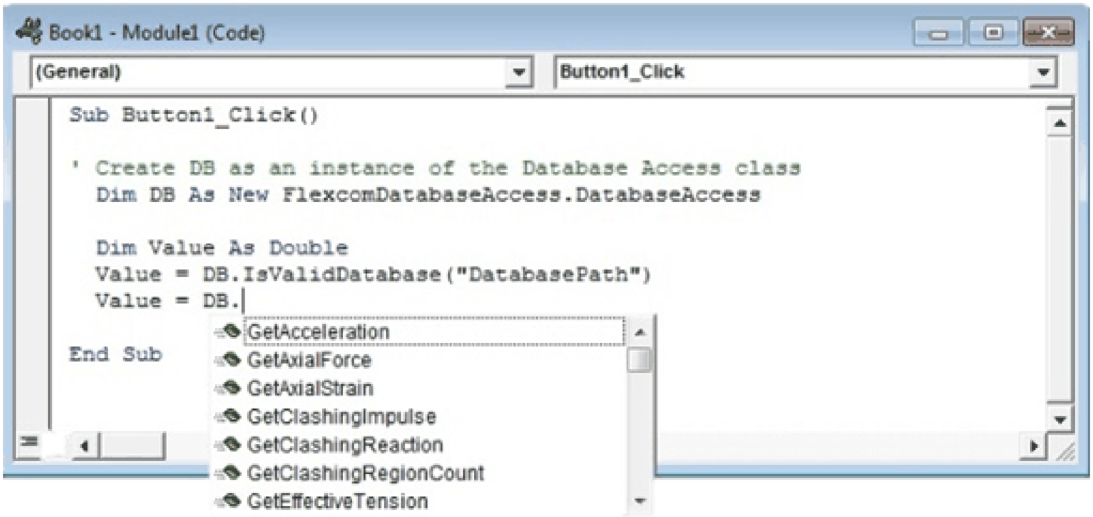
Open a module in the Visual Basic project

2. To create an instance of the DatabaseAccess class, define a new parameter, named “DB” in this example, as illustrated in the below figure.



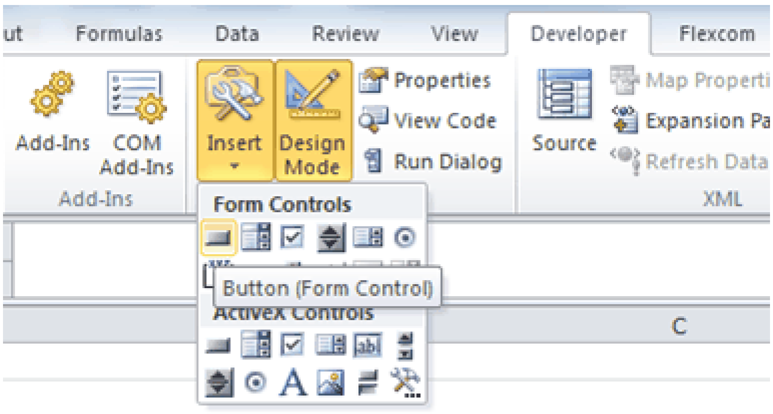
Create a new instance of the DatabaseAccess class

3. To access all the Flexcom Database Access functions use the “DB.” syntax. The figure below shows that once the dot is typed, VBA helps you by displaying a list of member functions to pick from.



VBA environment suggests the functions available

Once you are finished adding code to a macro, it may be ran by pressing the Run button or pressing the F5 key within the VBA environment. Alternatively, it is possible to create a button in the active Excel workbook and assigning the macro to it. To do this, click Insert Controls on the Developer tab and select Button (Form Control), highlighted in the figure below. On the dialog box that appears, choose the macro you want the button to run and click OK.



Creating a button in an Excel workbook

EXAMPLES

The Node Positions Example article gives an example of a simple VBA macro to extract node positions from the database and write them to files. The Statistics Example article gives an example of a more complex VBA macro to calculate statistics such as mean and standard deviation of effective tension.

Formula Recalculation - Important Information

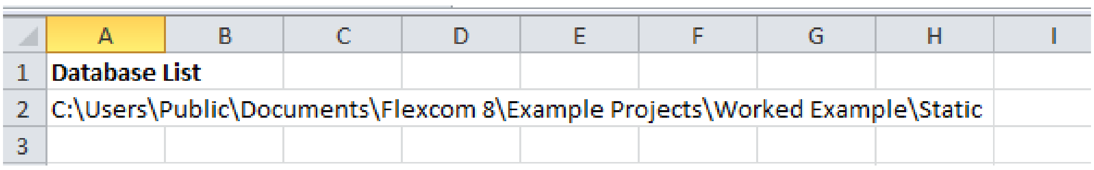
Microsoft Excel recalculates formulas only when the cells that the formula depends on have changed. This aspect of the program is very significant in the context of the Flexcom Excel Add-In. If you are using an Excel workbook to retrieve analysis results directly from a Flexcom database, the values presented in the spreadsheet will not be automatically updated if you re-run the analysis which created the database.

The Flexcom Excel Add-In is based on the use of functions (as opposed to macros) to retrieve information from the database files. Even the standard templates provided on the Flexcom ribbon control are based on a pre-defined body of functions. When you insert a function into a cell for the first time, the returned value is naturally correct. However if the database contents changes subsequently, the cell contents are not automatically updated.

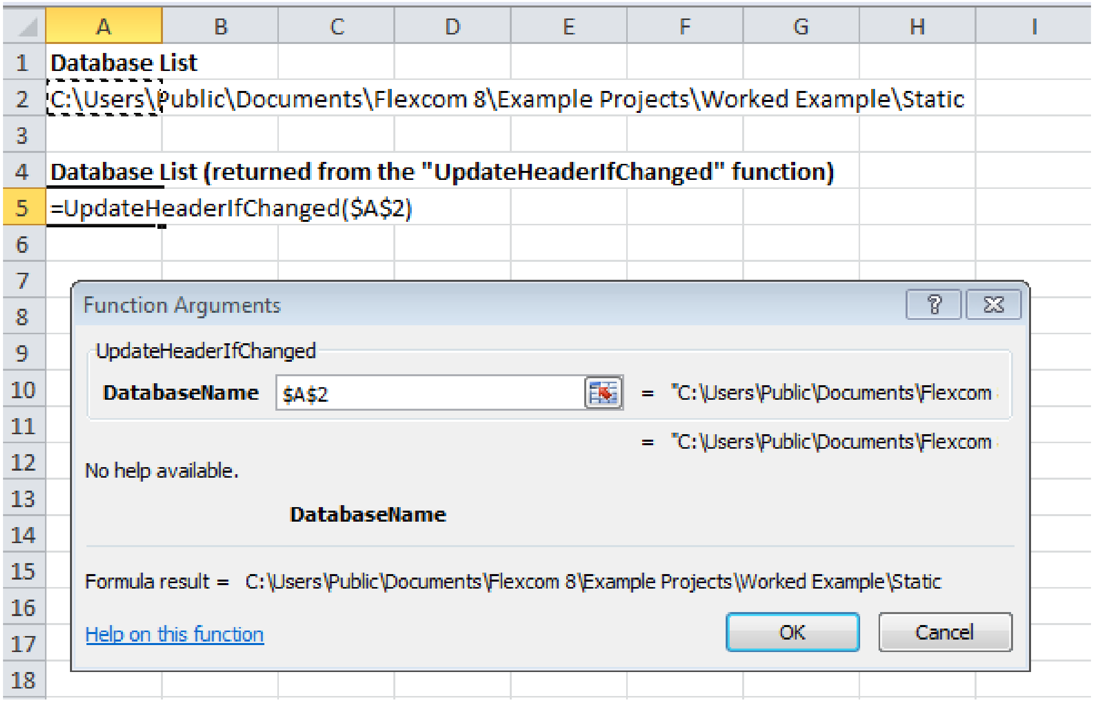
The use of either of the following procedures is recommended to ensure your worksheet is up to date:

Procedure (a)

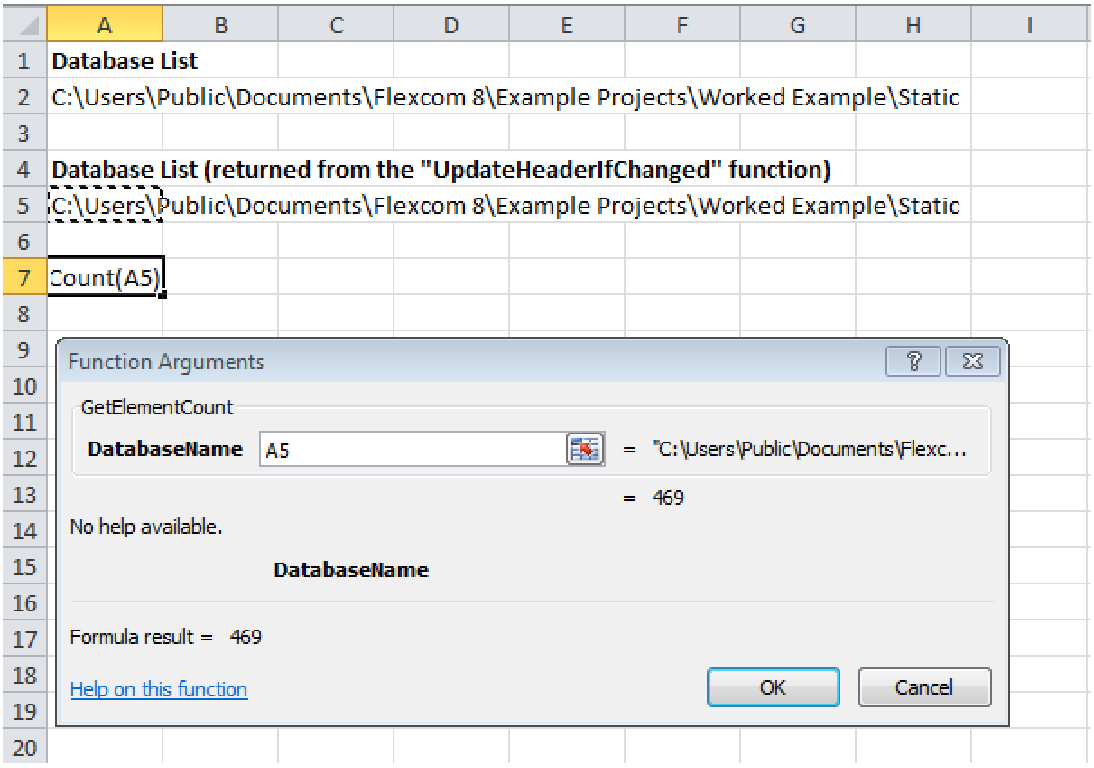
1. Fill a cell with the complete directory, including the filename, of the database you wish to interrogate.



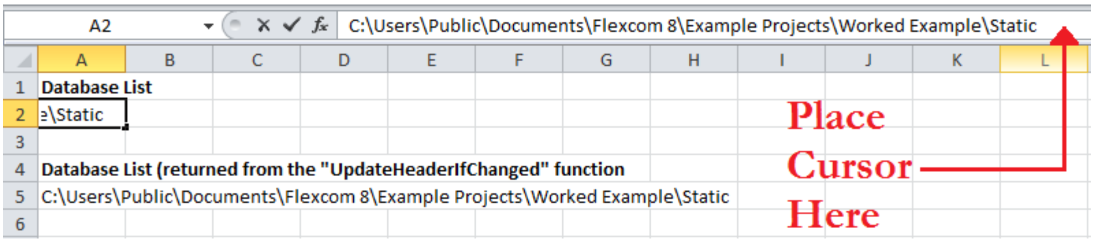
2. In a nearby cell, insert the function ‘UpdateHeaderIfChanged’. Its argument will be the cell containing the filename, as shown in Step 1 (i.e. cell A2). This function checks to see if the database contents have changed since it was loaded into Excel.



3. Any subsequent functions which use the database name as an argument should reference the cell created in Step 2 (i.e. cell A5).



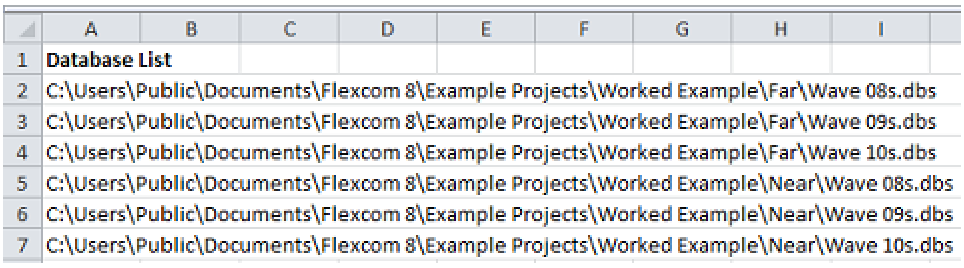
4. To update your workbook, select the cell created in Step 1 (i.e. A2). Place the cursor at the end of the function in the formula bar (as shown below). Press the Return key to manually force Microsoft Excel to update the cell contents.



Steps 1 to 4 should be repeated for any and all subsequent databases you wish to interrogate. If these steps are not followed, it is possible that erroneous data may be displayed.

Procedure (b)

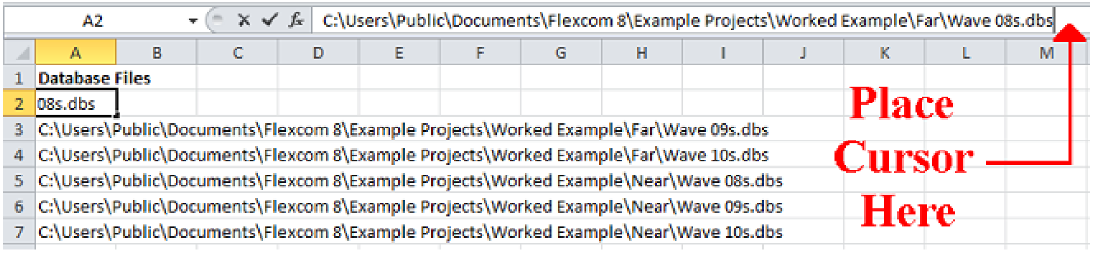
1. Reserve some governing cells for the storage of the names of the database files which your workbook references (as shown below). All cells within the workbook should reference one of these master cells, either directly or indirectly.



2. Save your workbook and close Microsoft Excel.

3. Launch Microsoft Excel and reopen your workbook.

4. Select the first cell in the master list of database files. Place the cursor at the end of the file name in the formula bar (as shown below). Press the Return key to manually force Microsoft Excel to update the cell contents. Repeat this procedure for every other cell in the master database list.



This may sound like a tedious procedure, but as long as you follow the guidelines above, it is a very reliable method and relatively quick to employ. The operation of Microsoft Excel in this regard is outside our control. Note also that if you are comfortable with the use of macros or Visual Basic scripts, it should be possible to write an algorithm which performs the updates automatically.

General Functions

This section contains information on:

- GetClashingRegionCount
- GetElementCount
- GetFlatGuideCount
- GetLabelCount
- GetModifiedDate
- GetNodeCount
- GetParameterType
- GetPIPConnectionCount
- GetZeroGapGuideCount
- IsValidDatabase
- IsValidDatabaseAsText
- UpdateHeaderIfChanged

Purpose

This function retrieves the number of clashing regions in the model.

Syntax

GetClashingRegionCount (string DatabaseName)

Parameters

DatabaseName (character string). The full path name to the relevant database.

Error Codes

-9999998 (if the specified database is invalid)

Purpose

This function retrieves the number of elements in the model.

Syntax

GetElementCount (string DatabaseName)

Parameters

DatabaseName (character string). The full path name to the relevant database.

Error Codes

-9999998 (if the specified database cannot be opened).

Purpose

This function retrieves the number of flat guide surfaces in the model.

Syntax

GetFlatGuideCount (string DatabaseName)

Parameters

DatabaseName (character string). The full path name to the relevant database.

Error Codes

-9999998 (if the specified database is invalid)

Purpose

This function retrieves the number of node and element labels in the model.

Syntax

GetLabelCount (string DatabaseName)

Parameters

DatabaseName (character string). The full path name to the relevant database.

Error Codes

-9999998 (if the specified database is invalid)

Purpose

This function retrieves the date/time at which the database was last modified.

Syntax

GetModifiedDate (string DatabaseName)

Parameters

DatabaseName (string). The full path name to the relevant database.

Return Value

Database modification date and time (real).

Purpose

This function retrieves the number of nodes in the model.

Syntax

GetNodeCount (string DatabaseName)

Parameters

DatabaseName (character string). The full path name to the relevant database.

Error Codes

-9999998 (if the specified database is cannot be opened)

Purpose

This function produces a descriptive title based on a specified force parameter index.

Syntax

GetParameterType (int parameterType)

Parameters

parameterType (integer). The parameter type index (1-17). The values are as follows:

1 Reaction Force

2 Axial Force

3 Local Y Shear Force

4 Local Z Shear Force

5 Torque

6 Local Y Bending Moment

7 Local Z Bending Moment

8 Effective Tension

9 Local Y Curvature

10 Local Z Curvature

11 Local Axial Strain

12 Flat Guide Reactions

13 Zero-Gap Guide Reactions

14 Pipe-in-Pipe Reactions

15 Clashing Reactions

16 Temperature

17 Pressure

Error Codes

'Invalid Parameter Type' (if the specified parameter type index is invalid)

Purpose

This function retrieves the number of pipe-in-pipe connections in the model.

Syntax

GetPIPConnectionCount (string DatabaseName)

Parameters

DatabaseName (character string). The full path name to the relevant database.

Error Codes

-9999998 (if the specified database is invalid)

Purpose

This function retrieves the number of zero-gap guides in the model.

Syntax

GetZeroGapGuideCount (string DatabaseName)

Parameters

DatabaseName (character string). The full path name to the relevant database.

Error Codes

-9999998 (if the specified database is invalid)

Purpose

This function checks the presence/status of a specified database file, denoted by an integer output. Specification of a file extension is optional, and if omitted, the function will check for either one of motion (DBM) or force (DBF) database files.

Syntax

IsValidDatabase (string DatabaseName)

Parameters

DatabaseName (string). The full path name to the relevant database.

Return Value

0 (if the database is valid)

Error Codes

1 (if the database files are present but either has changed)

-1 (if the force database is missing)

-2 (if the motion database is missing)

-9999970 (if the path or filename contains invalid characters)

-9999980 (if the database filename is empty)

-9999998.0 (if the both the force database and motion database are missing)

Purpose

This function checks the presence/status of a specified database file, denoted by an descriptive text output. Specification of a file extension is optional, and if omitted, the function will check for either one of motion (DBM) or force (DBF) database files.

Syntax

IsValidDatabaseAsText (string DatabaseName)

Parameters

DatabaseName (string). The full path name to the relevant database.

Return Value

‘Database is Valid’ (if the specified database is valid)

Error Codes

‘Database has changed’ (if the database file is present, but has been updated since Excel was launched)

‘Force Database is missing’ (if the force database is missing)

‘Motion Database is missing’ (if the motion database is missing)

’Database name contains invalid characters’ (if the path or filename contains invalid characters)

‘Database name is empty’ (if the database filename is empty)

‘Database is Invalid’ (if the specified database is invalid)

Purpose

By way of introduction, all databases created by Flexcom have “header” information detailing, among other things, the locations of blocks of data within the database. These blocks contain results for parameters such as axial strain, bending moments etc. Refer to Database Access Routines if you are interested in further details. When you load a database file into Excel, the “header” information is retrieved only once, for efficiency, and is stored in a “cache” or location in memory.

The purpose of this function is to clear the “cache”, and reread the “header” information from the database file. This is a very important function, as it may be used to ensure the contents of the Excel workbook if fully up to date.

Syntax

UpdateHeaderIfChanged (string DatabaseName)

Parameters

DatabaseName (string). The full path name to the relevant database.

Return Value

DatabaseName

Element Functions

This section contains information on:

- GetElementEndNode
- GetElementIndexFromUserElement
- GetElementInnerDiameter
- GetElementLength
- GetElementOuterDiameter
- GetElementStartNode
- GetUserElementNumber

Purpose

This function retrieves the end node of an element.

Syntax

GetElementEndNode (string DatabaseName, int ElementIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

ElementIndex (integer). The relevant index of the element.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the index is out of bounds)

Purpose

This function retrieves the internal index of an element based on a user-defined element number.

Syntax

GetElementIndexFromUserElement (string DatabaseName, int UserElementNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

UserElementNo (integer). The relevant user element number.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if no match is found)

Purpose

This function retrieves the internal diameter of an element.

Syntax

GetElementInnerDiameter (string DatabaseName, int ElementIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

ElementIndex (integer). The relevant index of the element.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the specified index is invalid)

Purpose

This function retrieves the (undeformed) length of an element.

Syntax

GetElementLength (string DatabaseName, int ElementIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

ElementIndex (integer). The relevant index of the element.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the specified index is invalid)

Purpose

This function retrieves the external diameter of an element.

Syntax

GetElementOuterDiameter (string DatabaseName, int ElementIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

ElementIndex (integer). The relevant index of the element.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the specified index is invalid)

Purpose

This function retrieves the start node of an element.

Syntax

GetElementStartNode (string DatabaseName, int ElementIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

ElementIndex (integer). The relevant index of the element.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the index is out of bounds)

Purpose

This function retrieves a user-defined element number based on a specified internal element index.

Syntax

GetUserElementNumber (string DatabaseName, int ElementIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

ElementIndex (integer). The relevant element index.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Label Functions

This section contains information on:

- GetLabel
- GetLabelType
- GetLabelTypeForLabel
- GetUserElementFromLabel
- GetUserNodeFromLabel

Purpose

This function retrieves a descriptive node or element label based on a specified label index.

Syntax

GetLabel (string DatabaseName, int LabelIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

LabelIndex (integer). The relevant index of the label.

Error Codes

Empty string (if a label for the specified index cannot be found)

Purpose

This function retrieves the type, whether node or element, of a descriptive label based on a specified label index.

Syntax

GetLabelType (string DatabaseName, int LabelIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

LabelIndex (integer). The relevant node/element label index.

Return Value

1 (for a node)

2 (for an element)

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the specified index is out of range)

Purpose

This function retrieves the type, whether node or element, of a descriptive label.

Syntax

GetLabelTypeForLabel (string DatabaseName, string Label)

Parameters

DatabaseName (character string). The full path name to the relevant database.

Label (string). The relevant node/element label.

Return Value

1 (for a node)

2 (for an element)

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the specified label cannot be found)

Purpose

This function retrieves a user-defined element number based on a specified descriptive element label.

Syntax

GetUserElementFromLabel (string DatabaseName, string Label).

Parameters

DatabaseName (character string). The full path name to the relevant database.

Label (character string). The label used to identify the element.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be found in the database)

Purpose

This function retrieves a user-defined node number based on a specified descriptive node label.

Syntax

GetUserNodeFromLabel (string DatabaseName, string Label).

Parameters

DatabaseName (character string). The full path name to the relevant database.

Label (character string). The label used to identify the node.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be found in the database)

Node Functions

This section contains information on:

- GetNodeIndexFromUserNode
- GetUserNodeNumber

Purpose

This function retrieves the internal index of a node based on a user-defined node number.

Syntax

GetNodeIndexFromUserNode (string DatabaseName, int UserNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

UserNodeNo (integer). The relevant user node number.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if no match is found)

Purpose

This function retrieves a user-defined node number based on a specified internal node index.

Syntax

GetUserNodeNumber (string DatabaseName, int NodeIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

NodeIndex (integer). The relevant node index.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Set Functions

This section contains information on:

- GetElementDistanceAlongSet
- GetNodeDistanceAlongSet
- GetSetCount
- GetSetElement

- GetSetElementCount
- GetSetName

Purpose

This function retrieves the distance of an element (start point) along a particular element set.

Syntax

GetElementDistanceAlongSet (string DatabaseName, string SetName, int ElementIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

SetName (character string). The full name of the relevant set.

ElementIndex (integer). The relevant index of the element.

Error Codes

-9999998 (if there is an issue with the database file)

-9999999 (if there is an issue with the indexes)

Purpose

This function retrieves the distance of a node along a particular element set.

Syntax

GetNodeDistanceAlongSet (string DatabaseName, string SetName, int NodeIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

SetName (character string). The full name of the relevant set.

NodeIndex (integer). The relevant node index.

Error Codes

-9999998 (if there is an issue with the database file)

-9999999 (if there is an issue with the nodes)

Purpose

This function retrieves the number of element sets in the model.

Syntax

GetSetCount (string DatabaseName)

Parameters

DatabaseName (character string). The full path name to the relevant database.

Error Codes

-9999998 (if there is an issue with the specified database file)

Purpose

This function retrieves a user-defined element number based on a specified internal element index within a particular element set.

Syntax

GetSetElement (string DatabaseName, string SetName, int ElementIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

SetName (character string). The full name of the relevant set.

ElementIndex (integer). The element index within the relevant set.

Error Codes

-9999990 (if the element number is out of bounds)

-9999998 (if the specified database is invalid)

-9999999 (if the specified set name is invalid)

Purpose

This function retrieves the number of elements within a particular element set.

Syntax

GetSetElementCount (string DatabaseName, string SetName)

Parameters

DatabaseName (character string). The full path name to the relevant database.

SetName (character string). The full name of the relevant set.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the specified set name is invalid)

Purpose

This function retrieves an element set name based on a specified internal element set index.

Syntax

GetSetName (string DatabaseName, int SetIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

SetIndex (integer). The relevant index of the set.

Error Codes

Empty string (if the specified index is out of range)

Empty string (if the specified database is invalid)

Time Functions

This section contains information on:

- GetSolutionTimeCount
- GetTime
- GetTimeIndex

Purpose

This function retrieves the number of solution times in the analysis.

Syntax

GetSolutionTimeCount (string DatabaseName)

Parameters

DatabaseName (character string). The full path name to the relevant database.

Error Codes

-9999998 (if the specified database cannot be opened)

Purpose

This function retrieves the solution time for a specified solution time index.

Syntax

GetTime (string DatabaseName, int TimeIndex)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the solution time index for a specified solution time.

Syntax

GetTimeIndex (string DatabaseName, float Time)

Parameters

DatabaseName (character string). The full path name to the relevant database.

Time (float). The relevant time value.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if a match cannot be found)

Force Functions

This section contains information on:

- GetAxialForce
- GetAxialStrain
- GetClashingImpulse
- GetClashingReaction
- GetEffectiveTension

- GetFlatGuideReaction
- GetForceValue
- GetLocalYBendingMoment
- GetLocalYCurvature
- GetLocalYShearForce
- GetLocalZBendingMoment
- GetLocalZCurvature
- GetLocalZShearForce
- GetNodeReaction
- GetPIPReaction
- GetPressure
- GetTemperature
- GetTorque
- GetZeroGapGuideReaction

Purpose

This function retrieves the axial force in an element at a particular time step.

Syntax

GetAxialForce (string DatabaseName, int TimeIndex, int ElementIndex, int LocalNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ElementIndex (integer). The relevant element index.

LocalNodeNo (integer). The relevant local node number (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the axial strain in an element at a particular time step.

Syntax

GetAxialStrain (string DatabaseName, int TimeIndex, int ElementIndex, int LocalNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ElementIndex (integer). The relevant element index.

LocalNodeNo (integer). The relevant local node number (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the contact impulse for a particular clashing region at a particular time step.

Syntax

GetClashingImpulse (string DatabaseName, int TimeIndex, int ClashingRegionNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ClashingRegionNo (integer). The relevant clashing region number.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the contact reaction force for a particular clashing region at a particular time step.

Syntax

GetClashingReaction (string DatabaseName, int TimeIndex, int ClashingRegionNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ClashingRegionNo (integer). The relevant clashing region number.

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the effective tension in an element at a particular time step.

Syntax

GetEffectiveTension (string DatabaseName, int TimeIndex, int ElementIndex, intLocalNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ElementIndex (integer). The relevant element index.

LocalNodeNo (integer). The relevant local node number (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the reaction force from a flat guide surface in a particular degree of freedom, at a particular time step.

Syntax

GetFlatGuideReaction (string DatabaseName, int TimeIndex, int guideNo, int DOFNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

guideNo (integer). The relevant guide index number.

DOFNo (integer). The Degree of Freedom (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This is a generic function capable of retrieving a wide variety of data from the force database. Refer to the discussion below for further details.

Syntax

GetForceValue (string DatabaseName, int TimeIndex, int ParameterType, int ItemNo, int DOFNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ParameterType (integer). The parameter type index. The values are as follows:

1 Reaction Force

2 Axial Force

3 Local Y Shear Force

4 Local Z Shear Force

5 Torque

6 Local Y Bending Moment

7 Local Z Bending Moment

8 Effective Tension

9 Local Y Curvature

10 Local Z Curvature

11 Local Axial Strain

12 Flat Guide Reactions

13 Zero-Gap Guide Reactions

14 Pipe-in-Pipe Reactions

15 Clashing Reactions

16 Temperature

17 Pressure

ItemNo (integer). The item number. The significant of this variable is dependent on the selected parameter type. The values are as follows:

Parameter Index	Parameter Type	Item Number
1	Reaction Forces	Constrained Node Number Index
2-11	Element Forces/Stresses	Element Position (1=Start, 2=Midpoint, 3=End)
12,13	Guide Reactions	Guide Index
14	Pipe-in-Pipe Reactions	Pipe-in-Pipe Connection Index
15	Clashing Reactions	Clashing Region Index
16,17	Temperature/Pressure	Element Position (1=Start, 2=Midpoint, 3=End)

DOFNo (integer). The Degree of Freedom (1 to 6).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database or if the inputs are out of range)

Purpose

This function retrieves the local–Y bending moment in an element at a particular time step.

Syntax

GetLocalYBendingMoment (string DatabaseName, int TimeIndex, int ElementIndex, int LocalNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ElementIndex (integer). The relevant element index.

LocalNodeNo (integer). The relevant local node number (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the local–Y curvature in an element at a particular time step.

Syntax

GetLocalYCurvature (string DatabaseName, int TimeIndex, int ElementIndex, int LocalNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ElementIndex (integer). The relevant element index.

LocalNodeNo (integer). The relevant local node number (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the local–Y shear force in an element at a particular time step.

Syntax

GetLocalYShearForce (string DatabaseName, int TimeIndex, int ElementIndex, int LocalNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ElementIndex (integer). The relevant element index.

LocalNodeNo (integer). The relevant local node number (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the local–Z bending moment in an element at a particular time step

Syntax

GetLocalZBendingMoment (string DatabaseName, int TimeIndex, int ElementIndex, int LocalNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ElementIndex (integer). The relevant element index.

LocalNodeNo (integer). The relevant local node number (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the local–Z curvature in an element at a particular time step.

Syntax

GetLocalZCurvature (string DatabaseName, int TimeIndex, int ElementIndex, int LocalNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ElementIndex (integer). The relevant element index.

LocalNodeNo (integer). The relevant local node number (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the local–Z shear force in an element at a particular time step.

Syntax

GetLocalZShearForce (string DatabaseName, int TimeIndex, int ElementIndex, int LocalNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ElementIndex (integer). The relevant element index.

LocalNodeNo (integer). The relevant local node number (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the reaction force at a constrained node in a particular degree of freedom, at a particular time step (based on the node's index within a list of constrained nodes).

Syntax

GetNodeReaction (string DatabaseName, int TimeIndex, int ConstrainedNodeIndex, intDOFNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ConstrainedNodeIndex (integer). The relevant node index in the boundary condition array.

DOFNo (integer). The Degree of Freedom (1 to 6).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the reaction force at a pipe-in-pipe connection in a particular degree of freedom, at a particular time step.

Syntax

GetPIPReaction (string DatabaseName, int TimeIndex, PIPConnectionNo, int DOFNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

PIPConnectionNo (integer). The guide index number from 1 to number of pipe-in-pipe connections in the database.

DOFNo (integer). The Degree of Freedom (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the internal or external pressure for an element at a particular time step.

Syntax

GetPressure (string DatabaseName, int TimeIndex, int ElementIndex, int LocalNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ElementIndex (integer). The relevant element index.

PressureParamater (integer). The possible values are:

1. Internal pressure at the local element start
2. External pressure at the local element start
3. Internal pressure at the local element middle
4. External pressure at the local element middle
5. Internal pressure at the local element end
6. External pressure at the local element end

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the temperature for an element at a particular time step.

Syntax

GetTemperature (string DatabaseName, int TimeIndex, int ElementIndex, int LocalNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ElementIndex (integer). The relevant element index.

LocalNodeNo (integer). The relevant local node number (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the torque in an element at a particular time step.

Syntax

GetTorque (string DatabaseName, int TimeIndex, int ElementIndex, int LocalNodeNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ElementIndex (integer). The relevant element index.

LocalNodeNo (integer). The relevant local node number (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the reaction force from a zero-gap guide in a particular degree of freedom, at a particular time step.

Syntax

GetZeroGapGuideReaction (string DatabaseName, int TimeIndex, int GuideNo, int DOFNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

GuideNo (integer). The relevant guide number.

DOFNo (integer). The Degree of Freedom (1 to 3).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Kinematic Functions

This section contains information on:

- GetAcceleration
- GetKinematicValue
- GetPosition
- GetVelocity

Purpose

This function retrieves the acceleration of a node in a particular degree of freedom at a particular time step.

Syntax

GetAcceleration (string DatabaseName, int TimeIndex, int NodeIndex, int DOFNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

NodeIndex (integer). The relevant node index.

DOFNo (integer). The Degree of Freedom (1 to 6).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the position, velocity or acceleration of a node in a particular degree of freedom at a particular time step.

Syntax

GetKinematicValue (string DatabaseName, int TimeIndex, int ParameterType, int NodeIndex, int DOFNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

ParameterType (integer). The Parameter Type Index for the motion database. The values are as follows:

1 Motion

2 Velocity

3 Acceleration

NodeIndex (integer). The relevant node index.

DOFNo (integer). The Degree of Freedom (1 to 6).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the position of a node in a particular degree of freedom at a particular time step.

Syntax

GetPosition (string DatabaseName, int TimeIndex, int NodeIndex, int DOFNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

NodeIndex (integer). The relevant node index.

DOFNo (integer). The Degree of Freedom (1 to 6).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Purpose

This function retrieves the velocity of a node in a particular degree of freedom at a particular time step.

Syntax

GetVelocity (string DatabaseName, int TimeIndex, int NodeIndex, int DOFNo)

Parameters

DatabaseName (character string). The full path name to the relevant database.

TimeIndex (integer). The relevant time index.

NodeIndex (integer). The relevant node index.

DOFNo (integer). The Degree of Freedom (1 to 6).

Error Codes

-9999998 (if the specified database is invalid)

-9999999 (if the requested value cannot be retrieved from the database)

Database Access Routines

THEORY

This section provides information regarding the use of Database Access Routines. This feature is intended for advanced users of the software, and facilitates the development of specialised postprocessing tools tailored to meet the specific requirements of individual users. It is divided into the following sections:

- Database Files introduces Flexcom's motion and force database files.
- Database Access Routines describes the collection of procedures that allow the retrieval of results directly from the database files.
- Motion Database File Structure documents the contents of the motion database file.
- Force Database File Structure documents the contents of the force database file.

The Excel Add-in for Flexcom is an alternative advanced method to extract information from the results database.

RELEVANT KEYWORDS

- `*DATABASE` is used to specify the frequency of database output.
- `*DATABASE CONTENT` is used to customise the contents of the database output files.

The Database Access Routines themselves do not have any associated keywords as such. Rather they are invoked from a user-defined application to facilitate customised postprocessing. If you would like to see an example of how the DARs are used in practice, refer to J02 - Advanced Database Postprocessing. This example actually contains two sub-sections, namely...

- Database Access via Excel VBA
- Database Access via Fortran DAR

Database Files

Flexcom can produce two database files, referred to as the motion database file and the force database file. Together, these can contain a detailed description of the analysis solution at all or selected solution times. These files are unformatted random access files, so you cannot type, print or edit them. The motion database file can contain information on the position, velocity and acceleration of all or selected nodes in the model (including runtime generated statistics of motion). The force database file can contain information on:

(i) Restoring forces in all or selected elements

(ii) Reactions at all restraints

(iii) Statistics of restoring forces generated during runtime

Restoring forces in this context may include axial forces, effective tensions, shears, bending moments relative to both local $\hat{y}$ and $\hat{z}$ axes, and curvatures relative to these same local axes. A number of further outputs may optionally be added to these parameters

Database Access Routines

The database access routines are a collection of procedures that allow the retrieval of results directly from the database files. These procedures are packaged in the dynamic link library (DLL) mcsdar.dll.

In order to build these routines into your FORTRAN or Visual C++ program you need:

1. To reference the symbol export library mcsdar.lib in your program, and
2. To call the routines using the predefined calling convention as described in each routine interface.

To run your program you need to place a copy of the mcsdar.dll in the same folder as your executable.

The calling convention of each routine is Standard Call by Reference. This allows all arguments to be passed by reference. Character strings are passed by reference along with their length. The length of the character strings is a hidden argument in FORTRAN. The name of each routine is decorated with an underscore prefix (_) and a @n suffix. The number n represents the length of the argument list in bytes. Integer and real data types are 4 bytes long. A single character is 1 byte long.

The procedures available are listed below:

- `GetDatabaseInfo` returns information about the database files and the model. This must be called first.
- `XIT` deallocates memory and closes the database files, if opened. This function must be called last after all data of interest has been retrieved.
- `GetAnalysisDetails` returns the analysis name/title, gravity constant and two flags indicating if this is a static analysis and if the time-step is fixed or variable.
- `GetProcessedDirectStressStrainData` returns all processed nonlinear direct stress-strain relationships in the format of axial strain - axial force
- `GetNodeProperties` returns the node properties in terms of initial co-ordinates, as specified directly by the user, or as computed by the Cable Pre-Static Step, if a cable was specified in the model

- `GetElementProperties` returns the properties associated with this element
- `GetRequestedHistory` returns the requested history
- `GetRequestedHistoryError` flags the errors that occurred during the last history request
- `GetRequestedSingleValue` returns the requested single value
- `GetRequestedSingleValueError` flags the errors that occurred during the last single value request

Each of these routines is now described in turn.

GETDATABASEINFO FUNCTION

Returns information about the database files and the model.

SYNTAX

```
Value = GetDatabaseInfo(
  revdbm,
  revdbf,
  versiondbm,
  versiondbf,
  no_parameters,
  model_parameters,
  filenamepath)
```

DESCRIPTION OF ARGUMENTS

Argument	Description
revdbm	Revision number of the motion database file. Integer scalar (output). -1 – file is corrupted. 0 – file does not exist. a positive value indicates a valid database file.

revdbf	Revision number of the force database file. Integer scalar (output). -1 – file is corrupted. 0 – file does not exist. a positive value indicates a valid database file.
versiondbm	Version information for the motion database file. Integer array of dimension 5 (output). The following are stored: 1 – Major Version Number 2 – Minor Version Number 3 – Maintenance Version Number 4 – Month of Release 5 – Year of Release
versiondbf	Version information for the force database file. Integer array of dimension 5 (output). The following are stored: 1 – Major Version Number 2 – Minor Version Number 3 – Maintenance Version Number 4 – Month of Release 5 – Year of Release
no_parameters	Number of model parameters. Integer scalar (input).
model_parameters	Integer array of dimension no_parameters (output), which holds model information. Items in model_parameters:

Index	Value represents
1	Number of outputted time-steps to the database
2	Number of elements in the model
3	Number of elements stored in the database
4	Number of nodes stored in the database
5	Number of auxiliary elements in the model
6	Number of auxiliary nodes in the model
7	Number of labels
8	Number of nodes with assigned boundary conditions
9	Number of flat guide surfaces
10	Number of zero-gap guides
11	Number of pipe-in-pipe sections
12	Number of pipe-in-pipe connections
13	Number of clashing regions
14	Number of maximum force/strain points resulting from the processing of the nonlinear direct stress-strain relationships

filenamepath	Analysis file name or the full path to the analysis filename. Character string of unspecified length (input). If the full path is not used, then this variable refers to a path relative to your Database Access executable.

RETURN VALUE

Value	Description
-3	Model details could not be read; check values of revdbm and revdbf
-2	The value specified as number of parameters is not a number
-1	Path is null
0	Successful
A positive value	The number of model parameters specified is too small. Call funtion again substituting the returned value in no_parameters

XIT FUNCTION

Deallocates memory and closes the database files, if opened.

SYNTAX

```
Value = XIT()
```

DESCRIPTION OF ARGUMENTS

This function has no arguments.

RETURN VALUE

Value	Description
0	Successful
1	Motion database file could not be closed
2	Force database file could not be closed
3	Both files could not be closed

GETANALYSISDETAILS

Returns the analysis name/title, gravity constant and two flags indicating if this is a static analysis and if the time-step is fixed or variable.

SYNTAX

```
Value = GetAnalysisDetails(
  static_flag,
  fixed_tstep,
  gravity,
  analysisname)
```

DESCRIPTION OF ARGUMENTS

Argument	Description
static_flag	Flag to indicate if this is a static analysis or not. Integer scalar (output) 1 – static analysis 0 – otherwise
fixed_tstep	Flag to indicate if a fixed time-step was used. Integer scalar (output) 1 – fixed time-step 0 – variable time-step
gravity	Gravity Constant. Real scalar (output)
analysisname	Analysis name/title. Character string (output)

RETURN VALUE

Value	Description
0	Successful

A positive value	Minimum length of the character string to be allocated before calling this function. Nothing is transfered in the string. Set the length of the character string at least of the returned value and call this function again.

Returns all processed nonlinear direct stress-strain relationships in the format of axial strain - axial force.

SYNTAX

```
Value = GetProcessedDirectStressStrainData(
  revdbm,
  revdbf,
  table_no,
  no_points,
  processed_data)
```

DESCRIPTION OF ARGUMENTS

<table>
<tr><th>Argument</th><th colspan="2">Description</th></tr>
<tr><td>revdbm</td><td colspan="2">Revision number of the motion database file. Integer scalar (input)</td></tr>
<tr><td>revdbf</td><td colspan="2">Revision number of the force database file. Integer scalar (input)</td></tr>
<tr><td>table_no</td><td colspan="2">Nonlinear material table number. Integer scalar (input)</td></tr>
<tr><td>no_points</td><td colspan="2">Number of data points. Dimension of the processed_data array. Integer scalar (input)</td></tr>
<tr><td rowspan="4">processed_data</td><td colspan="2">Real array of dimensions (2,no_points) (output).</td></tr>
<tr><td>First Index</td><td>Value represents</td></tr>
<tr><td>1</td><td>Axial Strain</td></tr>
<tr><td>2</td><td>Axial Force</td></tr>
</table>

RETURN VALUE

Value	Description
A negative value	Requested table was not found
0	Successful
A positive value	Minimum array dimension (no_points) to hold all data. Array is not filled in. Call this function again substituting this value in no_points.

Returns the node properties in terms of initial co-ordinates, as specified directly by the user or after the Cable Pre-Static Step, if a cable was specified in the model.

SYNTAX

```
Value = GetNodeProperties(
  revdbm,
  revdbf,
  node,
  auxiliaryflag,
  coord)
```

DESCRIPTION OF ARGUMENTS

Argument	Description
revdbm	Revision number of the motion database file. Integer scalar (input)
revdbf	Revision number of the force database file. Integer scalar (input)
node	Node number. Integer scalar (input)

auxiliaryflag	0 –standard node, 1 –auxiliary node. Integer scalar (input)		
coord	Array holding node initial co-ordinates, as specified directly by the user or after the Cable Pre-Static Step. Real array of dimension (3) (output).		
	Index	Value represents	
	1	X Co-ordinate	
	2	Y Co-ordinate	
	3	Z Co-ordinate	

RETURN VALUE

Value	Description
-2	Unknown value assigned to the auxiliary flag
-1	Data could not be accessed
0	Specified node is not in the database.
A positive value	Internal node index in the database. Co-ordinates are returned.

Returns the properties associated with this element.

SYNTAX

```
Value = GetElementProperties(
  revdbm,
  revdbf,
  element,
  auxiliaryflag,
  no_prop,
  elem_iprop,
  elem_prop)
```

DESCRIPTION OF ARGUMENTS

<table>
<tr><th>Argument</th><th colspan="2">Description</th></tr>
<tr><td>revdbm</td><td colspan="2">Revision number of the motion database file. Integer scalar (input)</td></tr>
<tr><td>revdbf</td><td colspan="2">Revision number of the force database file. Integer scalar (input)</td></tr>
<tr><td>element</td><td colspan="2">Element number. Integer scalar (input)</td></tr>
<tr><td>auxiliaryflag</td><td colspan="2">0 –standard element, 1 –auxiliary element. Integer scalar (input)</td></tr>
<tr><td>no_prop</td><td colspan="2">Dimension of property arrays. Integer scalar (input).</td></tr>
<tr><td rowspan="7">elem_iprop</td><td colspan="2">Array holding the element integer properties. Integer array of dimension (no_prop) (output). Each item in the array is initialised to -1 in the routine prior to any data assignment. Only properties 1-4 are applicable to auxiliary elements.</td></tr>
<tr><td>Index</td><td>Value represents</td></tr>
<tr><td>1</td><td>Internal element index in the database</td></tr>
<tr><td>2</td><td>Number of the first node belonging to this element</td></tr>
<tr><td>3</td><td>Number of the second node belonging to this element</td></tr>
<tr><td>4</td><td>Geometric element set number</td></tr>
<tr><td>5</td><td>Flag to indicate if user effective properties were defined

0 – No</td></tr>
</table>

		1 – Yes
	6	Table number of the processed nonlinear direct stress-strain data
	7	Flag to indicate the material type associated with this element 0 – Linear elastic 1 – Direct stress-strain 2 – Nonlinear bending stiffness 3 – Only nonlinear axial or torsion stiffness
elem_prop	Array holding the element real properties. Real array of dimension (no_prop) (output). Each item in the array is initialised to -1.0 in the routine prior to any data assignment. None of these properties are applicable to auxiliary elements.	
	Index	Value represents
	1	Length
	2	External fluid height
	3	External fluid density
	4	External fluid pressure
	5	Internal fluid height
	6	Internal fluid density
	7	Internal fluid pressure
	8	Bending stiffness

	9	Axial stiffness if this element has an associated linear material specified in the Rigid Riser format, or the cross-section area (as specified under *GEOMETRIC SETS) if this element has an associated nonlinear material specified in the Rigid Riser format. In all other instances, this entry is zero.
	10	Effective external diameter
	11	Effective internal diameter
	12	Effective cross-section area
	13	Second moment of area Iyy about the local y-axis
	14	Second moment of area Izz about the local z-axis
	15	Wall thickness

RETURN VALUE

Value	Description
-3	Unknown value assigned to the auxiliary flag
-2	Data could not be accessed
-1	Specified element is not in the database
0	Successful. Properties are returned.
A positive value	Number of properties is set too low. Call this function again substituting this value in no_prop.

Returns the requested history.

SYNTAX

```
Value = GetRequestedHistory(
  revdbm,
  revdbf,
  output_parameter,
  dof,
  node_no,
  element_no,
  local_node_no,
  stored_timesteps,
  no_timesteps,
  array,
  label)
```

DESCRIPTION OF ARGUMENTS

Argument	Description
revdbm	Revision number of the motion database file. Integer scalar (input)
revdbf	Revision number of the force database file. Integer scalar (input)
output_parameter	Requested output parameter. Integer scalar (input). See Table of Outputs for more details.
dof	Requested output parameter degree of freedom. Integer scalar (input). See Table of Outputs for more details.
node_no	Node number. Integer scalar (input/output). This may also be used to specify a guide surface, zero-gap guide or pipe-in-pipe connection number. If a node label is specified and a value of zero is specified for this variable, then the actual node number designated by the label is returned here.

element_no	Element number. Integer scalar (input).
local_node_no	Element local node number. Integer scalar (input).
stored_timesteps	Number of outputted time-steps to the database. Integer scalar (input). Call GetDatabaseInfo function to retrieve this value.
no_timesteps	Number of time steps requested for output. Integer scalar (input).
array	Output parameter history information. Real array of dimension (no_timesteps) (output).
label	Node label. Character string of unspecified length (input). This is relevant only if the node number was specifed as zero otherwise the node number specified takes precedence.

RETURN VALUE

Value	Description
0	Successful. History is returned
A positive value	Errors occured. Please call the GetRequestedHistoryError function to check which errors occured.

TABLE OF OUTPUTS

Output parameter	DOF	Location of data	Description.
1	N/A	Motion/Force Database	Time.
2	N/A	Motion/Force Database	Time-Step.
3	N/A	Motion/Force Database	Water Surface Elevation.
4	N/A	Motion/Force Database	Kinetic Energy.
5	1-7	Motion Database	Motion (DOFs 1-6, 7 - resultant rotation).
6	1-7	Motion Database	Velocity (DOFs 1-6, 7 - resultant angular velocity).
7	1-7	Motion Database	Acceleration (DOFs 1-6, 7 - resultant angular acceleration).

8	1-3	Motion Database	Guide Surface Origin Position (X,Y,Z).
8	4-6	Motion Database	Guide Surface Local-x Vector (X,Y,Z).
8	7-9	Motion Database	Guide Surface Local-y Vector (X,Y,Z).
9	1-3	Motion Database	Zero-Gap Guide Origin Position (X,Y,Z).
9	4-6	Motion Database	Zero-Gap Guide Direction (X,Y,Z)
10	1-4	Motion Database	Seabed contact flag and X, Y and Z reactions, on node-by-node basis.
10	5-8	Motion Database	Guide contact flag and X, Y and Z reactions, on node-by-node basis.
11	1	Motion Database	Clashing Clearance.

11	2-3	Force Database	Clashing Reaction and Impulse.
12	1-3	Motion Database	Motion of Auxiliary Nodes (X,Y,Z).
13	1-7	Force Database	Node Reactions (DOFs 1-6, 7 - resultant moment).
14	1-4	Force Database	Guide Surface Contact Reactions (X,Y,Z,magnitude), on guide-by-guide basis.
15	1-4	Force Database	Zero-Gap Guide Contact Reactions (X,Y,Z,magnitude), on guide-by-guide basis.

16	1-4	Force Database	Pipe-In-Pipe Connection Reactions (1-axial, 2-normal, 3-transverse, 4-magnitude).
17	1	Force Database	Axial Force.
17	2	Force Database	Local-y Shear Force.
17	3	Force Database	Local-z Shear Force.
17	4	Force Database	Torque.
17	5	Force Database	Local-y Bending Moment.
17	6	Force Database	Local-z Bending Moment.
17	7	Force Database	Effective Tension.
17	8	Force Database	Resultant Bending Moment.
17	9	Force Database	Resultant Curvature.

17	10	Force Database	Resultant Shear Force.
17	11	Force Database	Local-y Curvature.
17	12	Force Database	Local-z Curvature.
17	13	Force Database	Axial Strain.
17	14	Force Database	Temperature.
18	1-2	Force Database	Pressure (1-Internal, 2-External).

Returns the type of error/errors that occured during the last call to GetRequestedHistory Function.

SYNTAX

```
Value = GetRequestedHistoryError(
  ierror,
  maxerrors,
  errors)
```

DESCRIPTION OF ARGUMENTS

Argument	Description
ierror	Return value of GetRequestedHistory Function. Integer scalar (input)
maxerrors	Dimension of array errors. Integer scalar (input)

<table>
<tr>
<td>errors</td>
<td>
Array holding flags to indicate if a particular error occured or not. Integer array of dimensions (maxerrors) (output).

0 – error did not occur

1 – error occured.

The items in the array are listed in the following order:
<table>
<tr><td>Index</td><td>Description</td></tr>
<tr><td>1</td><td>1st argument of GetRequestedHistory is not a number.</td></tr>
<tr><td>2</td><td>2nd argument of GetRequestedHistory is not a number.</td></tr>
<tr><td>3</td><td>3rd argument of GetRequestedHistory is not a number or is out of range.</td></tr>
<tr><td>4</td><td>4th argument of GetRequestedHistory is not a number or is out of range.</td></tr>
<tr><td>5</td><td>5th argument of GetRequestedHistory is not a number or is out of range.</td></tr>
<tr><td>6</td><td>6th argument of GetRequestedHistory is not a number or is out of range.</td></tr>
<tr><td>7</td><td>7th argument of GetRequestedHistory is not a number or is out of range.</td></tr>
<tr><td>8</td><td>8th argument of GetRequestedHistory is not a number or is out of range.</td></tr>
<tr><td>9</td><td>9th argument of GetRequestedHistory is not a number or is out of range.</td></tr>
<tr><td>10</td><td>Reserved.</td></tr>
</table>
</td>
</tr>
</table>

	11	11th argument of GetRequestedHistory: unknown label or unspecified label when both the node number is zero or not a number.
	12	Requested output parameter is not in the database.
	13	Requested node is not in the database.
	14	Requested element is not in the database.
	15	Could not read from database.

RETURN VALUE

Value	Description
-1	Failed. Return value of GetRequestedHistory (ierror) is negative
0	Successful.
A positive value	Failed – maxerrors is to small. Call the function again substituting the returned value of the function in variable maxerrors.

Returns the requested single value.

SYNTAX

```
Value = GetRequestedSingleValue(
  revdbm,
  revdbf,
  output_parameter,
  dof,
  node_no,
  element_no,
  local_node_no,
  stored_timesteps,
  timestep_index,
  single_value,
  label)
```

DESCRIPTION OF ARGUMENTS

Argument	Description
revdbm	Revision number of the motion database file. Integer scalar (input)
revdbf	Revision number of the force database file. Integer scalar (input)
output_paramete r	Requested output parameter. Integer scalar (input). See Table of Outputs for more details.
dof	Requested output parameter degree of freedom. Integer scalar (input). See Table of Outputs in the GetRequestedHistory function for more details.
node_no	Node number. Integer scalar (input/output). This may also be used to specify a guide surface, zero-gap guide or pipe-in-pipe connection number. If a node label is specified and a value of zero is specified for this variable, then the actual node number designated by the label is returned here.
element_no	Element number. Integer scalar (input).
local_node_no	Element local node number. Integer scalar (input).
stored_timesteps	Number of outputted time-steps to the database. Integer scalar (input). Call GetDatabaseInfo function to retrieve this value.

timestep_index	The time-step index at which the output is requested. Integer scalar (input).
single_value	Output parameter value. Real scalar (output).
label	Node label. Character string of unspecified length (input). This is relevant only if the node number was specifed as zero otherwise the node number specified takes precedence.

RETURN VALUE

Value	Description
0	Successful. History is returned
A positive value	Errors occured. Please call the GetRequestedSingleValueError function to check which errors occured.

Returns the type of error/errors that occured during the last call to GetRequestedSingleValue Function.

SYNTAX

```
Value = GetRequestedSingleValueError(
  ierror,
  maxerrors,
  errors)
```

DESCRIPTION OF ARGUMENTS

Argument	Description

ierror	Return value of GetRequestedSingleValue Function. Integer scalar (input)
maxerrors	Dimension of array errors. Integer scalar (input)
errors	Array holding flags to indicate if a particular error occured or not. Integer array of dimensions (maxerrors) (output). 0 – error did not occur 1 – error occured. The items in the array are listed in the following order:

Index	Description
1	1st argument of GetRequestedSingleValue is not a number.
2	2nd argument of GetRequestedSingleValue is not a number.
3	3rd argument of GetRequestedSingleValue is not a number or is out of range.
4	4th argument of GetRequestedSingleValue is not a number or is out of range.
5	5th argument of GetRequestedSingleValue is not a number or is out of range.
6	6th argument of GetRequestedSingleValue is not a number or is out of range.
7	7th argument of GetRequestedSingleValue is not a number or is out of range.

	8	8th argument of GetRequestedSingleValue is not a number or is out of range.
	9	9th argument of GetRequestedSingleValue is not a number or is out of range.
	10	Reserved.
	11	11th argument of GetRequestedSingleValue: unknown label or unspecified label when both the node number is zero or not a number.
	12	Requested output parameter is not in the database.
	13	Requested node is not in the database.
	14	Requested element is not in the database.
	15	Could not read from database.

RETURN VALUE

Value	Description
-1	Failed. Return value of GetRequestedSingleValue (ierror) is negative
0	Successful.
A positive value	Failed – maxerrors is to small. Call the function again substituting the returned value of the function in variable maxerrors.

Motion Database File Structure

HEADER SECTION (COMMON TO BOTH MOTION AND FORCE DATABASE FILES)

Part 1: Index

Revision Block, Always written

Start Record=1, Block Length=1, Structure Length=1

Record Layout	Description
DBM/DBF Revision [int]	The two revision numbers can be different. 1-V797 and earlier, 2-is skipped, 3-V811 and later
Append data [int]	0-do not append results, 1-append results.
Fixed Time Step [int]	0-No, 1-Yes
Major Version No. [int]	Major Version Number (e.g. for Version 7.9.3, this is 7)
Minor Version No. [int]	Minor Version Number (e.g. for Version 7.9.3, this is 9)
Maintenance Version No. [int]	Maintenance Version Number (e.g. for Version 7.9.3, this is 3)
Month of Release [int]	Month of Release 1-January, 2-February, … and 12-December

Year of Release [int]	Year of Release (e.g. 2009 is stored as 2009)

Information Block, Always written

This record provides high level information to the header.

Start Record=2

Record Layout	Description
No. data blocks [int]	The total number of data blocks contained in the header.
No. of index records [int]	The total number of index records including this one and the Revision Block in the header.
<Expected Empty>	
<Expected Empty>	
<Expected Empty>	
<Expected Empty>	
<Expected Empty>	
<Expected Empty>	

Start Record=3, End Record=No. of index records. The data is written in compact form until the start records of all data blocks are written.

Record Layout	Description
Start Record for Block A [int]	First record of Block A
Start Record for Block B [int]	First record of Block B
Start Record for Block C [int]	First record of Block C
Start Record for Block D [int]	First record of Block D
Start Record for Block E [int]	First record of Block E
Start Record for Block F [int]	First record of Block F
Start Record for Block G [int]	First record of Block G
Start Record for Block H [int]	First record of Block H

Record Layout	Description
Start Record for Block I [int]	First record of Block I
Start Record for Block J [int]	First record of Block J
Start Record for Block K [int]	First record of Block K
Start Record for Block K1 [int]	First record of Block K1
Start Record for Block L [int]	First record of Block L
Start Record for Block M [int]	First record of Block M
Start Record for Analysis Title Block [int]	First record of Analysis Title Block
Start Record for Element Data Block [int]	First record of Element Data Block

Record Layout	Description
Start Record for Material Properties Block [int]	First record of Material Properties Block
Start Record for Nonlinear E Tables Data Block [int]	First record of Nonlinear E Tables Data Block
Start Record for Node Data Block [int]	First record of Node Data Block
Start Record for Auxiliary Elements Block [int]	First record of Auxiliary Elements Block
Start Record for Auxiliary Nodes Block [int]	First record of Auxiliary Nodes Block
Start Record for Seabed Data Block [int]	First record of Seabed Data Block
Start Record for Wave Harmonics Block [int]	First record of Wave Harmonics Block
Start Record for Wave Harmonics for the DSP Block [int]	First record of Wave Harmonics for the DSP Block

Record Layout	Description
Start Record for Contact Diameters Block [int]	First record of Contact Diameters Block
Start Record for Element Sets Data Block [int]	First record of Element Sets Data Block
Start Record for Node Sets Data Block [int]	First record of Node Sets Data Block
Start Record for Labels Data Block [int]	First record of Labels Data Block
Start Record for Point Masses Data Block [int]	First record of Point Masses Data Block
Start Record for Point Loads Data Block [int]	First record of Point Loads Data Block
Start Record for Boundary Conditions Block [int]	First record of Boundary Conditions Block
Start Record for Current Profile Block [int]	First record of Current Profile Block

Record Layout	Description

Start Record for Pipe-In-Pipe Block [int]	First record of Pipe-In-Pipe Block
Start Record for Guide Surfaces Data Block [int]	First record of Guide Surfaces Data Block
Start Record for Panel Data Block [int]	First record of Panel Data Block
Start Record for Zero-Gap Guides Block [int]	First record of Zero-Gap Guides Block
Start Record for Installation Criteria Block [int]	First record of Installation Criteria Block
Start Record for Floating Body Data Block [int]	First record of Floating Body Data Block
Start Record for Vessel Data Block [int]	First record of Vessel Data Block
Start Record for Stinger Block [int]	First record of Stinger Block

Record Layout	Description

Start Record for Target Spectrum Block [int]	First record of Target Spectrum Block
Start Record for Distributed Loads Block [int]	First record of Distributed Loads Block
Start Record for Point Buoys Block [int]	First record of Point Buoys Block
Start Record for Vessel Profiles Block [int]	First record of Vessel Profiles Block
Start Record for Body Profiles Block [int]	First record of Body Profiles Block
Start Record for Colour Contour Ranges Block [int]	First record of Colour Contour Ranges Block
Start Record for Defined Colour Data Block [int]	First record of Defined Colour Data Block
Start Record for General and Plasticity related Outputs Block [int]	First record of General and Plasticity related Outputs Block

Record Layout	Description

Start Record for Plastic Hardening General Data Block [int]	First record of Plastic Hardening General Data Block
Start Record for Plastic Isotropic Hardening Data Block [int]	First record of Plastic Isotropic Hardening Data Block
Start Record for the List of Damper Elements [int]	First record of the List of Damper Elements
Start Record for Binary Data Block [int]	First record of Binary Data Block
<Expected Empty>	
<Expected Empty>	
<Expected Empty>	
<Expected Empty>	

The last record is padded with zeros, as necessary.

Part 2: Data blocks

Please note these blocks are only referred by a letter only when they cannot be described by functionality. The start of each block is based on the information contained in the index, described in Part 1.

Block A, Always written

Block Length=1, Structure Length=1

Record Layout	Description
GUID	Stores the GUID in the first 16bytes of the record.

Block B, Always written

Block Length=1, Structure Length=1

Record Layout	Description
Analysis in progress flag [int]	1 = in progress, 0 for not in progress
First auto-generated node [int]	Auxiliary node number indicating the first automatically generated node
First auto-generated element [int]	Auxiliary element number indicating the first automatically generated element
First auto-generated panel [int]	Auxiliary panel number indicating the first automatically generated panel
No. of vessel profiles [int]	Number of vessels under *VESSEL,INTEGRATED that have a profile specified

No. of body profiles [int]	Number of bodies specified under *BODY,INTEGRATED
No. of binary data blocks[int]	Number of data blocks defined under *BINARYDATA
No. of binary data records[int]	Number of records needed to store all the binary data information

Block C, Always written

Block Length=1, Structure Length=1

Record Layout	Description
No. records required for the header [int]	Length in records of the database header. This is the same for both files.
No. records required for each time slice DBM [int]	Length in records of each time slice in the motion database file.
No. records required for each time slice DBF [int]	Length in records of each time slice in the force database file.
Run-time Statistics Flag [int]	1-Run-time statistics outputted, 0-Not outputted

No records required for the motion run-time statistics [int]	Length in records of the motion run-time statistics section.
No records required for the force run-time statistics [int]	Length in records of the force run-time statistics section.
Max. No. Panels [int]	
No. Panels [int]	

Block D, Always written

Block Length=1, Structure Length=1

Record Layout	Description
Analysis Start Time [real]	
Analysis Finish Time [real]	
Ramp Time [real]	Loads are ramped over this time.
Database Time Step [real]	Interval at which time slices are written

Time Step [real]	This is the time step (increment) in the analysis time. Flexcom solves at every time step.
Start Time for Run-time Statistics [real]	Run-time statistics can be created in addition to the database outputs, starting from different time index. They share the same file with the database results.
No. Elements in Model [int]	Refers to beam elements.
Total number of Stress Strain Points [int]	

Block E, Always written

Block Length=1, Structure Length=1

Record Layout	Description
Acceleration due to Gravity [real]	
Water Depth [real]	
Water Density [real]	
No. Guides [int]	This is the number of flat guides + number of articulated stinger guides.

No. ZGG Guides [int]	
No. PIP Connections [int]	No. of Pipe-In-Pipe Connections
No. PIP sections [int]	No. of Pipe-In-Pipe Sections
No. Clashing Regions [int]	

Block F, Always written

Block Length=1, Structure Length=1

Record Layout	Description
No. Elements in Database [int]	No. of elements selected for database output.
No. Nodes in Database [int]	This is derived from above from the elements in the database.
No. Nodes per Element [int]	This is always equal to two
No. Co-ordinates [int]	This is always equal to three (X,Y,Z)
No. DOF per Node [int]	This is always equal to six.

No. Nodes with BCs [int]	The total number of nodes assigned boundary conditions.
No. element sets [int]	
No. Node Sets [int]	

Block G, Always written

Block Length=1, Structure Length=1

Record Layout	Description
No. Time Slices [int]	No of Time Slices in the Database
Outputted by dedicated thread [int]	0 – File written by the main thread; 1 – File written by the dedicated thread
Keep the rest empty	Keep the rest empty

Block H, Always written

Block Length=1, Structure Length=1

Record Layout	Description
Static Analysis Flag [int]	1-Static analysis, 0-otherwise.
Sea Type Flag [int]	0-No Wave Defined, 1-Regular Wave, 2-Random Sea
Seabed Flag [int]	0-Not Defined or Horizontal, 1-Sloping, 3-Arbitrary, 4-2D Linear, 5-2D Cubic Bessel, 6-2D Cubic Spline, 7-3D Linear, 8-3D Cubic.
No. points defining the seabed [int]	0-No seabed defined, 2-Horizontal or Sloping Seabed, >2-Arbitrary seabed.
No. Waves Harmonics [int]	Total number of wave harmonics in the model (from sea spectra and regular waves)
Wave Direction [real]	The wave direction or the dominant direction in degrees.

No. Vessels in the Model [int]	
No. Sea States [int]	Total Number of sea states defined in the model. A regular wave is a sea state, too.

Block I, Always written

Block Length=1, Structure Length=1

Record Layout	Description
Motions Outputted Flag [int]	1-Outputted, 0-Not outputted
Velocities Outputted Flag [int]	1-Outputted, 0-Not outputted
Accelerations Outputted Flag [int]	1-Outputted, 0-Not outputted
Reactions Outputted Flag [int]	1-Outputted, 0-Not outputted
Axial Force Outputted Flag [int]	1-Outputted, 0-Not outputted
Local Shear-Y Outputted Flag [int]	1-Outputted, 0-Not outputted

Local Shear-Z Outputted Flag [int]	1-Outputted, 0-Not outputted
Torque Outputted Flag [int]	1-Outputted, 0-Not outputted

Block J, Always written

Block Length=1, Structure Length=1

Record Layout	Description
Local Y-Bending Moment Outputted Flag [int]	1-Outputted, 0-Not outputted
Local Z-Bending Moment Outputted Flag [int]	1-Outputted, 0-Not outputted
Effective Tension Outputted Flag [int]	1-Outputted, 0-Not outputted
Local Y-Curvature Outputted Flag [int]	1-Outputted, 0-Not outputted
Local Z-Curvature Outputted Flag [int]	1-Outputted, 0-Not outputted
Local Axial Strain Outputted Flag [int]	1-Outputted, 0-Not outputted
Temperature Outputted Flag [int]	1-Outputted, 0-Not outputted
Pressure Outputted Flag [int]	1-Outputted, 0-Not outputted

Block K, Always written

Block Length=1, Structure Length=1

Record Layout	Description
No. auxiliary nodes [int]	
No. auxiliary elements [int]	
No. Point Masses [int]	
No. Point Loads [int]	
No. Points in the Current Profile [int]	0-no current defined or defined in a user DLL, 1-Uniform current, >1-Piecewise linear current.
No. FBs [int]	No. of floating bodies.
No. criteria parameters [int]	No. of criteria parameters to be monitored.
No. Labels [int]	

Block K1, More Model Dimensions and Flags, Always written

Block Length=1, Structure Length=1

Record Layout	Description
Number of Damper Elements in the Database [int]	The number of damper elements for which database output was requested.
Damper Power Outputted Flag [int]	1-Outputted, 0-Not outputted
Vessel Derivatives Output Flag [int]	1-Outputted, 0-Not outputted
Number of output points per element [int]	The number of output points per element for which database output was requested.
Convected Axes Output Flag [int]	1-Outputted, 0-Not outputted

Block L, Always written

Block Length=1, Structure Length=1

Record Layout	Description
No. auxiliary bodies [int]	
No. Waves Harmonics for DSP [int]	Reduced number of wave harmonics in the model (from sea spectra and regular waves) for use of Dynamic Display.
No. Distributed Loads [int]	
No. Point Buoys [int]	
Author of this file [int]	0 – Flexcom; 1 – Modes.
Quasi-Static Analysis Flag [int]	1 – Quasi-Static Analysis; 0 - Otherwise
Total CPU time [real]	Computation time (in hours) taken to perform analysis
Auxiliary Element Colour Flag	0 – No; 1 - Yes

Block M, Always written

Block Length=1, Structure Length=1

Record Layout	Description
No. Target Spectrum Divisions [int]	0 – No target spectrum is available.
Colour Contour Flag [int]	0 – No; 1 - Yes
No. colours in defined colour table [int]	
Unit System used for analysis [int]	1 – Metric; 2 – Imperial; 3 – User-defined
Extension of original keyword file [int]	1 – keyx; 2 – keyxm; 3 – keyxi
Seabed Direction [real]	Seabed orientation measured in degrees counter-clockwise about the global X axis
No. Colour Contour Ranges [int]	No. of colour contour ranges available for Model View.

Block Analysis Title, Always written

Block Length=3, Structure Length=1

Record Layout	Description
Analysis Title [char1:28]	The maximum no. of characters stored per record is 32. This entry takes the first 28bytes of Record 12. Rest of the Record is empty.

Record Layout	Description
Analysis Title [char29:56]	This entry takes the first 28bytes of Record 13. Rest of the Record is empty.

Record Layout	Description
Analysis Title [char57:80]	This entry takes the first 24bytes of Record 14. Rest of the Record is empty.

Block Element Data, Always written

Block Length=No. Elements in Database, Structure Length=3

Record Layout	Description
Internal element no. [int]	The internal numbering system runs continuously, from 1 to total number of elements in the database.
User element no. [int]	As specified by the user
No. of first node of the element [int]	This is in internal numbering.
No. of second node of the element [int]	This is in internal numbering.
Element length [real]	
Effective external diameter [real]	Effective has the meaning of used in stress calculations.
Effective internal diameter [real]	
Colour Index [int]	Colour code for the Dynamic Display; this is also the Geometric set No.

Record Layout	Description

Internal Fluid Height [real]	
Internal Fluid Density [real]	
Internal Fluid Pressure [real]	
Bending Stiffness [real]	
User Defined Stress Properties Flag [int]	0-Element does not have assigned user stress properties; 1- Element has assigned user stress properties
Effective Cross-Section Area [real]	
Second Moment of Area-YY [real]	
Second Moment of Area-ZZ [real]	

Record Layout	Description

Element Thickness [real]	
Space reserved for external fluid.	

Block Material Properties, Always written

Block Length= No. Elements in Database, Structure Length=1

Record Layout	Description
Internal element No. [int]	
Nonlinear Material Table No. [int]	0-Linear or Only Nonlinear axial or torsion stiffness, the number of the Direct Stress Strain table otherwise.

Axial stiffness or the cross-section area [real]	This entry is the axial stiffness if this a rigid linear material, or the cross-section area (as specified under *GEOMETRIC SETS) if this is a rigid nonlinear material. In all other instances, this entry is zero.
Material Type [int]	0-Linear, 1-Nonlinear Young's Modulus, 2-Nonlinear bending stiffness, 3-Only Nonlinear axial or torsion stiffness
Poisson's Ratio [real]	Applicable to all material models.
Young's Modulus [real]	Only applicable to plastic strain hardening models. This value is zero for all other models.

Block Nonlinear E Tables Data, Written if Total number of Stress Strain Points > 0

Block Length= Total number of Stress Strain Points, Structure Length=1

Record Layout	Description
Table No. [int]	

No of Points [int]	
Strain [real]	This is the calculated strain that is used in the resultant nonlinear axial stiffness (EA curve)
Force [real]	This is the calculated force that is used in the resultant nonlinear axial stiffness (EA curve)

Block Node Data, Always written

Block Length=No. nodes in the database, Structure Length = 1

Record Layout	Description
Internal node No. [int]	See description above for element no.

Node Co-ordinate X [real]	As specified by user or from the Cable PreStatic Step.
Node Co-ordinate Y [real]	As specified by user or from the Cable PreStatic Step.
Node Co-ordinate Z [real]	As specified by user or from the Cable PreStatic Step.
User node no. [int]	
Restrained node flag [int]	0-No boundary conditions (BCs) specified at this node, Internal node No.-Boundary conditions specified. Important: The data written here is compacted. This means that the Internal node No, written here may not be the same as the value written at Position 1. E.g. If I have only one BC for Node 43 in the whole model, the number 43 will be written here in the very first record of this block.

Block Auxiliary Elements, Written if No. auxiliary elements>0

Block Length=2*No. auxiliary elements, Structure Length=2

Record Layout	Description
Auxiliary Element Internal No. [int]	
Auxiliary Element User No. [int]	
No. of first node of the element [int]	Internal numbering
No. of second node of the element [int]	Internal numbering
Element length [real]	=0.0
Effective external diameter [real]	=0.0
Effective internal diameter [real]	=0.0
Body No. Associated with Element [int]	

Record Layout	Description
Panel No. Associated with Element [int]	

Body Type [int]	Values of 0,1,2 correspond to None, Vessel, Node
Colour Index [int]	Colour code for the Dynamic Display {Values of 15-16}

Block Auxiliary Nodes, Written if No. auxiliary nodes>0

Block Length=No. auxiliary nodes, Structure Length=1

Record Layout	Description
Auxiliary Node User No. [int]	
Node Co-ordinate X [real]	As specified by user.
Node Co-ordinate Y [real]	As specified by user.

Node Co-ordinate Z [real]	As specified by user.

Block Seabed, Always written

Block Length= [No. Seabed Points * 2 /8] + 1, No repeating structure: seabed points are written compactly.

If no seabed, 2D seabed or 3D seabed is specified, one empty record is written.

If a horizontal seabed is specified:

Record Layout	Description
Point 1 Co-ordinate Y [real]	
Point 1 Co-ordinate X [real]	
Point 2 Co-ordinate Y [real]	
Point 2 Co-ordinate X [real]	

If a sloping seabed is specified:

Note that rotation to account for direction has not been applied to profile points.

Record Layout	Description
Point 1 Co-ordinate Y [real]	
Point 1 Co-ordinate X [real]	
Point 2 Co-ordinate Y [real]	
Point 2 Co-ordinate X [real]	
Slope [real]	Seabed slope measured in degrees counter-clockwise about the seabed local Z axis

If an arbitrary seabed is specified:

Record Layout	Description
Point 1 Co-ordinate Y [real]	
Point 1 Co-ordinate X [real]	
Point 2 Co-ordinate Y [real]	
Point 2 Co-ordinate X [real]	
…	
…	
…	
…	

The record is repeated until all points are written.

Block Wave Harmonics, Written if No. Waves Harmonics>0

Block Length= No. Wave Harmonics. Structure Length=1

Record Layout	Description
Wave Amplitude [real]	
Wave Number [real]	This is the scientific wave number, not its index in the array
Circular velocity [real]	Radians/sec
Phase Angle [real]	Radians
Wave Direction [real]	Degrees

Block Wave Harmonics for the DSP, Written if No. DSP Waves Harmonics>0

Block Length= No. DSP Wave Harmonics. Structure Length=1

Record Layout	Description
DSP Wave Amplitude [real]	
DSP Wave Number [real]	This is the scientific wave number, not its index in the array
DSP Circular velocity [real]	Radians/sec
DSP Phase Angle [real]	Radians
DSP Wave Direction [real]	Degrees

Block Contact Diameters, Always written

Block Length= [No. elements in database / 8]+1. Data is written in compact form.

Record Layout	Description
Contact Diameter Elem.1 [real]	
Contact Diameter Elem.2 [real]	
Contact Diameter Elem.3 [real]	
…	
…	
…	
…	

…	

Block Element Sets, Always written

Block length=No. element sets*(No. elements in database +9), Structure Length=No. elements in database + 9

Record Layout	Description
Element Set Name [char1:32]	

Record Layout	Description
Element Set Name [char33:64]	

Record Layout	Description
Element Set Name [char65:96]	

Record Layout	Description
Element Set Name [char98:128]	

Record Layout	Description
Element Set Name [char129:160]	

Record Layout	Description
Element Set Name [char161:192]	

Record Layout	Description

Element Set Name [char193:224]	

Record Layout	Description
Element Set Name [char225:255]	

Record Layout	Description
No. elements in the set and requested for database output [int]	This number can be less than the actual number of elements defined in the set, as some elements may not be included in the database output.

This record is repeated based on the number of elements above.

Record Layout	Description
User element no. [int]	

Block Node Sets, Always written

Block Length= No. node sets*(No. nodes in database +9) Structure Length=No. nodes in database + 9

Record Layout	Description
Node Set Name [char1:32]	

Record Layout	Description
Node Set Name [char33:64]	

Record Layout	Description
Node Set Name [char65:96]	

Record Layout	Description

Node Set Name [char98:128]	

Record Layout	Description
Node Set Name [char129:160]	

Record Layout	Description
Node Set Name [char161:192]	

Record Layout	Description
Node Set Name [char193:224]	

Record Layout	Description

Node Set Name [char225:255]	

Record Layout	Description
No. nodes in the set for database output [int]	This number can be less than the actual number of nodes defined in the set, as some elements may not be included in the database output.

This record is repeated based on the number of nodes above.

Record Layout	Description
User Node no. [int]	

Block Labels, Written if No. labels > 0

Block Length=No. labels * 9, Structure Length=No. labels

Record Layout	Description
Label Name [char1:32]	

Record Layout	Description
Label Name [char33:64]	

Record Layout	Description
Label Name [char65:96]	

Record Layout	Description
Label Name [char98:128]	

Record Layout	Description
Label Name [char129:160]	

Record Layout	Description
Label Name [char161:192]	

Record Layout	Description
Label Name [char193:224]	

Record Layout	Description
Label Name [char225:255]	

Record Layout	Description
Element/Node No. [int]	
Label Type [int]	Note that 1=Node, 2=Element

Block Point Masses, Written if No. Point Masses>0, to preserve the numerical record format set Point Masses Flag =1 if condition is true, and zero otherwise.

Block Length= Point Masses Flag *[No. Point Masses / 8]+1, information is written compactly until exhausted.

Record Layout	Description
User Node No [int] User Node No [int] User Node No [int]	User Node No. associated with a Point Mass.

…	
…	
…	
…	
…	

Block Point Loads, Written if No. Point Loads>0

Block Length=No. Point Loads, Structure Length=1.

Record Layout	Description
User Node No. [int]	Node at which the load is applied
Degree of Freedom [int]	DOF at which the load is applied
Load Value [real]	

Block Boundary Conditions, Written if No. Nodes with BCs >0, to preserve the numerical record format set No. Nodes with BCs Flag =1 if condition is true, and zero otherwise.

Block Length= No. Nodes with BCs Flag*[No. Nodes with BCs/8]+1, Information is compacted.

Record Layout	Description
User Node No. [int]	The number of a node, which has assigned a boundary condition.
User Node No. [int]	The number of a node, which has assigned a boundary condition.
User Node No. [int]	The number of a node, which has assigned a boundary condition.
User Node No. [int]	The number of a node, which has assigned a boundary condition.
…	

…	
…	
…	

Block Current Profile, Written if No. Points in the Current Profile >0

BlockLength= No. Points in the Current Profile, Structure Length=1

If one point

Record Layout	Description
Velocity [real]	
Direction [real]	

If more than one point

Record Layout	Description
Depth [real]	
Velocity [real]	
Direction [real]	

Block Pipe-In-Pipe, Written if No. PIP sections >0

BlockLength= No. elements in model*No. PIP sections, Structure Length= 1

Record Layout	Description
Internal Outer Element No[int]	This structure is repeated over the whole No. of Elements for Each PIP Section.

Internal Inner Element No[int]	

Block Flat and Articulated Guide Surfaces, Written if No. Guides >0, Set No. Guides Flag =1 if condition is true, and zero otherwise.

Block Length= No. Guides*11, Structure1 Length= 3 Structure2 Length=8

Structure 1 is repeated for each guide. The order the guides appear is: Vessel Driven followed by Node Driven

Record Layout	Description
Shape Flag [int]	1 – Flat; 2 - Cylindrical
Contact Node Set No. [int]	

Surface Height/Length [real]	Height for flat guide; Length for cylindrical guide
Surface Width/Radius [real]	Width for flat guide; Radius for cylindrical guide
Negative Local Y Axis Extended Flag [int]	1- Negative Local Y Axis Extended, 0-Not Extended
Positive Local Y Axis Extended Flag [int]	1- Positive Local Y Axis Extended, 0-Not Extended
Auxiliary Surface Flag [int]	1- Auxiliary Surface, 0-Not Auxiliary Surface
Initial Origin X [real]	

Record Layout	Description
Initial Origin Y [real]	
Initial Origin Z [real]	

Initial Axis 1 X [real]	
Initial Axis 1 Y [real]	
Initial Axis 1 Z [real]	
Initial Axis 2 X [real]	
Initial Axis 2 Y [real]	
Initial Axis 2 Z [real]	

Record Layout	Description
Thickness [real]	
Starting Angle [real]	Cylindrical guides only

Subtended Angle [real]	Cylindrical guides only

Structure 2 is repeated for all surfaces, after we are done with Structure 1.

Record Layout	Description
Surface Name [char1:32]	

Record Layout	Description
Surface Name [char33:64]	

Record Layout	Description
Surface Name [char65:96]	

Record Layout	Description
Surface Name [char98:128]	

Record Layout	Description
Surface Name [char129:160]	

Record Layout	Description
Surface Name [char161:192]	

Record Layout	Description
Surface Name [char193:224]	

Record Layout	Description
Surface Name [char225:255]	

Block Panel Data, Written if Max. No. Panels >0

Block Length= Max. No. Panels, Structure Length=1

Record Layout	Description
Panel Node No [int]	A panel is a triangular surface. It needs three nodes to be defined.
Panel Node No [int]	
Panel Node No [int]	

Panel Colour [int]	
Associated Body No. [int]	
Panel No. [int]	
Body Type [int]	Values of 0,1,2 correspond to None, Vessel, Node

Block Zero Gap Guides (ZGG), Written if No. ZGG >0, Set No. ZGG Flag =1 if condition is true, and zero otherwise.

Block Length=No. ZGG, Structure Length=1

Record Layout	Description
ZGG Contact Node Set No. [int]	
ZGG Height [real]	
ZGG Initial Origin X [real]	

ZGG Initial Origin Y [real]	
ZGG Initial Origin Z [real]	
ZGG Initial Axis X [real]	
ZGG Initial Axis Y [real]	
ZGG Initial Axis Z [real]	

Block Installation Criteria, Written if No. criteria parameters >0, Set No. criteria parameters Flag =1 if condition is true, and zero otherwise.

Block Length= No. criteria parameters*9+1 Structure1 Length=1, Structure2 Length=9

Structure 1 is written only one.

Record Layout	Description
Criteria Option [int]	

Criteria Vector No. [int]	
Criteria DOF [int]	
Criteria Adjustment No. [int]	
Criteria Max. Iterations [int]	

Structure 2 is written for each criterion.

Record Layout	Description
Criterion Type [int]	
Criterion Min Value [real]	

Criterion Max Value [real]	

Record Layout	Description
Name of Area where Criterion Applies [char1:32]	

Record Layout	Description
Name of Area where Criterion Applies [char33:64]	

Record Layout	Description
Name of Area where Criterion Applies [char65:96]	

Record Layout	Description
Name of Area where Criterion Applies [char98:128]	

Record Layout	Description
Name of Area where Criterion Applies [char129:160]	

Record Layout	Description
Name of Area where Criterion Applies [char161:192]	

Record Layout	Description
Name of Area where Criterion Applies [char193:224]	

Record Layout	Description
Name of Area where Criterion Applies [char225:255]	

Block Floating Body, Written if No. FBs >0

Block Length= No. FBs*4, Structure Length=4

Record Layout	Description
FB COG Internal Node No. [int]	

Record Layout	Description

FB Name[char1-32]	

Record Layout	Description
FB Heading [real]	

Record Layout	Description
FB COG User Node No. [int]	

Block Vessel Data, Written if No. Vessels in the Model >0

Block Length= No. Vessels in the Model*9, Structure Length=9

First Structure – Structures written consecutively

Record Layout	Description
Vessel Name [char1:32]	

Record Layout	Description
Vessel Name[char33:64]	

Record Layout	Description
Vessel Name[char65:96]	

Record Layout	Description
Vessel Name[char98:128]	

Record Layout	Description
Vessel Name[char129: 160]	

Record Layout	Description
Vessel Name[char161: 192]	

Record Layout	Description
Vessel Name[char193: 224]	

Record Layout	Description

Vessel Name[char225:255]	

Second Structure

Record Layout	Description
Vessel Initial Origin X [real]	
Vessel Initial Origin Y [real]	
Vessel Initial Origin Z [real]	
Vessel Initial Yaw [real]	Radians
Vessel Initial Roll [real]	Radians

Vessel Initial Pitch [real]	Radians
Vessel Angles Theory Flag [int]	1-Small Angles, 0-Large Angles

Block Stinger Plane Data, Always Written

Block Length=1, Structure Length=1

Record Layout	Description
X Axis Component 1 [real]	Components 1-3 defining the X axis
X Axis Component 2 [real]	
X Axis Component 3 [real]	

Y Axis Component 1 [real] Y Axis Component 2 [real] Y Axis Component 3 [real]	Components 1-3 defining the Y axis

Block Target Spectrum, Written if No. Target Spectrum Divisions > 0; if this is true then set Target Spectrum Flag = 1 otherwise = 0

Block Length=[(No. target divisions+1)*2/8]+1, Structure Length - data is written compactly

Record Layout	Description
Point1 Frequency [real]	

Point1 Energy [real]	
Point2 Frequency [real]	
Point2 Energy [real]	
…	
…	
…	
…	

Block Distributed Loads, Written if No. Distributed Loads > 0

Block Length= No. Distributed Loads, Structure Length=1

Record Layout	Description
User Element Number Start [int]	The load is applied from a “start” element to a “end” element

User Element Number End [int]	
DOF [int]	
Load Value [real]	

Block Point Buoys, Written if No. Point Buoys > 0

Block Length= No. Point Buoys, Structure Length=1

Record Layout	Description
User Node Number [int]	
Total Buoyancy [real]	

Total Weight [real]	

Block Vessel Profiles, Written if No. of vessel profiles > 0

Block Length= No. Of Vessel Profiles, Structure Length=2

Record Layout	Description
Centre of Profile component 1 [real]	
Centre of Profile component 2 [real]	
Centre of Profile component 3 [real]	

Height of Profile [real]	
Length of Profile [real]	
Width of Profile [real]	
X-Axis component 1 [real]	
X-Axis component 2 [real]	

Record Layout	Description
X-Axis component 3 [real]	
Y-Axis component 1 [real]	
Y-Axis component 2 [real]	

Y-Axis component 3 [real]	
Z-Axis component 1 [real]	
Z-Axis component 2 [real]	
Z-Axis component 3 [real]	

Block Body Profiles, Written if No. of Body Profiles > 0

Block Length= No. of Body Profiles, Structure Length=2

Record Layout	Description
Centre of Profile component 1 [real]	
Centre of Profile component 2 [real]	

Centre of Profile component 3 [real]	
Height of Profile [real]	
Length of Profile [real]	
Width of Profile [real]	
X-Axis component 1 [real]	
X-Axis component 2 [real]	

Record Layout	Description
X-Axis component 3 [real]	
Y-Axis component 1 [real]	

Y-Axis component 2 [real]	
Y-Axis component 3 [real]	
Z-Axis component 1 [real]	
Z-Axis component 2 [real]	
Z-Axis component 3 [real]	

Block Stress Colour Contour Ranges, Contains valid data if Colour Contour Flag =1

Block Length= [No. Elements in Database *No. Colour Contour Ranges * 2 / 8]+1; Data is written compactly.

Record Layout	Description
Element n, Axial force Min [real]	
Element n, Axial force Max [real]	

Element n, Local y-shear force Min [real]	
Element n, Local y-shear force Max [real]	
Element n, Local z-shear force Min [real]	
Element n, Local z-shear force Max [real]	
Element n, Torque Min [real]	
Element n, Torque Max [real]	

Record Layout	Description
Element n, Local y-bending moment Min [real]	
Element n, Local y-bending moment Max [real]	
Element n, Local z-bending moment Min [real]	

Element n, Local z-bending moment Max [real]	
Element n, Effective tension Min [real]	
Element n, Effective tension Max [real]	
Element n, Local y-Curvature Min [real]	
Element n, Local y-Curvature Max [real]	

Record Layout	Description
Element n, Local z-Curvature Min [real]	
Element n, Local z-Curvature Max [real]	
Element n, Local Axial Strain Min [real]	

Element n, Local Axial Strain Max [real]	
Element n, Internal Pressure Min [real]	
Element n, Internal Pressure Max [real]	
Element n, External Pressure Min [real]	
Element n, External Pressure Max [real]	

Record Layout	Description
Element n, Plastic Axial Strain Min [real]	
Element n, Plastic Axial Strain Max [real]	
Element n, Plastic Local-y Curvature Min [real]	

Element n, Plastic Local-y Curvature Max [real]	
Element n, Plastic Local-z Curvature Min [real]	
Element n, Plastic Local-z Curvature Max [real]	
Element n, Eqv. Plastic Axial Strain Min [real]	
Element n, Eqv. Plastic Axial Strain Max [real]	

Record Layout	Description
Element n, Eqv. Plastic Curvature Min [real]	
Element n, Eqv. Plastic Curvature Max [real]	
Data is written compactly	

Block Defined Colour table

Block Length=[No. colours in defined colour table /8] + 1, No repeating structure: defined colours are written compactly.

Record Layout	Description
Colour 1 ARGB [int] Colour 2 ARGB [int] Colour 3 ARGB [int] Colour 4 ARGB [int]	

Block List of Damper Elements in the Database, Written if Number of Damper Elements in the Database > 0

Block Length= [Number of Damper Elements in the Database / 8] +1 , Structure Length=1 (Data are written compactly)

Record Layout	Description
Internal Element Number [int] …… Data are written compactly	The internal element number of the first damper element in the database.

Block Binary Data, Written if No. of binary data blocks > 0

Block Length= No. of binary data blocks, Structure Length= 1+ No. of data stream records

Record Layout	Description
Block Type[int]	1 – Vessel Profile, 2 – Body Profile, 3 – Seabed Data File
Key[int]	

Length of data stream [int]	

Record Layout	Description
Data[char1:Length of data stream]	

Data that contains more than 32 bytes will span more than 1 record. No of data stream records = [(Length of data stream-1)/32] + 1

RUN-TIME STATISTICS

Block 1-Run-Time Statistics, Written if Run-time Statistics Flag=1 and only if any kinematics have been requested (See flags for Motion to Acceleration).

Start Record= No records required for the header+1, Block Length= No records required for the motion run-time statistics, Structure Length=No. Nodes in Database*3, Data is written compactly until exhausted.

For each node included in the database and each DOF of that node, we write the minimum, maximum, mean and standard deviation of the respective kinematic.

The structure below is repeated to write, in order: Motion, Velocity and Acceleration.

Record Layout	Description
First Node in Database-DOF1-Min [real]	
First Node in Database-DOF1-Max [real]	
First Node in Database-DOF1-Mean [real]	
First Node in Database-DOF1-Std Dev [real]	
First Node in Database-DOF2-Min [real]	
First Node in Database-DOF2-Max [real]	

First Node in Database-DOF2-Mean [real]	
First Node in Database-DOF2-Std Dev [real]	

Record Layout	Description
First Node in Database-DOF3-Min [real]	
First Node in Database-DOF3-Max [real]	
First Node in Database-DOF3-Mean [real]	
First Node in Database-DOF3-Std Dev [real]	
First Node in Database-DOF4-Min [real]	
First Node in Database-DOF4-Max [real]	

First Node in Database-DOF4-Mean [real]	
First Node in Database-DOF4-Std Dev [real]	

Record Layout	Description
First Node in Database-DOF5-Min [real]	
First Node in Database-DOF5-Max [real]	
First Node in Database-DOF5-Mean [real]	
First Node in Database-DOF5-Std Dev [real]	
First Node in Database-DOF6-Min [real]	
First Node in Database-DOF6-Max [real]	

First Node in Database-DOF6-Mean [real]	
First Node in Database-DOF6-Std Dev [real]	

Record Layout	Description
Second Node in Database-DOF1-Min [real]	
Second Node in Database-DOF1-Max [real]	
Second Node in Database-DOF1-Mean [real]	
Second Node in Database-DOF1-Std Dev [real]	
Second Node in Database-DOF2-Min [real]	

Second Node in Database-DOF2-Max [real] Second Node in Database-DOF2-Mean [real] Second Node in Database-DOF2-Std Dev [real]	

TIME SLICE

Time Slice Start= No records required for the header+ No records required for the motion run-time statistics+(Time Step Index – 1)* No. records required for each time slice + 1, Time Slice Length = No. records required for each time slice

Block 1-Motions, Written if Motions Outputted Flag=1

Start Record= Time Slice Start, Block Length= [No. Nodes in Database * No. DOF per Node /8] + 1, Data is written compactly.

Record Layout	Description
Motion-First Node in Database-DOF1 [real] Motion-First Node in Database-DOF2 [real]	

Motion-First Node in Database-DOF3 [real]	
Motion-First Node in Database-DOF4 [real]	
Motion-First Node in Database-DOF5 [real]	
Motion-First Node in Database-DOF6 [real]	
Motion-Second Node in Database-DOF1 [real]	
Motion-Second Node in Database-DOF2 [real]	And continues with DOF3-6, then move to the next node and so on.

Block 2-Velocities, Written if Velocities Outputted Flag=1

Start Record= Time Slice Start+Motions Outputted Flag*([No. Nodes in Database * No. DOF per Node /8] + 1), Block Length= [No. Nodes in Database * No. DOF per Node /8] + 1, Data is written compactly.

Record Layout	Description
Velocity-First Node in Database-DOF1 [real]	

Velocity-First Node in Database-DOF2 [real]	
Velocity-First Node in Database-DOF3 [real]	
Velocity-First Node in Database-DOF4 [real]	
Velocity-First Node in Database-DOF5 [real]	
Velocity-First Node in Database-DOF6 [real]	
Velocity-Second Node in Database-DOF1 [real]	
Velocity-Second Node in Database-DOF2 [real]	And continues with DOF3-6, then move to the next node and so on

Block 3-Accelerations, Written if Accelerations Outputted Flag=1

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1), Block Length= [No. Nodes in Database * No. DOF per Node /8] + 1, Data is written compactly.

Record Layout	Description
Acceleration-First Node in Database-DOF1 [real]	
Acceleration -First Node in Database-DOF2 [real]	
Acceleration -First Node in Database-DOF3 [real]	

Acceleration -First Node in Database-DOF4 [real]	
Acceleration -First Node in Database-DOF5 [real]	
Acceleration -First Node in Database-DOF6 [real]	
Acceleration -Second Node in Database-DOF1 [real]	
Acceleration -Second Node in Database-DOF2 [real]	And continues with DOF3-6, then move to the next node and so on

Block 4-Flat and Articulated Stinger Guides, Written if No. Guides > 0

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag+ Accelerations Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1), Block Length= [No. Guides* 9/8] + 1, Data is written compactly, first for the Flat Guides and then for the Articulated Guides. There are no zeros between the two.

Record Layout	Description
Guide1-Origin-X [real]	Guide can be a flat/cylindrical, vessel/node driven.
Guide1-Origin-Y [real]	
Guide1-Origin-Z [real]	
Guide1-Axis1-Component1 [real]	

Guide1-Axis1-Component2 [real]	
Guide1-Axis1-Component3 [real]	
Guide1-Axis2-Component1 [real]	
Guide1-Axis2-Component2 [real]	

Record Layout	Description
Guide1-Axis2-Component3 [real]	Guide can be a flat/cylindrical, vessel/node driven.
Guide2-Origin-X [real]	
Guide2-Origin-Y [real]	
Guide2-Origin-Z [real]	
Guide2-Axis1-Component1 [real]	
Guide2-Axis1-Component2 [real]	
Guide2-Axis1-Component3 [real]	

Guide2-Axis2-Component1 [real]	

Block 5-Zero Gap Guides, Written if No. ZGG >0

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag+ Accelerations Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1) + No. Guides Flag *([No. Guides* 9/8] + 1), Block Length=[No. ZGG*6/8]+1, Data is written compactly.

Record Layout	Description
Guide1-Origin-X [real]	
Guide1-Origin-Y [real]	
Guide1-Origin-Z [real]	
Guide1-Axis-Component1 [real]	
Guide1-Axis-Component2 [real]	
Guide1-Axis-Component3 [real]	
Guide2-Origin-X [real]	
Guide2-Origin-Y [real]	

Record Layout	Description
Guide2-Origin-Z [real]	
Guide2-Axis-Component1 [real]	
Guide2-Axis-Component2 [real]	
Guide2-Axis-Component3 [real]	
…	
…	
…	
…	

Block 6-Contact Flags and Reactions, Always Written

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag+ Accelerations Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1) + No. Guides Flag *([No. Guides* 9/8] + 1) + No. ZGG Flag *([No. ZGG*6/8]+1), Block Length=[No. Nodes in Database*8/8]+1, Data is written compactly.

Record Layout	Description
First Node in Database-Seabed Contact Flag [int]	0-not in contact, 1-in contact
First Node in Database-Seabed Reaction X [real]	

First Node in Database-Seabed Reaction Y [real]	
First Node in Database-Seabed Reaction Z [real]	
First Node in Database-Surface No. [int]	0-not in contact with a surface, otherwise the contact surface no.
First Node in Database-Surface Reaction X [real]	
First Node in Database-Surface Reaction Y [real]	
First Node in Database-Surface Reaction Z [real]	
…	Everything is repeated for the Next node in database.

Block 7-Clashing Clearance Data, Written if No. Clashing Regions>0, Set Clashing Flag=1

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag+ Accelerations Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1) + No. Guides Flag *([No. Guides* 9/8] + 1) + No. ZGG Flag *([No. ZGG*6/8]+1)+[No. Nodes in Database*8/8]+1, Block Length=[No. Clashing Regions/8]+1, Data is written compactly.

Record Layout	Description
Clearance Region 1 [real]	
Clearance Region 2 [real]	
…	

…	
…	
…	
…	
…	

Block 8-Motion of Auxiliary Nodes, Written if No. Auxiliary Bodies>0, Set Auxiliary Flag=1

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag+ Accelerations Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1) + No. Guides Flag *([No. Guides* 9/8] + 1) + No. ZGG Flag *([No. ZGG*6/8]+1)+[No. Nodes in Database*8/8]+1+ Clashing Flag*([No. Clashing Regions/8]+1), Block Length=[No. auxiliary nodes * No. DOF per Node/8]+1, Data is written compactly.

Record Layout	Description
Auxiliary Node 1-DOF1 [real]	
Auxiliary Node 1-DOF2 [real]	
Auxiliary Node 1-DOF3 [real]	
Auxiliary Node 1-DOF4 [real]	
Auxiliary Node 1-DOF5 [real]	
Auxiliary Node 1-DOF6 [real]	

Auxiliary Node 2-DOF1 [real] …	

Block 9-Water Surface Elevation, Always Written

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag+ Accelerations Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1) + No. Guides Flag *([No. Guides* 9/8] + 1) + No. ZGG Flag *([No. ZGG*6/8]+1)+[No. Nodes in Database*8/8]+1+ Clashing Flag*([No. Clashing Regions/8]+1)+ Auxiliary Flag*([No. auxiliary nodes * No. DOF per Node/8]+1), Block Length=1, Structure Length=1

Record Layout	Description
Time [real] Water Elevation [real] Time Step [real] Ramp Value [real] Total Kinetic Energy [real]	

Block 10-Vessel Data, Written if No. Vessels in the Model>0

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag+ Accelerations Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1) + No. Guides Flag *([No. Guides* 9/8] + 1) + No. ZGG Flag *([No. ZGG*6/8]+1)+[No. Nodes in Database*8/8]+1+ Clashing Flag*([No. Clashing Regions/8]+1)+ Auxiliary Flag*([No. auxiliary nodes * No. DOF per Node/8]+1)+1, Block Length= No. Vessels in the Model, Structure Length=1

Record Layout	Description
Vessel COG-X [real]	Co-ordinates of the Vessel COG
Vessel COG-Y [real]	
Vessel COG-Z [real]	
Vessel Orientation-Component 1 [real]	Vessel Orientation
Vessel Orientation-Component 2 [real]	
Vessel Orientation-Component 3 [real]	

Block 11-Auxiliary Element Colour, Written if Auxiliary Element Colour Flag >0,

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag+ Accelerations Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1) + No. Guides Flag *([No. Guides* 9/8] + 1) + No. ZGG Flag *([No. ZGG*6/8]+1)+[No. Nodes in Database*8/8]+1+ Clashing Flag*([No. Clashing Regions/8]+1)+ Auxiliary Flag*([No. auxiliary nodes * No. DOF per Node/8]+1)+1 + No. Vessels in the Model, Block Length=[No. auxiliary elements /8]+1, Data is written compactly.

Record Layout	Description
Colour for auxiliary element 1 [int]	Colour code for the Dynamic Display {Values of 0-13}
Colour for auxiliary element 2 [int]	Colour code for the Dynamic Display {Values of 0-13}
…	…
…	…
…	…
…	…
…	…
…	…

Block 12-Vessel Profile Data, Written if No. of Vessel Profiles >0,

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag+ Accelerations Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1) + No. Guides Flag *([No. Guides* 9/8] + 1) + No. ZGG Flag *([No. ZGG*6/8]+1)+[No. Nodes in Database*8/8]+1+ Clashing Flag*([No. Clashing Regions/8]+1)+ Auxiliary Flag*([No. auxiliary nodes * No. DOF per Node/8]+1)+1 + No. Vessels in the Model +[No. auxiliary elements /8] +1, Block Length=No. Of Vessel Profiles, Structure Length=2.

Record Layout	Description
Centre of Profile component 1 [real]	
Centre of Profile component 2 [real]	

Centre of Profile component 3 [real]	
X-Axis component 1 [real]	
X-Axis component 2 [real]	
X-Axis component 3 [real]	
Y-Axis component 1 [real]	
Y-Axis component 2 [real]	

Record Layout	Description
Y-Axis component 3 [real]	
Z-Axis component 1 [real]	
Z-Axis component 2 [real]	
Z-Axis component 3 [real]	

Block 13-Body Profile Data, Written if No. of Body Profiles >0,

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag+ Accelerations Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1) + No. Guides Flag *([No. Guides* 9/8] + 1) + No. ZGG Flag *([No. ZGG*6/8]+1)+[No. Nodes in Database*8/8]+1+ Clashing Flag*([No. Clashing Regions/8]+1)+ Auxiliary Flag*([No. auxiliary nodes * No. DOF per Node/8]+1)+1 + No. Vessels in the Model +[No. auxiliary elements /8] +1 + 2 * No. Of Vessel Profiles, Block Length=No. Of Body Profiles, Structure Length=2.

Record Layout	Description
Centre of Profile component 1 [real] Centre of Profile component 2 [real] Centre of Profile component 3 [real] X-Axis component 1 [real] X-Axis component 2 [real] X-Axis component 3 [real] Y-Axis component 1 [real] Y-Axis component 2 [real]	

Record Layout	Description
Y-Axis component 3 [real] Z-Axis component 1 [real] Z-Axis component 2 [real] Z-Axis component 3 [real]	

Block 14-Vessel Derivatives, Written if No. Vessels in the Model > 0 and Vessel Derivatives Outputted Flag=1,

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag+ Accelerations Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1) + No. Guides Flag *([No. Guides* 9/8] + 1) + No. ZGG Flag *([No. ZGG*6/8]+1)+[No. Nodes in Database*8/8]+1+ Clashing Flag*([No. Clashing Regions/8]+1)+ Auxiliary Flag*([No. auxiliary nodes * No. DOF per Node/8]+1)+1 + No. Vessels in the Model +[No. auxiliary elements /8] +1 + 2 * No. Of Vessel Profiles + 2 * No. Of Body Profiles, Block Length=No. Of Vessels in the Model, Structure Length=3.

Record Layout	Description

Vessel Velocity – DOF1 [real]	Vessel velocity in translational degrees of freedom (3 terms)
Vessel Velocity – DOF2 [real]	
Vessel Velocity – DOF3 [real]	
Vessel Acceleration – DOF1 [real]	Vessel acceleration in translational degrees of freedom (3 terms)
Vessel Acceleration – DOF2 [real]	
Vessel Acceleration – DOF3 [real]	
Vessel Rotational Velocity Matrix (1,1) [real]	Transformation matrix required to find translational velocity of a point on the vessel due to vessel rotations in yaw, roll & pitch (3x3 matrix).
Vessel Rotational Velocity Matrix (2,1) [real]	

Record Layout	Description
Vessel Rotational Velocity Matrix (3,1) [real]	
Vessel Rotational Velocity Matrix (1,2) [real]	

Vessel Rotational Velocity Matrix (2,2) [real]	
Vessel Rotational Velocity Matrix (3,2) [real]	
Vessel Rotational Velocity Matrix (1,3) [real]	
Vessel Rotational Velocity Matrix (2,3) [real]	
Vessel Rotational Velocity Matrix (3,3) [real]	
Vessel Rotational Acceleration Matrix (1,1) [real]	Transformation matrix required to find translational acceleration of a point on the vessel due to vessel rotations in yaw, roll & pitch (3x3 matrix).

Record Layout	Description
Vessel Rotational Acceleration Matrix (2,1) [real]	
Vessel Rotational Acceleration Matrix (3,1) [real]	

Vessel Rotational Acceleration Matrix (1,2) [real] Vessel Rotational Acceleration Matrix (2,2) [real] Vessel Rotational Acceleration Matrix (3,2) [real] Vessel Rotational Acceleration Matrix (1,3) [real] Vessel Rotational Acceleration Matrix (2,3) [real] Vessel Rotational Acceleration Matrix (3,3) [real]	

Block 15-Convected Axes, Written if Convected Axes Outputted Flag=1

Start Record= Time Slice Start+ (Motions Outputted Flag+ Velocities Outputted Flag+ Accelerations Outputted Flag)*([No. Nodes in Database * No. DOF per Node /8] + 1) + No. Guides Flag *([No. Guides* 9/8] + 1) + No. ZGG Flag *([No. ZGG*6/8]+1)+[No. Nodes in Database*8/8]+1+ Clashing Flag*([No. Clashing Regions/8]+1)+ Auxiliary Flag*([No. auxiliary nodes * No. DOF per Node/8]+1)+1 + No. Vessels in the Model +[No. auxiliary elements /8] +1 + 2 * No. Of Vessel Profiles + 2 * No. Of Body Profiles + 3 * No. Of Vessels in the Model, Block Length= [No. Elements in Database * 6 /8] + 1, Structure Length=1, Data is written compactly.

Record Layout	Description

X-Axis component 1 [real] X-Axis component 2 [real] X-Axis component 3 [real] Y-Axis component 1 [real] Y-Axis component 2 [real] Y-Axis component 3 [real] *Space reserved.*	

Force Database File Structure

HEADER SECTION (COMMON TO BOTH MOTION AND FORCE DATABASE FILES)

As detailed in Motion Database File Structure.

RUN-TIME STATISTICS

Block 1-Run-Time Statistics, Written if Run-time Statistics Flag=1 and only if the restoring force has been requested (See flags for Local Shear-Y to Effective Tension).

Start Record= No records required for the header+1, Block Length= No records required for the force run-time statistics, Structure Length=[No. Elements in Database*12/8]+1, Data is written compactly until exhausted.

For each element included in the database and each location (1, 2 or 3) on that element, we write the minimum, maximum, mean and standard deviation of the restoring force.

The structure below is repeated to write, in order: Local Shear-Y, Local Shear-Z, Torque, Local Y-Bending Moment, Local Z-Bending Moment and Effective Tension.

Record Layout	Description
First Element in Database-LOC1-Min [real]	
First Element in Database-LOC1-Max [real]	
First Element in Database-LOC1-Mean [real]	
First Element in Database-LOC1-Std Dev [real]	
First Element in Database-LOC2-Min [real]	
First Element in Database-LOC2-Max [real]	
First Element in Database-LOC2-Mean [real]	

First Element in Database-LOC2-Std Dev [real]	

Record Layout	Description
First Element in Database-LOC3-Min [real]	
First Element in Database-LOC3-Max [real]	
First Element in Database-LOC3-Mean [real]	
First Element in Database-LOC3-Std Dev [real]	
Second Element in Database-LOC1-Min [real]	
Second Element in Database-LOC1-Max [real]	
Second Element in Database-LOC1-Mean [real]	

Second Element in Database-LOC1-Std Dev [real]	

TIME SLICE

Time Slice Start= No records required for the header+ No records required for the force run-time statistics+(Time Step Index – 1)* No. records required for each time slice + 1, Time Slice Length = No. records required for each time slice

Block - Reactions, Written if Reaction Outputted Flag=1

Start Record= Time Slice Start, Block Length= [No. Nodes with BCs * No. DOF per Node /8] + 1, Data is written compactly.

Record Layout	Description
First Node-DOF1 [real]	There are outputted for nodes with BCs, not for all nodes in the database.
First Node-DOF2 [real]	
First Node-DOF3 [real]	
First Node-DOF4 [real]	

First Node-DOF5 [real]	
First Node-DOF6 [real]	
Second Node-DOF1 [real]	
Second Node-DOF2 [real]	And continues with DOF3-6, then move to the next node and so on.

Block - Axial Force, Written if Axial Force Outputted Flag =1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly.

Record Layout	Description
First Element in Database-LOC1 [real]	
First Element in Database-LOC2 [real]	
First Element in Database-LOC3 [real]	

Second Element in Database-LOC1 [real]	
Second Element in Database-LOC2 [real]	
Second Element in Database-LOC3 [real]	
Third Element in Database-LOC1 [real]	
Third Element in Database-LOC2 [real]	

Block - Local Shear-Y Force, Written if Local Shear-Y Outputted Flag=1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ Axial Force Outputted Flag *([No. Elements in Database*3/8]+1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly as in the previous block.

Block - Local Shear-Z Force, Written if Local Shear-Z Outputted Flag=1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag)*([No. Elements in Database*3/8]+1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly as in the previous block.

Block - Torque, Written if Torque Outputted Flag=1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag)*([No. Elements in Database*3/8]+1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly as in the previous block.

Block - Local Y-Bending Moment, Written if Local Y-Bending Moment Outputted Flag=1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag)*([No. Elements in Database*3/8]+1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly as in the previous block.

Block - Local Z-Bending Moment, Written if Local Z-Bending Moment Outputted Flag=1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag)*([No. Elements in Database*3/8]+1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly as in the previous block.

Block - Effective Tension, Written if Effective Tension Outputted Flag=1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag)*([No. Elements in Database*3/8]+1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly as in the previous block.

Block - Local Y-Curvature, Written if Local Y-Curvature Outputted Flag =1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag)*([No. Elements in Database*3/8]+1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly as in the previous block.

Block - Local Z-Curvature, Written if Local Z-Curvature Outputted Flag =1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag)*([No. Elements in Database*3/8]+1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly as in the previous block.

Block - Local Axial Strain, Written if Local Axial Strain Outputted Flag=1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag)*([No. Elements in Database*3/8]+1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly as in the previous block.

Block - Flat Guide Reactions, Written if Reaction Outputted Flag=1 and No. Guides>0

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1), Block Length=[No. Guides*3/8]+1, Data is written compactly.

Record Layout	Description
Reaction-Surface1-DOF1 [real]	
Reaction-Surface1-DOF2 [real]	
Reaction-Surface1-DOF3 [real]	
Reaction-Surface2-DOF1 [real]	
Reaction-Surface2-DOF2 [real]	
Reaction-Surface2-DOF3 [real]	
Reaction-Surface3-DOF1	

[real] Reaction-Surface3-DOF2 [real]	

Block - ZGG Reactions, Written if Reaction Outputted Flag=1 and No. ZGG Guides>0

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Guides*3/8]+1), Block Length=[No. ZGG Guides *3/8]+1, Data is written compactly.

Record Layout	Description
Reaction-Surface1-DOF1 [real] Reaction-Surface1-DOF2 [real] Reaction-Surface1-DOF3 [real] Reaction-Surface2-DOF1 [real]	

Reaction-Surface2-DOF2 [real]	
Reaction-Surface2-DOF3 [real]	
Reaction-Surface3-DOF1 [real]	
Reaction-Surface3-DOF2 [real]	

Block - PIP Reactions, Written if Reaction Outputted Flag=1 and No. PIP Connections>0

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Guides*3/8]+1+[No. ZGG Guides *3/8]+1), Block Length=[No. PIP Connections *3/8]+1, Data is written compactly.

Record Layout	Description
Reaction-PIP1-DOF1 [real]	
Reaction-PIP1-DOF2 [real]	

Reaction-PIP1-DOF3 [real]	
Reaction-PIP2-DOF1 [real]	
Reaction-PIP2-DOF2 [real]	
Reaction-PIP2-DOF3 [real]	
Reaction-PIP3-DOF1 [real]	
Reaction-PIP3-DOF2 [real]	

Block - Clashing Reactions, Written if Reaction Outputted Flag=1 and No. Clashing Regions>0

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Guides*3/8]+1+[No. ZGG Guides *3/8]+1+[No. PIP Connections *3/8]+1), Block Length=[No. Clashing Regions*2/8]+1, Data is written compactly.

Record Layout	Description
Clashing1-Reaction [real]	
Clashing1-Impulse [real]	
Clashing2-Reaction [real]	
Clashing2-Impulse [real]	
…	
…	
…	
…	

Block - Temperature, Written if Temperature Outputted Flag=1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Flat Guides*3/8]+1+[No. ZGG Guides *3/8]+1+[No. PIP Connections *3/8]+1+[No. Clashing Regions*2/8]+1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly.

Record Layout	Description
Temperature-Element1-LOC1 [real] Temperature-Element1-LOC2 [real] Temperature-Element1-LOC3 [real] Temperature-Element2-LOC1 [real] Temperature-Element2-LOC2 [real] Temperature-Element2-LOC3 [real] …	

...	

Block - Pressure, Written if Pressure Outputted Flag=1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Flat Guides*3/8]+1+[No. ZGG Guides *3/8]+1+[No. PIP Connections *3/8]+1+[No. Clashing Regions*2/8]+1)+ Temperature Outputted Flag*([No. Elements in Database*3/8]+1), Block Length=[No. Elements in Database*3*2/8]+1, Data is written compactly.

Record Layout	Description
Element1-LOC1-InternalPressure [real]	
Element1-LOC1-ExternalPressure [real]	
Element1-LOC2-InternalPressure [real]	
Element1-LOC2-ExternalPressure [real]	

Element1-LOC3-InternalPressure [real]	
Element1-LOC3-ExternalPressure [real]	
Element2-LOC1-InternalPressure [real]	
Element2-LOC1-ExternalPressure [real]	

Block – Plastic Axial Strain, Written if Plastic Generalised Strains Flag =1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Flat Guides*3/8]+1+[No. ZGG Guides *3/8]+1+[No. PIP Connections *3/8]+1+[No. Clashing Regions*2/8]+1)+ Temperature Outputted Flag*([No. Elements in Database*3/8]+1)+ Pressure Outputted Flag*([No. Elements in Database*3*2/8]+1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly.

Record Layout	Description
First Element in Database-LOC1 [real]	

First Element in Database-LOC2 [real]	
First Element in Database-LOC3 [real]	
Second Element in Database-LOC1 [real]	
Second Element in Database-LOC2 [real]	
Second Element in Database-LOC3 [real]	
Third Element in Database-LOC1 [real]	
Third Element in Database-LOC2 [real]	

Block – Plastic Local-y Curvature, Written if Plastic Generalised Strains Flag =1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Flat Guides*3/8]+1+[No. ZGG Guides *3/8]+1+[No. PIP Connections *3/8]+1+[No. Clashing Regions*2/8]+1)+ Temperature Outputted Flag*([No. Elements in Database*3/8]+1)+ Pressure Outputted Flag*([No. Elements in Database*3*2/8]+1)+ Plastic Generalised Strains Flag*([No. Elements in Database*3/8] +1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly as in the previous block.

Block – Plastic Local-z Curvature, Written if Plastic Generalised Strains Flag =1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Flat Guides*3/8]+1+[No. ZGG Guides *3/8]+1+[No. PIP Connections *3/8]+1+[No. Clashing Regions*2/8]+1)+ Temperature Outputted Flag*([No. Elements in Database*3/8]+1)+ Pressure Outputted Flag*([No. Elements in Database*3*2/8]+1)+ Plastic Generalised Strains Flag*2*([No. Elements in Database*3/8] +1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly as in the previous block.

Block –Longitudinal Stress, Written if Plastic Generalised Strains Flag =1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Flat Guides*3/8]+1+[No. ZGG Guides *3/8]+1+[No. PIP Connections *3/8]+1+[No. Clashing Regions*2/8]+1)+ Temperature Outputted Flag*([No. Elements in Database*3/8]+1)+ Pressure Outputted Flag*([No. Elements in Database*3*2/8]+1)+ Plastic Generalised Strains Flag*3*([No. Elements in Database*3/8]+1), Block Length= Plastic Generalised Strains Flag*8* ([No. Elements in Database*3/8]+1) , Data is written compactly.

Record Layout	Description
First Element in Database-000-degrees-LOC1 [real]	
First Element in Database-000-degrees -LOC2 [real]	
First Element in Database-000-degrees -LOC3 [real]	
Second Element in Database-000-degrees -LOC1 [real]	
Second Element in Database-000-degrees -LOC2 [real]	
Second Element in Database-000-degrees -LOC3 [real]	

...	
...	

Record Layout	Description
First Element in Database-045-degrees-LOC1 [real]	
First Element in Database-045-degrees -LOC2 [real]	
First Element in Database-045-degrees -LOC3 [real]	
Second Element in Database-045-degrees -LOC1 [real]	
Second Element in Database-045-degrees -LOC2 [real]	
Second Element in Database-045-degrees -LOC3 [real]	
...	
...	

Record Layout	Description
First Element in Database-090-degrees-LOC1 [real]	
First Element in Database-090-degrees -LOC2 [real]	
First Element in Database-090-degrees -LOC3 [real]	
Second Element in Database-090-degrees -LOC1 [real]	
Second Element in Database-090-degrees -LOC2 [real]	
Second Element in Database-090-degrees -LOC3 [real]	
...	
...	

Record Layout	Description

First Element in Database- 135-degrees-LOC1 [real] First Element in Database- 135-degrees -LOC2 [real] First Element in Database- 135-degrees -LOC3 [real] Second Element in Database- 135-degrees -LOC1 [real] Second Element in Database- 135-degrees -LOC2 [real] Second Element in Database- 135-degrees -LOC3 [real]	

Record Layout	Description
First Element in Database- 180-degrees-LOC1 [real]	

First Element in Database-180-degrees -LOC2 [real]	
First Element in Database-180-degrees -LOC3 [real]	
Second Element in Database-180-degrees -LOC1 [real]	
Second Element in Database-180-degrees -LOC2 [real]	
Second Element in Database-180-degrees -LOC3 [real]	
...	
...	

Record Layout	Description
First Element in Database-225-degrees-LOC1 [real]	
First Element in Database-225-degrees -LOC2 [real]	

First Element in Database- 225-degrees -LOC3 [real] Second Element in Database- 225-degrees -LOC1 [real] Second Element in Database- 225-degrees -LOC2 [real] Second Element in Database- 225-degrees -LOC3 [real]	

Record Layout	Description
First Element in Database- 270-degrees-LOC1 [real] First Element in Database- 270-degrees -LOC2 [real] First Element in Database- 270-degrees -LOC3 [real]	

Second Element in Database-270-degrees -LOC1 [real]	
Second Element in Database-270-degrees -LOC2 [real]	
Second Element in Database-270-degrees -LOC3 [real]	
...	
...	

Record Layout	Description
First Element in Database-315-degrees-LOC1 [real]	
First Element in Database-315-degrees -LOC2 [real]	
First Element in Database-315-degrees -LOC3 [real]	
Second Element in Database-315-degrees -LOC1 [real]	

Second Element in Database-315-degrees -LOC2 [real]	
Second Element in Database-315-degrees -LOC3 [real]	
...	
...	

Block – Equivalent Plastic Axial Strain, Written if Equivalent Plastic Generalised Strains Flag =1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Flat Guides*3/8]+1+[No. ZGG Guides *3/8]+1+[No. PIP Connections *3/8]+1+[No. Clashing Regions*2/8]+1)+ Temperature Outputted Flag*([No. Elements in Database*3/8]+1)+ Pressure Outputted Flag*([No. Elements in Database*3*2/8]+1)+ Plastic Generalised Strains Flag*11*([No. Elements in Database*3/8]+1), Block Length=[No. Elements in Database*3/8]+1

Record Layout	Description
First Element in Database-LOC1 [real]	

First Element in Database-LOC2 [real]	
First Element in Database-LOC3 [real]	
Second Element in Database-LOC1 [real]	
Second Element in Database-LOC2 [real]	
Second Element in Database-LOC3 [real]	
Third Element in Database-LOC1 [real]	
Third Element in Database-LOC2 [real]	

Block – Equivalent Plastic Curvature, Written if Equivalent Plastic Generalised Strains Flag =1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Flat Guides*3/8]+1+[No. ZGG Guides *3/8]+1+[No. PIP Connections *3/8]+1+[No. Clashing Regions*2/8]+1)+ Temperature Outputted Flag*([No. Elements in Database*3/8]+1)+ Pressure Outputted Flag*([No. Elements in Database*3*2/8]+1)+ Plastic Generalised Strains Flag*11*([No. Elements in Database*3/8]+1))+ Equivalent Plastic Generalised Strains Flag*([No. Elements in Database*3/8]+1), Block Length=[No. Elements in Database*3/8]+1, Data is written compactly as in the previous block.

Block -Water Surface Elevation, Always Written

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Flat Guides*3/8]+1+[No. ZGG Guides *3/8]+1+[No. PIP Connections *3/8]+1+[No. Clashing Regions*2/8]+1)+ Temperature Outputted Flag*([No. Elements in Database*3/8]+1)+ Pressure Outputted Flag*([No. Elements in Database*3*2/8]+1))+ Plastic Generalised Strains Flag*11*([No. Elements in Database*3/8]+1))+ Equivalent Plastic Generalised Strains Flag*2*([No. Elements in Database*3/8]+1), Block Length=1, Data is written compactly.

Record Layout	Description
Time [real] Water Elevation [real] Time Step [real]	

Ramp Value [real] Total Kinetic Energy [real]	

Block – Damper Power, Written if Damper Power Outputted Flag = 1

Start Record= Time Slice Start+Reaction Outputted Flag*([No. Nodes with BCs * No. DOF per Node /8] + 1)+ (Axial Force Outputted Flag+Local Shear-Y Outputted Flag+ Local Shear-Z Outputted Flag+ Torque Outputted Flag+ Local Y-Bending Moment Outputted Flag+ Local Z-Bending Moment Outputted Flag+ Effective Tension Outputted Flag+ Local Y-Curvature Outputted Flag+ Local Z-Curvature Outputted Flag+ Local Axial Strain Outputted Flag)*([No. Elements in Database*3/8]+1)+ Reaction Outputted Flag *([No. Flat Guides*3/8]+1+[No. ZGG Guides *3/8]+1+[No. PIP Connections *3/8]+1+[No. Clashing Regions*2/8]+1)+ Temperature Outputted Flag*([No. Elements in Database*3/8]+1)+ Pressure Outputted Flag*([No. Elements in Database*3*2/8]+1)+ Plastic Generalised Strains Flag*11*([No. Elements in Database*3/8]+1))+ Equivalent Plastic Generalised Strains Flag*2*([No. Elements in Database*3/8]+1) + 1, Block Length=[No. Damper Elements in Database/8]+1, Data is written compactly.

Record Layout	Description
First Damper Element - Power [real]	Power output of the first damper element

Second Damper Element - Power [real]	Power output of the second damper element
...	
...	
...	
...	
...	
...	

1.10 Examples

This section of the documentation describes a set of examples which illustrate some of the range of applications for which Flexcom may be used. The examples are intended to provide a representative sample of the capabilities of the program. There are, however, many program features that are not described in the examples and the range of application of the program is by no means limited to the type of structures described here.

The Flexcom examples set is divided into the following sections:

- A - Top Tensioned Risers

1.10.1 A - Top Tensioned Risers

Section A contains some examples of top tensioned risers, including:

1.10.1.1 A01 - Deepwater Drilling Riser

This example describes example deepwater drilling riser analyses, and demonstrates a number of features of Flexcom specifically suited to the modelling of drilling risers. The overall layout of the example is as follows:

- Introduction gives an overview of the drilling riser analysis, and notes some of the more important features of Flexcom which are relevant to the analysis.
- Model Summary describes the model in more detail, and discusses the relevant analytical capabilities of the software.
- Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions to which the riser is subjected.
- Results presents pertinent results from the various analyses performed and discusses their significance.
- Input Data contains the input data for the drilling riser model.
- Calculation of Flexcom Input Data describes how the required Flexcom inputs are computed from the model data presented in Input Data.

Introduction

This example considers the analysis of a drilling riser in 7500 ft of water. Connected mode (including drift-off) and disconnected mode analyses are performed. The loads experienced by the riser include current loading, wave loading due to both regular waves and random seas, and forces induced by vessel motions including vessel offsets and the first and second order vessel response to waves.

This example illustrates a number of important modelling features in Flexcom. Specifically, the following capabilities are demonstrated:

- Modelling of soil structure interaction defined in terms of P-y curves.
- Modelling of complex riser tensioning system.
- Simulation of vessel drift-off via the inclusion of an explicitly modelled moored vessel in an analysis.
- Simulation of an emergency-disconnect scenario and subsequent riser recoil.
- Performing dynamic analyses in the frequency domain, and comparing results with corresponding time domain dynamic analyses.
- Performing a modal analysis using Modes to calculate the natural frequencies and mode shapes of a top tensioned riser.

- Performing a time domain fatigue analysis using LifeTime to estimate fatigue damage and predict fatigue life.
- Performing a frequency domain fatigue analysis using LifeFrequency to estimate fatigue damage and predict fatigue life.

A schematic of the riser stack-up is shown in the Riser Stack-Up Schematic figure.

Model Summary

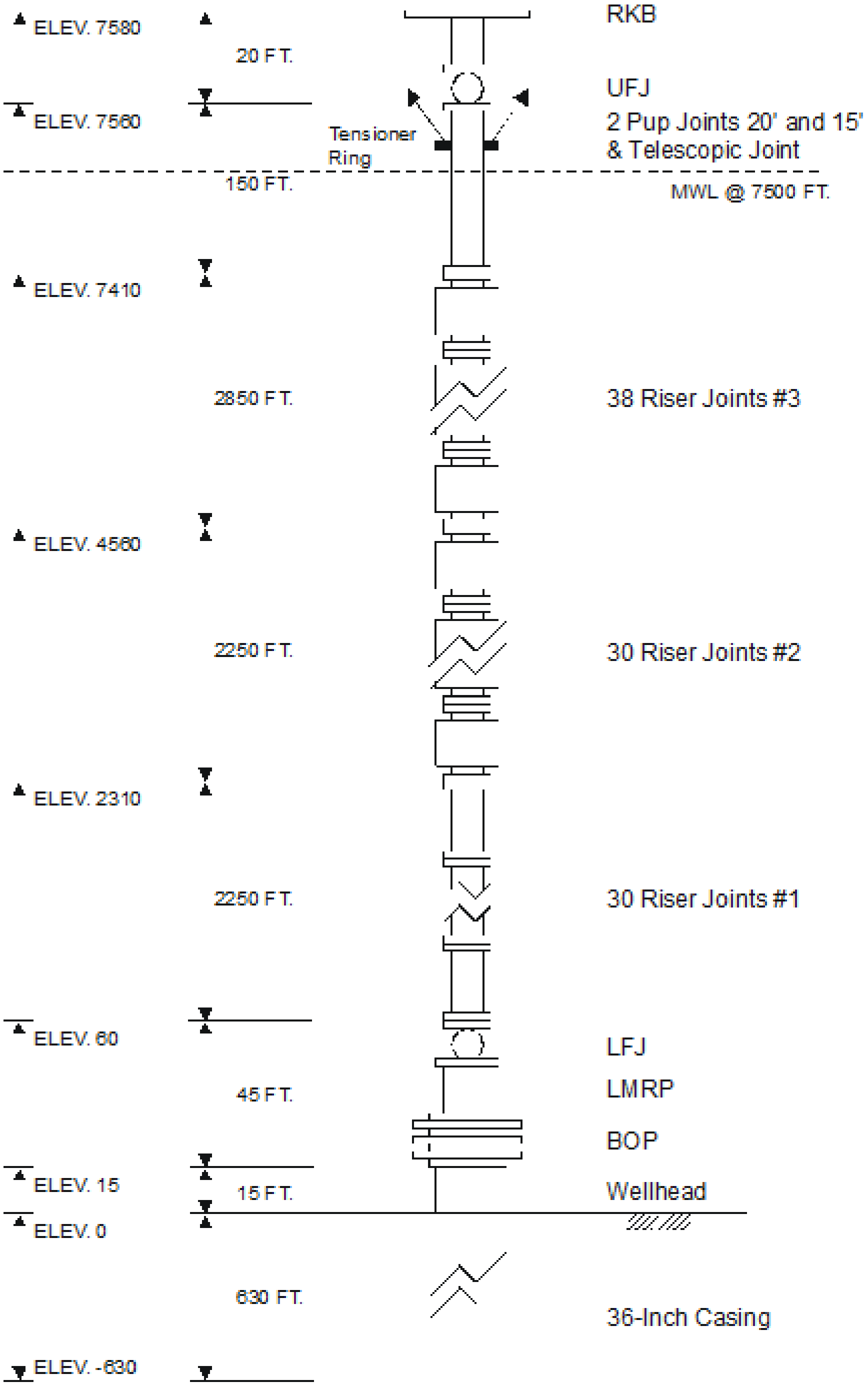
The main body of the riser consists of three different joint types, denoted Riser Joints #1 - #3 in figure 1-1. Types #1 and #2 are buoyed joints, while Joint #3 is unbuoyed. The riser is fixed to a wellhead at the seabed via a lower flex joint (LFJ) and lower marine riser package (LMRP). This model also includes 36" casing modelled to a depth of 630 ft below the mudline. The riser is attached to a floating vessel via an upper flex joint (UFJ), at a distance of 60 ft above the MWL. The riser tensioning system and slip ring is modelling as an assemblage of beam elements.

Further information is provided in the following sections.

- Drilling Riser Joints
- Soil Structure Interaction
- Riser Tensioning System
- Vessel Drift-Off
- Emergency Disconnect

Drilling Riser Joints

The data specification for the riser joint properties consists only of the riser OD and wall thickness, the OD and ID of the various peripheral lines, and the amount and distribution of buoyancy material on buoyed joints. All of the required Flexcom inputs are then calculated from this data, in order to demonstrate the process and to provide an insight into the significance of these inputs. Calculation of Flexcom Input Data later in the example describes this process.



Riser Stack-Up Schematic

Soil Structure Interaction

The soil structure interaction modelling is governed by the specification of P-y curves, which define the soil resistance to lateral motion of the structure, over a range of depths below the mudline. The appropriate (non-linear) soil stiffness is then applied to nodes of the finite element model at locations along the length of the conductor/casing corresponding to the depths at which P-y curves are defined.

Each P-y curve may be specified at a particular depth below the mudline, or alternatively at a node of the FE discretisation. The latter approach permits a more general specification of P-y curves, for example, with models of multiple risers, and is the approach used here. The actual curves themselves are presented in Input Data.

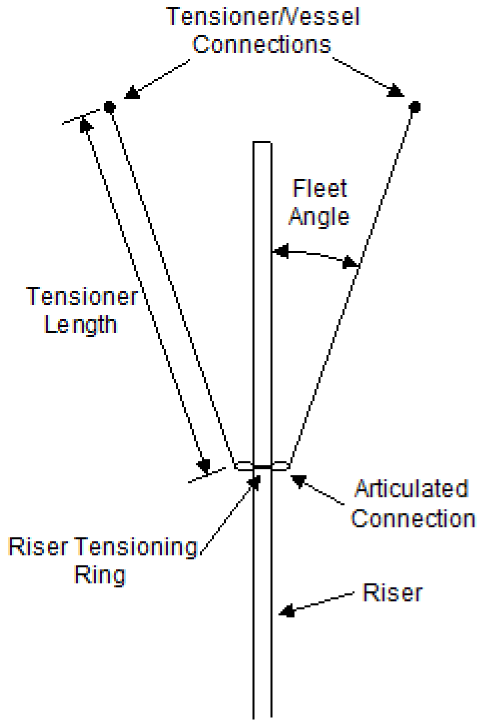
Riser Tensioning System

The riser tensioning system is modelled using an arrangement of Non-Linear Elements to represent the individual tensioning lines. The tensioning lines are arranged symmetrically about the riser and connected to a slip ring via Hinge Elements at the nominated connection point. The tops of the tensioning lines are connected to the vessel. From the geometry, you may compute the nominal length and fleet angle of each of the tensioning lines. Each of the tensioning lines has a very low axial stiffness, allowing it to apply a constant tension to the riser irrespective of any increase or decrease in the length of the tensioning line. The tension applied by each line is given by the expression:

$$F = \frac{T}{n\cos\theta} \qquad (1)$$

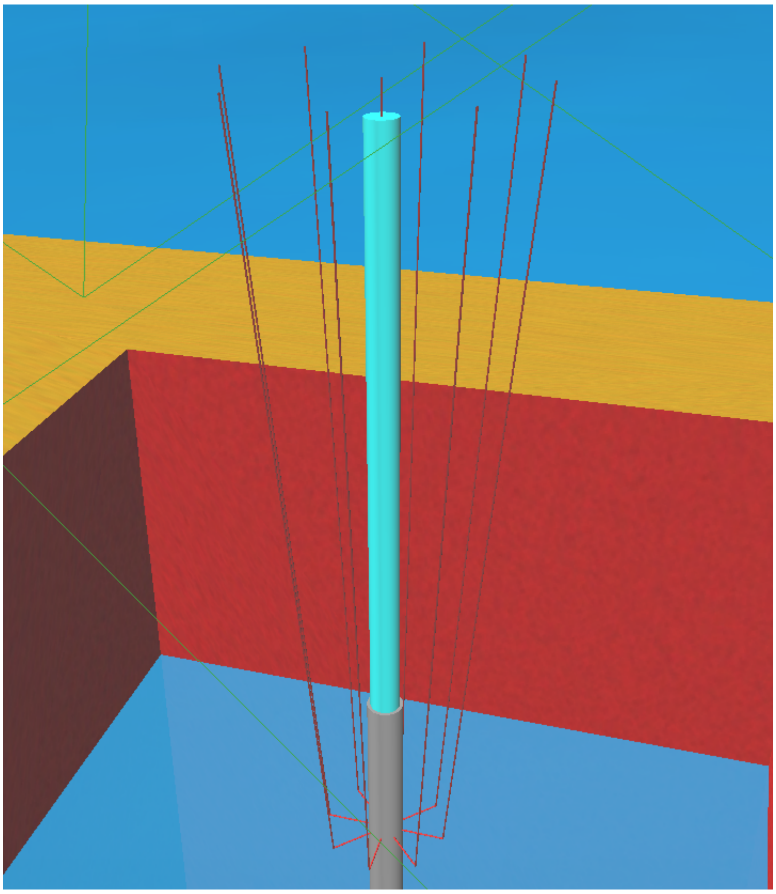
where F is the tension in each tensioning line, T is the specified total vertical tension to be applied to the riser, n is the number of tensioning lines, and θ is the fleet angle. This ensures that the total vertical force applied to the riser is equal to the required top tension value. If you examine the Flexcom keyword file, you will see how Parameters and Equations have been used to compute the required parameters such as fleet angle, and tension within each line.

A schematic illustration of the riser tensioning system is as follows.



Riser Tensioning System Schematic

The tensioning lines are represented by mass-less beam elements, which are characterised by high bending and torsional stiffness coupled with a non-linear axial stiffness. A non-linear curve is used to specify an effectively constant (non-zero) tension in each of the lines for all deflections. The tensioner is modelled in this way in order to capture the changing orientation of the tensioning lines relative to the riser in the drift-off case. Modelling the top tension as a vertical point load might probably be sufficient in the normal operating mode analysis.

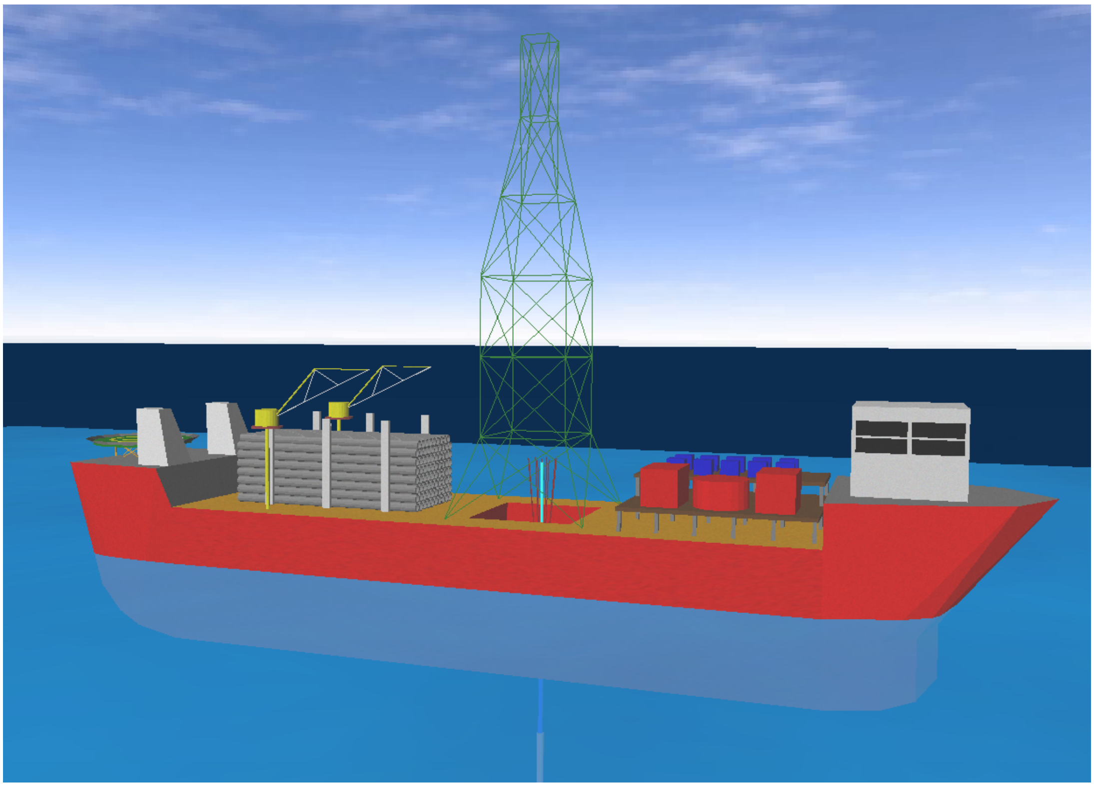


Riser Tensioning System modelled in Flexcom

Vessel Drift-Off

Simulation of vessel drift-off is achieved via the inclusion of an explicitly modelled moored vessel in the analysis. The objective is to compute the drift motion of the vessel due to current, wind, and second order drift forces computed from a wave spectrum. The vessel itself is modelled as an assemblage of rigid mass-less beam elements, with a node located at the vessel CoG. These “vessel” elements connect the vessel CoG to those riser nodes whose motions are defined by the motions of the vessel. The wind loads are computed using the wind velocity, air density, wind coefficients and the moored vessel dimensions.

The current loads are computed by a combination of the current velocity, seawater density, current coefficients and the moored vessel dimensions. The wave drift loads are calculated from user-specified Quadratic Transfer Functions (QTFs) and the discretised wave spectrum.

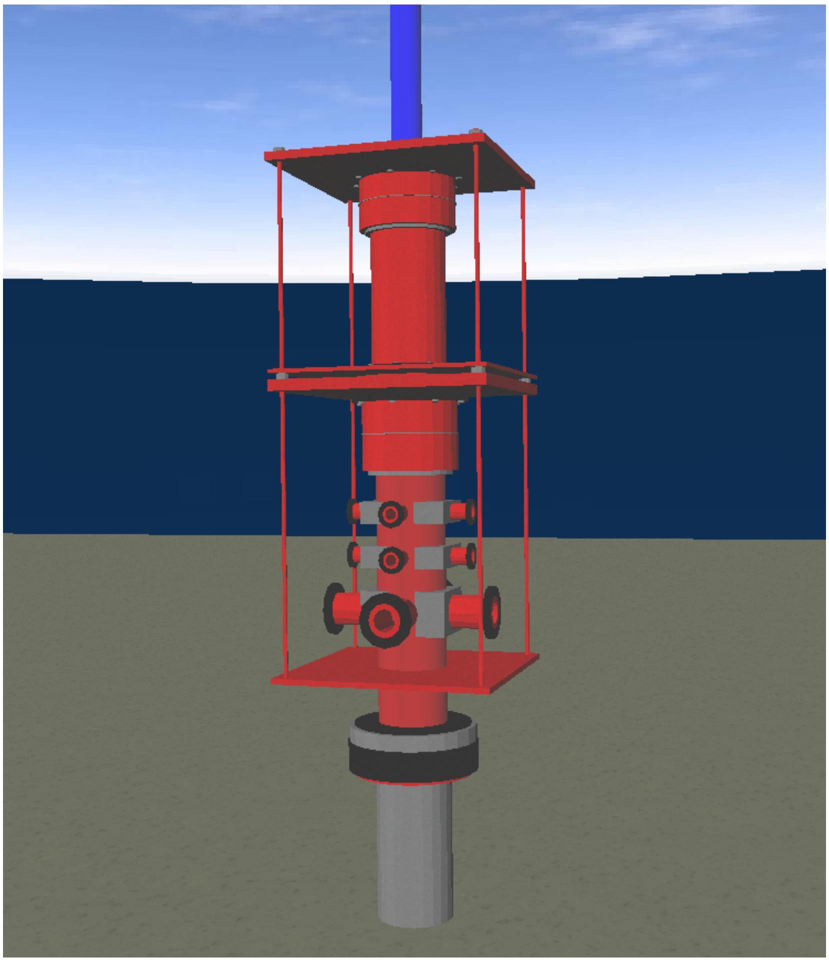


Drill Ship

Emergency Disconnect

This portion of the example simulates an emergency disconnect scenario, following failure of the dynamic positioning system on the drillship. In such situations, the riser is quickly disconnected from the wellhead to prevent any potential damage. Simulation of riser recoil is achieved by initially modelling the riser in the connected mode, before suddenly removing the boundary conditions at the base of the riser. This allows the BOP and the LMRP to move freely in all degrees of freedom. The main objective of this analysis is to compute the motion of the base of the riser immediately following disconnect.

Note: Although Flexcom can provide an approximate simulation of the disconnect event and subsequent riser recoil, a more comprehensive simulation may be performed using our DeepRiser software, which specialises in the analysis of top-tensioned risers. DeepRiser provides a detailed hydro-pneumatic tensioner, which models all the major hydraulic and pneumatic components. It can model each tensioning cylinder independently, and can simulate the behaviour of the anti-recoil control system.



LMRP, BOP & Wellhead

Analyses

Four separate scenarios are considered as follows.

- Normal Operating. This scenario represents normal drilling operations in the connected mode.
- Disconnected Mode. In the disconnected case, the wellhead, casing and tensioning system are not included.
- Drift-Off. This case simulates a connected mode drift-off analysis. The model is similar to the first case, and in addition, the drill ship is explicitly included in the model as a moored vessel.

- Emergency Disconnect. This model is initially in the connected state, but the riser is subsequently detached from the BOP.

Normal Operating

INITIAL STATIC ANALYSIS

The bottom of the casing is fixed in all 6 degrees of freedom. Vessel boundary conditions are specified in all 6 degrees of freedom at the top of the upper flex joint. Restraints for the riser tensioning system are included via the specification of vessel boundary conditions at the upper ends of the tensioning lines. The initial position of the vessel reference point and RAOs are specified.

OFFSET AND CURRENT ANALYSIS

A vessel offset of 225 ft (3% of water depth) is applied in the horizontal (DOF 2) direction. A piecewise-linear current is also included. All of the boundary conditions are unchanged and carry through automatically from the initial static analysis.

TIME DOMAIN DYNAMIC ANALYSIS

No specification of boundary conditions is required in either of the two dynamic analyses conducted here, a regular wave analysis and a random sea run. All of the boundary condition data carries through from the previous runs. Since vessel boundary conditions and RAO data have been specified previously, dynamic motions are automatically applied in both dynamic analyses with the onset of wave loading.

MODAL ANALYSIS

A modal analysis is carried out to calculate the natural frequencies and mode shapes of the drilling riser in connected mode. The riser is naturally designated as a top-tensioned riser (TTR) for the purposes of Shear 7 output. The output is based on the first 50 modes and 500 equally spaced segments. The first 100 eigenpairs are requested, as recommended practice is to specify twice the number of natural frequencies as you are actually interested in. Note that output is requested for the set entitled Riser only, as this comprises drilling riser elements only, and excludes the casing, BOP, LMRP, flex joints, tensioning system etc.

FATIGUE ANALYSIS

A time domain fatigue analysis is performed using LifeTime to estimate fatigue damage and predict fatigue life. Metocean data which underpins the fatigue assessment is typically presented as a scatter diagram. The metocean data for this example is imported directly from a spreadsheet via the Spreadsheet Based Variations feature, with a single master input file used to automatically generate all the requisite input data. A total of 43 seastates are contained within the scatter diagram, representing various combinations of significant wave height H_s and mean zero crossing period T_z.

For efficiency reasons a representative sample of just 9 seastates are considered here for illustrative purposes. Additionally, the time domain simulations are analysed for ½ hour only, whereas 3 hour simulations would generally be considered recommended practice.

In order to minimise storage requirements, only axial force and bending moment are requested for storage in the program database.

A corresponding frequency domain fatigue analysis is also performed with LifeFrequency using the Stress Spectra mode.

Disconnected Mode

INITIAL STATIC ANALYSIS

In this case the base of the riser is free in all DOFs. Vessel boundary conditions are specified in all 6 degrees of freedom at the top of the upper flex joint. The initial position of the vessel reference point and the RAO file name are again specified.

CURRENT ANALYSIS

A piecewise-linear current is included in this analysis. All boundary conditions remain unchanged and are carried through automatically from the initial static analysis.

TIME DOMAIN DYNAMIC ANALYSIS

No specification of boundary conditions is required either in the disconnected mode dynamic analysis, which is a regular wave run. All of the boundary condition data carries through from the current analysis. Since vessel boundary conditions and RAO data have been specified previously, dynamic motions are automatically applied with the onset of wave loading.

FREQUENCY DOMAIN DYNAMIC ANALYSIS

Similar to the time domain, no specification of boundary conditions is required in the frequency domain dynamic analysis. All of the boundary condition data carries through from the current analysis.

Drift-Off

INITIAL STATIC ANALYSIS

The base of the riser is fixed in all 6 degrees of freedom. No other boundary conditions are required. The moored vessel centre of gravity (COG) node is also identified.

Coefficient data is used to specify low frequency wind, wave, and current loading on the moored vessel during the drift-off analysis. No “normal” or “standard” vessel data, such as vessel initial position, offset or RAO data, is specified in this run.

OFFSET AND CURRENT ANALYSIS

Offsets of -160ft, 191ft and 130° respectively are applied as constant boundary conditions at the CoG node in DOFs 2, 3 and 4 (representing the moored vessel surge, sway and yaw directions). A piecewise-linear current is also included.

TIME DOMAIN DYNAMIC ANALYSIS

The specification of boundary conditions for the dynamic analysis is the same as for the initial static run: the CoG node is released to displace further from its offset position. Drift motions are computed from the current, wind, and wave drift loading.

Emergency Disconnect

INITIAL STATIC ANALYSIS

The base of the riser, and both ends of the riser/conductor below the mudline, are initially fixed in all degrees of freedom. Vessel boundary conditions are specified in 6 degrees of freedom at the top of the upper flex joint, and restraints for the riser tensioning system are included by specifying boundary conditions at the upper ends of the tensioning lines. The initial position of the vessel reference point and the vessel RAOs are also specified.

OFFSET AND CURRENT ANALYSIS

A vessel offset of 225 ft (3% of water depth) is applied in the horizontal (DOF 2) direction. A piecewise-linear current is also included. All of the boundary conditions are unchanged and carry through automatically from the initial static analysis.

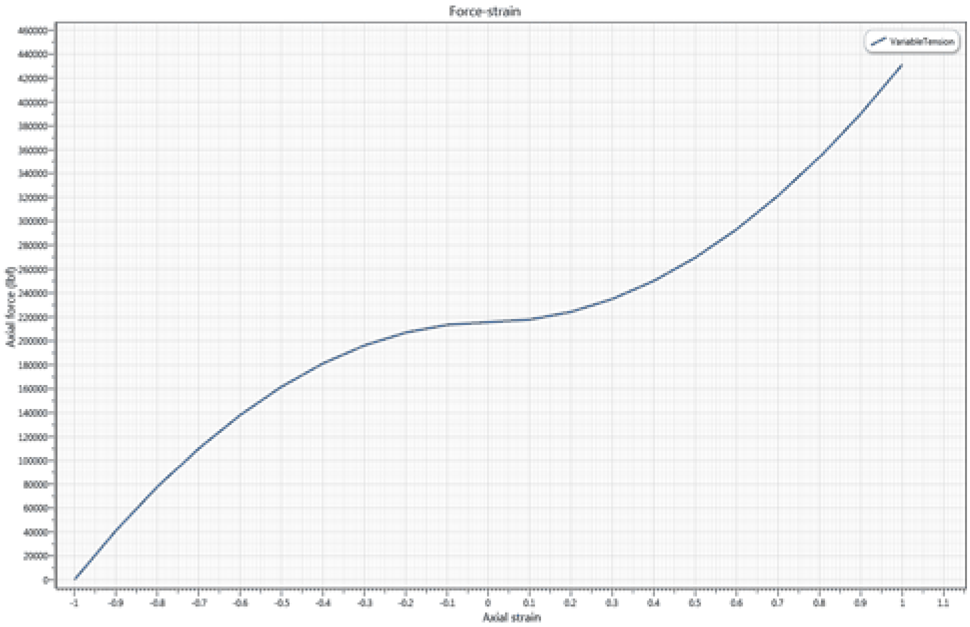
TIME DOMAIN DYNAMIC ANALYSIS

A recoil analysis is simulated for a single load case. (In reality it would be usual to look at the disconnect occurring at a number of different times over one wave cycle).

A connected state is first analysed under the action of a specified current and a regular wave, described in the Input Data section below. All of the boundary condition data carries through from the previous runs and no change is required. Since vessel boundary conditions and RAO data have been specified previously, dynamic motions are automatically applied in the dynamic analysis with the onset of wave loading.

A disconnected state is then analysed under the same current and wave loading. The boundary conditions at the base of the riser are removed and all other boundary conditions unchanged. The emergency disconnect is specified to occur at peak vessel heave, after the initial solution has reached steady-stage (the connected state is simulated for approximately five wave periods). The disconnected state is then analysed for a further five wave periods to examine the subsequent riser response.

A total top tension of 1700 kips is applied via the riser tensioning system. The tensioning system is modelled using an assembly of non-linear spring elements. These elements apply a constant tension during normal operation, and also incorporate an increased stiffness as the system approaches maximum or minimum stroke-out. The nonlinear relationship between axial force and strain is shown in the figure below.



Axial Force-Strain Relationship for Tension Line

Each of the eight tensioning lines also include damper elements. These dampers help to damp the motion of the riser after the disconnection has taken place.

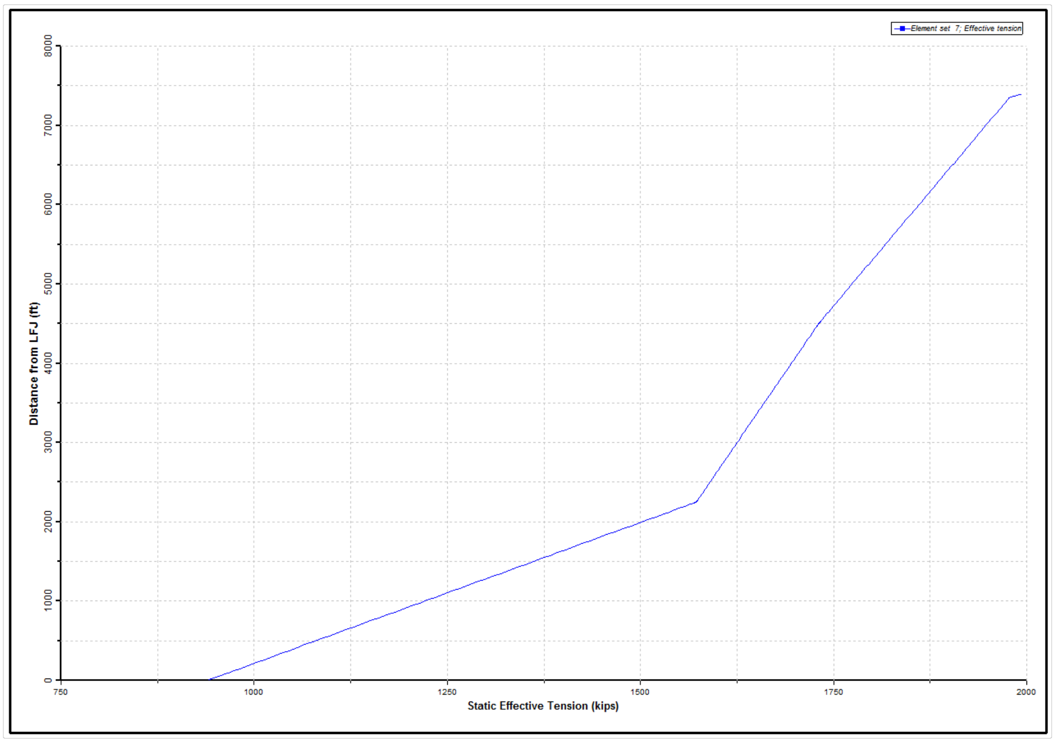
Results

This following sections contains results from the various simulations:

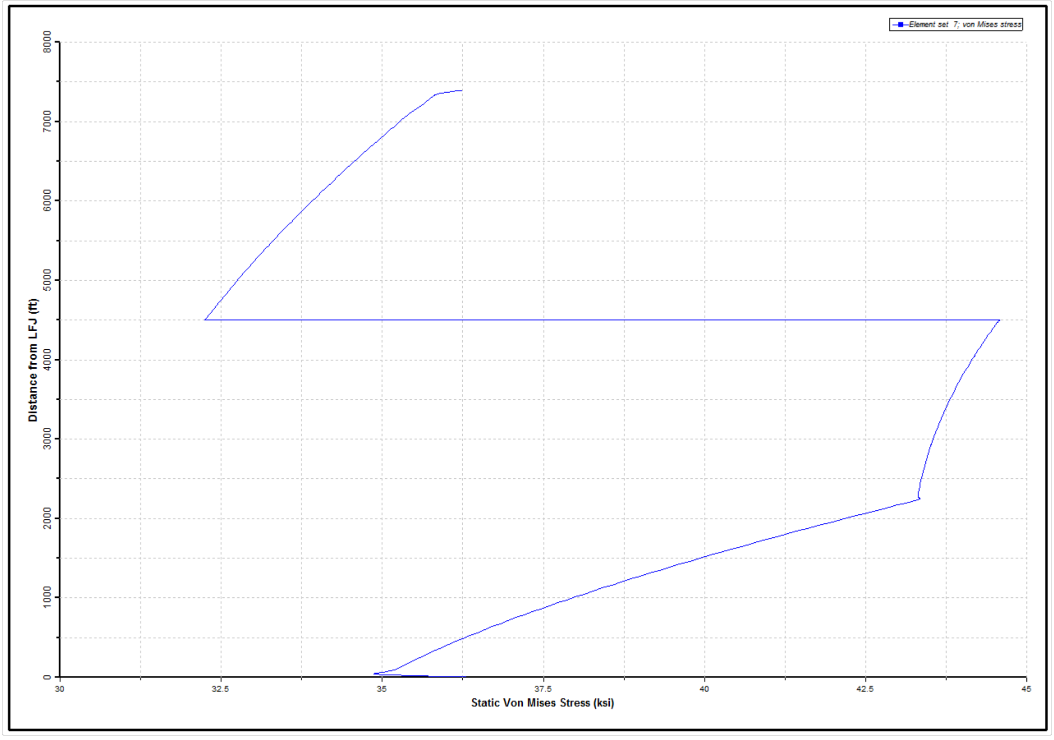
- Connected Mode, Normal Operating
- Disconnected Mode
- Connected Mode, Drift-Off
- Emergency Disconnect

Connected Mode, Normal Operating

Results from the connected mode analysis are presented in the following figures. The Static Effective Tension and Static Von Mises Stress figures below are from the static restart with offset and current, and plot respectively the distributions of effective tension and von Mises stress in the riser. The effective tensions are everywhere positive as required, and the stress values are acceptable. The discontinuity in the stress snapshot at approximately 4500 ft above the LFJ is due to the change in wall thickness here between joint Types #2 and #1 – the maximum von Mises stress clearly occurs at the inner circumference. Note that these figures, and indeed all snapshots and statistics plots presented hereafter, are for a user-defined set Riser which includes all riser joints between the lower flex joint and the telescopic joint, exclusive.



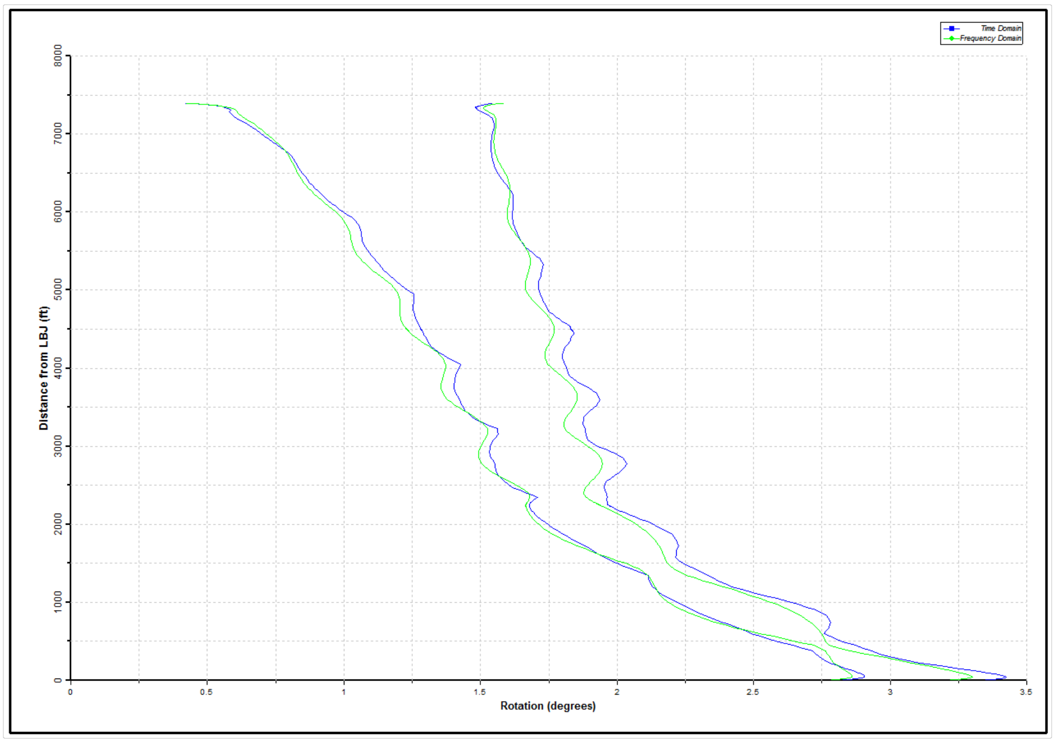
Static Effective Tension



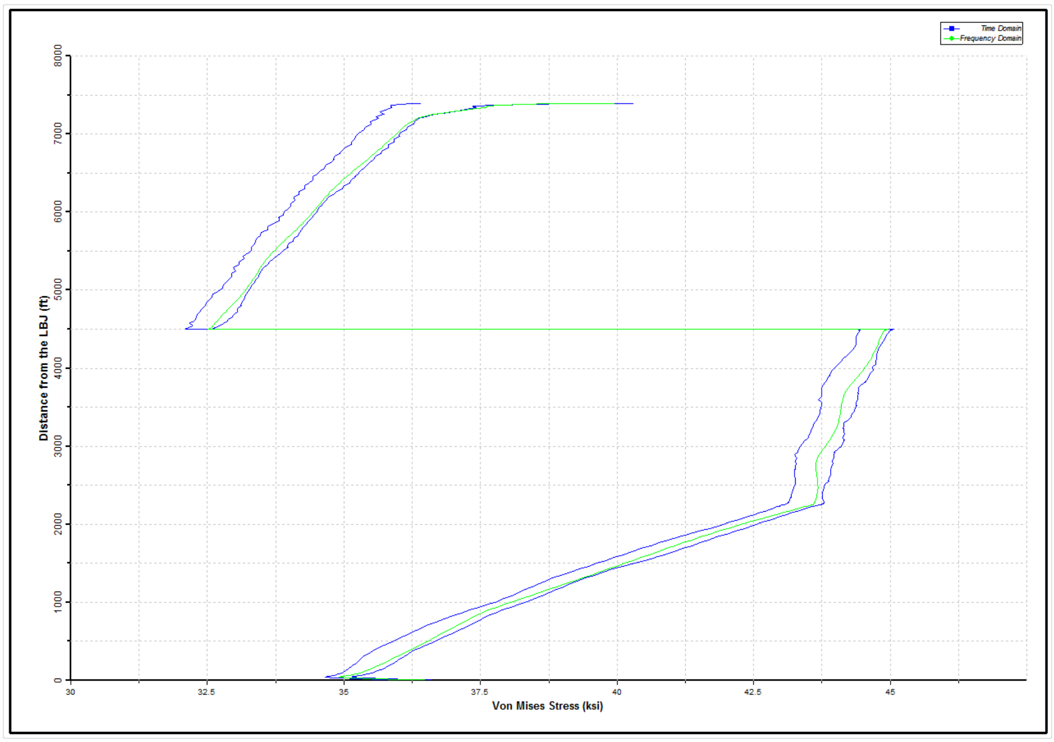
Static Von Mises Stress

REGULAR WAVE ANALYSIS

The first figure below compares maxima and minima of riser rotation from the time and frequency domain solutions. The distributions show considerable agreement. The maximum predicted rotation at the LFJ is approximately 3.4 degrees for the time domain solution, and 3.3 degrees for the frequency domain solution. (In postprocessing the time domain results, statistics are generated over the last two wave periods). The second figure below compares the von Mises stress distribution along the riser from the time and frequency domain solutions. Again considerable agreement is observed. The time domain solution shows that the dynamic variation in stress is small, since the values are dominated by the axial stresses due to top tension. The actual stress values are within acceptable limits.



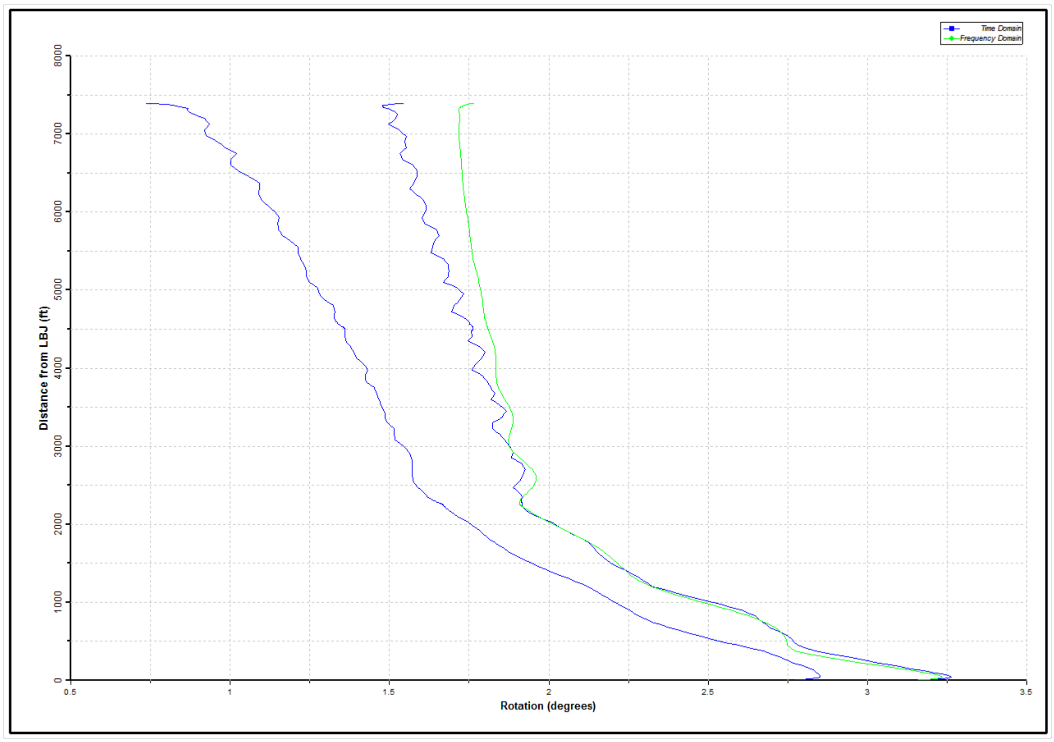
Comparing Rotations along Riser



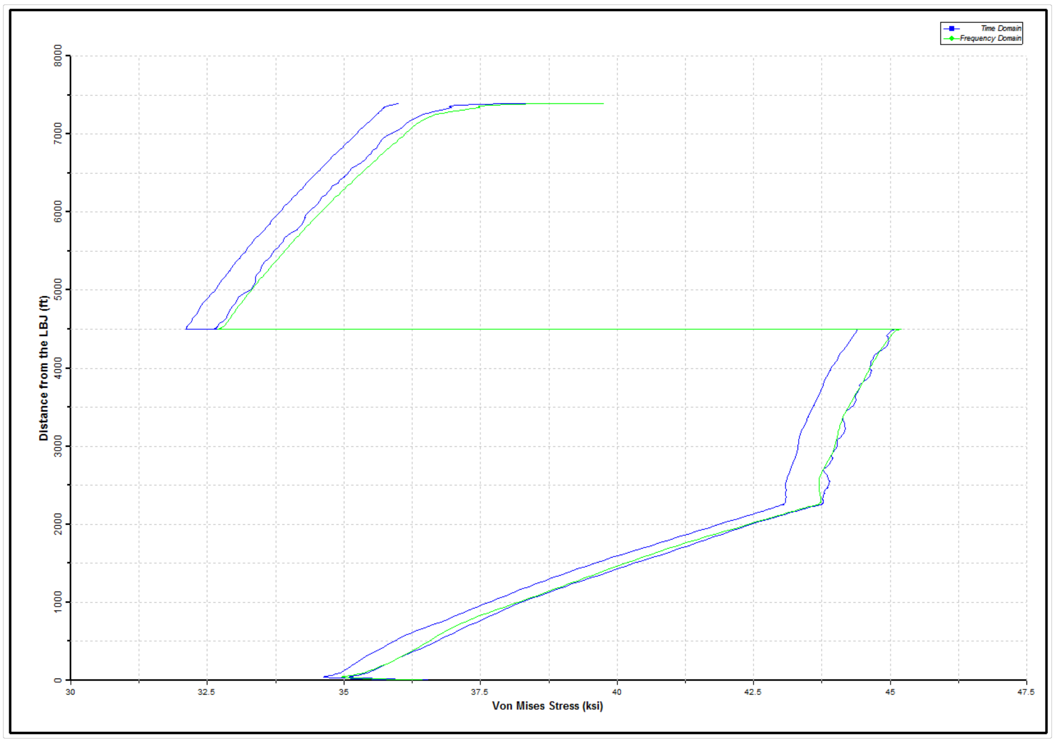
Comparing Von Mises Stresses along Riser

RANDOM SEA ANALYSIS

In figure 1-6, a maximum rotation at the LBJ of approximately 3.25 degrees is predicted by both time domain and frequency domain solutions. The distributions show considerable agreement. (In postprocessing the time domain results, statistics exclude the first 100 seconds of response, leaving a record of 2500 seconds for postprocessing). In the Comparing Von Mises Stresses along Riser figure below, the von Mises stress distribution is acceptable as before. Again there is excellent agreement between the distribution of von Mises stress predicted by the time domain and frequency domain approaches.



Comparing Rotations along Riser



Comparing Von Mises Stresses along Riser

MODAL ANALYSIS

The table below presents the natural frequencies of the drilling riser in connected mode for the first 15 modes. For a TTR, pure bending modes are assumed to occur in identical or nearly identical pairs. So Modes searches for these pairs and categorises one of them as Bending and the other as Unknown.

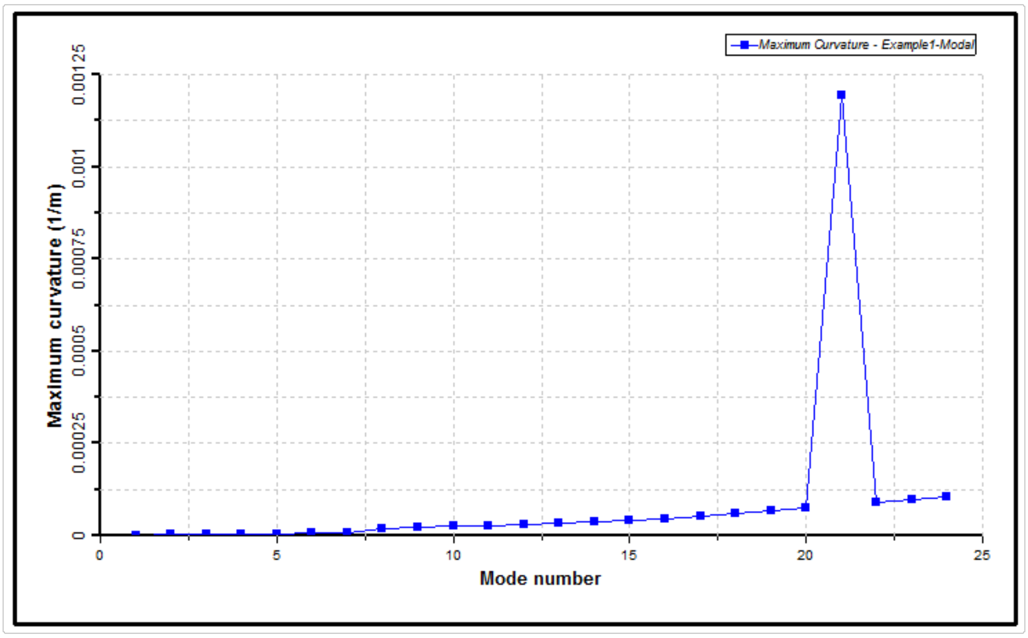
The first figure below plots maximum curvature as a function of mode number (this plot is generated automatically by Modes). This type of plot is created to help in identifying so-called 'mixed modes' (refer to Modal Analysis for further information), where mixed modes tend to appear as local maxima or spikes. The maximum modal curvature should be monotonically increasing with mode number if pure bending modes only are considered.

Riser Natural Frequencies

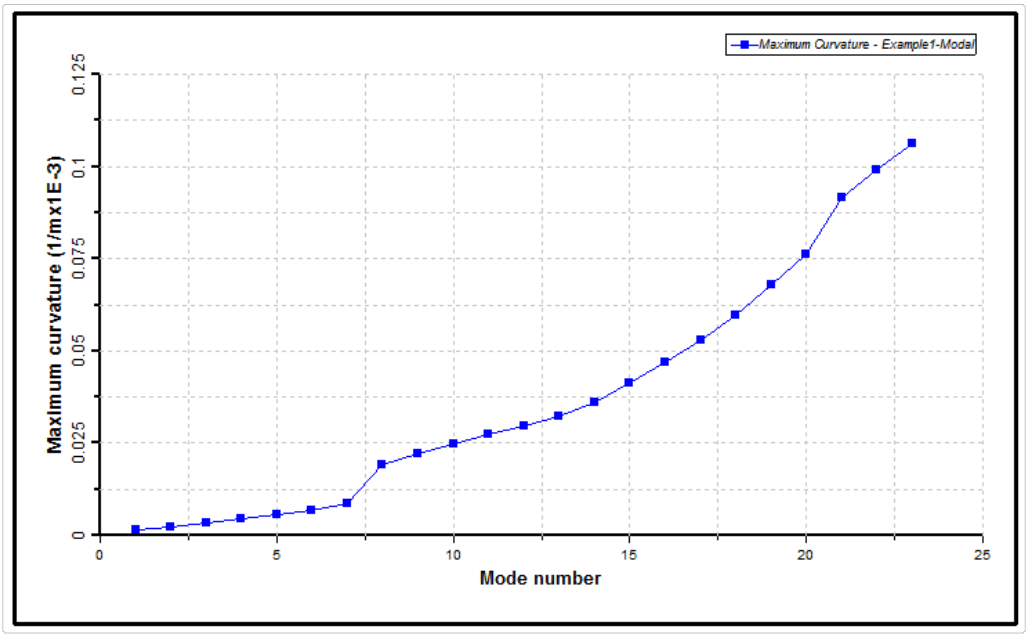
Mode No.	Period (s)

1	78.03
2	36.59
3	24.09
4	18.44
5	14.73
6	12.12
7	10.42
8	9.19
9	8.12
10	7.26
11	6.64
12	6.09
13	5.59
14	5.19
15	4.86

One such spike is clearly identifiable in the first figure below, corresponding to Mode No. 21. This mode may be excluded from the Shear 7 output by invoking the TTR Modes – Exclude option, and the modal analysis may be performed again. As the objective this time is to recreate the Shear7 output only (a purely postprocessing option), the Repeat Run capability is utilised. This facility allows you to indicate that you do not want the program to redo the full eigensolution. Instead the program is to read the eigensolution details from an earlier run, but to recalculate the Shear7 data, in this case with certain modes excluded. The first figure below plots maximum curvature as a function of mode number, with the mixed mode excluded. The maximum curvature is now monotonically increasing with mode number, suggesting that only pure bending modes are now included in this output.

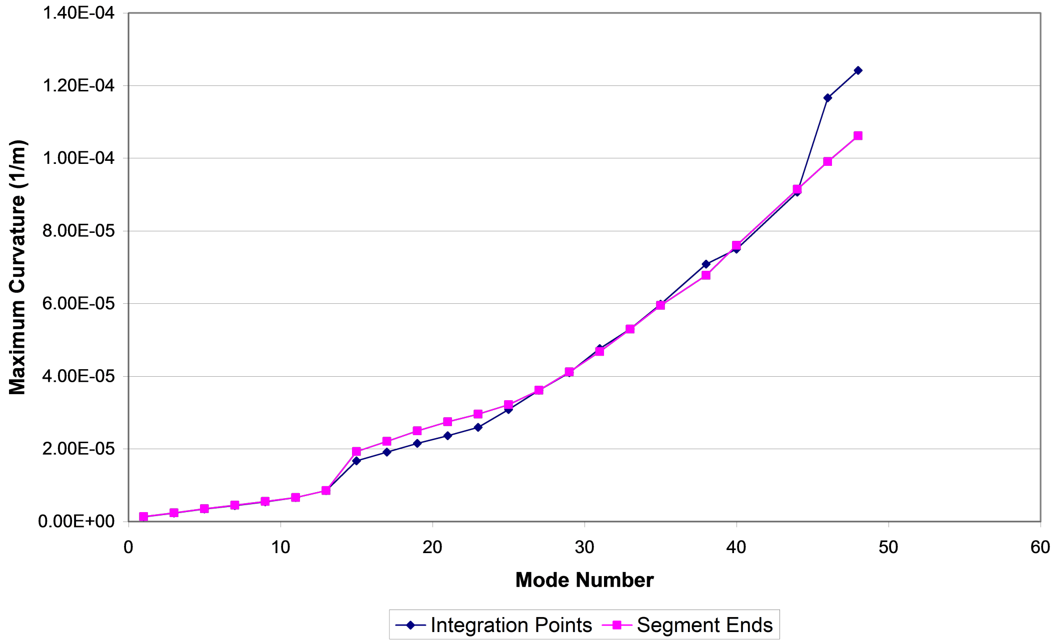


Maximum Curvatures



Maximum Curvatures, Excluding Mixed Modes

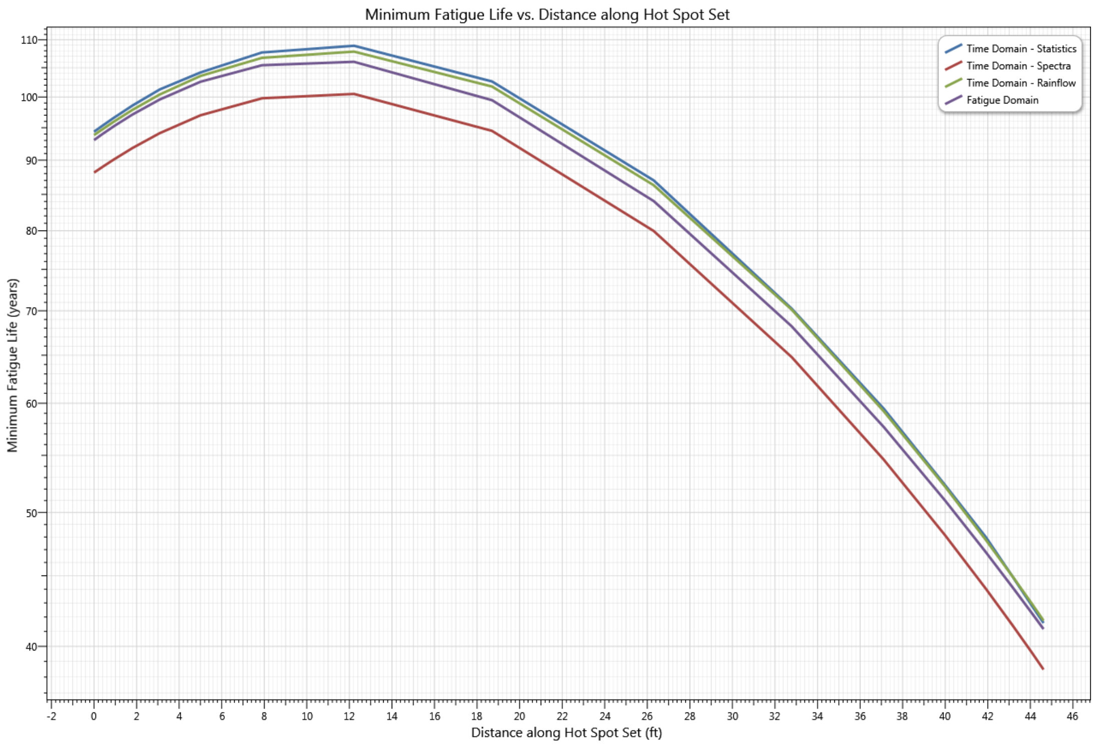
The figure below is a graph produced in Excel from tabular output presented in the output file. It plots the maximum curvature in each mode of the eigensolution, comparing the results at the element integration points, and the equal length segments used in the creation of Shear7 output. The intention here is to allow you to decide if you have specified enough segments to accurately capture the actual modal curvature distribution. In the present case the two sources agree reasonably well, so the specified number of segments appears reasonable.



Maximum Curvatures, Examining Segments

FATIGUE ANALYSIS

The below figure presents a summary of the fatigue analysis results.

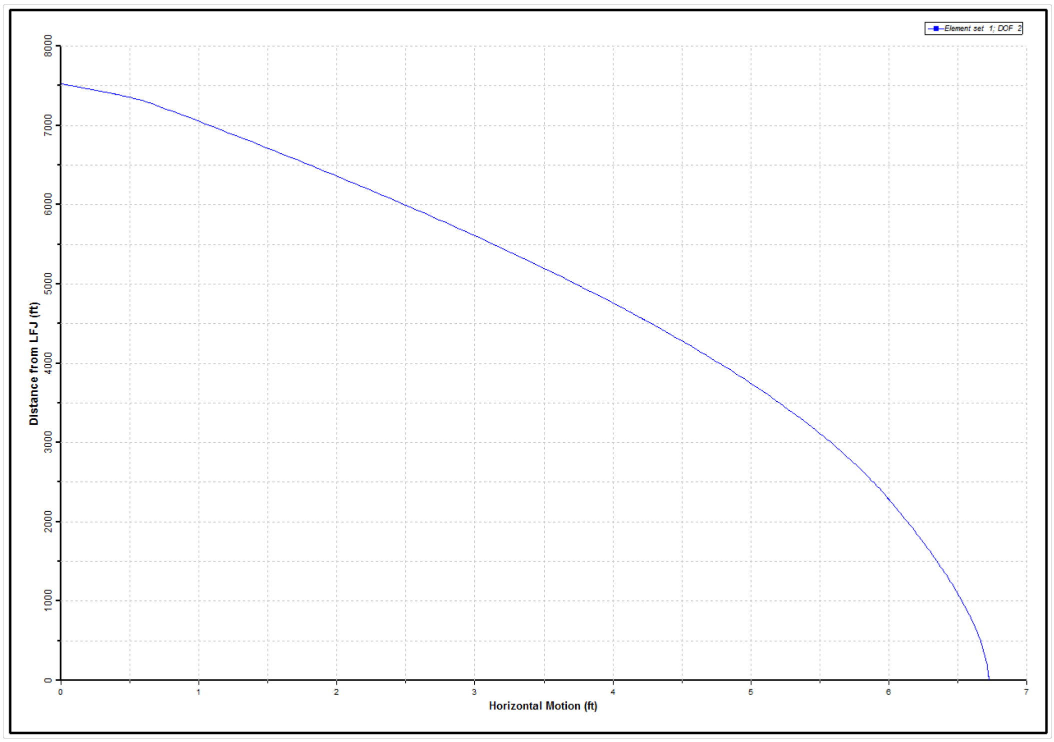


Fatigue Analysis Results

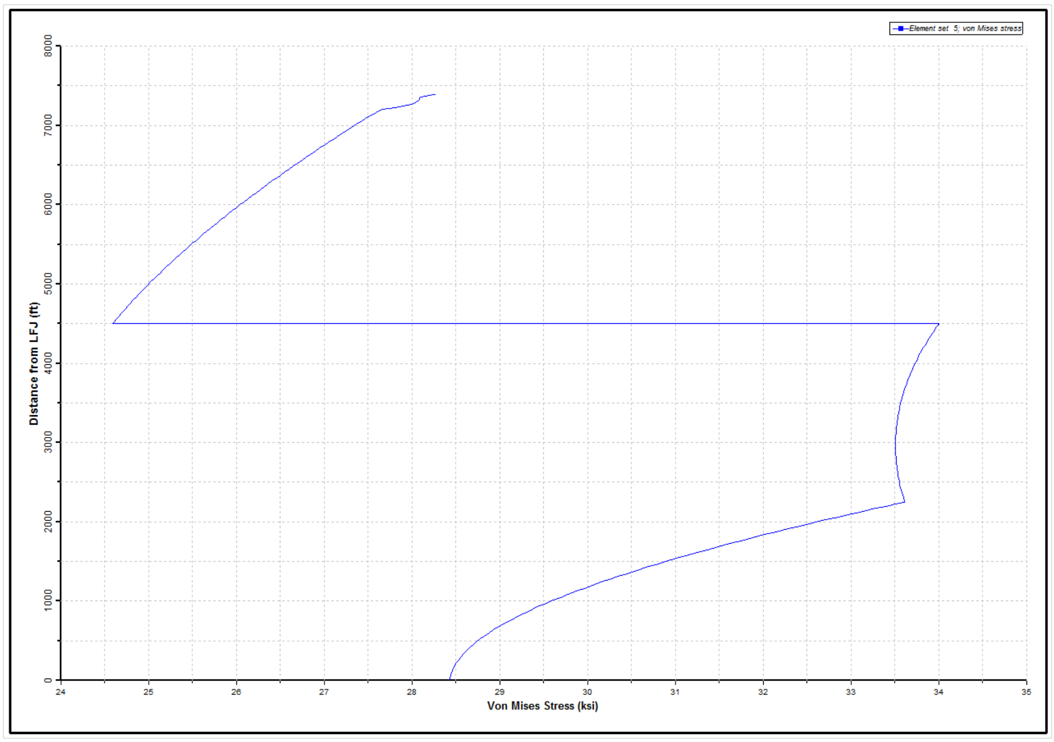
The lowest fatigue life occurs near the upper flex joint. The LifeFrequency and LifeTime results show very close agreement, particularly in view of the differences between the estimates produced by the different methods used within LifeTime itself. Note also that the computational effort required by the two fatigue analyses differ dramatically (i.e. only a few minutes for the frequency domain simulation as opposed to a few hours for the time domain analysis).

Disconnected Mode

Results from the disconnected mode analysis are presented in the Comparing Tension along Riser figure (below) to the Drift-Off LFJ Angle figure. The Drift-Off LFJ Angle and Horizontal Motion along Riser figure (below) are from the static restart with current, and plot respectively horizontal displacement and von Mises stress. The displacement plot shows that the LMRP moves approximately 6.7 ft under the action of the 100-year current. The von Mises stresses are again acceptable.



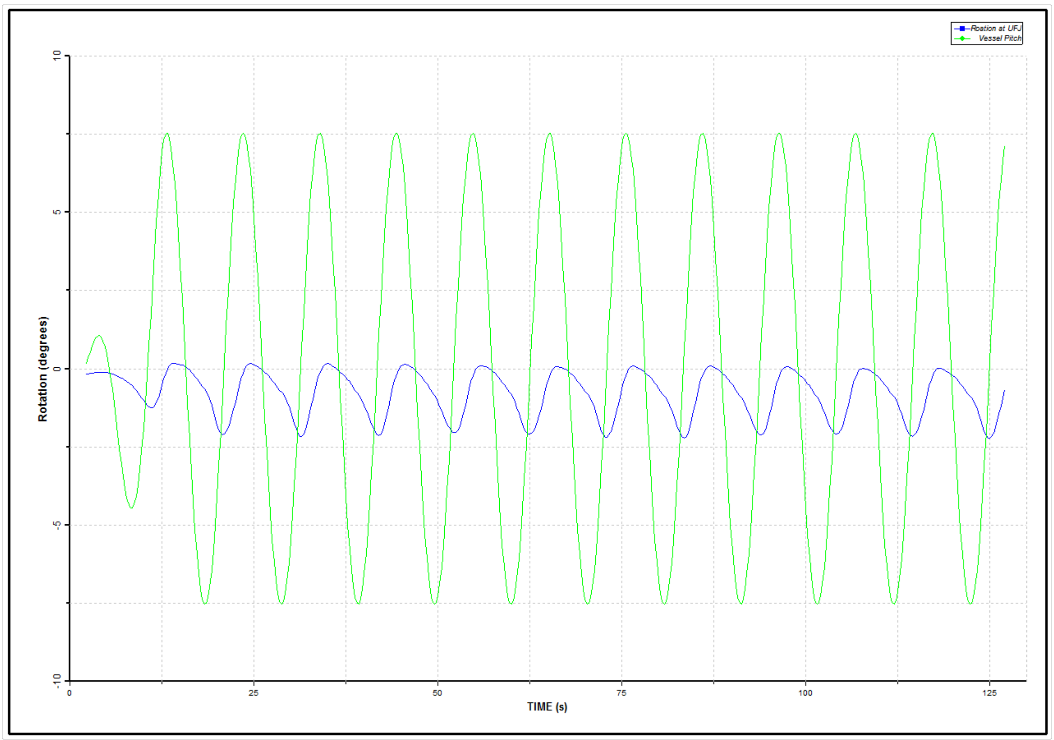
Horizontal Motion along Riser



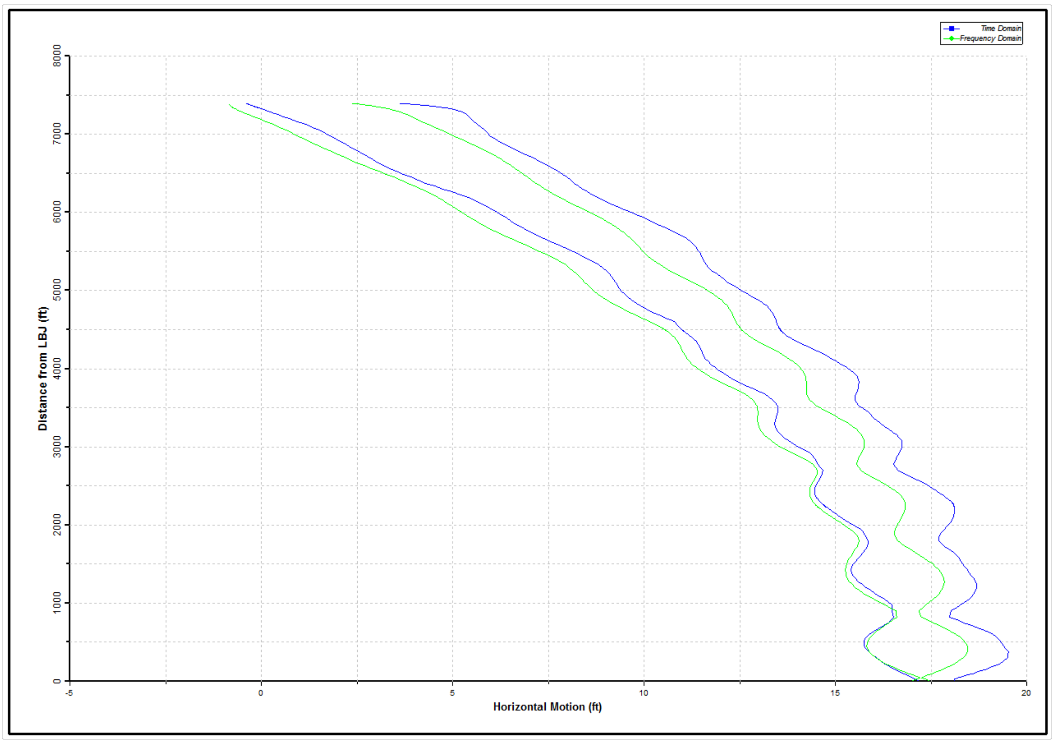
Static Von Mises Stress

The 'Static Von Mises Stress' figure compares timetraces of vessel pitch and the rotation below the UFJ from the time domain solution. The pitch varies between –7.5 degrees and +7.5 degrees, but the corresponding rotations below the UFJ are much less, varying between –2.5 degrees and about 0.5 degrees.

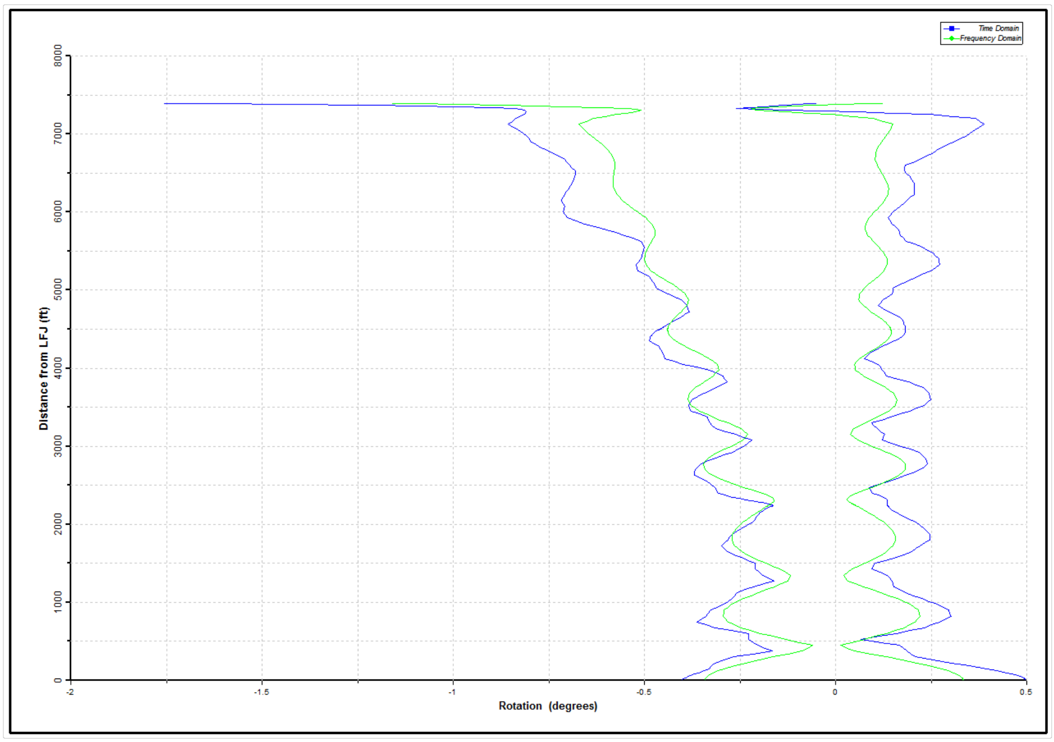
The remaining figures present results from the time and frequency domain dynamic analyses. The Vessel Pitch and Rotation below the UFJ to Comparing Tension along Riser figure plot respectively the distributions of horizontal motion, rotation, effective tension and von Mises stresss in the riser. (In postprocessing the time domain results, envelopes are generated over the last two wave periods). As with the connected mode analysis there is very good agreement between the time and frequency domain approaches. The large dynamic variation in effective tension (Comparing Rotations along Riser figure) due to the vessel heave is noteworthy. The von Mises stress distribution (Comparing Tension along Riser figure below) also shows a relatively large dynamic variation due to the tension variation.



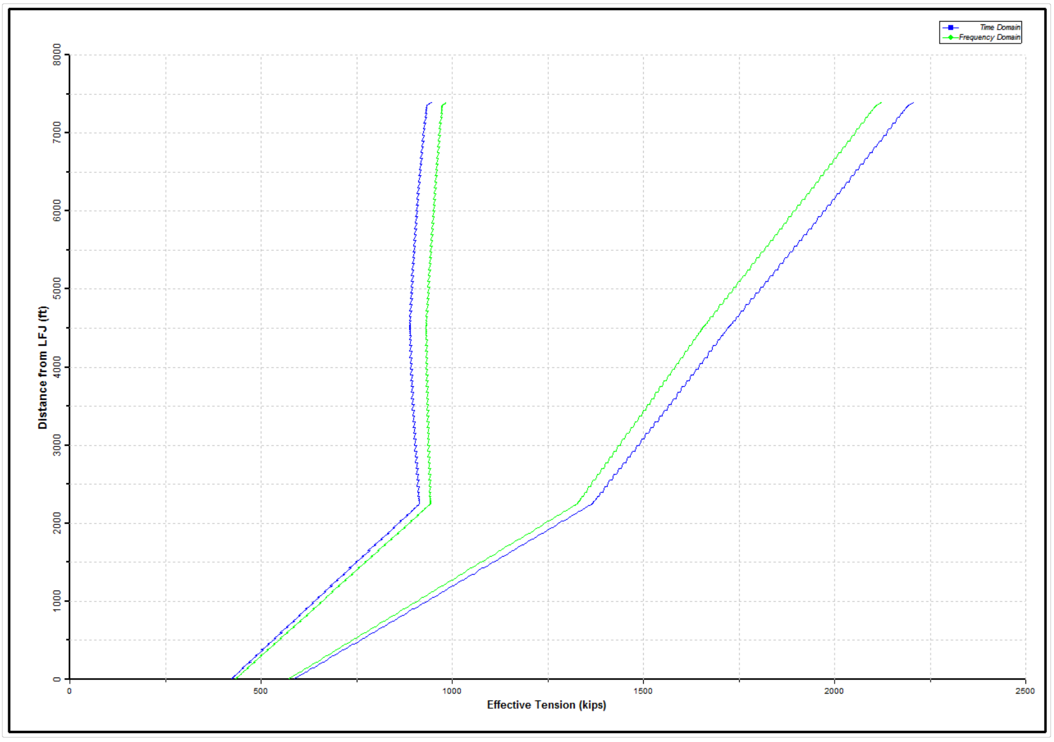
Vessel Pitch and Rotation below the UFJ



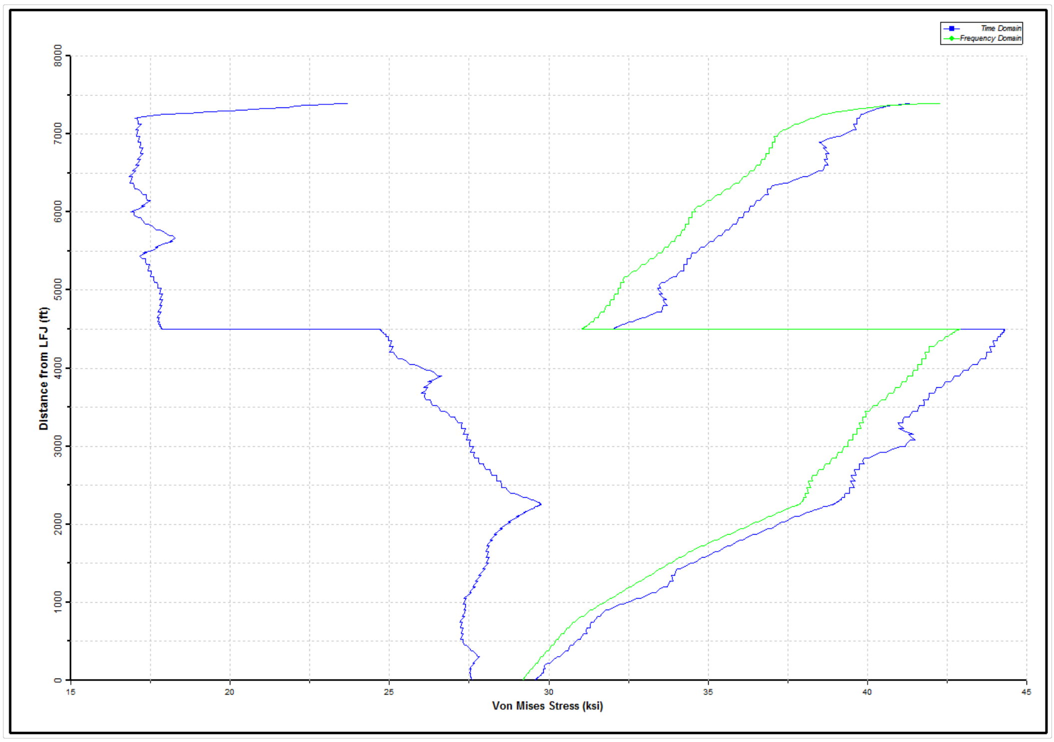
Comparing Horizontal Motion along Riser



Comparing Rotations along Riser



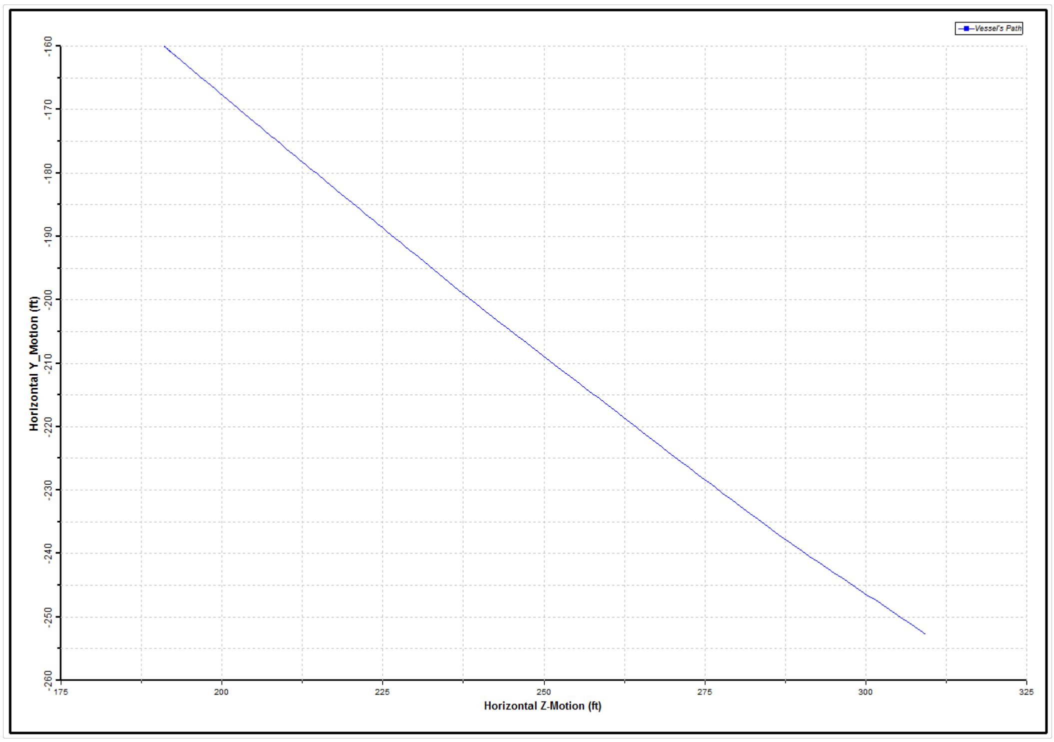
Comparing Tension along Riser



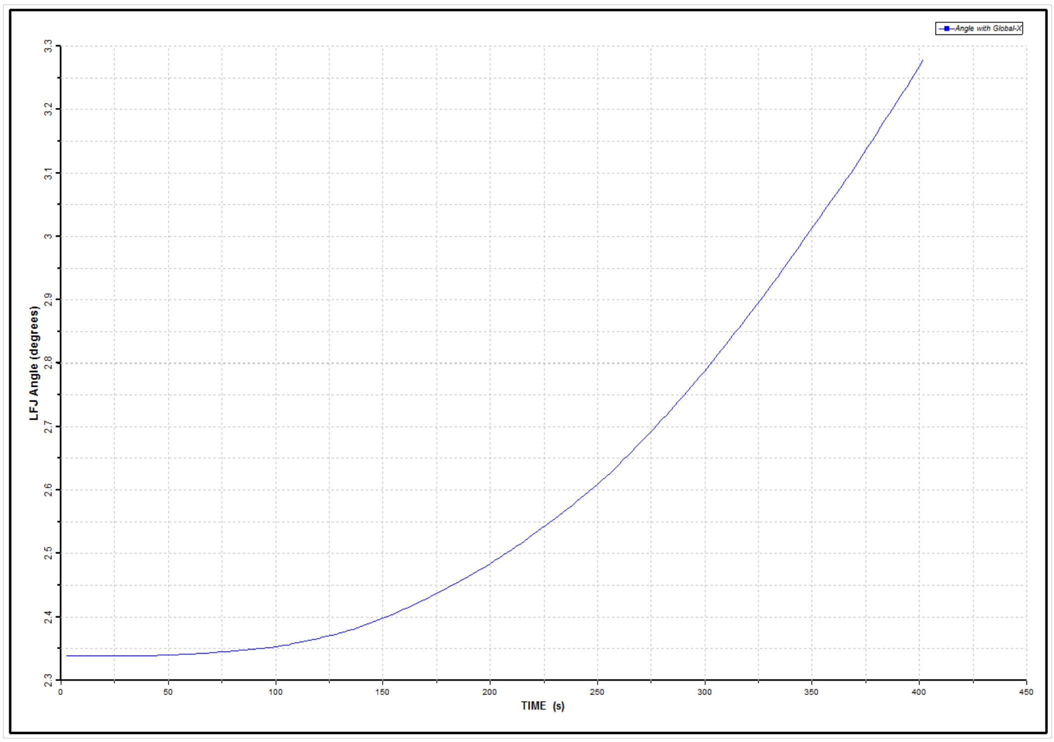
Comparing Von Mises Stress along Riser

Connected Mode, Drift-Off

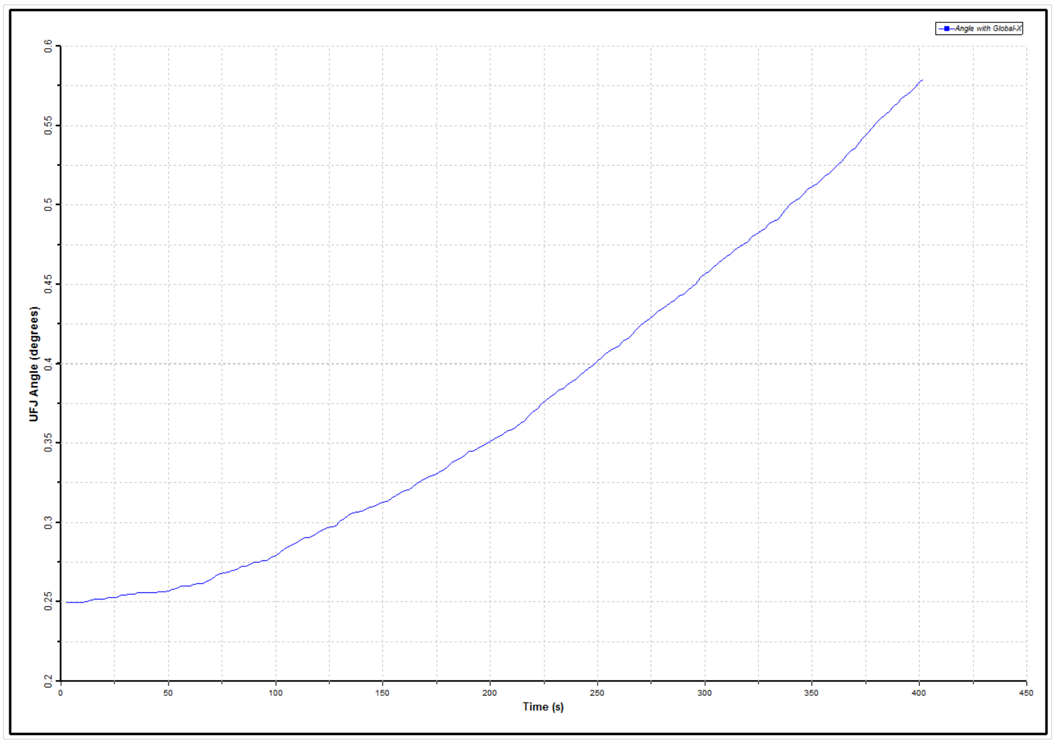
Results from the drift-off mode analysis are presented in the Comparing Von Mises Stress along Riser figure to the Drift-Off UFJ Angle figure (below). The Comparing Von Mises Stress along Riser figure shows a plot of the drift-off path of the vessel. The Drift-Off Path figure (below) and Drift-Off LFJ Angle figure show timetrace plots of the lower and upper flex joint angles, respectively. The Drift-Off LFJ Angle and Drift-Off UFJ Angle figures below show timetrace plots of the bending moments within the BOP and LMRP connectors, respectively.



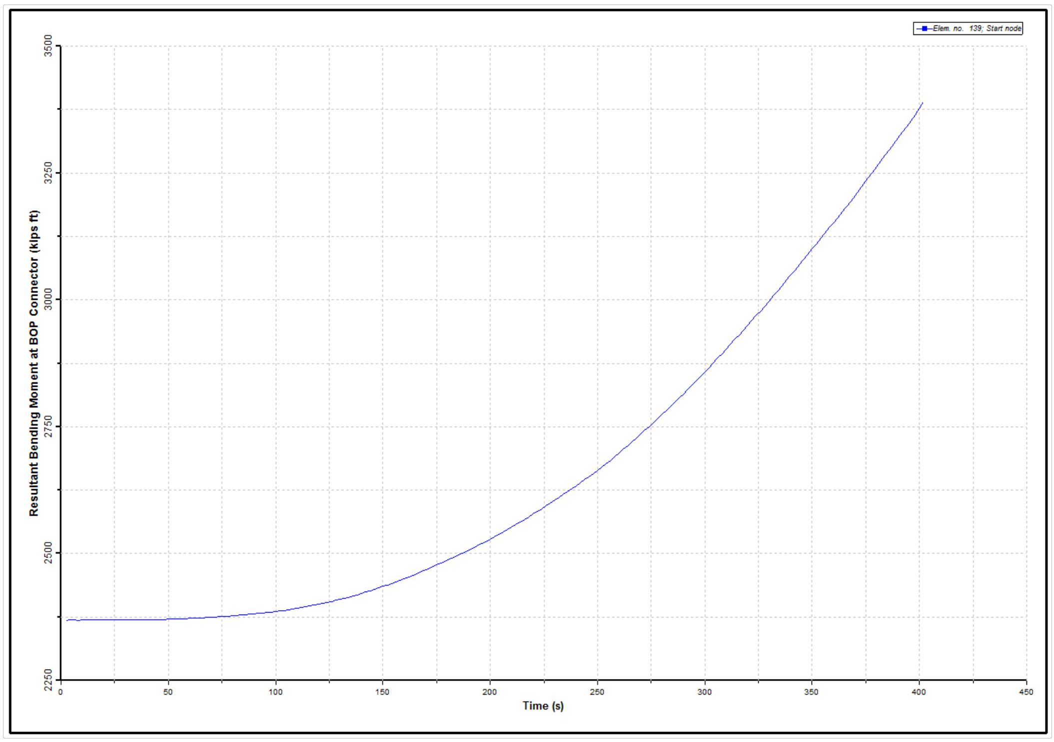
Drift-Off Path



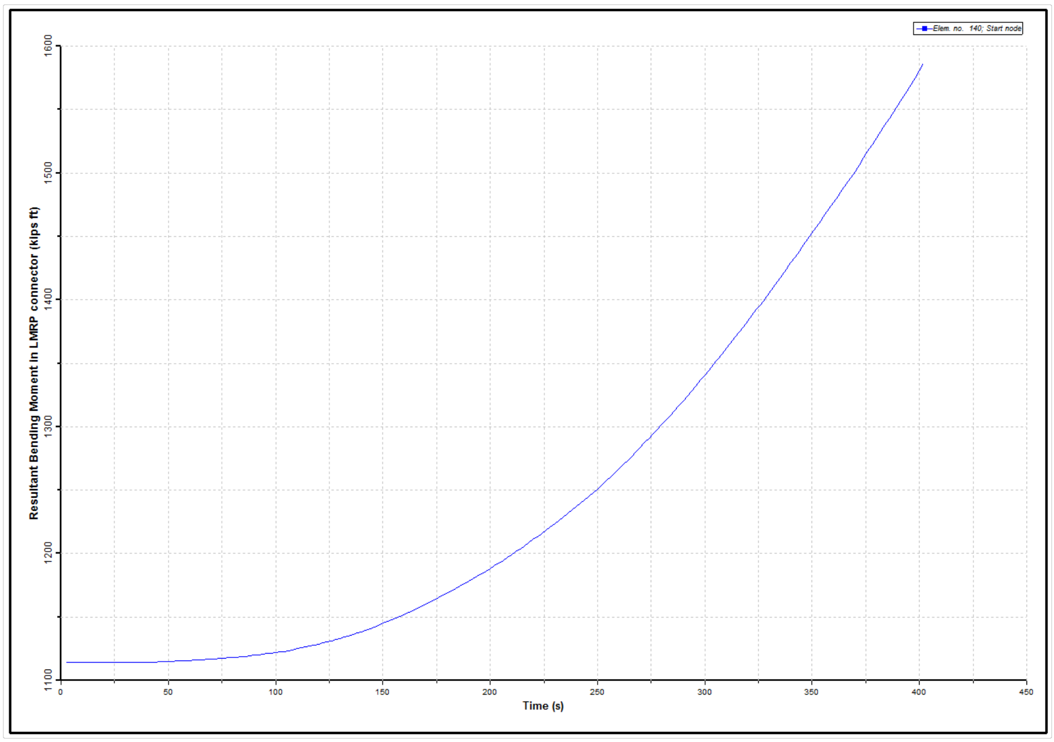
Drift-Off LFJ Angle



Drift-Off UFJ Angle



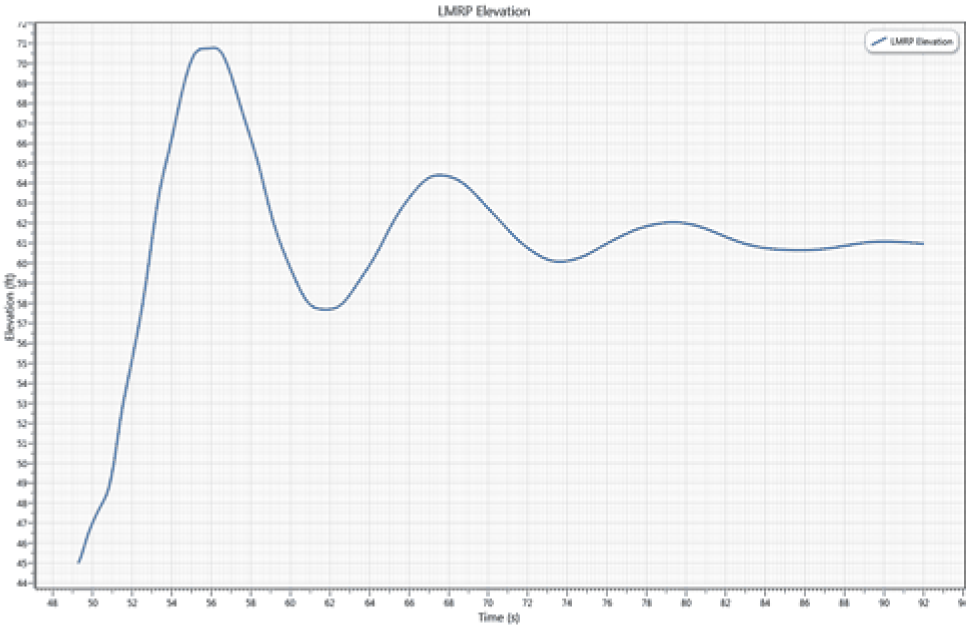
Bending Moment in BOP Connector



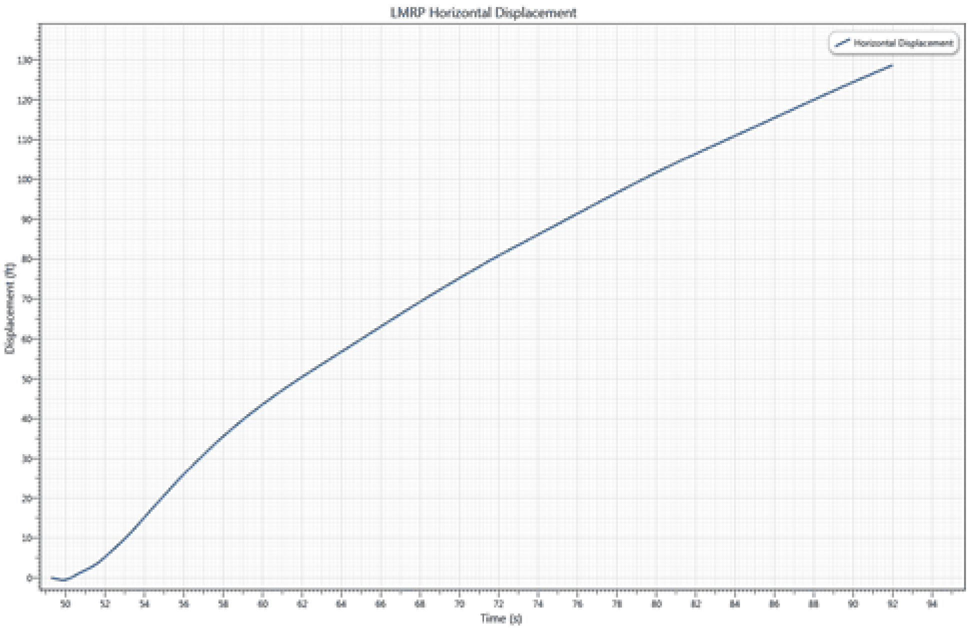
Bending Moment in LMRP Connector

Emergency Disconnect

Results from the riser recoil analysis are presented in the plots below. The first figure plots a time history of the elevation of the LMRP following disconnect, while the second figure plots the horizontal displacement.

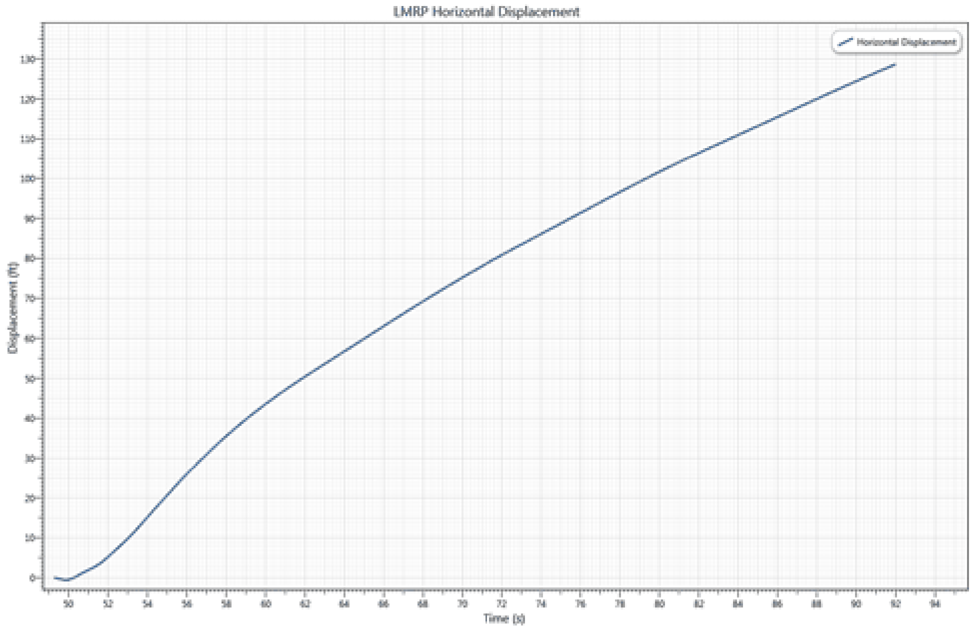


LMRP Elevation



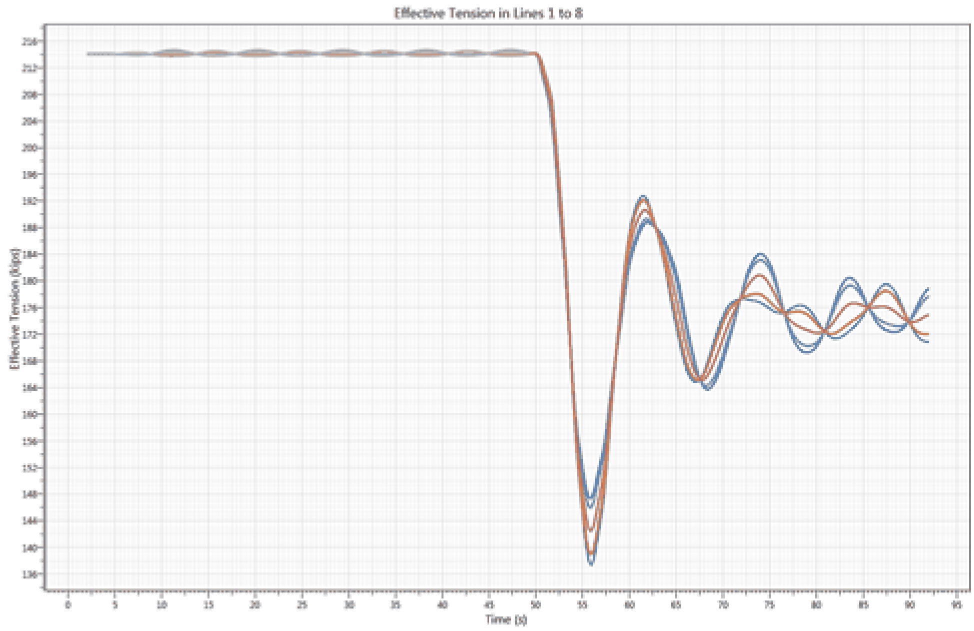
LMRP Horizontal Displacement

The figure below plots the length of the telescopic joint's inner barrel over time. It is shown that the minimum length of the inner barrel immediately after disconnect is approximately 6.3 feet, so it does not collapse completely.



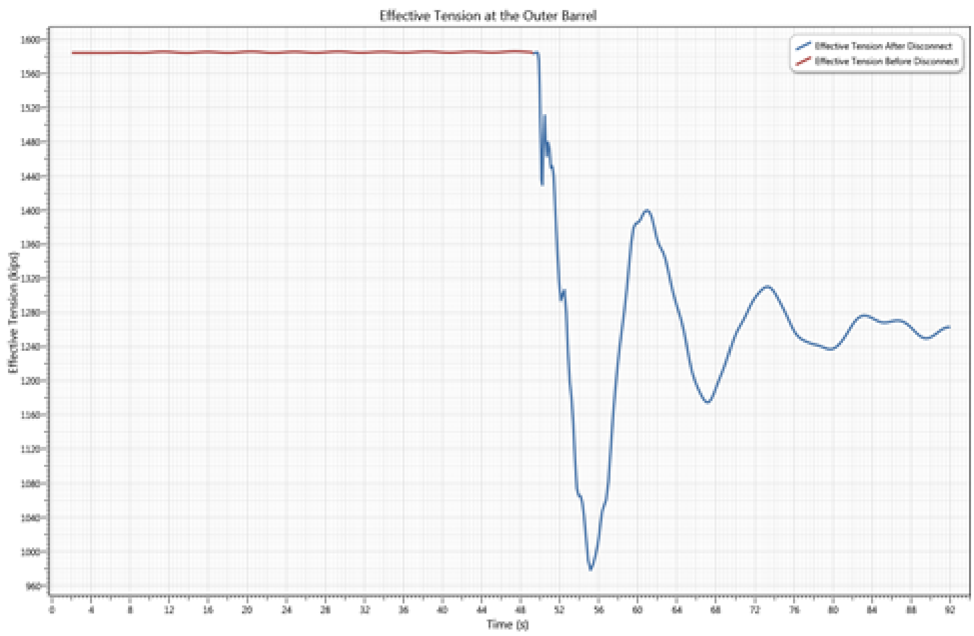
Length of Inner Barrel

Plots of effective tension in each tensioning line are also requested. These are all superimposed onto a single graph for comparison in the figure below. It is seen that, before disconnection, the tension in each line only varies slightly, and in a regular fashion, due to the wave loading and associated vessel response. After disconnection, the tension drops rapidly as the BOP and LMRP become detached from the wellhead and travel upwards. It gradually reaches steady state, as the riser motion is damped out by the damper elements. At all times, the effective tension remains positive, which implies that no slack develops in the tensioning lines.

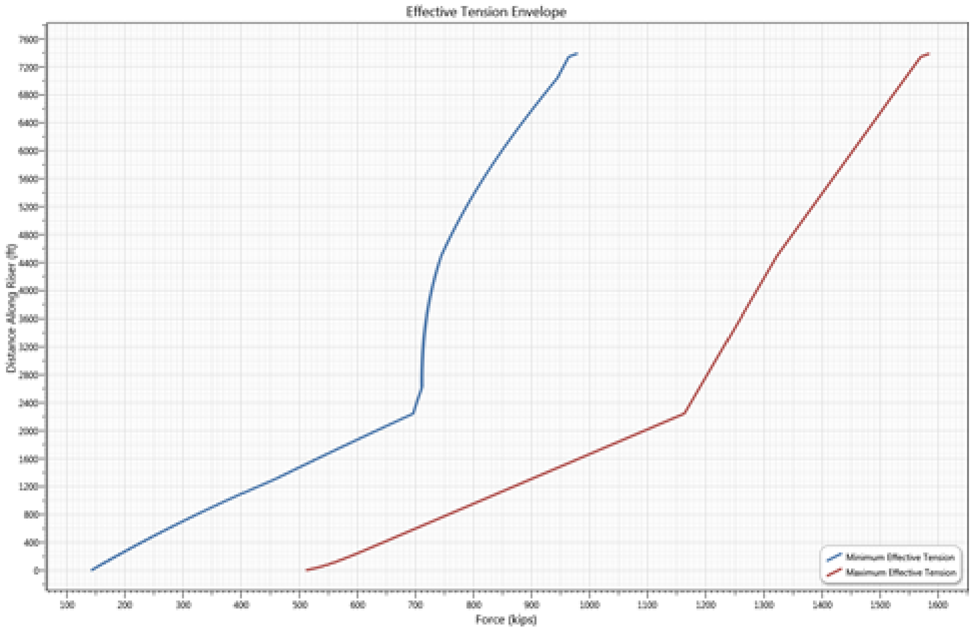


Effective Tension in Tensioning Lines

The figure below shows the variation of effective tension at the interface of the telescopic joint's outer barrel and the slip ring of the tensioning system. The envelope of tension over the complete riser is then plotted in second figure below.



Effective Tension at Outer Barrel/Slip Ring Interface



Envelope of Effective Tension in Riser

Effective compression does not occur in the main riser stack-up, with the minimum reported value approximately 150 kips.

Note: Although Flexcom can provide an approximate simulation of the disconnect event and subsequent riser recoil, a more comprehensive simulation may be performed using our DeepRiser software, which specialises in the analysis of top-tensioned risers. DeepRiser provides a detailed hydro-pneumatic tensioner, which models all the major hydraulic and pneumatic components. It can model each tensioning cylinder independently, and can simulate the behaviour of the anti-recoil control system.

Input Data

This section contains information on:

- Drilling Riser Joints
- Telescopic Joint
- Tensioner
- LMRP
- BOP

- Wellhead Connector
- Conductor/Casing
- Flex Joints
- Internal Fluids
- Vessel – Connected, Normal Operating and Disconnected Modes
- Vessel – Connected Mode, Drift-Off Analysis
- Soil
- Current Loading
- Wave Loading
- Wind Loading
- Wave Scatter Diagram

Drilling Riser Joints

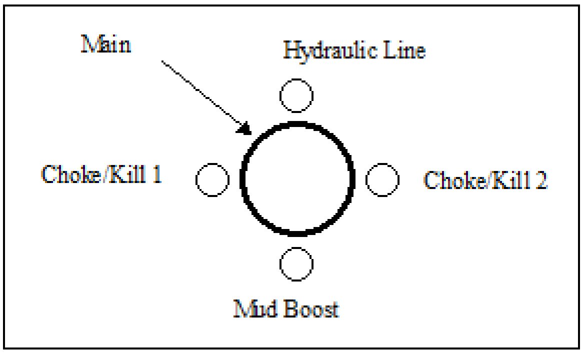
Four drilling riser joints are used to model the main body of the riser. Three correspond to the joints identified as Joints #1, #2 and #3 in the Riser Stack-Up Schematic figure, and a fourth “Termination Joint” models the 2 short pup pieces above these. The table below describes each of these joints in terms of outer diameter, wall thickness, joint length, and whether or not the joint has buoyancy modules attached.

Sample Element Lengths Riser Joint Data

Tubular	Outer Diameter (in)	Thickness (in)	Length (ft)	Buoyancy
Riser Joint # 1	21	0.875	75	✓

Riser Joint # 2	21	0.625	75	✓
Riser Joint # 3	21	0.625	75	×
Termination Joint	21	0.875	45	×

A number of peripheral lines surround the main tubular. These consist of a hydraulic line, two choke and kill lines, and a mud boost line. These peripherals are laid out as shown in the figure below. The 'Peripheral Line Data' table below gives outer and inner diameters for the various lines.



Peripheral Lines

Peripheral Line Data

Tubular	Outer Diameter (in)	Inner Diameter (in)
Choke/Kill 1	5	3

Choke/Kill 2	5	3
Mud Boost Line	4.5	3.75
Hydraulic Line	3.5	3

The buoyancy modules attached to Joints # 1 and # 2 are described in the table below. This gives details of the buoyancy module dimensions, the number of modules per riser joint, and the so-called buoyancy material “densities”. Note that “density” in this context reflects common usage, but the values quoted are not strictly densities as rigorously defined in Flexcom. In Flexcom, density is always taken to mean mass per unit volume, whereas the “density” values in the table below represent instead weight per unit volume.

Buoyancy Module Data

Outside Diameter	Volume per Module (ft^3)	No. of Modules per Joint	“Density” (lb/ft^3)	
			Joint # 1	Joint # 2
47	48	10	26	31

Telescopic Joint

The table below describes the properties of the telescopic joint.

Telescopic Joint Properties

General Properties:	
Normal Drag	1
Normal Inertia	2

Outer Barrel Properties:	
Length	70 ft
Length to Slip Ring	60 ft
External Diameter, Do	26.25 in
Internal Diameter, Di	24 in
Inner Barrel Properties:	
Length	35 ft
External Diameter, Do	21.5 in
Internal Diameter, Di	19.5 in

Note that the inner barrel axial stiffness is not calculated from the above geometric properties. Instead it is assigned an arbitrarily low axial stiffness value so that it extends freely under axial load.

Tensioner

The riser tensioning system is modelled via the inclusion of a group of elements modelling the slip ring and tensioning lines. The slip ring is modelling using rigid elements which connect the tensioning lines to the riser at the telescopic joint. The weight of the slip ring is modelled as a point mass at its connection location. Hinges connect tensioning line and slip ring elements.

Damper elements are included in the recoil analysis. These elements are used to dampen the motion of the riser system following the emergency disconnect.

The properties of the tensioning system are shown below in the table below. Note that the co-ordinates of the tensioner/vessel connections points are given in the global axes.

Details of the Riser Tensioning System

Diameter of Slip Ring	21 in		
Slip Ring Mass	2200 slugs		
Tensioner/Vessel Connections	X [ft]	Y [ft]	Z [ft]
Tension Line 1	7561.75	5.21	0.0
Tension Line 2	7561.75	0.0	5.21
Tension Line 3	7561.75	-5.21	0.0
Tension Line 4	7561.75	0.0	-5.21

LMRP

The geometric and hydrodynamic properties of the Lower Marine Riser Package (LMRP) are shown below in the table below. The LMRP is essentially a large mass connected to the end of the riser. The stresses within the LMRP are not of interest in the analysis, rather the effect that its weight has on the rest of the riser.

LMRP Properties

Length	20 ft
Outer Diameter	60 in
Inner Diameter	18 in
Vertical CdAd	0 ft^2

Vertical CaVin	0 ft^3
Horizontal CdAd	35 ft^2
Horizontal CmVin	96.2 ft^3
Horizontal CaVin	0 ft^3

BOP

The geometric and hydrodynamic properties of the Blow Out Preventer (BOP) are shown below in the table below.

BOP Properties

Length	23.25 ft
Outer Diameter	60 in
Inner Diameter	18 in
Vertical CdAd	0 ft^2
Vertical CaVin	0 ft^3
Horizontal CdAd	116.25 ft^2
Horizontal CmVin	913 ft^3
Horizontal CaVin	0 ft^3

Wellhead Connector

The properties of the wellhead connector are shown below in the table below.

Wellhead Connector Properties

Length	15 ft
Outer Diameter	30 in
Inner Diameter	18 in

Conductor/Casing

The parameters of the casing in this drilling riser model are given in the table below.

Casing Properties

Outer Diameter	36 in
Inner Diameter	19.25 in

Flex Joints

The upper and lower flex joints are modelled using linear flex joint elements. Both joints have a length of 1.75 ft. The rotational stiffnesses are as given in the table below. Nominal small values are assigned for the weights in air and water of both joints.

Flex Joint Properties

Lower Flex Joint (LFJ) Stiffness	30,000 ft lb/°
Upper Flex Joint (UFJ) Stiffness	10,000 ft lb/°

Internal Fluids

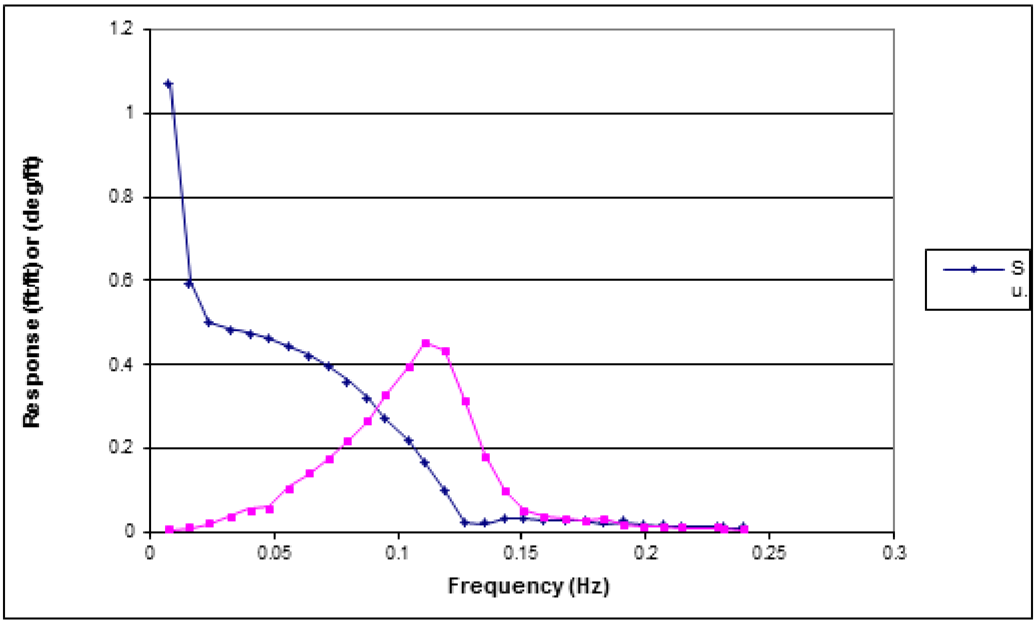
Two internal fluids are used. The first fluid is used to model the drilling mud when the drilling riser is connected. The second fluid has properties corresponding to those of seawater and is used when the drilling riser is disconnected. The fluid densities are given in the table below.

Internal Fluid Data

Connected Mode Density	2.981 slugs/ft^3 (12.827 ppg)
Disconnected Mode Density	1.9876 slugs/ft^3 (seawater)

Vessel – Connected, Normal Operating and Disconnected Modes

The drill ship for these models is defined with its reference point over the origin and 100 ft above the MWL. The first order vessel motions are defined by the specification of an RAO file. The analysis surge and pitch RAOs are plotted in the figure below.



Surge and Pitch RAOs

Vessel – Connected Mode, Drift-Off Analysis

The dimensions and mass parameters of the moored vessel for this model are respectively specified in the two tables below.

Vessel Dimensions

Length from Vessel Centre of Gravity to Forward Perpendicular	300.0 ft
Length from Vessel Centre of Gravity to Aft Perpendicular	300.0 ft
Beam	110.0 ft
Draft	45.0 ft
Longitudinal Wind Area	17500.0 ft^2
Transverse Wind Area	61500.0 ft^2

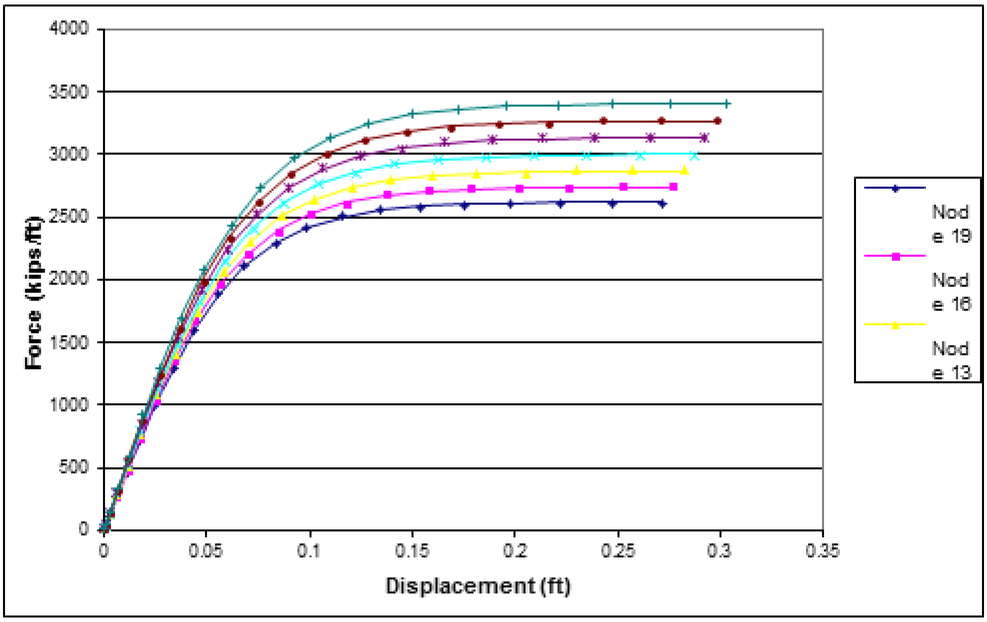
Vessel Mass Parameters

Mass	$1.32x10^5$ slugs
Yaw Inertia	$3.1x10^8$ $slugs.ft^2$
Surge Added Mass	$9.2x10^3$ slugs
Sway Added Mass	$7.9x10^4$ slugs
Yaw Added Mass	$2.5x10^8$ $slugs.ft^2$

Coupled Sway-Yaw Added Mass	0.22 slugs.ft

Soil

Soil structure interaction is modelled by the specification of force-deflection (P-y) curves which define the soil resistance to lateral motion of the structure over a range of depths below the mudline. The figure below presents an illustrative sample of the P-y curve data used in this example.



Sample P-y Curves

Current Loading

Three currents are used in the various analyses. A ‘1-Year Storm’ current profile is used where the riser is connected and operating normally. A ‘10-Year Hurricane’ current profile is used for the disconnected case. A third current is used in the drift-off analysis.

All currents are defined as piecewise-linear using the data given in the table below. Note that each current point is defined by distance below MWL rather than elevation above the seabed.

Current Profiles

1-Year Storm

Depth (ft)	Current (ft/s)
0	1.0
250	1.0
300	0.5
7500	0.5

10-Year Hurricane

Depth (ft)	Current (ft/s)
0	2.093
180	2.093
200	0.304
7500	0.304

Drift-Off

Depth (ft)	Current (ft/s)	Direction (°)
0	2.0	130.0
250	2.0	130.0
300	1.0	130.0
7500	1.0	130.0

Wave Scatter Diagram

A total of 43 seastates are contained within the scatter diagram, representing various combinations of significant wave height H_s and mean zero crossing period T_z. The fatigue seastate data consists of the names of the dynamic analyses and the percentage occurrences of the associated seastates.

Wave Scatter Diagram

		Tz (s)									
		4.1	5.5	6.3	7.4	8.0	9.0	10.1	11.3	12.1	13.0
	1.1	136076	252100	255434	235832						
	2.7		321287	334477	298020	74558					
	4.3		272100	362849	383204	148557	167188				
Hs	5.6			174127	296125	308157	275432	110168			
(ft)	7.4				175352	264871	260697	126533			
	8.6				117157	233065	231621	110175			
	10.1					111420	143196	102307	93240		
	11.8						83161	90355	80073		
	13.3							67966	52943		
	14.5							32699	36523		
	16.1								11986	20976	
	17.9									13739	
	19.1									10200	
	20.5									1520	279

Wave Loading

Four waves are used for the time and frequency domain dynamic analyses. Two are regular waves for ‘1-Year Storm’ and ‘10-Year Hurricane’ conditions respectively. The regular waves use an amplitude of half the maximum wave height given in the table below in conjunction with the corresponding wave period.

The ‘1-Year Storm’ and ‘Drift-Off’ waves are modelled as random seas using the Pierson-Moskowitz wave spectrum with equal area discretisation. The significant wave height and zero crossing period are used as the inputs for these waves. 100 harmonics are requested and the default values are used for the rest of the discretisation parameters.

Wave Data

Parameter		1-Year Storm	10-Year Hurricane	Drift-Off

Maximum Wave Height	(ft)	29.0	45.0	-
Period for H_{max}	(s)	9.0	10.4	-
Significant Wave Height, H_s	(ft)	16.0	26.4	6.0
Zero Crossing Period, T_z	(s)	6.5	11.9	4.0
Direction	(°)	0.0	0.0	130.0

Wind Loading

Wind loading is defined for the drift-off analysis using the data in the table below.

Wind Data

Mean Wind Speed	10 ft/s
Mean Wind Direction	130.0 °

Calculation of Flexcom Input Data

This section describes how the required geometric properties for input to Flexcom are calculated from the data in the Buoyancy Module Data Table to the Telescopic Joint Properties Table. In specifying geometric properties using the Flexcom Rigid Riser format, values are explicitly input for:

- E (Young’s Modulus)
- G (Shear Modulus)
- Do (External Diameter)
- Di (Internal Diameter)
- ρ (Mass Density)
- Dd (Drag Diameter)
- Db (Buoyancy Diameter)

In specifying geometric properties using the Flexcom Flexible Riser format, values are explicitly input for:

- EIyy (Bending Stiffness about Local-y Axis)
- EIzz (Bending Stiffness about Local-z Axis)
- GJ (Torsional Stiffness)
- EA (Axial Stiffness)
- M (Mass per Unit Length)
- P (Polar Inertia of Cross-Section per Unit Length)
- Di (Internal Diameter)
- Dd (Drag Diameter)
- Db (Buoyancy Diameter)

The following abbreviations are also referred to during calculations:

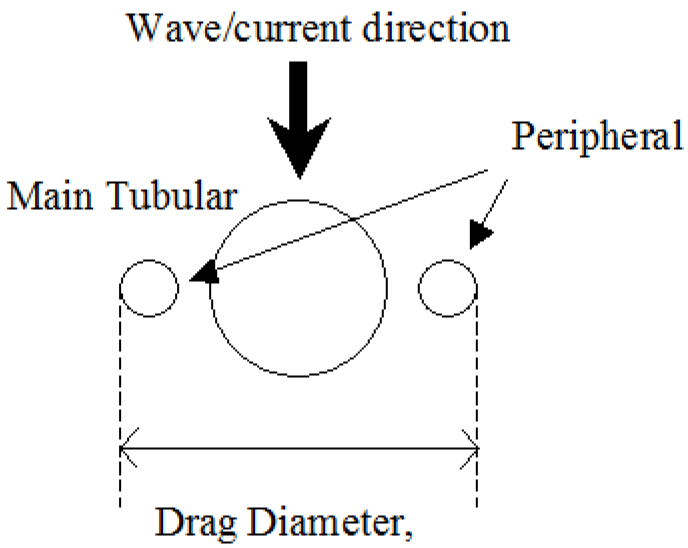
- E_{steel} (Young’s Modulus of Steel)
- G_{steel} (Shear Modulus of Steel)
- ρ_{steel} (Mass Density of Steel)
- T (Thickness)

- A (Cross Sectional Area)
- V (Displaced Volume per Unit Length)
- M_P (Mass per Unit Length of Peripheral Lines)
- V_P (Displaced Volume per Unit Length of Peripheral Lines)
- M_{MOD} (Mass of Single Buoyancy Module)
- V_{MOD} (Volume of Single Buoyancy Module)
- N_{MOD} (Number of Buoyancy Modules per Joint)

General

The mass density calculations are based on first computing a figure for the total mass of each joint, including main tubular, peripherals and buoyancy material if present.

The calculation of equivalent drag diameter depends on whether buoyancy material is attached to a joint or not. For the bare joints, the calculation is based on the below figure. For these joints, Dd is input as (21" + 2*5") = 31", on the basis that this is the most conservative combination of main tubular OD and peripheral OD. For the buoyed joints, the drag diameter is simply specified as the buoyancy module OD of 47".



Drag Diameter Definition

The buoyancy diameter calculations are based on first computing a figure for the total volume of water displaced by each joint, including main tubular, peripherals and buoyancy material if present. An equivalent buoyancy diameter that will give the same volume of displaced water is then calculated for input to Flexcom.

Peripheral Lines

ρ_{steel} = 15.23 slugs/ft^3

Line	Do (in)	Di (in)	CSA (in^2)	EXA (in^2)
Choke/kill 1	5	3	12.566	19.635
Choke/kill 2	5	3	12.566	19.635
Mud boost	4.5	3.75	4.86	15.904
Hydraulic	3.5	3	2.553	9.621
		Total	32.545	64.795

$M_\rho = \rho_{steel}$ * CSA = 3.442 slugs/ft

V_ρ = 0.450 ft^2

Buoyancy Modules

Outside Diameter (in)	Volume per Module (ft^3)	No. of Modules per Joint	"Density" (lb/ft^3)	
			Joint # 1	Joint # 2
47	48	10	26	31

$M_{MOD} = V_{MOD} * N_{MOD} * \rho_{MOD} / L$

= 5.172 slugs /ft (Joint #1)

= 6.166 slugs /ft (Joint #2)

Drilling Riser Joint 1

E_{steel} = 4.32 * 10^9 lb / ft^2

G_{steel} = 1.73 * 10^9 lb / ft^2

Do = 1.750 ft

T = 0.875 in

Di = Do – 2*T = 1.604 ft

A = π/4*(Do^2-Di^2) = 0.384 ft^2

M = (ρ_{steel} * A) * 110% (couplings) + M_P + M_{MOD} = 15.050 slugs /ft

ρ = M / A = 39.175 slugs /ft^3

Dd = 3.917 ft

V = (π/4*(Do^2)) + (V_{MOD} * N_{MOD} / L) + V_P

= (2.405 + 6.400 + 0.450) = 9.255 ft^2

Db = 3.433 ft

Drilling Riser Joint 2

E_{steel} = 4.32 * 10^9 lb / ft^2

G_{steel} = 1.73 * 10^9 lb / ft^2

Do = 1.750 ft

T = 0.625 in

Di = Do – 2*T = 1.646 ft

A = π/4*(Do^2-Di^2) = 0.278 ft^2

EA =1.200 * 10^9 lb

M = (ρ_{steel} * A) * 110% (couplings) + M_P + M_{MOD} = 14.263 slugs /ft

ρ = M / A = 51.337 slugs /ft^3

Dd = 3.917 ft

V = (π/4*(Do^2)) + (V_{MOD} * N_{MOD} / L) + V_P

= (2.405 + 6.400 + 0.450) = 9.255 ft^2

Db = 3.433 ft

Drilling Riser Joint 3

E_{steel} = 4.32 * 10^9 lb / ft^2

G_{steel} = 1.73 * 10^9 lb / ft^2

Do = 1.750 ft

T = 0.625 in

Di = Do – 2*T = 1.646 ft

A = π/4*(Do^2-Di^2) = 0.278 ft^2

M = (ρ_{steel} * A) * 110% (couplings) + M_P = 8.097 slugs /ft

ρ = M / A = 29.142 slugs /ft^3

Dd = 2.583 ft

V = (π/4*(Do^2)) + V_P = (2.405 + 0.450) = 2.855 ft^2

Db = 1.907 ft

Termination Joint

E_{steel} = 4.32 * 10^9 lb / ft^2

G_{steel} = 1.73 * 10^9 lb / ft^2

Do = 1.750 ft

T = 0.875 in

Di = Do – 2*T = 1.604 ft

A = $\pi/4*(Do^2-Di^2)$ = 0.384 ft^2

M = $(\rho_{steel}$ * A) * 110% (couplings) + M_P = 9.878 slugs /ft

ρ = M / A = 25.713 slugs /ft^3

Dd = 2.583 ft

V = $(\pi/4*(Do^2)) + V_P$ = (2.405 + 0.450) = 2.855 ft^2

Db = 1.907 ft

Outer Barrel

E_{steel} = 4.32 * 10^9 lb / ft^2

G_{steel} = 1.73 * 10^9 lb / ft^2

Do = 2.188 ft

Di = 2.000 ft

A = $\pi/4*(Do^2-Di^2)$ = 0.617 ft^2

M = (ρ_{steel} * A) * 110% (couplings) = 10.331 slugs /ft

ρ = M / A = 16.753 slugs /ft^3

Dd = 2.188 ft

V = π/4*(Do^2) = 3.758 ft^2

Db = 2.188 ft

Inner Barrel

Do = 1.792 ft

Di = 1.625 ft

E_{steel}= 4.32 * 10^9 lb / ft^2

Iyy = π/64*(Do^4-Di^4) = 0.164 ft4

EIyy = E_{steel} * Iyy = 7.065 * 10^8 lb.ft^2

EIzz = EIyy =7.065 * 10^8 lb.ft^2

G_{steel}= 1.73 * 10^9 lb / ft^2

J = 2 * Iyy =0.327 ft^4

GJ = 5.652 * 10^8 lb.ft^2/rad

A = π/4*(Do^2-Di^2) = 0.447 ft^2

EA =100 lb (nominal)

M = (ρ_{steel} * A) * 110% (couplings) = 7.493 slugs /ft

Dd = 1.792 ft

V = π/4*(Do^2) = 2.521 ft^2

Db = 1.792 ft

Tensioner

EIyy = EIzz = $1.000 * 10^{10}$ lb.ft^2 (nominal)

GJ = $1.000 * 10^{10}$ lb.ft^2 /rad (nominal)

Do = Di = Db = Db = 0.000 ft (nominal)

M = 0.000 slugs /ft (nominal)

Slip Ring

EIyy = EIzz = EA = $1.000 * 10^9$ lb.ft^2 (nominal)

GJ = $1.000 * 10^9$ lb.ft^2 /rad (nominal)

Do = Db = Db = 0.010 ft (nominal)

Di = 0.001 ft (nominal)

M = 0.010 slugs /ft (nominal)

Conductor/Casing

E_{steel} = $4.32 * 10^9$ lb / ft^2

G_{steel} = $1.73 * 10^9$ lb / ft^2

Do = 3.000 ft

Di = 1.604 ft

A = $\pi/4*(Do^2-Di^2)$ = 5.047 ft^2

M = $(P_{steel} * A)$ = 76.873 slugs /ft

ρ = M / A = 15.230 slugs /ft^3

Dd = 3.000 ft

V = $\pi/4*(Do^2)$ = 7.069 ft^2

Db = 3.000 ft

Wellhead

E_{steel} = 4.32 * 10^9 lb / ft^2

G_{steel} = 1.73 * 10^9 lb / ft^2

Do = 2.500 ft

Di = 1.500 ft

A = $\pi/4*(Do^2-Di^2)$ = 3.142 ft^2

M = $(P_{steel}$ * A) * 148.5% (Wellhead structure) = 71.073 slugs /ft

ρ = M / A = 22.623 slugs /ft^3

Dd = 2.500 ft

V = $(\pi/4*(Do^2 - Di^2))$ * 148.5% (Wellhead structure) = 4.667 ft^2

Db = 2.438 ft

BOP

E_{steel} = 4.32 * 10^9 lb / ft^2

G_{steel} = 1.73 * 10^9 lb / ft^2

Do = 5.000 ft

Di = 1.500 ft

A = $\pi/4*(Do^2-Di^2)$ = 17.868 ft^2

M = (P_{steel} * A) * 150.0% (BOP structure) = 408.095 slugs /ft

ρ = M / A = 22.840 slugs /ft^3

Dd = 5.000 ft

V = (π/4*(Do^2 -Di^2)) * 148.5% (BOP structure) = 26.795 ft^2

Db = 5.841 ft

LMRP

E_{steel} = 4.32 * 10^9 lb / ft^2

G_{steel} = 1.73 * 10^9 lb / ft^2

Do = 5.000 ft

Di = 1.500 ft

A = π/4*(Do^2-Di^2) = 17.868 ft^2

M = (P_{steel} * A) * 150.0% (LMRP structure) = 408.164 slugs /ft

ρ = M / A = 22.844 slugs /ft^3

Dd = 5.000 ft

V = (π/4*(Do^2 -Di^2)) * 148.5% (BOP structure) = 26.800 ft^2

1.10.1.2 A02 - Spar Production Riser

This example describes an example analysis of a spar production riser, and demonstrates a number of features of Flexcom specifically suited to this application. The overall layout of the example is as follows:

- Introduction gives an overview of the production riser analysis, and notes some of the more important features of Flexcom which are relevant to the analysis.

- Model Summary describes the model in more detail, and discusses the relevant analytical capabilities of the software.
- Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions to which the riser is subjected.
- Results presents pertinent results from the various analyses performed and discusses their significance.

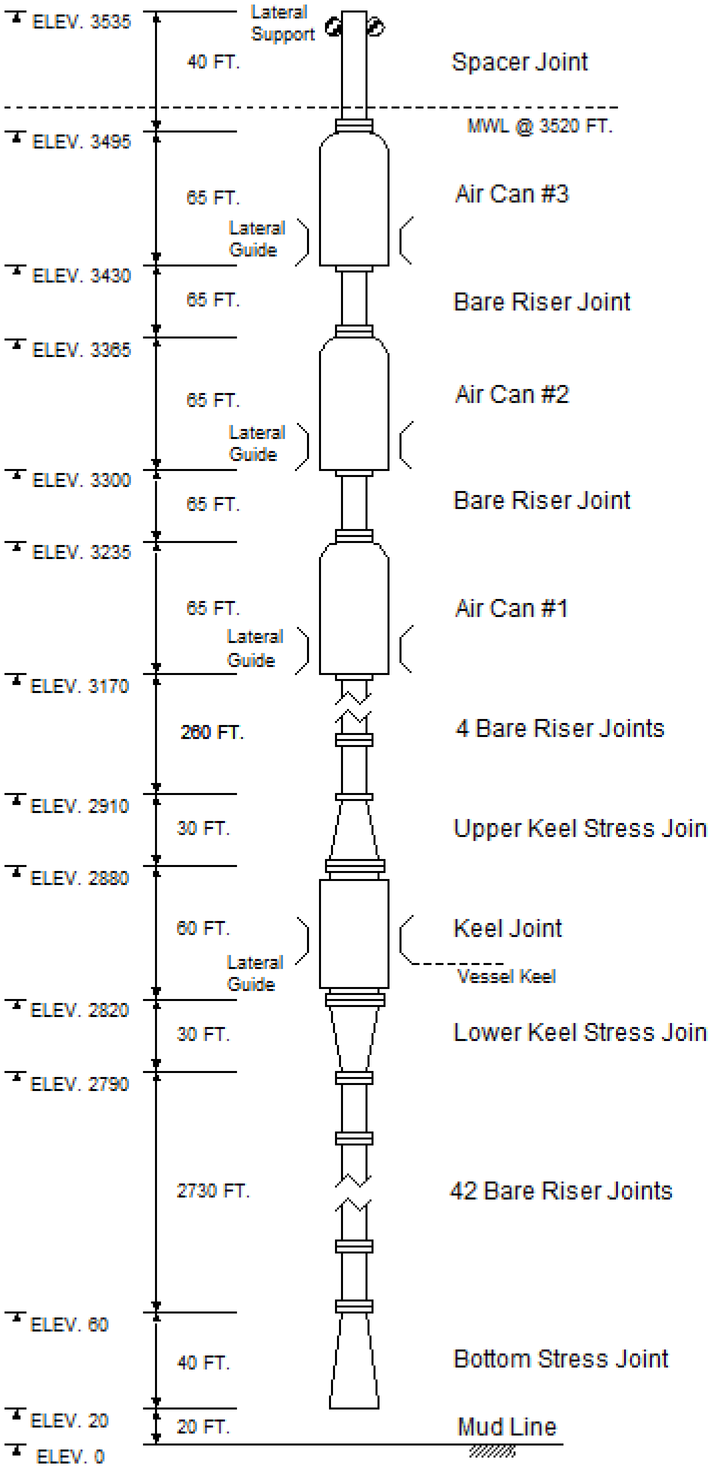
Introduction

This example considers the analysis of a buoyantly tensioned spar production riser. The example consists of a static analysis of the riser subject to current loads and vessel offset, and a dynamic analysis of the riser subject to wave loading and first-order vessel motions.

This example illustrates a number of important modelling features in Flexcom relevant to this type of analysis. Specifically, the following capabilities are demonstrated:

- Modelling of contact between the riser/air cans and hull-mounted guides.
- Modelling of hydrodynamic loading within the spar moonpool.

A schematic of the riser stack-up is shown in the below figure.



Riser Stack-Up Schematic

Model Summary

GENERAL

The riser in this example is a model of a buoyantly tensioned spar production riser. The riser is oil filled and of single-casing type, and is situated in 3520 ft of water. A stress joint is located at the riser base and between this and the vessel keel are 42 bare riser joints. A keel joint is located at the point of entry of the riser into the vessel moonpool, with tapered stress joints above and below the keel joint. A riser guide, which is attached to the spar, is located around the keel joint at its midpoint. Above the upper keel stress joint are a further 4 bare riser joints. This section of the riser is followed by a section comprising 3 riser joints fitted with air cans. The riser joints with air cans are each separated by a single bare riser joint. Hull-mounted air can guides, which are located towards the bottom of each of the air cans, provide lateral support. Finally a spacer joint is located above the top air can.

RISER GUIDES

The riser guides are modelled using zero-gap guides which provide lateral support. A zero-gap guide may be thought of as a cylindrical sheath positioned around a section of riser, with the internal diameter of the sheath equal to the external diameter of the riser. So there is a contact clearance of zero between the guide and the riser. The zero-gap contact algorithm works by checking the position of nodes at each solution step for contact with zero-gap supports. If a node comes into contact with a zero-gap guide, appropriate boundary conditions are applied at the node in local axes which are both perpendicular to the longitudinal guide axis. Unlike flat guide surfaces, reaction forces are not considered when releasing the restraints. Rather, the node is free to move in the axial direction (subject to frictional restraints) through the zero-gap guide, and whenever the node leaves the contact region, the boundary conditions are removed.

MOONPOOL HYDRODYNAMICS

Hydrodynamic loading within the spar moonpool is modelled by assigning moonpool hydrodynamic coefficients to elements of the structure that may be subjected to hydrodynamic loading within the vessel moonpool. The program checks at each solution time whether each integration point on a given element is within the volume enclosed by the vessel moonpool at that particular time, or if it is in the so-called transition region, or if it is completely outside the influence of the vessel moonpool. If it is within the volume enclosed by the moonpool, then the water particle velocities and accelerations used in calculating the Morison's Eq. force at the integration point are calculated from the vessel motions. If the integration point is outside the influence of the moonpool, the water particle velocities and accelerations are calculated from the ambient wave field. If the integration point is within the transition region, then the water particle velocities and accelerations are interpolated linearly from those in the moonpool and those in the wave field. In this way, the program automatically accounts for elements that may move in and out of the vessel moonpool during the course of an analysis.

Analyses

INITIAL STATIC ANALYSIS

The base node, Node 1, is fixed in all 6 degrees of freedom. Vessel BCs are specified at the uppermost node, Node 507, in DOFs 2, 3, 5, and 6. The initial position of the vessel reference point and the RAOs are also specified.

OFFSET ANALYSIS

The only change here is that a vessel offset of 50 ft is applied in the horizontal (DOF 2) direction. The BCs remain unchanged and are carried through automatically from the initial static analysis.

CURRENT ANALYSIS

A piecewise-linear current is applied to the riser. The current is ramped on over ten increments. The BCs again remain unchanged and are carried through automatically from the offset analysis.

DYNAMIC ANALYSIS

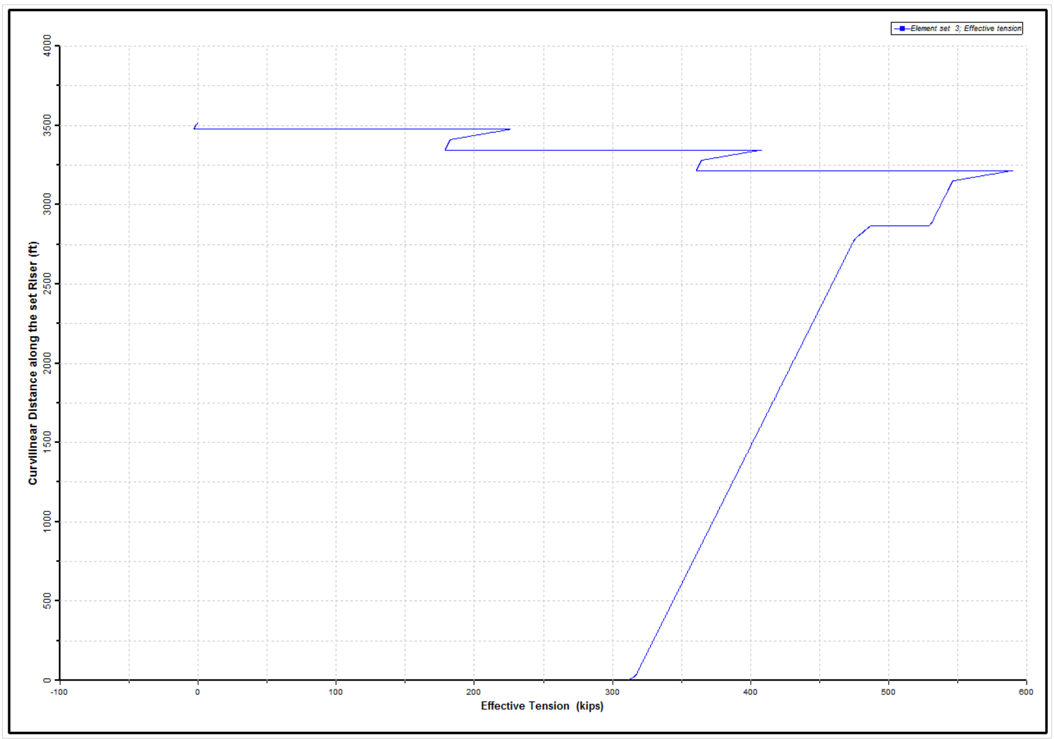
Once again the BCs remain unchanged and are carried through automatically from the current analysis. Since vessel BCs and RAO data have previously been input, dynamic motions are automatically applied with the onset of wave loading.

Results

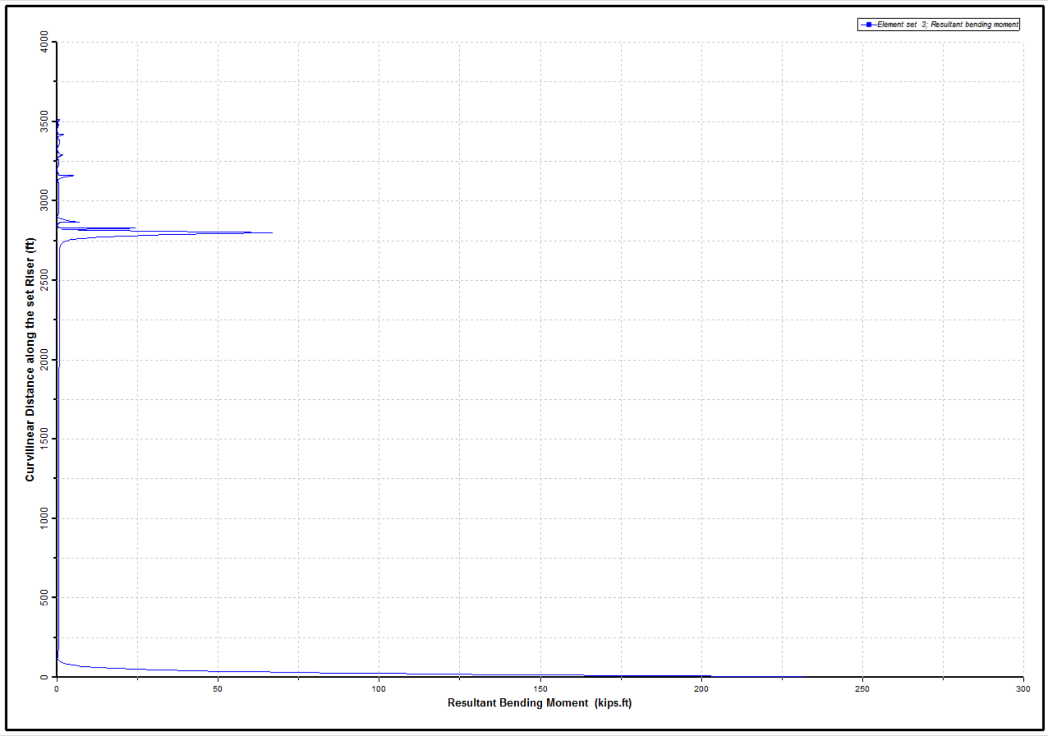
CURRENT ANALYSIS

The first figure below shows the distribution of effective tension in the riser after the current analysis. The tensioning effect of each of the three air cans is evident, as is the transfer of load from the keel sleeve to the riser at the top of the keel joint.

The second figure below shows the distribution of static resultant bending moment in the riser. Note the peak in bending moment around the location of the keel joint. This indicates that, as the vessel has been offset horizontally, the contact between the keel sleeve and its lateral guide has resulted in a horizontal contact reaction that gives rise to the peak in bending moment.



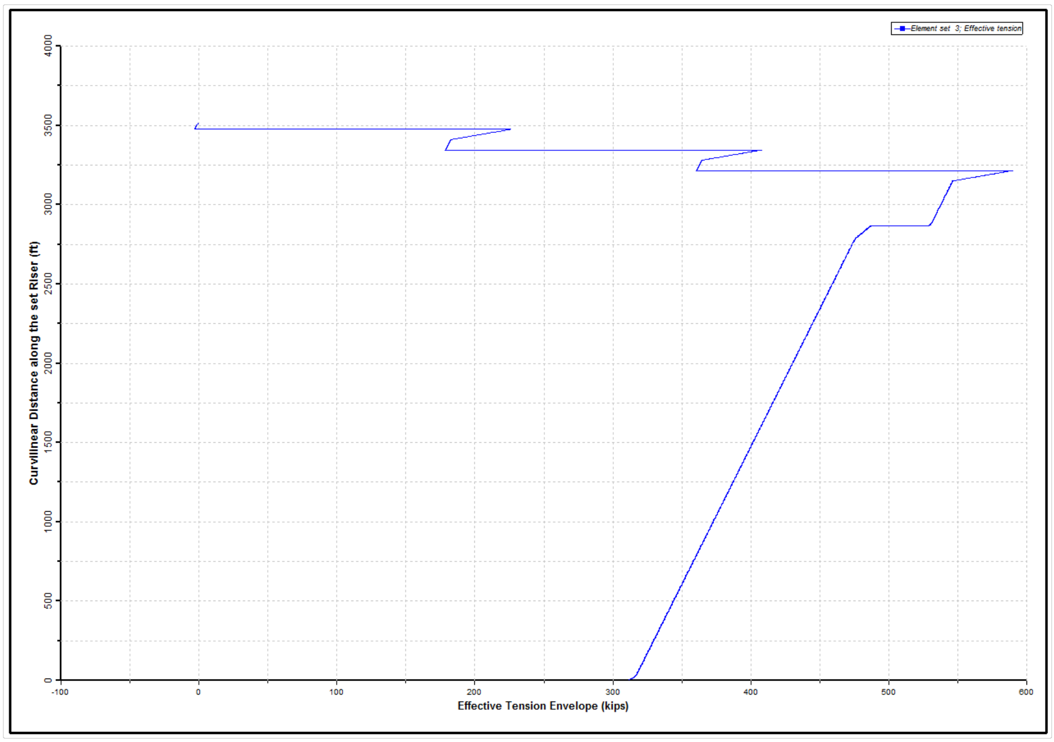
Static Effective Tension



Static Resultant Bending Moment

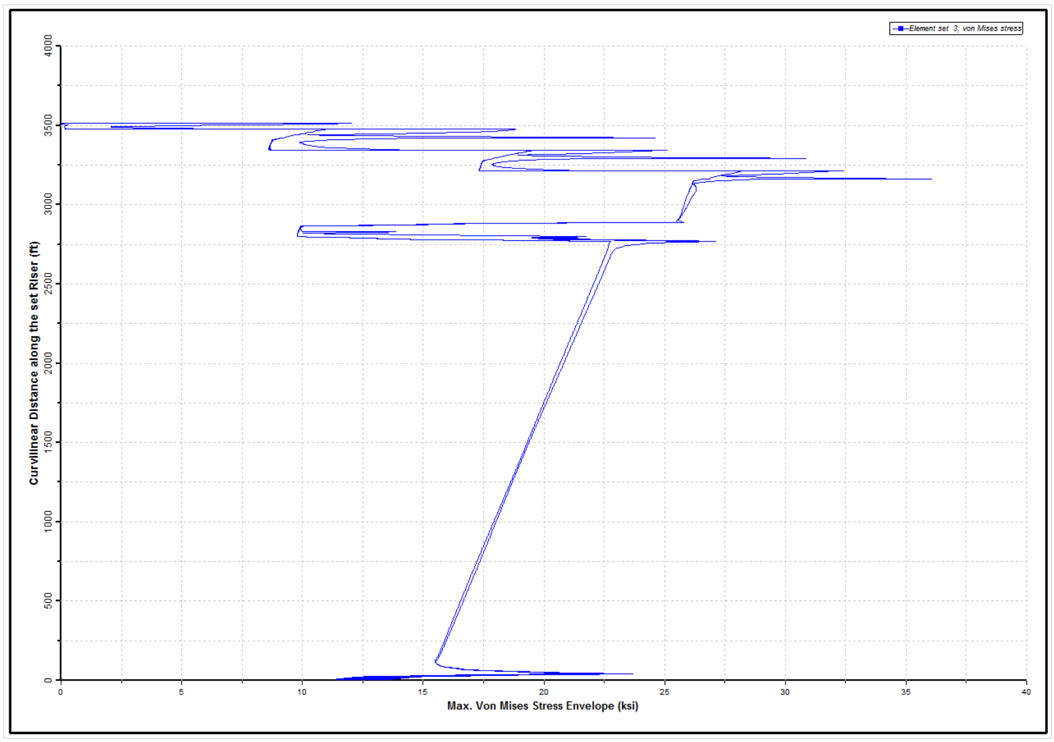
DYNAMIC ANALYSIS

The envelopes of dynamic effective tension in the riser are plotted in the figure below, which shows a relatively small variation in effective tension over much of the riser for the dynamic analysis.

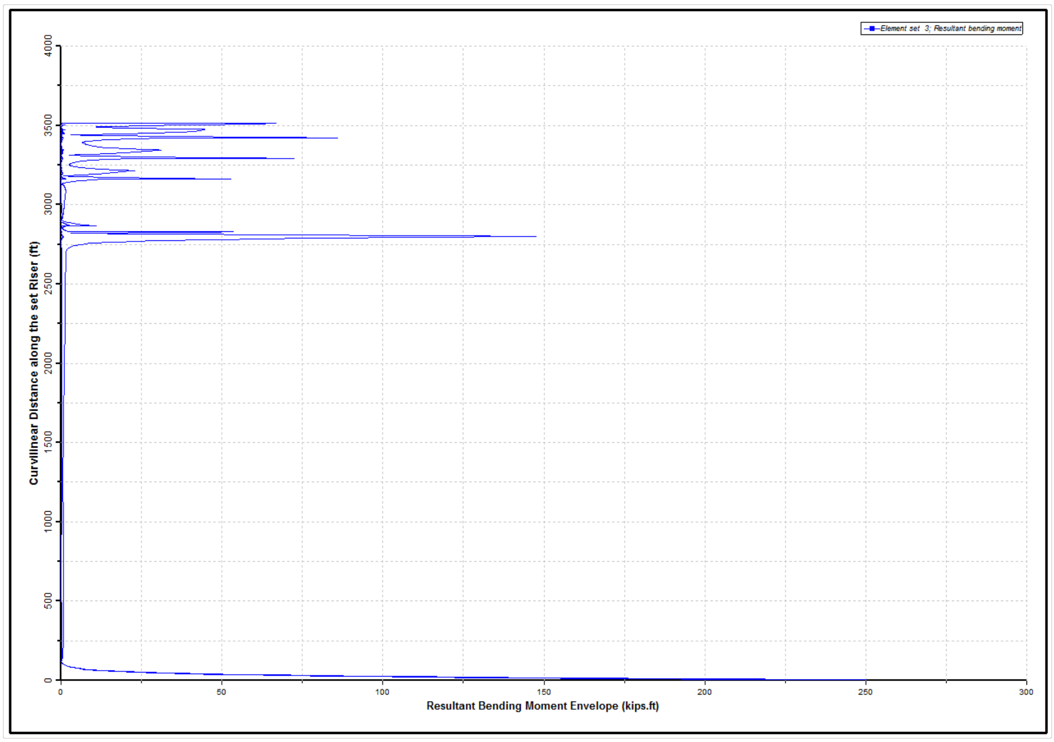


Effective Tension Envelope

The first figure below plots envelopes of von Mises stress. Note the peaks in stress in the region of the air cans, which can be largely attributed to areas of locally high bending moment resulting from contact between the air cans and lateral guides. Finally, the second figure below shows the dynamic bending moment envelopes. Comparison with the Static Resultant Bending Moment figure above shows that contact is now occurring at the air cans in addition to the keel joint.



Maximum Von Mises Stress Envelope



Resultant Bending Moment Envelope

1.10.1.3 A03 - Pipe-in-Pipe Production Riser

This example describes analyses of a dual barrier Spar production riser which invoke the Flexcom pipe-in-pipe modelling option. The overall layout of the example is as follows:

- Introduction gives an overview of the example, and notes some of the more important relevant features of Flexcom.
- Model Summary describes the actual model in more detail.
- Analyses briefly describes the various analyses performed, discussing the environmental and loading conditions to which the riser is subjected.
- Results presents pertinent results from the example analyses and discusses their significance.

Introduction

OVERVIEW

This example considers the analysis of a dual barrier production riser, which is modelled as three concentric tubes using the Flexcom pipe-in-pipe modelling facility. The model comprises a 12¾" OD outer riser, an $8^5/_8$" OD inner riser and 4½" OD production tubing; each has a thickness of ½". The outer and inner risers extend from the spar deck to the mudline, while the tubing extends from the deck to a packer 8000 ft below the mudline (the water depth is 4000 ft). The annulus between outer and inner risers contains fluid with a density of 9 ppg, while the inner riser/tubing annulus has fluid with a density of 2 ppg. The fluid in the tubing itself is 7 ppg.

The example is sub-divided into two main parts:

- Standard analysis: A static analysis of the riser subjected to current loads and vessel offset, followed by a dynamic analysis including wave loading and first-order vessel motions.
- Specialised case: A series of analysis stages culminating in the fatigue assessment of the inner riser subject to current-induced vortex induced vibration (VIV).

MODELLING FEATURES

The main purpose of the example is to demonstrate the program options for analysing pipe-in-pipe systems. There are two main aspects to pipe-in-pipe modelling in Flexcom, both of which are invoked here. The first involves identifying which sections of the model are contained within which other sections (i.e. Pipe-in-Pipe Sections). So in the present example you specify that the tubing above the mudline is contained within the inner riser, and that the inner riser is everywhere contained within the outer riser. The second modelling aspect involves specifying pipe-in-pipe connections between nodes on the respective sections. Linear and non-linear connections are available, and these can be used in combination, as they are here. Linear connections are used at regular intervals to model spacers or centralisers. Elsewhere non-linear connections are specified: these have a very low stiffness when the sections are in their respective undeformed positions, but the stiffness increases exponentially as the gap between sections approaches zero, to provide a simple contact model.

The example also showcases a modelling approach for estimating fatigue damage on inner pipes induced by VIV. This is a specialised methodology in which the VIV motions of the outer pipe, as estimated by a modal analysis and Shear7, are applied to the inner pipe using fabricated sinusoidal motion time histories, and then dynamic fatigue is computed in the usual manner via rainflow cycle counting. Refer to VIV Induced Fatigue of Pipe-in-Pipe Systems for further details.

MODEL SIMPLIFICATIONS

This example system is intended to represent a production riser operating on a spar. However since the main purpose is to demonstrate pipe-in-pipe modelling, a number of aspects of spar riser analysis (particularly those illustrated in Example A02 - Spar Production Riser) are only very simply modelled here. In a realistic spar production riser model most or all of these simplifications would be addressed.

Standard Analysis

- Although the tubing is modelled to 8000 ft below the mudline, soil/tubing interaction is not explicitly modelled using P-y curves. Instead a simple lateral (DOF 2) restraint is applied at every 100 ft between packer and mudline.
- Likewise interaction (contact) between the outer riser and guides in the spar hull is not explicitly modelled - again a simple lateral restraint is applied at 100ft intervals within the hull.
- A keel joint is not included in the model.
- The top tension is modelled as a simple vertical point load.

Specialised VIV Simulation

- As current-induced VIV is the focus, the tubing below the mudline is omitted.

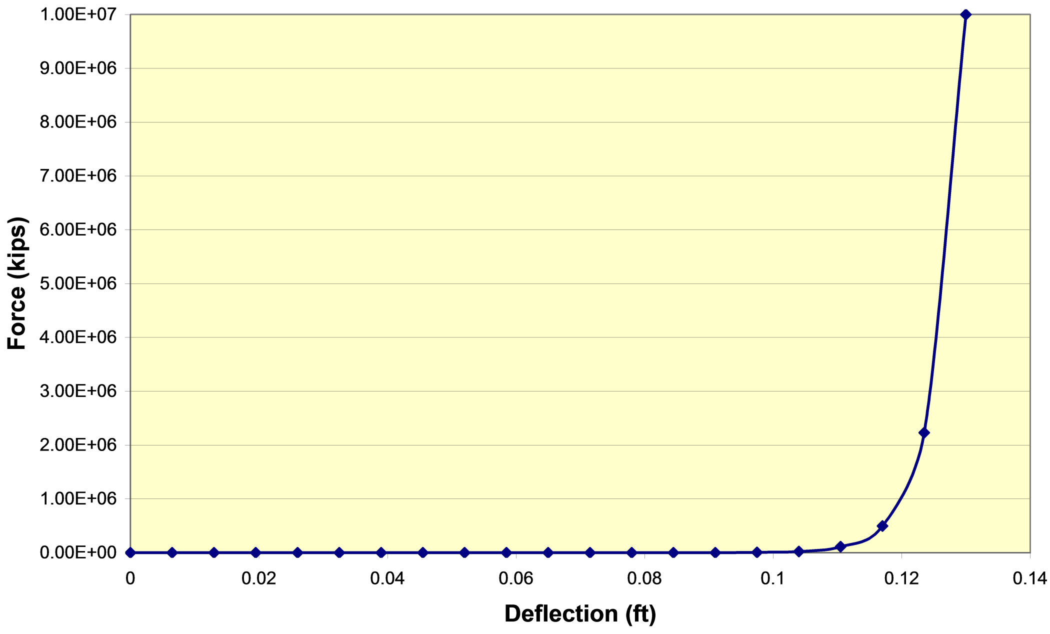
- Contact between the outer riser and guides in the spar hull is simulated using linear node springs, rather than attempting to use contact surfaces. As the modal analysis is inherently linear by nature, any non-linear effects caused by intermittent contact cannot be captured in the VIV assessment. In order to ensure conservative results, node springs of relatively high stiffness are used, as this will cause elevated curvatures to appear in the region of the lowest riser guide on the spar hull (i.e. at the intersection between restrained and unrestrained sections). Defining the rigidity/flexibility of these linearised connections is a matter of engineering judgment. If you re-run the simulation using different input values, you will be able to examine the sensitivity of fatigue damage estimates to this parameter.

- Only one current profile is considered in Shear7, whereas in reality a range of current profiles would be simulated to cover the range of loading conditions likely to be experienced by the riser system.

Model Summary

In the simplified riser model used here, each of the three concentric sections is modelled using elements 25’ in length. This gives a total of 813 elements – in a more realistic model very many more elements would likely be used. In defining the nodal co-ordinates for the production tubing, the lowermost node (Node 1) is given an initial vertical co-ordinate 4’ above the packer level at -8000’, and is then moved to the packer by the application of a boundary condition. The 4’ is the amount by which the 12000’ of tubing will elongate under its own weight, and the definition of co-ordinates in this way ensures the tubing is slack at the packer. Likewise the lowermost node on the inner casing (Node 1001) has an initial vertical co-ordinate of 0.5’ above the mudline, and is again moved to the mudline via a boundary condition. This pre-stretch is to ensure acceptable base tension in the inner riser over its full working life.

The three sections are connected together by making the nodes at the top of each section equivalent. The nodes connected in this way are Node 485 on the tubing, Node 1165 on the inner riser, and Node 2165 on the outer. There is then a further single node above these for the application of the top tension of 1120 kips. Of course the lateral (horizontal) motions of the three sections are coupled by the pipe-in-pipe connections, which are applied at all nodes between the mudline and the three equivalent nodes at the Spar deck. As previously mentioned, these are of two types, linear and non-linear, which alternate over the full length from mudline to deck. The linear connections have a stiffness of 100,000 kips/ft. The non-linear are characterised by the force-deflection relationship shown below.



Nonlinear Connection Characteristic

In this specification, the force increases exponentially as the deflection approaches what is the same initial separation for both sets of sections, approximately 0.13 ft. The above plot is not necessarily intended to represent a physically reasonable specification; it is instead intended to show how a very simple contact model could be approximated. Note also that all the deflection values are positive – symmetry is assumed.

The model in the VIV portion of the example is similar to that discussed above, with two minor deviations:

- The tubing below the mudline is omitted, so the lowermost end is represented by Node 321 rather than Node 1.
- Node 321 is assigned an initial vertical co-ordinate of 1.333ft above its mean position on the mudline, which is subsequently adjusted via a boundary condition. This length is consistent with the approach discussed above, adjusted to account for the new length of tubing which is now 4000' rather than 12000'.

Analyses

STANDARD ANALYSIS

- **Initial Static Analysis**: The base nodes, Nodes 1, 1001 and 2001, are fixed in the 3 translational degrees of freedom. Nodes on the inner tubing at 100’ intervals between the packer and the mudline are also restrained, in DOF 2 only. Vessel BCs are specified at the uppermost node, Node 3001, in DOFs 2 and 6. Vessel BCs in DOF 2 only are also specified at nodes at 100’ intervals below Node 3001, down to the spar keel at 400 ft below MWL (500 ft below deck). The initial position of the vessel reference point (at the MWL) and the RAOs are also specified.

- **Offset and Current Analysis**: The only change here is that (i) a vessel surge offset of 120 ft (3% of water depth) is applied, and (ii) a piecewise-linear current is specified. The current velocity is 3.5 ft/s for the first 1000 ft below the MWL, decreasing then to 1 ft/s over the next 640 ft, and remaining constant thereafter. All of the BCs are unchanged and carry through automatically from the initial static analysis.

- **Time Domain Dynamic Analysis**: A wave of amplitude 20 ft and period 13s is specified. The BCs remain unchanged and are carried through automatically from preceding analyses. Since vessel BCs and RAO data have previously been input, dynamic motions are automatically applied with the onset of wave loading.

- **Frequency Domain Dynamic Analysis**: A wave of amplitude 20 ft and period 13s is specified. Similar to the time domain, no specification of boundary conditions is required in the frequency domain dynamic analysis. All of the BC data carries through from the current analysis.

SPECIALISED VIV SIMULATION

- **Initial Static Analysis**: Similar to above, with the exception that any tubing below the mudline is omitted, so the base nodes, Nodes 321, 1001 and 2001, are fixed in the 3 translational degrees of freedom. Again vessel BCs are specified at the uppermost node, Node 3001. Additionally, node springs rather than vessel BCs are used to restrain the outer riser laterally in DOF 2 at nodes at 100’ intervals down to the spar keel. No RAOs are necessary as wave loading is not considered.

- **Modal Analysis**: A modal analysis of the full system, including outer riser, inner riser and tubing is performed. Naturally the modal solution is dominated by the outer riser, which is the largest, heaviest and stiffest component of the system. 100 eigenpairs are requested in the modal solution, with 25 modes being requested for transfer to Shear7. The inner riser is sub-divided into 164 segments of equal length, in order to facilitate a more straightforward extraction of data later from Shear7 (i.e. no interpolation will be required as the Shear7 element mesh corresponds to the Flexcom element mesh).

- **Shear7 Analysis**: A VIV fatigue assessment is performed in Shear7, using the modal solution as input. Other important parameters such as the current distribution with depth, and the S-N curve, are also supplied to Shear7.

- **Modal Postprocessing**: Firstly, a series of parameters are defined to store data variables produced by Shear7. Although these parameters are extracted manually, it is hoped to automate this process in a future edition of Flexcom. Secondly, a series of displacement and curvature plots are requested from Modes, corresponding to the bending modes of interest. Key variables from Shear7 include:

    - Numbers of the excited modes (taken from section "2.2.1 Results of the Mode Interaction Analysis")

    - Modal time share probabilities (taken from section "2.2.1 Results of the Mode Interaction Analysis")

    - Modal frequencies (taken from section "9. Modal damping ratio "zeta", modal mass, and modal frequency for the mainly excited modes")

    - Modal amplitudes (taken from "11. Modal Displacement Amplitude")

- **Static Offset Analyses**: Based on the information retrieved in the previous stage, a series of static offset analyses (5 in total here) are used to deform the riser system into shapes which corresponds to the bending modes of interest. A number of manual steps are required (again it is hoped to automate this process in a future edition of Flexcom) as follows:

    - View each displacement plot in turn using the Plotting module

    - Export the plot data to CSV format using the CSV button

    - Open the CSV file in Excel, scale the displacement data by the relevant modal amplitude predicted by Shear7, then copy the displacement column

- Paste the displacement data into the relevant file under the *BOUNDARY keyword (it is helpful to first paste the data into a standard text editor, before re-copying and pasting into the keyword file)

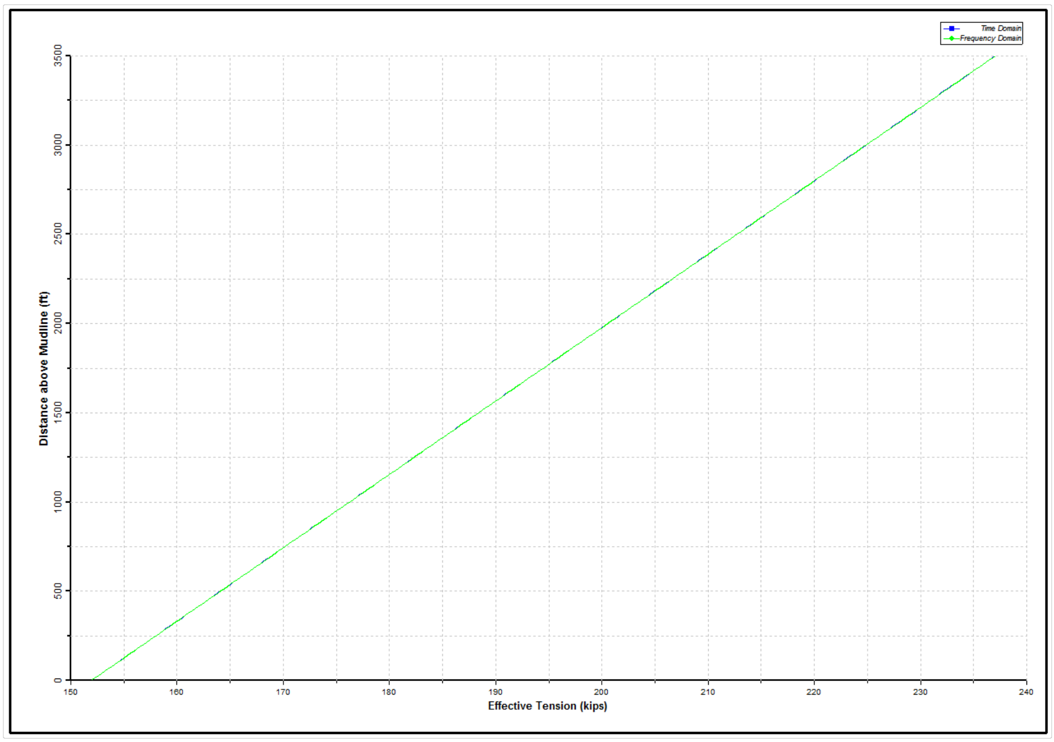
- **Fatigue Analysis**: Perform a fatigue analysis using the list of static offset analysis as input 'seastates' via the *SEASTATE FILES keyword, including the relevant modal frequencies and time share probabilities predicted by Shear7.

Results

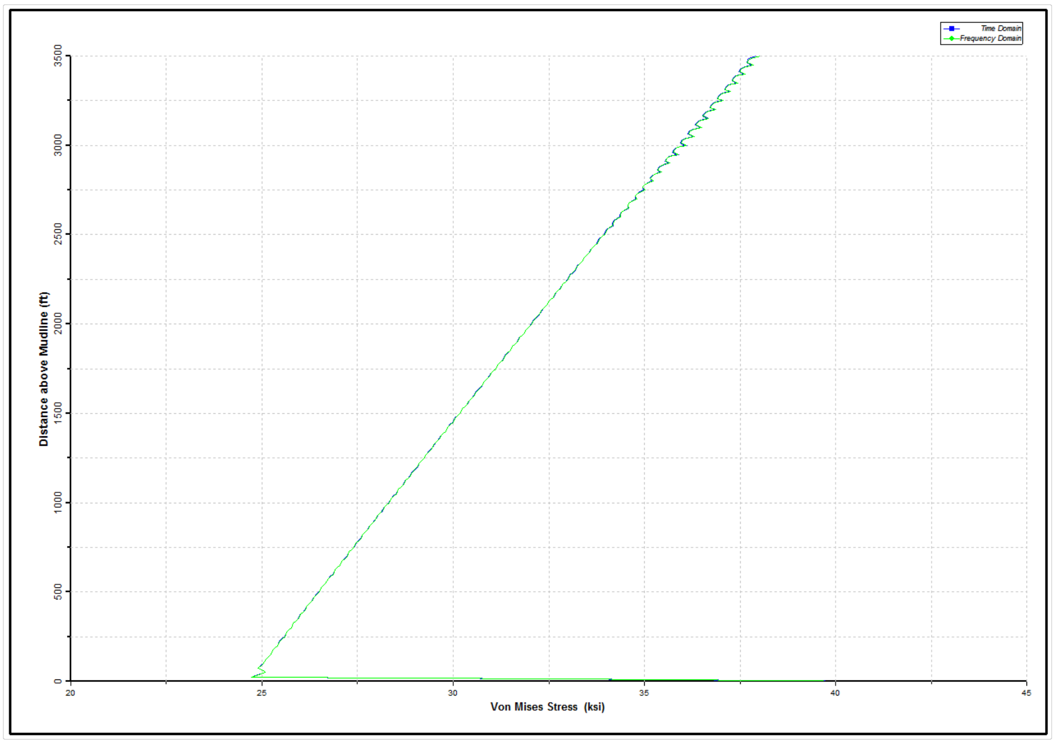
STANDARD ANALYSIS

Dynamic results from both time domain and frequency domain analyses are presented below in terms of envelopes of effective tension and von Mises stress for the three sections. Results in all cases are plotted for elements up to 100 ft below the Spar keel; above this point unrealistically high stresses could be expected due to the crude nature of the Spar/riser interaction model used here. In the case of the tubing, results are presented for elements above the mudline only.

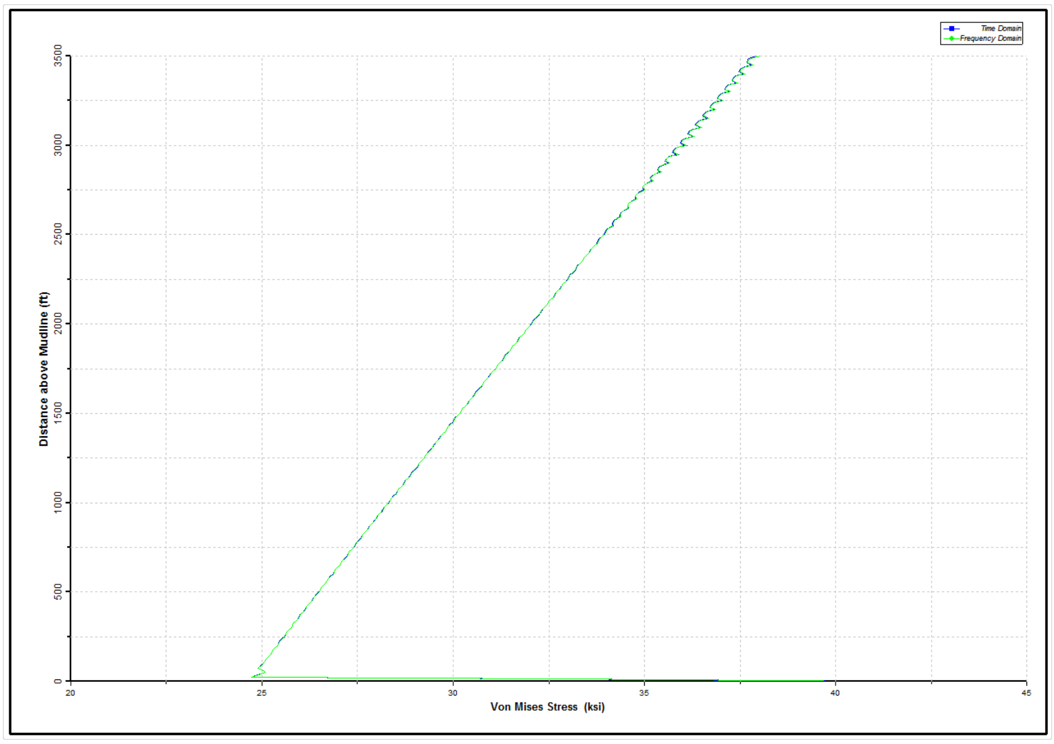
The Dynamic Tensions in Production Tubing figure and Dynamic Stresses in Production Tubing figure below show the results for the production tubing. Assuming the steel in the riser is Grade 80, the stresses are everywhere below allowable. Overall the dynamic variation is small, as might be expected with 8,000 ft of tubing below the mudline included in the model. Next, the corresponding plots for the inner riser are presented in the Dynamic Tensions in Inner Riser figure and Dynamic Stresses in Inner Riser figure below. The somewhat “jagged” plot is due to the variation in bending moment caused by the two different and alternating pipe-in-pipe connection types. Finally, the outer riser results are shown in the Dynamic Tensions in Outer Riser figure and the Dynamic Stresses in Outer Riser figure below. The agreement of results obtained from the time domain and frequency domain analyses is shown to be excellent.



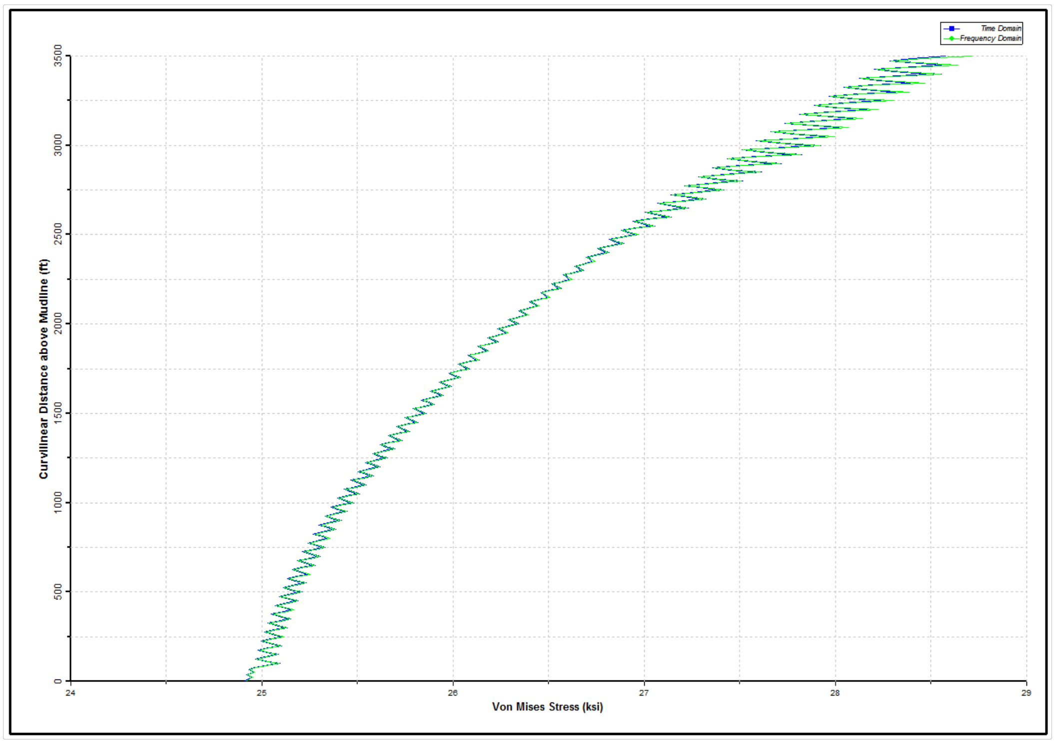
Dynamic Tensions in Production Tubing



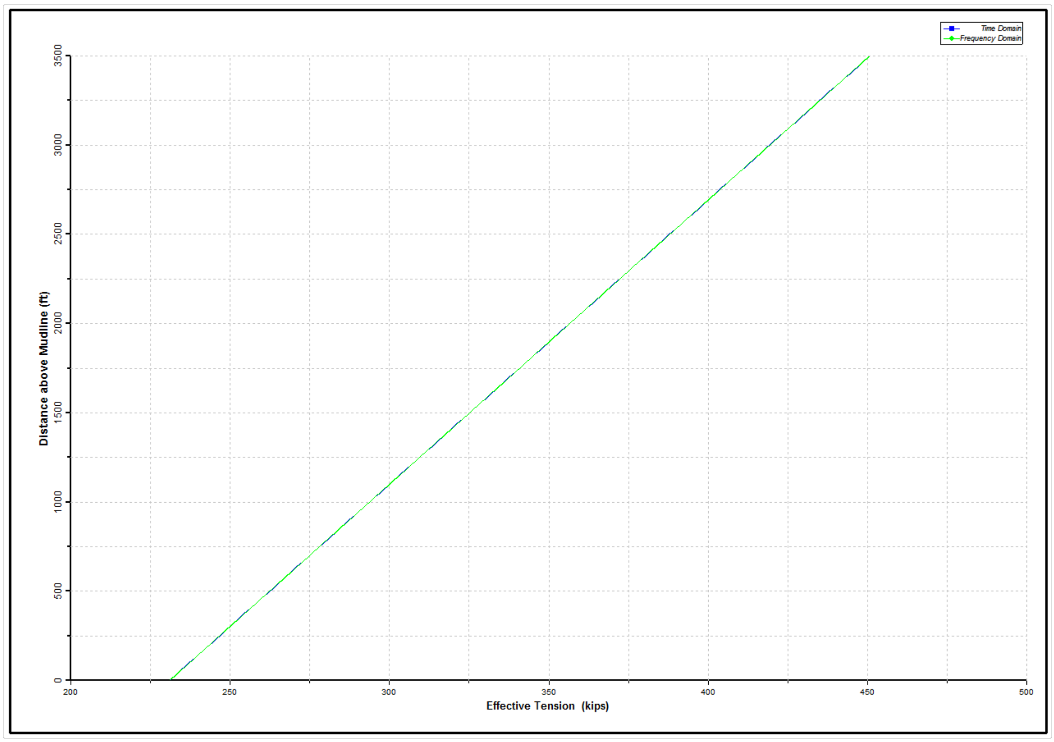
Dynamic Stresses in Production Tubing



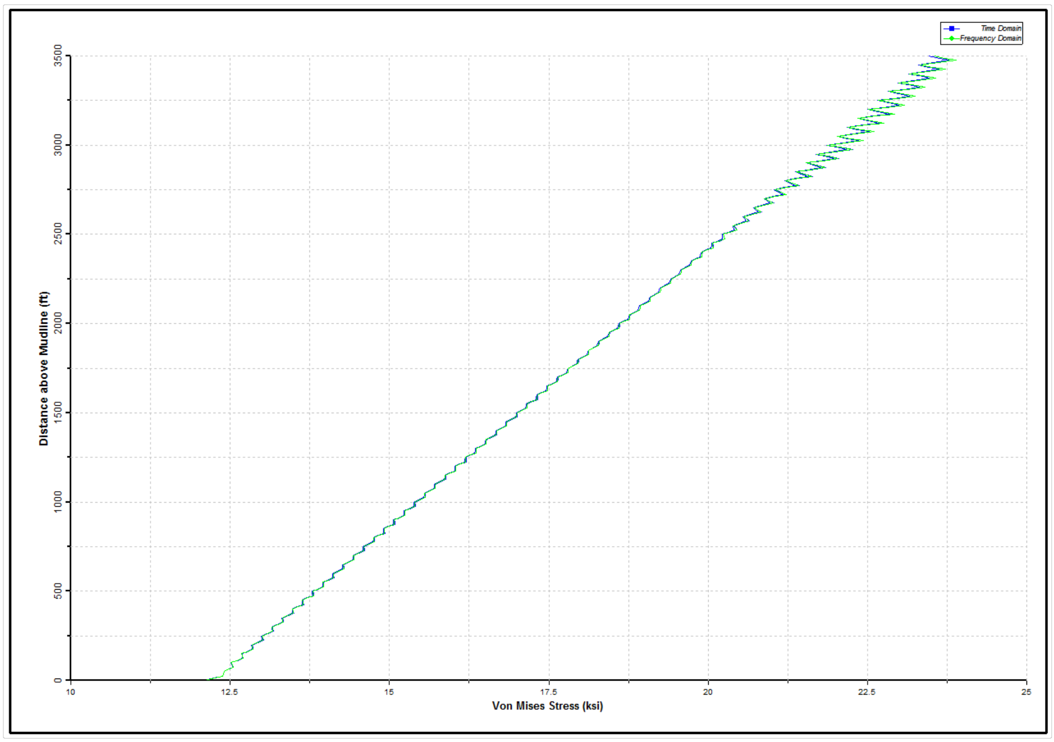
Dynamic Tensions in Inner Riser



Dynamic Stresses in Inner Riser



Dynamic Tensions in Outer Riser



Dynamic Stresses in Outer Riser

SPECIALISED VIV SIMULATION

Modal Analysis

Results from the modal analysis are summarised below. It is clear from the table that the bending modes occur in identical pairs at each modal frequency, which is expected behaviour for a top-tensioned riser. This data is subsequently passed to Shear7 via the MDS file.

Modal Solution Summary

```
                                        *** Summary of Results ***
                                        --------------------------

 ----------------------------------------------------------------------------------

                                                                           Max
 Eigenpair          Eigenvalue          Period      Type               Curvatu
                                          (s)                           (1/ft)
 ----------------------------------------------------------------------------------
      1               0.0927            20.6340     Bending            0.3894E-
      2               0.0932            20.5799     Unknown            0.0000E+
```

3	0.3714	10.3103	Bending	0.7561E-
4	0.3733	10.2833	Unknown	0.0000E+
5	0.8361	6.8715	Bending	0.1123E-
6	0.8405	6.8534	Unknown	0.0000E+
7	1.4874	5.1519	Bending	0.1489E-
8	1.4952	5.1384	Unknown	0.0000E+
9	2.3258	4.1200	Bending	0.1854E-
10	2.3381	4.1091	Unknown	0.0000E+
11	3.3523	3.4317	Bending	0.2220E-
12	3.3700	3.4227	Unknown	0.0000E+
13	4.5677	2.9399	Bending	0.2584E-
14	4.5920	2.9321	Unknown	0.0000E+
15	5.9734	2.5708	Bending	0.2953E-
16	6.0052	2.5640	Unknown	0.0000E+
17	7.5706	2.2836	Bending	0.3312E-
18	7.6111	2.2775	Unknown	0.0000E+
19	9.3609	2.0536	Bending	0.3705E-
20	9.4112	2.0481	Unknown	0.0000E+
21	11.3461	1.8653	Bending	0.4057E-
22	11.4073	1.8603	Unknown	0.0000E+
23	13.5280	1.7083	Bending	0.4420E-
24	13.6013	1.7037	Unknown	0.0000E+
25	15.4122	1.6005	Torsional	0.0000E+

Shear7 Analysis

From the perspective of a VIV-induced fatigue assessment of the inner riser, the results of interest from Shear include:

- Numbers of the excited modes (taken from section "2.2.1 Results of the Mode Interaction Analysis")
- Modal time share probabilities (taken from section "2.2.1 Results of the Mode Interaction Analysis")
- Modal frequencies (taken from section "9. Modal damping ratio "zeta", modal mass, and modal frequency for the mainly excited modes")
- Modal amplitudes (taken from "11. Modal Displacement Amplitude")

Relevant extracts from the Shear7 output file are reproduced here in order to help you to familiarise yourself with the various data outputs (assuming you are not an experienced user of Shear7).

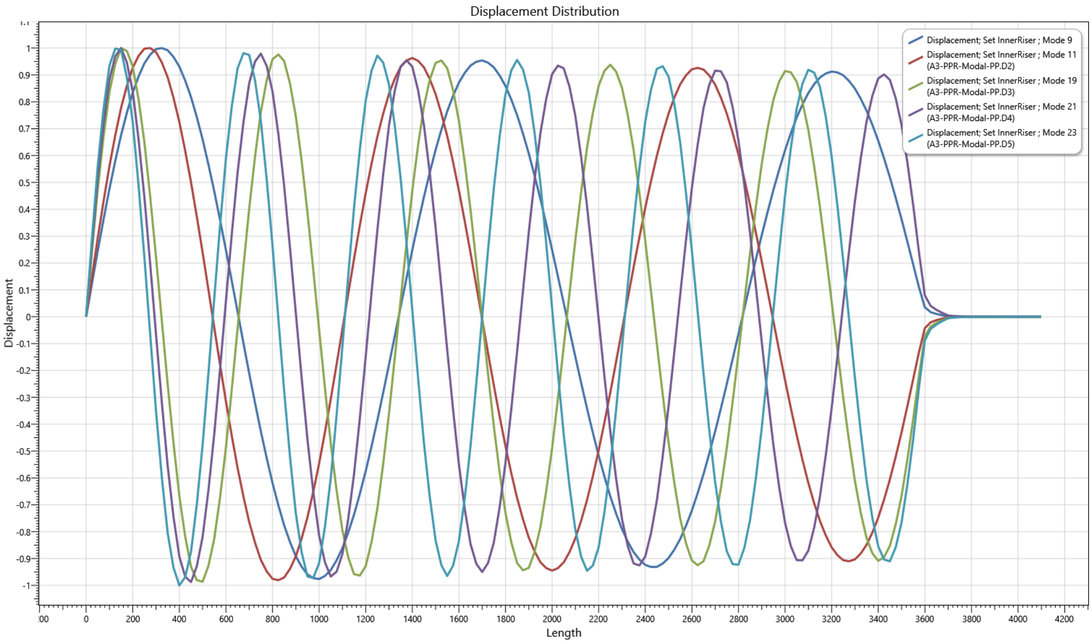
Shear7 Output File Extracts

```
2.2.1 Results of the Mode Interaction Analysis:
```

```
Based on the non-zero power-in lengths and the power cutoff value of: 0.
 the number of modes above cutoff is:           5
 These modes are: Time Share                       Excitation Dominant Moc
                Probabilities:                        Zone #     Amplitude:
         5           0.6776                             1       0.1000E+01
         6           0.1154                             1       0.9805E+00
        10           0.0529                             1       0.5515E+00
        11           0.0731                             1       0.5394E+00
        12           0.0811                             1       0.5237E+00
                    --------                                     --------
 Cumulative sum:     1.0000     Primary zone amplitude limit:   0.3000
 Lowest And Highest Excited Mode Number
 Nmmin=  5     Nmmax= 12

9. Modal damping ratio "zeta", modal mass,
   and modal frequency for the mainly excited modes.

  mode no.  damping ratio  modal mass(slug) frequency (Hz)
  --------------------------------------------------------
     5       0.15389          2568.612        0.24273
     6       0.10722          2565.540        0.29140
    10       0.05928          2557.277        0.48695
    11       0.05150          2535.415        0.53610
    12       0.04567          2588.692        0.58537


11. Modal Displacement Amplitude

  Mode No.   Amplitude (ft or m)
  ---------------------------------
     5             0.38520
     6             0.38644
    10             0.08868
    11             0.09435
    12             0.08762
```

Modal Postprocessing

The solution database produced by Modes contains information on all modes (whether bending, axial, torsional or mixed modes), while the Shear7 output relates specifically to pure bending modes. In order to extract the relevant mode shapes from Modes, the relevant mode number are obtained from the Modal Solution Summary table, using the Shear7 mode number as an index in the table. For example, the first VIV mode identified by Shear7 is Mode No.5, and this corresponds to Mode No.9 in the output from Modes. All 5 mode shapes of interest overlaid in a single plot are shown below. Although not an issue in this case, it is advisable to examine the various mode shapes to verify that they represent pure bending modes (viewing two separate planes side-by-side in the Model View is quite helpful in this respect) and reduce the possibility of mixed modes being included.



Mode Shapes

Fatigue Analysis

Results from the fatigue analysis are summarised below. The fatigue life predicted by the Rainflow Cycle Counting method is just over 8 years, so fatigue is clearly an issue for the riser system as it currently stands. The engineering design team would probably look into VIV suppression mechanisms such as strakes or fairings. It should be noted that this is a highly simplified example which is provided for illustrative purposes only. Only one current profile is considered in Shear7, whereas in reality a range of current profiles would be simulated to cover the range of loading conditions likely to be experienced by the riser system. Your attention is also drawn to the other Model Simplifications listed earlier.

Fatigue Results Summary

```
              Results Summary
              ===============


Damage Calculated Directly from Time History

Maximum predicted damage  :        .14738E+00
Corresponding fatigue life:      6.79 years
Occurring at element      :  1145
Stress point              :      1


Damage Calculated from Stress Spectrum

Maximum predicted damage  :        .21535E+00
Corresponding fatigue life:      4.64 years
Occurring at element      :  1145
Stress point              :      1


Damage Calculated from Rainflow Counting

Maximum predicted damage  :       .12238E+00
Corresponding fatigue life:      8.17 years
Occurring at element      :  1145
Stress point              :      1
```

1.10.1.4 A04 - TTR Wake Interference

This example considers a TLP riser system in which a production riser and a drilling riser operate in close proximity to each other. Wake interference effects are modelled in order to accurately determine the clearance between both risers. The example is divided into the following sections:

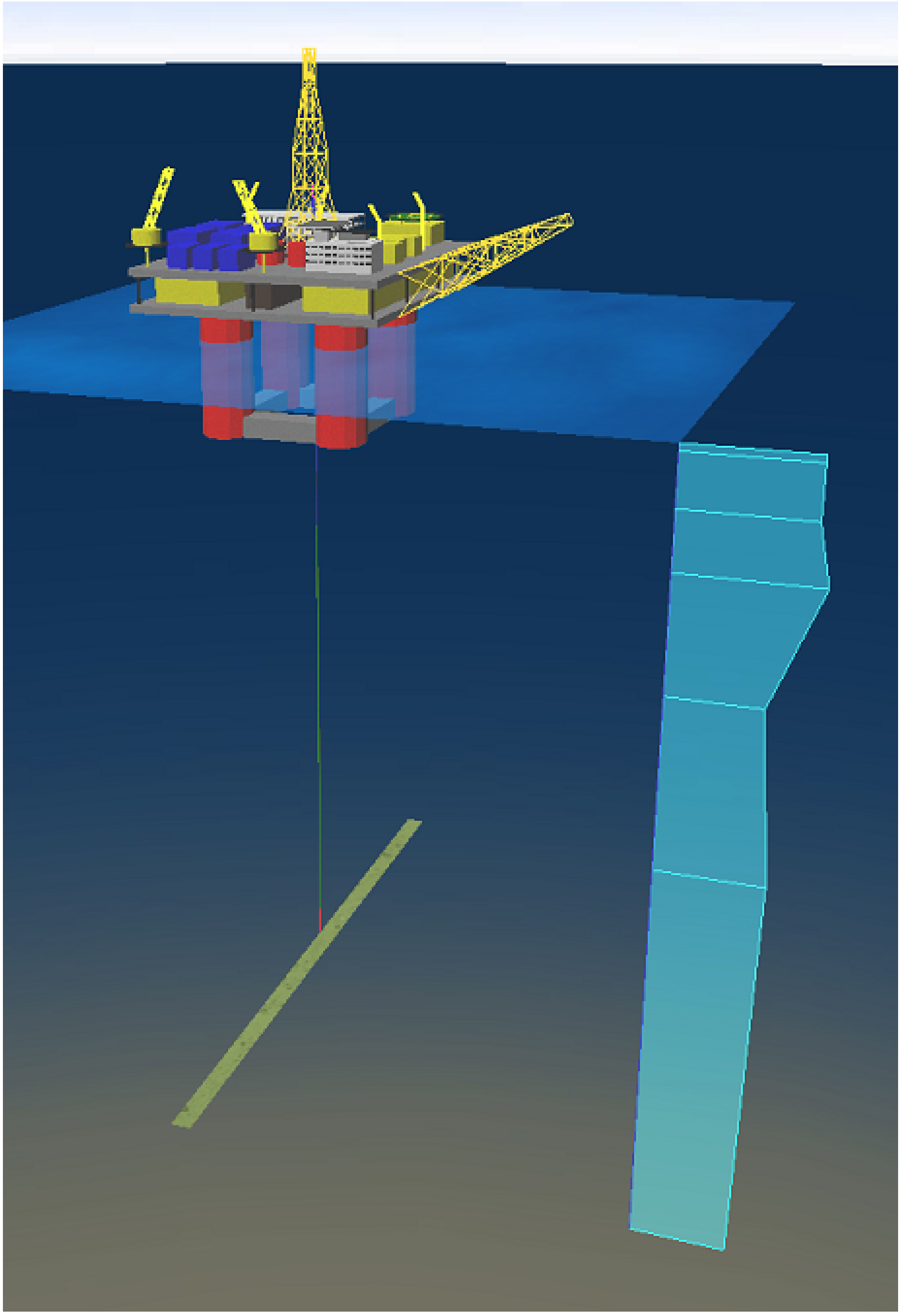
- Introduction gives an overview of the analysis.

- Model Summary describes the model in more detail.
- Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions.
- Results presents pertinent results from the various analyses performed.

Introduction

This example considers the riser system of a TLP vessel situated in 350 m of water. Two risers are included in the project, a drilling riser and a production riser. The main objective of the analysis is to accurately determine the clearance between the risers, taking wake interference effects into account. The upstream (drilling) and downstream (production) risers are defined in separate models as they are structurally independent. The drilling riser model, including a schematic of the applied current profile, is shown in the figure below.

The analysis sequence considers the drilling riser first in order to determine its static configuration subject to static current loading and VIV drag data. The VIV drag data is obtained independently, by running a Shear7 analysis. Next, the production riser is considered. An initial static analysis is performed, and this is then followed by a current static analysis in which the wake effects from the drilling riser are considered. The wake effects are modelled using Blevins, (2005) formulation. This applies a reduced current on the downstream riser, in addition to a lift force which attempts to align the risers in the direction of the applied current.



Drilling Riser and Current Profile

Model Summary

DRILLING RISER

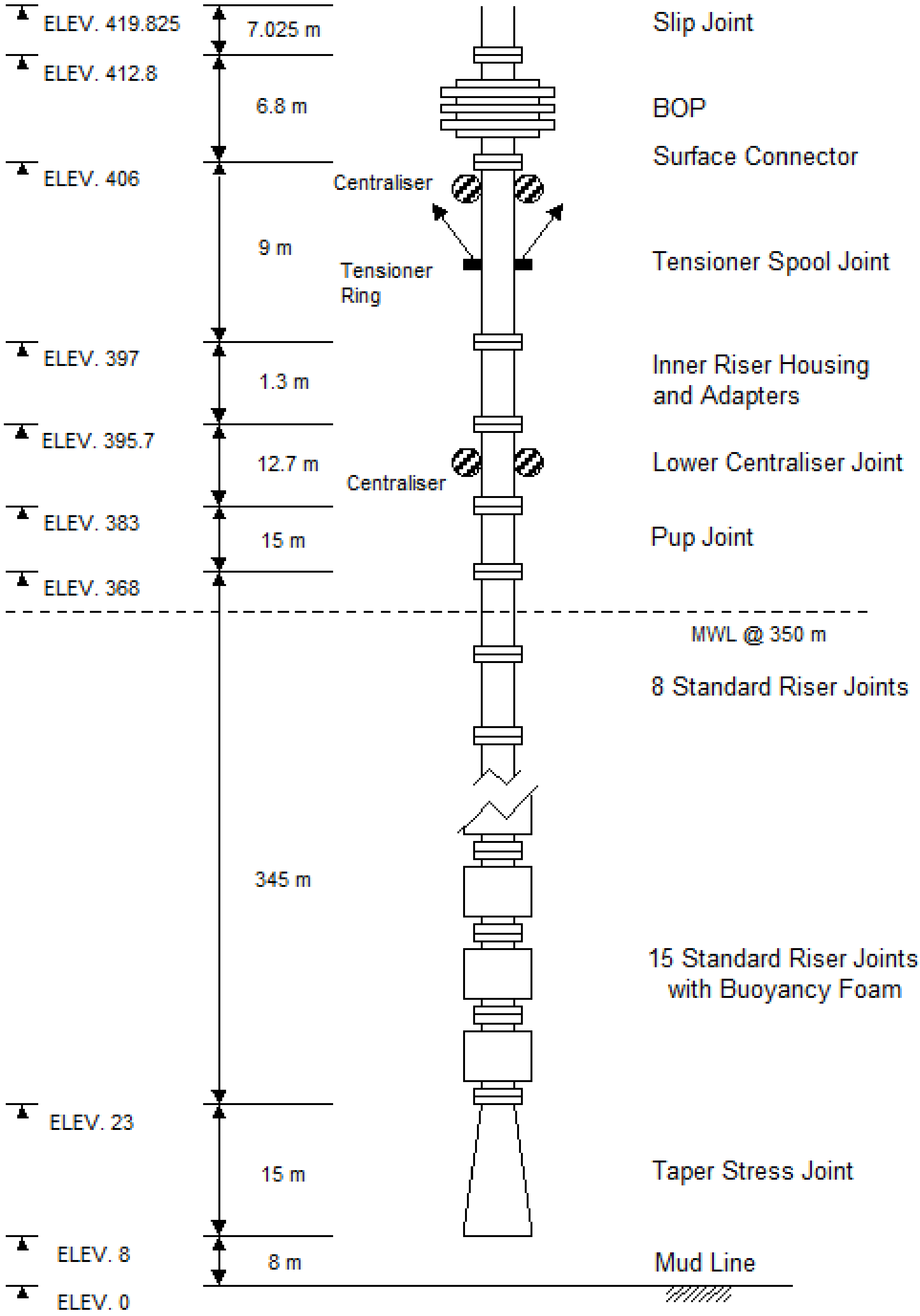
The drilling riser is fixed at the base of a tapered stress joint to a subsea wellhead via a subsea connector. Above the stress joint, the stack-up comprises 23 standard riser joints, the lower fifteen of which are fitted with buoyancy foam. Between the top of the standard riser joints and the tensioner spool joint are a pup joint, a lower centraliser joint and the inner riser housing and adapters. The riser tensioning ring and a centraliser are located on the tensioner spool joint at mid-height and at the top of the joint respectively. A BOP and slip joint are fitted above the tensioner spool joint. The Drilling Riser Stack-Up Schematic figure below shows a schematic of the drilling riser stack-up.

PRODUCTION RISER

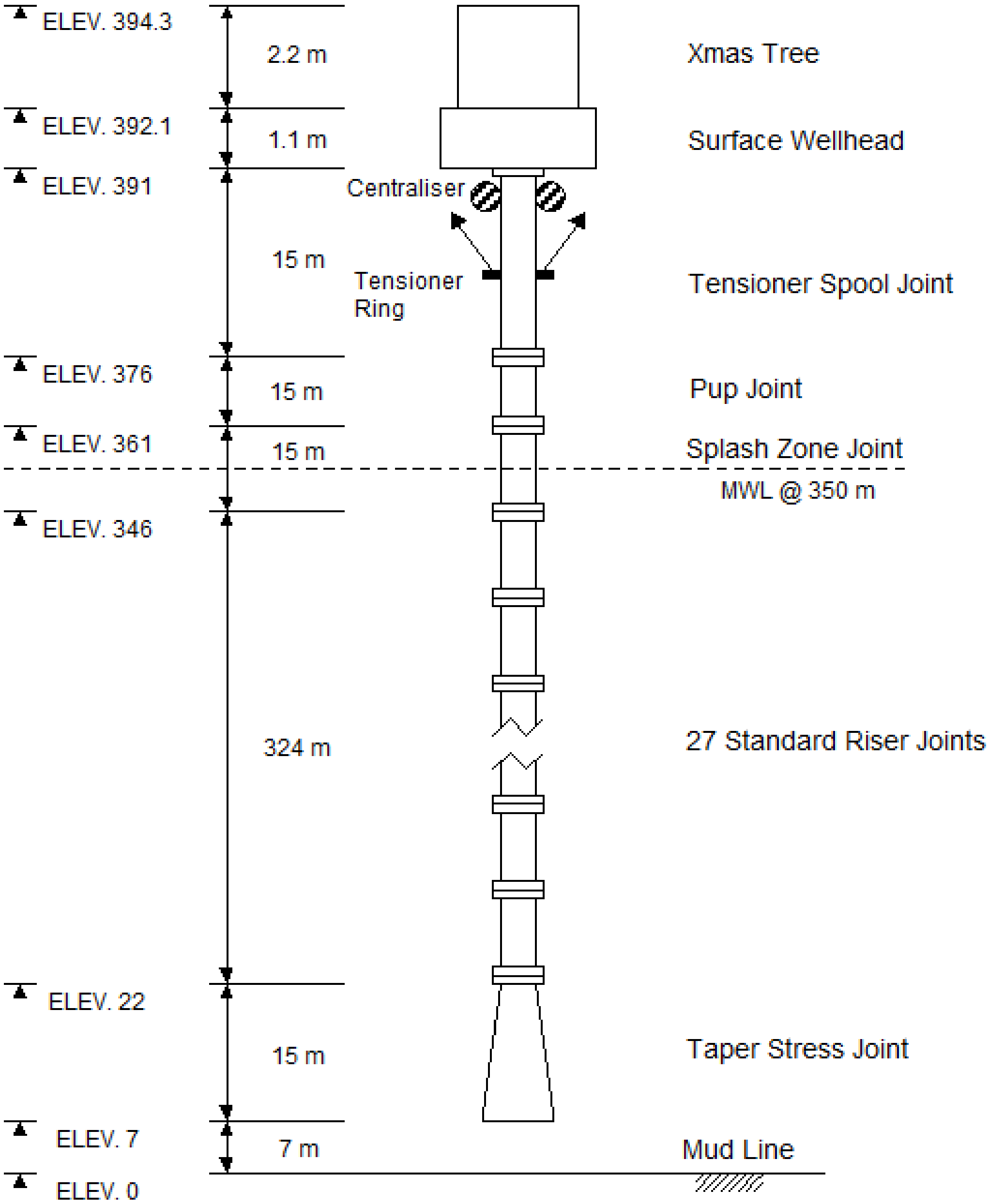
The production riser under consideration is of the dual casing type with the outer annulus containing brine and the inner annulus containing gas. A tapered stress joint is located at the riser base, above which are 27 standard riser joints. Above the standard riser joints, at the MWL, is a splash zone joint and above this are located a pup joint, the tensioner spool joint, surface wellhead and Xmas tree respectively. The riser tensioning ring is located midway along the tensioner spool joint and a centraliser is located at the top of this joint also. The Production Riser Stack-Up Schematic figure below shows a schematic of the production riser stack-up.

TLP VESSEL

The TLP vessel is approximately 75 m long by 75 m wide with a draft of 30m, and is connected to the seabed with tethers of length 300 m.



Drilling Riser Stack-Up Schematic



Production Riser Stack-Up Schematic

Analyses

DRILLING RISER STATIC ANALYSIS

The base of the riser is fixed in all degrees of freedom. The upper ends of the tensioner elements are attached to the vessel. In addition an internal fluid of density 1970kg/m^3 is contained in the riser.

DRILLING RISER STATIC CURRENT ANALYSIS

The boundary conditions remain unchanged and are carried through automatically from the initial static analysis. A piecewise-linear cross current is also applied to the drilling riser. The direction of the current is constant with depth and is flowing in the positive sense of the global Y-axis.

DRILLING RISER VIV ANALYSIS

VIV effects are considered, as a modal analysis of the drilling riser is performed using Modes, and enhanced drag coefficients are computed using Shear7. As not all users will have both programs installed, this portion of the analysis is performed in advance, and the Shear7 output file (Example16-DR-shear.plt) is simply referenced subsequently in the static current analysis of the production riser.

PRODUCTION RISER STATIC ANALYSIS

The base of the riser is fixed in all degrees of freedom. The upper ends of the tensioner elements are attached to the vessel. An internal fluid of density 1400 kg/m^3 is contained in the riser.

PRODUCTION RISER MODAL ANALYSIS

As part of a wake interference analysis which includes VIV effects, Flexcom automatically performs a modal analysis of the downstream structure, and proceeds to perform a Shear7 analysis to compute enhanced drag coefficients.

In order for this to be possible, the modal analysis file must be created in advance so that Flexcom can access it subsequently as part of the wake interference analysis.

In the modal analysis, the riser type is designated as TTR, 100 eigenpairs and 20 modes are requested, and the riser is divided into 80 equally spaced segments for the purposes of generating Shear7 input data.

PRODUCTION RISER STATIC CURRENT ANALYSIS

The boundary conditions remain unchanged and are carried through automatically from the initial static analysis. A piecewise-linear cross current is also applied to the production riser. The direction of the current is constant with depth and is aligned with the global Z-axis.

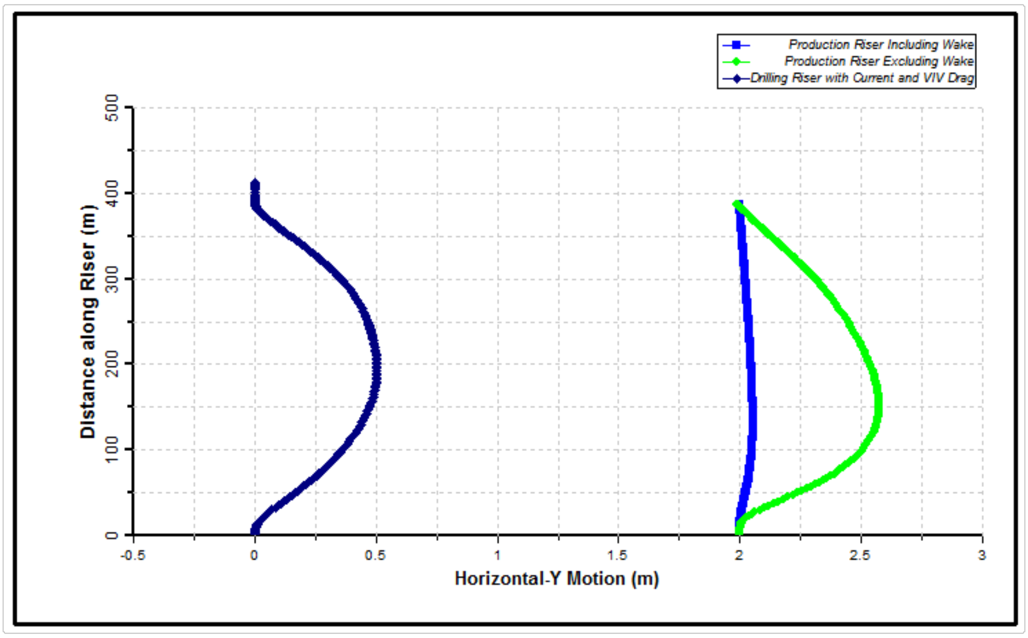
This analysis is performed twice; ignoring the presence of the drilling riser in the first instance, and then taking wake interference effects into account. It is interesting to quantify the influence of wake interference on the overall configuration of the production riser and consequently its effect on the clearance between the risers.

CLEARANCE POSTPROCESSING

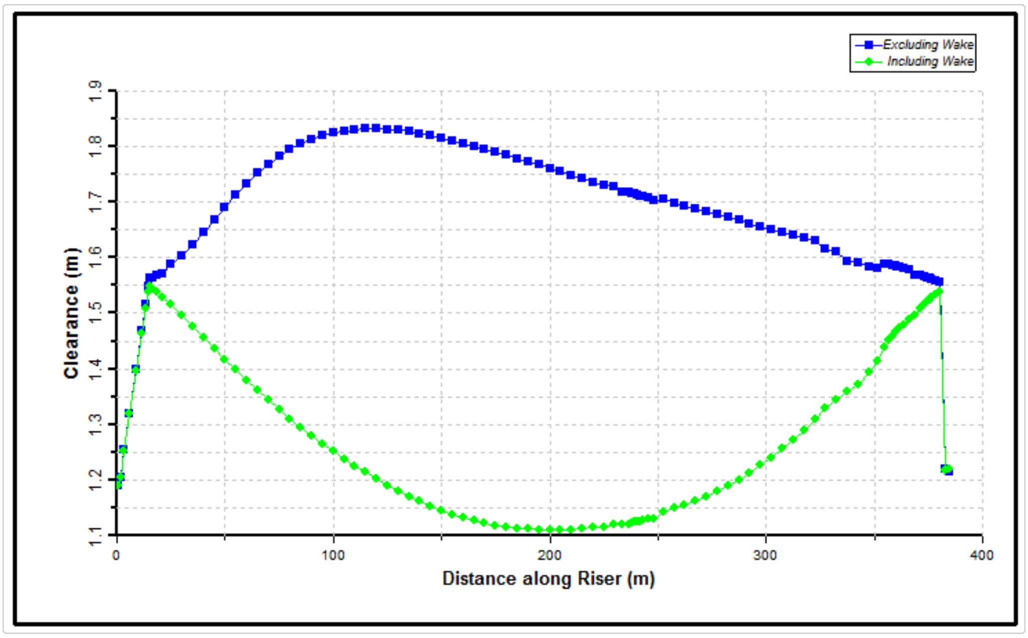
The clearance between the upstream (drilling) and downstream (production) risers is computed using Clear, the Flexcom postprocessor for clearance/interference calculations. Separate runs are performed for the cases including and excluding wake interference effects.

Results

The first figure below presents the overall configuration of the TLP riser system, including and excluding wake interference effects. Naturally the configuration of the drilling riser is the same in both cases, but the configuration of the production riser varies considerably. The second figure presents the clearance between risers.



TLP Riser System Configuration



Production/Drilling Riser Clearance

1.10.1.5 A05 - Marine Riser with Landing String

This example considers the deployment of landing string through a marine riser. The simulation is performed in a series of separate stages, considering different lengths of landing string deployed, in conjunction with various degrees of vessel offset. The example is divided into the following sections:

- Introduction gives an overview of the riser simulation, and notes some of the more important features of Flexcom which are relevant to this type of analysis.
- Model Summary describes the model in more detail, and describes how parameters and variations are used to quickly assemble a relatively complex model, and create all the keyword files necessary to simulate the landing string deployment.
- Analyses briefly describes the various steps in the analysis sequence.
- Results presents pertinent outputs from the various analyses performed, created using summary postprocessing and collation, which is central to the assessment of the simulation matrix.

Introduction

This example considers the deployment of landing string through a marine riser. The simulation is performed in a series of separate stages, considering different lengths of landing string deployed, in conjunction with various degrees of lateral vessel offset. The following points are noteworthy:

- The model is heavily parameterised, making extensive use of Parameters and Equations. In addition to being QA friendly, the template file readily accommodates any changes which may need to be made subsequently. For example, the number of riser joints is parameterised (e.g. setting RJ1_N = 8 means that the riser stack-up contains 8 riser joints of 'type 1'. So it's very easy to alter the number of joints present in the stack-up.
- Explicit units are assigned to every base/independent parameter in the model. Not only does this provide clarity to an engineer reading the keyword data, it also ensures consistency of units when defining secondary/dependent parameters. Metric units are used throughout this model, but it is also possible to mix units subsequently if you wish (e.g. by defining a wall thickness in inches).

- Each riser joint is modelled in detail, including its bare, buoyant and flange sections. To achieve this effect, line section groups are used to model Repeating Sub-Sections. These subsections are then used to automatically assemble the riser joint stack-up with relatively little effort required from the user.

- Pipe-in-pipe is used to model the interaction between the marine riser and landing string, in terms of both contact modelling and hydrodynamic forces.

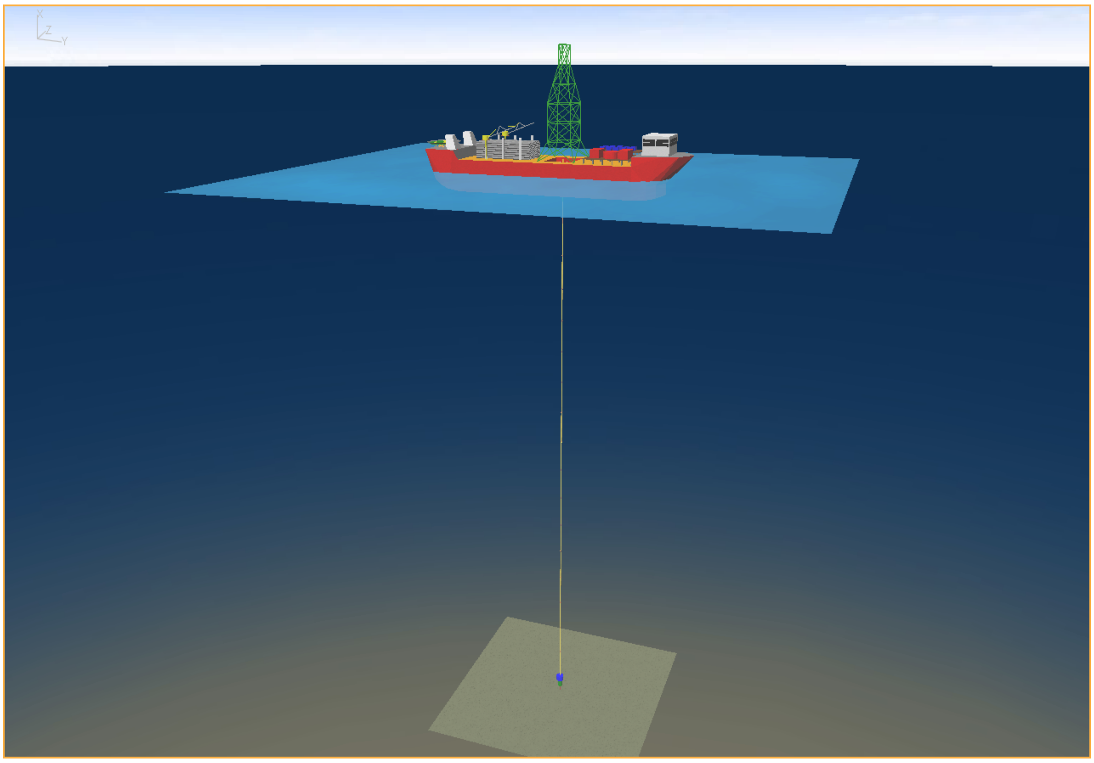
- Separate Parameters are used to define to the number of landing string joints present within the marine riser and the horizontal vessel offset. The initial model has 4 landing string joints included, corresponding to 56m of landing string. It is gradually lowered into the marine riser, by incrementally adding 4 new joints each time, until finally there are 32 joints present, corresponding to 448m of landing string. Lateral vessel offsets of up to 12m are considered, in increasing increments of 2m from the mean vessel position on station. Keyword Based Variations are used to define all the required parameters, as this facility is ideally suited to parameters which vary in fixed increments. The combinations input is used to control the names and locations of generated keyword files. In this case, a separate folder is established for every landing string deployment stage, and individual file names within each folder reflect the various vessel offset cases considered.

- Summary Postprocessing is used to examine flex joint angles, effective tension and bending moment, for every individual simulation. After each individual analysis has completed, Flexcom produces a text-based Summary Output File for visual inspection, but more importantly it also creates a Summary Database File, which is effectively a binary version of the same data, for subsequent collation. Landing string deployment length and vessel offset are also defined as key parameters which uniquely identify each individual simulation.

- The Summary Postprocessing Collation facility subsequently assembles all the output data into a single Summary Collation Spreadsheet. It also presents the data graphically in the form of 3-dimensional Summary Collation Plots.

Model Summary

The overall riser system is shown below.



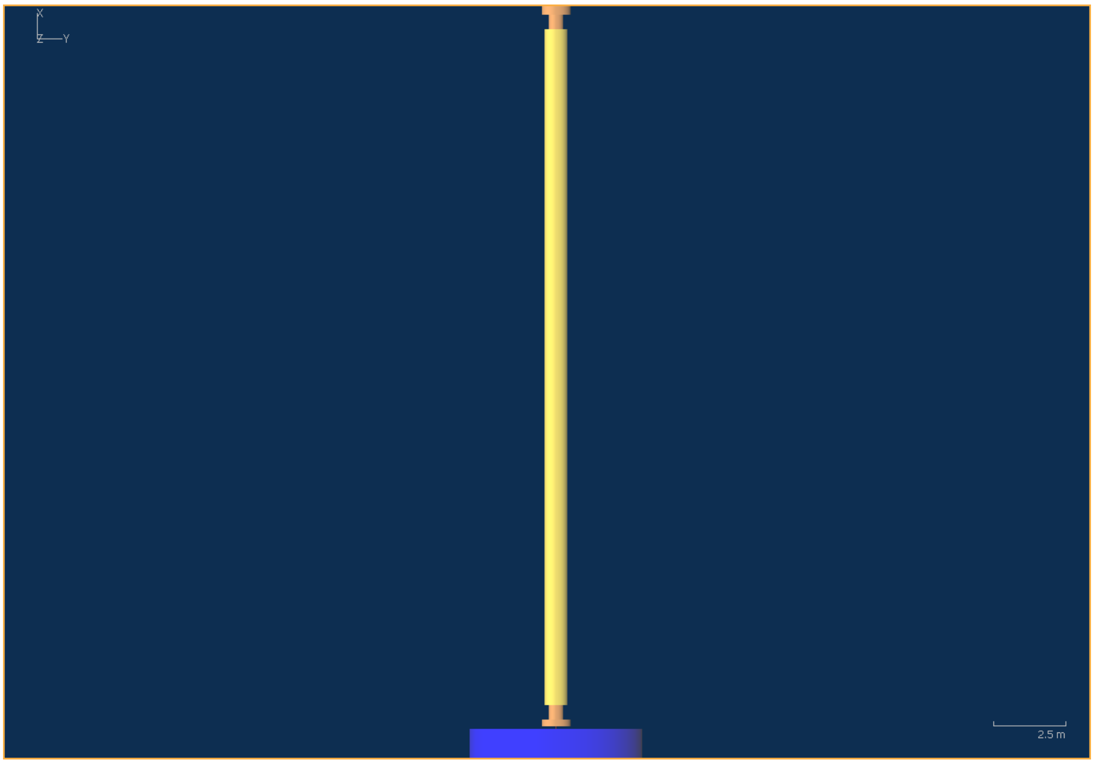
Marine Riser

The marine riser stack-up is comprised of the following sections:

- Wellhead connector
- Blow Out Preventor (BOP)
- Lower Marine Riser Package (LMRP)
- Lower flex joint
- Stack of Riser Joints (8) of type 1
- Stack of Riser Joints (6) of type 2
- Stack of Riser Joints (2) of type 3
- Stack of Pup Joints (3)
- Telescopic joint

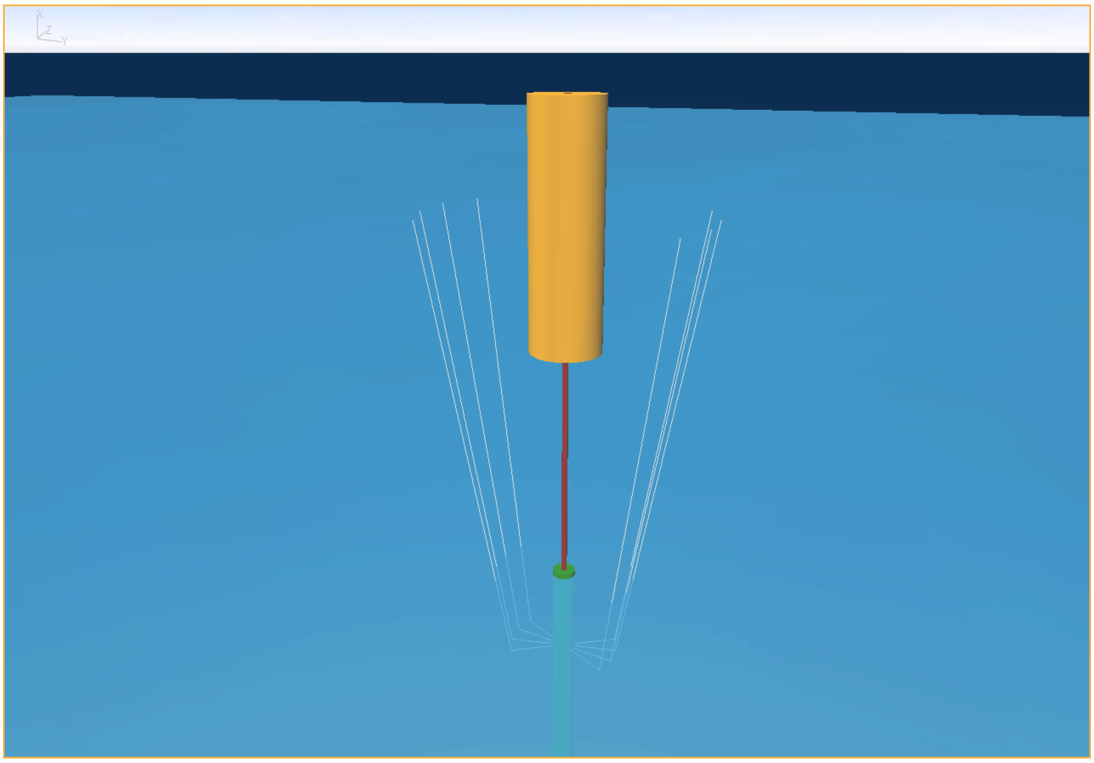
- Upper flex joint
- Diverter housing
- Tensioning system which is comprised of 8 separate tensioning lines.

A close-up of the first riser joint, just above the LMRP, is shown below. Each riser joint is modelled in detail, including its bare, buoyant and flange sections. To achieve this effect, line section groups are used to model Repeating Sub-Sections. These subsections are then used to automatically assemble the riser joint stack-up with relatively little effort required from the user.



Single Riser Joint

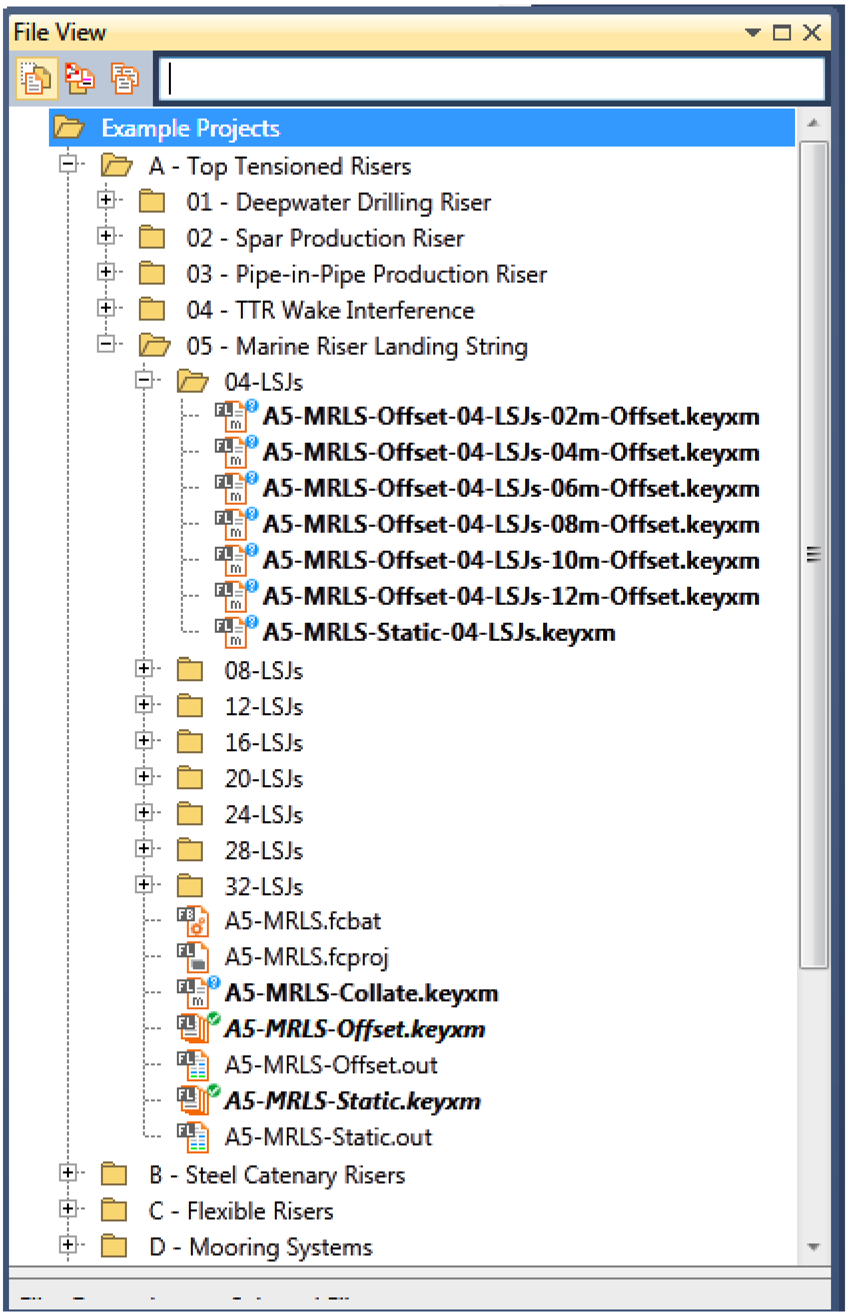
A close-up of the tensioning system is shown below.


Riser Tensioning System

Analyses

Separate Parameters are used to define to the number of landing string joints present within the marine riser and the horizontal vessel offset. The initial model has 4 landing string joints included, corresponding to 56m of landing string. It is gradually lowered into the marine riser, by incrementally adding 4 new joints each time, until finally there are 32 joints present, corresponding to 448m of landing string. Lateral vessel offsets of up to 12m are considered, in increasing increments of 2m from the mean vessel position on station. Keyword Based Variations are used to define all the required parameters, as this facility is ideally suited to parameters which vary in fixed increments. The combinations input is used to control the names and locations of generated keyword files. In this case, a separate folder is established for every landing string deployment stage, and individual file names within each folder reflect the various vessel offset cases considered.

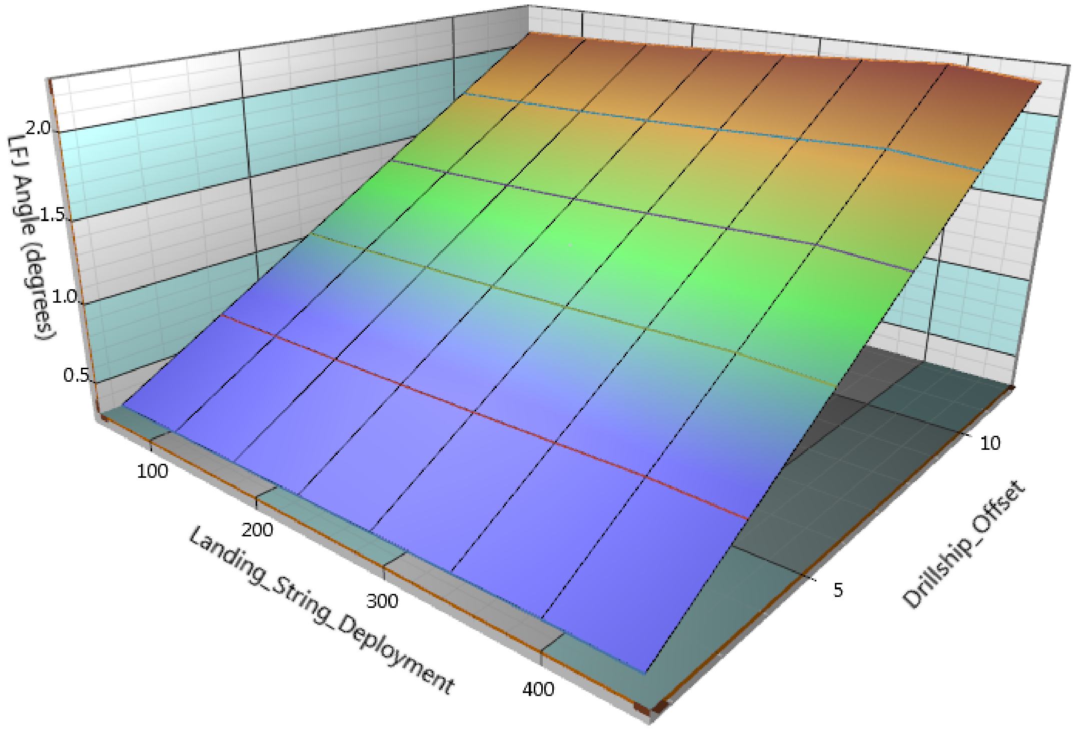
The overall file and folder structure is shown below (only the first folder is shown in expanded form).



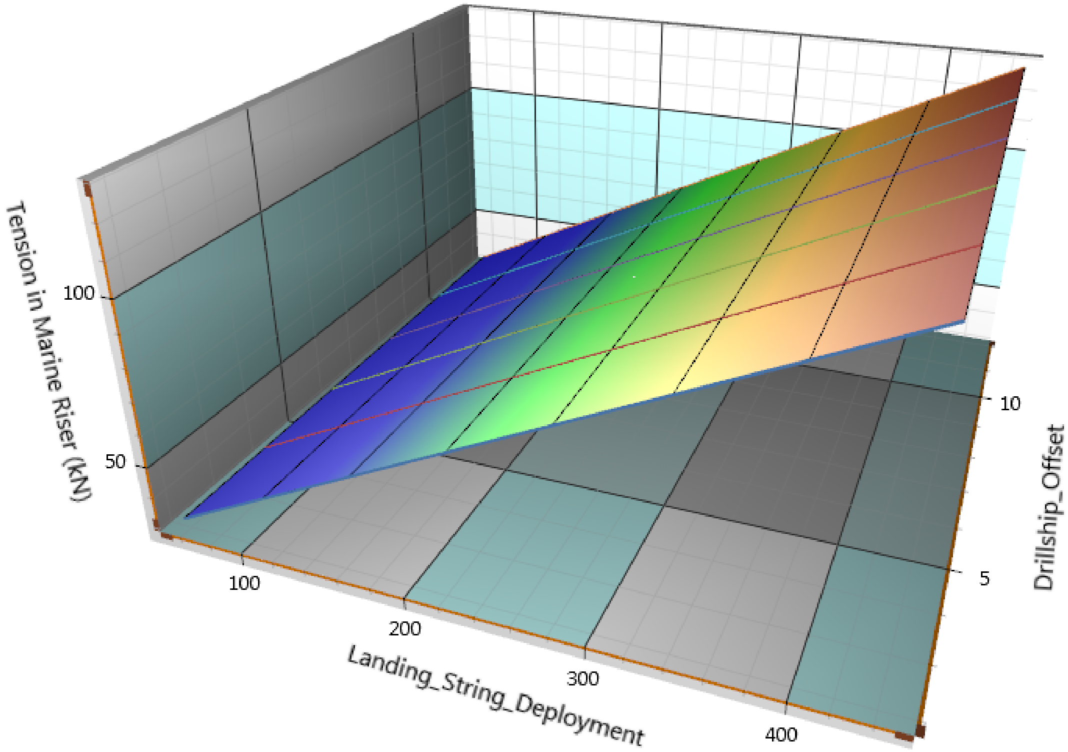
File & Folder Structure

Results

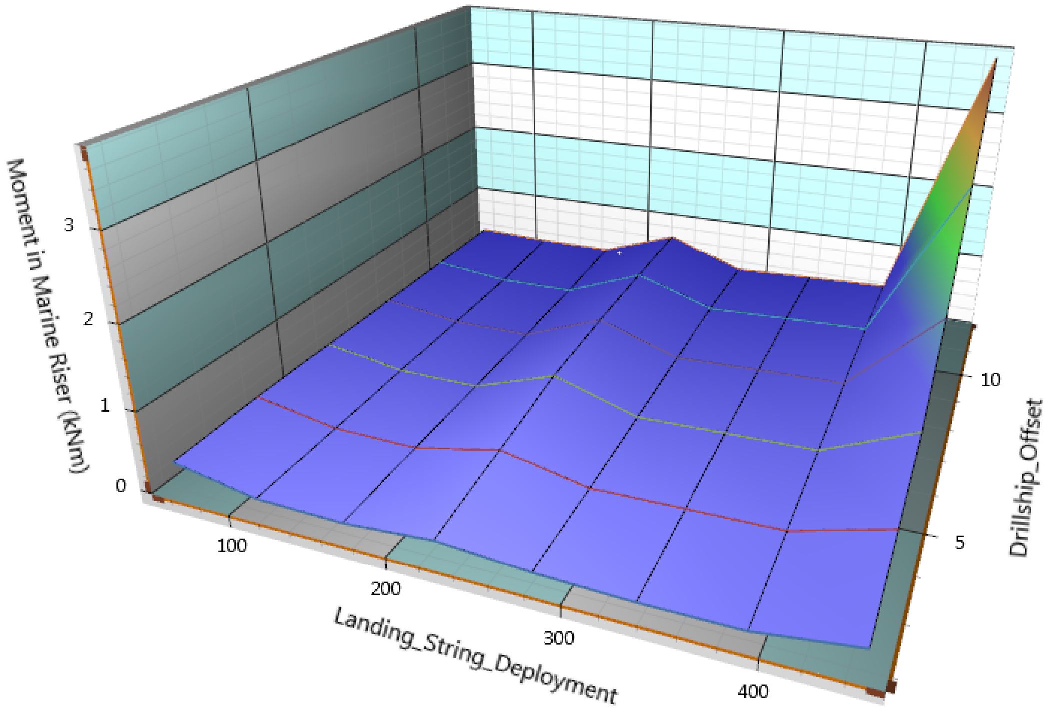
The following figures present the maximum lower flex joint angle, maximum effective tension in the landing string, and maximum bending moment in the landing string, over the course of the deployment process. The parameter of interest is shown on the vertical axis, while the model variations (i.e. deployed length of landing string and lateral vessel) are shown on the horizontal axes.



Lower Flex Joint Angle



Effective Tension in Landing String



Bending Moment in Landing String

1.10.2 B - Steel Catenary Risers

Section B contains some examples of steel catenary risers, including:

- B01 - Steel Catenary Riser
- B02 - SCR Transfer

1.10.2.1 B01 - Steel Catenary Riser

This example describes analyses of a free hanging steel catenary riser (SCR). The overall layout of the example is as follows:

- Introduction gives an overview of the example and describes the structure under consideration.
- System Modelling describes the structure from a modelling viewpoint in Flexcom.

* Analyses briefly describes the various analyses performed, discussing the environmental and loading conditions to which the riser is subjected.

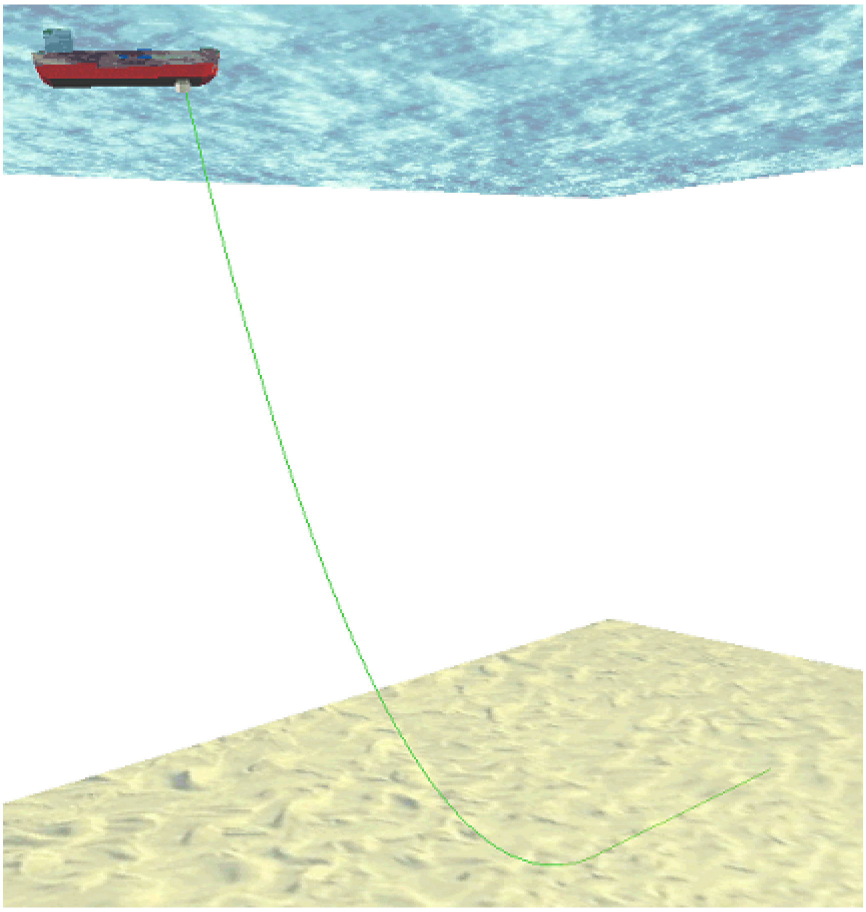
* Results presents pertinent results from the example analyses and discusses their significance.

Introduction

The system modelled in this example is an SCR hanging from a Floating Production, Storage and Offloading (FPSO) unit. It is oriented in a free hanging configuration in 1200m of water. The SCR hang-off is located midships at starboard, where a linear flex joint is connected. The SCR leaves the I-tube exit at an angle and proceeds through the flex joint towards touchdown with the seabed.

System Modelling

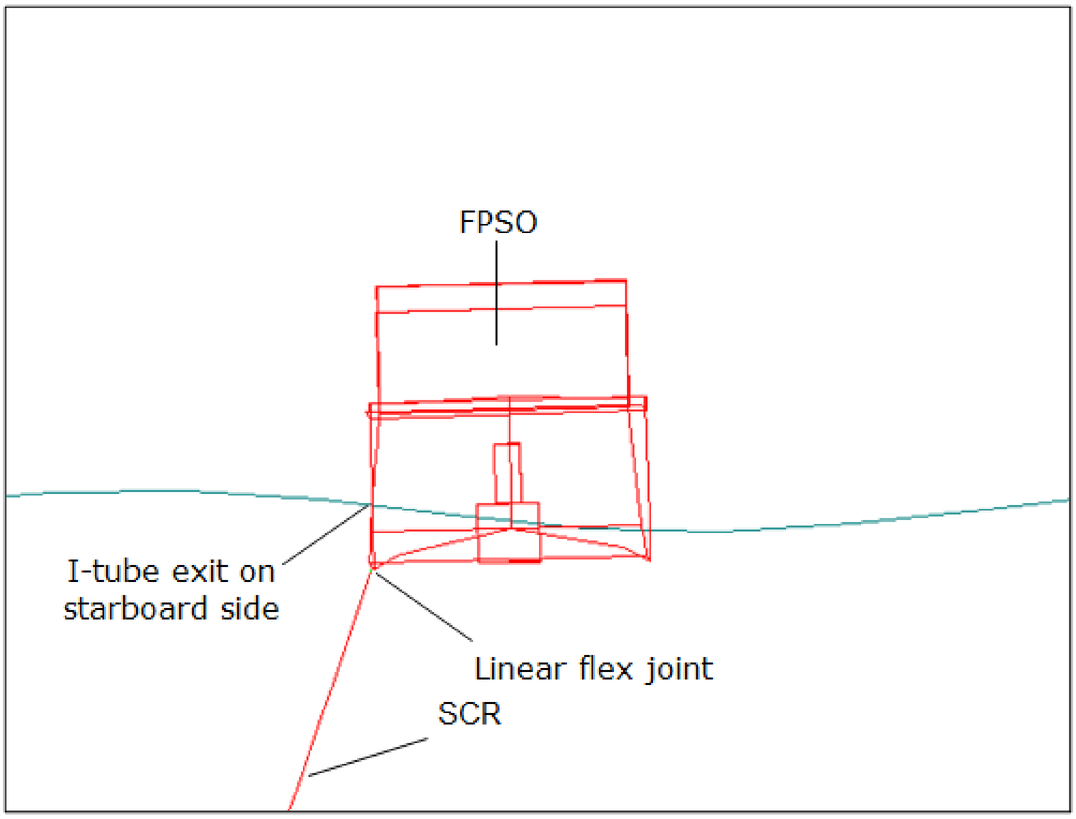
The riser hang-off at the top of the SCR incorporates an I-tube which is modelled in Flexcom as a single rigid beam. The riser has a natural hang-off angle in the initial position of 12°, but an 18.5° hang-off angle is subsequently achieved via the Solution Criteria Automation feature. The seabed is modelled as flat and elastic, with both longitudinal and transverse friction coefficients specified. The riser is filled with seawater. The figure below shows the SCR layout.



SCR Layout

The seabed touchdown region and at the vessel hang-off are the areas where the SCR response due to vessel motions is most critical. The riser design requires that von Mises and bending stresses cannot exceed allowable values at any point along the SCR. Likewise effective compression must be avoided at all costs. From the point of view of riser modelling, sufficient element refinement at touchdown is important to ensure predicted values in this region are realistic.

The figure below shows the SCR system at hang-off as the FPSO undergoes environmental loading. It is important to closely monitor this region as the vessel motions, which are dictated by the applied environment, can induce large moments and stresses in the SCR at this point.



FPSO under Environmental Loading

Analyses

INITIAL STATIC ANALYSIS

The base node is fixed in the 3 translational degrees of freedom. Vessel boundary conditions are specified in the first three DOFs at the uppermost node. The initial position of the vessel reference point, at approximately 9m above the MWL, and the RAOs are also input.

OFFSET ANALYSIS

The only change here is that an 18.5° hang-off angle is requested via the Solution Criteria Automation feature. All of the boundary conditions are unchanged and carry through automatically from the initial static analysis.

HANGOFF ANALYSIS

Additional rotational vessel boundary conditions are used to fix the upper end of the riser at the desired 18.5° hang-off angle.

CURRENT ANALYSIS

A piecewise-linear current, varying between 0.15 m/s at the mudline to 0.3 m/s at the MWL, is applied to the riser. The boundary conditions again remain unchanged and are carried through automatically from the hang-off analysis.

TIME DOMAIN DYNAMIC ANALYSIS

Three random sea analyses are performed. In each case the seastate is characterised by a Pierson-Moskowitz wave spectrum with a T_z of 9s. Three different H_s values are specified, namely 1m, 2m and 3m. The boundary conditions remain unchanged and are carried through automatically from preceding analyses. Since vessel boundary conditions and RAO data have previously been input, dynamic motions are automatically applied with the application of wave loading. Purely for efficiency reasons, the time domain simulation is analysed for ½ hour only, whereas a 3 hour simulation would generally be considered recommended practice.

FREQUENCY DOMAIN DYNAMIC ANALYSIS

Again, three Pierson-Moskowitz spectra, with a T_z of 9s and H_s values of 1m, 2m and 3m, are specified, for comparison with the time domain. Like the time domain, no specification of boundary conditions is required in the frequency domain: all of the boundary condition data carries through from the current analysis.

MODAL ANALYSIS

A modal analysis is carried out to calculate the natural frequencies and mode shapes of the steel catenary riser. The riser is naturally designated as an SCR for the purposes of Shear 7 output. The output is based on the first 50 modes and 500 equally spaced segments. The first 100 eigenpairs are requested, as recommended practice is to specify twice the number of natural frequencies as you are actually interested in. Note that output is requested for the set entitled SCR only, as this comprises riser elements only, and excludes the upper flex joint and rigid I-tube element. The composition of this set is actually defined in the Flexcom static analysis, and as the definition automatically carries through, there is no need to redefine it again in the modal analysis.

FREQUENCY DOMAIN FATIGUE ANALYSIS

A frequency domain fatigue analysis is performed using LifeFrequency to estimate fatigue damage and predict fatigue life. This example illustrates the operation of LifeFrequency in the Postprocessor with Stress Spectra mode. An initial static analysis precedes a series of near and far static offsets, each of which is then followed by a Flexcom random sea analysis, to determine the dynamic riser response to a fatigue load case matrix. LifeFrequency is then run in the program Postprocessor with Stress Spectra mode of operation, to accumulate the fatigue damage from the individual seastates.

The model used is very similar to that of the time domain dynamic analysis examined in the preceding section of this example. In this case however, the vessel has an initial yaw orientation of zero, so that offsets in the local vessel surge axis correspond to near and far offsets of the SCR. Also the specified RAO data is heading independent, and corresponds to head seas only.

A relatively simplistic fatigue matrix of 12 load cases defines the long-term seastate environment. In practical applications many more would typically be used. Six of the cases represent ‘Far’ loading conditions and six are ‘Near’ cases. For each individual case, an individual wave spectrum, mean offset, drift amplitude and drift frequency are defined. The table below presents the fatigue load case matrix.

Fatigue Load Case Matrix

Load Case	Offset Case	Seastate Data		Offset (m)	Drift Data		Percentage Occurrence
		H_s (m)	T_z (s)		Amp. (m)	Period (s)	
1	Near	0.75	5.24	4.84	0.16	138.50	4.4

2	Near	1.25	5.27	5.01	0.24	174.22	10.4
3	Near	1.75	5.77	5.28	0.51	196.46	9.6
4	Near	2.25	6.26	5.63	0.83	193.05	8.4
5	Near	2.75	6.89	6.00	1.15	193.05	5.8
6	Near	3.25	7.72	6.31	1.36	180.18	3.4
7	Far	0.75	5.24	4.39	0.16	136.80	7.8
8	Far	1.25	5.27	4.20	0.24	183.15	14.4

9	Far	1.75	5.77	4.01	0.51	193.05	13.2
10	Far	2.25	6.26	3.79	0.85	196.46	12.0
11	Far	2.75	6.89	3.62	1.17	196.46	7.4
12	Far	3.25	7.72	3.52	1.31	200.00	3.2

For the nominal location in question, the current distribution is assumed to be very nearly constant throughout the year, so the same current definition is used in all the Flexcom random sea analyses.

Fatigue data comprises the analysis SCF and S-N curve. For this example a stress concentration factor of 1.2 is used. The analysis S-N curve is log-linear, with m and K values of 4 and 1.15×10^{15} respectively. No endurance limit is specified. Thickness effects are ignored in this analysis.

FREQUENCY DOMAIN CYCLE COUNTING ANALYSIS

A sample frequency domain cycle counting analysis is performed using Histogram to illustrate its operation. It is based on the same fatigue load case matrix as the preceding fatigue analysis, and the results are output in the form of response histograms. In this example, a response histogram of bending moment is requested for the location of minimum fatigue life as predicted in the preceding fatigue analysis. 20 “bins” or divisions are specified going from a minimum of 0 to a maximum of 100. the bending moment data is assigned a scale factor of 0.001, so the range of the corresponding response histogram is from 0 to 100kNm.

Results

This section contains information on:

- Dynamic Analysis
- Modal Analysis
- Frequency Domain Fatigue Analysis
- Frequency Domain Cycle Counting Analysis

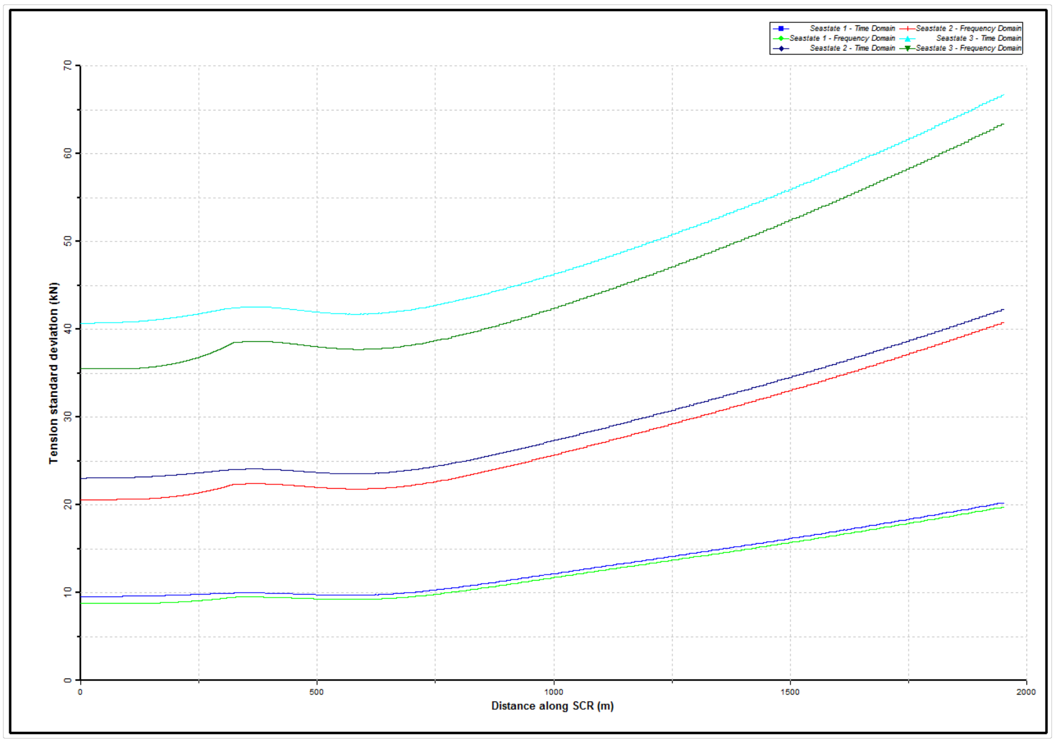
Dynamic Analysis

Results from the dynamic analyses are presented in the figures below; each plot compares results from the time and frequency domain analyses. In this example, von Mises stress and effective tension are the critical parameters. The seabed touchdown and hang-off regions are the key locations.

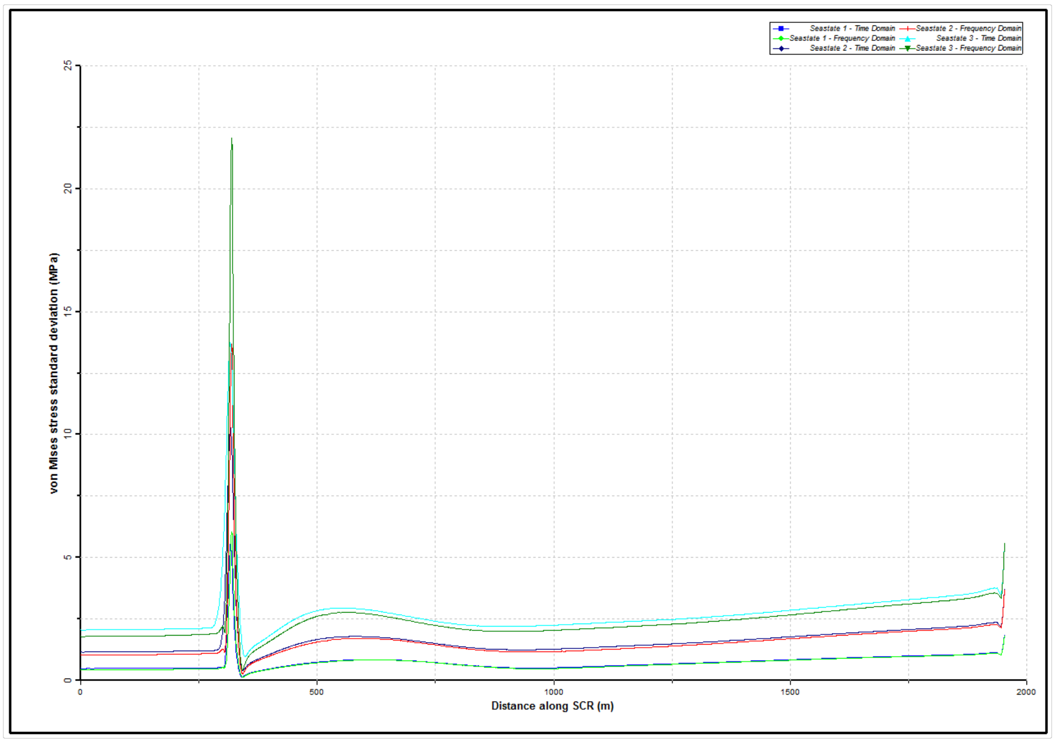
The Dynamic Tensions figure below shows a plot of standard deviation of effective tension from the three analyses. Note that in this plot ‘Seastate 1’ refers to the 1m H_s, ‘Seastate 2’ to the 2m H_s, and ‘Seastate 3’ to the 3m H_s. Excellent agreement is observed between time and frequency domain analyses for the lowest H_s, the agreement is still good for the 2m case, while for the 3m case the agreement is less good but still reasonable. This is discussed in more detail presently.

The Dynamic von Mises Stresses figure below shows a plot of standard deviation of von Mises stress. There are noticeable peaks, located approximately 315m along the riser, which are typical of this type of simulation. These peaks naturally correspond to the SCR touchdown region. A second, lesser peak can be observed at the riser hang-off.

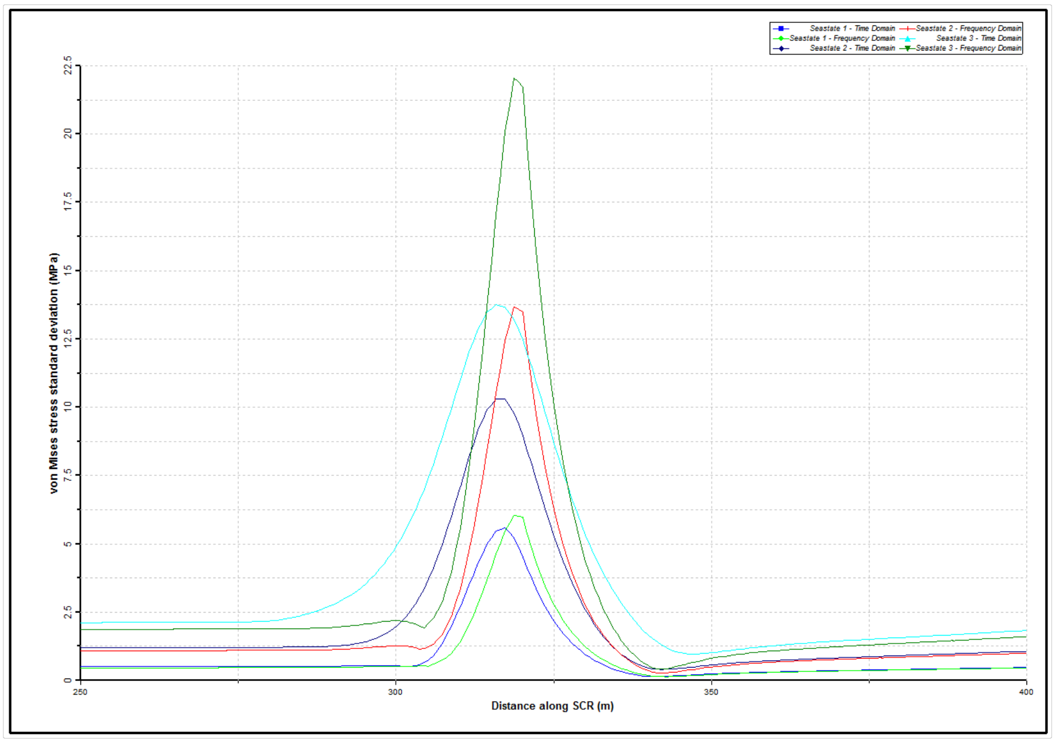
Time and frequency domain results are in close agreement everywhere except in the touchdown region. The almost exact agreement at the vessel connection is particularly noteworthy. The touchdown region stresses are more closely examined in the Dynamic von Mises Stresses (Detail) figure below, which shows a detail from Dynamic von Mises Stresses figure below. Here again it is clear than the time and frequency approaches give almost identical results for the 1m case, are in reasonable agreement for the 2m, but differ significantly for the highest H_s.



Dynamic Tensions



Dynamic von Mises Stresses



Dynamic von Mises Stresses (Detail)

Differences between the time and frequency domain results can be attributed to the fact that nodes that are on the seabed in the static configuration remain on the seabed throughout the frequency domain analysis. In the time domain on the other hand the changing nature of the riser/seabed interaction is accurately modelled. This means that dynamic stress variations are spread over a greater length of riser in the time domain, with a consequent reduction in the peak values – compare for example the differing extents of the touchdown region in the two graphs for Seastate 3 in the Dynamic von Mises Stresses (Detail) figure above. Obviously this effect is more pronounced in the higher seastates because the vessel motions are larger and the riser/seabed interaction more significant. It is worth pointing out that the SCR dynamic analyses in the frequency domain takes only a couple of minutes on a standard PC, compared to about an hour for the corresponding time domain runs.

Modal Analysis

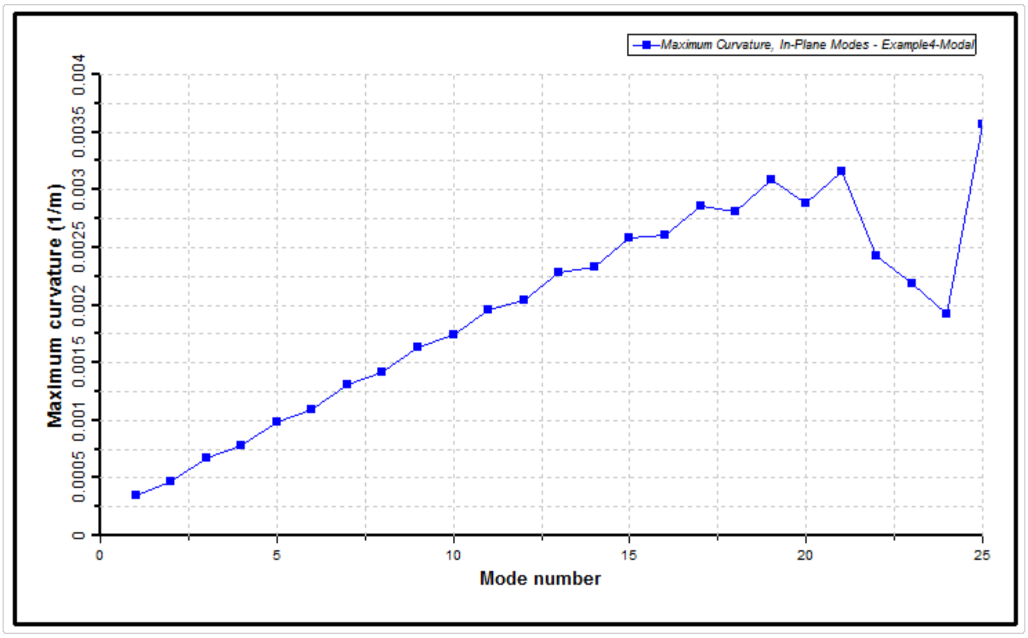
The table below presents the natural frequencies of the SCR for the first 20 modes. For an SCR, the division of bending modes into In-Plane and Out of Plane modes is a relatively straightforward operation. For In-Plane modes, displacements occur only in the plane of the SCR. Likewise Out of Plane modes have displacements normal to the initial plane of the SCR only. Any mode combining motions in all three directions is deemed Unknown.

Riser Natural Frequencies

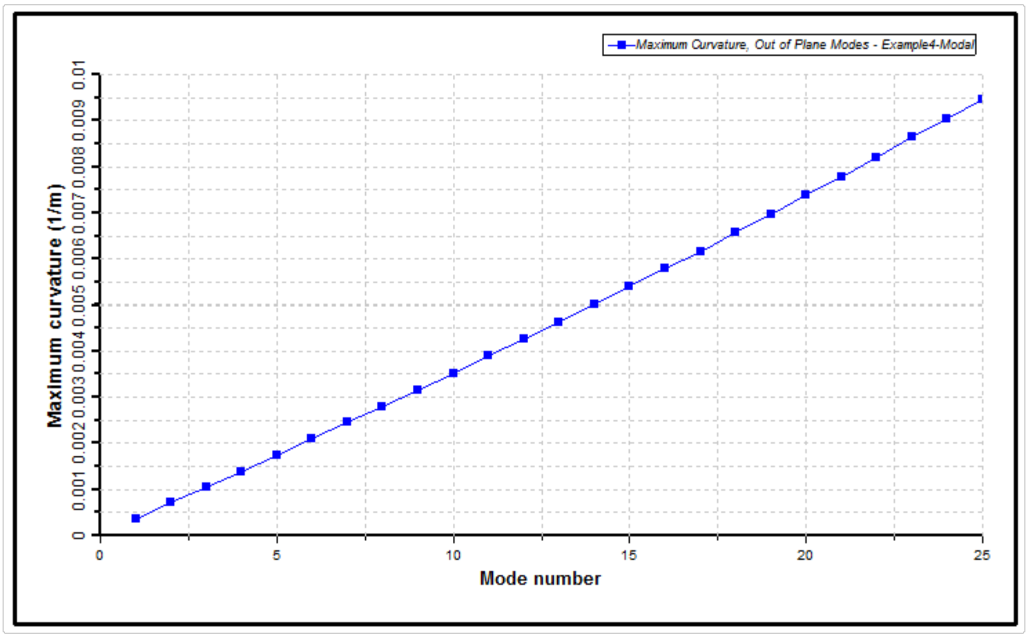
Mode No.	Period (s)	Type
1	45.98	Out of Plane
2	27.43	In-Plane
3	23.20	Out of Plane
4	16.97	In-Plane
5	15.50	Out of Plane
6	12.12	In-Plane
7	11.63	Out of Plane
8	9.62	In-Plane
9	9.30	Out of Plane
10	7.91	In-Plane
11	7.75	Out of Plane
12	6.77	In-Plane

13	6.64	Out of Plane
14	5.88	In-Plane
15	5.80	Out of Plane
16	5.22	In-Plane
17	5.15	Out of Plane
18	4.68	In-Plane
19	4.63	Out of Plane
20	4.25	In-Plane

The two figures below plot maximum curvature as a function of mode number for the in-plane and out of plane bending modes respectively (these plots are generated automatically by Modes). This type of plot is created to help in identifying so-called 'mixed modes' (refer to Modal Analysis for further information), where mixed modes tend to appear as local maxima or spikes. The maximum modal curvature should be monotonically increasing with mode number if pure bending modes only are considered.



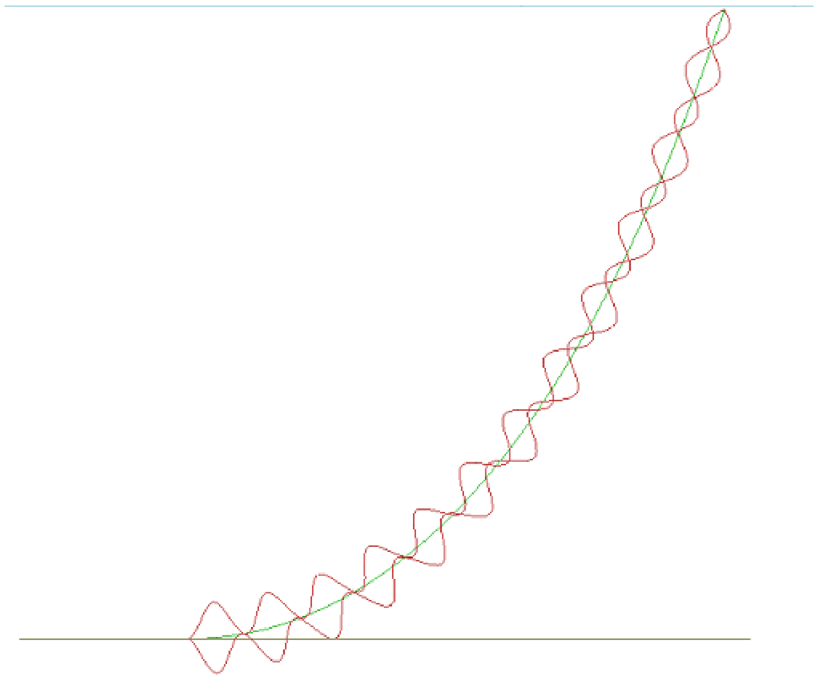
Maximum Curvatures, In-Plane Modes



Maximum Curvatures, Out of Plane Modes

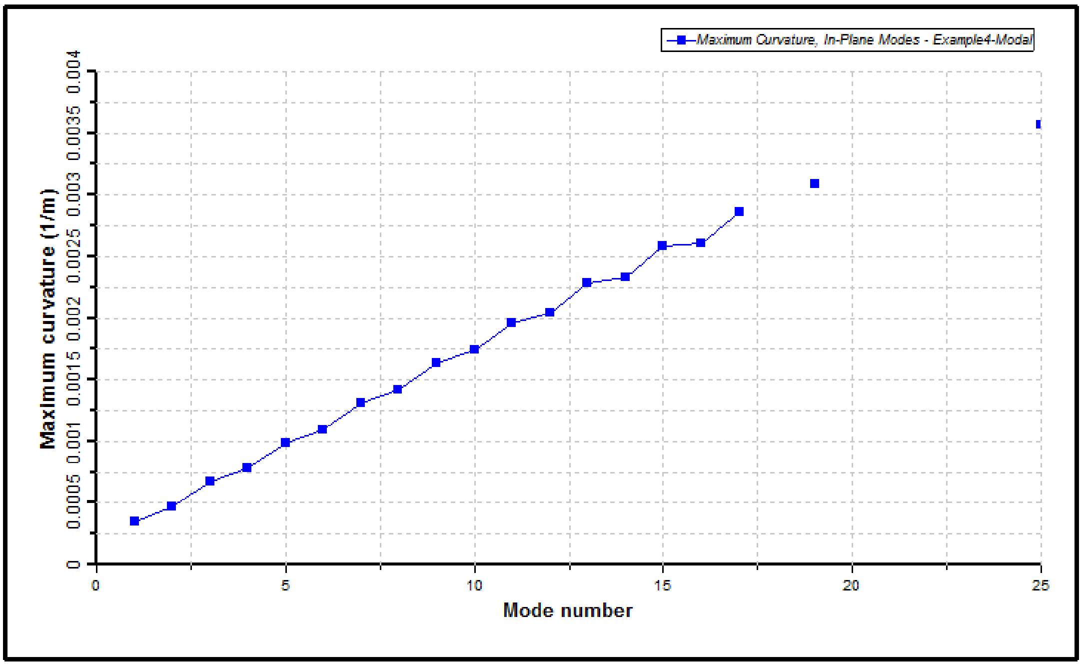
Some of the in-plane bending modes at the higher frequencies appear to be unreliable, as the monotonic trend is not followed. As it is difficult to clearly identify spikes in the plot, it is advisable to visually inspect the modal response manually. The procedure is relatively straightforward – you simply open two views of the structure on screen together. One should be a front elevation (i.e. a view normal to the plane of the SCR) and the other an end elevation (i.e. a view parallel to the plane of the SCR). Pure in-plane bending modes will be visible in the front elevation view, and pure out of plane bending modes will be visible in the end elevation view.

Additionally, if there are any combined axial-bending modes present, these will contain both bending deformation and axial deformation. In a pure bending mode, all of the nodes of the mode (as opposed to the nodes of the finite element model) will be stationary, so a standing wave effect will be exhibited. In a coupled axial-bending mode, there will be modal nodes which are not stationary, as there will be an axial response component present also. The figure below shows the modal response for the in-plane Mode No. 24. Standing wave behaviour is exhibited towards the top of the riser, but there are no stationary nodes in touchdown region, so clearly there is some axial response present.



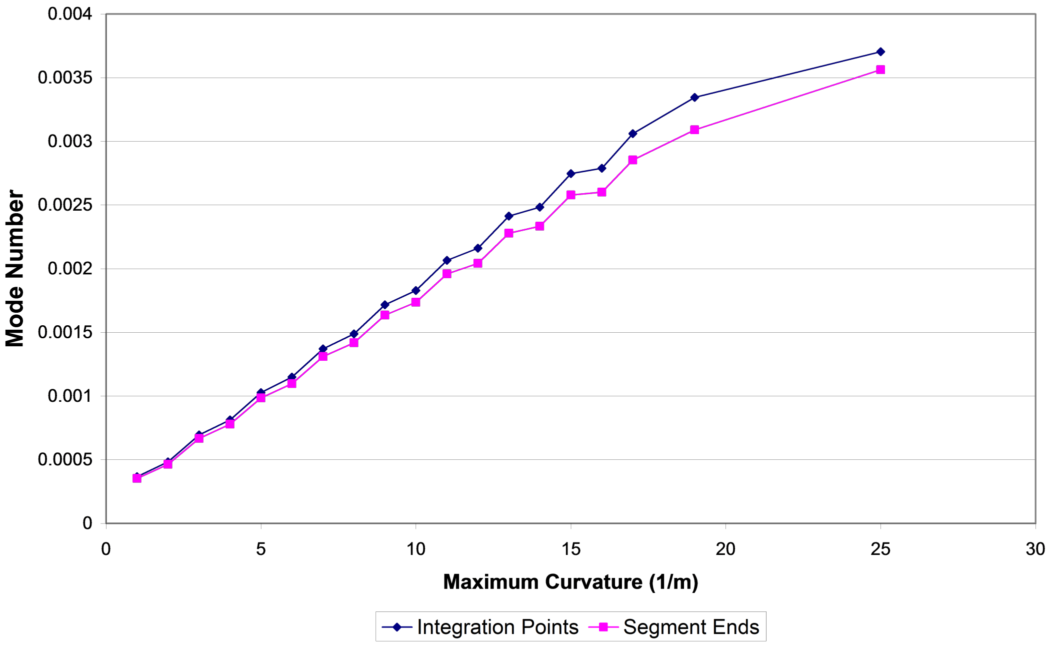
Modal Response for In-Plane Mode No. 24

Based on the above, some modes may be excluded from the Shear 7 output by invoking the SCR Modes – Exclude In-Plane option, and the modal analysis may be performed again. As the objective this time is to recreate the Shear7 output only (a purely postprocessing option), the Repeat Run capability is utilised. This facility allows you to indicate that you do not want the program to redo the full eigensolution. Instead the program is to read the eigensolution details from an earlier run, but to recalculate the Shear7 data, in this case with certain modes excluded. Note that a Repeat Run is practically instantaneous, as only postprocessing operations are performed. The figure below plots maximum in-plane curvature as a function of mode number, with some mixed modes excluded. The maximum curvature is now monotonically increasing with mode number, suggesting that only pure bending modes are now included in this output. Note also that although the excluded modes are not plotted anymore, the mode numbering remains unchanged; so if it is necessary to exclude further modes, it is still the ‘original’ mode numbers that are used.

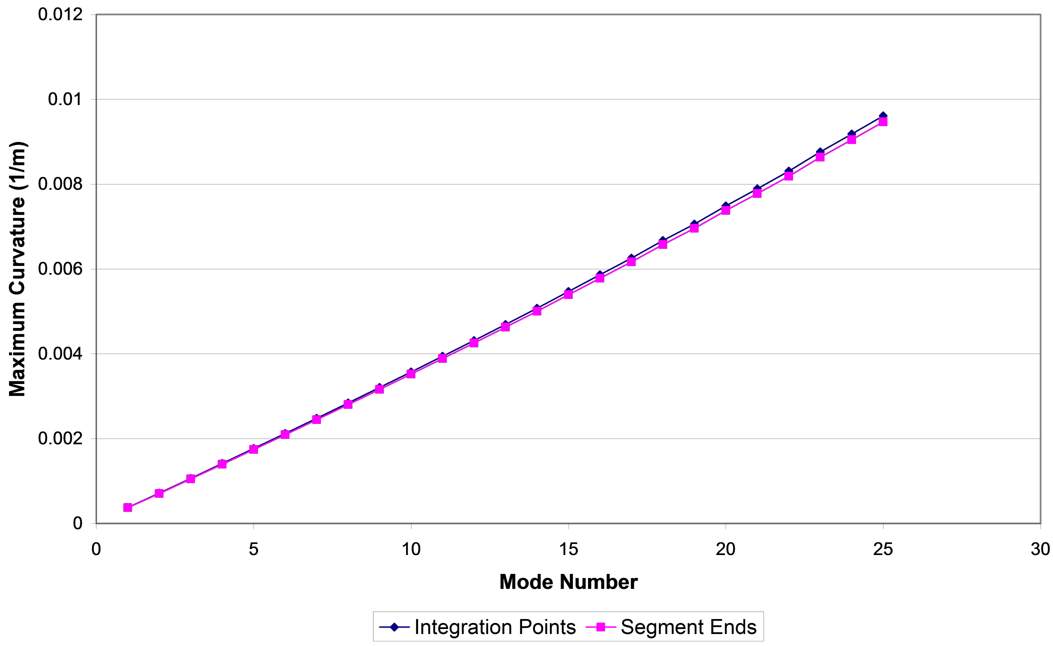


Maximum Curvatures, In-Plane Modes, Excluding Mixed Modes

The two figures below are graphs produced in Excel from tabular output presented in the output file. They plot the maximum in-plane and out of plane curvatures in each mode of the eigensolution, comparing the results at the element integration points, and the equal length segments used in the creation of Shear7 output. The intention here is to allow you to decide if you have specified enough segments to accurately capture the actual modal curvature distribution. In the present case the two sources agree reasonably well, so the specified number of segments appears reasonable.



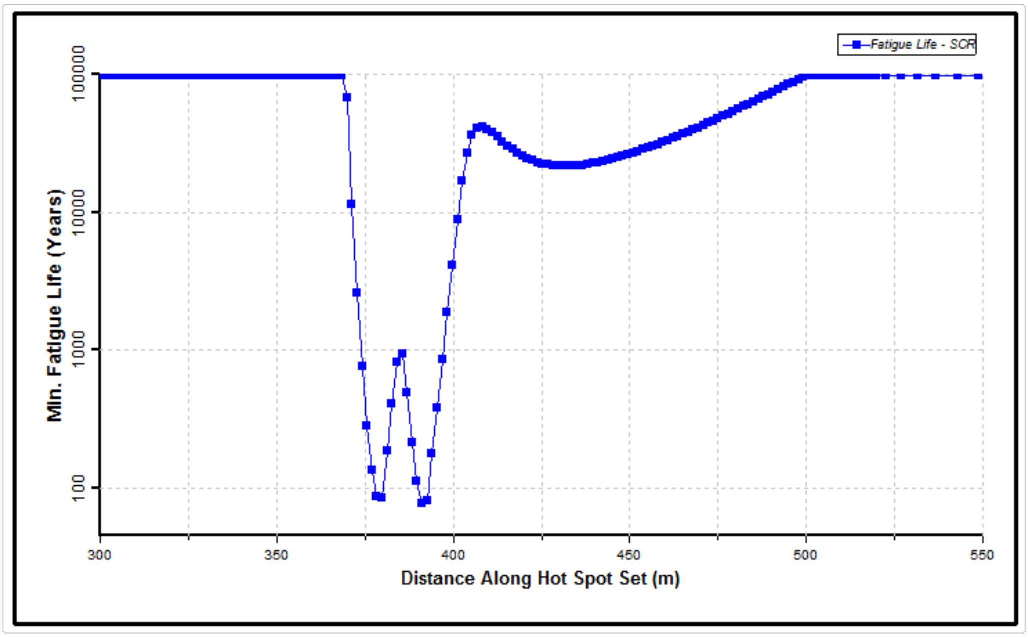
Maximum Curvatures, In-Plane Modes, Examining Segments



Maximum Curvatures, Out of Plane Modes, Examining Segments

Frequency Domain Fatigue Analysis

Fatigue results are requested for all nodes of the SCR finite element model, but over most of the structure LifeFrequency reports that fatigue lives are Infinite. In the program terminology this represents a predicted fatigue life of over 99,999 years. The shortest fatigue lives occur for hot spots in the touchdown region, as expected. The figure below plots the minimum predicted fatigue life in the touchdown region (note the log scale on the vertical axis).



Minimum Fatigue Life in Touchdown Region

Frequency Domain Cycle Counting Analysis

In this example, a response histogram of bending moment is requested for the location of minimum fatigue life as predicted in the preceding fatigue analysis. The results from the analysis are presented in tabular (Histogram in Tabular Format table below) and graphical (Histogram in Graphical Format figure below) format. It is clear from these outputs that the bins definition is reasonable for the histogram.

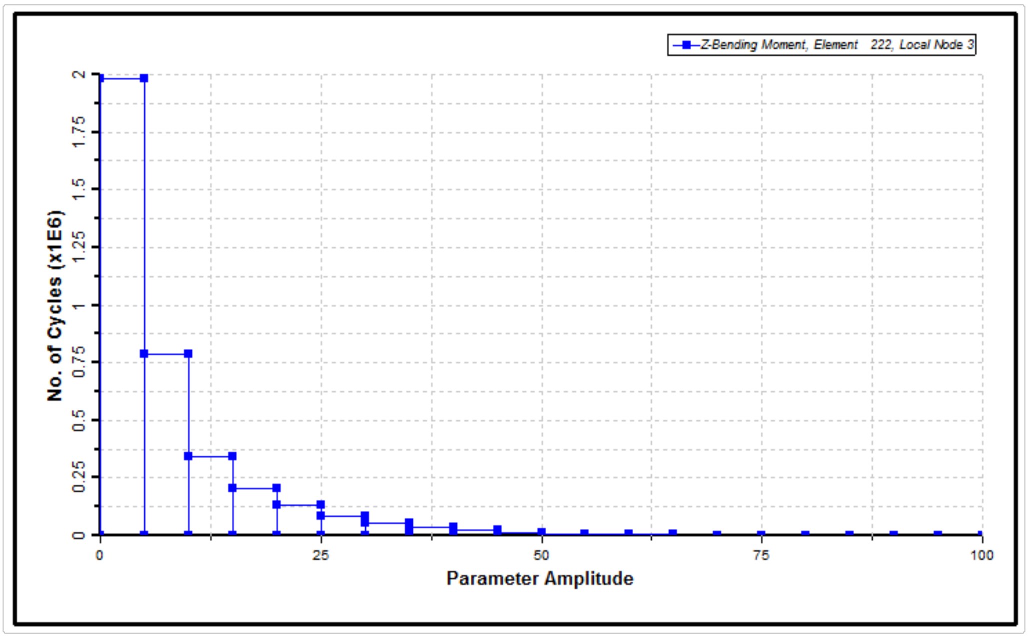
Histogram in Tabular Format

Parameter Histograms

Bin #	Parameter Amplitude	No. of cycles
1	0.0000 - 5.0000	1981324
2	5.0000 - 10.0000	786066
3	10.0000 - 15.0000	344830
4	15.0000 - 20.0000	204646
5	20.0000 - 25.0000	130258
6	25.0000 - 30.0000	83581
7	30.0000 - 35.0000	53924
8	35.0000 - 40.0000	34639
9	40.0000 - 45.0000	22158
10	45.0000 - 50.0000	14111
11	50.0000 - 55.0000	8899
12	55.0000 - 60.0000	5527
13	60.0000 - 65.0000	3371
14	65.0000 - 70.0000	2018
15	70.0000 - 75.0000	1185
16	75.0000 - 80.0000	681
17	80.0000 - 85.0000	382
18	85.0000 - 90.0000	208
19	90.0000 - 95.0000	110
20	95.0000 - 100.0000	56

Total number of cycles 3677974

Number of cycles outside the region 50



Histogram in Graphical Format

1.10.2.2 B02 - SCR Transfer

This example describes an analysis of the transfer of a steel catenary riser (SCR) from an installation vessel to a floating production unit. The overall layout of the example is as follows:

- Introduction gives a summary of the example analysis and describes the structure under consideration.
- Model Summary describes the structure from a modelling viewpoint in Flexcom. Important modelling techniques for this type of analysis are noted, as well as relevant Flexcom features and capabilities.
- Analysis Summary briefly describes the various analyses performed, discussing the environmental and loading conditions to which the riser is subjected.
- Results presents pertinent results from the example analyses and discusses their significance.

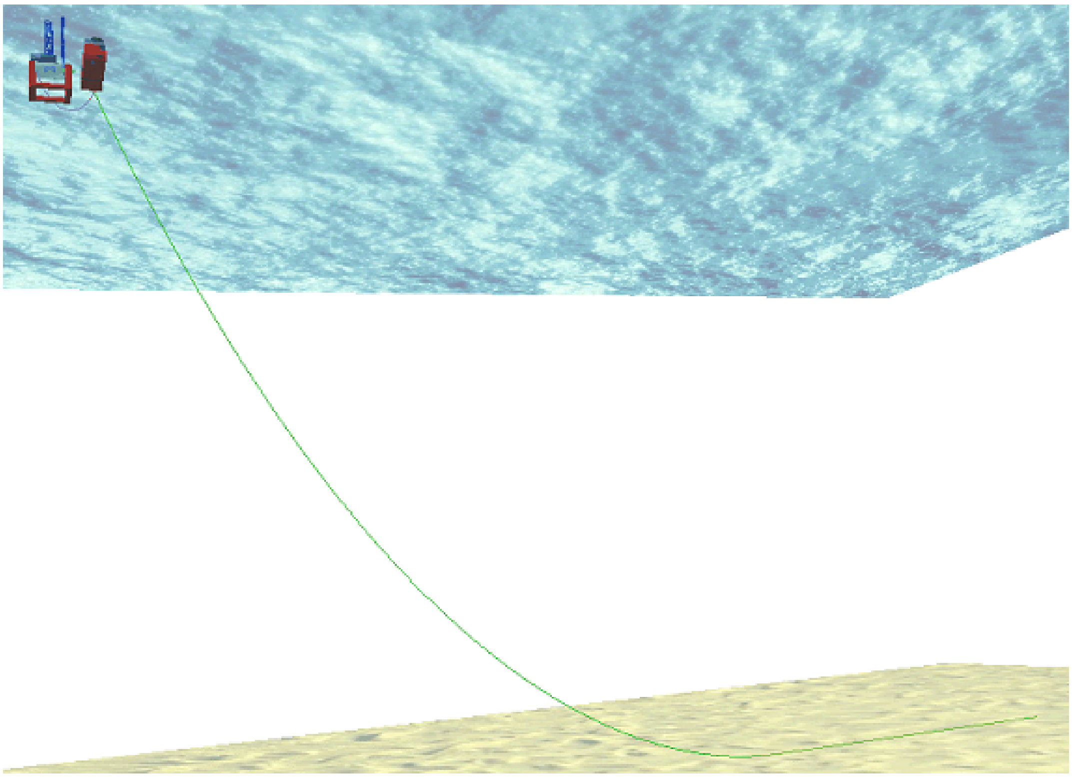
Introduction

The system modelled in this example is a simple SCR hanging from an installation vessel in 1500m of water. The SCR has been fully paid out by the installation vessel and is being transferred to a floating production unit (FPU). The installation vessel is oriented beam-on to the semi-submersible in preparation for transfer. The SCR is then first paid out to the transfer depth by a winch on board the installation vessel. It is then transferred to the host platform (FPU) winch and pulled in to its hang-off elbow on the host platform.

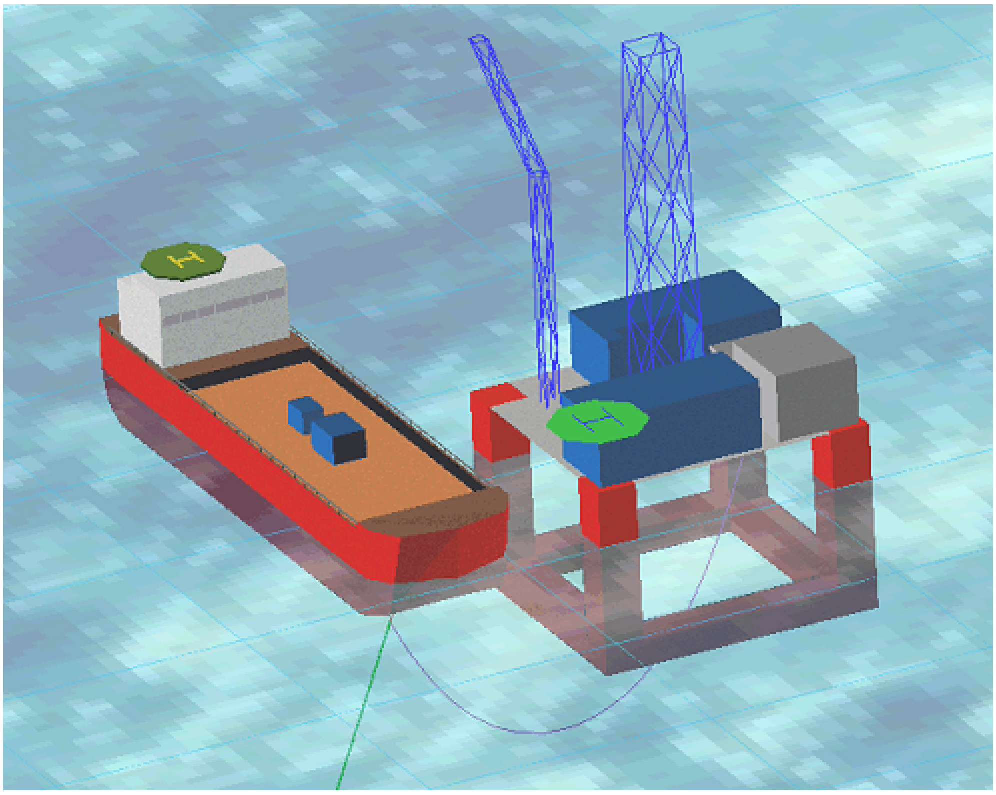
Model Summary

GENERAL

The first figure below shows the system that is being modelled here. The seabed has a slope of 2°. The transfer analysis starts from the point where the end of the SCR has been lowered by the installation vessel winch below the wave zone, and the vessel has been rotated beam-on to the host platform. Both the payout winch (on the installation vessel) and the pull-in winch (on the host platform) are modelled using winch elements. The second figure below shows a close-up of the installation vessel and the host platform.

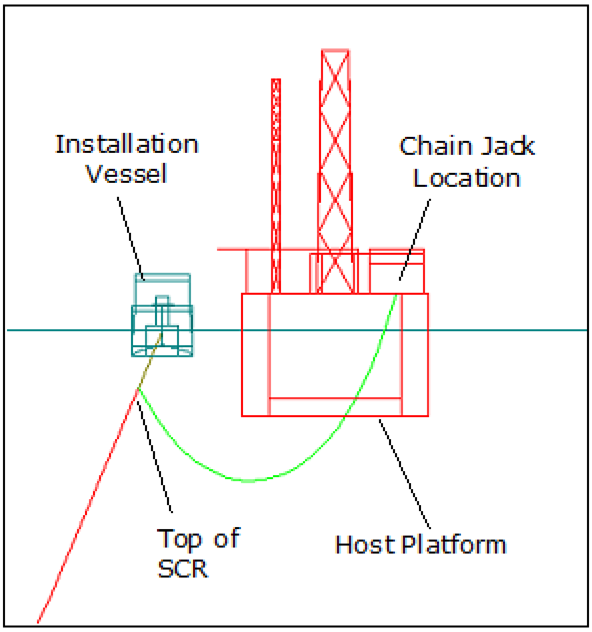


SCR Transfer Model

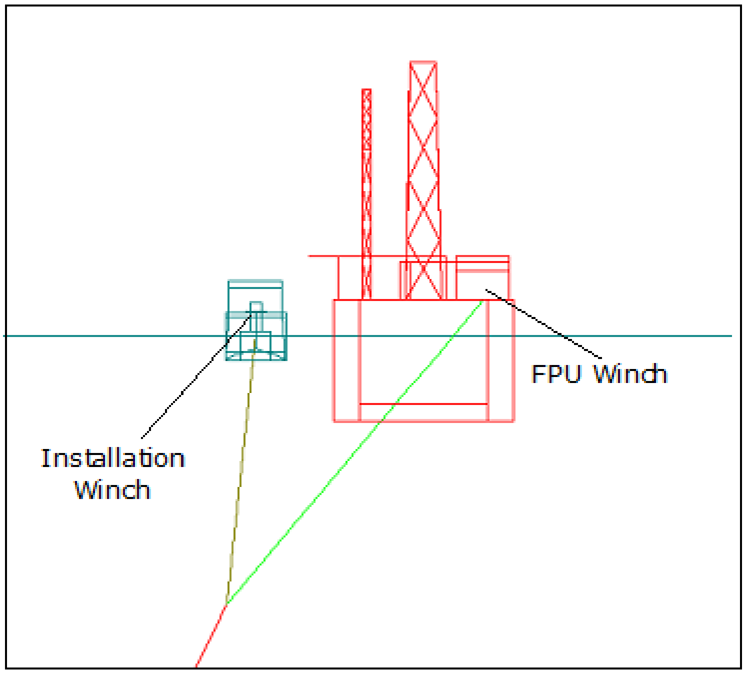


Installation Vessel and FPU

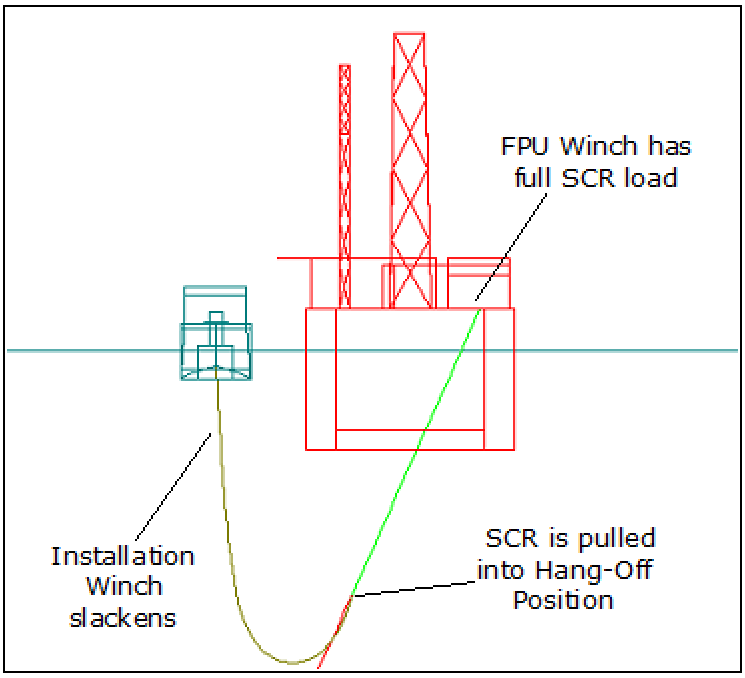
The three figures below illustrate the transfer procedure. The first figure below shows the system at the start of the analysis. The installation vessel has transferred some of the load to the host platform in the second figure. Finally in the third figure the host platform has assumed the full load of the SCR and the installation winch slackens and is eventually disconnected.



Vessel Supports SCR Load



Transferring SCR Load



FPU Supports SCR Load

Bending stresses, von Mises stresses and bending strains along the SCR, particularly in the sagbend region, cannot exceed allowable values.

WINCH ELEMENTS

Winch elements are normal beam-column elements which have the unique property that their lengths can vary during an analysis. In a dynamic analysis, the variation in length is defined in terms of a maximum winch velocity and a winching time sequence. The time sequence consists of (i) a ramp-up time when winch velocity is increased from zero to the maximum value; (ii) a time during which the velocity remains at this maximum, and (iii) a ramp-down time when the velocity returns to zero. The operation in a static analysis is less complex, with an overall change in length being applied linearly from the analysis start time to the end time. Winch elements can be used, for example, in pipelaying applications, for example in simulating the transfer of an SCR from a lay vessel to a TLP or semi-sub.

Analysis Summary

INITIAL STATIC ANALYSIS

The base node of the riser is fixed in the 3 translational degrees of freedom. Vessel boundary conditions (BCs) are specified in the translational DOFs at the ends of the two winches, and for the moonpool rigid elements. The initial positions of the two vessel reference points and the RAOs are also input. The SCR is specified as being full of seawater.

DYNAMIC ANALYSIS

A regular wave of amplitude 3.21 m and period 5.75s is specified. The BCs remain unchanged and are carried through automatically from preceding analyses. Since vessel BCs and RAO data have previously been input, dynamic motions are automatically applied with the onset of wave loading.

The payout of the installation winch takes places at 2.2 m/s over a 65s interval, including ramping up and down intervals of 5s. The pull-in of the FPU winch takes place at 1.25 m/s over 92s, again with 5s ramping up and down intervals. These values are for demonstration purposes only, to give an analysis of reasonable duration. Much longer winching times would be used in reality.

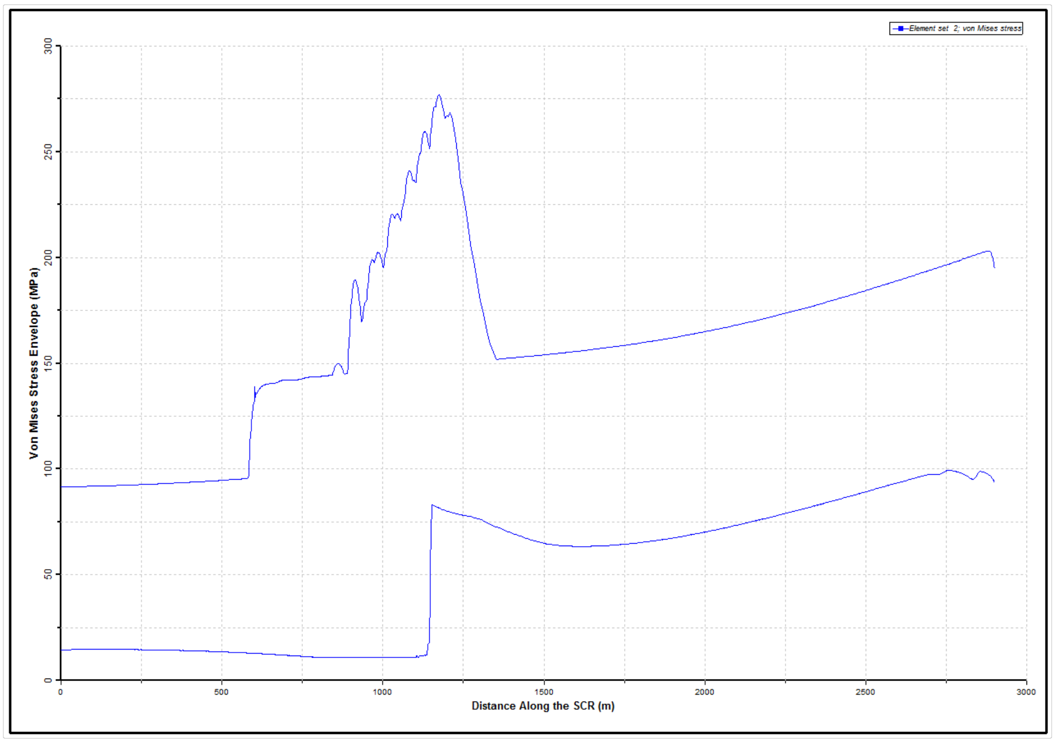
CLEARANCE POSTPROCESSING

Adequate clearance of the winches from their respective moonpools has to be ensured. This is checked here using Clear, the Flexcom postprocessor for clearance/interference calculations. Note that two rigid beam elements are included in the model to represent the edge of the moonpools on the two vessels, for the purpose of the clearance analyses.

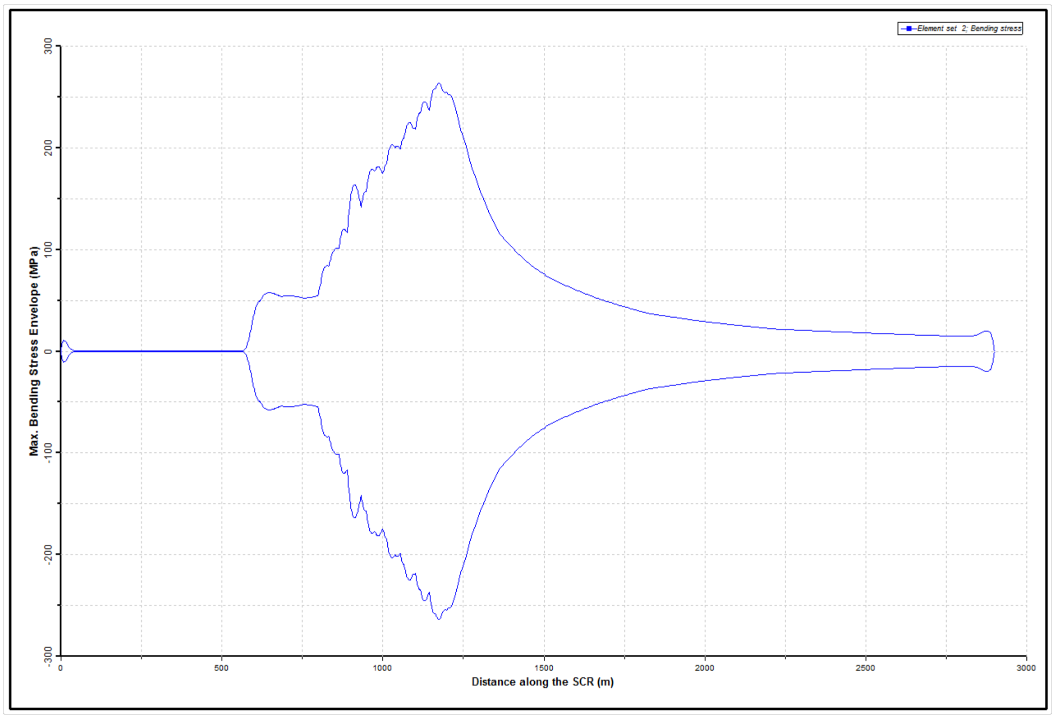
Results

DYNAMIC ANALYSIS

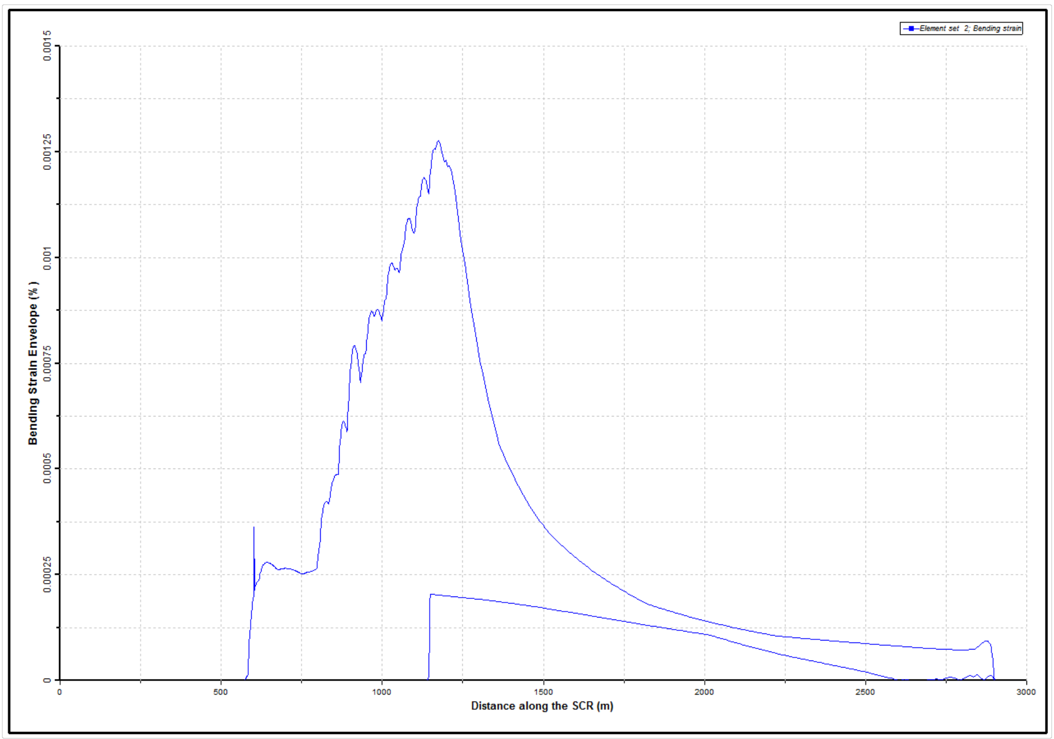
The maximum SCR bending stress, von Mises stress and bending strain are the critical results. The sagbend region is the key SCR location. The first figure below shows a plot of von Mises stress as a function of distance along the riser. The second figure below shows a corresponding plot of bending stress, while the third figure below shows the bending strains. Noticeable peaks of stresses and strains can be observed approximately 1150m along the riser, corresponding naturally to the touchdown region.



Von Mises Stress Envelope

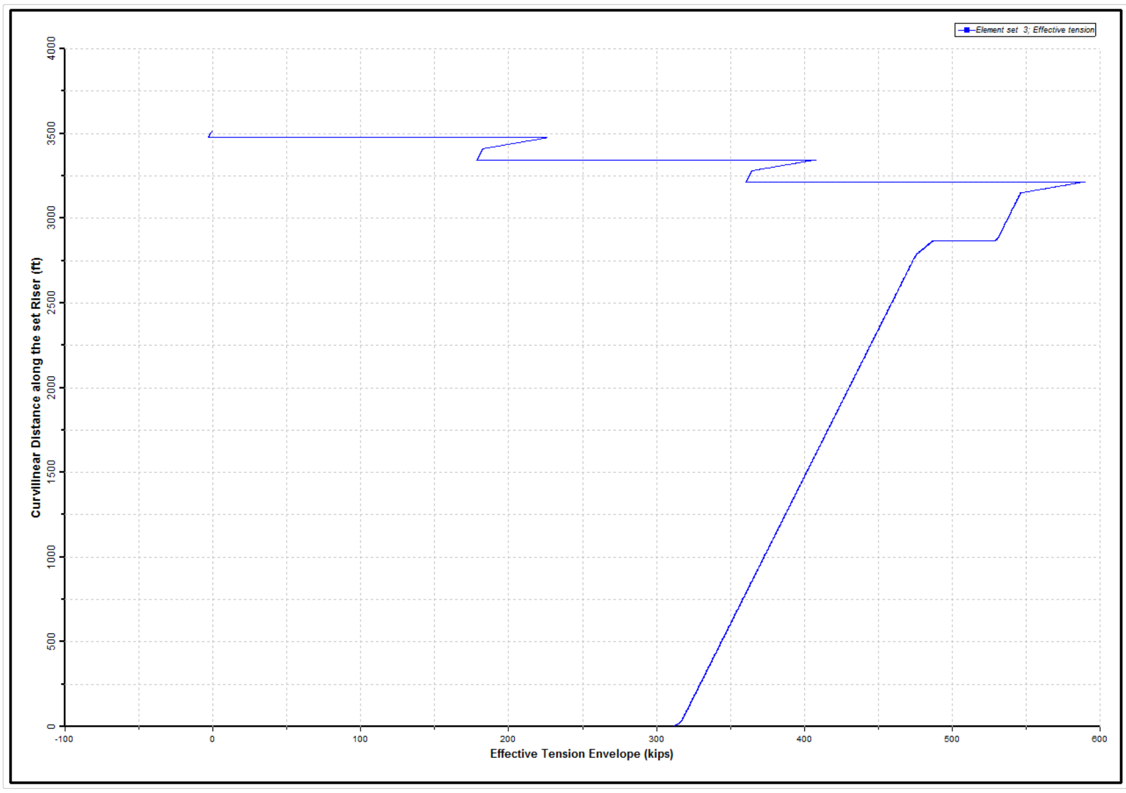


Bending Stress Envelope



Bending Strain Envelope

To avoid buckling it is essential that the riser does not go into significant compression at any point. The below figure shows the dynamic effective tension envelope, which shows that the riser remains in tension throughout.



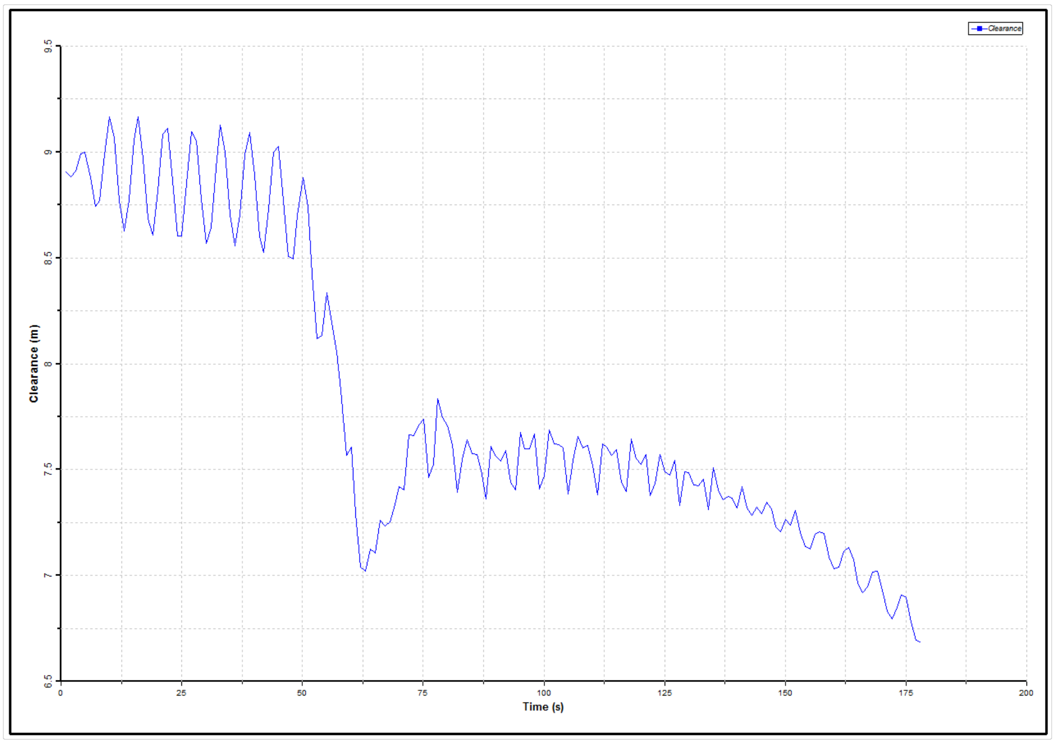
Effective Tension Envelope

CLEARANCE POSTPROCESSING

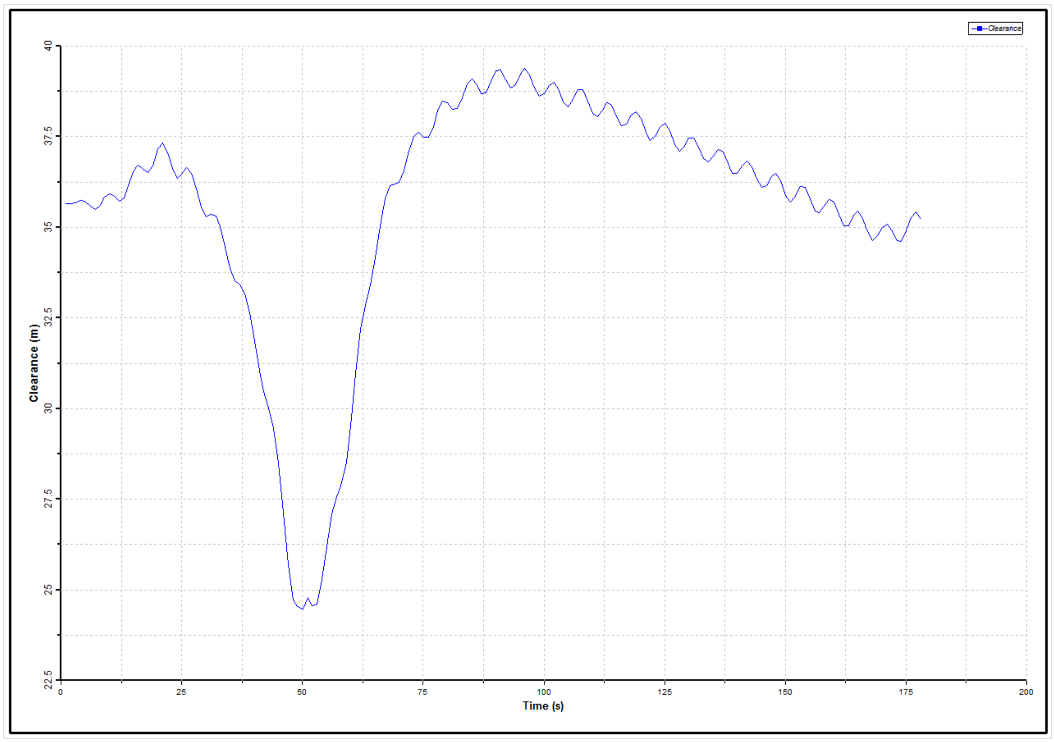
The clearance of the winches in their respective moonpools is also critical. Minimum clearances between the winches and moonpools are shown in the Winch Clearances Table below. Clearance timetrace plots from Clear are given in the Installation Winch Clearance figure and the FPU Winch Clearance figure below.

Winch Clearances

	Minimum Clearance from Moonpool (m)
Installation Winch	6.7
FPU Winch	24.5



Installation Winch Clearance



FPU Winch Clearance

1.10.3 C - Flexible Risers

Section C contains some examples of flexible risers, including:

- C01 - Free Hanging Catenary
- C02 - Multi-Line Flexible System
- C03 - Turret Disconnect

1.10.3.1 C01 - Free Hanging Catenary

This example describes static and dynamic analyses of a free-hanging catenary riser, and discusses a number of issues of particular relevance to the modelling of flexible risers. The overall layout of this example is as follows:

- Introduction gives an overview of the free-hanging catenary analysis, and notes some of the more important features of Flexcom which are relevant to the analysis.

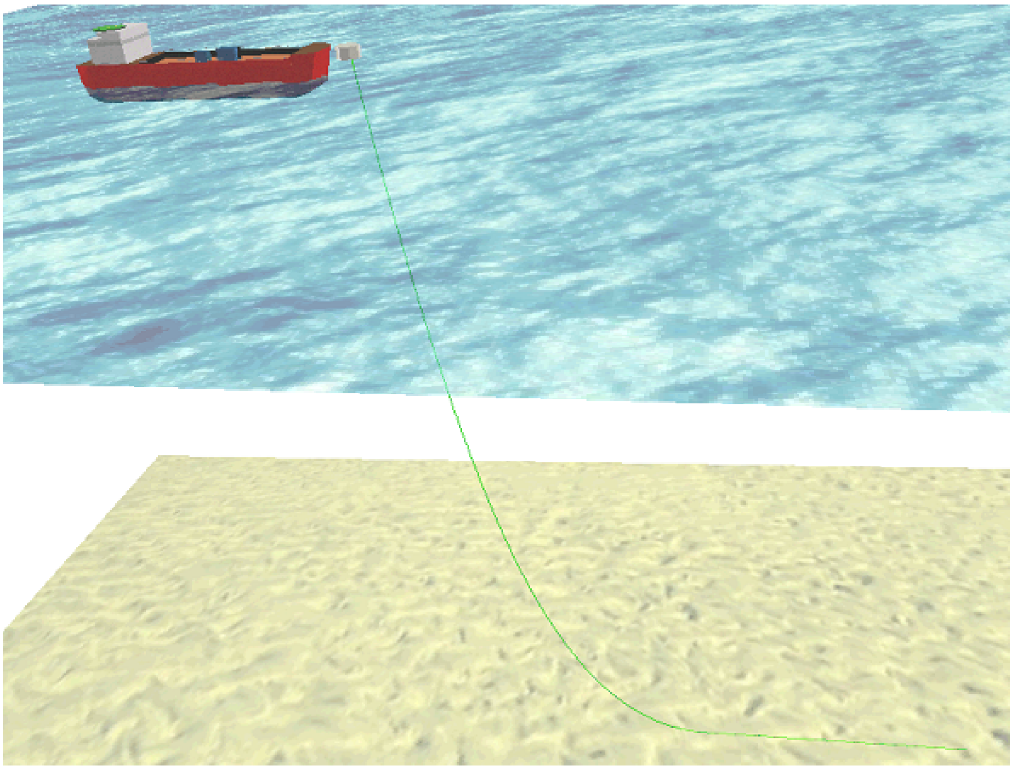
- Model Summary describes the model in more detail, and discusses the issue of modelling compression in flexible risers.
- Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions to which the riser is subjected.
- Results presents pertinent results from the various analyses performed and discusses their significance.

Introduction

This example considers the analysis of a flexible riser deployed in a free-hanging catenary configuration. The riser is deployed in 705m of water and is connected to an FPSO. The example consists of a static analysis of the riser subjected to vessel offset and current, and a dynamic analysis of the riser subjected to wave loading and first-order vessel motions.

A number of important features of Flexcom are illustrated here. Of particular significance is the program's ability to model buckling, both the onset of buckling instability and post-buckling behaviour. Other important program features invoked here are the non-linear bend stiffener facility, and the option to model a sloping seabed.

The free-hanging catenary riser configuration is shown in the figure below.



Free-Hanging Catenary

Model Summary

GENERAL

The riser in this example is a 6'' production riser, 1080m in length, sited in 705m of water. It is filled with oil and attached to an FPSO. The seabed is modelled as a rigid surface and has a 1.7° slope downwards from PLEM to vessel. The riser analysis is performed here in four stages. An initial static analysis applies gravity and buoyancy forces, and two static restart analyses are then used to introduce vessel offset and current loading. Finally, a dynamic analysis finds the response of the riser to vessel motions and wave loading.

MODELLING OF BUCKLING

The free hanging catenary configuration represents both the simplest and the cheapest option for the installation and procurement of a flexible riser in deep water. However, the catenary configuration couples the mean, low frequency offset and first order motion of the vessel to the seabed touchdown point response. Where vertical motion of the hangoff point is significant, compression forces can arise at touchdown that, when large, can cause global buckling of the pipe section, significantly reducing the dynamic pipe minimum bend radius. The phenomenon is particularly significant for FPSOs, which tend to have considerable heave response. In addition, the vertical motion is a combination of both the heave and pitch, so catenary risers attached to FPSOs, particular those where the riser hangoff is offset from the vessel centre of gravity, are likely to experience compressive loading and possible buckling.

The issue of modelling compression in flexibles was the subject of the publication McCann et al., (2003). The example data used here is based on one of the examples described by McCann et al, and the results presented here are a small subset of the results presented in the paper. Interested readers are referred to the relevant proceedings for a full version of this material.

McCann et al list a number of conclusions, some of which are repeated because of their significance to this particular example.

1. Flexcom can accurately predict both the onset of buckling and post-buckling behaviour, provided sensible values of element length and time-step are used, and provided the Euler buckling within each individual element is not exceeded.

   McCann et al demonstrate this by comparing Flexcom output for a column instability analysis with analytical and third-party solutions. Complete agreement with independent solutions is reported.

2. The paper proposes a non-dimensional velocity parameter ($V_{Hangoff}/V_{Terminal}$) which is both a simple and useful measure for assessing the likelihood of compressive buckling. $V_{Hangoff}$ is the maximum heave velocity of the riser hang-off, whereas $V_{Terminal}$ is a riser “terminal velocity” as defined by McCann et al. Where this parameter is greater than 1, buckling is expected. Further details are contained in McCann et al., (2003). For the riser and vessel data used in this example, $V_{Hangoff}$ is 4m/s and $V_{Terminal}$ is 2.72m/s. So the ratio is 1.47, making compression likely according to the McCann et al criterion. That compression does indeed occur is demonstrated in the dynamic analysis results.

3. Accurate modelling of compression is sensitive to element length and dynamic analysis time-step. McCann et al demonstrated this with a range of sensitivity analyses. In this example, the effect of time-step size is investigated.

4. Large variation between results from deterministic analysis can exist when buckling occurs. McCann et al recommend that a stochastic approach be used to extrapolate extreme curvature based on the standard deviation of curvature in the touchdown region, even in the case of regular wave analysis. In the dynamic analyses reported here, both max/min and standard deviation plots are presented to illustrate this recommendation.

Analyses

INITIAL STATIC ANALYSIS

The base of the riser is fixed in all translational degrees of freedom, while the upper end is attached in all DOFs to the vessel with a hangoff angle of 7°. The initial position of the vessel is specified with an undisplaced orientation of 0° to the global Y-axis in the YZ plane. The vessel RAOs are also input.

OFFSET ANALYSIS

The only change here is that a vessel offset of 24.8 m is applied in the horizontal (DOF 2) direction. The offset is ramped on over ten increments. All of the BCs are unchanged and carry through automatically from the initial static analysis.

CURRENT ANALYSIS

A piecewise-linear current is applied to the riser, varying from 1.2m/s at the MWL to 0m/s at the mudline. The current direction is from PLEM to vessel. The BCs remain unchanged and are carried through automatically from the offset analysis.

DYNAMIC ANALYSES

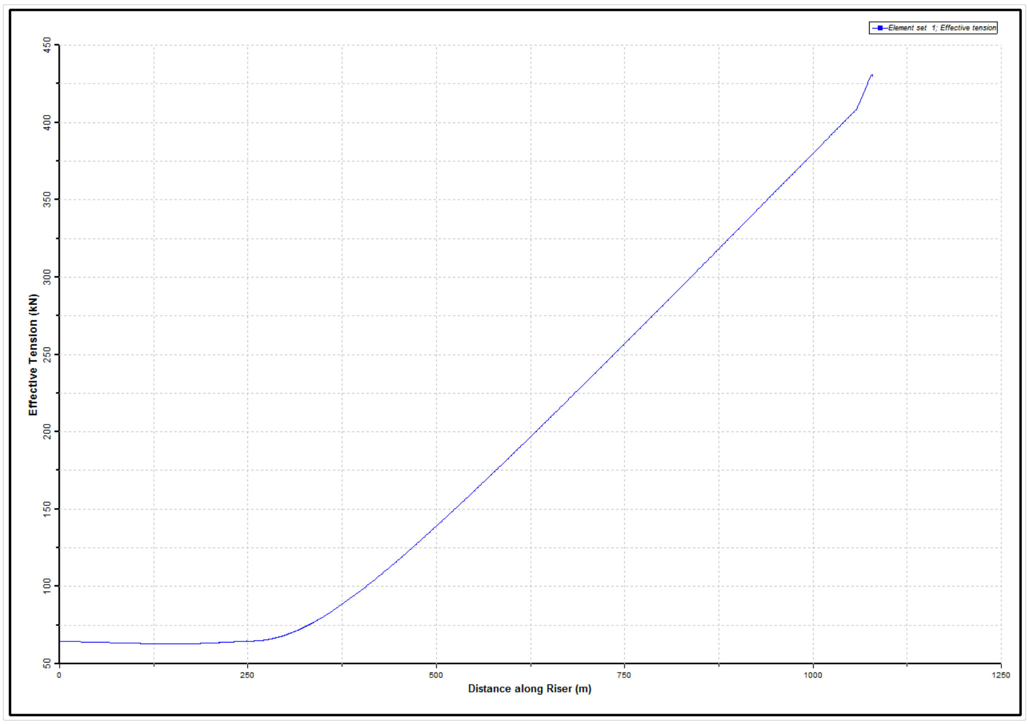
A regular wave of amplitude 5m and period 14s is specified. The analysis is run for 4 wave periods, with results from the last period used to generate response statistics. The BCs remain unchanged and are carried through automatically from the current analysis. Since vessel BCs and RAO data have previously been input, dynamic motions are automatically applied with the onset of wave loading.

As noted in the Model Summary Section earlier, a sensitivity study is performed on dynamic analysis time-step. The base case value used is 0.01s, and analyses are also performed with step values of 0.02s, 0.04s, and 0.005s. The sensitivity study is facilitated via the Keyword Based Variations feature, which allows several variations of the base case input file to be produced automatically.

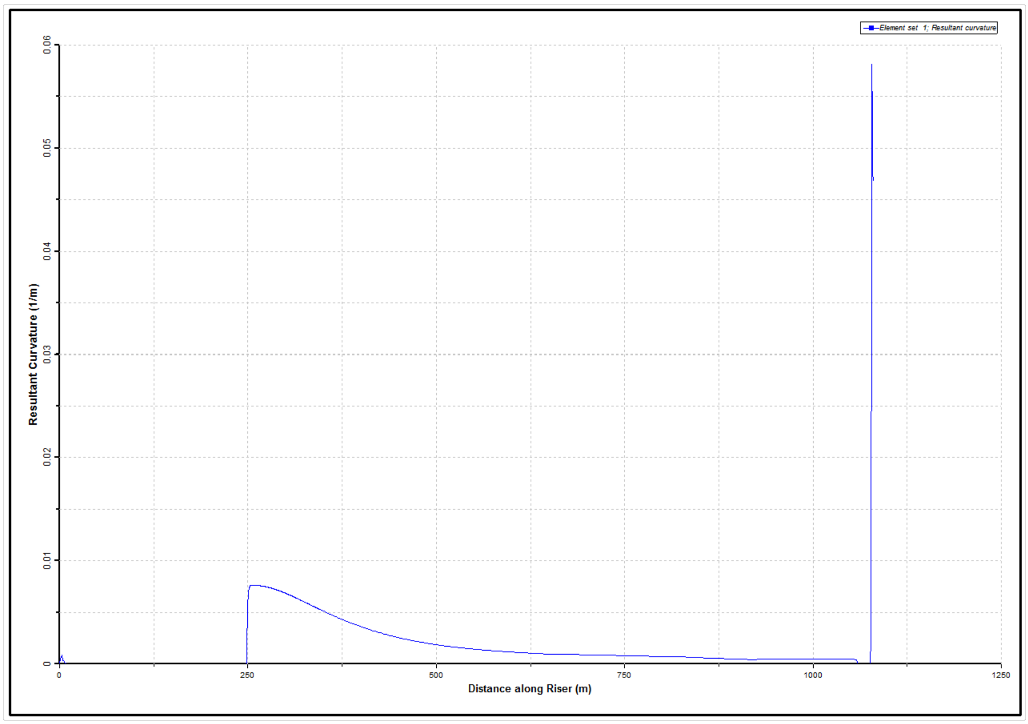
Results

CURRENT ANALYSIS

The static riser solution is shown in the Static Effective Tension Distribution figure to Dynamic Effective Tension Envelope Base Case figures below, which plot respectively the static structure configuration and the static distributions of tension and curvature in the riser. The Static Resultant Curvature Distribution figure shows that in its mean static position the riser is everywhere in effective tension. The large peak in curvature at the fixed vessel hangoff, where the bend stiffener is located, is also noteworthy.



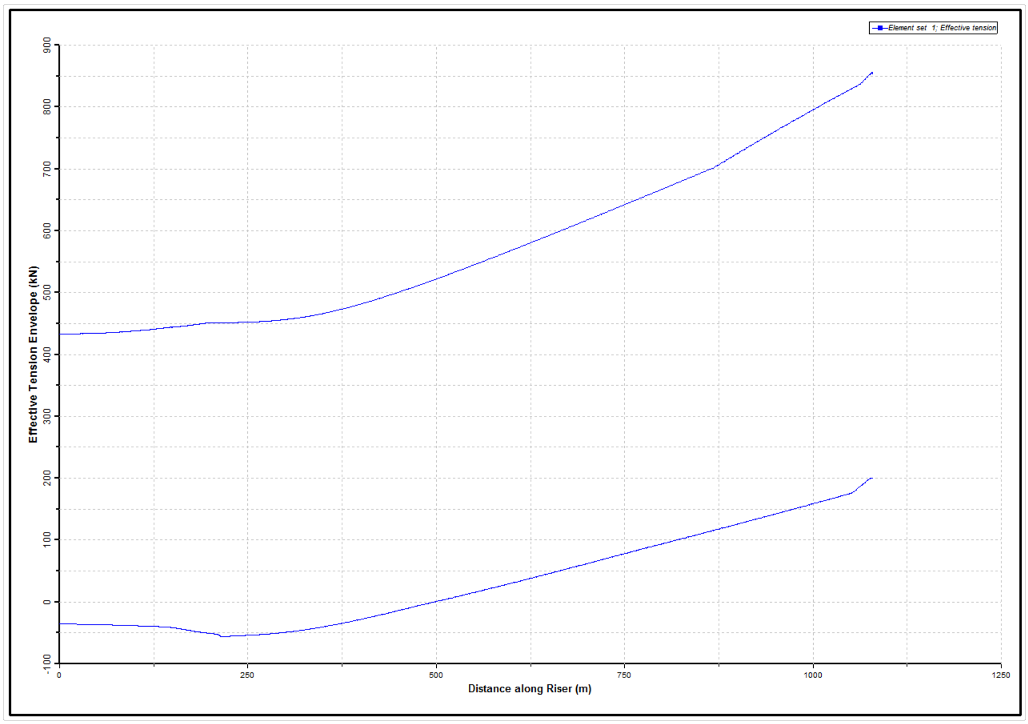
Static Effective Tension Distribution



Static Resultant Curvature Distribution

DYNAMIC ANALYSES

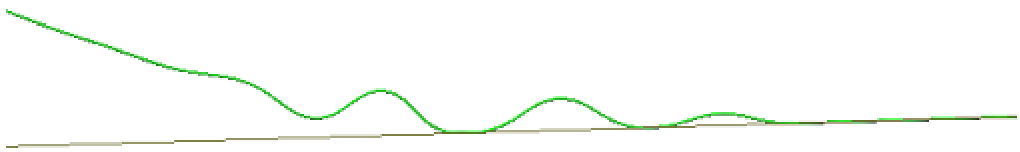
The Dynamic Effective Tension Envelope Base Case figure below presents the envelope of effective tension from the base case dynamic analysis, with the 0.01s time-step.



Dynamic Effective Tension Envelope Base Case

Clearly effective compression is predicted in the riser on the seabed. The minimum predicted value is approximately (-)52kN. Note that in the model used here, element lengths on the seabed vary between 0.5m and 3m. For an element of length 3m and bending stiffness 57.82kNm^2, the critical Euler buckling load is ~63kN. For an element of length 0.5m, it is ~2280kN. So the Euler buckling load is not exceeded in any individual element here, as per the findings of McCann et al., (2003).

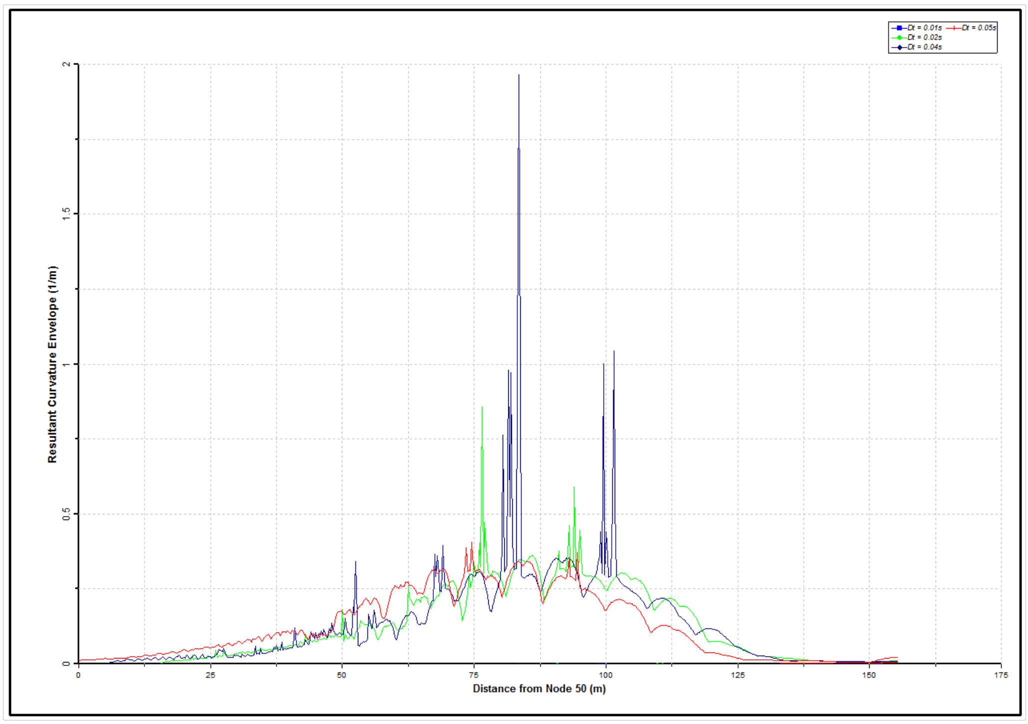
The Localised Buckling in Touchdown Zone figure below illustrates the configuration in the touchdown zone at a sample solution time during which the riser is in compression. The presence of localised buckling is clearly evident.



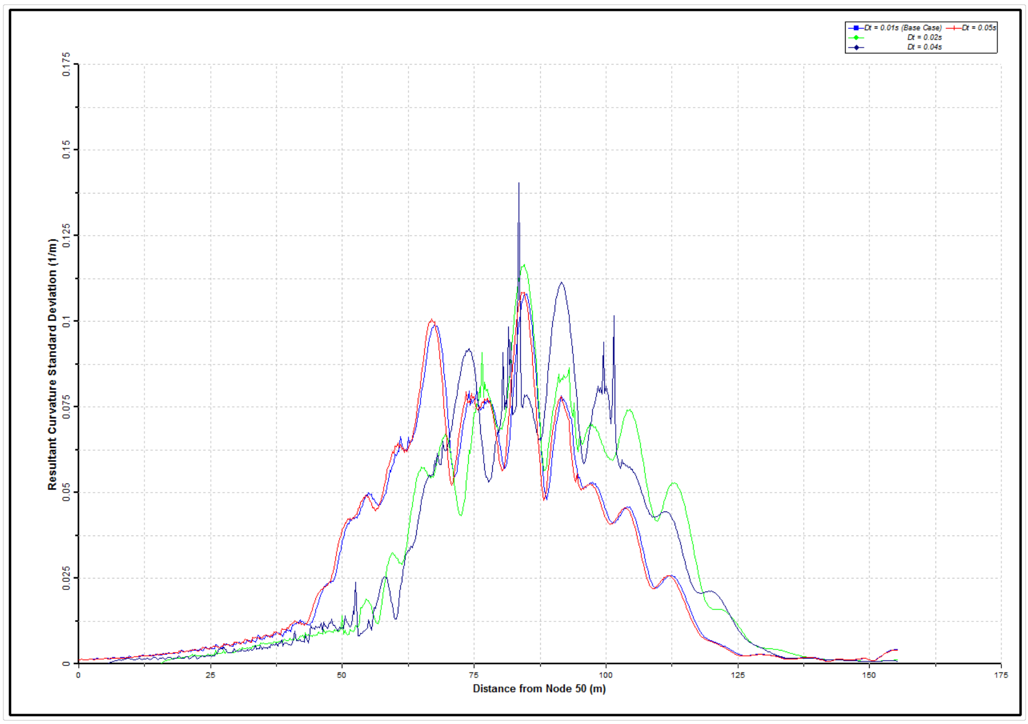
Localised Buckling in Touchdown Zone

The remaining figures show the effect of time-step size on curvature statistics. The Curvature Envelopes – Sensitivity to Time-Step figure shows max/min envelopes for the four time-step values used in this study, while the Curvature Deviations – Sensitivity to Time-Step figure plots the corresponding curvature standard deviations. A significant variation is observed in the Curvature Envelopes – Sensitivity to Time-Step figure with changing time-step. This is due to differences in the post-buckling behaviour of the riser. McCann et al show that the way in which buckling progresses is strongly influenced by the deformed position of the riser as buckling is initiated (McCann et al., 2003). So slight differences in deformed position between analyses with different time-steps lead to quite different behaviour thereafter.

On the other hand the Curvature Deviations – Sensitivity to Time-Step figure shows that standard deviations tend to be more uniform than the corresponding curvature distribution, even for the analysis with the large time-step (0.04s), the results from which in the Curvature Envelopes – Sensitivity to Time-Step figure appear untrustworthy. McCann et al report a similar conclusion from a sensitivity study based on element length. On that basis, they conclude that using a deterministic approach in selecting the most onerous curvature condition can lead to large errors, and that the use of a statistical extrapolation based on standard deviation provides a more robust method of deriving maximum curvature (McCann et al., 2003).



Curvature Envelopes – Sensitivity to Time-Step



Curvature Deviations – Sensitivity to Time-Step

1.10.3.2 C02 - Multi-Line Flexible System

This example describes a sample multi-line flexible riser system, and demonstrates how a complex flexible riser and mid-water arch system is readily built using Flexcom. The overall layout of this example is as follows:

- Introduction gives an overview of the flexible riser system.
- Model Summary describes the model in more detail, and discusses the use of the parameters used in setting up the model.
- Analyses briefly describes the various analyses performed, discussing the environmental and loading conditions to which the riser system is subjected.
- Results presents pertinent results from the analyses performed.

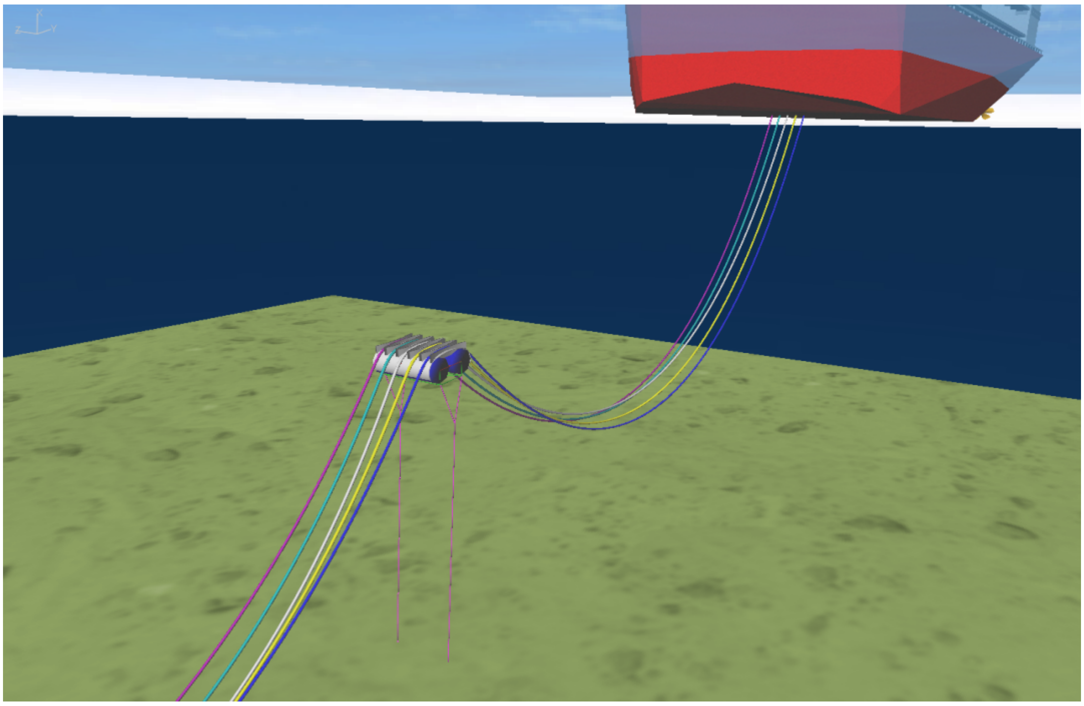
Introduction

This example considers the analysis of a multiple riser system in a Lazy-S configuration. This configuration, which is achieved by draping risers over the tethered mid-water arch, allows the upper catenary risers to absorb the vessel motions while isolating the lower catenary sections and any attached flowlines. The example consists of a static analysis, subject to gravity and buoyancy loads only, followed by a dynamic analysis subject to wave loading and first-order vessel motions.

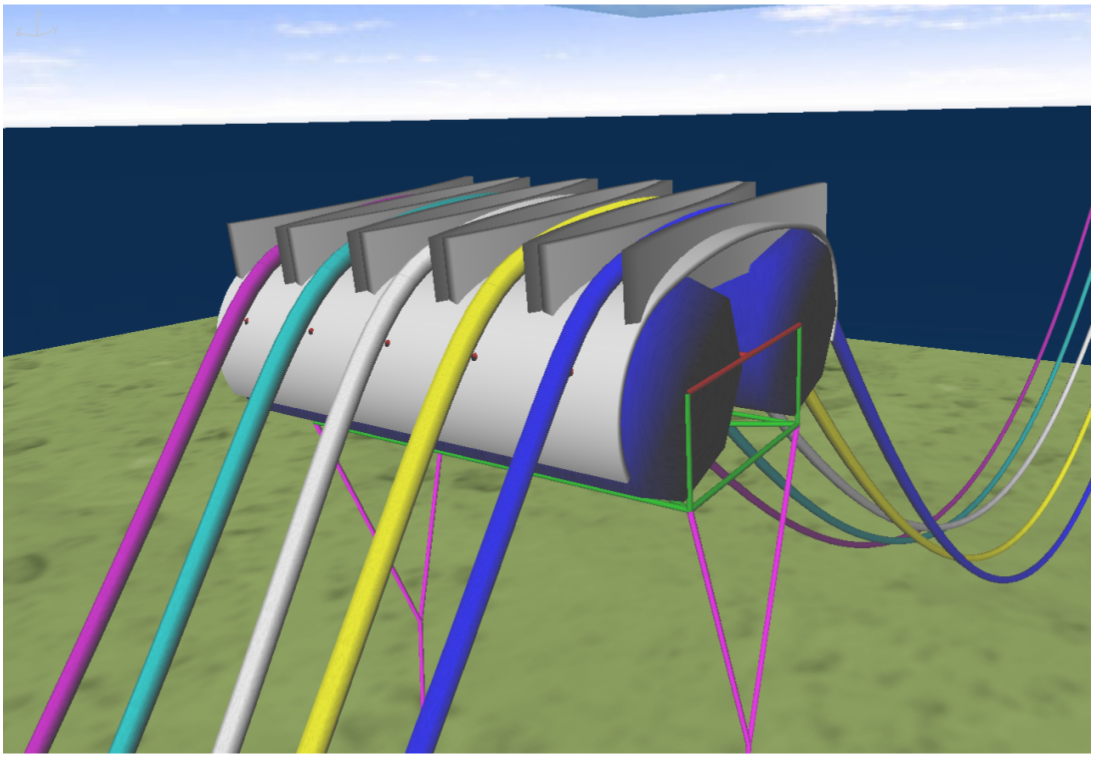
This example illustrates a number of important modelling features in Flexcom, specifically:

- Modelling of mid-water arch systems.
- Curved contact surface modelling.

The multiple riser system is shown in the first figure below, while a more detailed view of the mid-water arch is shown in the second figure.



Multiple Riser System



Mid-water Arch Detail

Model Summary

This section contains information on:

GENERAL

This example demonstrates the analysis of a multiple riser system in shallow water. The riser system consists of three production risers and two water injection risers, all of which are connected to the FPSO which is aligned with the risers. The system is sited in 85m of water and each riser passes over the subsea mid-water arch. The production risers are 260m long and are filled with oil. The water injection risers are 1m longer in the upper catenary section and are filled with seawater.

The analysis is performed in three stages. The static configuration of the system, subjected to gravity and buoyancy loading, is established in the initial static analysis where the mid-water arch is held in position using boundary conditions. A subsequent static restart analysis is performed to release the arch by removing these boundary conditions. Finally, the dynamic analysis finds the response of the system to regular wave loading.

MID-WATER ARCH

The mid-water arch model is composed of a framework of beam elements connected to a series of curved guide surfaces and two buoyancy tanks, all of which are tethered to the seabed. The model is highly parameterised using the `*PARAMETERS` keyword. The arch is modelled as a function of several parameters, the riser diameters, the midwater arch frame width, curved guide radius and subtended angle.

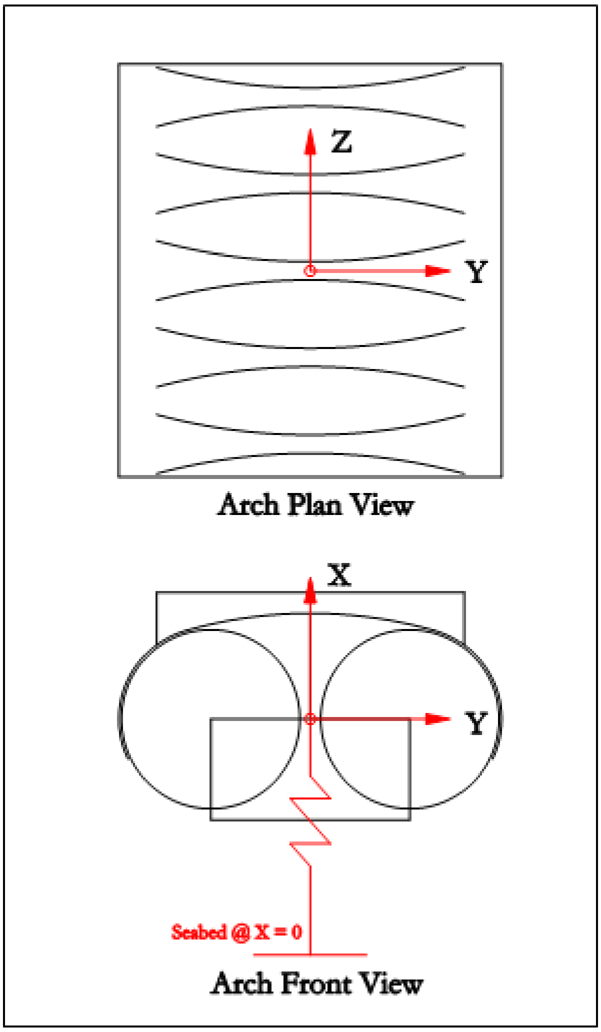
The riser diameters are stored in the parameters L1_Do to L5_Do. Changing the value of the outer diameter will change dependant parameters such as the length of the curved guide surface under that riser as well as the gutter gap and its curvature.

The frame width is defined as the distance between the two buoyancy tanks (centre-to-centre) and is stored in the parameter FrameWidth. Changing the value of this parameter will change the overall shape of the arch and the associated top guide surfaces as well as the gutter wall heights and lengths.

The side guide surfaces are defined by two parameters, the radius and subtended angle, stored in the parameters SideGuideRadius and SideGuideSubtAng respectively. Again, the shape of the arch can be affected by changing these parameters.

Also parameterised is the tether height and bridle length to facilitate easy adjustment of the arch in the vertical direction.

The mid-water arch is built around the origin of the global axis system, such that the axes are located as shown in the figure below. There are three curved guide surfaces supporting each riser, one top and two side curved guide surfaces, one vessel side and one well side. The curved guide surfaces, defined using the *GUIDE keyword and TYPE=CYLINDRICAL option, are connected to the frame by beam elements and driver nodes. The risers are defined in three sections, the lower and upper catenary sections (defined using the *LINES keyword) and the central section of riser which passes over the arch (defined using individual nodes and elements). The arch section nodes are parameterised so as to follow the curved arch, a distance of the riser radius above the arch, so as to facilitate ease of initial static solution. The riser is connected to the mid-water arch with a hinge element representing the friction clamp which is used to keep risers in place. It is also possible to remove this hinge element to allow the riser to move freely over the arch to model a situation where the friction clamp may have failed. The sections of the catenary lines which can potentially come into contact with the arch are prescribed a finer mesh to accurately capture contact on the arch region. The lower catenary risers are supported on an elastic seabed while the upper portions of the risers are connected to the FPSO.



Global Axis System in relation to Midwater Arch

The hydrodynamic properties of the mid-water arch are very different in the vertical and horizontal directions, therefore the arch hydrodynamic coefficients are normally specified separately for each of three directions (one vertical, two horizontal). Furthermore, for each direction the data is often specified as three products, namely:

i. CdAd (drag coefficient by frontal area) for the hydrodynamic loading due to relative fluid/arch velocity.

ii. CmVin (inertia coefficient by reference volume) for the hydrodynamic loading due to arch acceleration.

iii. CaVin (added mass coefficient by reference volume) for the hydrodynamic loading due to water particle acceleration.

This means a mid-water arch may be characterised by nine hydrodynamic coefficients or products. Flexcom transforms the arch data specified in terms of these nine coefficients into equivalent hydrodynamic properties for each element of the arch structure set, and the analysis then proceeds in the usual way. In this way phase differences between wave forces on different parts of the arch, and also the changes in the arch hydrodynamic properties as the arch displaces and rotates, are very accurately modelled.

Analyses

STATIC ANALYSES

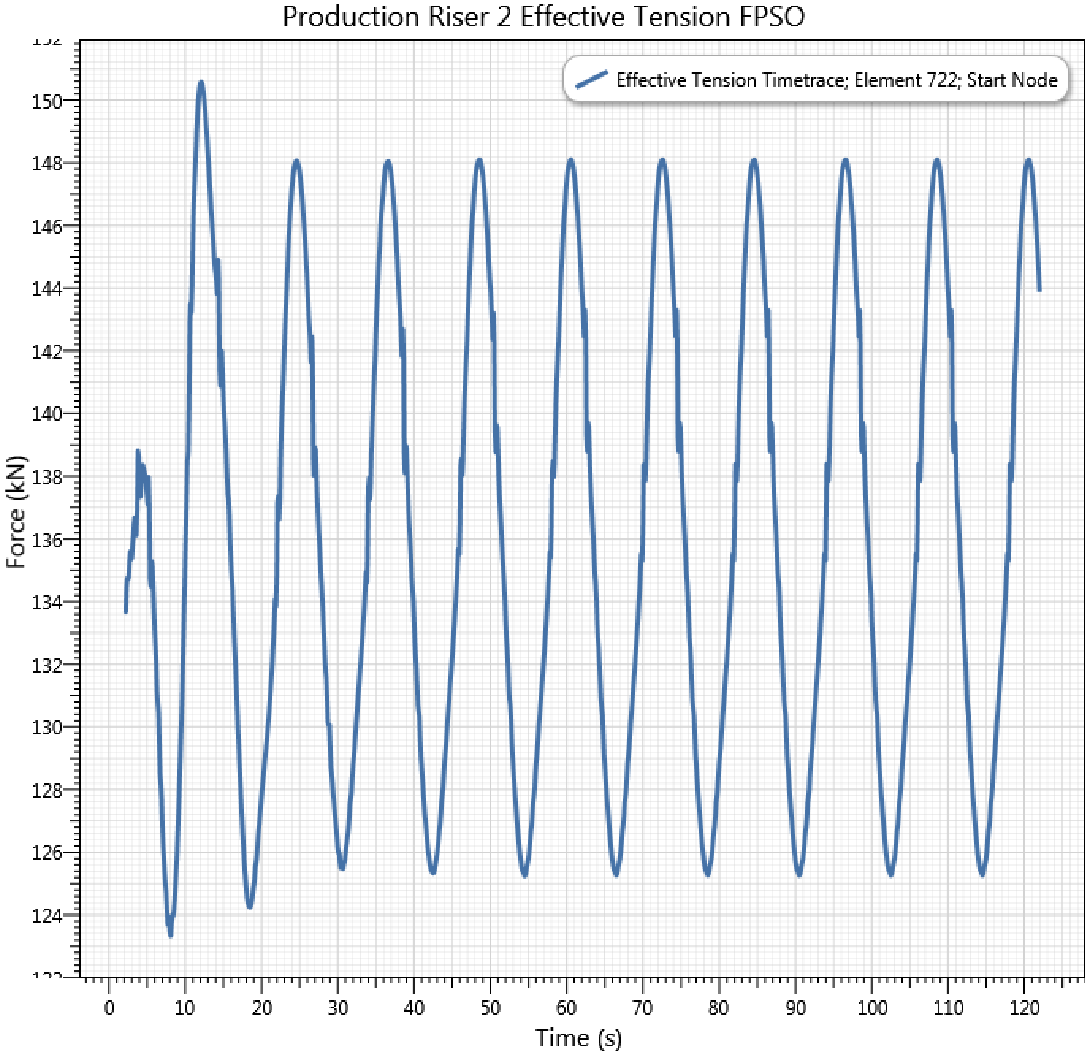
The base of each riser and tether is fixed in all translational degrees of freedom. The upper ends of the risers are attached to the vessel. The initial position of the vessel and the RAOs are also specified. The mid-water arch is fixed in all translational degrees of freedom at four corners in the initial static analysis, to allow the risers to solve statically. The arch is then released in a subsequent restart static analysis to allow the entire system to attain static equilibrium.

DYNAMIC ANALYSIS

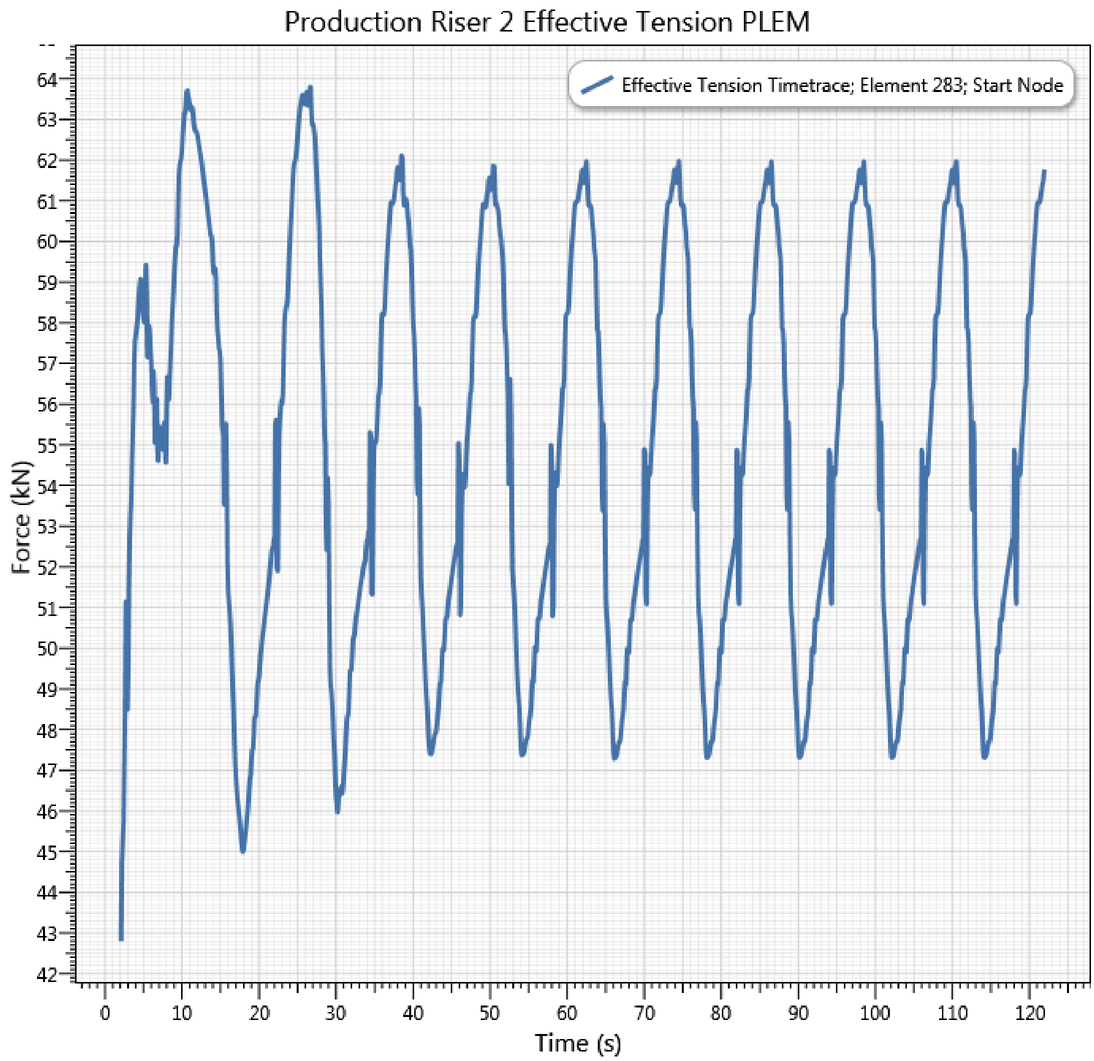
In this dynamic stage, a regular wave is applied to the structure. The wave amplitude is 8 metres, the period is 10 seconds and the wave direction is 45° anti-clockwise from the global Y-axis. The boundary conditions remain unchanged and are carried through automatically from the previous analysis. A variable time step is used to allow the software to use an optimal time step – this will vary over the course of analysis as the risers interact with the arch structure under the influence of wave loading.

Results

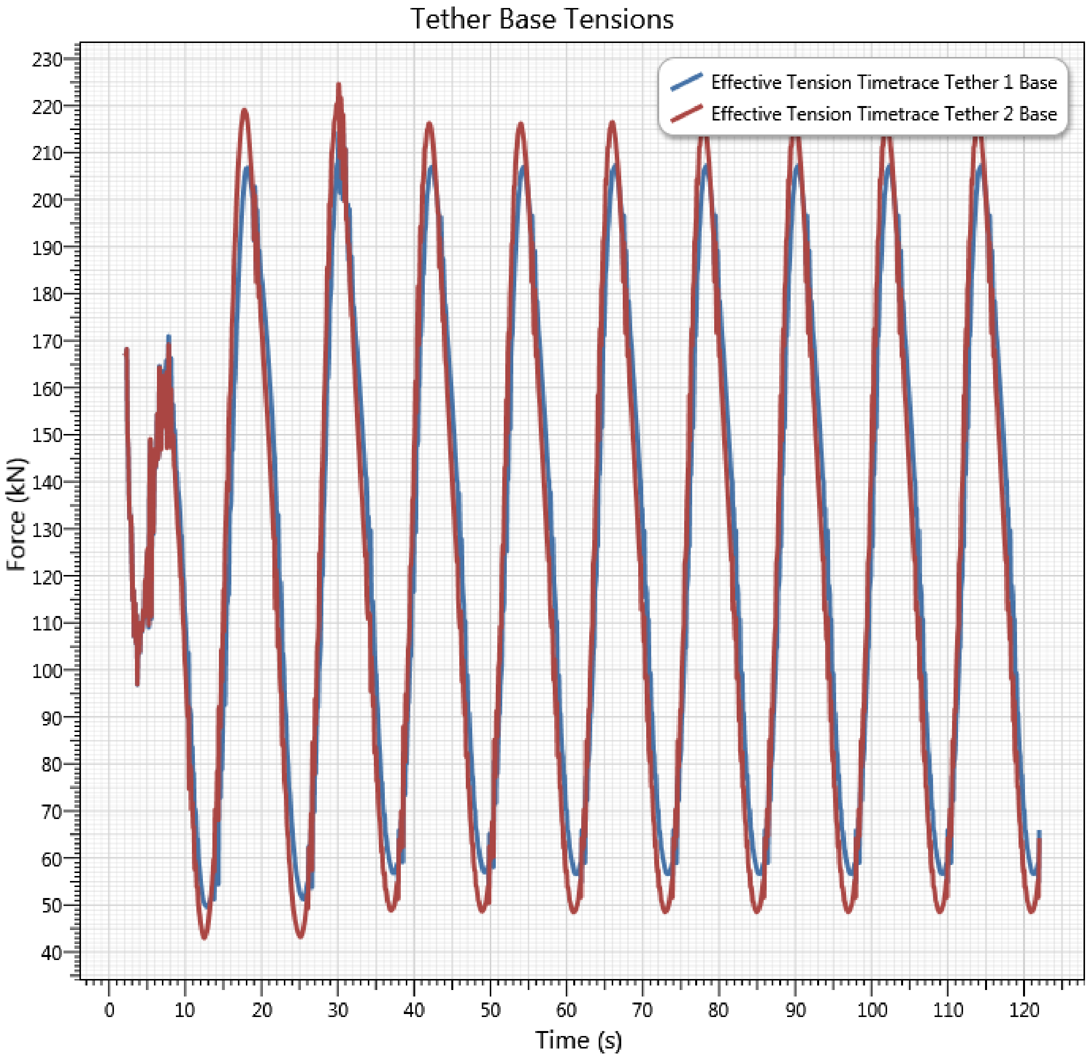
The following figures contain a selection of results from the dynamic analysis. The first two figures respectively show time traces of the effective tension at the vessel connection and pipeline end manifold (PLEM) for Production Riser 2. The graphs of tension in the riser show a periodic variance as the vessel is subject to regular wave loading. The isolation effects of the mid-water arch on the riser system can be seen in the reduced tension at the PLEM.



Riser 2 Effective Tension at Vessel



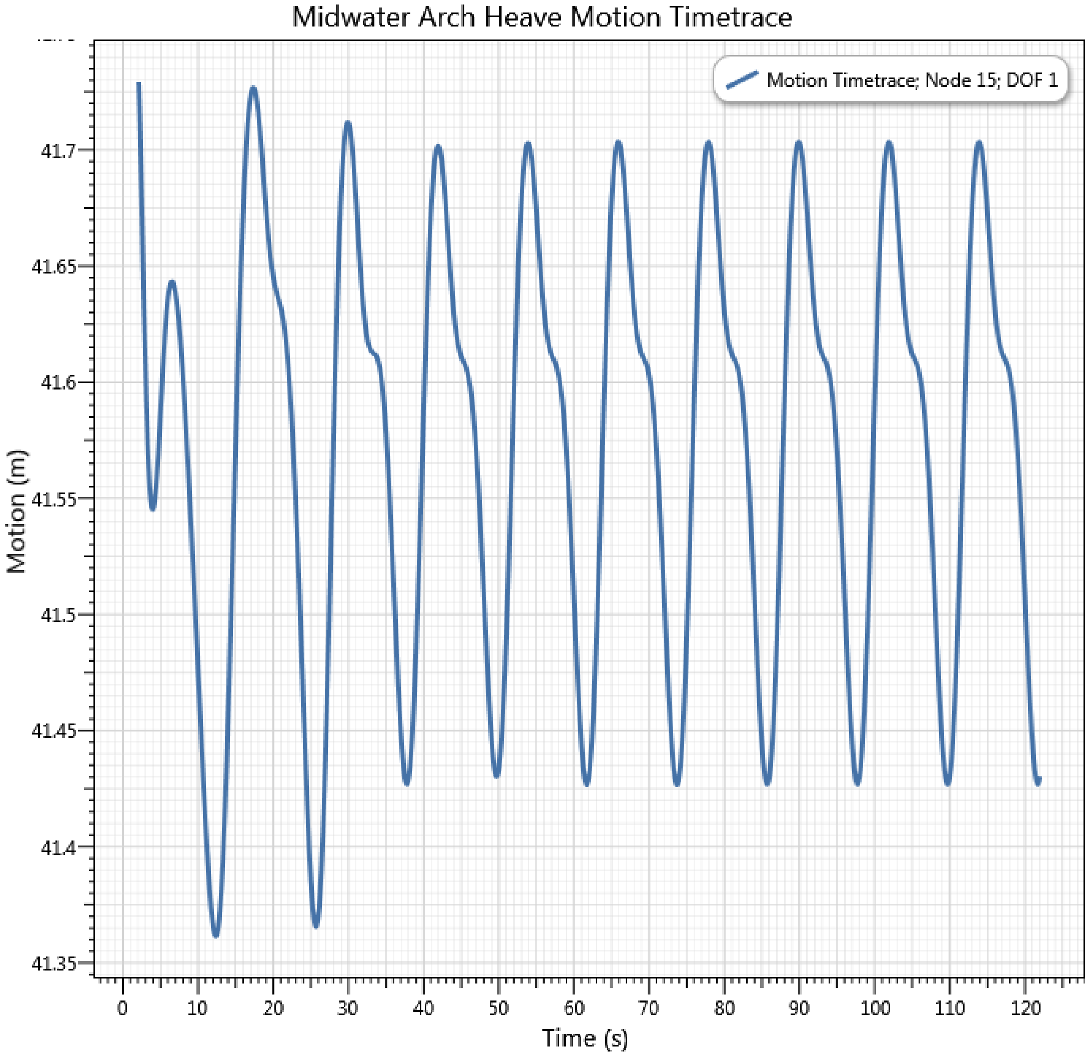
Riser 2 Effective Tension at PLEM



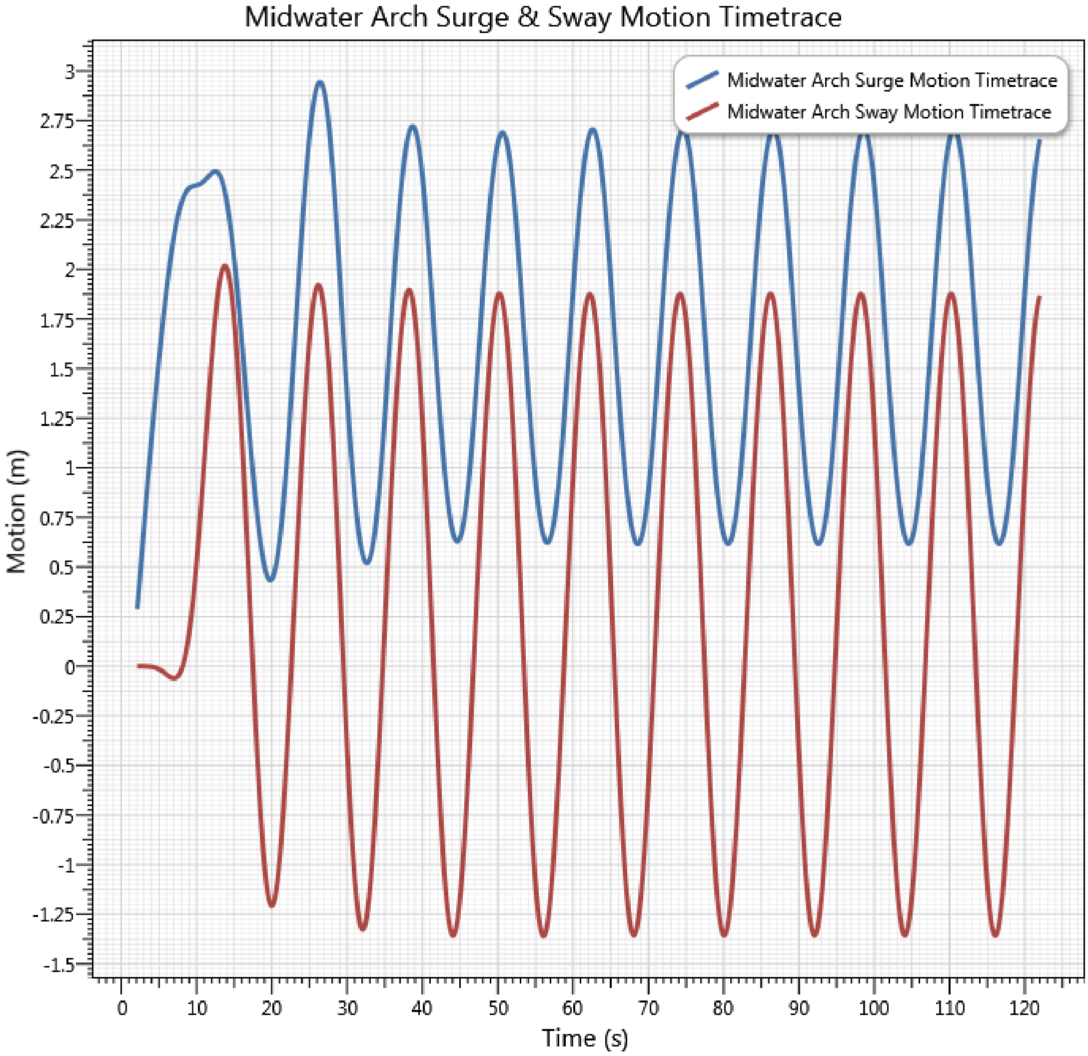
Effective Tension in Midwater Arch Tethers

The first figure below shows the tension in the mid-water arch tethers which again varies periodically due to wave loading. The free floating motion of the arch causes differences between the tether responses.

The final two figures show the heave, surge and sway motions of the mid-water arch. The arch motions while supporting the risers, allows vessel movement under wave loading to be minimised, reducing the loading effects on the risers in the touchdown zone and protecting the risers' bend radii.



Arch Heave Motion



Arch Surge & Sway Motion

1.10.3.3 C03 - Turret Disconnect

This section describes a sample turret moored vessel and demonstrates how the turret disconnect is readily facilitated using Flexcom. Equations and parameters are also used throughout the model to illustrate how such a model setup can allow for large changes to the design/model with minimal effort.

The overall layout of this example is as follows:

- Introduction gives an overview of the spider buoy configuration.
- Model Summary describes the model setup in more detail.

- Analyses briefly describes the various analyses performed.
- Results presents pertinent results.

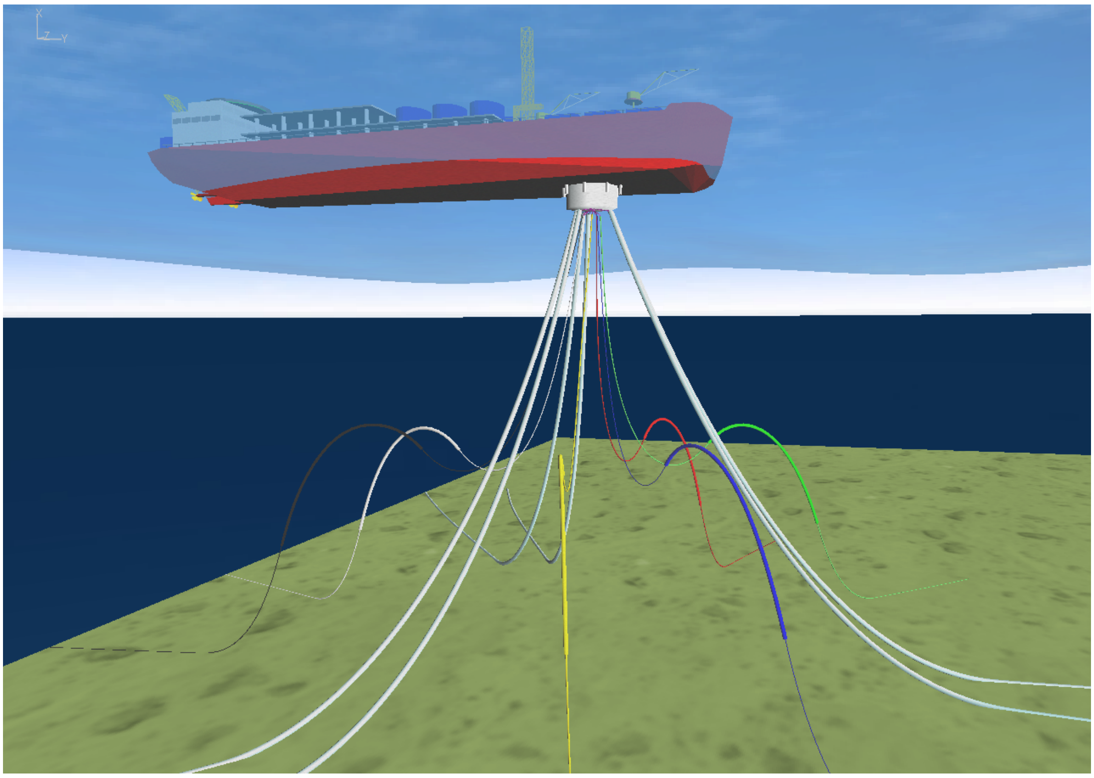
Introduction

This example considers an internal turret moored FPSO and examines a turret disconnect scenario. The turret, integrated towards the bow of the vessel, allows to vessel to weathervane into the prevailing environment. Additionally, the turret is disconnectable as this design allows the vessel to disconnect from the turret in inclement weather, such as an approaching iceberg, or for service of the FPSO. This model includes the vessel, turret cage, and buoy, which is comprised of 6 chain mooring lines, 2 production risers, 2 injection risers, an umbilical, and 1 gas lift line.

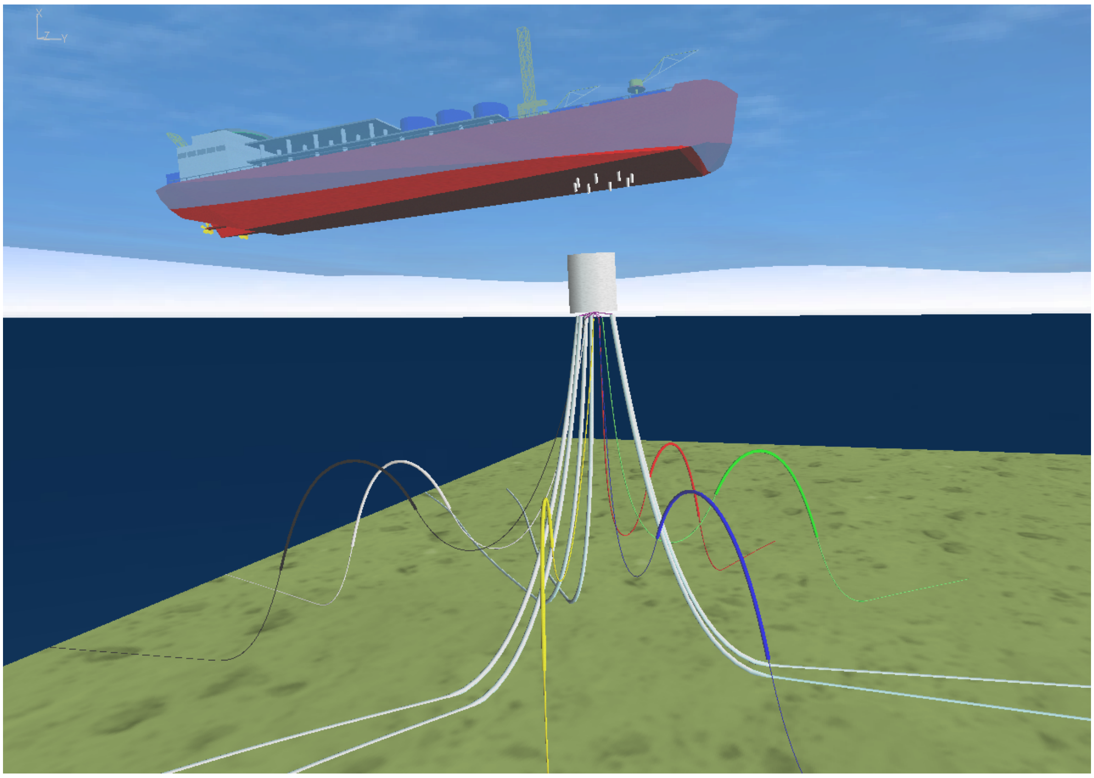
The analysis is performed in two phases. In the first phase, the buoy is connected to the vessel and subjected to wave and current loading. In the second phase of the analysis, the buoy is released from the vessel and is allowed to fall out of the moonpool of the vessel as it continues to be subjected to wave and current loading.

This example illustrates several features in Flexcom which allow for the ease of model set up and show how investing time in the initial set up of parameters can lead to significant time saving changes during any future alterations to the model. Additionally, such a model setup is also very conducive to quality assurance.

The connected and disconnect configurations are shown in the respective figures below.



Spider Buoy Connected Configuration



Spider Buoy Disconnected Configuration

Model Summary

GENERAL

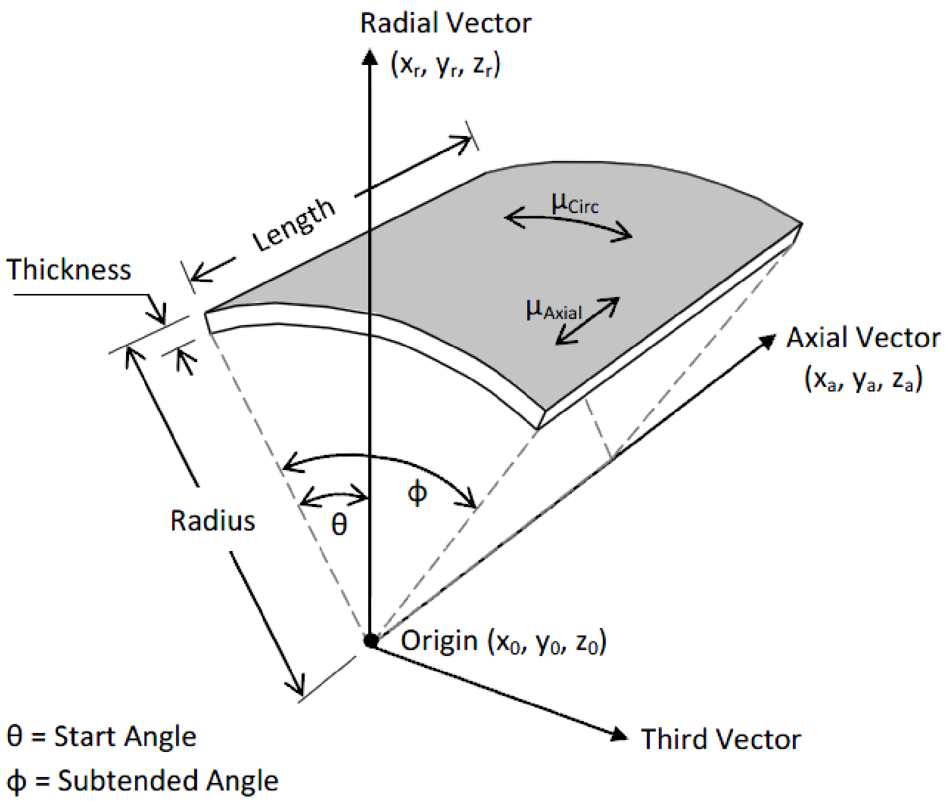
This example considers an internal turret moored FPSO in 150m of water. The internal turret is modelled as a large cylindrical structure, fixed to the vessel and positioned within the cylindrical moonpool. The turret connects 5 risers and 1 umbilical, in a lazy wave configuration, to the FPSO. Contact with the moonpool frame is also modelled via eight perpendicular, cylindrical guide surfaces positioned around the turret.

MOORING

The turret is moored with a 6 leg chain mooring configuration arranged in 3 bundles of 2 legs each. The two adjacent legs in each bundle are separated by 10°, with 120° between each bundle axis. The mooring spread is 300m and each leg has a length of 370m. This is done easily in Flexcom by defining angles in the *PARAMETERS command in terms of the mooring spread radius and angles. The connection at the buoy is modelled with a hinge element in order to avoid the transfer of any moment or torsion into the buoy.

MOONPOOL FRAME

The turret frame is modelled with eight cylindrical guide surfaces arranged symmetrically. The guide surfaces are defined in a similar fashion as the mooring lines in terms of a radius and radial coordinates. From here, the guide surfaces are defined using the *GUIDE keyword using an origin, and two perpendicular axial and radial vectors. This is shown in the figure below. For the ease of defining the radial vector, a subtended angle of 360 has been used.



Cylindrical Guide Surface Definitions

TURRET BODY

The turret body is modelled as large diameter, stiff beam elements, with stiff connection elements connecting the risers and moorings to the buoy. Mass and inertia properties are defined using the *FLOATING BODY keyword. Additionally, in order to appropriately model the tangential hydrodynamics and mass distribution, the buoy has been divided into an upper, middle and lower section element sets.

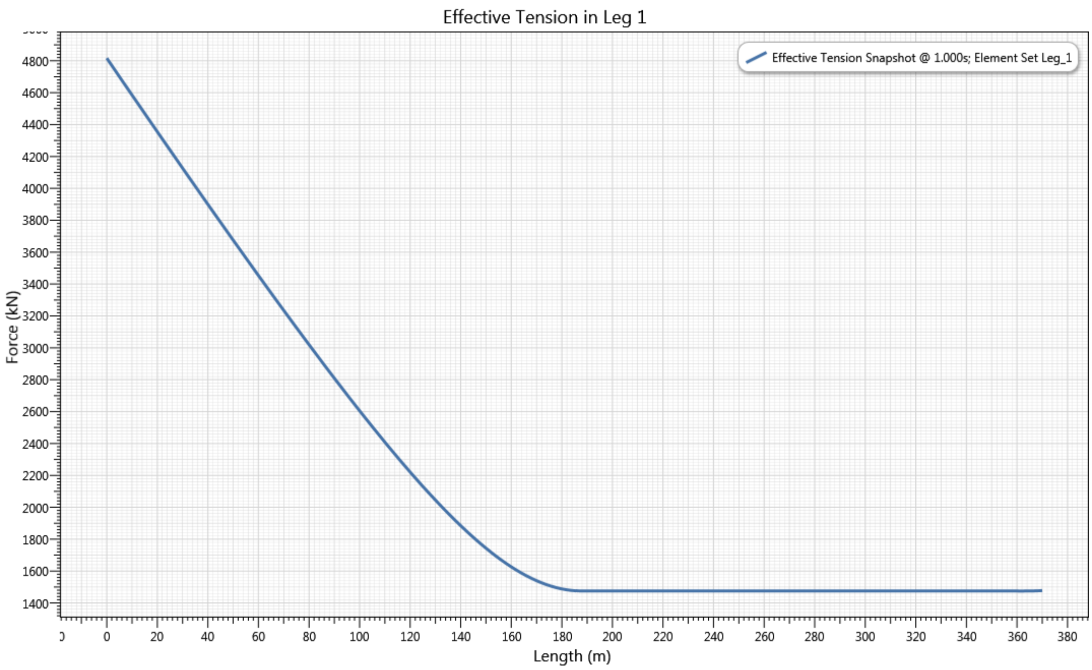
Analyses

The analysis is broken into 4 different stages:

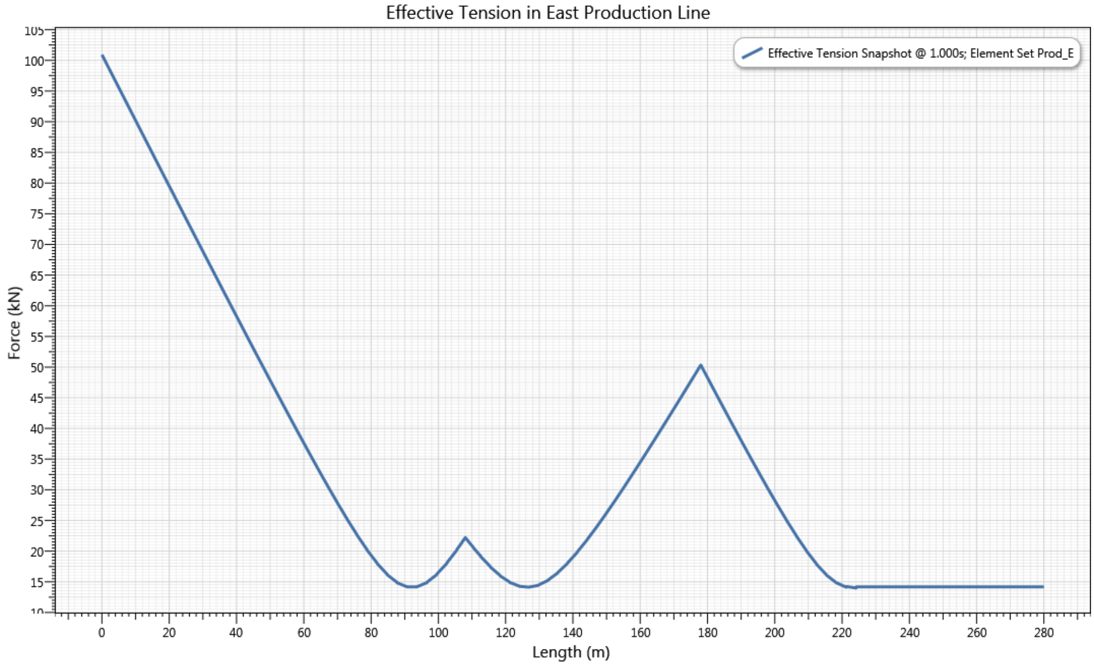
- Static Analysis – In the initial static stage, the lines are allowed to assume their static configurations. Static tensions for the mooring lines, and curvatures for the risers/umbilical are checked to ensure the static configuration is reasonable.
- Current Analysis – In this stage, the current is applied to the model.
- Connected Dynamic Analysis – Next, the connected turret is subjected to combined current, waves and vessel motion. Buoy motions, effective tensions of the lines, and curvatures envelopes are evaluated..
- Disconnect Dynamic Analysis – Finally, the boundary condition linking the turret to the vessel is released and the buoy is allowed to drop out of the moonpool. Contact with the moonpool frame, buoy motions, envelopes of effective tension in the moorings and risers, and envelopes of curvatures are also evaluated.

Results

The following figures contain a selection of results. The first figure below shows the static effective tension in one of the mooring lines. Effective tension in one of the production lines is shown in the second figure.

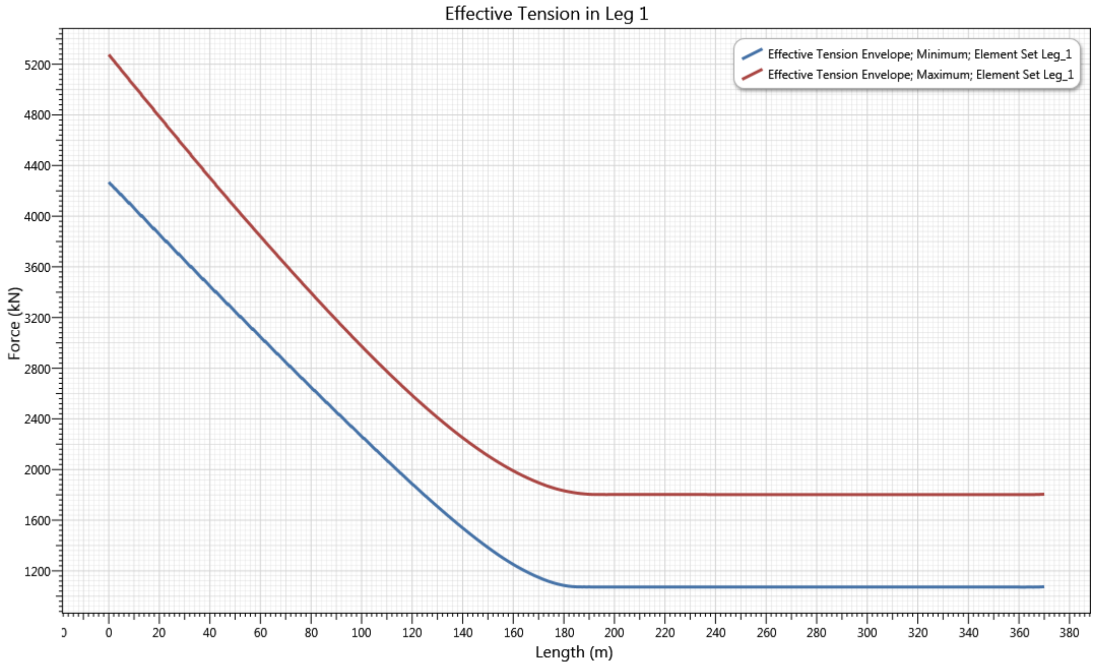


Static Mooring Line Tension

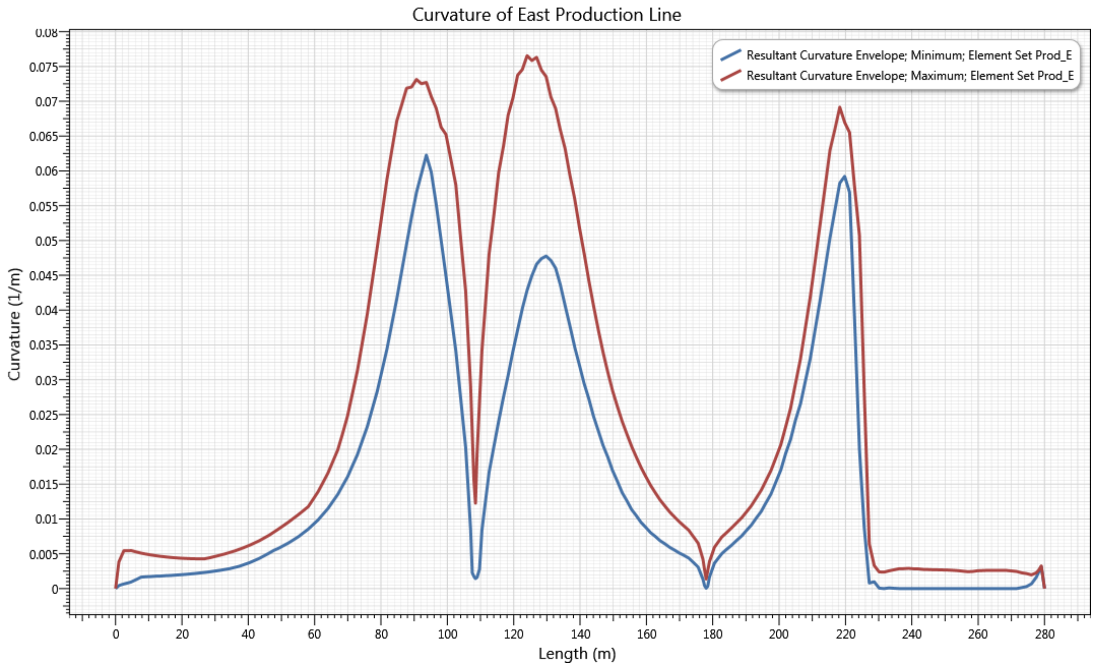


Static Production Line Tension

The first figure below shows the mooring line tension envelope for line 1 the connected buoy. The second figure shows an envelope of curvature for the east production line.



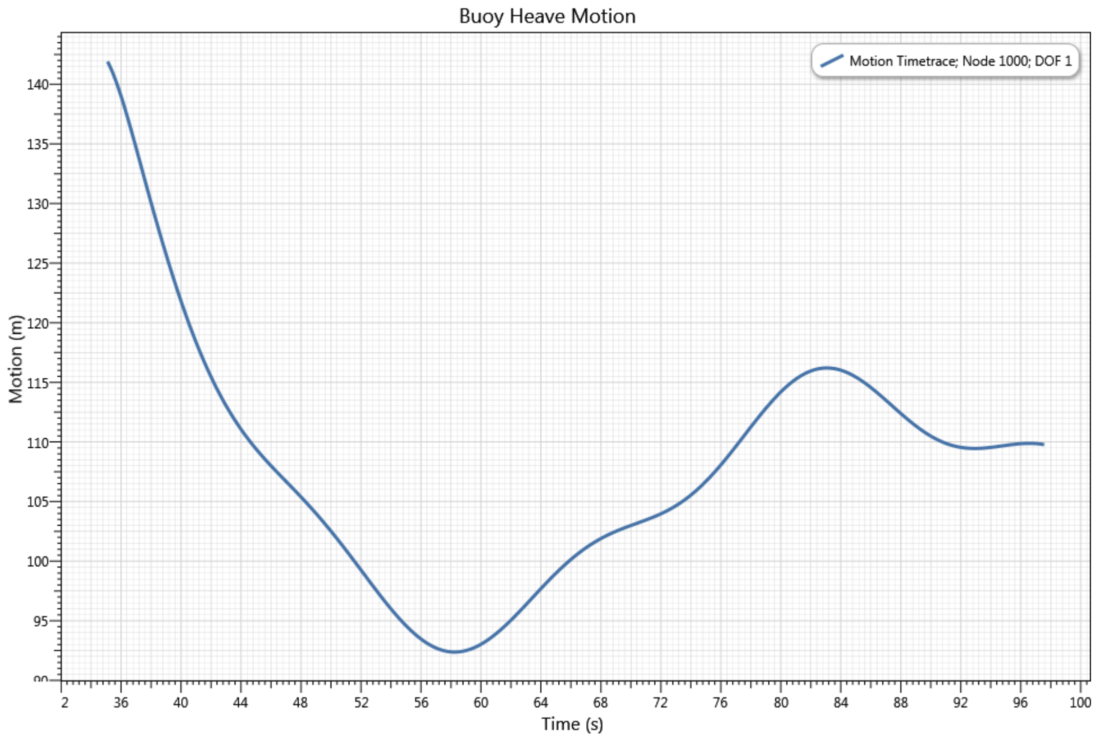
Mooring Line Tension Envelope (Connected)



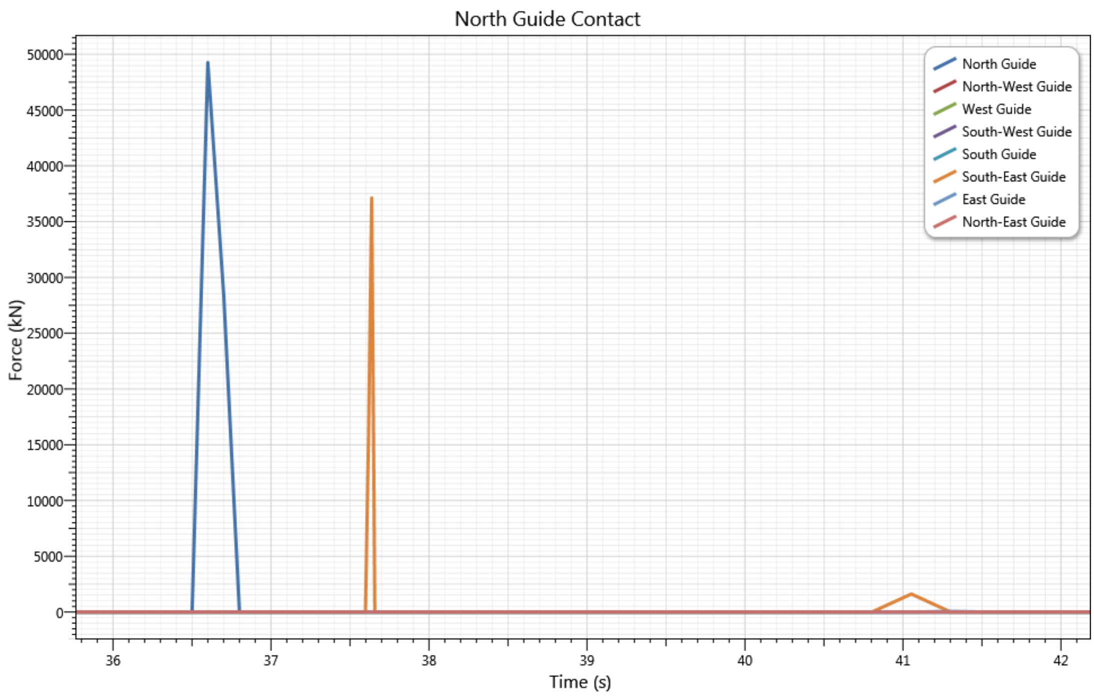
Production Line Curvature Envelope (Connected)

In this disconnect analysis, several parameters are monitored to ensure the adequacy of the model. The vertical motion of the buoy is one of the easiest parameters to check to ensure satisfactory behaviour. Additionally, any contact forces between the buoy and the moonpool are examined. These results are presented in the first and second figures below.

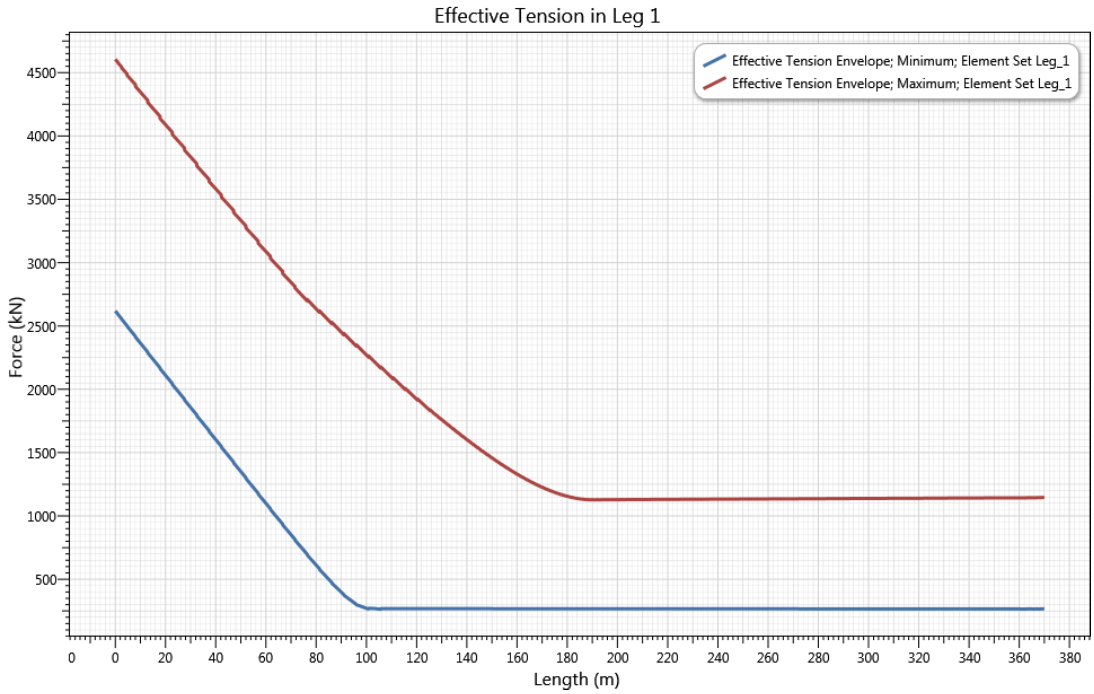
Once the engineer is satisfied these parameters are consistent with expectations, the system is then checked with respect to mooring line tensions and riser curvatures. Sample results are presented in the final two figures.



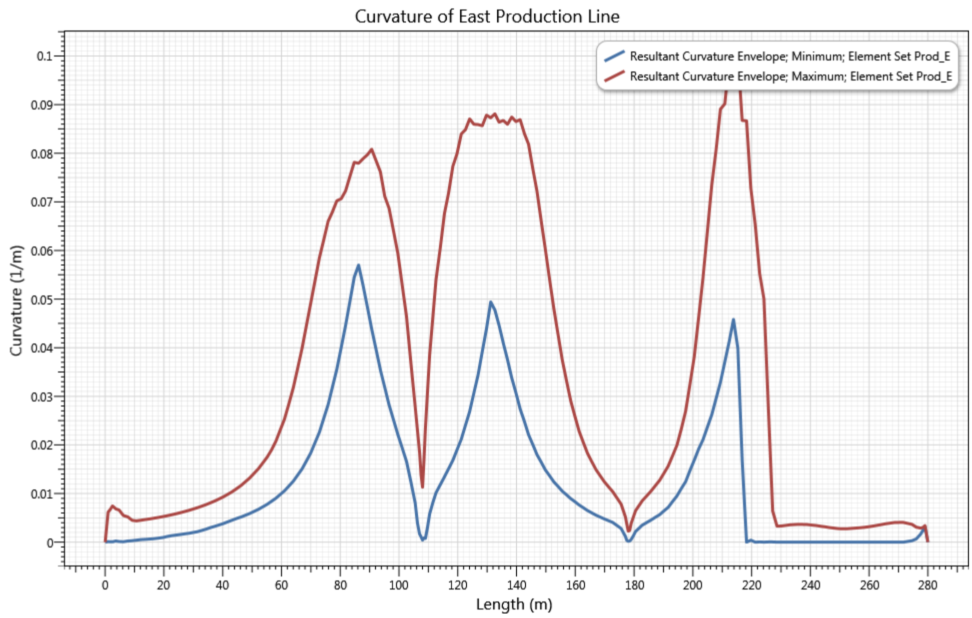
Vertical Motion of the Buoy



Moonpool Contact Forces



Mooring Line Tension Envelope (Disconnected)



Production Line Curvature Envelope (Disconnected)

1.10.4 D - Mooring Systems

Section D contains an example of a mooring system, including:

- D01 - Moored Vessel

1.10.4.1 D01 - Moored Vessel

This example describes the analysis of a moored vessel, and demonstrates the use of Flexcom in analysing mooring systems, with particular regard to determining vessel second order motions due to drift forces, wind and current. The overall layout of this example is as follows:

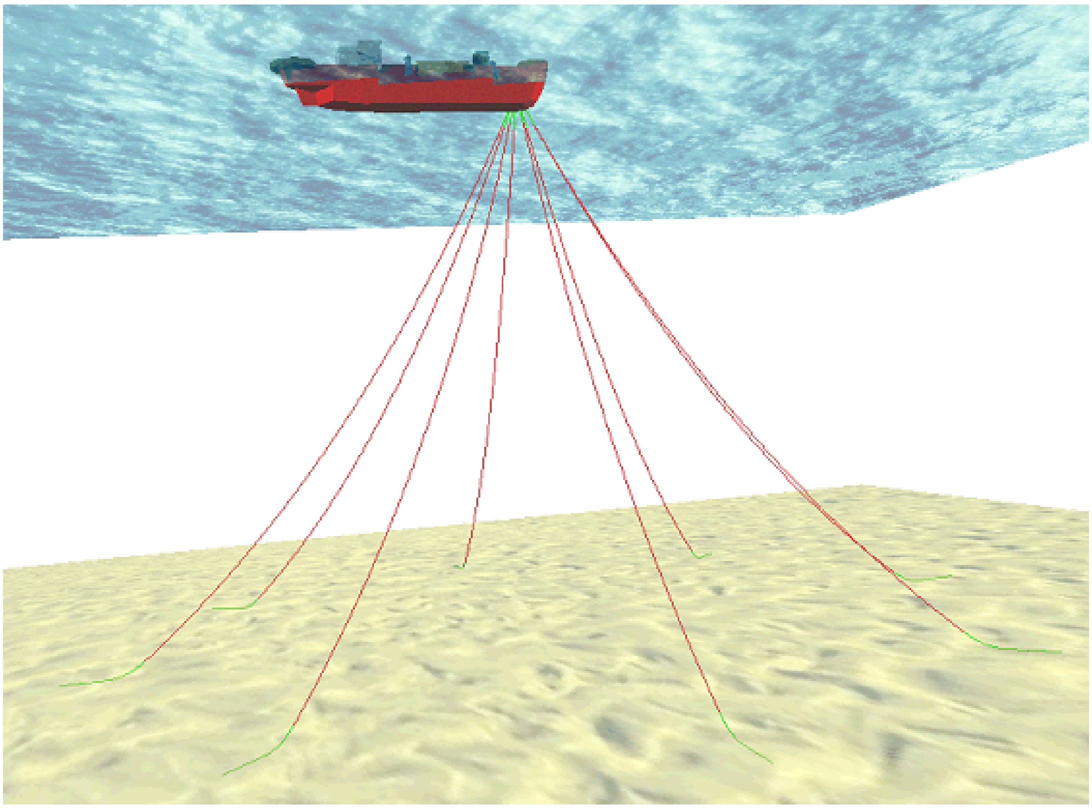
- Introduction gives an overview of the moored vessel analysis, and notes some of the more important features of Flexcom which are relevant to the analysis.
- Model Summary describes the model in more detail.
- Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions to which the vessel and mooring system is subjected.
- Results presents pertinent results from the various analyses performed and discusses their significance.

Introduction

This example considers the analysis of an eight-line mooring system. The analysis is performed in two phases. In the first, the moored vessel is explicitly included in the model, and is subjected to wind, wave and current loads in order to calculate the second order vessel motions in surge, sway and yaw. In the second phase, the computed second order motions are combined with high frequency vessel motions to simulate the combined vessel response. A reduced model is analysed in this second phase; the moored vessel is no longer explicitly included, and the response of only the most critical mooring line – that is, the one experiencing the greatest tension in the first phase - is examined.

This example illustrates the use of Flexcom in analysing mooring systems, with particular reference to the program ‘moored vessel’ capability. A detailed discussion of this facility is provided in Coupled Analysis.

The vessel and mooring system is shown in the below figure.

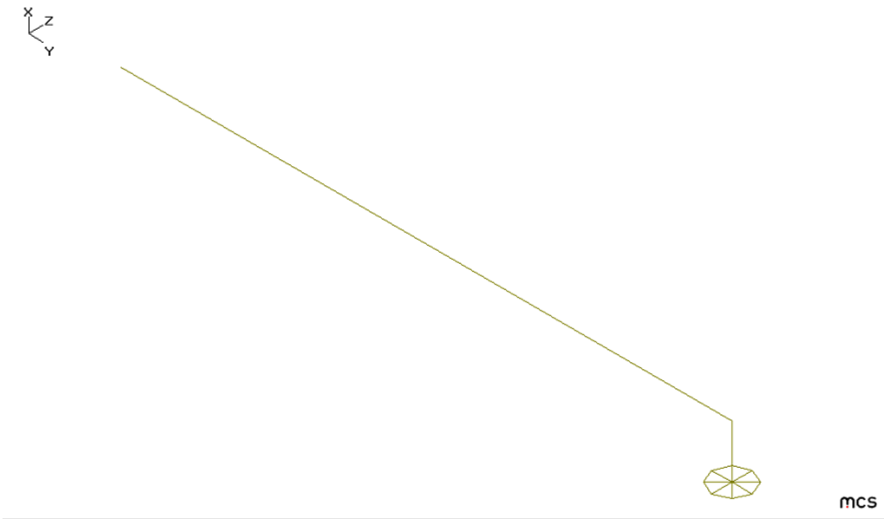


Vessel and Mooring System

Model Summary

PHASE 1

The moored vessel is approximately 340m in length, with a draft of 21m, and is situated in a water depth of 810m. The mooring system consists of eight lines connected to a turret. The angular spread between the lines is 45º. Each line is 1100m in length, of which 160m is chain of diameter 152mm and the remainder is wire of diameter 89mm. Mooring line pre-tension of each line is about 750kN. The anchoring radius from the turret centreline is 700m, with each fairlead 8m below the MWL. The horizontal distance between the turret centreline and the vessel CoG is 150m. In the finite element model of this system, the vessel is modelled using an assemblage of 19 beam elements; a snapshot of this arrangement is shown in the figure below. Of the 19 elements, 16 model the turret “carousel” to which the mooring lines are attached via hinges. There is 1 vertical element, and this is connected via a linear flex joint to the final, horizontal, element, which connects the turret to the CoG.



Schematic of Vessel and Turret

The only constraints applied to the mooring system are those at the mooring line anchor points; there are no other prescribed motions applied to the vessel or mooring lines. The vessel is subjected to wind, current and low frequency second order wave loads in order to identify the vessel response in surge, sway and yaw. These motions are subsequently combined with first order motions due to wave loading to simulate the combined vessel response.

PHASE 2

In this analysis only one mooring line is considered, the most loaded line from Phase 1. A more detailed model incorporating more elements is employed in this case. The moored vessel is not included in the model, but the vessel motions from Phase 1 are combined with motions calculated from first order RAOs to give the combined vessel response.

Analyses

PHASE 1: DETERMINATION OF SECOND ORDER MOTIONS

Initial Static Analysis

This first analysis is a so-called Static Fixed moored vessel analysis, as described in Coupled Analysis. The moored vessel data is specified, and Node 1 is identified as the vessel centre of gravity (CoG) node. The anchor nodes of each of the mooring lines are fixed in all degrees of freedom. There are no other user-specified restraints, although Flexcom does automatically restrain the CoG in all translational and rotational DOFs. Wind and current coefficients and vessel QTFs are also input. No RAO data is specified.

Wind and Current Analysis

Collinear wind and current loading is applied to the model. These are directed along the negative vessel surge axis. All boundary conditions (BCs) are unchanged and carry through automatically from the initial static analysis. In the terminology of Coupled Analysis this would in many cases be a Static Mooring analysis, but the vessel and excitation under consideration here necessitate that the static response be found dynamically (or quasi-statically). So a Dynamic Mooring analysis is specified, over a relatively long simulation time (20000s) but with a large time-step (1000s), so that the static equilibrium position is very quickly achieved.

Dynamic Analysis

A random sea characterised by a Pierson-Moskowitz wave spectrum, with a H_s of 5m and a T_z of 7s, is specified. BCs are unchanged and carry through automatically from the current analysis. A 3-hour simulation is performed with a relatively large time-step of 2.5s – the vessel response rather than the response of the mooring system is the primary output of interest. The vessel response in all translational and rotational degrees of freedom is in fact automatically output to a file with the extension .mor for subsequent use in the Phase 2 dynamic analysis.

PHASE 2: ANALYSIS OF MOORING LINE 1 WITH COMBINED VESSEL MOTIONS

Initial Static Analysis

The base node of the mooring line is fixed in all translational degrees of freedom. Vessel BCs are specified at the topmost node. The vessel initial position is specified and the vessel RAOs are input.

Current Analysis

A piecewise linear current is applied to the mooring system. The current is ramped on over ten increments. Boundary conditions are unchanged and carry through from the initial static phase. Note that it is necessary to run this analysis from 1s to 20001 (in steps of 2000s), in order to have the subsequent dynamic analysis start at the same time as the dynamic analysis of Phase 1: otherwise the second order motions from Phase 1 will not be applied.

Dynamic Analysis

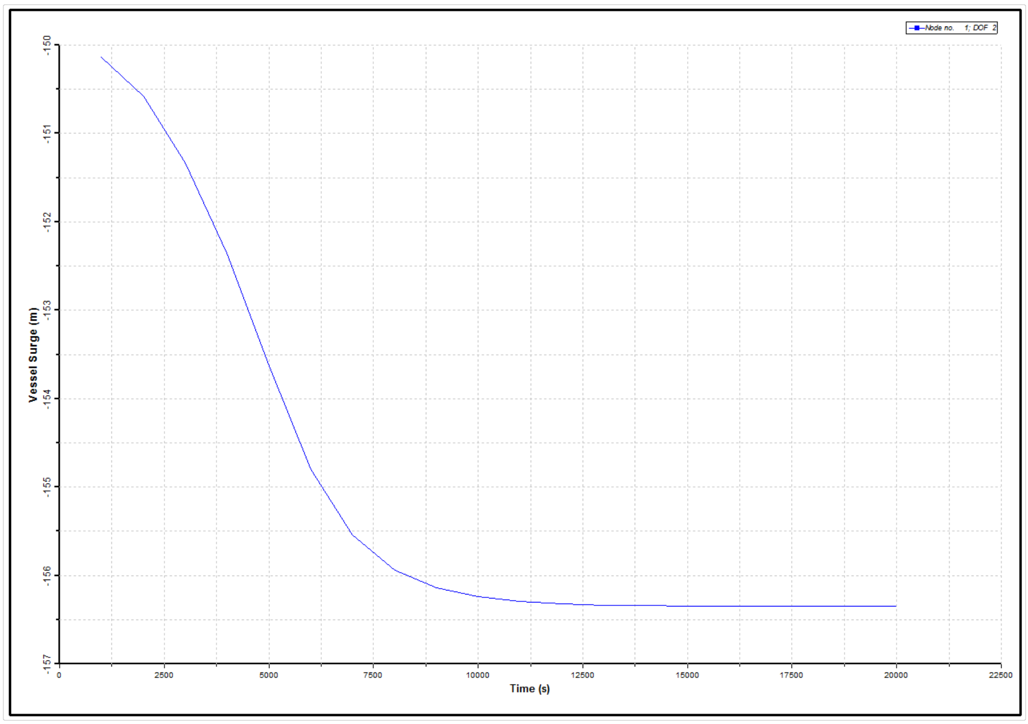
The same wave parameters as were specified in the Phase 1 dynamic analysis are naturally repeated here. Vessel timetrace drift motions are now specified, with the actual timetrace file being the .mor file from the Phase 1 dynamic. Purely for efficiency reasons, the time domain simulation is analysed for 1 hour only, whereas a 3 hour simulation would generally be considered recommended practice.

Note that is not necessary to apply the mean vessel offset from the Phase 1 wind and current analysis as a static offset in Phase 2 prior to this dynamic run. The mean offset will be the starting point of the drift motions in the timetrace file. However these will be ramped on with the wave loading. So if an offset is specified prior to the dynamic run, it will be immediately removed. So it is better not to apply the offset statically, rather just let it be ramped on in this dynamic phase.

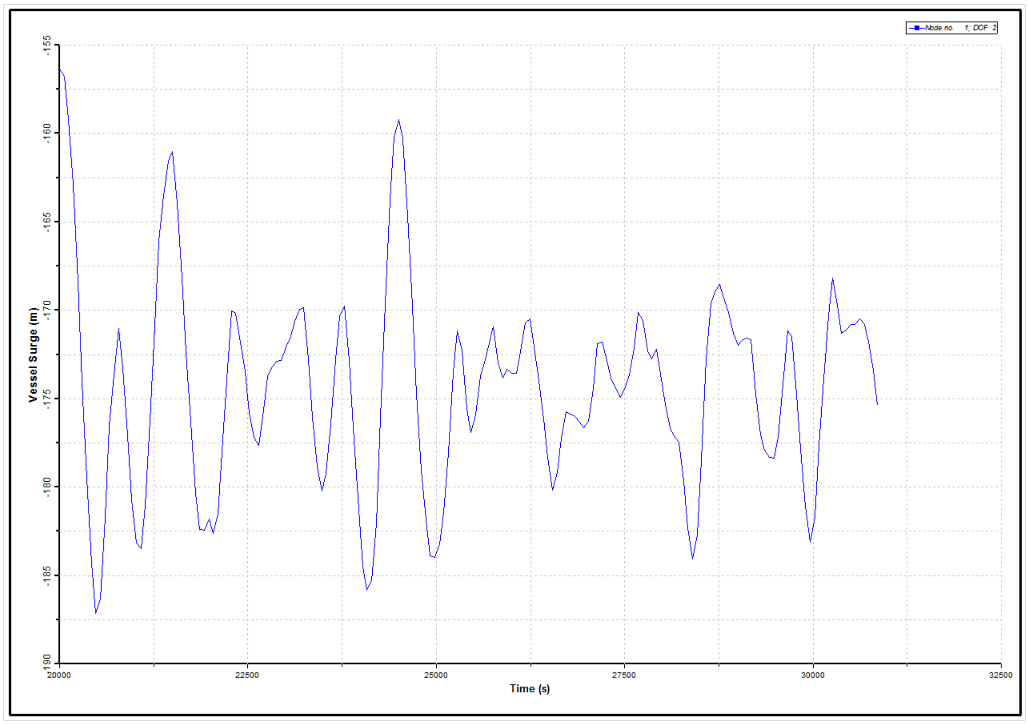
Results

PHASE 1

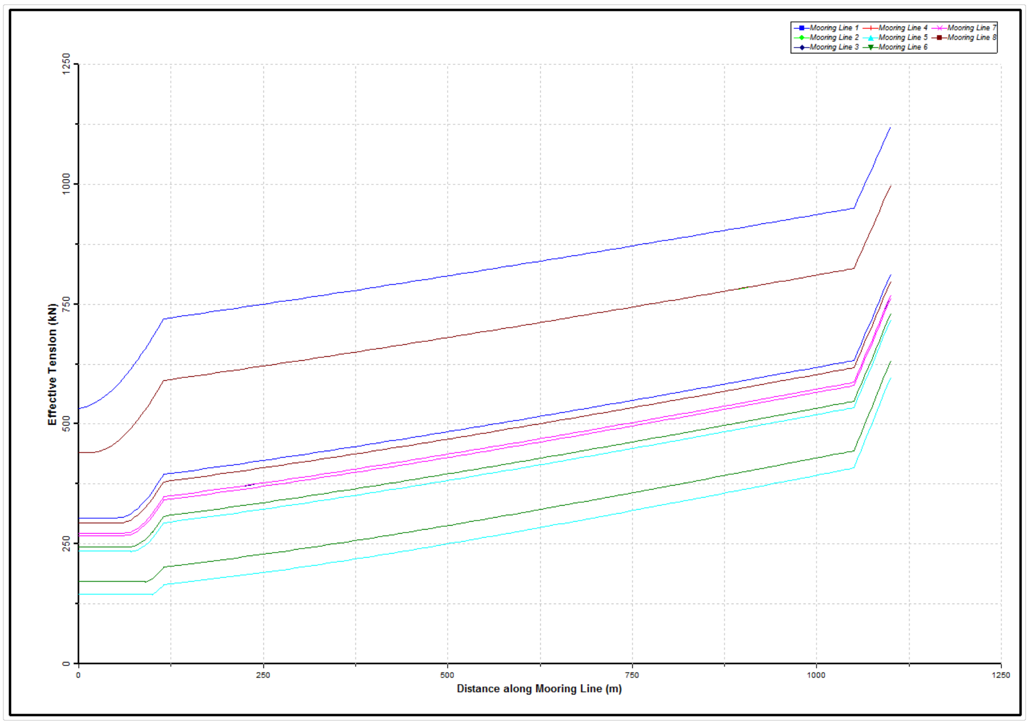
Results from the Phase 1 analyses are presented in the figures below. The Mean Vessel Surge figure shows the mean vessel surge due to wind and current (because of the direction of the environment relative to the vessel, sway and yaw are effectively zero). The final value is (-)6.35m. The Vessel Dynamic Surge figure plots the dynamic vessel response in surge. The Vessel Dynamic Surge figure shows an envelope plot of effective tension in each of the mooring lines, identifying that the maximum tension is experienced by Mooring Line 1. The Tension at Fairlead, Mooring Line 1 figure below shows a timetrace plot of effective tension at the fairlead of this line.



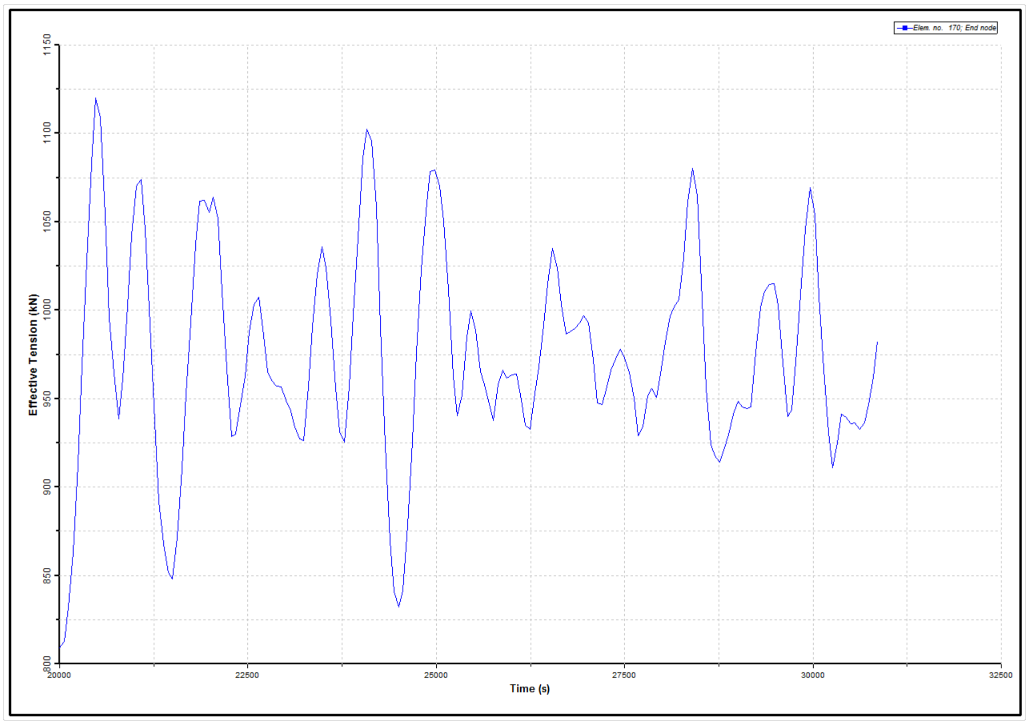
Mean Vessel Surge



Vessel Dynamic Surge



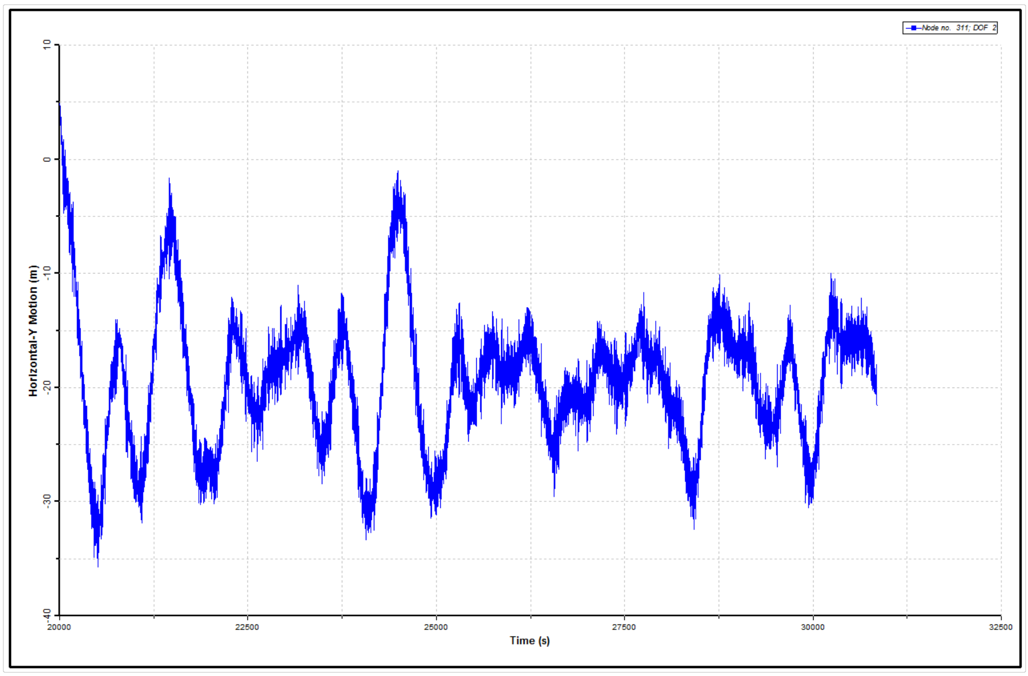
Mooring Line Tension Envelopes



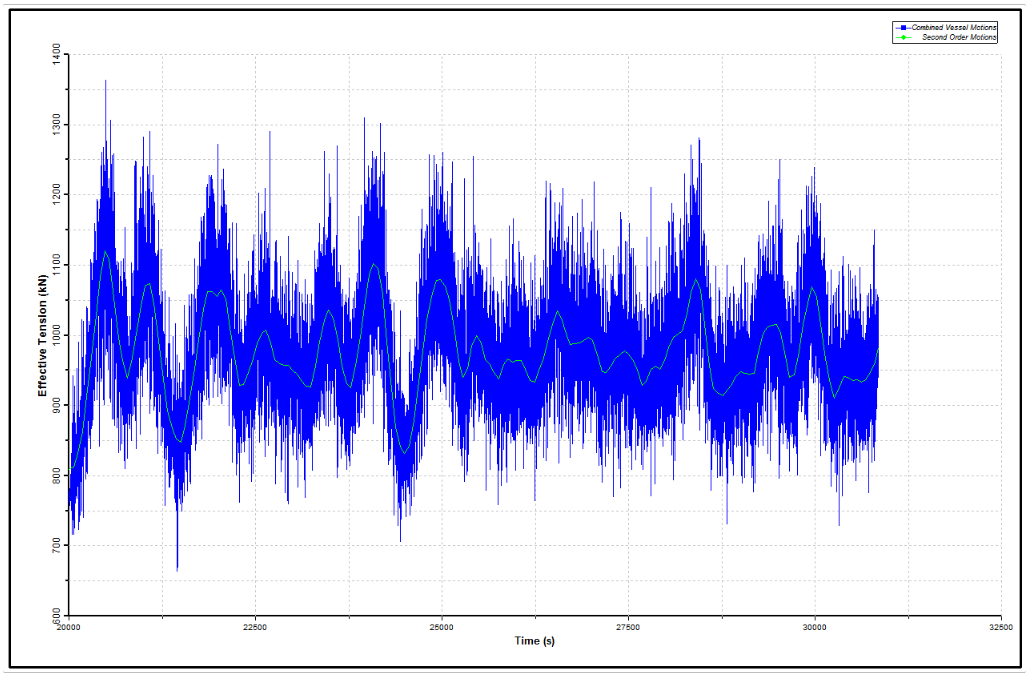
Tension at Fairlead, Mooring Line 1

PHASE 2

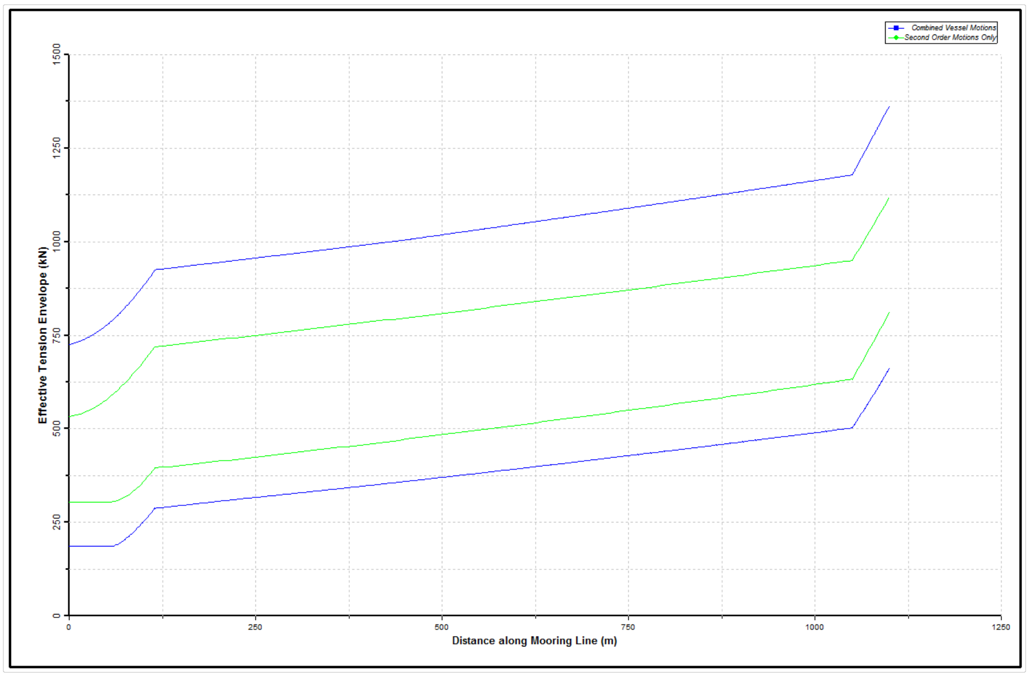
The Dynamic Fairlead Motion figure below shows a timetrace plot of the motion of the mooring line fairlead in the vessel surge direction. This is clearly the Vessel Dynamic Surge figure above with the first order vessel motions superposed. Likewise the 'Dynamic Fairlead Tension' figure below shows a timetrace plot of fairlead tension, with the second order tension superposed for ease of comparison. Finally the Mooring Line Tension Envelopes figure shows envelopes of tension over the full mooring line, with again the curves from the Phase 1 and 2 analyses superposed. The maximum tension shows an almost 22% increase, from 1120kN to 1365kN. Although this is clearly not a design case, the results indicate the importance of having a facility to predict the second order vessel motions, and also a facility to combine first and second order motions in a dynamic mooring line analysis. Both capabilities are provided by Flexcom and make it a robust tool for mooring system design.



Dynamic Fairlead Motion



Dynamic Fairlead Tension



Mooring Line Tension Envelopes

1.10.5 E - Offloading Systems

Section E contains some examples of offloading systems, including:

- E01 - CALM Buoy - Simple
- E02 - CALM Buoy - Complex
- E03 - Floating Hose

1.10.5.1 E01 - CALM Buoy - Simple

This example describes analyses of an offloading system incorporating a catenary anchor leg mooring (CALM) buoy and a steel oil offloading line, and demonstrates a number of features of Flexcom particularly suited to the modelling of these types of systems. The overall layout of the example is as follows:

- Introduction gives an overview of offloading system analysis, and notes some of the more important features of Flexcom relevant to this type analysis.

* Model Summary describes the actual model used here in more detail.
* Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions to which the offloading system is subjected.
* Results presents pertinent results from the various analyses performed and discusses their significance.

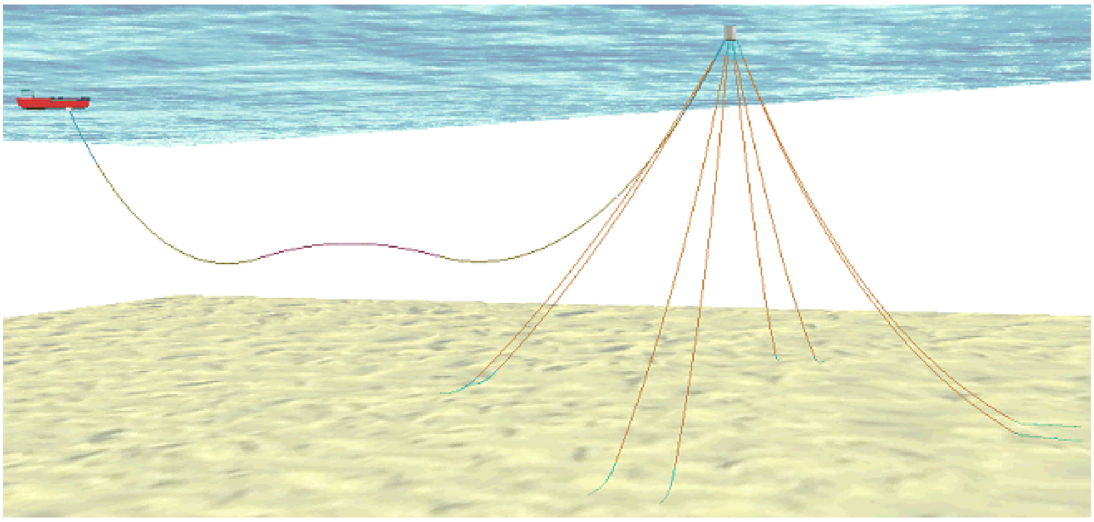
Introduction

This example considers the analysis of an oil offloading facility which includes a steel offloading line and a CALM buoy. Complex systems such as these which are characterised by hydrodynamic and mechanical coupling between components have traditionally been designed by analysing the individual components in isolation. The displacements of the individual floating components, most notably the CALM buoy, would have been calculated from RAOs derived for model tests and/or radiation/diffraction analyses. However, such an approach has the potential to be highly conservative. Typically for example the inertia of the offloading line or lines is of a similar order of magnitude to the CALM buoy, so the presence of these lines affects the buoy response in a way that would not be captured in the derivation of RAOs.

A better solution is to use a coupled analysis approach, and this is facilitated by the Flexcom CALM buoy facility and illustrated in this example. 'Coupled' in this context means that the CALM buoy is explicitly included in the model, and its motions are calculated from forces and reactions on the buoy, including those from the buoy mooring lines and the offloading lines, rather than from RAOs. Further details on the actual model used and the analyses performed are provided in the next sections. The figure below shows the system under consideration. The material in the example is based on a publication by Connaire et al., (2003).

This example illustrates a number of important modelling features in Flexcom, specifically:

* The CALM Buoy facility, which calculates the high frequency response of a floating buoy subject to first order wave forces.
* The Truss Element, which is ideally suited to modelling mooring chains and wires.

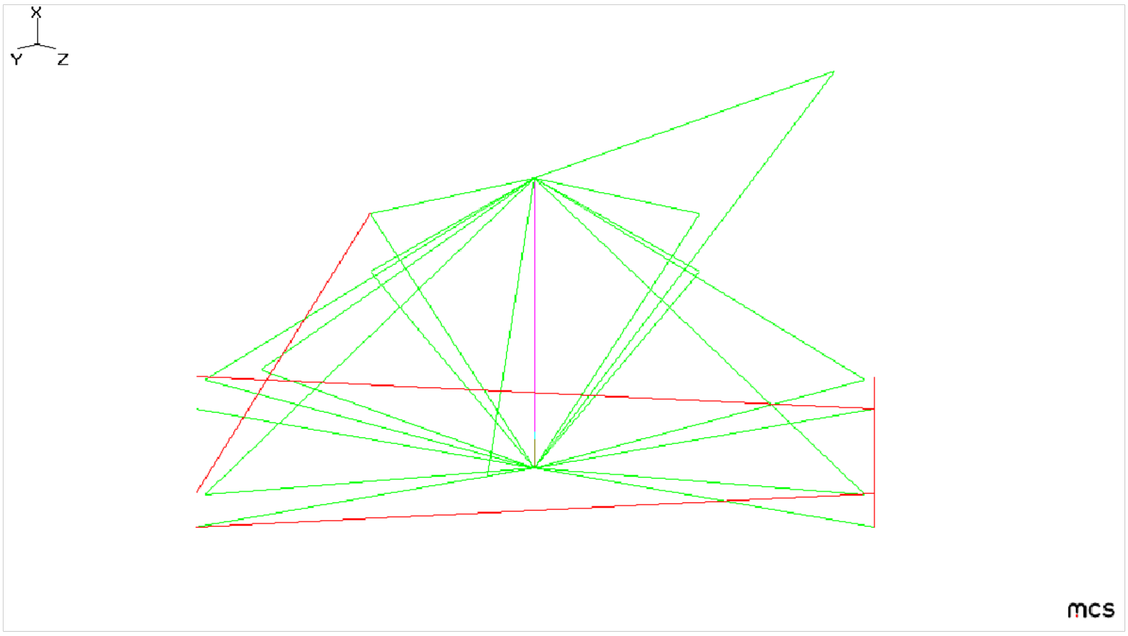


CALM Buoy Offloading System

Model Summary

GENERAL

The system, which characterises a sample oil offloading system, comprises a CALM buoy, which is located a distance of approximately 2000m from an FPSO, in a water depth of 800m. Fluid is transferred between the FPSO and the CALM buoy by means of a steel transfer line that hangs in a wave type configuration. The CALM buoy is moored using chain and steel wire rope mooring lines. The mooring lines are configured in four groups with two lines in each group. Mooring line pre-tension in each line is 800kN. A close-up of the finite element model in the region of the CALM buoy is presented in the figure below.



Close-up of CALM Buoy FEM

Because the main focus here is on the CALM buoy motions, FPSO motions are not considered in the various analyses of the system; instead the end of the transfer line remote from the buoy is fixed throughout. The only other constraints applied to the model are those at the mooring line seabed interface. Unlike a decoupled approach, there are no other prescribed vessel motions applied to the CALM buoy, mooring lines or steel transfer line.

Connaire et al., (2003) describes a range of different types of analyses of this system, of which only one type is performed here. This involves examining the response of the system to a unit height regular wave over a range of periods from 4s to 20s. The purpose of this is to determine CALM buoy RAOs for use in subsequent random sea fatigue analyses of the steel transfer line in either time or frequency domains.

In fact two separate systems are analysed in the example. The first comprises all of the elements described above. In the second the offloading line is omitted. Comparison of results from the two models will demonstrate the effect of the line on the CALM buoy motions, and emphasise the requirement for a coupled analysis capability for this type of system.

Analyses

INITIAL STATIC ANALYSIS

The base of each of the mooring line is fixed in all translational degrees of freedom. The CALM buoy CoG is also fixed in all degrees of freedom. A displacement of 10m is applied in the vertical or buoy heave direction at the CoG; this serves to develop tension in the mooring lines making for greater solution robustness in the next analysis stage. If the steel transfer line is included in the model, the remote end of the line is fixed in all translational degrees of freedom.

RELEASE ANALYSIS

The only change here is that the heave, surge and pitch constraints are removed from the CALM buoy CoG. The system mean equilibrium position is obtained as the buoy settles under the influence of gravity and buoyancy.

DYNAMIC ANALYSES

There is a total of 14 dynamic analyses of each model (with and without offloading line), making 28 analyses in total. Periods range from 4s to 20s (the values in the range omitted to keep the number of total runs down are 15s, 17s and 19s). In all cases, the BCs remain unchanged and are carried through automatically from the release analysis. Each dynamic simulation is run for 10 wave periods, using a variable time-step, and a ramp time of one wave period.

As noted previously, the main variables of interest are the buoy CoG motions in heave, surge and pitch, to define buoy motion RAOs. For this reason, the Summary Postprocessing facility is invoked, to generate a summary output file containing just the required variables of interest. The CALM buoy motions in heave, surge and pitch correspond to the motions of Node 8004 in DOFs 1, 2 and 6.

Each dynamic analysis runs 10 wave periods, but in generating the summary output only the results over the last 2 wave periods are considered. Note also that the Delete Database option is invoked. The 28 dynamic analyses generate considerable database output. The Delete Database option allows you to delete the database files for each run immediately the required results are extracted by the Summary Postprocessing facility, so minimising disk usage.

COLLATION OF SUMMARY OUTPUTS

The summary outputs of the dynamic analyses are collated so that the results can be viewed all at once as follows: the results where the offloading lines are present are collated separately from the results where the offloading lines are missing. This is achieved using Flexcom's Summary Collation capability. The collated results are presented in a collation spreadsheet and a 3D collation plot.

Results

Each dynamic analysis generates a summary output file, an example of which is shown in the table below – this one corresponds to the 20s period for the model including the offloading line.

```
                         F L E X C O M   Version 8.10.5

                 Academic Edition, Not Intended For Commercial Use

                      Time Domain Finite Element Analysis

                              (c) Wood 2019

CALM Buoy (Basic)with Offloading Line; Dynamic, T=20s

Summary of results from analysis:  E1-CBB-WithLine-Dynamic-T=20s

Summary Collate Plot Axes Data                      Value
------------------------------                      -----
Period                                              20.00
Dummy                                                1.00

Variable                                 Minimum      Maximum      Range      Standard     Scale      Units
                                                                              Deviation

(1) Kinematics
-----------
CALM Buoy Heave                          800.572      801.566      0.994        0.351     0.100E+01   m
CALM Buoy Surge                           55.427       57.736      2.309        0.816     0.100E+01   m
CALM Buoy Pitch                            0.865        1.579      0.714        0.241     0.100E+01   deg

(2) Forces
----------
Mooring Line 1 Fairlead Tension          480.702      489.007      8.305        3.052     0.100E+01   kN

Notes:
------
(1) Parameters calculated over time interval     160.179 to     200.000
```

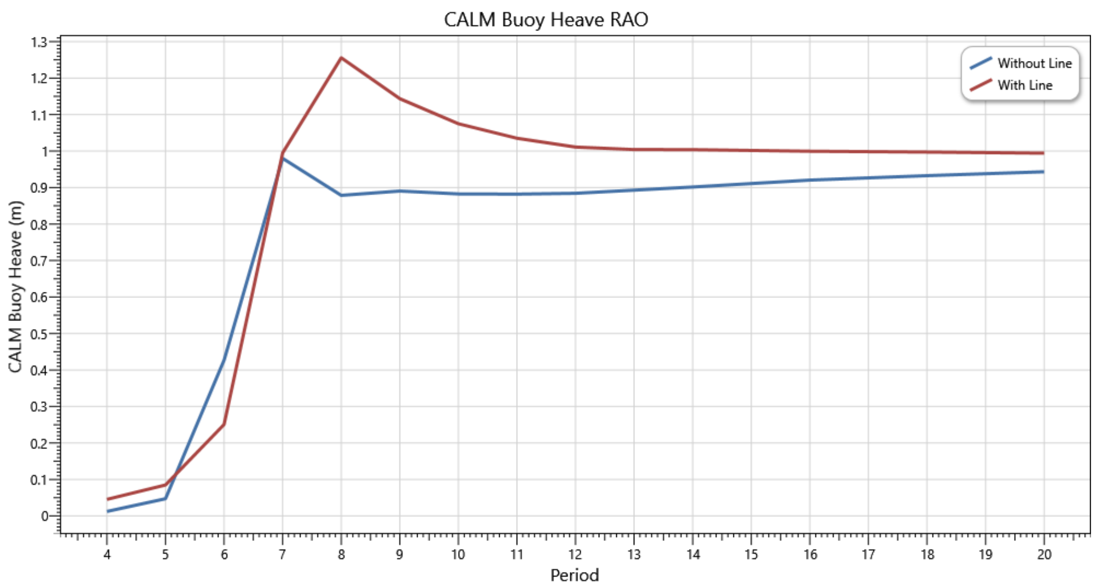
Sample Summary Output File

Summary collation may be used to automatically present the RAO data shown below. The RAOs are based on the 'Range' parameter, as the regular waves have 0.5m amplitude in all the dynamic simulations. Some observations are as follows:

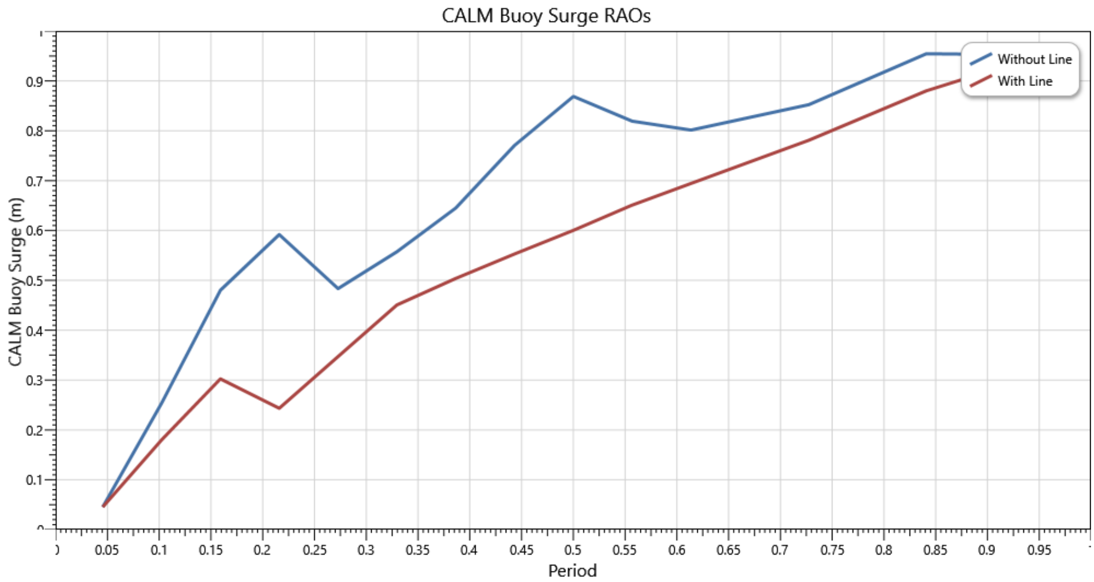
- Heave RAOs are generally larger with the presence of the steel transfer line. Additionally the RAO peak has shifted slightly from approximately 7s to 8s due to the steel transfer line.

- Surge RAOs are lower with the steel transfer line. The impact of this result is of particular significance for the detailed design of the transfer line itself. Because the transfer line is fatigue sensitive (particularly for seastates with lower peak periods) it is critical to identify the most appropriate RAOs for use in detailed design, in order to ensure feasibility. Unlike the case of heave, the natural period in surge does not occur in the wave frequency region.

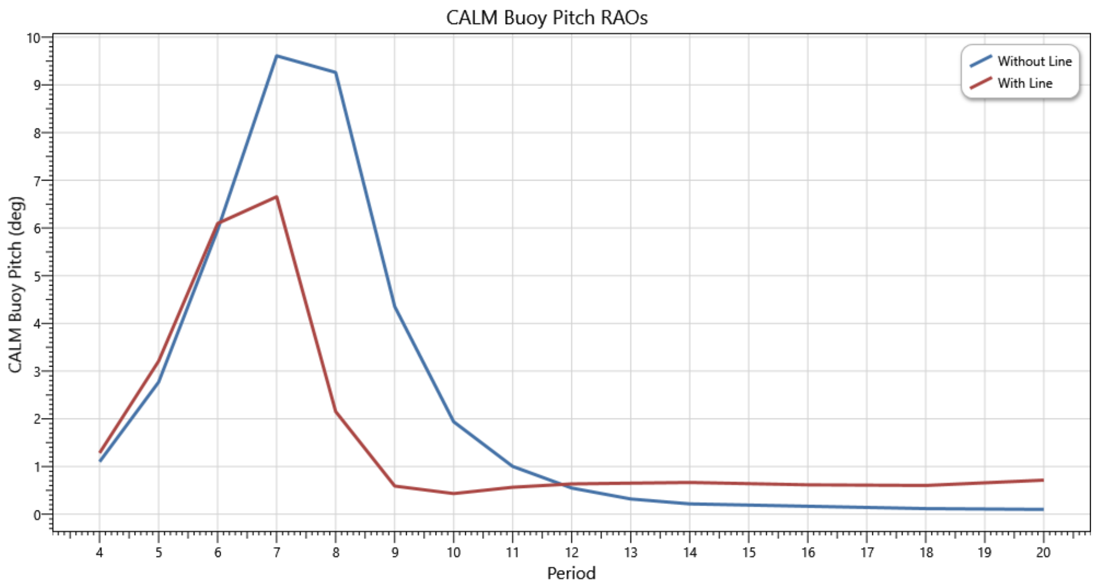
- Pitch RAOs are generally reduced by the presence of the steel transfer line, particularly at the peak and surrounding frequencies. More pitch is introduced at lower frequencies but this is less critical.



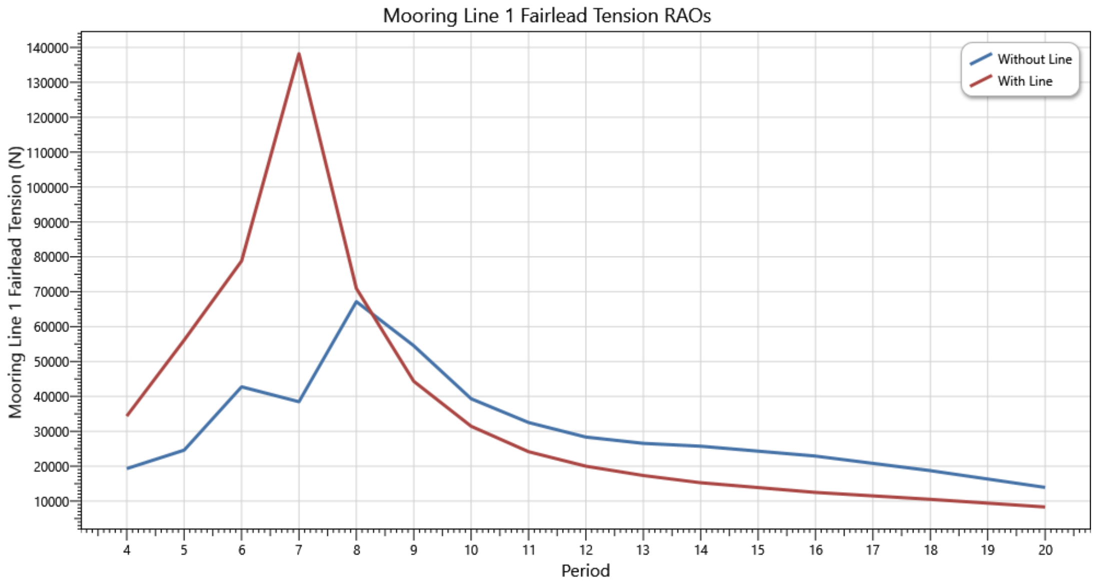
CALM Buoy Heave RAOs



CALM Buoy Surge RAOs



CALM Buoy Pitch RAOs



Mooring Line 1 Fairlead Tension RAOs

1.10.5.2 E02 - CALM Buoy - Complex

The objective of this example is to demonstrate the floating body modelling capability in Flexcom. The system under consideration is similar to that used in the preceding example, however the CALM buoy is modelled using the more generalised floating body. The overall layout of the example is as follows:

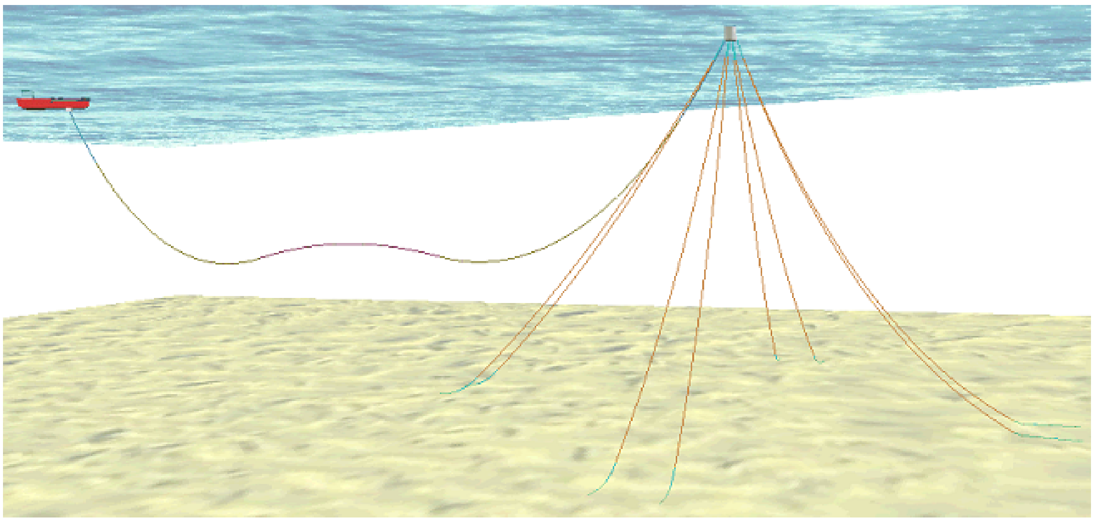
- Introduction gives a brief overview of the floating body analysis capability.
- Model Summary describes the finite element model.
- Analyses describes the various analyses performed, discussing the environmental and loading conditions to which the system is subjected.
- Results presents the pertinent results from the various analyses performed and discusses their significance.

Introduction

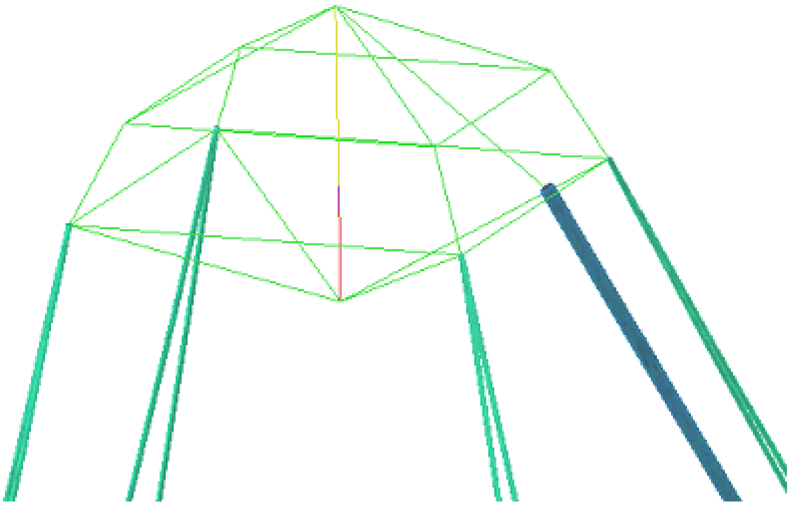
Flexcom includes a fully integrated floating body analysis capability in both the time and frequency domain. This is a considerably more powerful facility than that of the CALM buoy or moored vessel described in the preceding sections. Floating body analysis is applicable to both the time and frequency domain (both the CALM buoy and moored vessel facilities are restricted to the time domain). First and second order forces can be applied to the floating body (loading on the CALM buoy is restricted to first order, while only low frequency drift forces may be applied to the moored vessel). Refer to Coupled Analysis for a discussion of the underlying theory.

Model Summary

The model system used in this example is largely similar to that in the CALM Buoy Offloading System example and is shown in the first figure below. A close up of the floating body finite element model is shown in the second figure. Fluid is transferred between a FPSO and a floating body (i.e. a CALM buoy) by means of a steel transfer line that hangs in a wave type configuration. The buoy is moored using chain and steel wire rope mooring lines. Because the main focus here is on the floating body motions, FPSO motions are not considered. Instead the end of the transfer line remote from the buoy is fixed throughout. There are no prescribed vessel motions applied to the floating body. Floating body RAO motions in heave, surge and pitch are determined for both regular wave and random sea states in the time and frequency domains. The hydrodynamic characteristics are determined from a WAMIT analysis.



CALM Buoy Offloading System



Close-up of floating body model

Analyses

INITIAL STATIC ANALYSIS

The bases of each of the mooring lines are fixed in all translational degrees of freedom. The steel transfer line included in the model has the remote end fixed in all translational degrees of freedom. The floating body CoG node is also fixed in all degrees of freedom. A displacement of 12m is applied in the vertical or floating body heave direction at the CoG; this serves to develop tension in the mooring lines making for greater solution robustness in the next analysis stage.

RELEASE ANALYSIS

The only change here is that all the floating body degrees of freedom are now released. The system mean equilibrium position is obtained as the buoy settles under the influence of gravity, buoyancy and the static forces exerted by the mooring and offloading lines.

REGULAR WAVE ANALYSIS

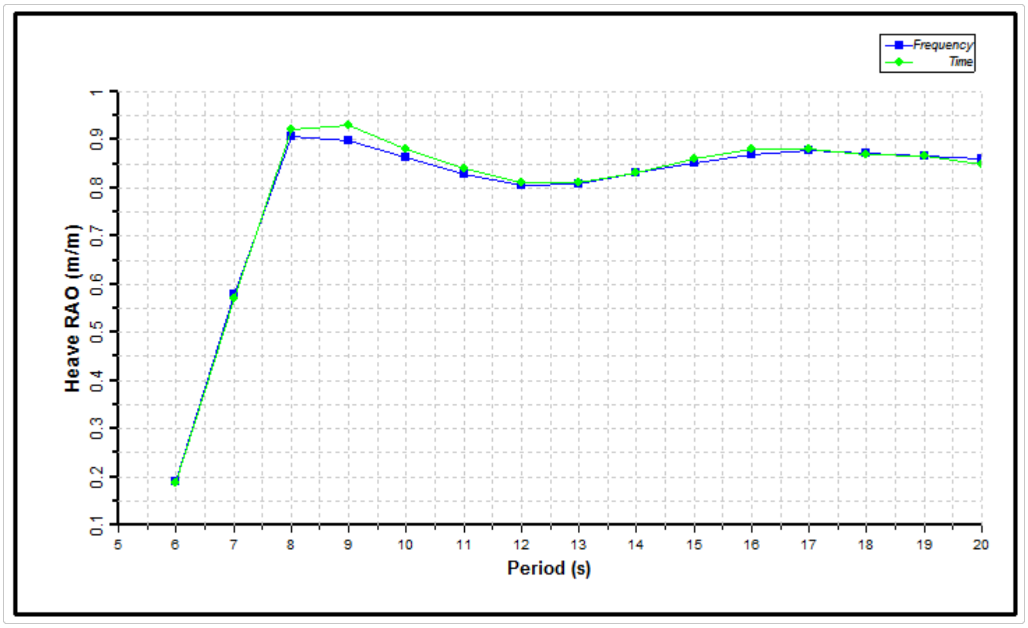
There are a total of 22 regular wave dynamic analyses for this model (11 each for the time and frequency domain). Periods range from 6s to 20s. In all cases, the boundary conditions remain unchanged and are carried through automatically from the release analysis. Each time domain dynamic simulation is run for 250s with a time step of 0.1s, and a ramp time of two wave periods. {For efficiency, only the 8s wave period input file is included, but you can easily recreate any of the others if you so wish}. The principal variables of interest are the floating body centre of gravity motions in heave, surge and pitch. In the time domain the RAO amplitude is based on the response over the last five wave periods.

RANDOM SEA ANALYSIS

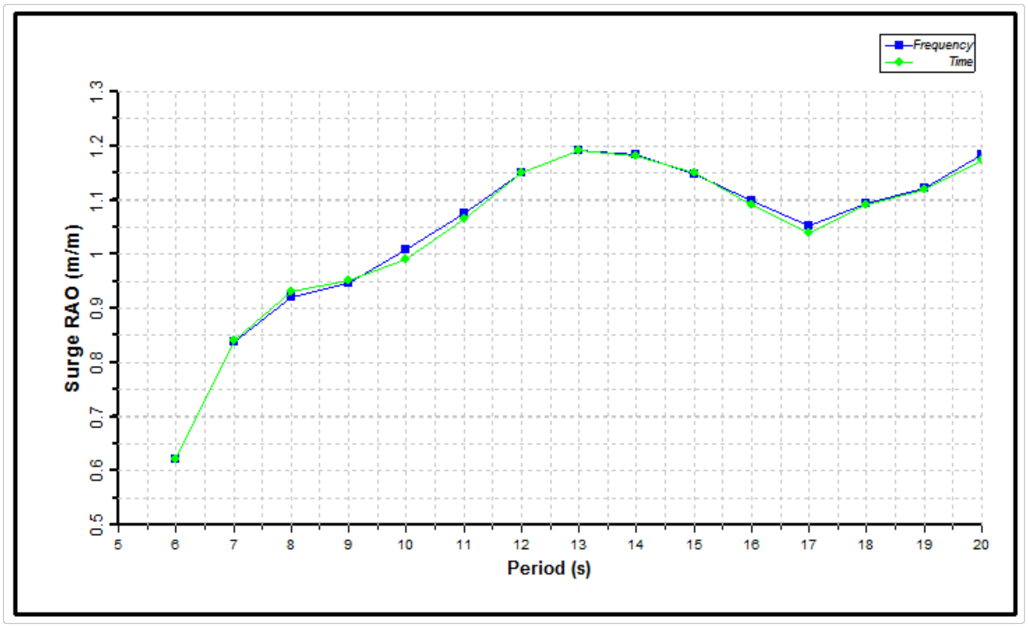
Two random sea analyses are performed, one each in the time and frequency domain. A Pierson-Moskowitz spectrum is used, with a significant wave height of 1.5m and a mean zero up crossing period of 10s. Purely for efficiency reasons, the time domain simulation is analysed for ¼ hour only, whereas a 3 hour simulation would generally be considered recommended practice.

Results

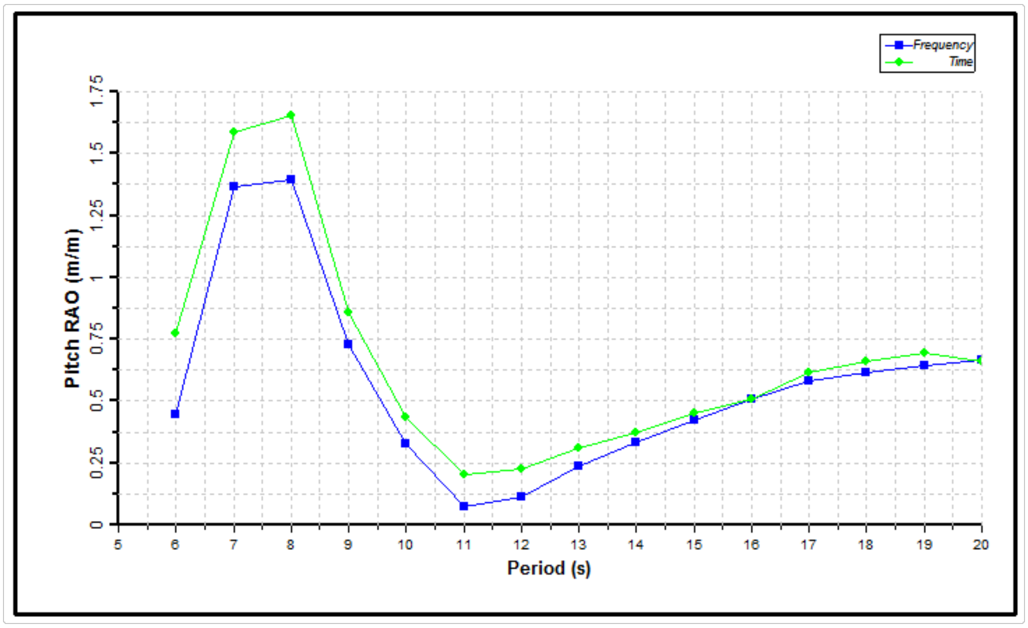
The RAOs from all the dynamics analyses are summarised in the RAO plots in the figures below. The RAOs determined from the regular wave analyses are shown in the Heave RAOs (regular wave analyses) figure to the Pitch RAOs (regular wave analyses) figure. The remaining figures, Heave RAOs (random sea analyses) figure to the Pitch RAOs (random sea analyses) figure, show corresponding data for the random sea analyses. It can be seen that there is very close agreement between the RAOs predicted by the time and frequency domain analyses.



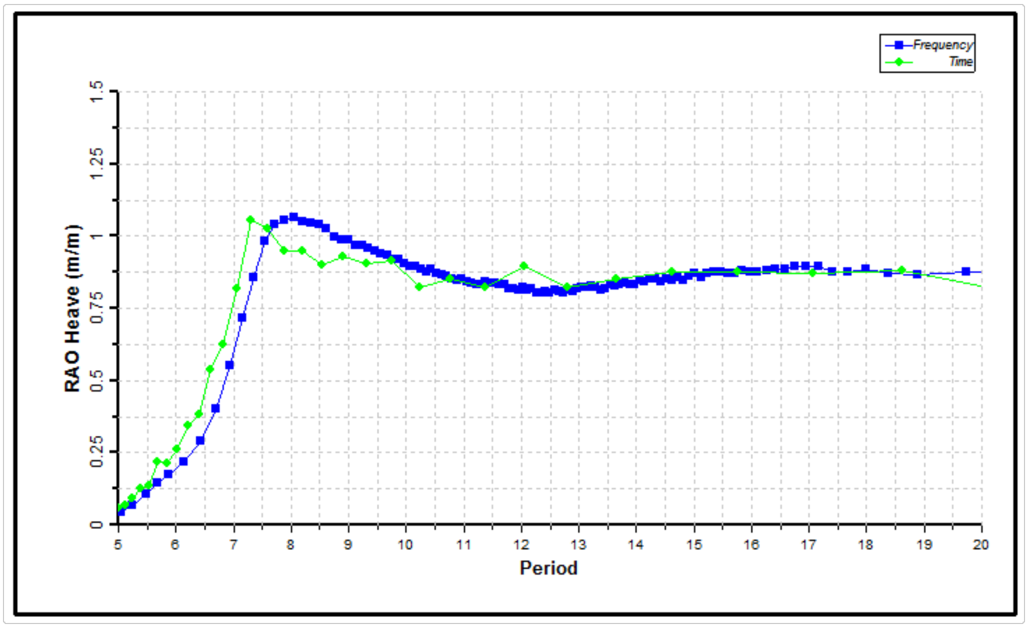
Heave RAOs (regular wave analyses)



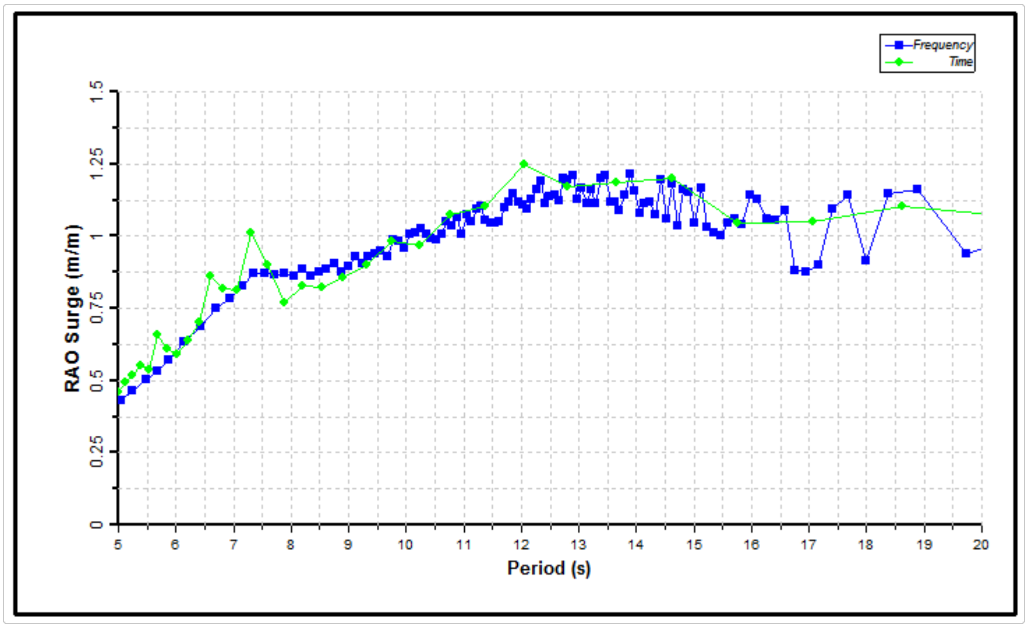
Surge RAOs (regular wave analyses)



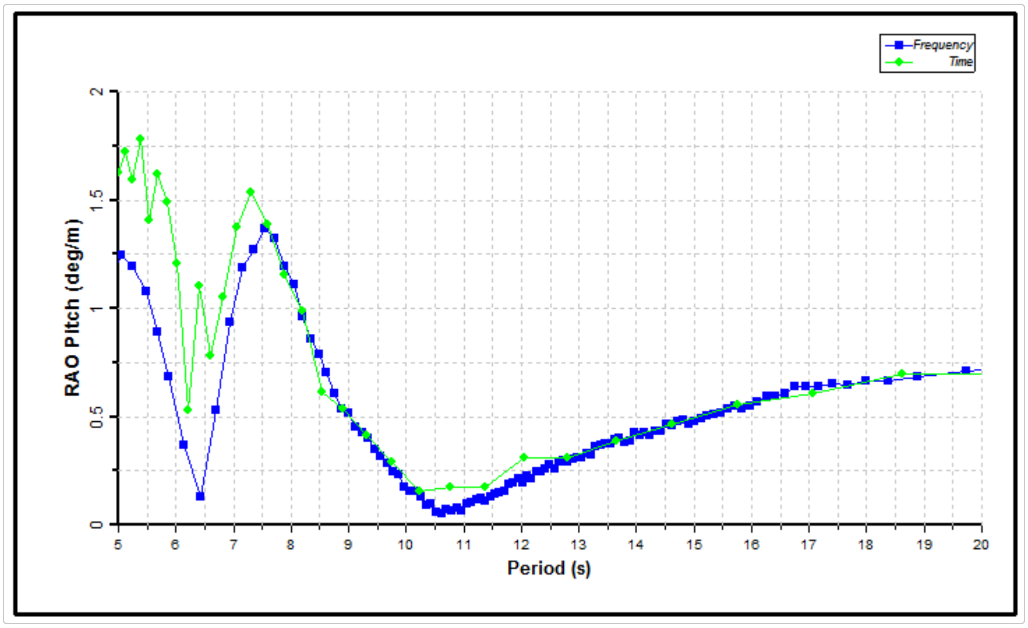
Pitch RAOs (regular wave analyses)



Heave RAOs (random sea analyses)



Surge RAOs (random sea analyses)



Pitch RAOs (random sea analyses)

1.10.5.3 E03 - Floating Hose

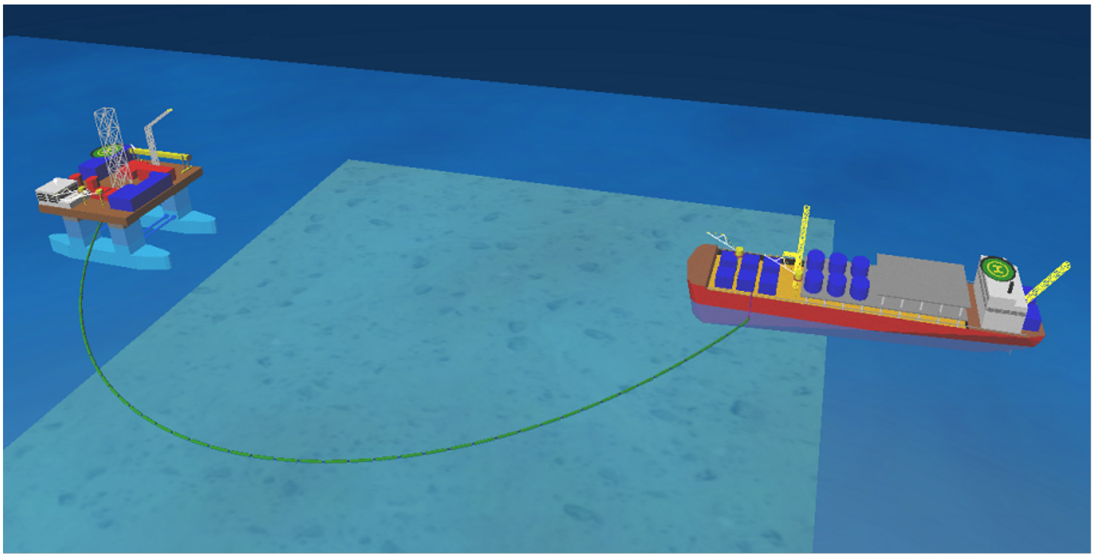
This section describes a sample floating hose system, and demonstrates how such a system is readily built using Flexcom. The overall layout of this document is as follows:

- Introduction gives an overview of the floating hose model.
- Model Summary describes the model in more detail, and discusses the use of the parameters used in setting up the model.
- Analyses briefly describes the various analyses performed, outlining the environmental and loading conditions to which the system is subjected.
- Results presents pertinent results from the analyses performed.

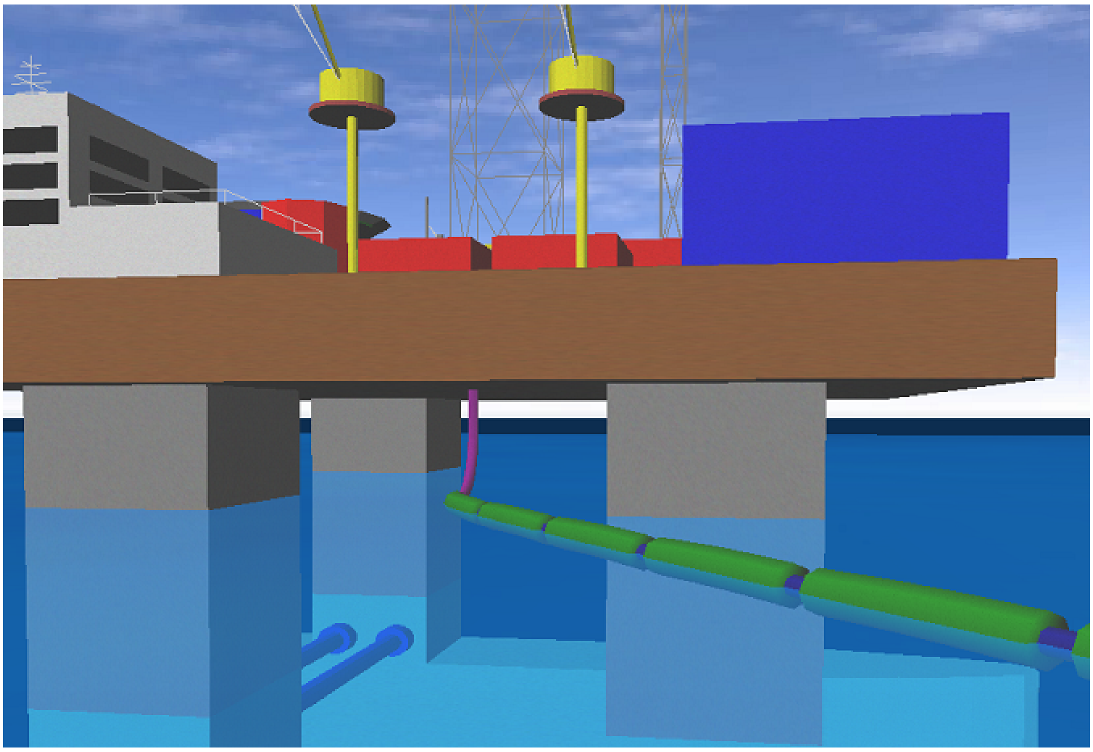
Introduction

This example considers the analysis of a floating hose system which connects a semi-submersible to an FPSO, a configuration sometime employed for offloading oil from a floating production unit. The example consists of an initial quasi-static analysis to facilitate model setup, followed by some dynamic analyses where the hose is subjected to regular wave loading.

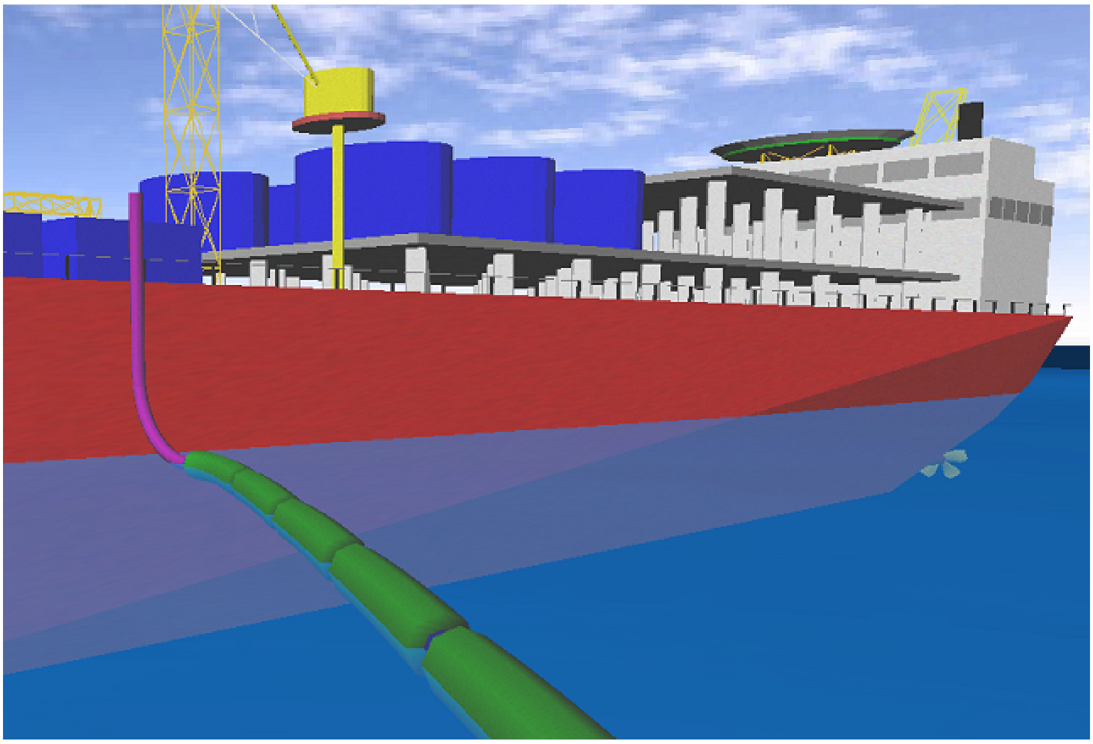
The overall hose configuration is shown in first figure below, while a more detailed view of the hose connections at either end are shown in final two figures.



Floating Hose System



Semi-Sub Connection



FPSO Connection

Model Summary

This example demonstrates the analysis of a floating hose system connecting a semi-sub and an FPSO, approximately 438m apart. The system consists of 52 flexible hose sections configured in a semi-circular shape. The sections have a relatively large diameter of 800mm due to buoyancy, which reduces to 400mm at the end of each section. Each section is 11.5m long, leading to a total hose length of 598m. The whole system is situated in 1000m of water.

The analysis is performed in several stages. The initial configuration of the system is established in a quasi-static analysis where the general shape of the floating hose is established before any dynamic loading is applied. A quasi-static approach is adopted due to the sensitivity of the buoyancy forces to any small displacements in the vertical direction. The initial analysis also includes static current loading.

A single master template file is then used to generate several regular wave input files of varying wave period. After the individual regular wave simulations have been performed, the relevant results are assembled centrally using the summary collation facility.

Analyses

QUASI-STATIC ANALYSIS

The end points of the floating hose are fixed to their respective vessels in all degrees of freedom. The positions of the semi-sub and the FPSO, and the relevant RAO data, are also specified. The initial analysis includes static loads such as gravity, buoyancy and current forces. The system is analysed quasi-statically to allow the hose to converge to its equilibrium configuration.

REGULAR WAVE TEMPLATE

The master template file generates several regular wave input files of varying wave period, including wave periods of 8, 9, 10, 12, 14 and 16 seconds.

DYNAMIC ANALYSES

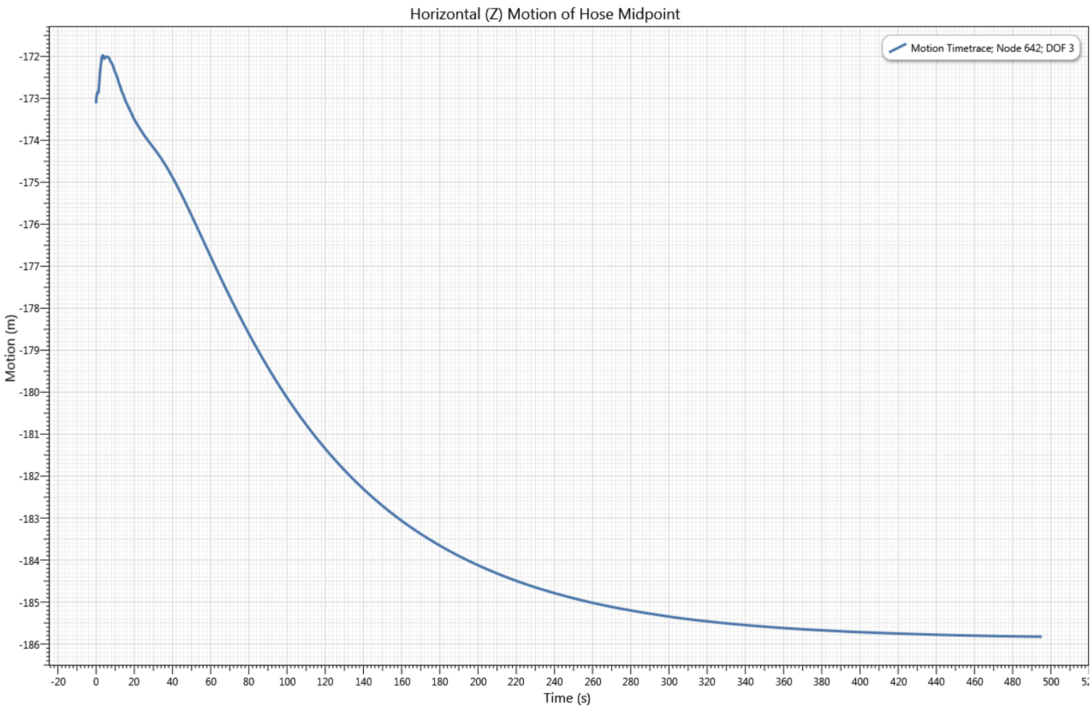
In each dynamic stage, a regular wave is applied. The wave amplitude is 3 metres, the period changes with each file, and the wave direction is aligned with the global Y-axis. The boundary conditions remain unchanged and are carried through automatically from the previous analysis stage. A variable time step is used to allow the software to use an optimal time step.

SUMMARY COLLATION

The outputs of the dynamic analyses, specifically the bending moments at either end of the floating hose, are collated into a single spreadsheet.

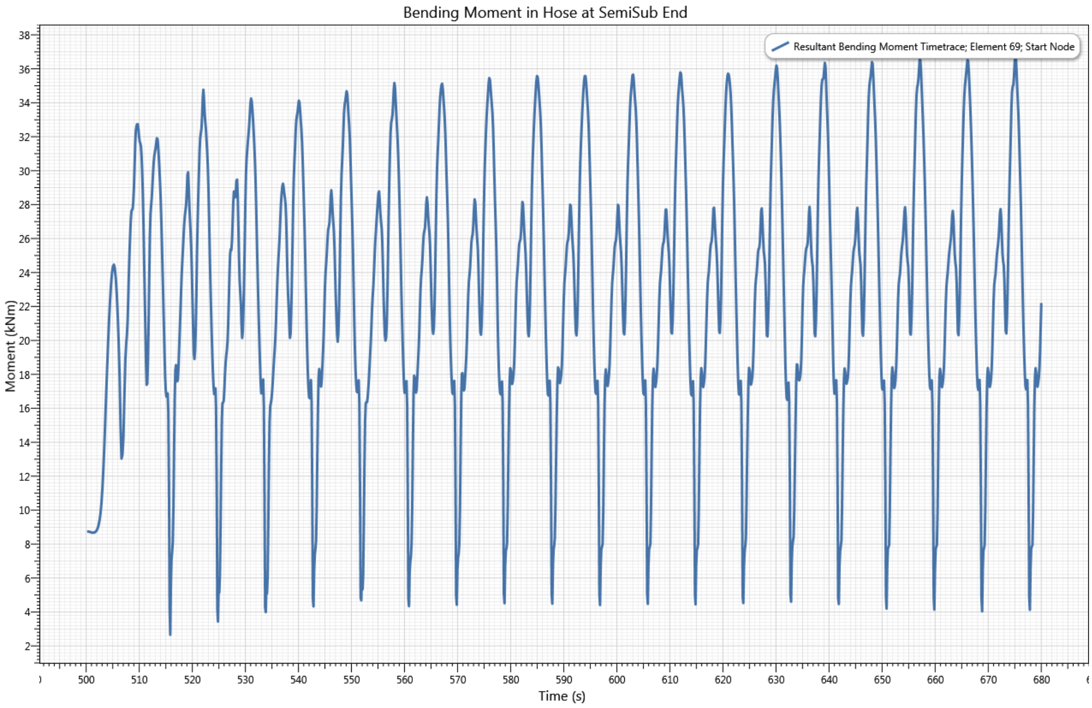
Results

The figure below shows the horizontal motion of the mid-point of the hose as it approaches static equilibrium, starting from the initial approximate semi-circular shape.

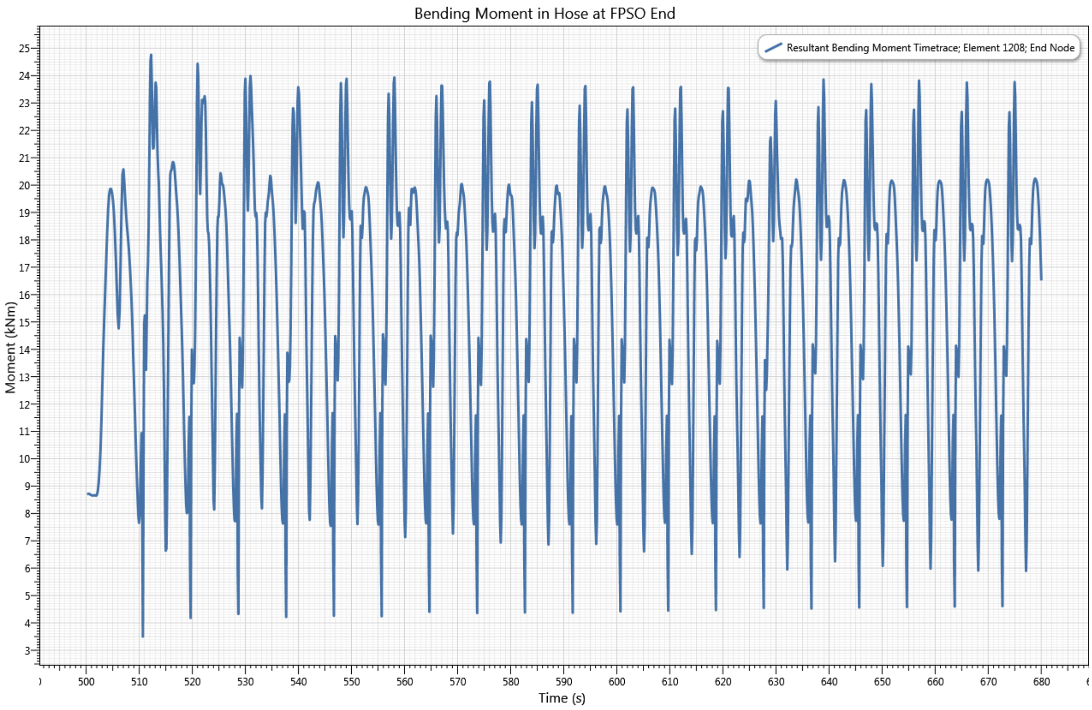


Horizontal (Z) Motion of Hose Midpoint

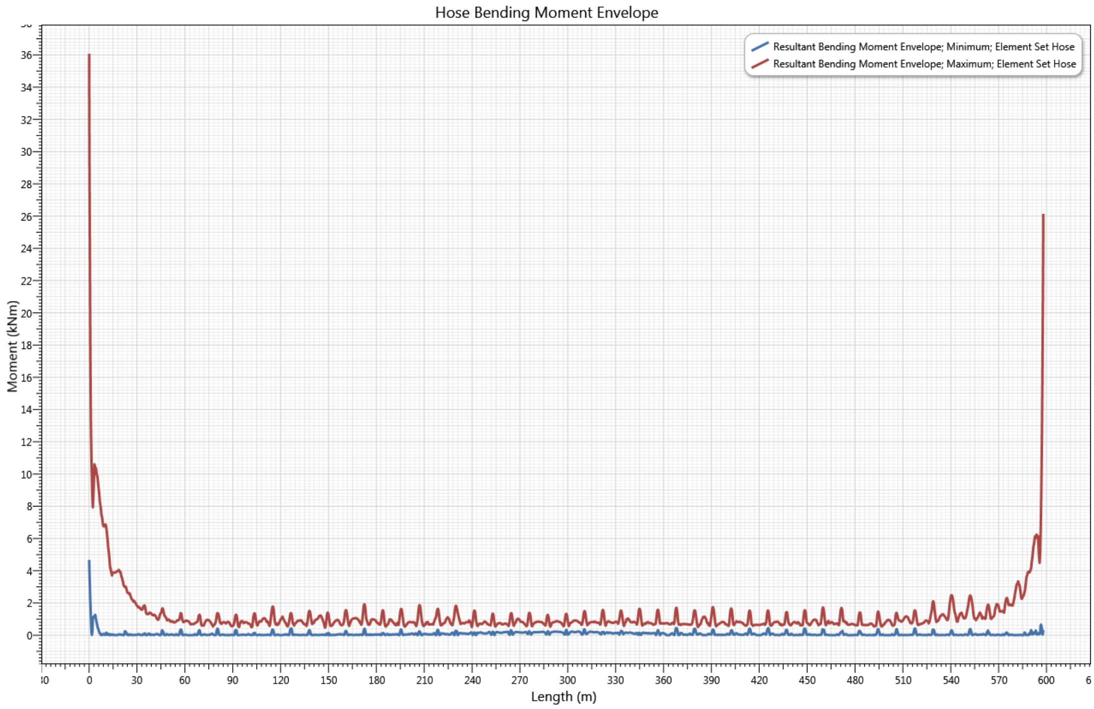
The following figures contain a selection of results from the dynamic analysis of a regular wave of period 9 seconds. The first and second figures below show the bending moment in the hose at the semi sub end and the FPSO end, respectively, while the third figure shows the moment envelope for the entire hose. The most severe moments occur at the vessel connection points as expected.



Bending Moment Semi-Sub End

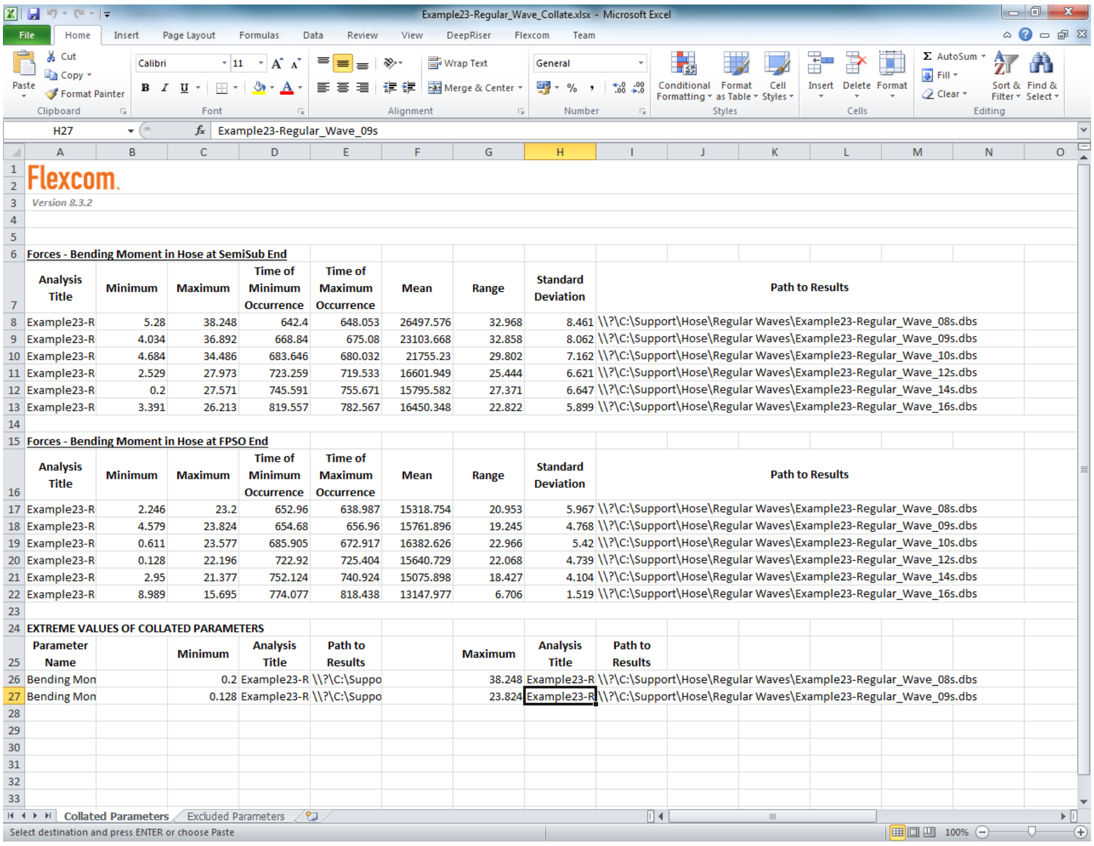


Bending Moment FPSO End



Bending Moment Envelope

The final figure below shows the summary collation spreadsheet. As highlighted in the extreme values section, the maximum bending moments occur for the 8s and 9s regular wave cases.



Flexcom

Version 8.3.2

Forces - Bending Moment in Hose at SemiSub End

Analysis Title	Minimum	Maximum	Time of Minimum Occurrence	Time of Maximum Occurrence	Mean	Range	Standard Deviation	Path to Results
Example23-R	5.28	38.248	642.4	648.053	26497.576	32.968	8.461	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_08s.dbs
Example23-R	4.034	36.892	668.84	675.08	23103.668	32.858	8.062	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_09s.dbs
Example23-R	4.684	34.486	683.646	680.032	21755.23	29.802	7.162	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_10s.dbs
Example23-R	2.529	27.973	723.259	719.533	16601.949	25.444	6.621	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_12s.dbs
Example23-R	0.2	27.571	745.591	755.671	15795.582	27.371	6.647	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_14s.dbs
Example23-R	3.391	26.213	819.557	782.567	16450.348	22.822	5.899	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_16s.dbs

Forces - Bending Moment in Hose at FPSO End

Analysis Title	Minimum	Maximum	Time of Minimum Occurrence	Time of Maximum Occurrence	Mean	Range	Standard Deviation	Path to Results
Example23-R	2.246	23.2	652.96	638.987	15318.754	20.953	5.967	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_08s.dbs
Example23-R	4.579	23.824	654.68	656.96	15761.896	19.245	4.768	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_09s.dbs
Example23-R	0.611	23.577	685.905	672.917	16382.626	22.966	5.42	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_10s.dbs
Example23-R	0.128	22.196	722.92	725.404	15640.729	22.068	4.739	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_12s.dbs
Example23-R	2.95	21.377	752.124	740.924	15075.898	18.427	4.104	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_14s.dbs
Example23-R	8.989	15.695	774.077	818.438	13147.977	6.706	1.519	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_16s.dbs

EXTREME VALUES OF COLLATED PARAMETERS

Parameter Name		Minimum	Analysis Title	Path to Results		Maximum	Analysis Title	Path to Results
Bending Mon		0.2	Example23-R	\\?\C:\Suppo		38.248	Example23-R	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_08s.dbs
Bending Mon		0.128	Example23-R	\\?\C:\Suppo		23.824	Example23-R	\\?\C:\Support\Hose\Regular Waves\Example23-Regular_Wave_09s.dbs

Summary Collation

1.10.6 F - Pipelines

Section F contains some examples of pipelines, including:

- F01 - As-Laid Span Analysis
- F02 - Upheaval Buckling

1.10.6.1 F01 - As-Laid Span Analysis

This example describes an as-laid span analysis, and demonstrates a number of features of Flexcom particularly suited to the modelling of pipelines. The overall layout of this example is as follows:

- Introduction gives an overview of as-laid span analysis, and notes some of the features of Flexcom which are relevant to the analysis.
- Model Summary describes the model in more detail.
- Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions to which the pipeline is subjected.
- Results presents pertinent results from the various analyses performed.

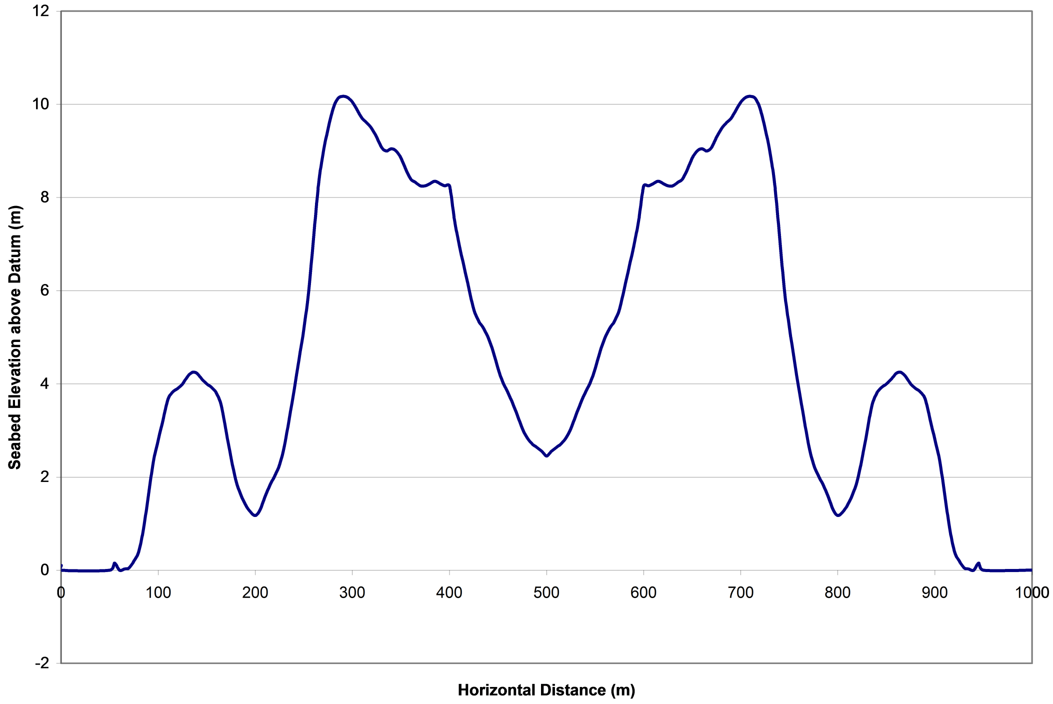
Introduction

This example considers the analysis of a pipeline lying on an arbitrary seabed. The pipeline is subjected to static loads of gravity, buoyancy and current, along with dynamic regular wave loading. The effects of Poisson’s ratio and temperature loading (due to the temperature differential between external and entrained seawater), are included. The example illustrates the use of Flexcom in the analysis of pipelines, in this case in an as-laid configuration. The ease of specifying a wholly arbitrary seabed bathymetry is demonstrated.

Model Summary

This example considers the analysis of 1000m of pipeline lying on an arbitrary rigid seabed in a water depth of 100m. The analysis is conducted in a number of stages, as follows. An initial static analysis is first performed where the pipeline is located and restrained horizontally just above the arbitrary seabed profile. The vertical constraints on the pipeline are then removed in a subsequent quasi-static analysis, and the pipe is allowed to descend to the seabed under the influence of gravity and buoyancy. A further static analysis is then performed in which current loading is introduced, and the static equilibrium position of the pipeline on the seabed is obtained. Finally, a dynamic regular wave analysis is performed to investigate influence of wave loading on the pipeline. The effects of temperature loading and Poisson’s ratio are included from the initial static stage.

The seabed profile is plotted in the figure below. Obviously the horizontal and vertical scales are very different, in order to render the bathymetry detail visible.



Arbitrary Seabed Profile

Analyses

INITIAL STATIC ANALYSIS

The nodes at the pipeline extremities are fixed in all translational degrees of freedom, and a rotational constraint in DOF 5 is included at the first node to prevent static determinacy. Additional restraints in the vertical direction are applied at regular intervals of 20m along the pipeline to position the pipeline above the arbitrary seabed.

QUASI-STATIC ANALYSIS

All vertical constraints along the length of the pipeline and the horizontal constraints at the second end are removed. Significant mass damping is specified to minimise initial transients, and the pipeline settles quickly over the seabed under the action of gravity and buoyancy.

CURRENT ANALYSIS

A piecewise-linear current is applied to the pipeline. As pipeline motion is constrained by seabed friction, all boundary conditions, except the DOF 5 restraint at the first end, are removed.

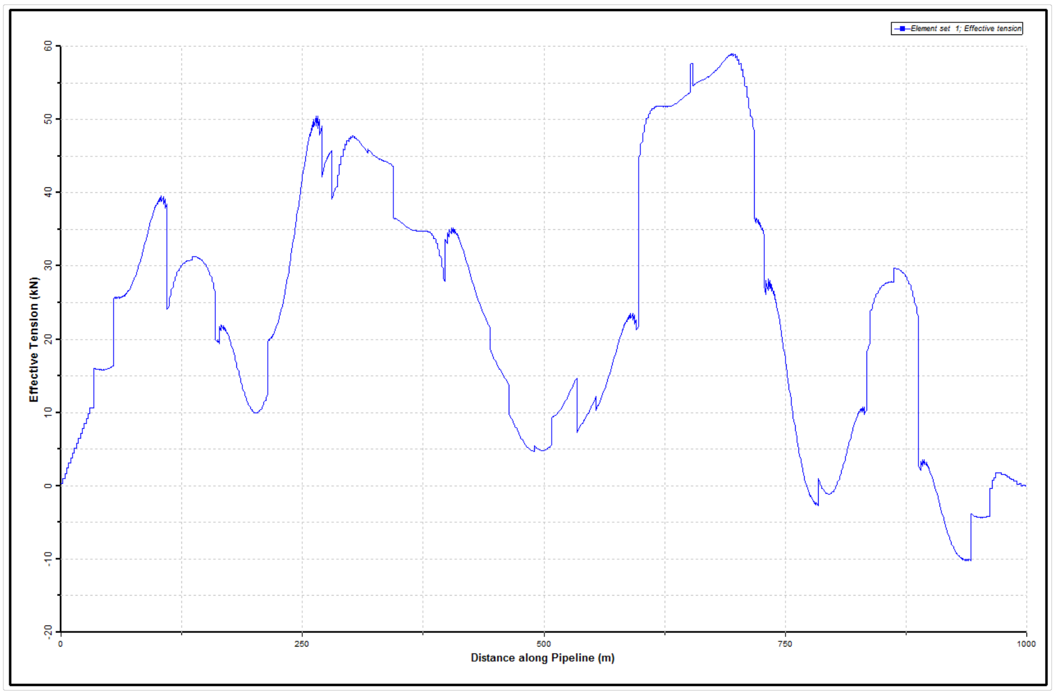
DYNAMIC ANALYSIS

A regular wave of amplitude 5m and period 18s is applied. The boundary conditions remain unchanged and are carried through automatically from the current analysis. The dynamic analysis is run for 15 wave periods, at a time-step of 0.5s, to ensure steady conditions become established.

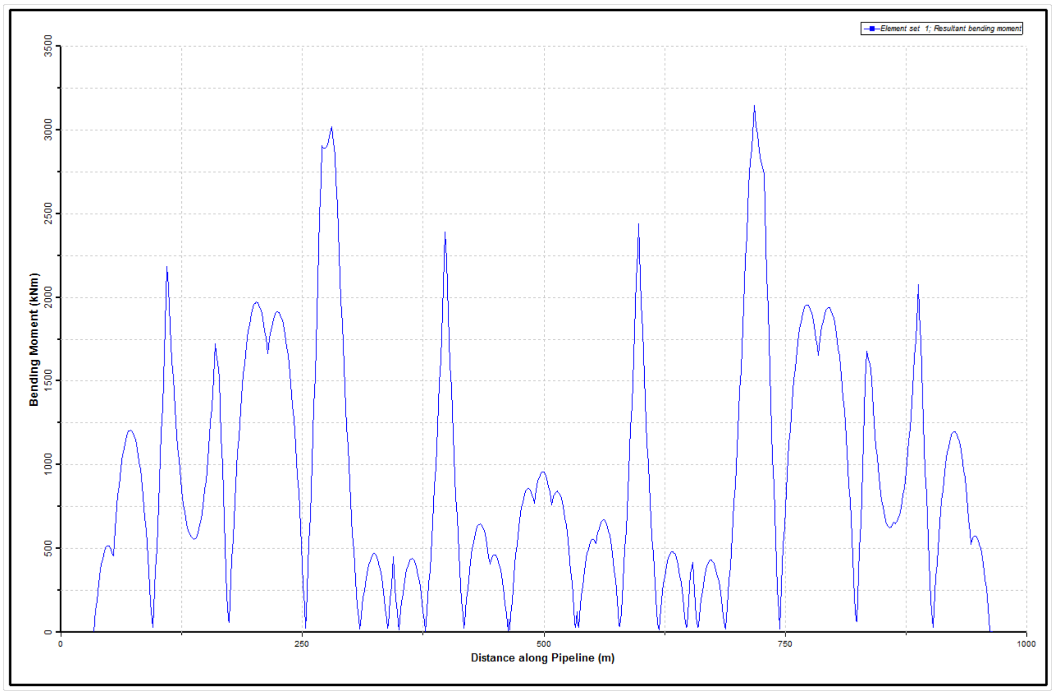
Results

CURRENT

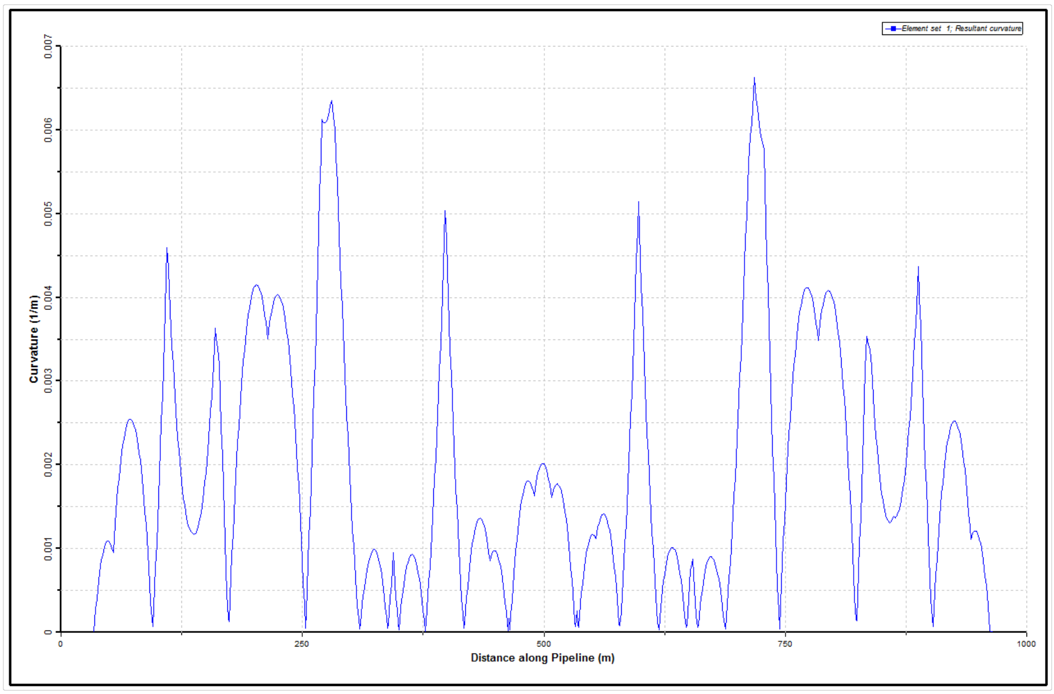
Results from the current analysis are presented in the figures below. The first figure shows a plot of the effective tension distribution in the pipeline, the second figure shows the static bending moments in the pipeline, while the third figure shows a plot of the static curvature distribution in the pipeline.



Static Effective Tension Distribution



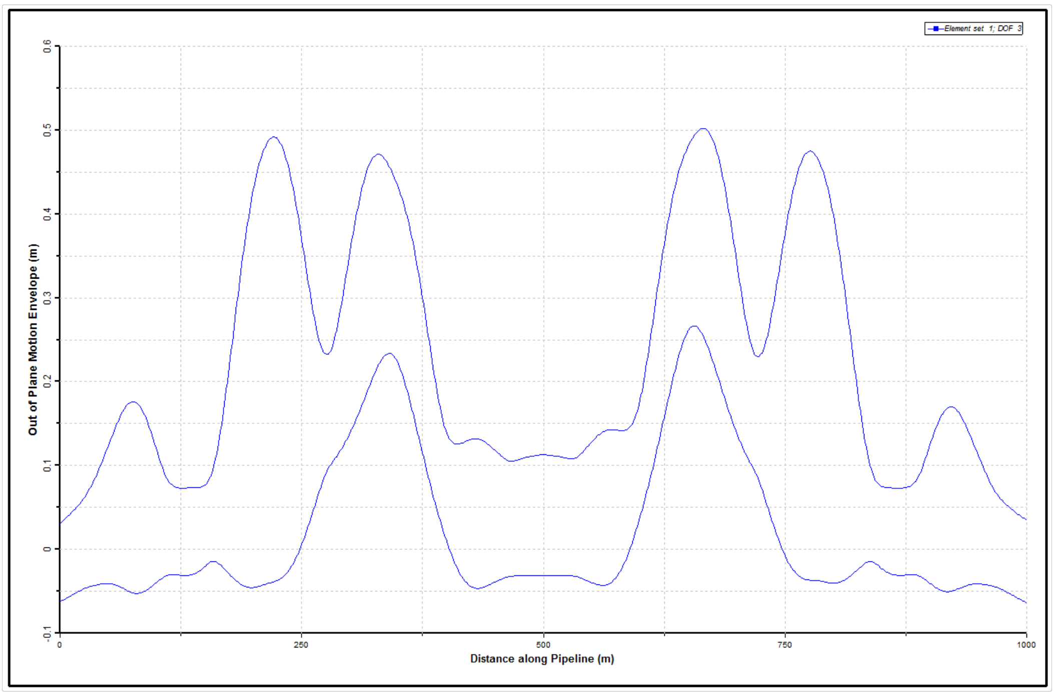
Static Bending Moment Distribution



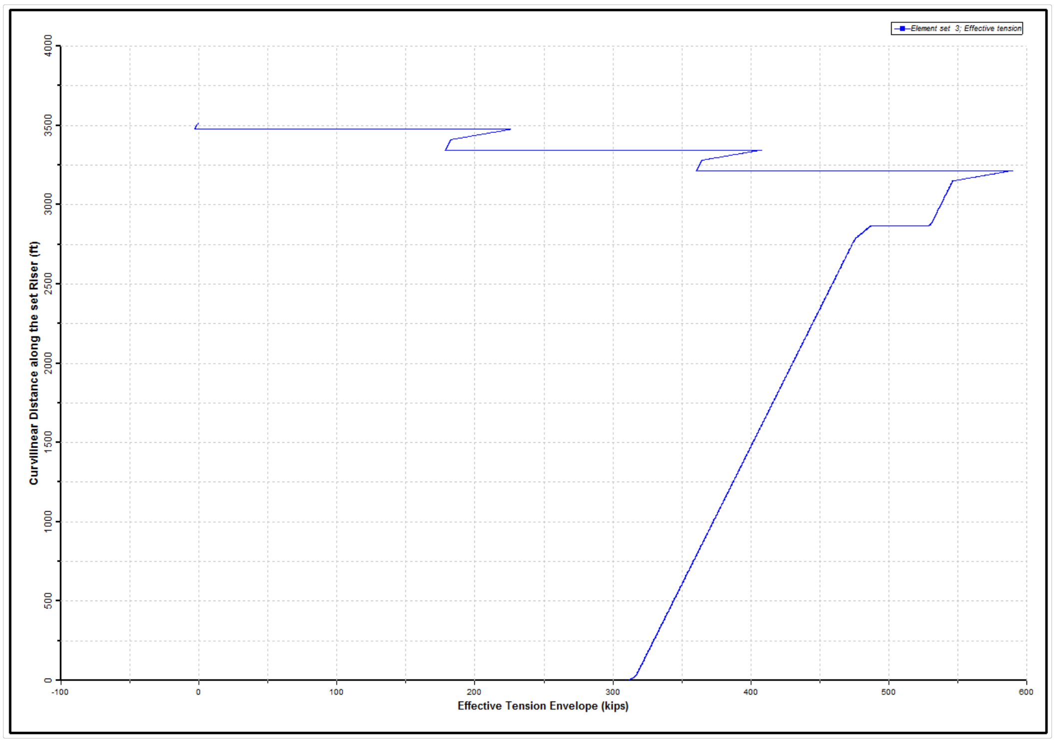
Static Curvature Distribution

DYNAMIC

Results from the dynamic analysis are presented in the figures below. The first figure shows an envelope of motion in the global Z-direction, that is, in the wave direction, transverse to the initial plane of the pipeline. The second figure shows an envelope of effective tension. Note that both plots are generated over the last three wave periods.



Horizontal Z-Motion Envelope



Effective Tension Envelope

1.10.6.2 F02 - Upheaval Buckling

This example describes an analysis to determine if a pipeline lying on the seabed is susceptible to upheaval buckling, and to demonstrate that Flexcom can be used successfully in this type of pipeline study. The overall layout of this example is as follows:

- Introduction gives an overview of the upheaval buckling analysis.
- Model Summary describes the model in more detail, and discusses some relevant analytical capabilities of the software.
- Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions to which the pipeline is subjected.
- Results presents pertinent results from the various analyses performed and discusses their significance in relation to upheaval buckling analysis.

Introduction

This example illustrates the use of Flexcom in an analysis to predict if upheaval buckling will occur in a pipeline installed in a water depth of 88.5m, for a user-defined seabed imperfection height and slug flow regime. The objective is not to demonstrate how such analyses should be performed as part of a pipeline design project; rather it is to show Flexcom can be readily and successfully employed in this type of study. To this end a somewhat simplified scenario is considered, as described in subsequent sections.

A multi-stage analysis procedure is employed, consisting of four static steps followed by a dynamic run which is the actual upheaval buckling analysis. The sequence of static analyses is used to develop a distribution of tension (or more correctly compression) in the pipeline. Such a distribution results from the pipeline weight in water, the seabed friction characteristics, the degree of pipeline end restraint, and the level of backfill. Classical pipeline calculations would be used in determining this distribution, a process which is not described here. In the analysis approach used here, the tension due to buoyancy, gravity and internal fluid is adjusted to the required distribution by the application of a temperature loading. This does not represent a genuine physical load condition, but is simply a convenient method to develop a used-defined compression distribution in the pipeline.

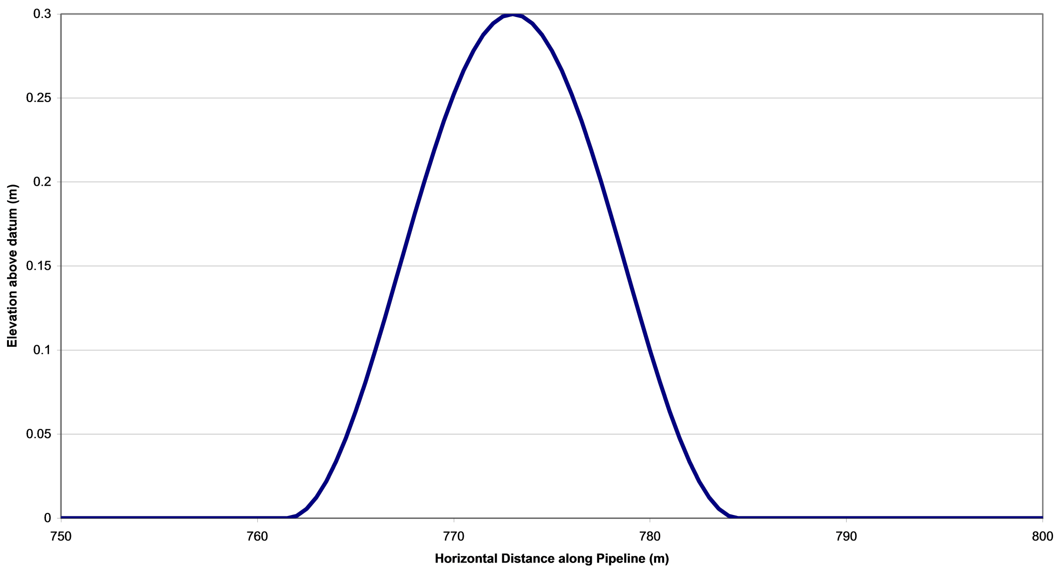
This example and the previous one in the manual (of an as-laid span) illustrate how Flexcom can be readily employed in analysing pipelines under various conditions. This present analysis also demonstrates the program facility to model multi-phase slug flow.

Model Summary

This example considers the analysis of 1546   m of pipeline lying in a water depth of 88.5m. The pipeline is lying on a rigid seabed which is everywhere flat except over a length of 23m at the centre of the model. Here an imperfection of 0.3m in height is modelled as a simple sine curve, as shown in the figure below.

An initial static analysis is first performed where the pipeline is located and restrained horizontally just above the seabed. The vertical restraints are then removed in a subsequent quasi-static run, and the pipe ‘drops’ to the seabed under the influence of gravity and buoyancy. A further static analysis is then performed in which the pipeline internal fluid is introduced. In a final static run, a temperature loading is specified in order to build up the required compressive forces in the pipeline, and the static equilibrium position of the pipeline on the seabed is obtained.

At this stage a dynamic analysis is performed to investigate the influence of slug flow on the pipeline. To keep the run time reasonable, the slug is not introduced at the end of the pipeline – instead it enters the pipeline at a distance of about 30m from the start of the imperfection. The slug speed is 3.0m/s, so the front of the slug arrives at the start of the imperfection about 10s after entering the pipeline, and at the mid-point of the imperfection after a further 4s. Examination of the pipeline response in and around this time region will indicate whether upheaval buckling occurs or not.



Seabed Imperfection Profile

Analyses

INITIAL STATIC ANALYSIS

The pipeline is defined as being 1m above the seabed datum, and so is 0.7m above the highest point of the imperfection. The nodes at the pipeline extremities are fixed in all degrees of freedom. Additional restraints in the vertical direction are applied at regular intervals of 10m along the pipeline to position the pipeline above the seabed.

QUASI-STATIC ANALYSIS

All constraints imposed on the pipeline are removed, and the pipe is allowed to drop to the seabed under the influence of gravity and buoyancy.

FLUID ANALYSIS

The internal fluid is added to the pipeline. No boundary conditions are applied and the pipeline is restrained by the seabed friction.

TEMPERATURE ANALYSIS

Temperature loading is introduced and compressive forces build up in the pipeline.

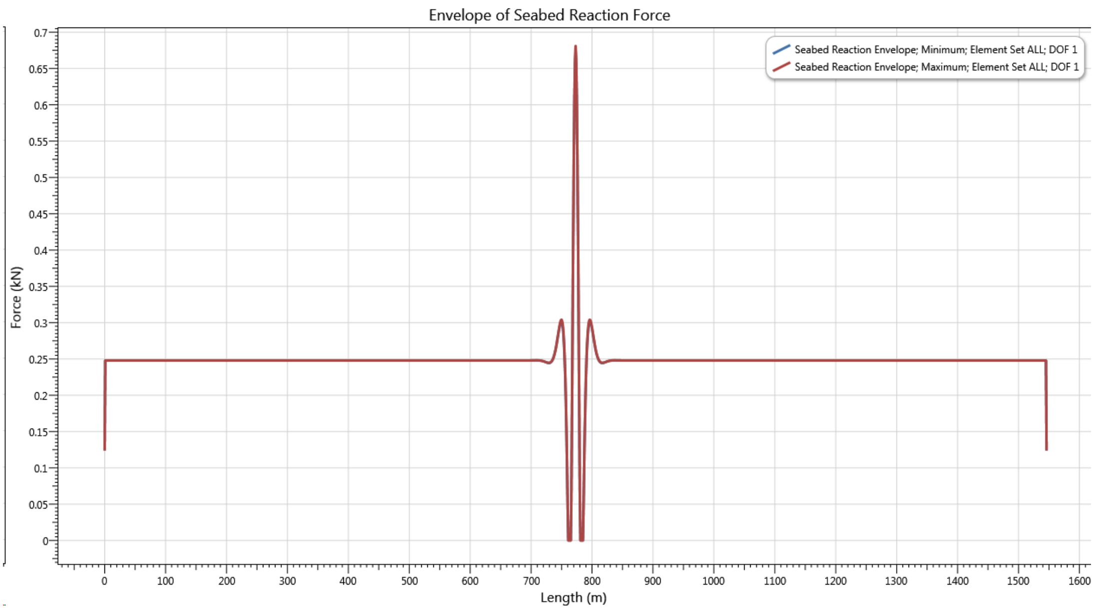
DYNAMIC ANALYSIS

The influence of slug flow on the pipeline is investigated. Two successive slugs of length 24m, with respective densities of 850kg/m^3 and 300kg/m respectively, pass through the pipeline at a speed of 3.0m/s.

Results

FLUID ANALYSIS

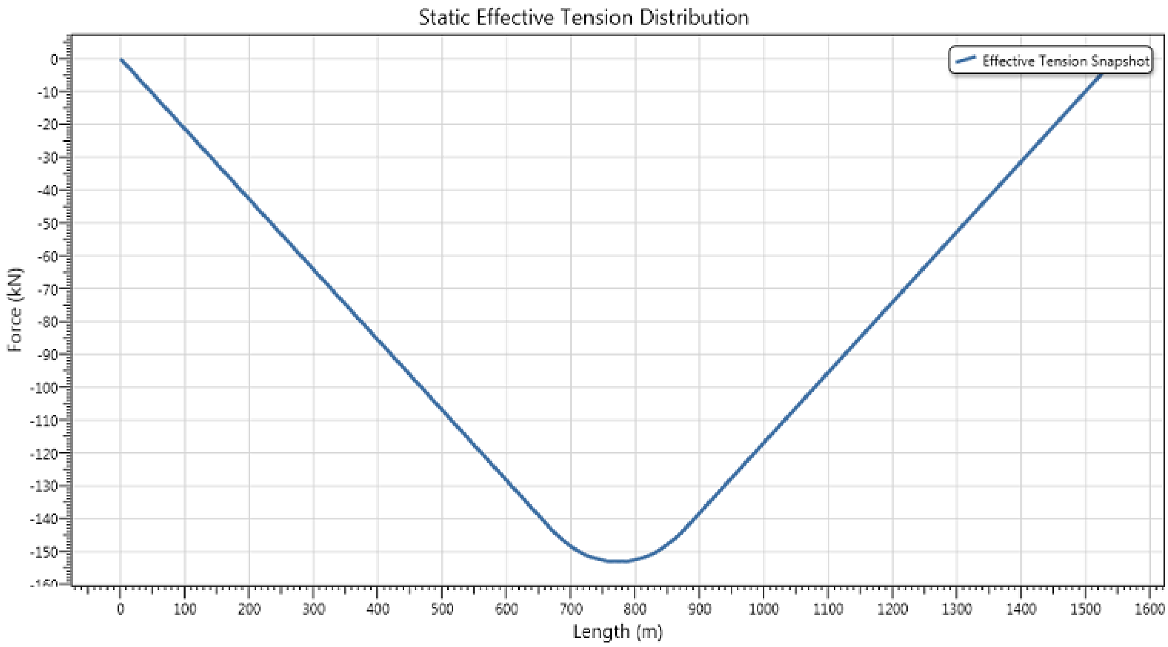
With the internal fluid present, the pipeline is fully weighted, and vertical reaction forces are exerted by the seabed.



Seabed Reaction Force

TEMPERATURE ANALYSIS

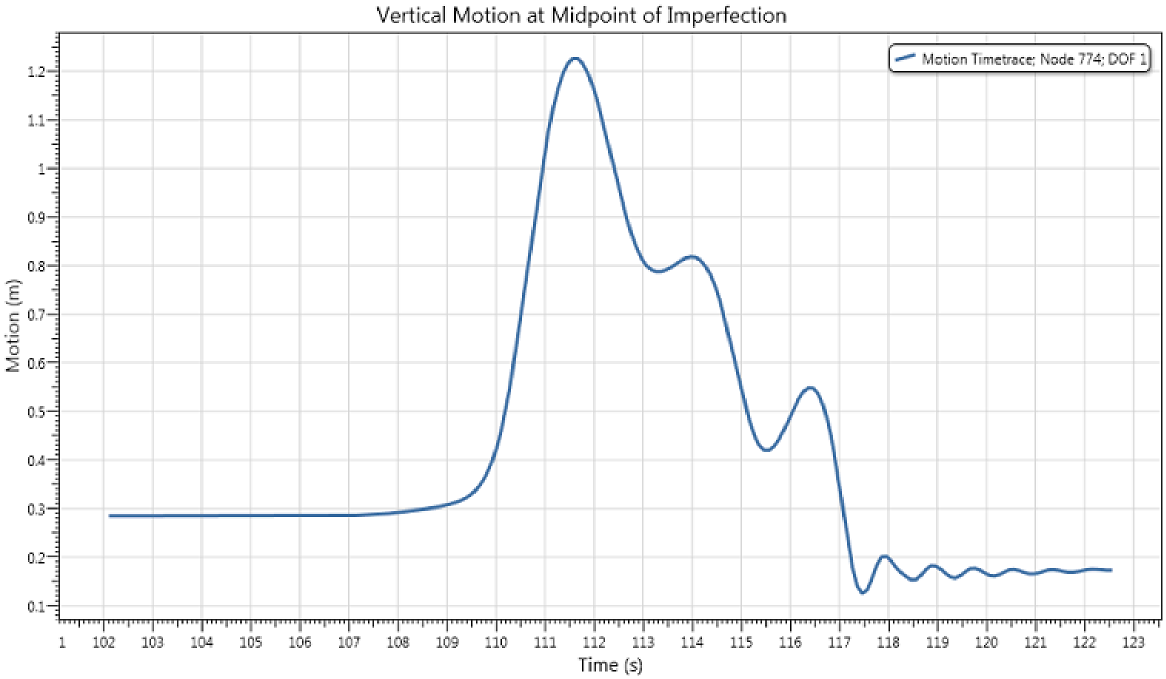
Results from the static analysis in which the temperature loading is introduced are presented in the figure below, which shows a plot of the effective tension distribution in the pipeline. There is clearly a significant compressive axial load as required due to the temperature loading and the seabed friction.



Static Effective Tension Distribution

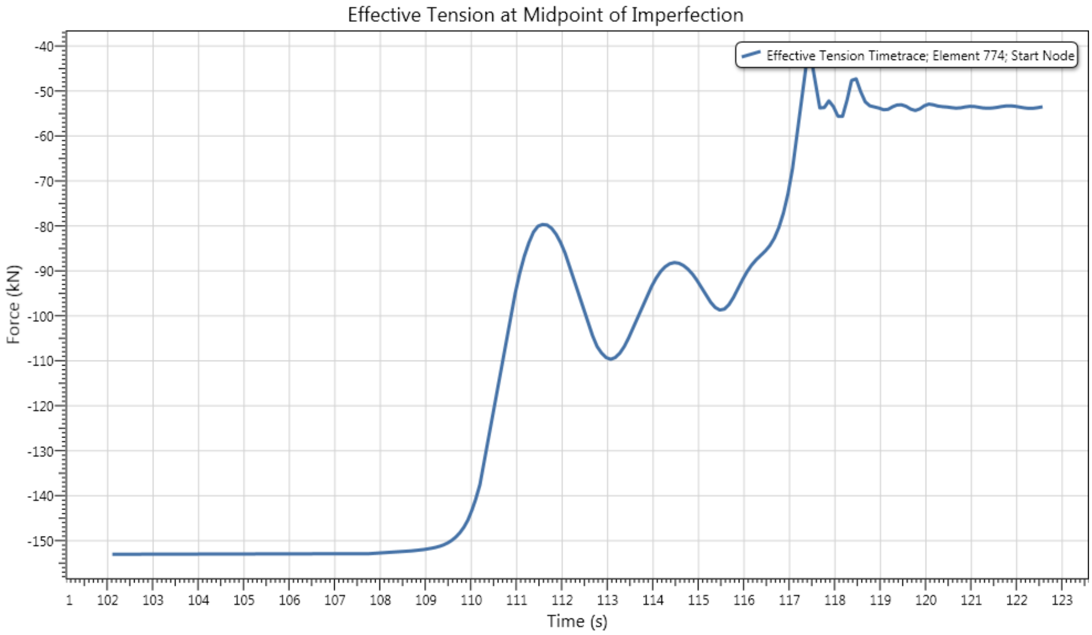
DYNAMIC ANALYSIS

Results from the dynamic analysis are presented in the figures below. The figure below shows a timetrace plot of vertical motion at the mid-point of the pipeline. It is clear that the pipeline lifts a considerable distance off the seabed as the slug passes, indicating that the pipeline is indeed susceptible to upheaval buckling.

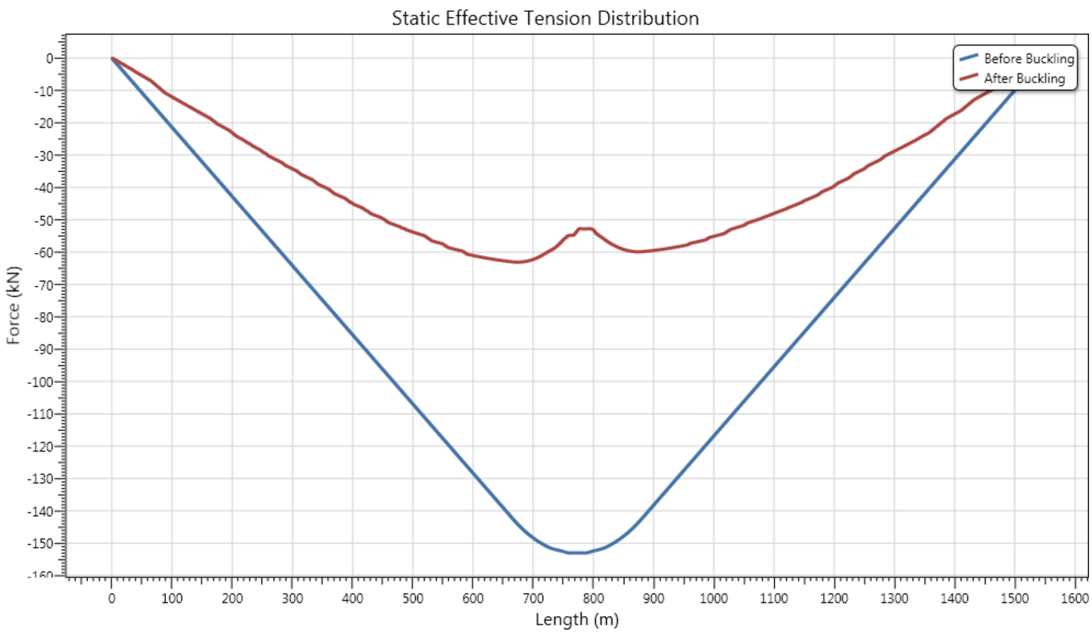


Vertical Motion at Midpoint of Imperfection

The first figure below shows a timetrace plot of effective tension at the mid-point of the pipeline. Consistent with the figure above, the plot indicates a substantial release of compression due to buckling. The second figure below shows a snapshot of effective tension at the end of the analysis, comparing it with the tension distribution before the dynamic analysis. Again a substantial release of compression is indicated, demonstrating that Flexcom has predicted upheaval buckling for this pipeline configuration and loading.



Effective Tension at Midpoint of Imperfection



Pipeline Compression following Onset of Buckling

1.10.7 G - Hybrid Riser Systems

Section G contains some examples of hybrid riser systems, including:

- G01 - Jumper Clashing
- G02 - Jumper Wake Interference
- G03 - Rigid Spool

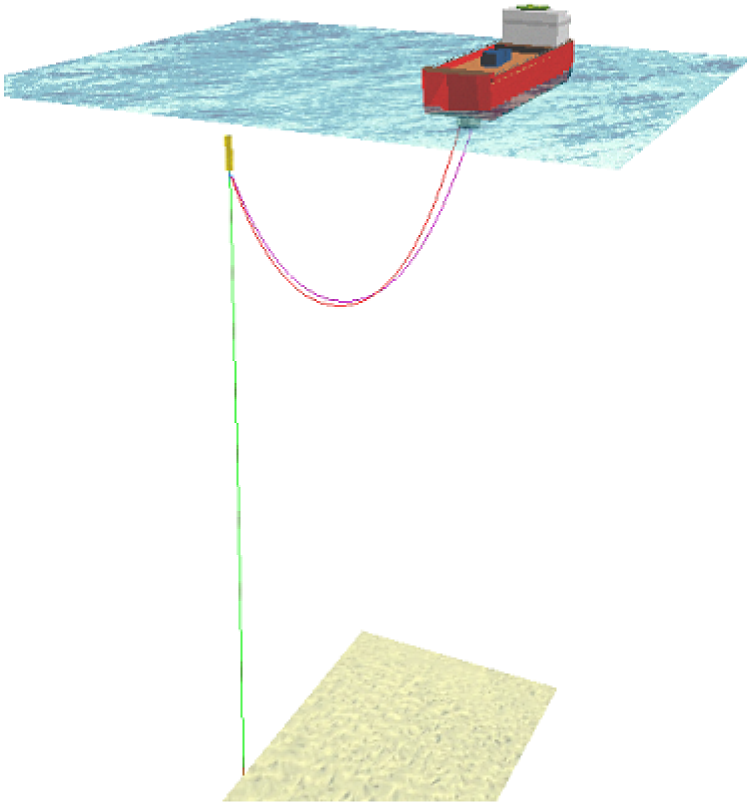
1.10.7.1 G01 - Jumper Clashing

This example considers the analysis of two jumper hoses operating in close proximity as part of the same hybrid riser system. Interference between the jumper hoses is modelled using the clashing facility. The example is divided into the following sections:

- Introduction gives an overview of the analysis.
- Model Summary describes the model in more detail.
- Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions.
- Results presents pertinent results from the various analyses performed.

Introduction

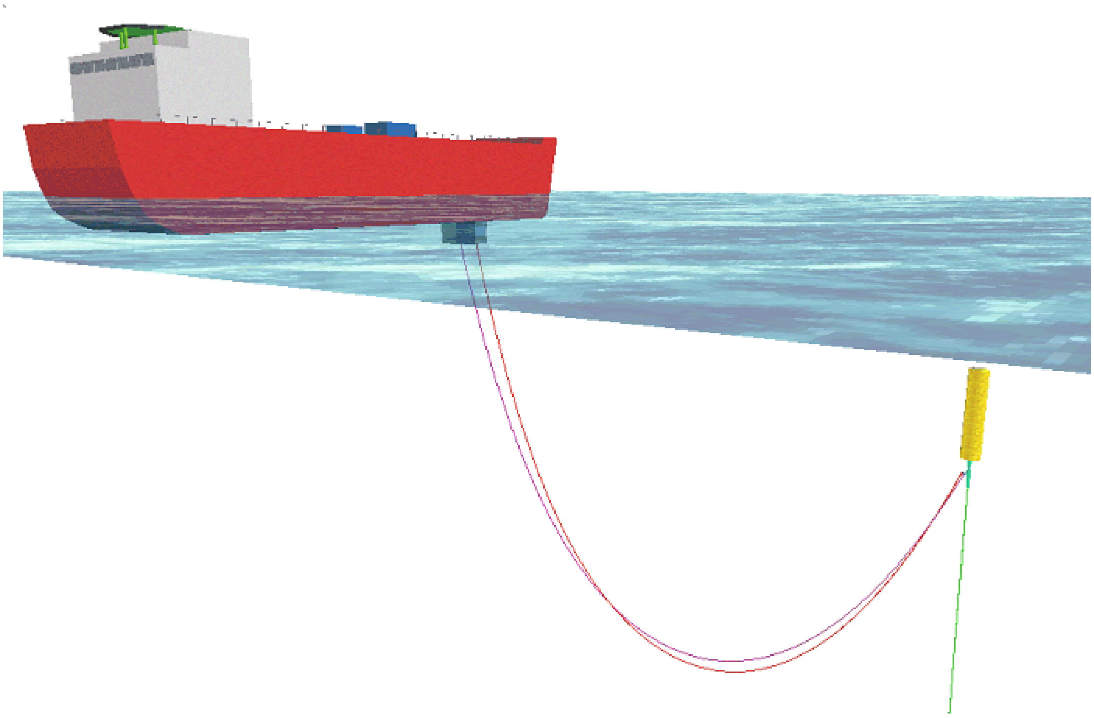
This example considers a hybrid riser system situated in 700m of water. The analysis compares the behaviour of the model, including and excluding the effects of clashing. The analysis sequence consists of an initial static analysis to determine the overall static configuration, followed by a dynamic analysis of the system subject to wave loading and first-order vessel motions. The overall hybrid riser system configuration is shown in the figure below.



Hybrid Riser System Configuration

Model Summary

The model is a top tensioned riser concept, based on a free standing hybrid riser arrangement as shown in the Hybrid Riser System Configuration figure. A close up of the upper section is shown in the figure below, with the region of minimum clearance between the jumper hoses highlighted. The system consists of a single vertical rigid steel riser, anchored to the seabed and tensioned by means of a buoyancy tank with a connection to an FPSO through two flexible jumpers. One of the jumpers takes oil from the wellhead, while the other injects gas into the production stream. The top of the buoyancy tank is positioned approximately 50m below the water surface.



Close up of Jumper Hoses and Buoyancy Tank

Analyses

INITIAL STATIC ANALYSIS

The base of the vertical riser is fixed in all degrees of freedom. The upper ends of the jumper hoses are attached to the vessel. The initial position of the vessel is specified with an undisplaced orientation of 90° to the global Y-axis in the YZ plane. The vessel RAOs are also specified.

The contact regions on the jumper hoses are defined in terms of element sets. When defining contact regions, it is important to specify reasonable data to prevent excessive runtimes, by ensuring that the program is not checking for contact between points that will never approach.

Depending on whether or not clashing is being modelled, a relatively high contact stiffness is specified (100 kN/m) or an insignificant one (specification of a token stiffness facilitates clearance computations for comparison purposes even though contact is effectively ignored).

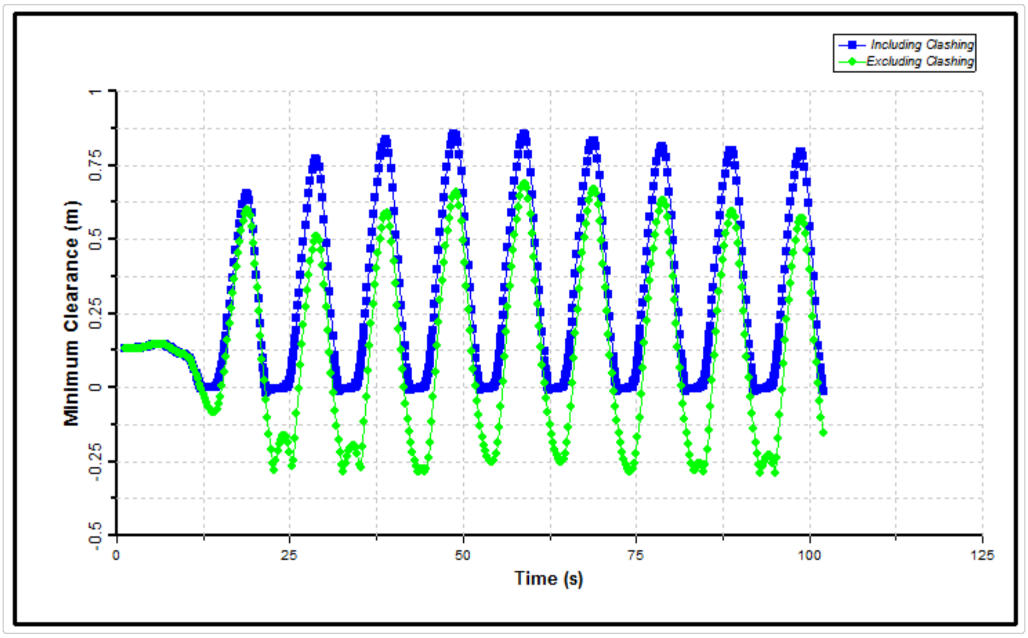
DYNAMIC ANALYSIS

The boundary conditions remain unchanged and are carried through automatically from the initial static analysis. Since vessel boundary conditions and RAO data have previously been input, dynamic motions are automatically applied with the onset of wave loading.

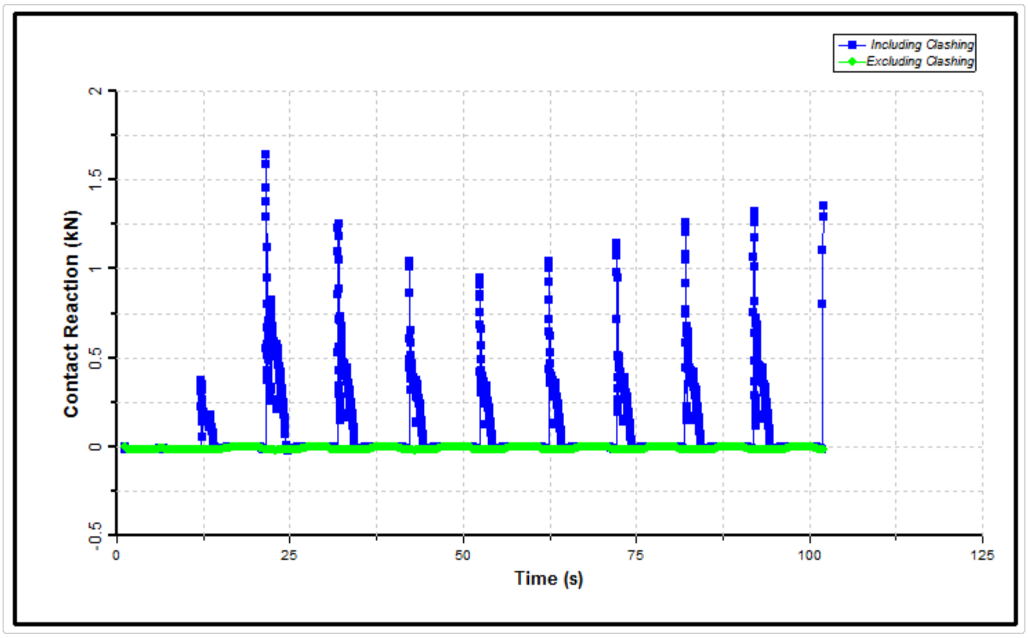
Where clashing effects are not being modelled, a fixed time step of 0.25s is used throughout the analysis. For the analysis which includes clashing, a variable time step is specified, and this allows the interaction between the jumper hoses to be accurately captured. Flexcom continually monitors the relative velocity of approaching lines, allowing the time step to be gradually reduced in anticipation of contact. Once the lines have separated after impact, the time step begins to increase again. This approach facilitates a robust and accurate contact model, while also ensuring an efficient simulation overall.

Results

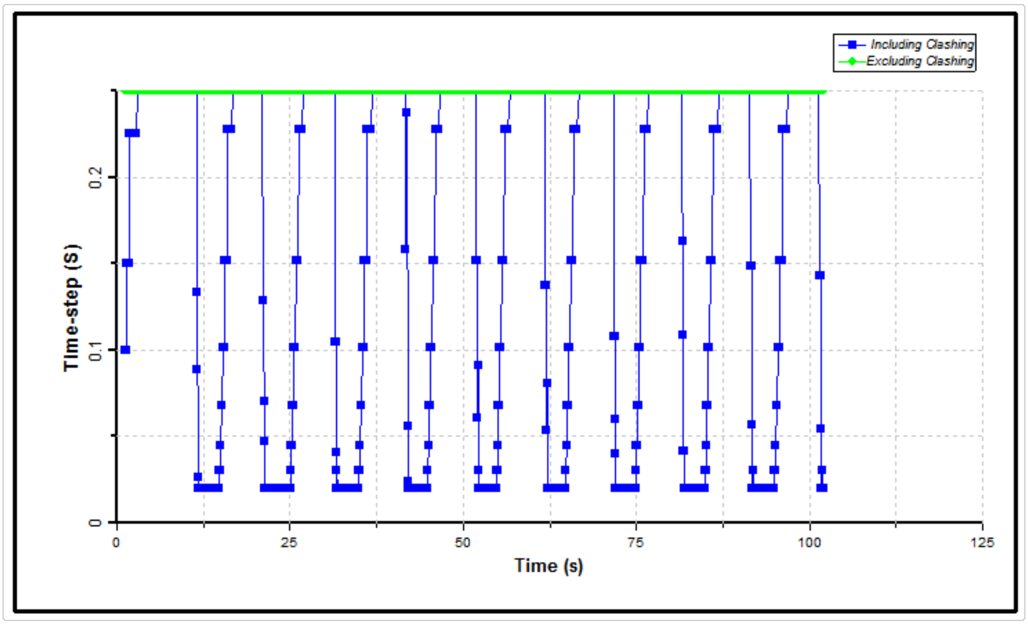
The Minimum Clearance figure presents a plot of the minimum clearance as a function of time. The jumper hoses contact each other intermittently when the effects of clashing are modelled. When clashing effects are neglected, the jumper hoses pass through each other freely. The Contact Reaction figure plots the contact reaction as a function of time. The maximum contact reaction is approximately 1.7kN when the effects of clashing are modelled. When clashing effects are neglected, the contact reaction is always negligible. The Analysis Time Step figure plots the analysis time step as a function of time. This illustrates Flexcom's ability to gradually reduce the time step in anticipation of contact, maintain a relatively small time step during contact, and to increase the time step when separation has developed. The Jumper Hoses in Contact figure shows a close-up of the jumper hoses in contact.



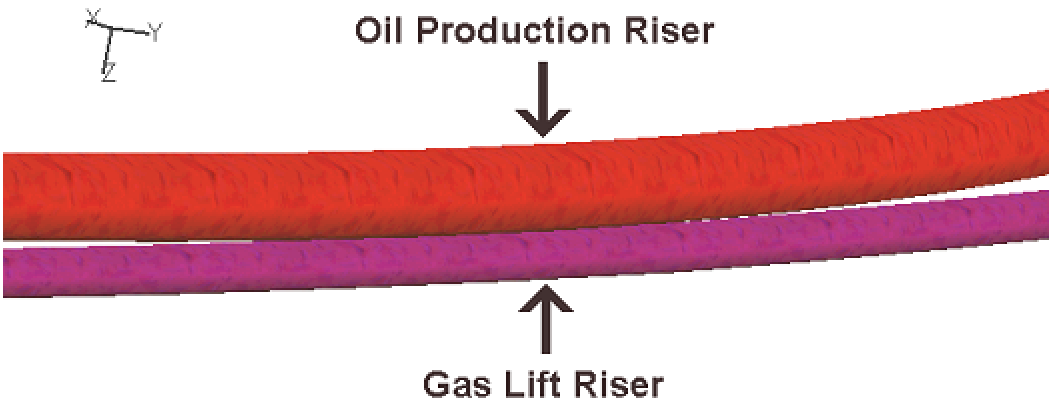
Minimum Clearance



Contact Reaction



Analysis Time Step



Jumper Hoses in Contact

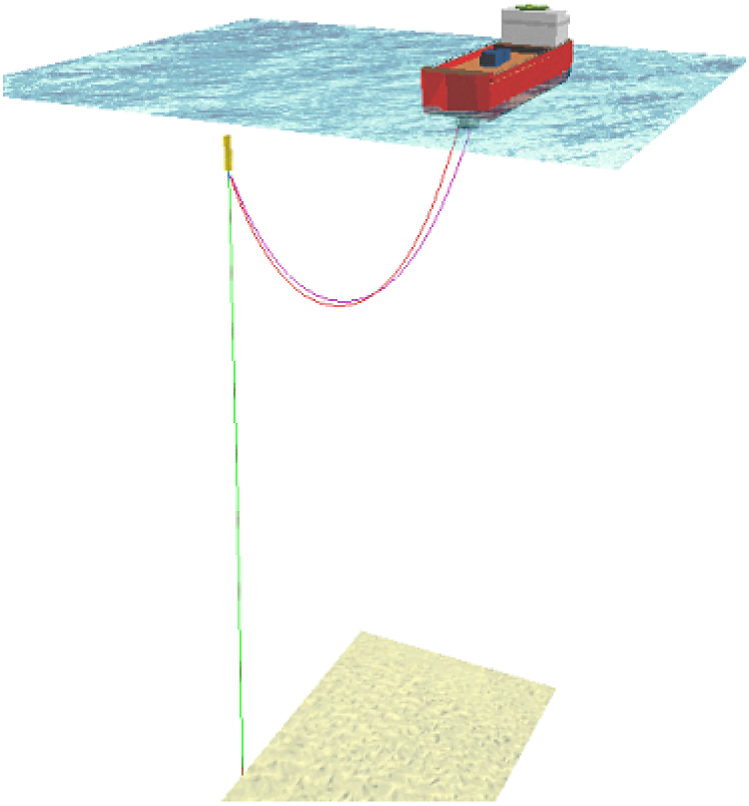
1.10.7.2 G02 - Jumper Wake Interference

This example considers the analysis of two jumper hoses operating in close proximity as part of the same hybrid riser system. Unlike the TLP riser system of the previous example, both hoses are defined within the same Flexcom model as they are not structurally independent. A wake interference analysis is performed in order to accurately determine how the clearance between the jumper hoses changes under the action of current loading. The example is divided into the following sections:

- Introduction gives an overview of the analysis.
- Model Summary describes the model in more detail.
- Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions.
- Results presents pertinent results from the various analyses performed.

Introduction

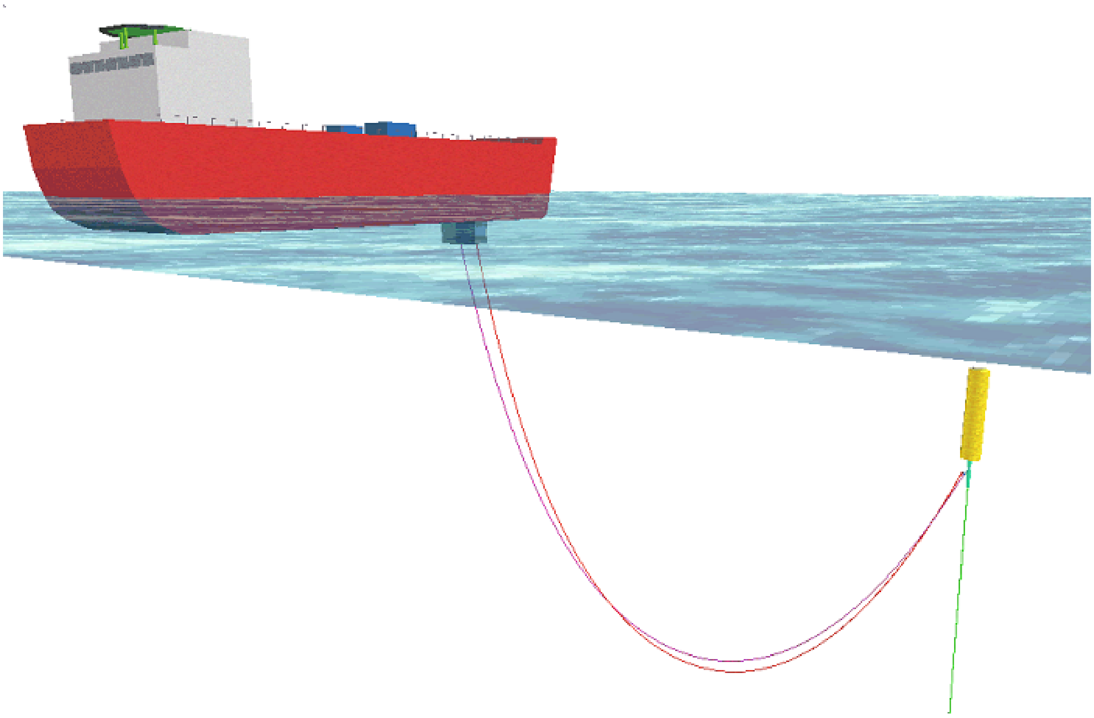
This example considers a hybrid riser system situated in 700m of water. The analysis compares the clearance between the jumpers, including and excluding the effects of wake interference. The analysis sequence consists of an initial static analysis to determine the overall static configuration, followed by a further static analysis in which current loading is introduced. Clearances between the jumpers are determined subsequently using Clear. The overall hybrid riser system configuration is shown in the figure below.



Hybrid Riser System Configuration

Model Summary

The model is a top tensioned riser concept, based on a free standing hybrid riser arrangement as shown in the Hybrid Riser System Configuration figure. A close up of the upper section is shown in the figure below. The system consists of a single vertical rigid steel riser, anchored to the seabed and tensioned by means of a buoyancy tank with a connection to an FPSO through two flexible jumpers. One of the jumpers takes oil from the wellhead, while the other injects gas into the production stream. The top of the buoyancy tank is positioned approximately 50m below the water surface. The wake interference effects are modelled using Blevins, (2005) formulation.



Hybrid Riser System Configuration

Analyses

STATIC ANALYSIS

The base of the vertical riser is fixed in all degrees of freedom. The upper ends of the jumper hoses are fixed in all translational degrees of freedom.

CURRENT ANALYSIS

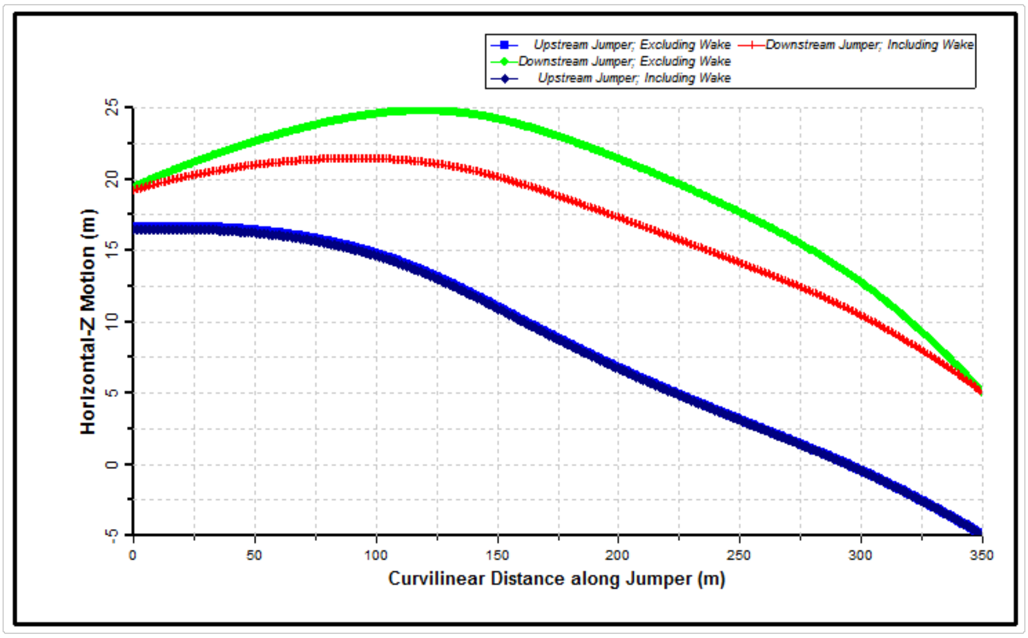
In this analysis the boundary conditions remain unchanged and are carried through automatically from the initial static analysis. A piecewise-linear cross current is also applied to the hybrid riser system. The direction of the current is constant with depth and is aligned with the positive sense of the global Z-axis. Two current analyses are performed, one with the Blevins wake model applied and the other without. The affects of including the wake action in this model are detailed in Results.

CLEARANCE POSTPROCESSING

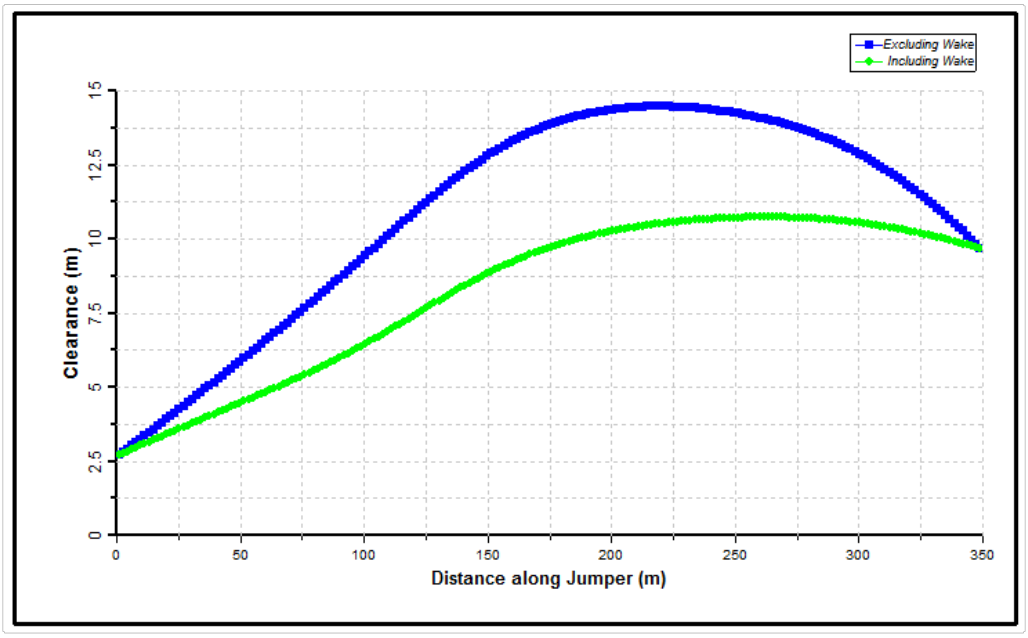
The clearance between the jumper hoses is computed using Clear, the Flexcom postprocessor for clearance/interference calculations. Separate runs are performed for the cases including and excluding wake interference effects.

Results

The first figure below presents the overall configuration of the jumper hoses, illustrating the influence of wake interference effects. Naturally the configuration of the upstream jumper hose is the same in both cases, but the configuration of the downstream jumper hose varies considerably. The second figure presents the clearance distribution between the jumper hoses.



Jumper Hose Configurations



Jumper Hose Clearance Distribution

The table below shows a sample of the detailed wake output for the downstream structure. Here L is the distance in the direction of the current between an element on the downstream structure and the corresponding upstream element. T is the distance normal to the direction of the current between the downstream and upstream elements. It can be seen that the effect of the wake has been to reduce the current operating on the downstream jumper, and also to introduce a lift force. The effect of the lift force is to align the risers in the direction of the current (i.e. reduce the magnitude of T).

Detailed Wake Analysis Output

```
                    *** WAKE INTERFERENCE OUTPUT ***
                    --------------------------------

Wake Model : BLEVINS

Element  Integration  Elevation       L            T        Reduced Current          Lift Forces
            Point                                                          X            Y            Z

  537        2        617.8062     2.8036       1.0873       0.7014       0.2584       0.6380       0.0000
  538        2        616.1911     2.8810       1.0756       0.6871       0.2914       0.7129       0.0000
  539        2        614.5788     2.9571       1.0625       0.6728       0.3282       0.7932       0.0000
  540        2        612.9696     3.0322       1.0488       0.6583       0.3669       0.8750       0.0000
  541        2        611.3636     3.1062       1.0349       0.6438       0.4063       0.9559       0.0000
  542        2        609.7610     3.1794       1.0207       0.6293       0.4458       1.0342       0.0000
  543        2        608.1620     3.2516       1.0063       0.6147       0.4847       1.1089       0.0000
  544        2        606.5666     3.3232       0.9918       0.6000       0.5227       1.1787       0.0000
  545        2        604.9749     3.3940       0.9771       0.5853       0.5592       1.2429       0.0000
  546        2        603.3873     3.4641       0.9623       0.5706       0.5938       1.3006       0.0000
  547        2        601.8037     3.5337       0.9473       0.5558       0.6261       1.3510       0.0000
  548        2        600.2243     3.6027       0.9322       0.5410       0.6557       1.3936       0.0000
  549        2        598.6493     3.6713       0.9169       0.5300       0.6922       1.4488       0.0000
  550        2        597.0790     3.7395       0.9015       0.5248       0.7430       1.5310       0.0000
  551        2        595.5134     3.8072       0.8860       0.5195       0.7935       1.6094       0.0000
  552        2        593.9527     3.8746       0.8702       0.5141       0.8434       1.6832       0.0000
  553        2        592.3971     3.9415       0.8544       0.5086       0.8924       1.7519       0.0000
  554        2        590.8469     4.0082       0.8384       0.5031       0.9400       1.8149       0.0000
```

1.10.7.3 G03 - Rigid Spool

This section describes a sample rigid spool model, and demonstrates how internal fluid and slug flow is readily modelled using Flexcom. The model is also parameterised to facilitate scalability, illustrating the benefit of parameters and equations during model building.

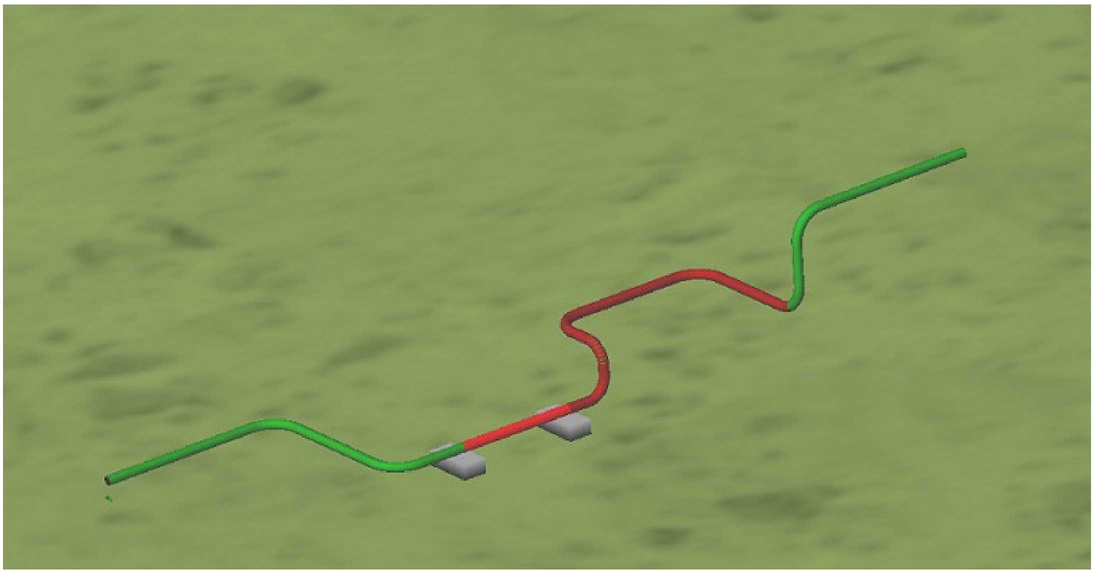
The overall layout of this example is as follows:

- Introduction gives an overview of the spool system.
- Model Summary describes the model setup in more detail.
- Analyses briefly describes the various analyses performed, discussing the internal fluid and slug loading to which the spool is subjected.
- Results presents pertinent results from the analyses performed.

Introduction

Flexcom provides a comprehensive internal fluid modelling capability. Stationary internal fluids, uniform steady state internal fluid flow and multi-phase slug flow may all be modelled. If slugs are present in an analysis, they are displayed visually in the structural animation.

This example considers the analysis of a rigid spool subjected to internal fluid and slug flow. The system is situated at a depth of 1500m, and consists of nine straight sections and eight 90-degree bends. Two contact surfaces are used to simulate the supports which help to reduce the possibility of seabed contact. The figure below shows the overall configuration of the model, with a slug of flowing through it. The example consists of a static analysis, subject to gravity and buoyancy loads only, followed by a dynamic analysis subject to internal fluid and slug loading.



Spool Configuration

If you are interested in further information regarding the effects of slug flow in rigid spools, please refer to the technical publication listed in References (Kavanagh et al., 2013).

Model Summary

GENERAL

In the case of this model, the background fluid is actually gas (shown in green), while the slug is actually oil (shown in red). The flow direction is from left to right in the figure below. Note also that the slug colour is slightly lighter at the tail end – denoting that the tail of slug has a lower density than the main body of the slug – fluid density in the tail section decreases linearly from oil to gas. This approach is used to capture the effect of the oil and gas mixing together in a transition region at the back of the slug. The slug speed is also reduced slightly as it passes around each bend (3% decrease), leading to an overall decrease in flow velocity across the length of the slug. This aspect demonstrates the program's ability to model time varying slugs which facilitate a more realistic slug flow definition than that of idealised flow of constant velocity.

GEOMETRY

Several inputs relating to the overall spool geometry are defined initially using the parameters feature. The straight sections are modelled using the lines feature, while the bends are modelled using explicitly defined nodes and elements. Each bend is subdivided into 10-degree sub-sections, with the ends of each adjacent section/element sharing a common node. The straight and curved sections are connected together via the nodal equivalence feature. Both ends of the spool are restrained in all degrees of freedom, while the intermediate supports are modelled using guide surfaces.

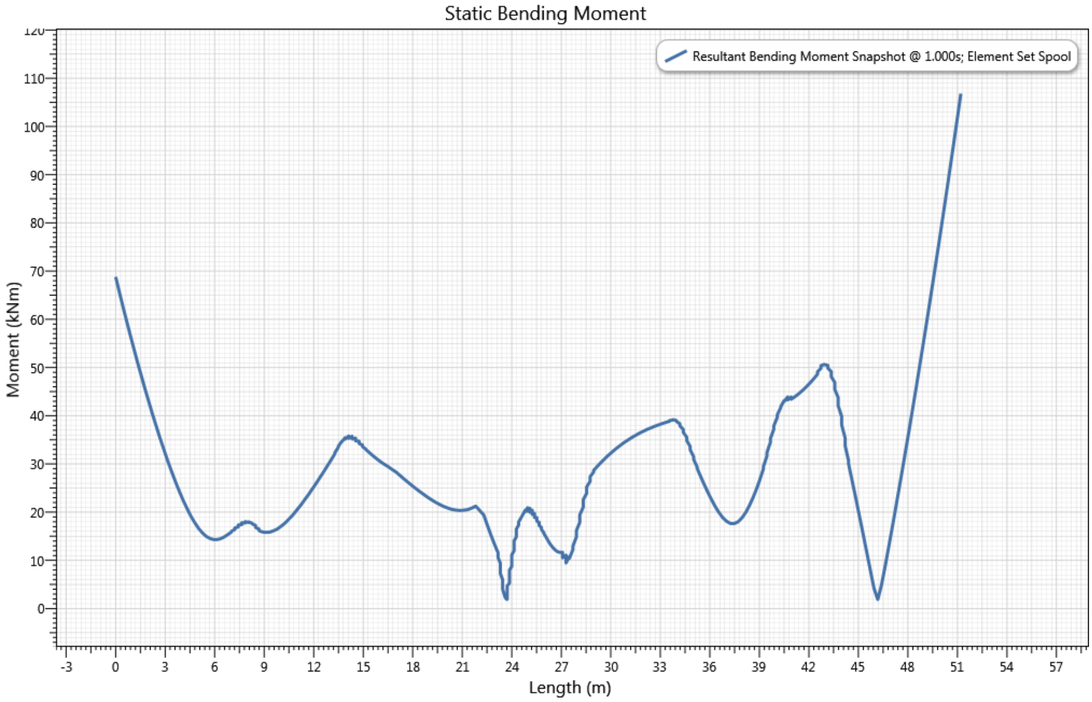
Analyses

An initial static analysis is performed in which the model is subjected to gravity and buoyancy loads only. This is followed by a dynamic restart analysis in which five successive oil slugs are passed through the spool. The time delay between each slug is such that a gap of three times the total slug length is maintained between each consecutive slug.

Each slug is modelled as 25 metres long. To model the decaying density of the slug towards its tail end, two slugs of different density are used, with a time delay between them. The main body of the slug is 20m in length and this portion has uniform density. A secondary 5m slug is used to model the tail portion, whose density varies linearly from oil to gas.

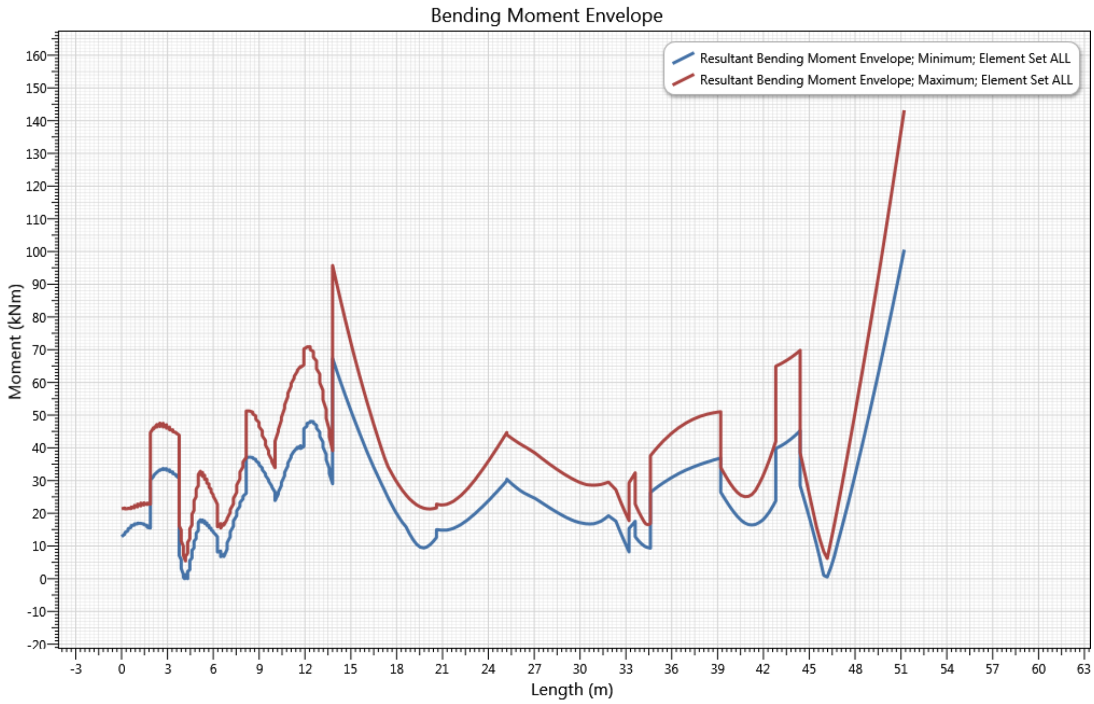
Results

The following figures contain a selection of results. The figure below shows the static bending moment distribution in the spool. The highest moments tend to occur at each end due to the applied fixations and the self-weight of the structure.



Static Bending Moment

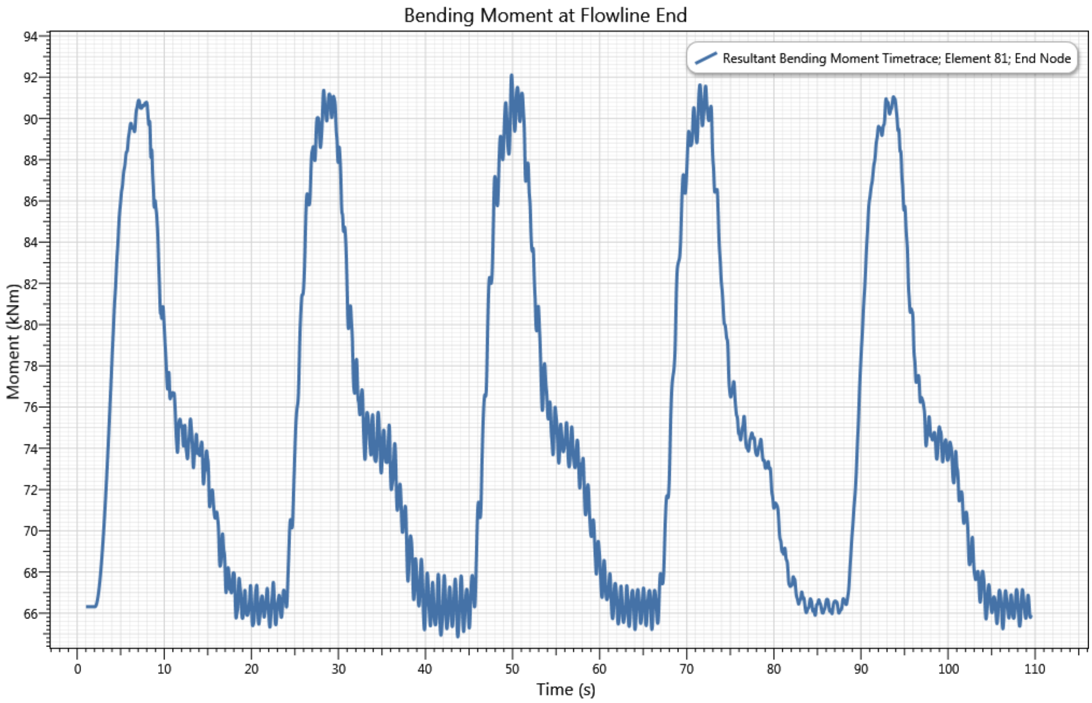
The figure below shows the bending moment envelope throughout the spool model during the dynamic analysis. Significant bending moment variations are evident at all locations.



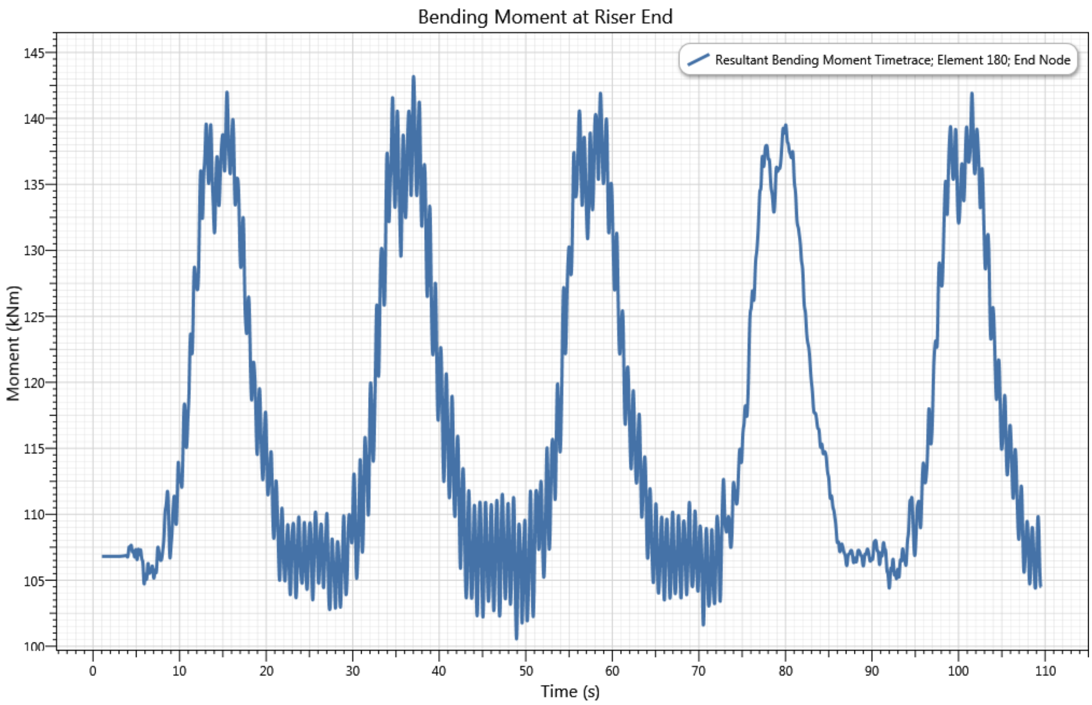
Dynamic Bending Moment Envelope

The first and second figures below show time histories of the bending moment at either end of the spool. Elevated moments tend to coincide with the times when the entire slug is present in the model, and when it is positioned nearest to one of the ends.

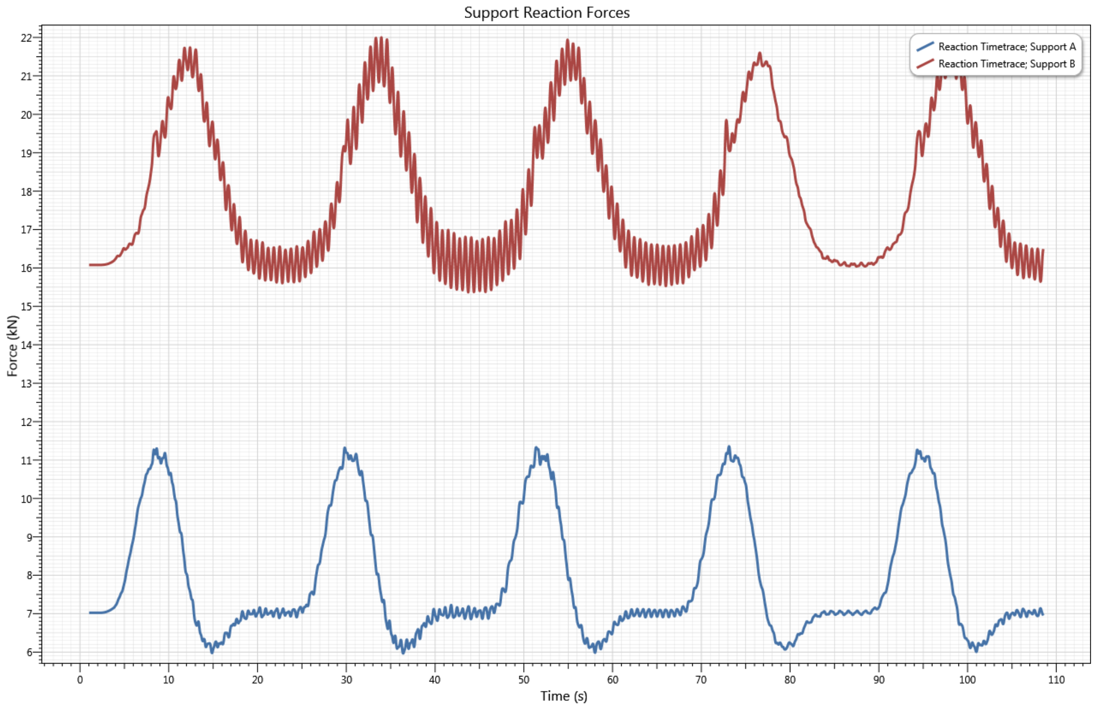
The final figure shows time histories of the reaction forces at the support locations. Although both supports have similar elevations, the second support (the one nearest to the spool centre) bears about twice the load of the first one.



Bending Moment (Flowline End)



Bending Moment (Riser End)



Support Reactions

1.10.8 H - Installation Analysis

Section H contains some examples of installation analysis, including:

- H01 - Bundle Tow-Out and Installation
- H02 - J-Tube Pull-In
- H03 - Articulated Stinger
- H04 - Pipe Laying
- H05 - Steel Pipe Installation with Plastic Deformation

1.10.8.1 H01 - Bundle Tow-Out and Installation

This example describes a tow-out and installation analysis, and demonstrates a number of features of Flexcom specifically suited to such analyses. The overall layout of this example is as follows:

- Introduction gives an overview of the tow-out and installation analysis, and notes some of the more important features of Flexcom which are relevant to the analysis.
- Model Summary describes the model in more detail.
- Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions to which the riser bundle is subjected.
- Results presents pertinent results from the various analyses performed and discusses their significance.

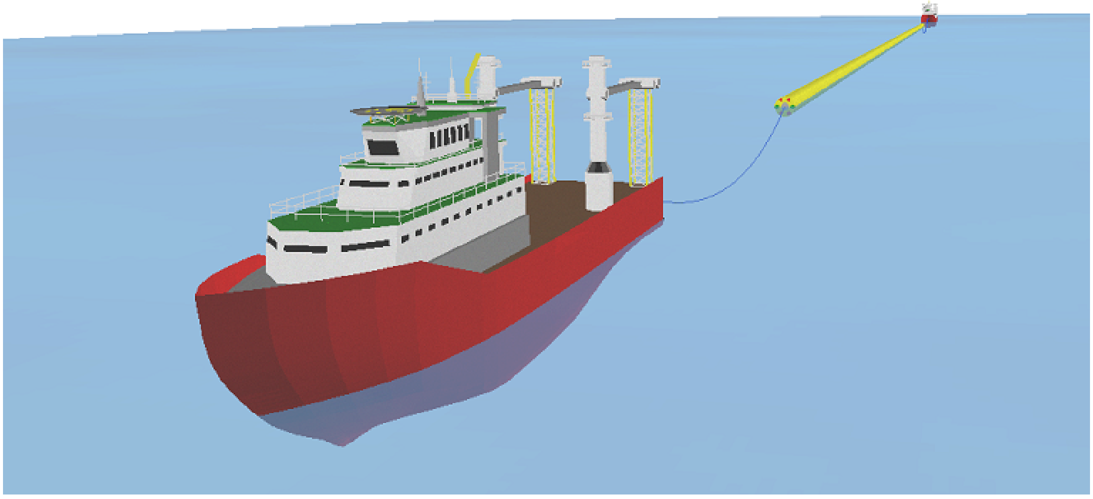
Introduction

This example considers a tow-out and installation of a riser bundle. The analysis is performed in two phases. Firstly, a shallow water tow configuration is considered, where the forward tow speed is simulated by an in-line uniform reverse current, and the riser bundle is subjected to regular wave loading. Secondly, a deep water installation procedure is analysed in which the riser bundle is upended from an initial floating configuration into a vertical position.

This example illustrates a number of important modelling features in Flexcom which are useful in this type of analysis. Specifically, the following capabilities are demonstrated:

- Modelling of partial submergence of sections. The buoyancy formulation in Flexcom means that partial submergence of an element cross-section can be accurately modelled. This is particularly useful when, as in this case, elements float on the sea surface.
- Winch element facility. The winch capability previously demonstrated in the SCR Transfer example is used again here in the deep water installation phase of the analysis.

A schematic of the system at the start of the second, deep water phase is shown in the figure below.



Tow-Out Configuration

Model Summary

This example considers the tow-out and installation of a riser bundle. The riser bundle is 850m in length and is attached at both ends to tug vessels by 150m cables. The shallow water tow-out takes place in a water depth of 35m at a constant tow speed of 1.8m/s. An initial static analysis is first performed with the bundle restrained vertically to obtain an approximate starting configuration. The initial constraints are subsequently removed in a quasi-static analysis, and the riser bundle attains its floating position at the mean water line. A further quasi-static analysis is then performed where the forward tow speed is simulated by an in-line uniform current. Finally, regular wave loading is introduced in a dynamic analysis.

The deep water installation procedure takes place in a water depth of 875m. A 75t dead anchor weight is attached to front end of the riser bundle to facilitate the upending.

An initial static analysis with vertical constraints is again performed, and is again followed by a quasi-static analysis in which the riser bundle reaches its equilibrium position. Two dynamic analyses are then performed to reposition the riser bundle into a vertical configuration.

Analyses

SHALLOW WATER TOW-OUT

Initial Static Analysis

Vessel boundary conditions are specified at the cable extremities in the translational degrees of freedom. Additional restraints in the vertical direction are applied at regular intervals to restrain the riser bundle in the mean water line region.

Quasi-Static Analysis

The vertical constraints on the riser bundle are removed. All of the other boundary conditions are unchanged. The riser quickly attains its equilibrium position, with significant mass and stiffness damping specified to minimise transients.

Current Analysis

A uniform current of 1.8m/s is applied to simulate the forward tow speed. The BCs remain unchanged and are carried through automatically from the quasi-static analysis. Again a quasi-static analysis with significant damping is specified.

Dynamic Analysis

The riser bundle is subjected to regular wave loading, of amplitude 6m and period 11s. The BCs remain unchanged and carry through automatically from the (first) quasi-static analysis. No first order vessel motions are applied in this case. The analysis is run for six wave periods (66s), with results statistics sampled over the last two.

DEEP WATER INSTALLATION

Initial Static Analysis

Vessel boundary conditions are specified at the cable extremities in the translational degrees of freedom. Additional restraints in the vertical direction are applied at regular intervals to restrain the riser bundle in the mean water line region.

Quasi-Static Analysis

The vertical constraints on the riser bundle are removed. All of the other BCs are unchanged. The riser quickly attains its equilibrium position, with significant mass and stiffness damping specified to minimise transients.

Dynamic Analyses

Two dynamic analyses are performed to reposition the riser bundle into a vertical configuration. The boundary conditions remain unchanged and are carried through automatically from the initial static analysis. The dynamic analysis conditions differ as follows.

In the first analysis, 725m of cable is winched out from the lead tug at 0.5m/s. Then 130m of cable is winched in by the rear tug at 0.25m/s. While the winching is in progress, the lead tug moves a distance of 630m towards the rear tug at a constant speed of 0.3m/s.

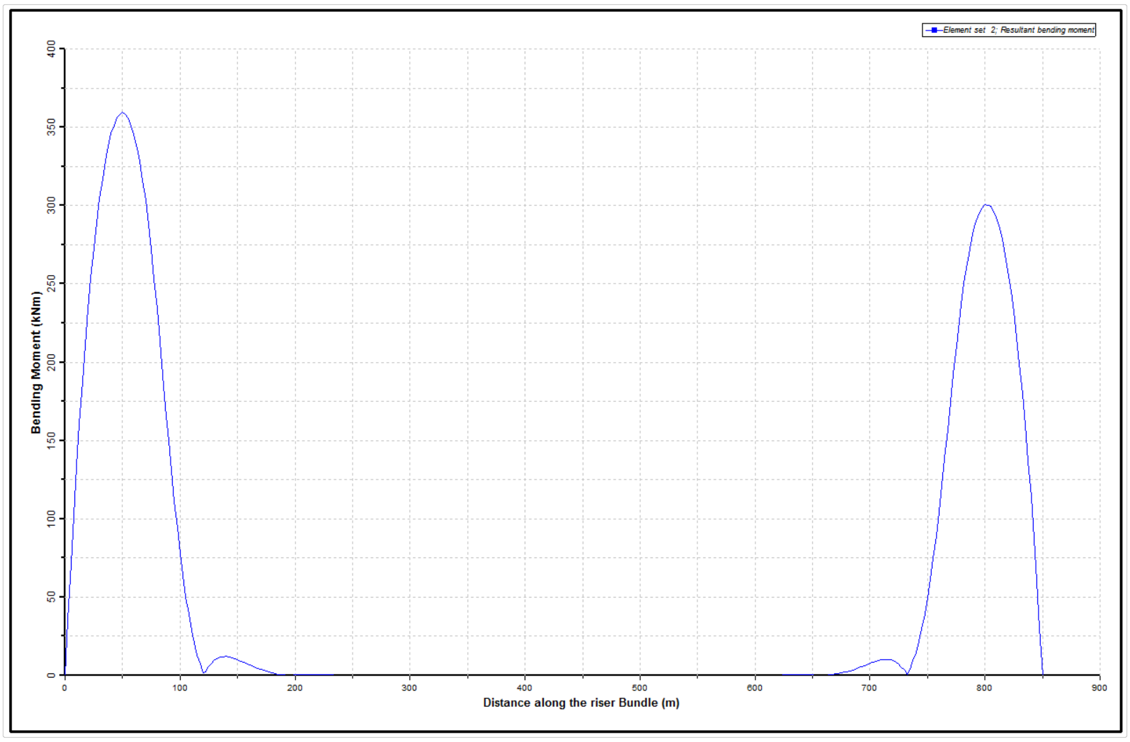
In the second run, a further 175m of cable is winched out from the lead tug at 0.2m/s. At the same time the rear tug moves a distance of 190m towards the lead tug at a constant speed of 0.2m/s.

Results

SHALLOW WATER TOW-OUT

Current Analysis

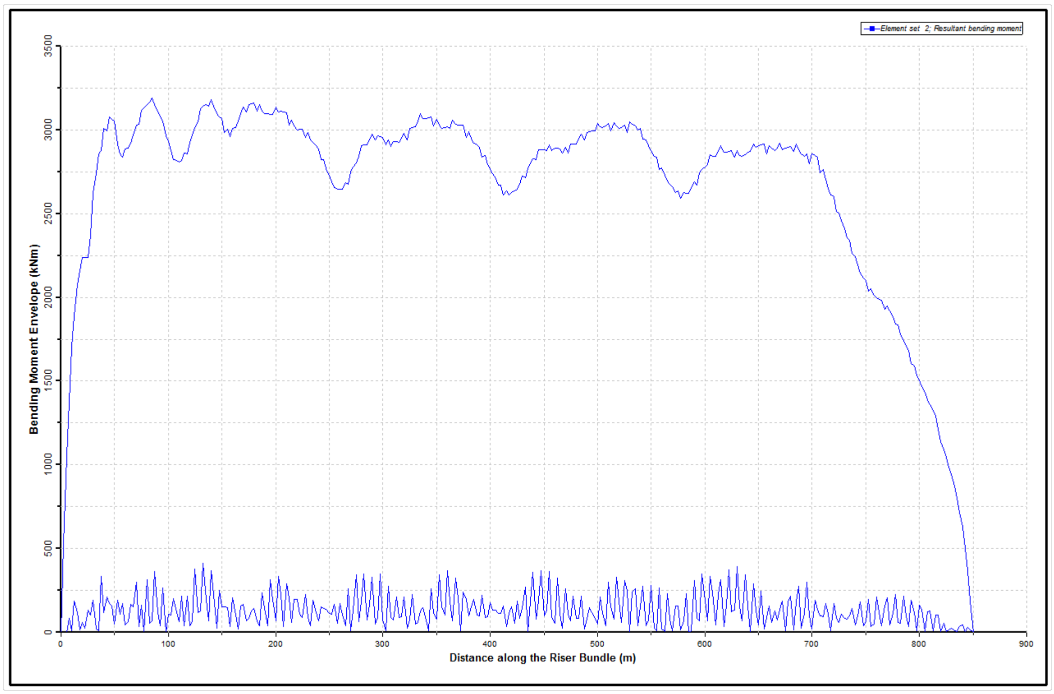
The figure below shows a plot of the bending moment distribution in the riser bundle in the shallow water tow-out configuration. The maximum bending moment is approximately 360kNm, occurring approximately 50m from the lead cable connection point. The majority of the bundle experiences little or no bending as the bundle floats on the MWL.



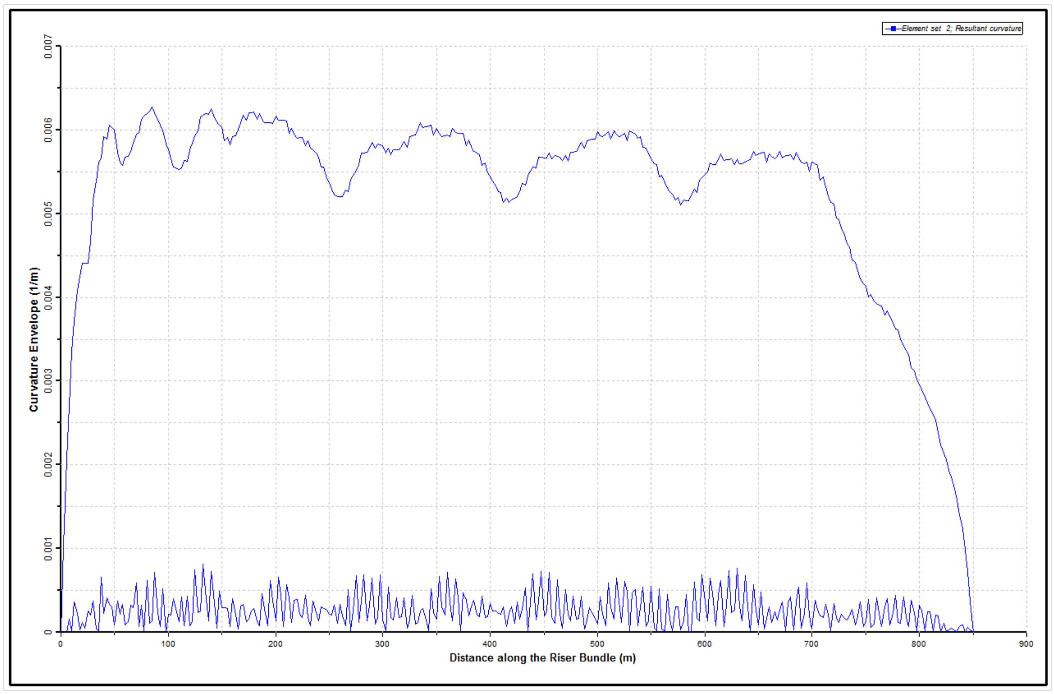
Static Bending Moments

Dynamic Analysis

Results from the dynamic analysis are presented in the figures below. The first figure shows envelopes of bending moment from the last two wave periods. The maximum dynamic bending moment experienced by the riser bundle is over 3000kNm. The second figure shows envelopes of curvature, which naturally show a similar profile to the bending moments.



Dynamic Bending Moments

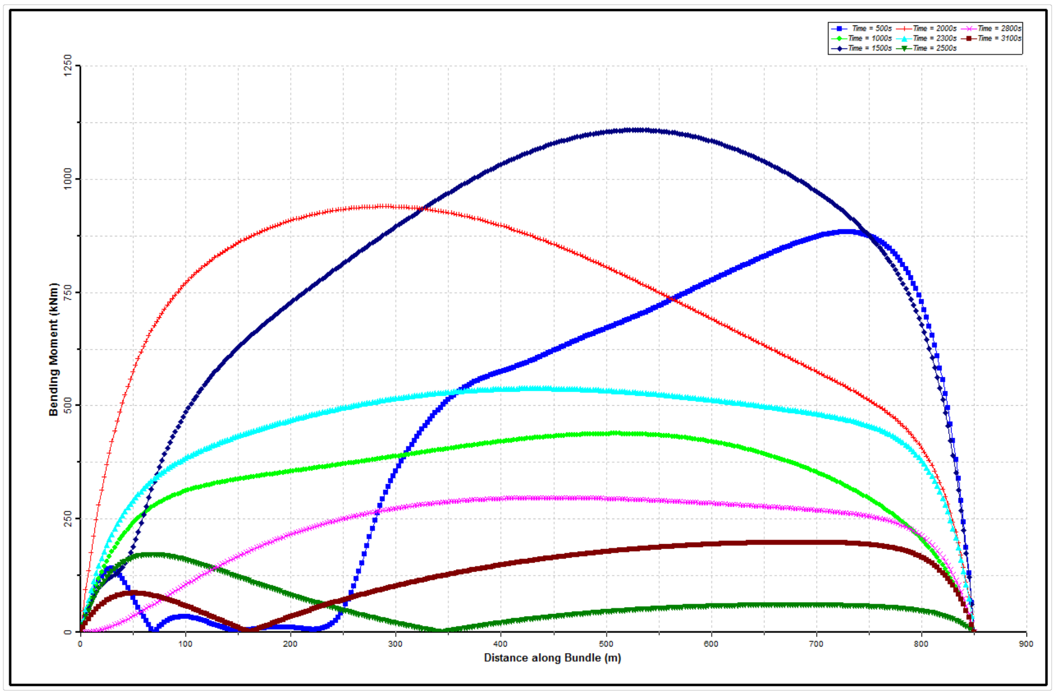


Peripheral Lines Dynamic Curvatures

DEEP WATER INSTALLATION

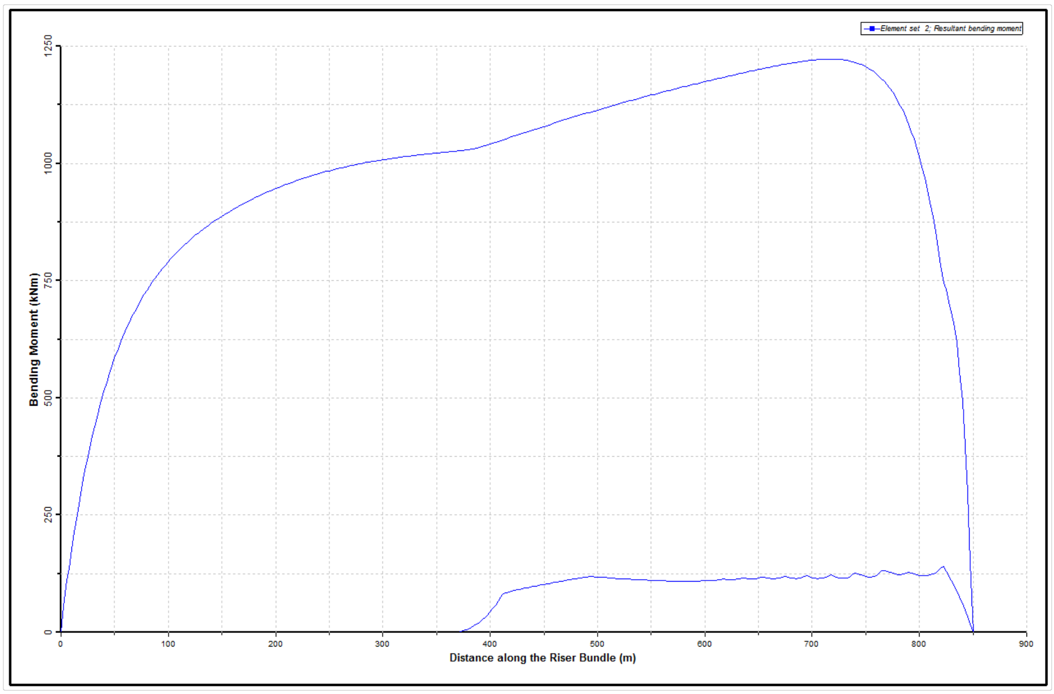
Dynamic Analyses

The figure below compares snapshots of bending moment distribution in the riser bundle at various stages of the deep water installation. The moments experienced by the bundle are most severe in the first dynamic analysis stage, between 1000s and 2000s simulation time.



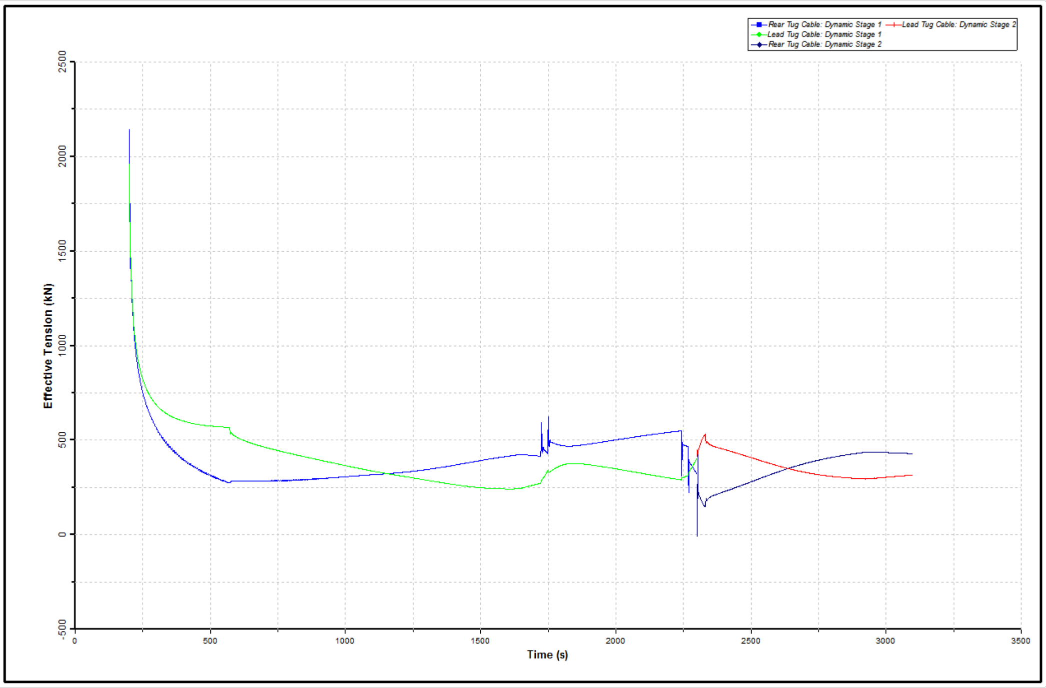
Bending Moments during Installation

The figure below shows envelopes of bending moment during the first dynamic analysis. The maximum bending moment experienced by the riser bundle during the upending process is approximately 1250kNm.



Dynamic Bending Moments

The figure below compares timetrace plots of effective tension in the rear and lead tug cables. The tension in both cables is greatest at the beginning of the dynamic analyses, because most of the bundle is floating at MWL: there is little uplift due to buoyancy, and the full weight of the anchor is borne by the cables. The tension in both cables decreases sharply as cable is paid-out from the lead tug and the foremost portion of the bundle becomes submerged. The tension continues to decrease steadily in the lead cable as the anchor and bundle weight is transferred to the rear cable. For a brief period at approximately 1720s simulation time, which is the transition period between lead tug cable winch out and rear cable winch in, the distance between the tugs continues to reduce, and the lead cable tension increases and the rear cable tension reduces temporarily. This effect is again evident towards the end of the first dynamic analysis, as the tug vessels continue to close together in the absence of winching. The lead cable tension reduces steadily as the upending procedure approaches completion, with the rear cable tension conversely increasing over the same period.



Winching Cable Tensions

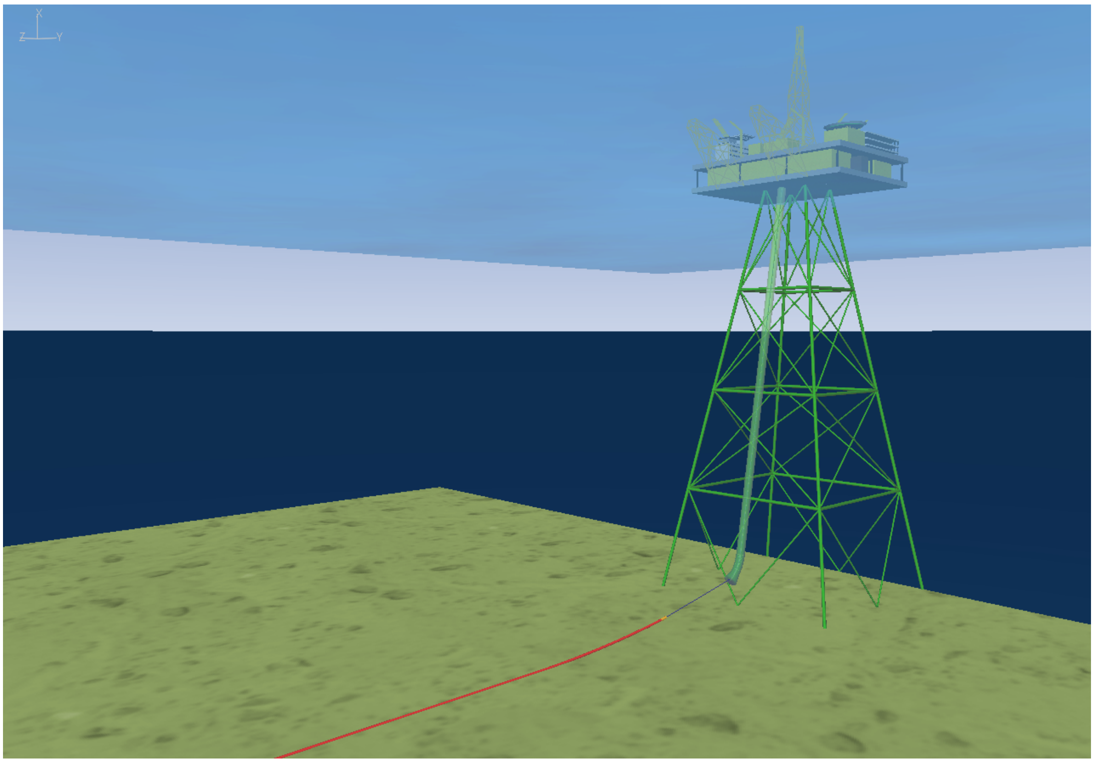
1.10.8.2 H02 - J-Tube Pull-In

This example considers the insertion of a flexible riser through a J-tube. Contact between the flexible and J-tube is modelled using the pipe-in-pipe modelling facility. The example is divided into the following sections:

- Introduction gives an overview of the analysis.
- Model Summary describes the structure from a modelling viewpoint in Flexcom. Important modelling techniques for this type of analysis are noted, as well as relevant Flexcom features and capabilities.
- Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions.
- Results presents pertinent results from the various analyses performed.

Introduction

This example considers a J-tube connected to a fixed platform in shallow water. A flexible riser is pulled through the J-tube using a winch mounted on the platform. The analysis sequence consists of an initial setup analysis to determine the overall static configuration, followed by a dynamic analysis of the system as the flexible is pulled through the J-tube. The overall system configuration in its initial state is shown in the figure below.



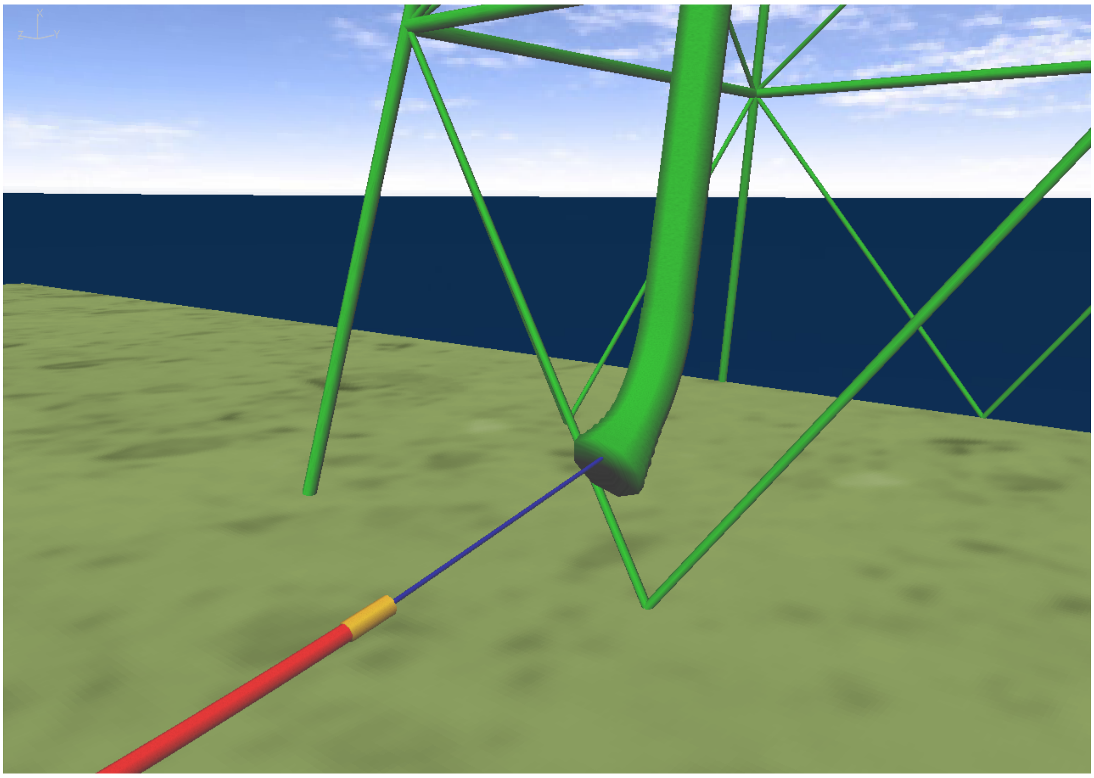
Initial Configuration

Model Summary

GENERAL

The platform itself is considered rigid, so it is included for visual effect only, and does not affect the finite element computations. The analysis commences from a point where the flexible is lying flat on the seabed, with the winching cable already positioned within the J-tube. The winching cable is connected to the flexible at one end via a rigid end fitting, passes through the J-tube, and connected to a winch at its upper end. In order to maintain reasonable element lengths in the contact region, the actual winching process is modelled by a single (initially very long) winch element above the platform. As this element contracts over time, the upper end of the flexible is gradually pulled along the J-tube.

Contact between the flexible and J-tube is modelled using the pipe-in-pipe modelling facility. As the degree of relative axial motion between the components is obviously significant, sliding connections are used to model the interaction between the inner and outer sections. No pipe-in-pipe sections are defined, as the J-tube is flooded with seawater, so the external ambient fluid for the flexible remains unchanged as it passes through the tube. A close up of the bellmouth region is shown in the figure below, with the various components clearly visible, as the upper end of the flexible approaches its entry to the J-tube.



Close up of J-Tube Region

PIPE-IN-PIPE MODELLING

A useful feature in Flexcom is the ability to model pipe-in-pipe configurations. Internal and external sections are modelled separately, and interact with each other by means of special stiffness connections simulating both linear and non-linear resistance to relative motion. Pipe-in-pipe connections can be considered to operate like springs, whether linear or non-linear. However these springs do not directly connect the two nodes (in many cases the nodes are collinear, at least initially, and so cannot be connected by a spring). What happens is that a spring stiffness is added directly into the stiffness matrix at locations corresponding to the appropriate motions of the two nodes. The direction of the spring is computed as being normal to the structure at the outer node.

Sliding connections are similar to standard ones, with the added advantage that the connections are interchangeable. This is appropriate for modelling scenarios where there is significant relative axial motion between the inner and outer pipes. Based on the proximity of the various inner and outer nodes at the beginning of the analysis, the program creates an initial set of (effectively standard) connections between the inner and outer pipes. The program continually monitors the relative axial locations of the inner and outer nodes over the course of the analysis, and the set of connected inner and outer nodes is updated as and when required.

WINCH ELEMENTS

Winch elements are normal beam-column elements which have the unique property that their lengths can vary during an analysis. In a dynamic analysis, the variation in length is defined in terms of a maximum winch velocity and a winching time sequence. The time sequence consists of (i) a ramp-up time when winch velocity is increased from zero to the maximum value; (ii) a time during which the velocity remains at this maximum, and (iii) a ramp-down time when the velocity returns to zero. The operation in a static analysis is less complex, with an overall change in length being applied linearly from the analysis start time to the end time. Winch elements can be used, for example, in pipelaying applications, for example in simulating the transfer of an SCR from a lay vessel to a TLP or semi-sub.

Analyses

INITIAL ANALYSIS

As the J-tube is rigidly connected to the platform (which itself is considered rigid), both ends of the J-tube are fixed in all translational degrees of freedom. The upper end of the winch element is also constrained, simulating the attachment point with the platform.

The model is first analysed quasi-statically for a short duration to help initiate contact between the winch cable and the J-tube. This is then followed by a static analysis to ensure the model has reached static equilibrium before proceeding to the pull-in stage.

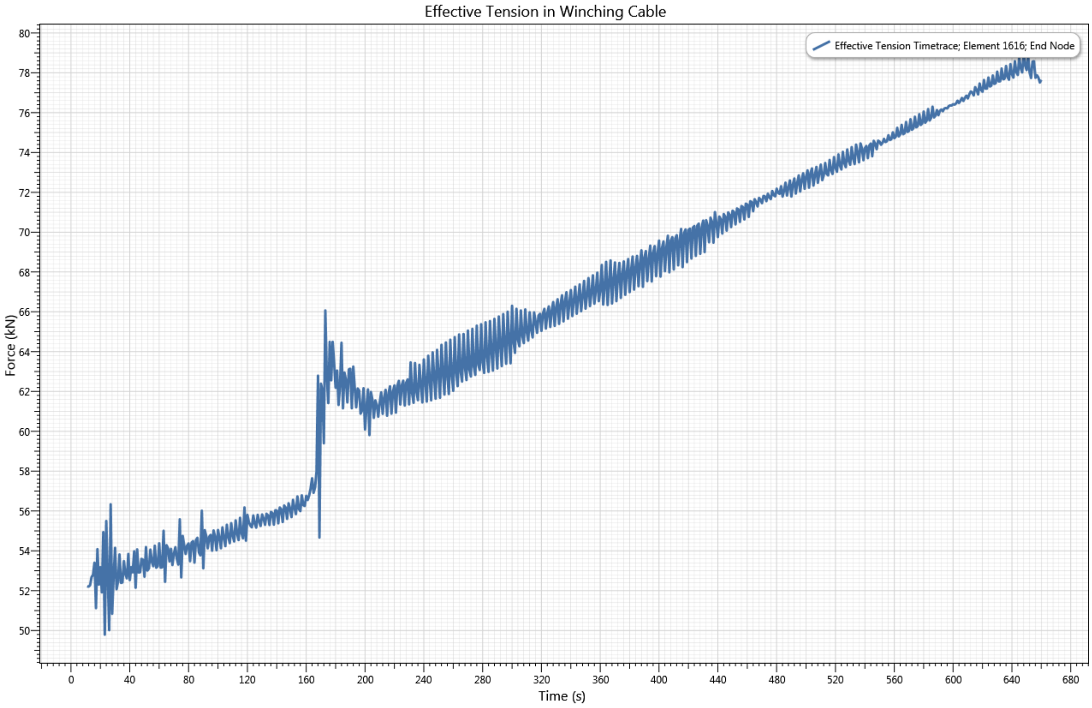
DYNAMIC ANALYSIS

The pull-through of the flexible takes place at 0.1m/s, and the entire process occurs over 650 seconds. Note that these values are for demonstration purposes only, and may not necessarily represent actual installation times in reality.

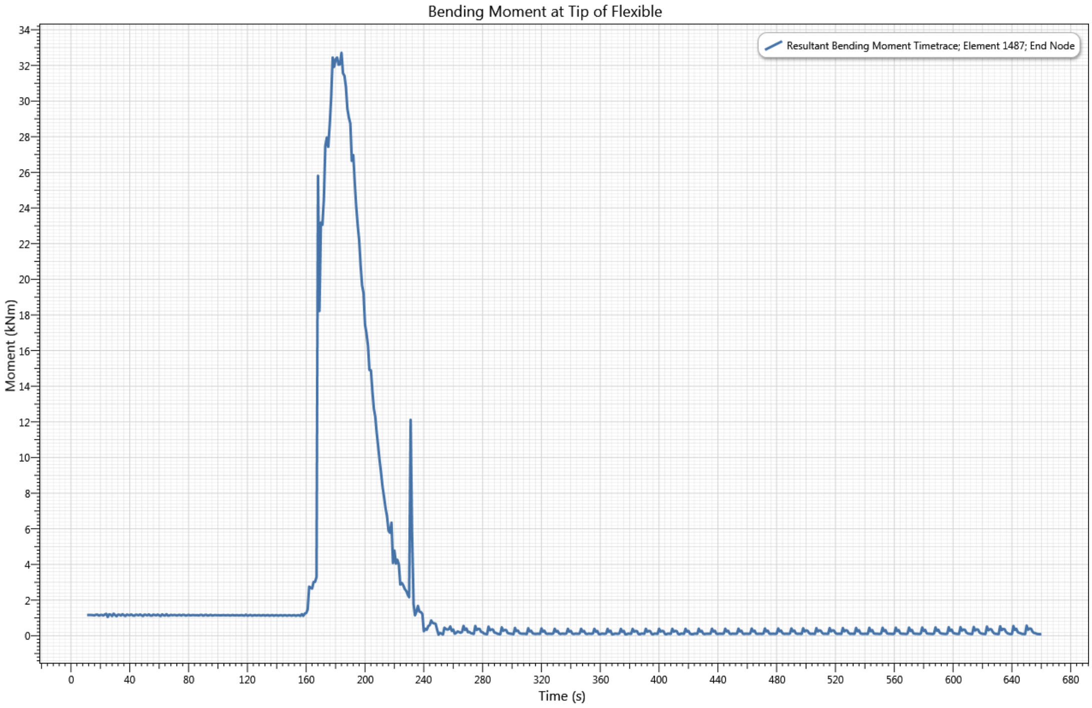
Results

The first figure below presents a plot of the effective tension in the winching cable. In general, the overall trend is that the tension increases in a fairly linear manner, which reflects the increasing weight of flexible suspended from the platform. The graph fluctuates continually, due to the node-based nature of the pipe-in-pipe connections – essentially the flexible moves in short strain lines along the rigid elements of the J-tube, rather than following a truly smooth curve. It is also interesting to note the slight increase in tension (deviating from the overall trend) during the analysis – this corresponds to the period where the end fitting moves around the curved section of the tube into the straight vertical section.

The second figure below presents the bending moment at the tip of the flexible (nearest the end fitting). The bending moment is naturally highest as the front end passes through the curved section of the J-tube, but otherwise quite low – e.g. when the flexible is located within the straight upper section of the J-tube.



Effective Tension in Winching Cable



Bending Moment at Tip of Flexible

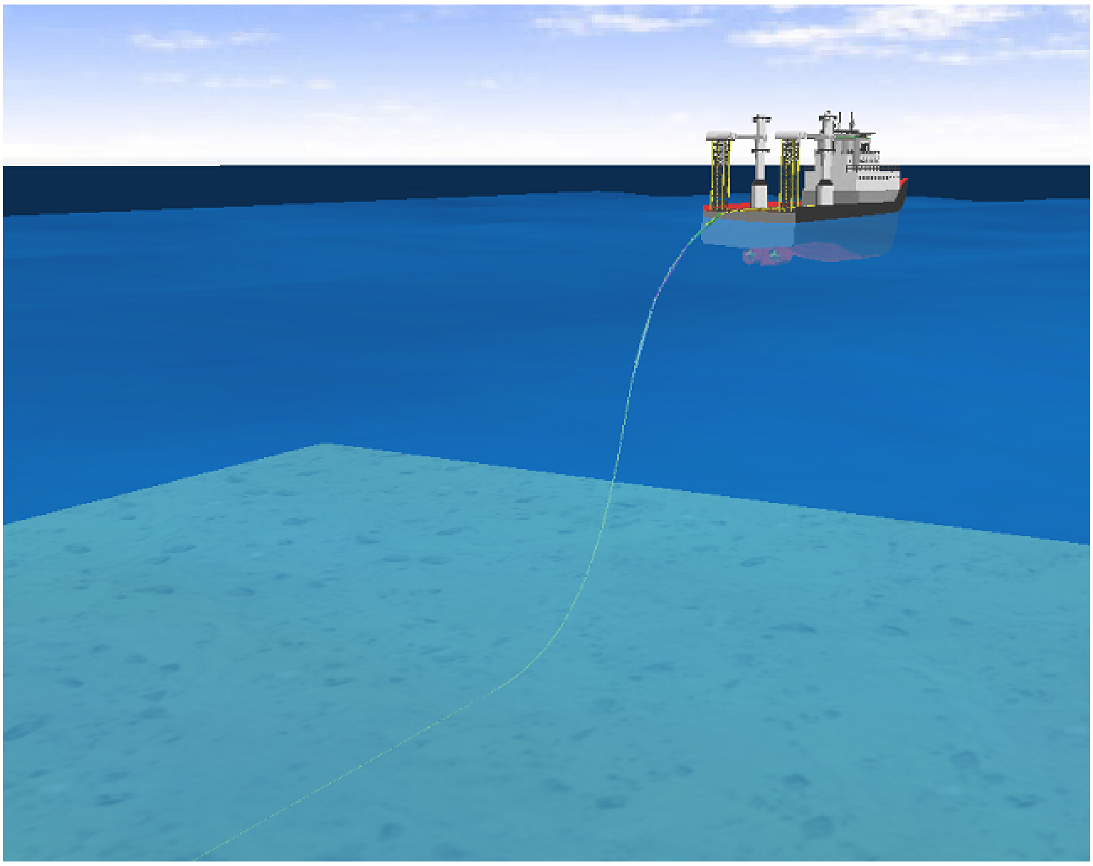
1.10.8.3 H03 - Articulated Stinger

This example considers the installation of a pipeline in an S-Lay installation configuration using an articulated stinger. Contact between the pipeline and stinger is modelled using guide surfaces. The example is divided into the following sections:

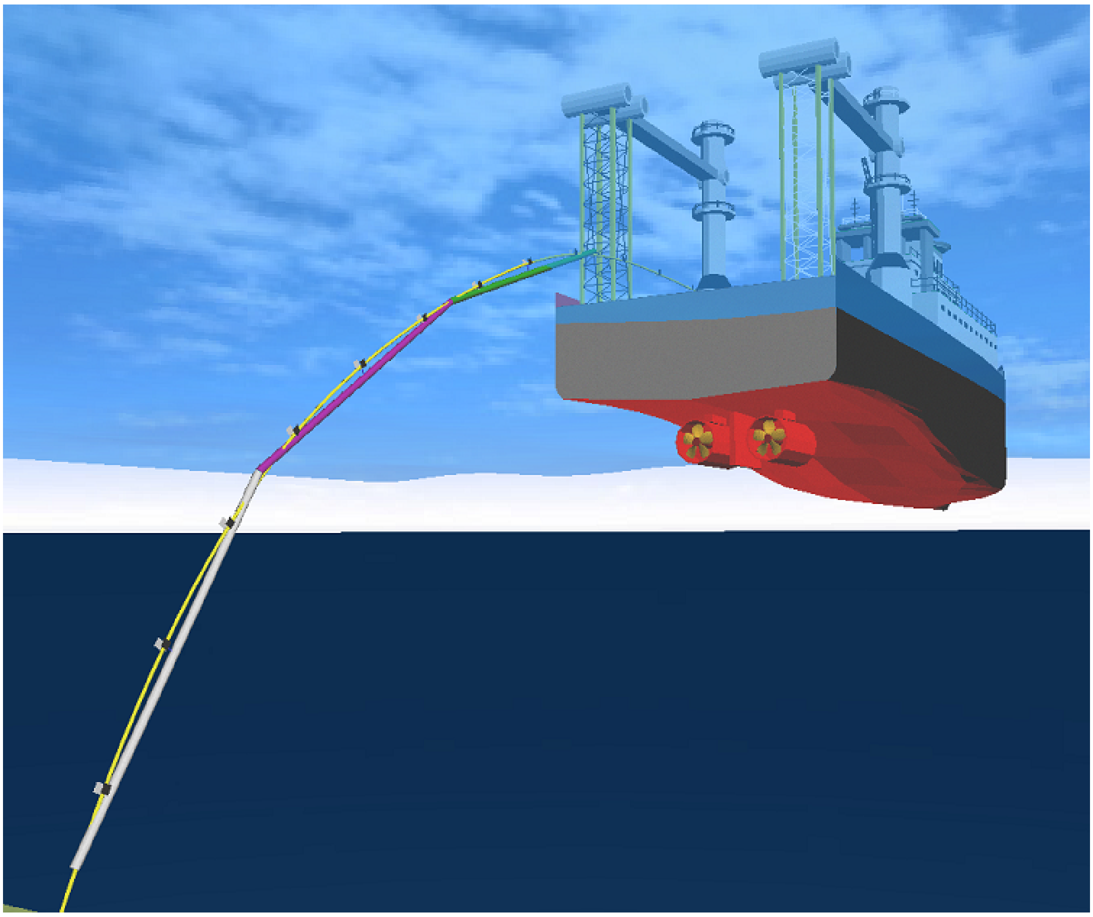
- Introduction gives an overview of the analysis.
- Model Summary describes the structure from a modelling viewpoint in Flexcom. Important modelling techniques for this type of analysis are noted, as well as relevant Flexcom features and capabilities.
- Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions.
- Results presents pertinent results from the various analyses performed.

Introduction

This example simulates an articulated S-Lay installation scenario. The model includes a lay vessel equipped with an articulated S-Lay stinger. The stinger is comprised of three stinger sections with three U-shaped roller boxes on each section. Hinges are placed at the start of each section in order to allow the sections to move independently of one another. The pipeline runs along a set of vessel supports before passing over the stinger sections and down onto the seabed. The pipeline is restrained on the vessel deck by a tensioner. The analysis sequence consists of initial static/quasi-static analyses in order to determine the overall static configuration of the system, followed by a dynamic analysis in which regular wave loading is applied. The overall system configuration is shown in the first figure below. A close up of the stinger region is shown in the second figure.



System Configuration



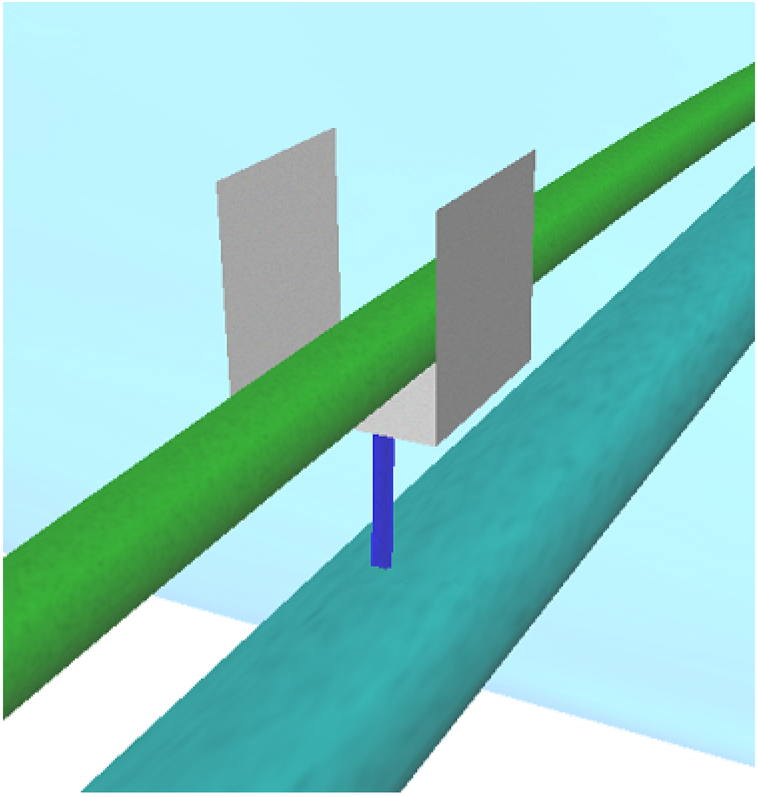
Stinger Region

Model Summary

GENERAL

The articulated stinger is composed of three rigid sections which are linked together using flexible connections. The stinger is also connected to the vessel at its upper end, and is free to oscillate vertically subject under the downward load of the pipeline weight and the uplift provided by the stinger's buoyancy.

Rollerbox supports are located at regular intervals along the stinger and vessel, and each support is modelled using a combination of three flat guide surfaces in the form of a “U” shape. The figure below shows a close up of one of the rollerboxes. The supports on the vessel move rigidly with the vessel as it translates and rotates in response to the applied wave loading. The supports on the stinger are associated with structural nodes which are connected to the stinger sections, so the motions of these supports actually form part of the solution (i.e. their motions are not prescribed via the vessel RAO data).

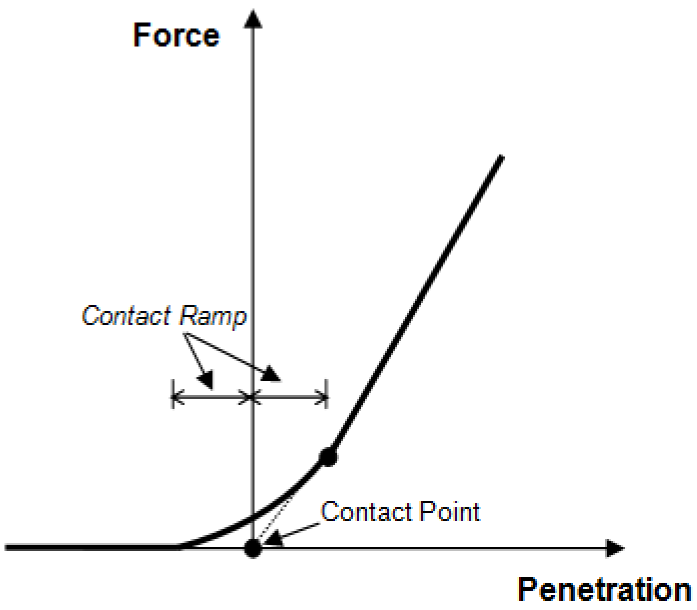


Rollerbox Model

The pipeline is modelled as a series of consecutive lines running from the seabed, over the stinger and vessel, and connected to the vessel using a tensioner (non-linear spring) element. In order to ensure an efficient solution, potential contact regions are associated with each support by defining relatively small contact element sets.

CONTACT MODELLING

An important aspect of the contact algorithm is how Flexcom handles the interaction between the structure and the guide surfaces. Each node that is potentially in contact with a guide surface may be considered to be connected to the surface by means of a non-linear spring which has an orientation normal to the surface. Theoretically, the force in the spring is zero until the node makes contact with the surface, at which point the linear spring stiffness is activated. While this formulation is theoretically correct, it is not especially robust, so Flexcom can use a modified spring force-deflection curve shown in the figure below instead. Adopting a non-linear approach tends to lessen the effects of instantaneous impact, reduce high frequency noise, and consequently allows the solution to proceed at larger time step increments.



Non-Linear Contact Model

Further details are available in Contact Modelling.

Analyses

INITIAL STATIC ANALYSIS

The lower end of the pipeline is fixed in all translational degrees of freedom, and the fixed end of the tensioner element is connected to the vessel.

In order to initiate contact between the pipeline and stinger, some additional constraints are applied to the ends of each stinger section. This allows the pipeline to be supported by the stinger, but the stinger is artificially restrained rigidly to the vessel for the time being.

The pipeline is also temporarily connected rigidly to the vessel, thereby rendering the tensioner element provisionally redundant.

STATIC RELEASE ANALYSIS

In obtain a static equilibrium configuration of the pipeline and stinger, the temporary constraints on the stinger are removed. This allows the stinger to find an equilibrium position under the influence of gravity and buoyancy loading.

QUASI-STATIC RELEASE ANALYSIS

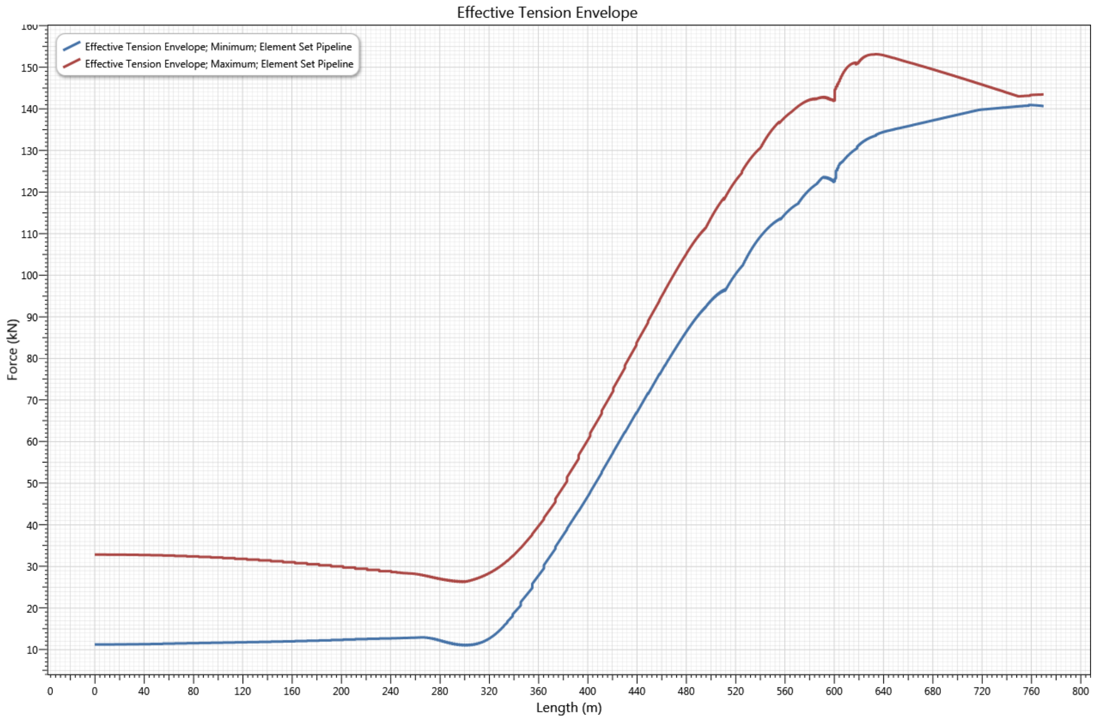
The additional constraints temporarily applied to the tensioner element are removed. This allows the pipeline to move in the axial direction subject to the constant restoring force provided by the tensioner element.

DYNAMIC ANALYSIS

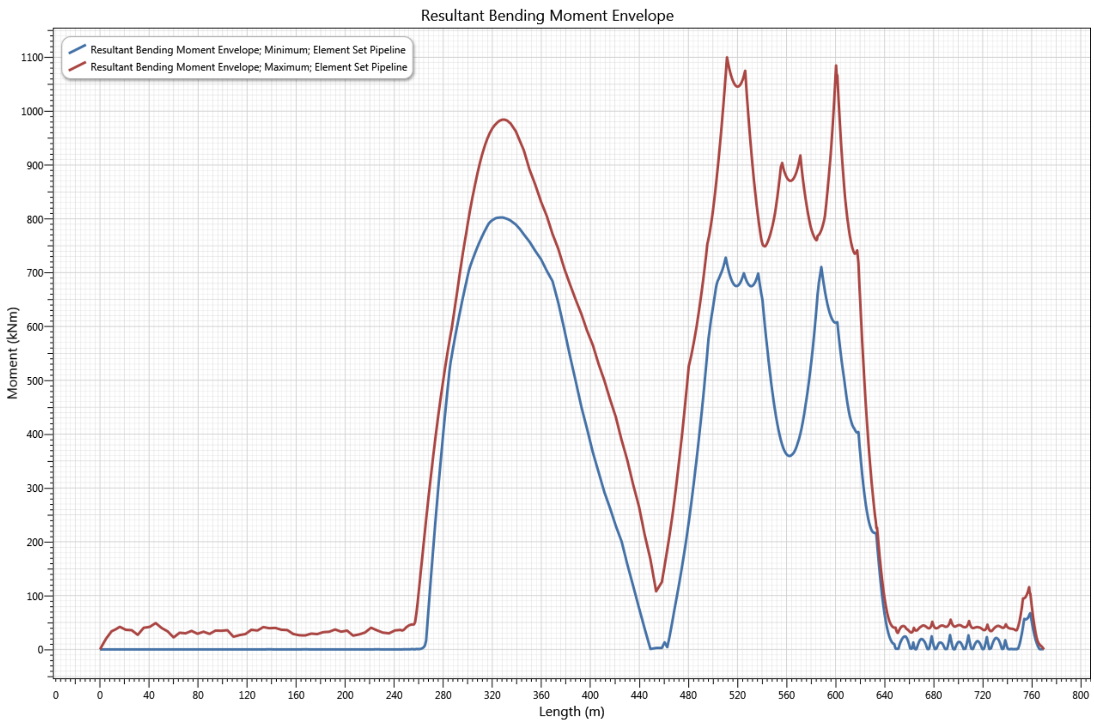
The boundary conditions remain unchanged and are carried through automatically from the preceding analysis. A sample regular wave loading is applied, of 2m amplitude and 10s period. The incident wave heading is applied at 90°, simulating a beam sea condition.

Results

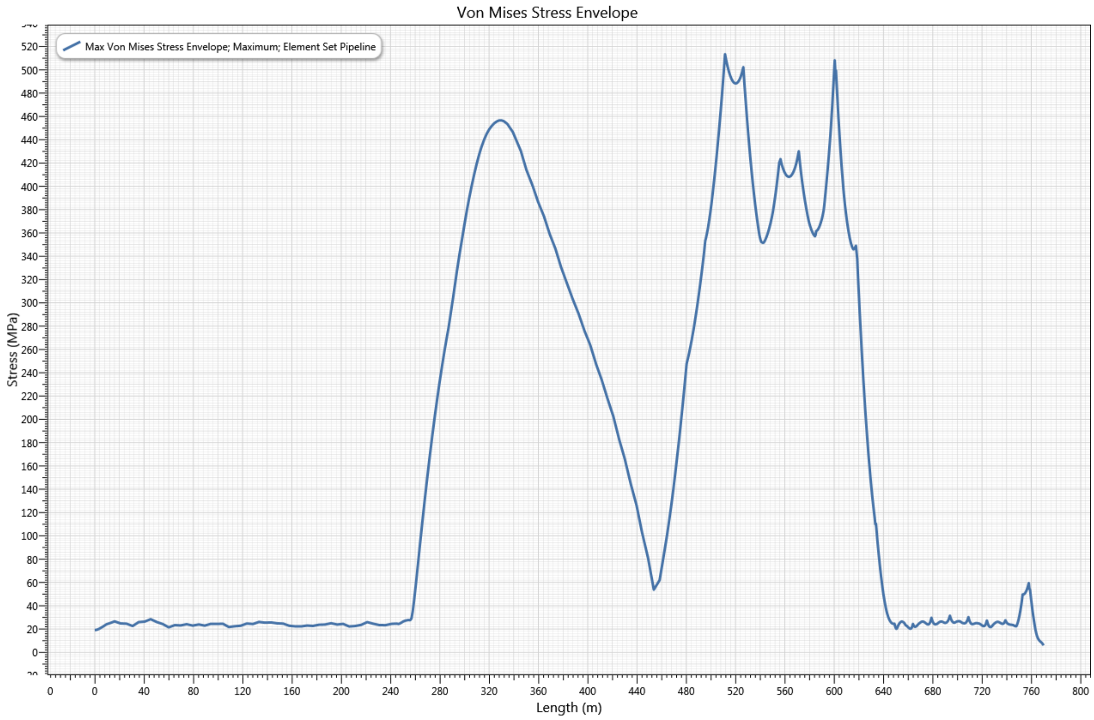
The figures below present envelope plots of effective tension, bending and von Mises stress respectively. At the tensioner end of the pipeline the maximum and minimum values are fairly consistent. This is consistent with expectations as the tensioner applies a constant force equal to the static tension. The tension extremes at all points along the pipeline are always positive, so the pipeline never goes into compression. The bending moment distribution varies significantly in the overbend (stinger) region due to wave induced motions, as the individual stinger sections are free to rotate about their hinged connections. Considerable variations occur in the sagbend region also, and the touchdown zone in particular. The final figure plots a sample time history of the contact reaction at the lowermost support (in the stinger tip region).



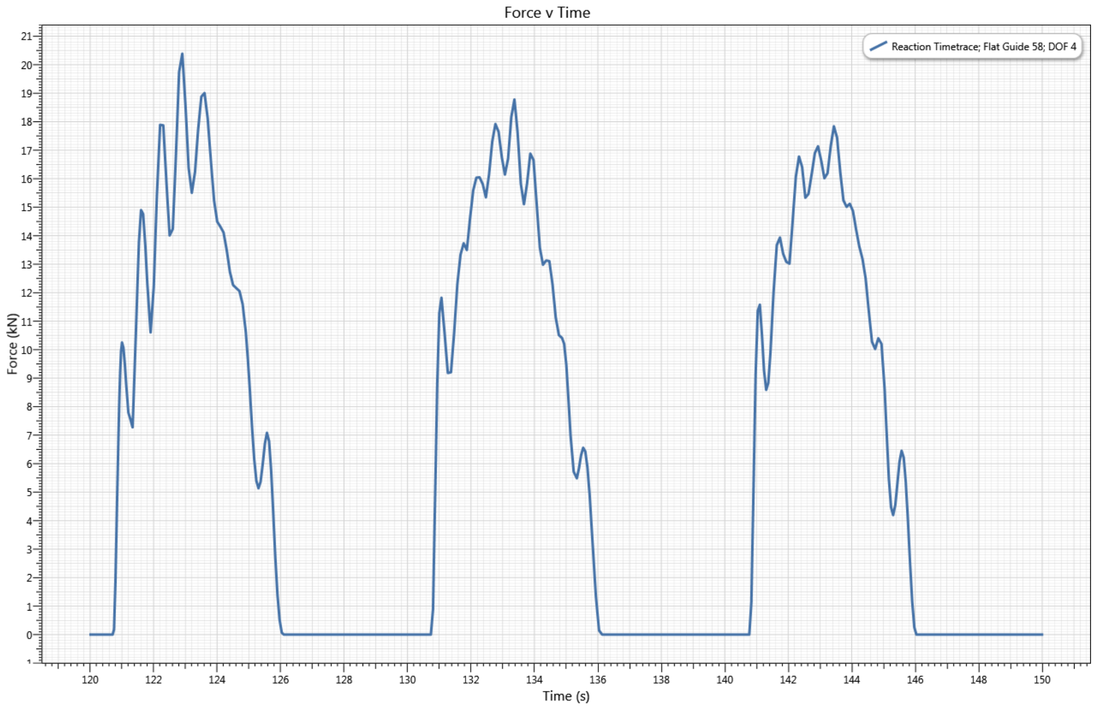
Effective Tension



Bending Stress



von Mises Stress



Sample Contact Reaction

1.10.8.4 H04 - Pipe Laying

This example considers the installation of a pipeline over an arbitrary seabed topography. It is divided into the following sections:

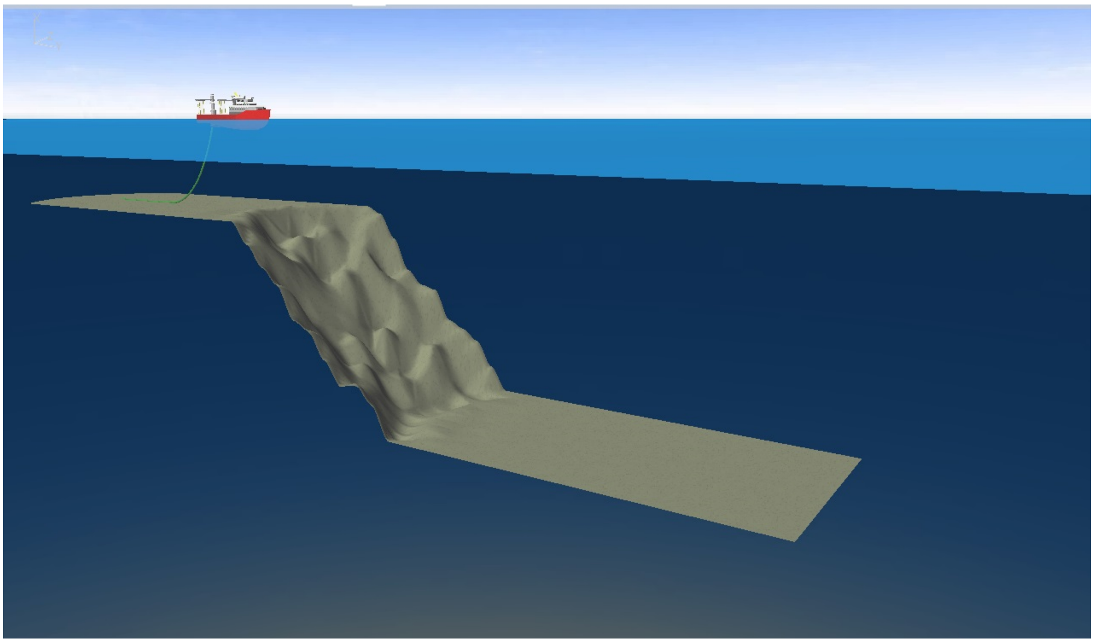
- Introduction gives an overview of the analysis.
- Model Summary describes the structure from a modelling viewpoint in Flexcom. Important modelling techniques for this type of analysis are noted, as well as relevant Flexcom features and capabilities.
- Analyses briefly describes the various analyses performed, discussing the environmental and loading conditions.
- Results presents pertinent results from the various analyses performed.

Introduction

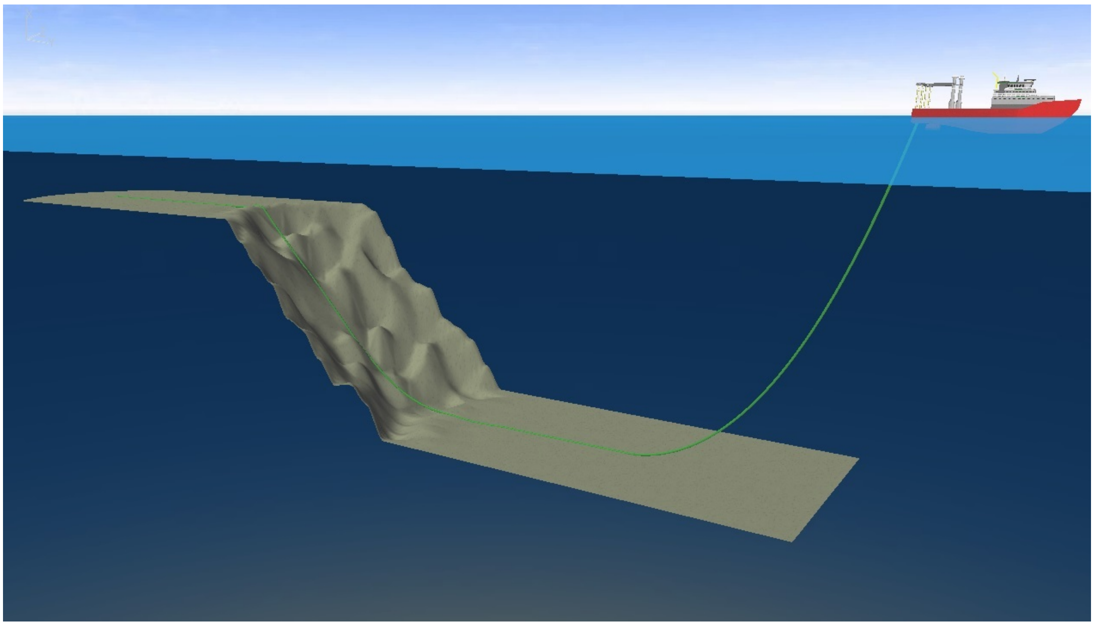
This example simulates an installation scenario in which a lay vessel gradually lowers a pipeline onto the seabed. The analysis is performed in four stages. The static configuration of the system is established in the initial static analysis. Two subsequent restart analyses are then performed in order to readjust some element lengths. The pipeline installation is then simulated in a subsequent dynamic analysis. The initial system configuration is shown in the first figure below, while the final configuration is presented in the second figure below.

This example illustrates a number of important features in Flexcom, specifically:

- Winch elements
- \- Arbitrary seabed modelling



Initial System Configuration



Final System Configuration

Model Summary

GENERAL

This example demonstrates the analysis of a pipe installation over a subsea precipice. The model consists of an installation vessel which lays the pipeline using a series of winch elements.

The pipeline is modelled using four consecutive line sections. The first section represents the initial pipeline configuration. The remaining three sections are comprised of very short elements, to allow for subsequent expansion to simulate the laying process.

The three-dimensional elastic seabed is generated using the Seabed Utility application. Refer to Elastic Seabed Profile for further information on the arbitrary seabed modelling feature in Flexcom.

Metric units are used throughout the model.

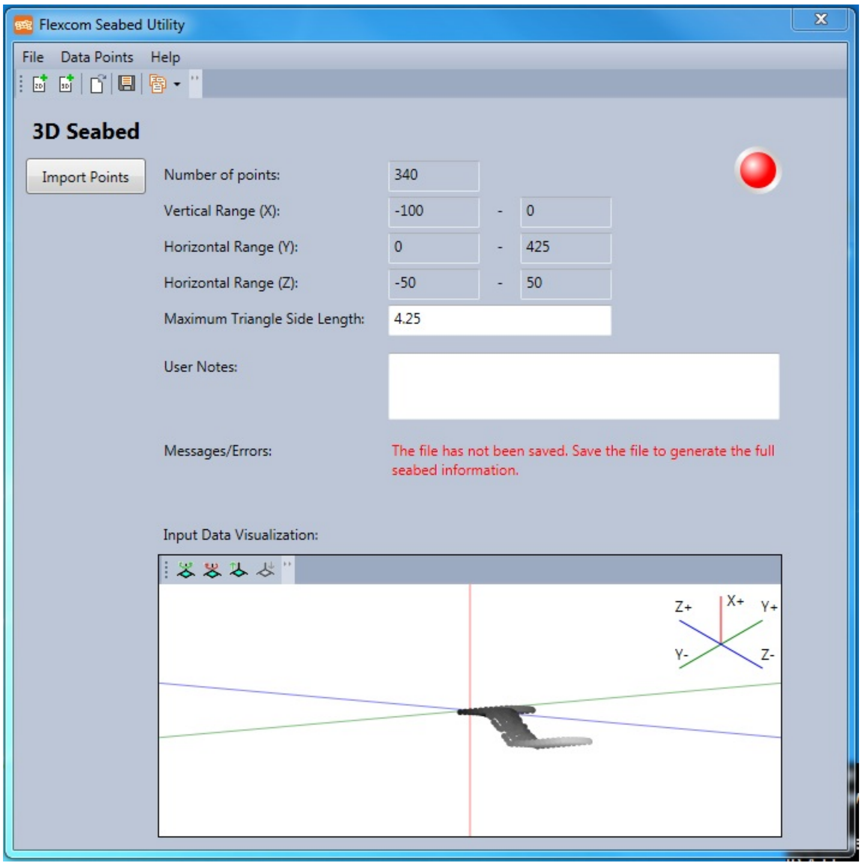
WINCH ELEMENTS

Winch elements are normal beam-column elements which have the unique property that their lengths can vary during an analysis. In a dynamic analysis, the variation in length is defined in terms of a maximum winch velocity and a winching time sequence. The time sequence consists of (i) a ramp-up time when winch velocity is increased from zero to the maximum value; (ii) a time during which the velocity remains at this maximum, and (iii) a ramp-down time when the velocity returns to zero. The operation in a static analysis is less complex, with an overall change in length being applied linearly from the analysis start time to the end time. Winch elements can be used, for example, in pipe-laying applications.

3D SEABED MODEL

Flexcom possesses a powerful 3D seabed modelling facility which can handle complex seabed topographies with ease. You simply specify an arbitrary cloud of data points, and Flexcom's triangulation algorithm automatically generates a continuous profile via linear or cubic spline interpolation.

Seabed profile data is specified in terms of arbitrary {x,y,z} data points in a standard text file. This generic input style offers complete flexibility, and means that Flexcom can readily accept seabed data from a range of third party software. Flexcom is accompanied by a helpful Seabed Utility application, shown in the figure below. This performs the required seabed meshing, and passes a compiled seabed data file to the main Flexcom program.



Seabed Utility Application

Analyses

MODEL SETUP

In the initial model setup, properties of the ocean, pipeline, seabed, installation vessel and internal fluid are assigned. The water depth is 50m at the initial vessel position, and increases to 150m as the vessel moves horizontally. The lower end of the pipeline is fixed at the seabed, and the upper end is connected to the vessel. The pipeline is filled with seawater. The file *3D_Seabed.fcsbd* specifies the seabed topography.

RE-MESHING ANALYSES

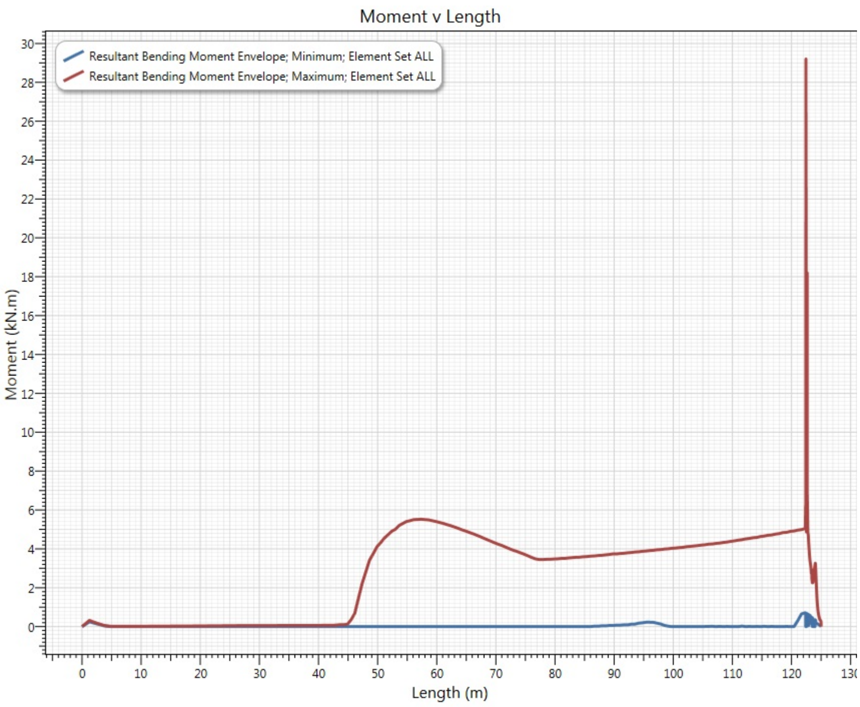
Due to the operation of the meshing algorithm, which attempts to maintain element length consistency between adjacent line sections, some elements in the upper region of the pipeline are quite short. In order to lengthen these elements, a static restart analysis is performed. Specifically, elements 87 to 98 were extended to all use a standard element length. A further static analysis is also performed to adjust the total length of the pipeline to compensate for the increase in pipe length from the previous analysis.

INSTALLATION ANALYSIS

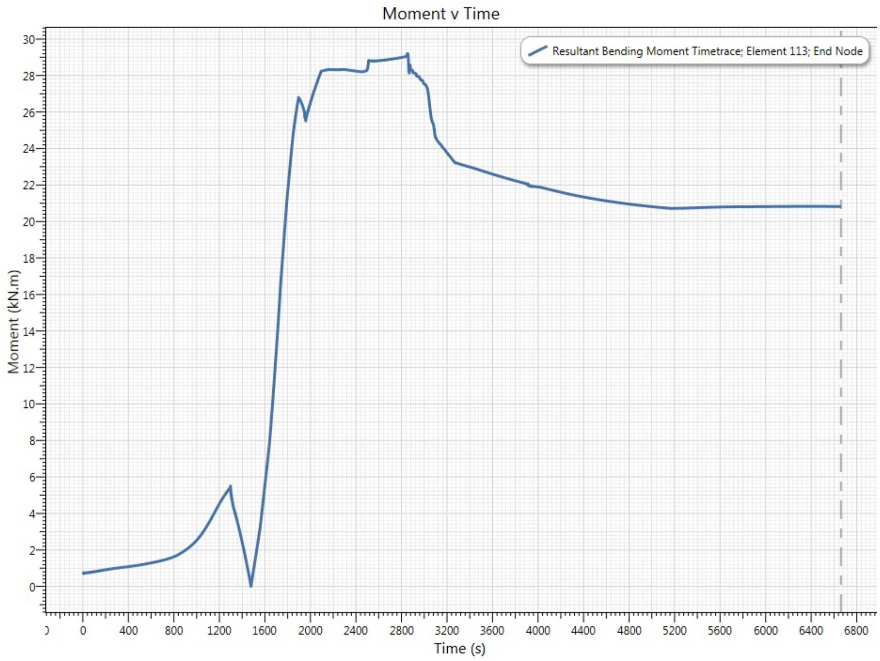
The model includes three winch element sets, named *WinchA*, *WinchB* and *WinchC*. Winch velocity as a function of time is specified for each set. Vessel motion is specified in terms of a series of horizontal displacements at various times in a file called *Vessel_motion.incx*.

Results

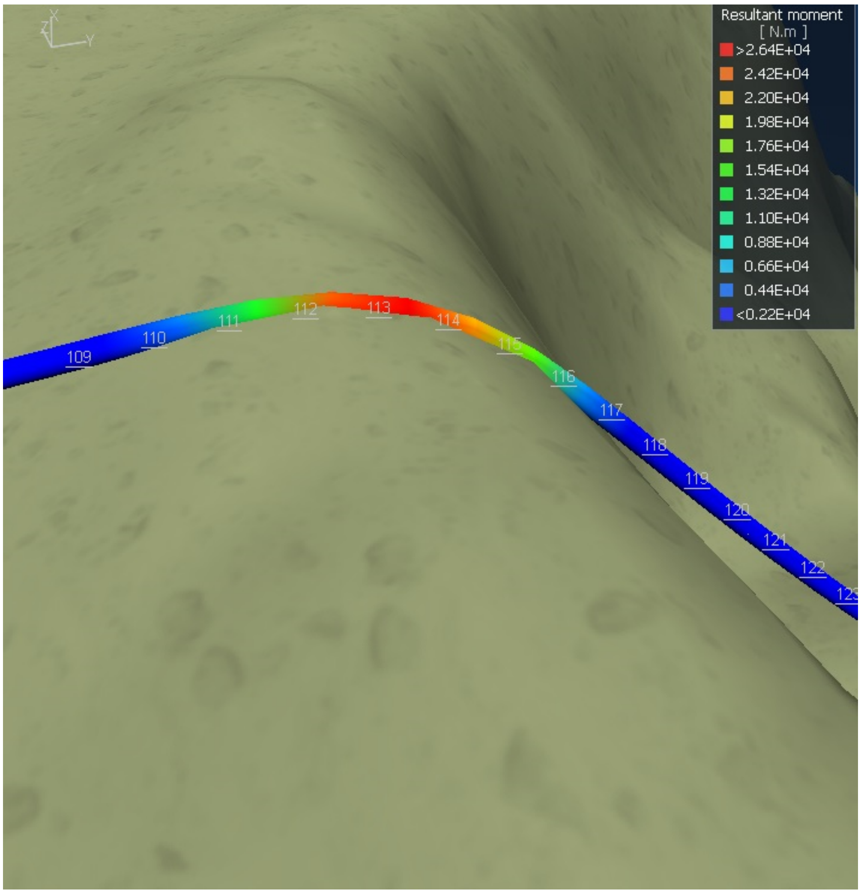
The first figure below presents the envelope plot of bending moment for the pipeline. A peak can be noticed on this plot, and this is further explored using a time history plot of bending moment, as shown in second figure below. Bending moment at this element reaches maximum value as pipeline bends over the precipice and the pipeline length increases, as illustrated in the third figure below.



Bending Moment Envelope

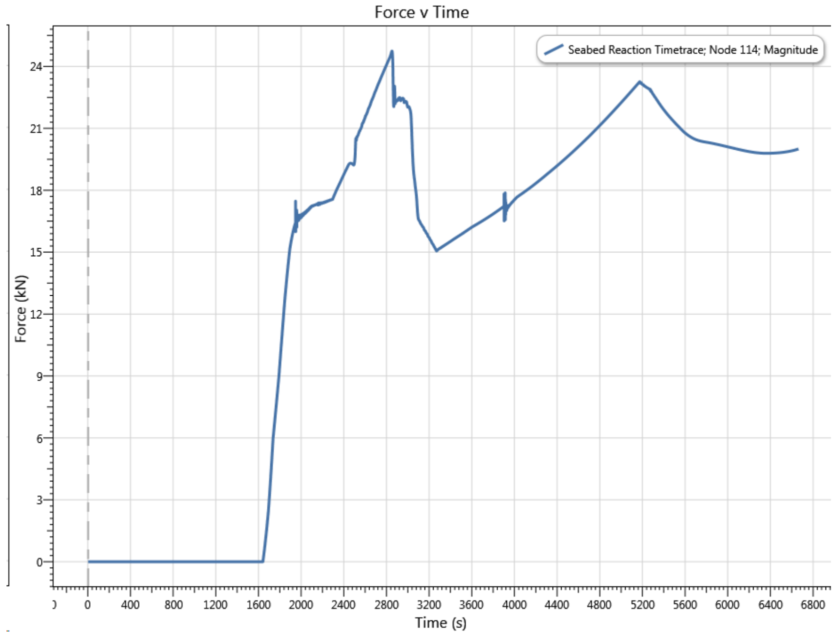


Time History of Bending Moment at Maximum Location



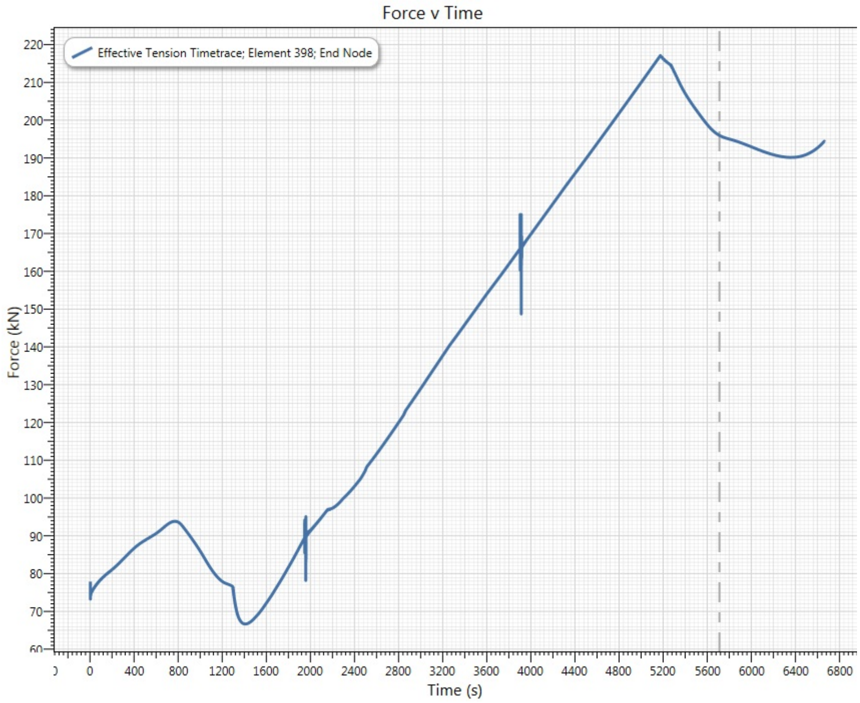
Maximum Bending Moment Location

The next figure shows a time history of seabed reaction force at the critical node, a measure of the maximum localised force being exerted on the pipe in this region. The time of peak reaction force corresponds with the peak in bending moment.



Time History of Seabed Reaction at Critical Location

The time history of effective tension at the vessel connection is shown in the figure below. Tension increases fairly linearly as the pipe is laid over the subsea precipice, due the increase in pipe suspended length. There are some local peaks in effective tension, correspond to the start and end times of the second winch. These could be smoothed had ramping been applied to the winch velocity.



Time History of Effective Tension at Vessel End

1.10.8.5 H05 - Steel Pipe Installation with Plastic Deformation

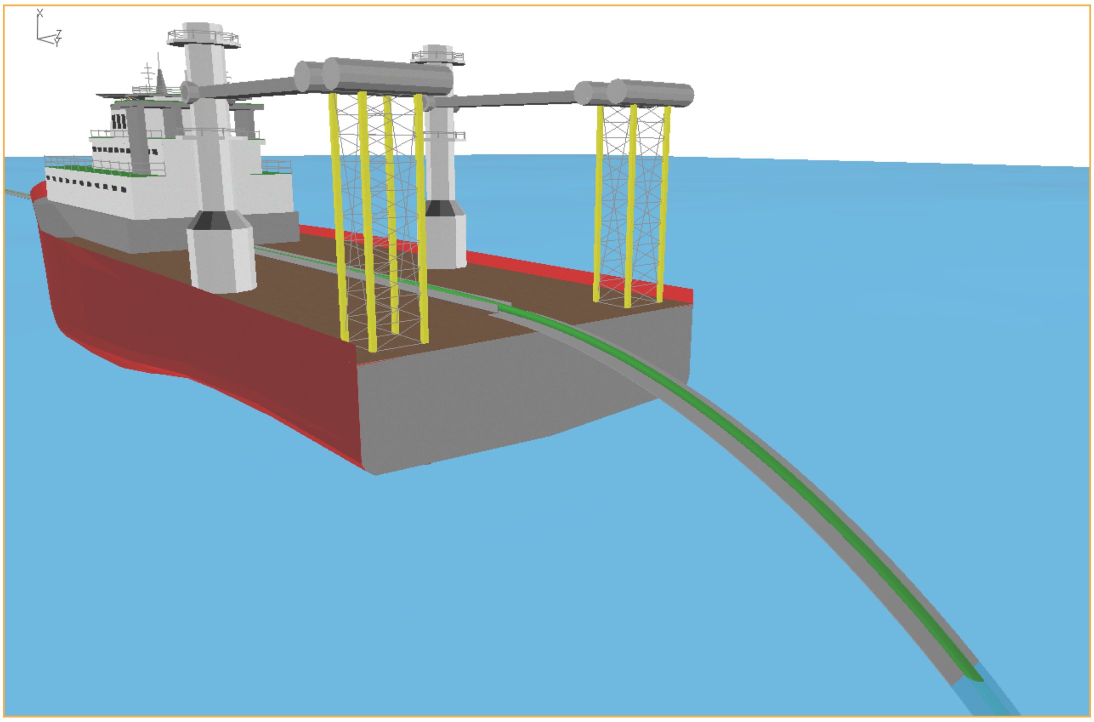
This example considers the installation of a steel pipe to examine the plastic deformation of the pipe as it passes over the vessel stinger and is landed onto the seabed. It is divided into the following sections:

- Introduction gives an overview of the analysis.
- Model Summary describes the structure from a modelling viewpoint in Flexcom. Important modelling techniques for this type of analysis are noted, as well as relevant Flexcom features and capabilities.
- Analyses briefly describes the various analyses performed, discussing the environmental and loading conditions.
- Results presents pertinent results from the various analyses performed.

Introduction

This example illustrates an installation scenario where a steel pipe is payed out over the vessel stinger and lowered onto the seabed to simulate a typical installation process. An initial static followed by a quasi-static analysis are used to establish the model configuration. This is then followed by a series of restart dynamic analyses to model the pipe payout over the stinger and lowering onto the seabed.

This example illustrates the plastic material hardening model in Flexcom.



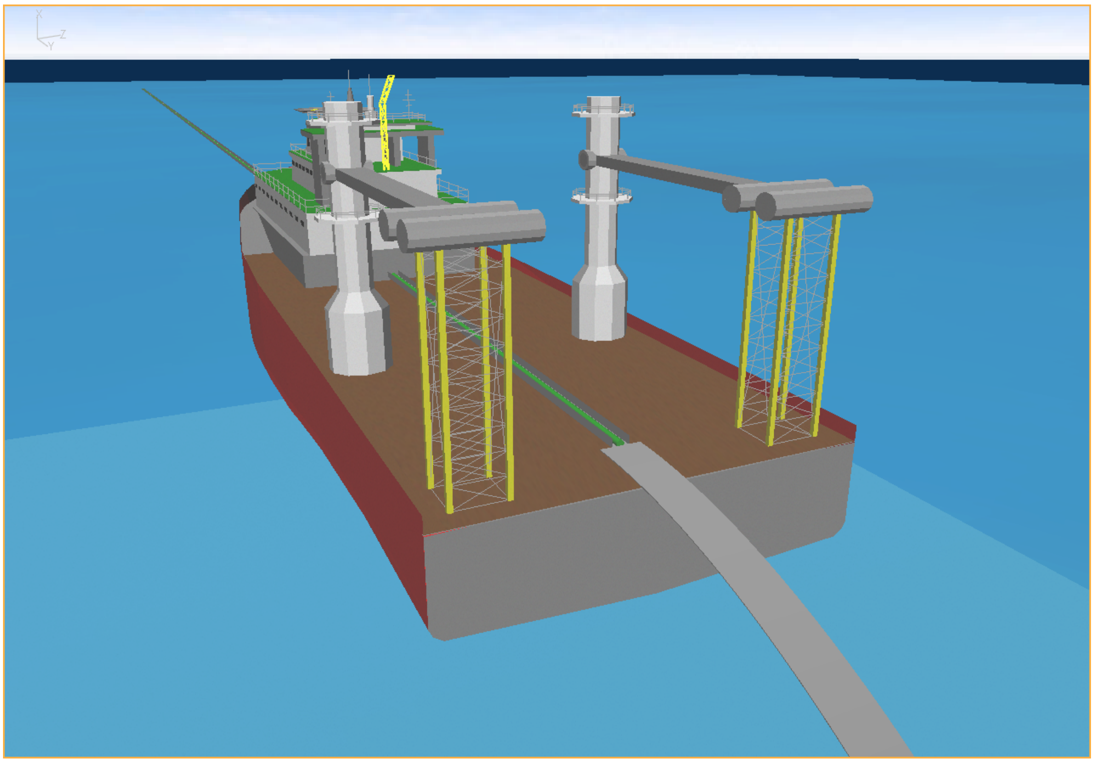
Model Configuration

Model Summary

The static configuration is established in the analysis file entitled *Model*. The pipe initial configuration is in its undeformed (straight) state and is supported by a flat guide surface which simulates the vessel deck level. The guide extends beyond the bow of the vessel in order to support the long length of pipe which will eventually be passed over the stinger. This approach facilitates modelling the pipe moving along the firing line in a straight undeformed state and then bending over the stinger which is the area of interest to the installation engineer.

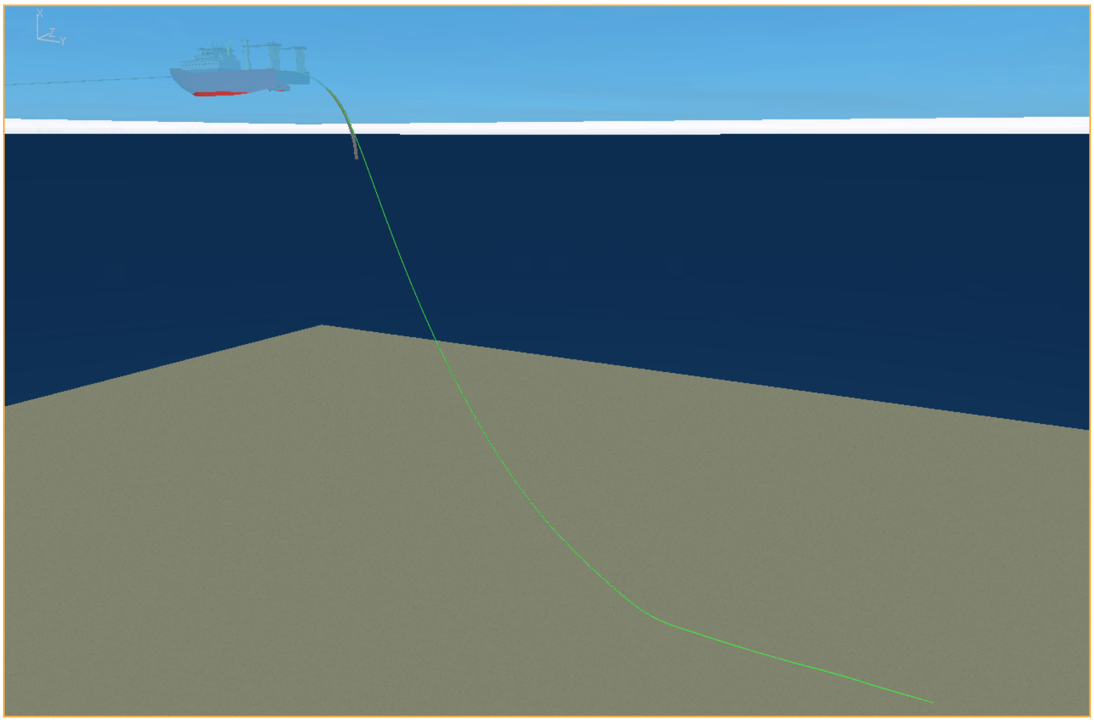
The pipe is initially held along its length with boundary conditions in the vertical degree of freedom, which are subsequently released in the restart analysis entitled *Contact* to ensure a robust numerical solution.

The application of the plastic hardening model is achieved via the *PLASTIC HARDENING keyword. The stinger is modelled using a curved guide surface via *GUIDE, TYPE=CYLINDRICAL where the stinger radius of curvature is easily specified.



Initial Model Configuration - Pipe in Undeformed State

Once the static configuration is established the pipe is then incrementally payed out over the stinger in a series of 7 restart analyses. The first payout analysis uses a displacement boundary condition in the surge direction to pay the pipe over the stinger and lower it to the seabed. In the subsequent payout analyses, the pipe end is connected to the seabed with a boundary condition and the pipe is further lowered onto the seabed using a combination of vessel offsets in the negative surge direction and displacement boundary conditions.



Model Configuration after all Payout Analyses

Analyses

INITIAL STATIC ANALYSIS

The pipe is modelled in the initial static analysis for its entire lay length as a straight pipe which extends beyond the vessel bow along a flat guide surface. A series of vertical degree of freedom boundary conditions are used to suspend the pipe in close proximity to the guide surface. The plastic material properties are defined via the *PLASTIC HARDENING keyword using an X65 (steel grade) non-linear stress-strain relationship.

QUASI-STATIC RELEASE ANALYSIS

A restart quasi-static analysis is used to release the vertical degree of freedom boundary conditions and allow the pipe to settle onto the flat guide surface. Solution robustness is aided with the addition of some mass damping on the pipe.

DYNAMIC ANALYSIS

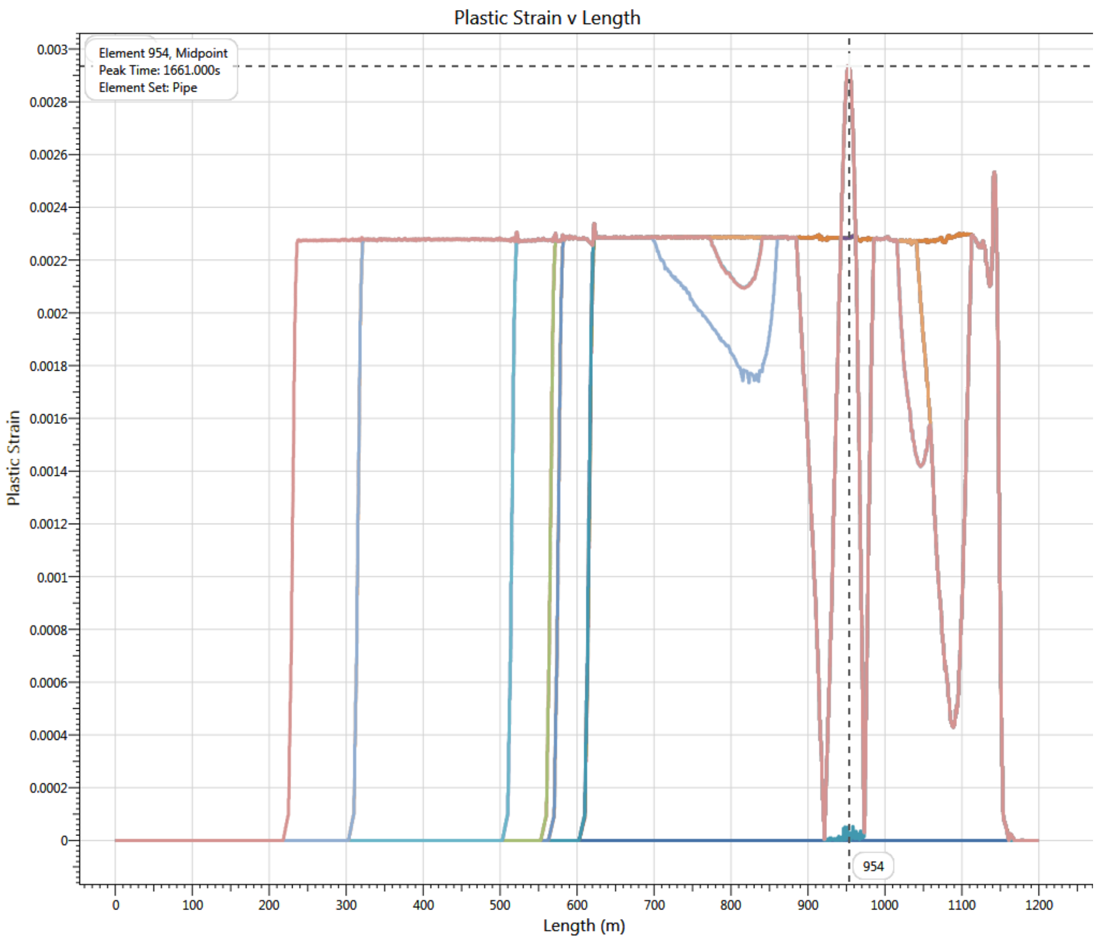
A series of dynamic restart analyses are used to simulate the payout of the pipe over the vessel stinger and lowering to the seabed. The initial payout analysis is restarted from the quasi-static release analysis and a displacement boundary condition is applied to the bow end of the pipe to push it over the stinger and lower to the seabed. Once the pipe reaches the seabed, the seabed end is attached to the seabed using a pinned boundary condition and the vessel is offset in the negative surge direction while also applying a displacement boundary condition at the bow end of the pipe. This procedure is carried out in incremental restart analysis steps from the second payout analysis to the last until the pipe is installed in an S-lay configuration.

The full series of installation stages is as follows:

Analysis Name	Payout(m)	Vessel Offset(m)
Payout 1	600	
Payout 2	100	100
Payout 3	60	100
Payout 4	40	50
Payout 5	50	100
Payout 6		200
Payout 7	15	100

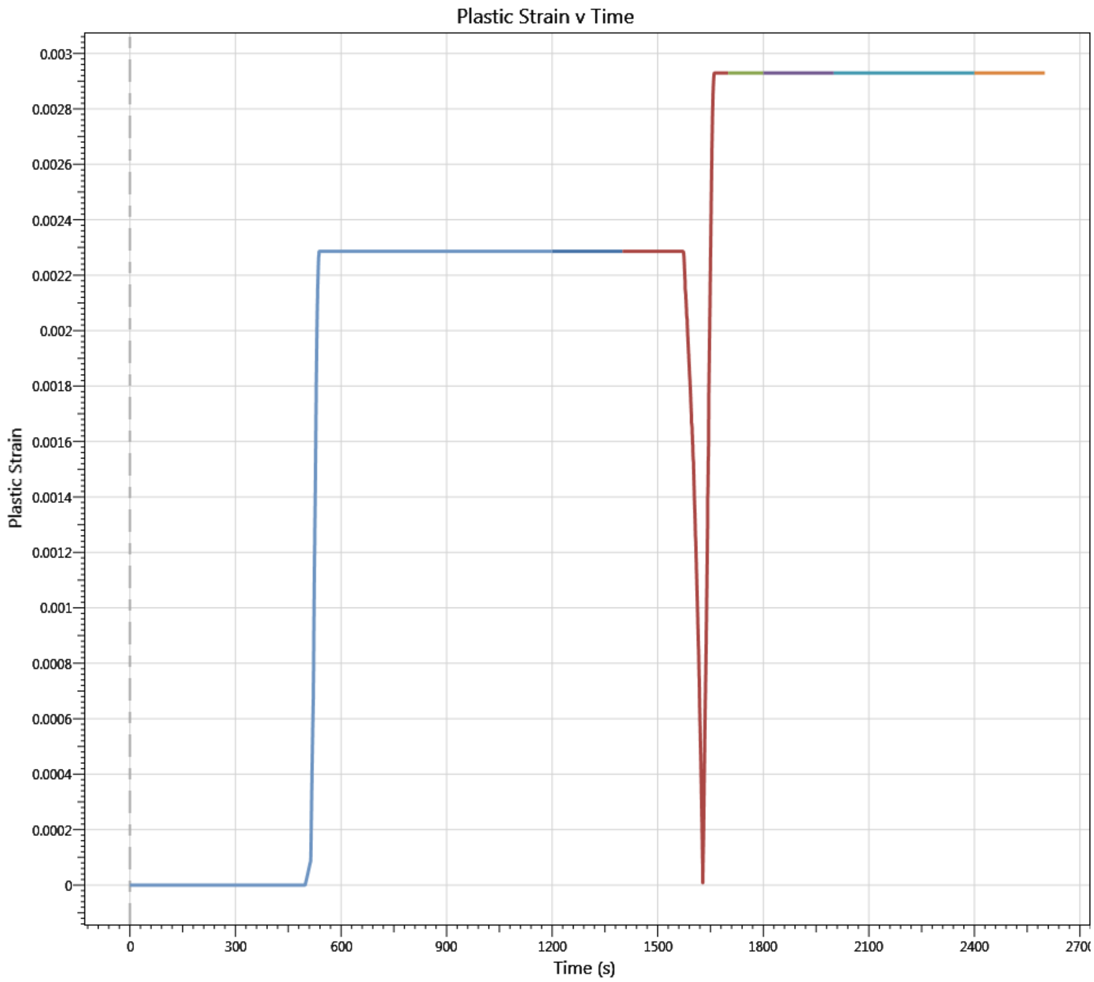
Results

The figure below shows an envelope plot of maximum plastic bending strain for all payout analyses superimposed over each other. Using the Tooltips on the plot viewer over the region of largest strain we are able to identify the location on the pipe which experiences the largest plastic deformation throughout the installation process.



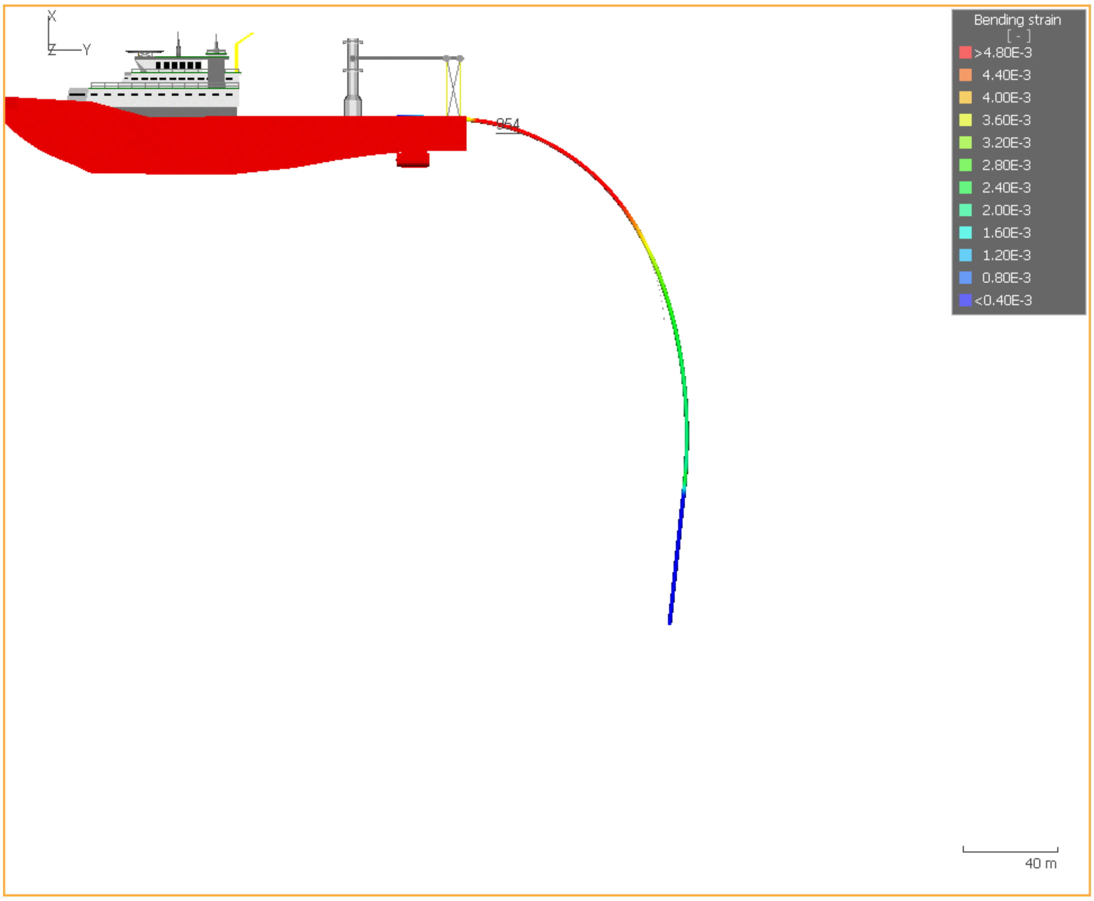
Plastic Strain vs Pipe Length for all Payout Analyses

The first significant plastic strain is experienced around time 537s, as shown in the time history of plastic bending strain at the critical location we identified along the pipe.



Timetrace of Plastic Strain at Element 954

Examining the model view at this time for that location on the pipe we can see that this occurs when the pipe is deforming over the stinger under its own self weight during the payout.



Model View - Critical Plastic Strain Location - Payout at 537s

At a time of 1628s if we examine the time trace of plastic strain we see that total bending strain has returned to zero, and examining the model view at this location and time we see that the pipe is straightening out as it is being lowered onto the seabed.

Model View - Critical Plastic Strain Location - Payout at 1628s

By the end of the simulation, the direction of curvature has now fully reversed from hogging to sagging and we are left with residual strain at this location as shown in the timetrace of plastic strain.

Model View - Critical Plastic Strain Location - Payout Complete

1.10.9 I - Offshore Structures

Section I contains an example of an offshore structure:

- I01 - Jack-Up Platform

1.10.9.1 I01 - Jack-Up Platform

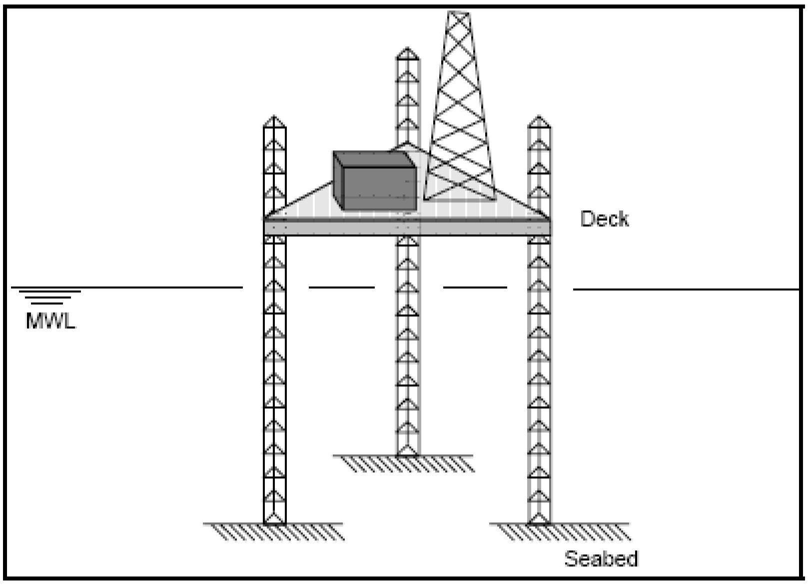
This example describes the analysis of a jack-up platform, and demonstrates the use of the program in modelling 3D frame-type rigid structures. The overall layout of this example is as follows:

- Introduction gives an overview of the jack-up platform model.

* Model Summary describes the model in more detail, and discusses the relevant analytical capabilities of the software.
* Analyses briefly describes the various analyses performed, discussing the various environmental and loading conditions to which the model is subjected.
* Results presents pertinent results from the various analyses performed.

Introduction

This example considers the analysis of a jack-up platform. The example consists of a static analysis of the structure subject to wind and current loads, and time domain and frequency dynamic analyses of the structure subject to wave loading. A schematic of the jack-up platform is shown in figure below.This example considers the analysis of a jack-up platform. The example consists of a static analysis of the structure subject to wind and current loads, and time domain and frequency dynamic analyses of the structure subject to wave loading. A schematic of the jack-up platform is shown in the figure below.



Schematic of Jack-Up Platform

Model Summary

This example demonstrates the analysis of a jack-up platform. The jack-up is sited in a water depth of 110m. The overall height is 153m with the deck at a height of 135m above the seabed. The structure is subjected to static and dynamic loading, which includes wind, current and regular wave loading.

The deck and legs of the jack-up platform are modelled using rigid beam elements. The deck of the jack-up platform is connected to the three support legs via six articulations elements. Each leg is connected to the deck at two nodes. This model of the jack-up allows the non-linear p-δ effect to be accounted for.

Analyses

INITIAL STATIC ANALYSIS

The jack-up is subjected to gravity and buoyancy loading. The seabed nodes of the three legs are fixed in all the translational degrees of freedom (DOFs) and rotational DOF 4 (to ensure the model is statically determinate).

CURRENT ANALYSIS

In addition to the initial static loads, wind and current loads are applied to the structure. The current profile has a piecewise-linear distribution. The wind forces are applied as point loads to the upper portions of each of the three legs. The BCs remain unchanged and are carried through automatically from the initial static analysis.

TIME DOMAIN DYNAMIC ANALYSIS

In addition to the above, regular wave loading is applied to the jack-up. The BCs remain unchanged and are carried through automatically from the current analysis

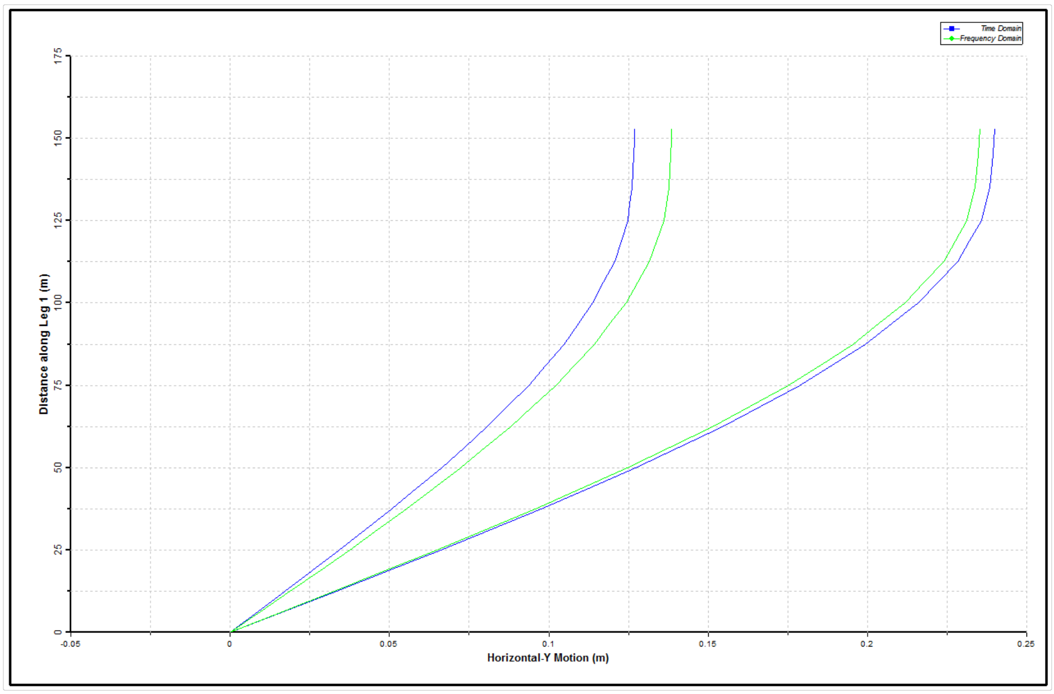
FREQUENCY DOMAIN DYNAMIC ANALYSIS

Similar to the time domain, no specification of boundary conditions is required in the frequency domain analysis. All of the BC data carries through from the current analysis.

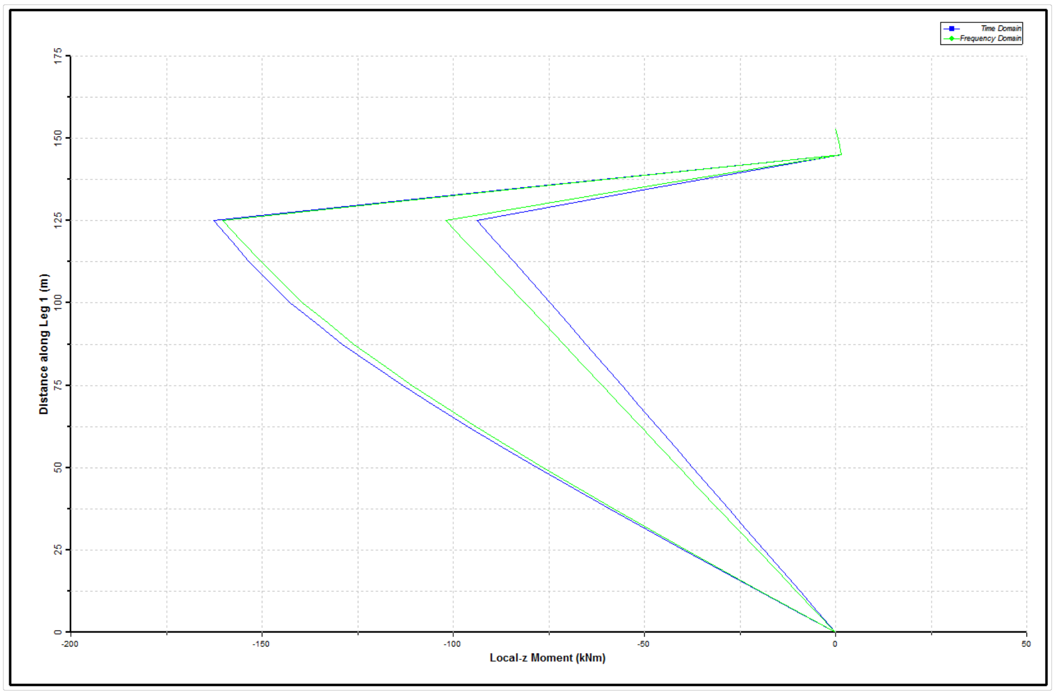
Results

Results from the dynamic analyses of the jack-up are presented in the figures below. The first figure below gives a comparison of the horizontal motion of Leg 1 obtained using time and frequency domain analyses. The second figure presents a comparison of the max/min envelopes of the local-z bending moment distribution in Leg 1, for both time domain and frequency domain analyses.

The results demonstrate close agreement between the time and frequency domain approaches. Discrepancies in the solutions may be attributed to the non-linear p-δ effect, which cannot be captured in the frequency domain. Specifically, the p-δ effect means that lateral displacements of the jack-up platform result in second order overturning moments being generated. The effect is emphasised by the fact that a large proportion of the structure mass is concentrated on the platform deck (the deck elements are almost 4 times as heavy as the leg elements).



Envelopes of Horizontal-Y Motion in Leg 1



Envelopes of Local-Z Moment in Leg 1

1.10.10 J - Specialised Examples

Section J illustrates some specialised modelling scenarios, including:

- J01 - Dropped Object and Recovery
- J02 - Advanced Database Postprocessing
- J03 - Summary Postprocessing Collation
- J04 - User Solver Variables

1.10.10.1 J01 - Dropped Object and Recovery

This example illustrates a dropped object scenario and its recovery modelled using Flexcom. It is divided into the following sections:

- Introduction gives an overview of the analysis.
- Model Summary describes the model set-up in more detail.

- Analyses briefly describes the dropped object analysis, including selection of appropriate solution parameters.
- Results presents pertinent results from the simulation.

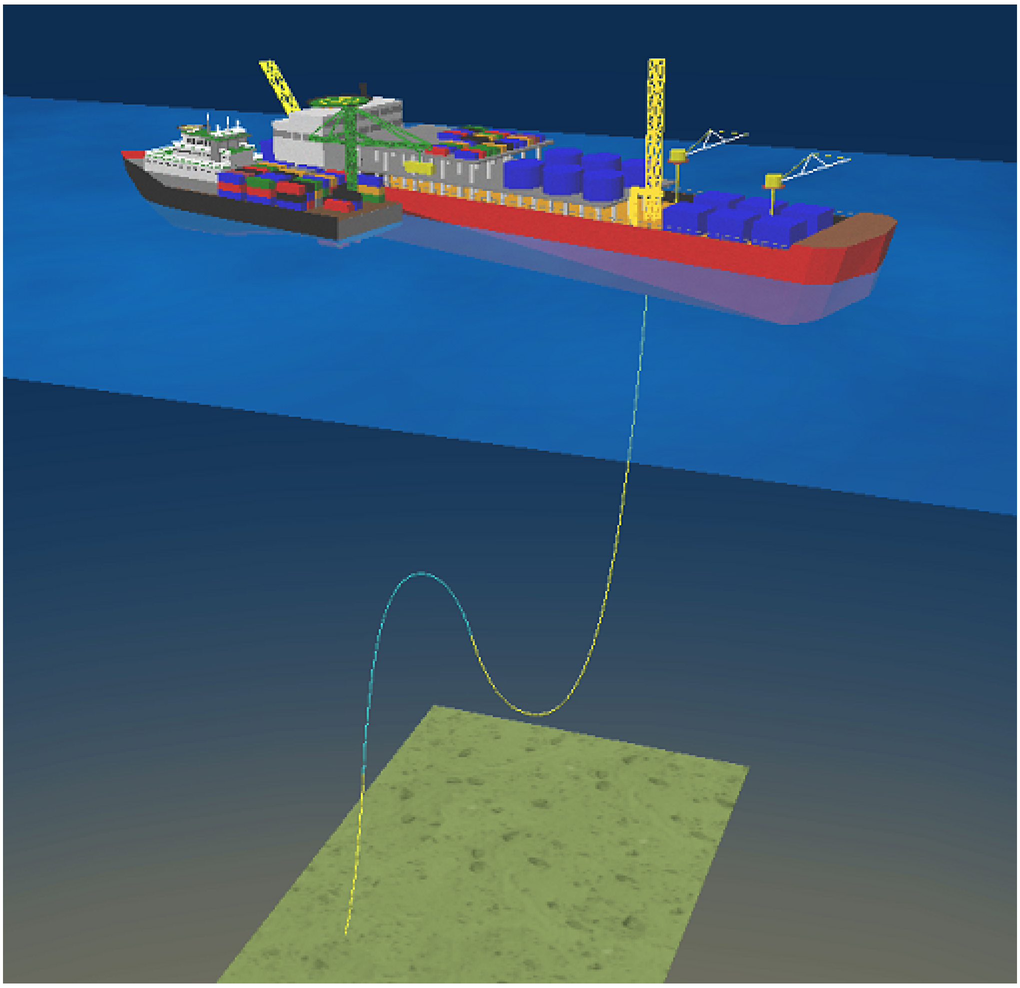
Introduction

This example illustrates a dropped object scenario and its recovery modelled using Flexcom. There are numerous hazards associated with the transfer of cargo at offshore locations, posing health and safety risks to both offshore personnel and to equipment.

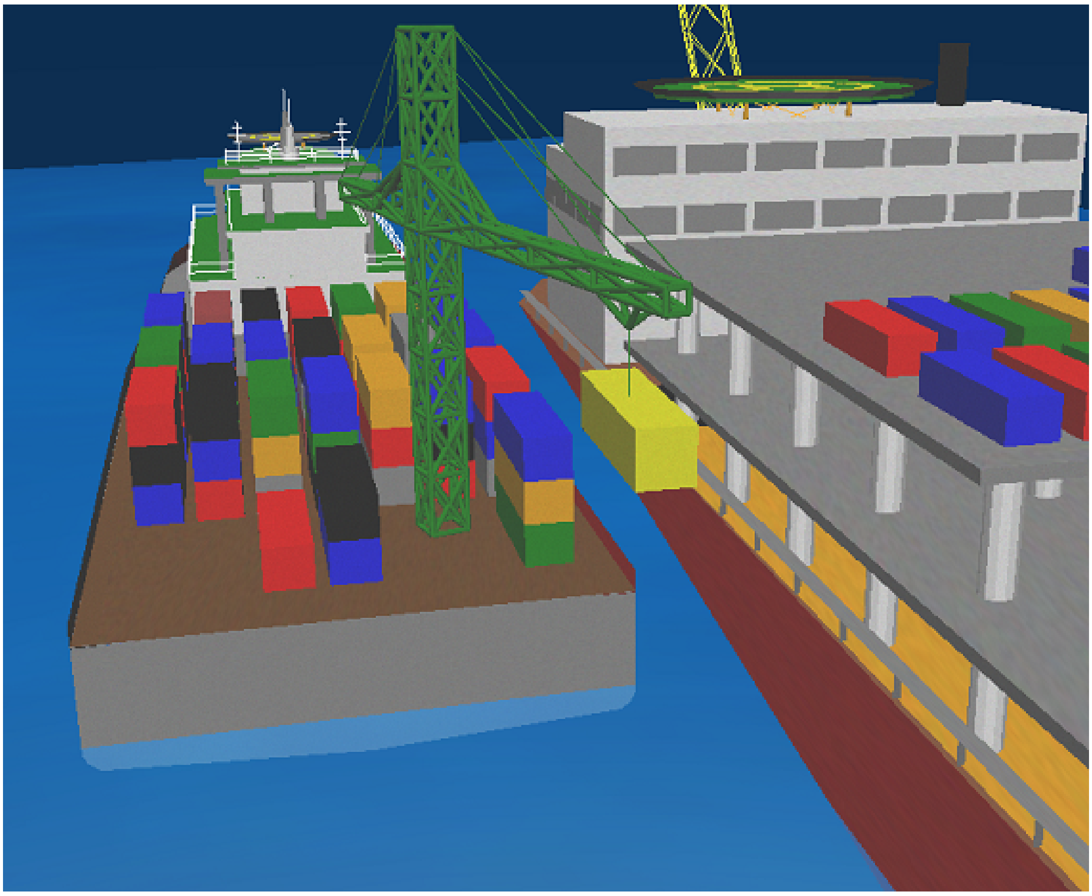
The model contains a flexible riser arranged in a steep wave configuration, which is connected to an FPSO. A support vessel is docked alongside the FPSO, and is transferring freight containers across to it. At some point during the transfer process, the weight bearing cable snaps, and the container drops from the crane, and sinks down into the ocean.

The second component of this analysis looks at the recovery of this dropped container by performing a dynamic lift using a salvage air bag which is attached to the container and slowly inflated to raise the container from seabed to surface.

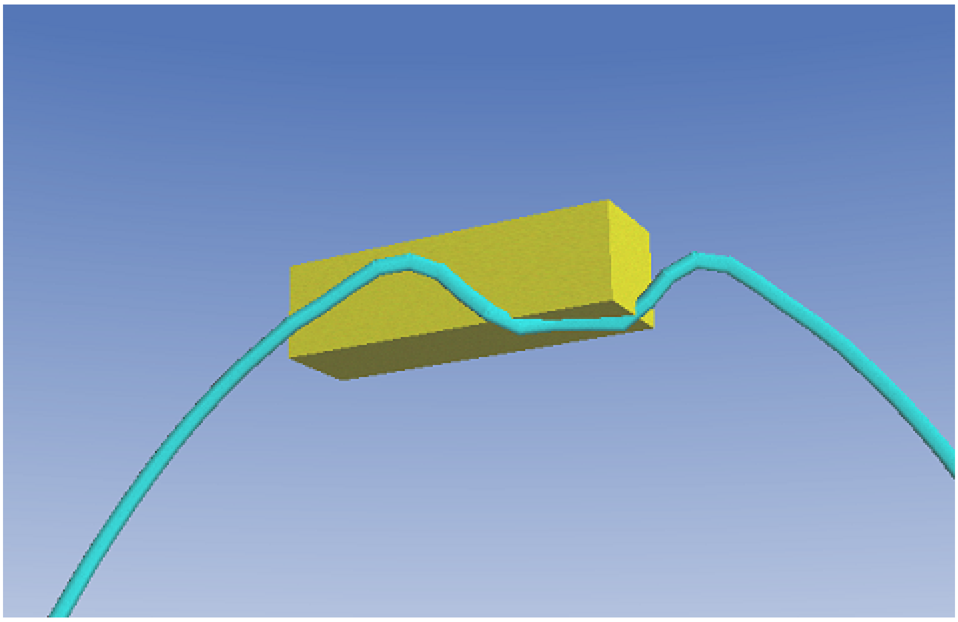
Some illustrations from the simulation are presented in the figures below.



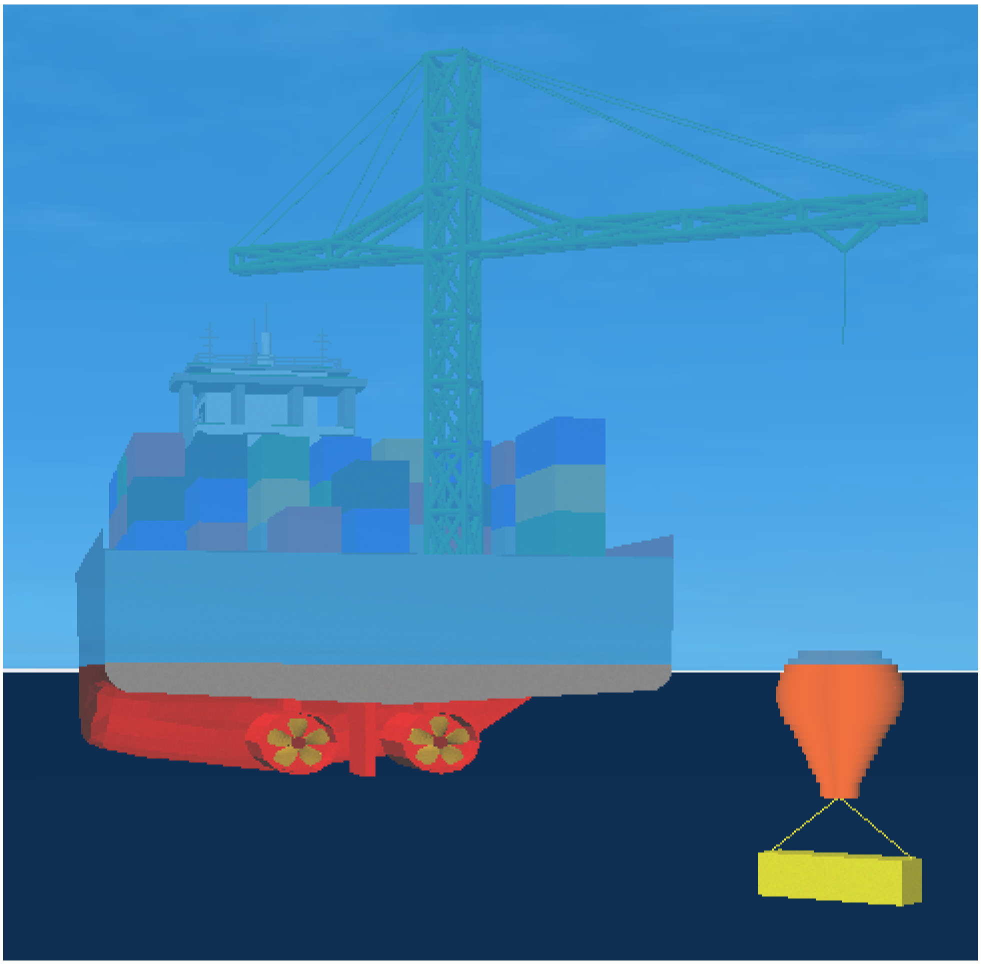
System Overview



Cargo Transfer



Container colliding with Riser



Container Recovery Bouyancy Lift Bag

Model Summary

The riser in this example is 430m in length and is sited in 300m of water. It is filled with oil and attached to a production vessel. In order to accurately capture the deformation of the flexible riser, relatively short elements are used in the contact region.

The container is modelled as a rigid body using elements of high stiffness. Hydrodynamic properties are assigned via the buoy structure facility.

Contact between the falling container and the flexible riser is modelled using Flexcom's line clashing feature. Potential contact areas are defined in the model, along with a relatively high contact stiffness. For added realism, auxiliary elements are used to visually replicate the crane structure.

The container recovery model concentrates on the lift aspect only and the riser is not included. It uses User Defined Elements where the buoyancy diameter of the salvage balloon elements are changed during the dynamic analysis via a user dll to replicate the inflation of the balloon. This allows the container to be lifted off the seabed and floated to the surface.

Analyses

The dropped object analysis is performed in two stages. The initial static determines the static equilibrium configuration of the system. The dynamic analysis models the collision between the falling container and the riser.

The base of the riser is fixed in all translational degrees of freedom, while the upper end is attached to the vessel. The container is restrained from motion in the initial static analysis, and then allowed to free-fall in the subsequent dynamic analysis.

Clashing is a complex and highly non-linear phenomenon, so the use of relatively small time steps is essential, in order to accurately capture the moment of impact, and the subsequent structural response. Flexcom automatically reduces the solution time step in anticipation of contact, and once the components have separated after impact, the time step begins to increase again. This approach facilitates a robust and accurate contact model, while also ensuring an efficient simulation.

The recovered object analysis is performed in two stages. The initial static is used to set the model configuration and define the balloon elements attached to the container for later modification in a subsequent dynamic analysis.

The dynamic analysis is used to load the dll which changes the element buoyancy diameters on the balloon as required throughout the analysis to replicate the inflating lift balloon. The source code for the user defined element dll is shown below.

```
subroutine
user_defined_element(iter,time,ramp,nnode,nncon,ncord,nndof,nelmn,necon,nedof,ndamp,elm
con,    &
  nodcon,intoun,ietoue,etype,edamp,cord,displacement,velocity,acceleration,trgb,tglu,di
sp_prev,pos_prev, &
  vel_prev,acc_prev,axial_force,y_shear,z_shear,torque,y_bending,z_bending,eff_tension,
y_curvature,      &
  z_curvature,length,eiyy,eizz,gj,ea,mass,polar,dint,ddrag,dbuoy,dout,dcont,fluid,dampe
r)
!dec$ attributes dllexport, stdcall, reference :: user_defined_element
  implicit none

! Input variables - cannot be modified within this subroutine

  integer, intent(in)     :: iter
  !dec$ attributes value :: iter
  real(8), intent(in)     :: time
  !dec$ attributes value :: time
  real(8), intent(in)     :: ramp
  !dec$ attributes value :: ramp
  integer(4), intent(in) :: nnode
  !dec$ attributes value :: nnode
  integer(4), intent(in) :: nncon
  !dec$ attributes value :: nncon
  integer(4), intent(in) :: ncord
  !dec$ attributes value :: ncord
  integer(4), intent(in) :: nndof
  !dec$ attributes value :: nndof

  integer(4), intent(in) :: nelmn
  !dec$ attributes value :: nelmn
  integer(4), intent(in) :: necon
  !dec$ attributes value :: necon
  integer(4), intent(in) :: nedof
  !dec$ attributes value :: nedof
  integer(4), intent(in) :: ndamp
  !dec$ attributes value :: ndamp

  integer(4), intent(in), dimension(nncon, nelmn) :: elmcon
  !dec$ attributes reference :: elmcon
  integer(4), intent(in), dimension(necon,nnode)  :: nodcon
  !dec$ attributes reference :: nodcon
  integer(4), intent(in), dimension(nnode)        :: intoun
  !dec$ attributes reference :: intoun
  integer(4), intent(in), dimension(nelmn)        :: ietoue
  !dec$ attributes reference :: ietoue
  integer(4),  intent(in), dimension(nelmn)       :: etype
  !dec$ attributes reference :: etype
  integer(4),  intent(in), dimension(nelmn)       :: edamp
  !dec$ attributes reference :: edamp
  real(8),   intent(in), dimension(ncord,nnode)   :: cord
  !dec$ attributes reference :: cord
  real(8),   intent(in), dimension(nndof,nnode)   :: displacement
  !dec$ attributes reference :: displacement
  real(8),   intent(in), dimension(nndof,nnode)   :: velocity
  !dec$ attributes reference :: velocity
  real(8),   intent(in), dimension(nndof,nnode)   :: acceleration
  !dec$ attributes reference :: acceleration
  real(8),   intent(in), dimension(3,3,nelmn)     :: trgb
```

```
  !dec$ attributes reference :: trgb
  real(8),  intent(in), dimension(3,3,nelmn)      :: tglu
  !dec$ attributes reference :: tglu
  real(8),  intent(in), dimension(nndof,nnode)    :: disp_prev
  !dec$ attributes reference :: disp_prev
  real(8),  intent(in), dimension(nndof,nnode)    :: pos_prev
  !dec$ attributes reference :: pos_prev
  real(8),  intent(in), dimension(nndof,nnode)    :: vel_prev
  !dec$ attributes reference :: vel_prev
  real(8),  intent(in), dimension(nndof,nnode)    :: acc_prev
  !dec$ attributes reference :: acc_prev
  real(8),  intent(in), dimension(nelmn,3)        :: axial_force
  !dec$ attributes reference :: axial_force
  real(8),  intent(in), dimension(nelmn,3)        :: y_shear
  !dec$ attributes reference :: y_shear
  real(8),  intent(in), dimension(nelmn,3)        :: z_shear
  !dec$ attributes reference :: z_shear
  real(8),  intent(in), dimension(nelmn,3)        :: torque
  !dec$ attributes reference :: torque
  real(8),  intent(in), dimension(nelmn,3)        :: y_bending
  !dec$ attributes reference :: y_bending
  real(8),  intent(in), dimension(nelmn,3)        :: z_bending
  !dec$ attributes reference :: z_bending
  real(8),  intent(in), dimension(nelmn,3)        :: eff_tension
  !dec$ attributes reference :: eff_tension
  real(8),  intent(in), dimension(nelmn,3)        :: y_curvature
  !dec$ attributes reference :: y_curvature
  real(8),  intent(in), dimension(nelmn,3)        :: z_curvature
  !dec$ attributes reference :: z_curvature

! Output variables - can be modified within this subroutine if required

  real(8),  intent(inout), dimension(nelmn)       :: length
  !dec$ attributes reference :: length
  real(8),  intent(inout), dimension(nelmn)       :: eiyy
  !dec$ attributes reference :: eiyy
  real(8),  intent(inout), dimension(nelmn)       :: eizz
  !dec$ attributes reference :: eizz
  real(8),  intent(inout), dimension(nelmn)       :: gj
  !dec$ attributes reference :: gj
  real(8),  intent(inout), dimension(nelmn)       :: ea
  !dec$ attributes reference :: ea
  real(8),  intent(inout), dimension(nelmn)       :: mass
  !dec$ attributes reference :: mass
  real(8),  intent(inout), dimension(nelmn)       :: polar
  !dec$ attributes reference :: polar
  real(8),  intent(inout), dimension(nelmn)       :: dint
  !dec$ attributes reference :: dint
  real(8),  intent(inout), dimension(nelmn)       :: ddrag
  !dec$ attributes reference :: ddrag
  real(8),  intent(inout), dimension(nelmn)       :: dbuoy
  !dec$ attributes reference :: dbuoy
  real(8),  intent(inout), dimension(nelmn)       :: dout
  !dec$ attributes reference :: dout
  real(8),  intent(inout), dimension(nelmn)       :: dcont
  !dec$ attributes reference :: dcont
  real(8),  intent(inout), dimension(nelmn,4)     :: fluid
  !dec$ attributes reference :: fluid
  real(8),  intent(inout), dimension(ndamp,4)     :: damper
```

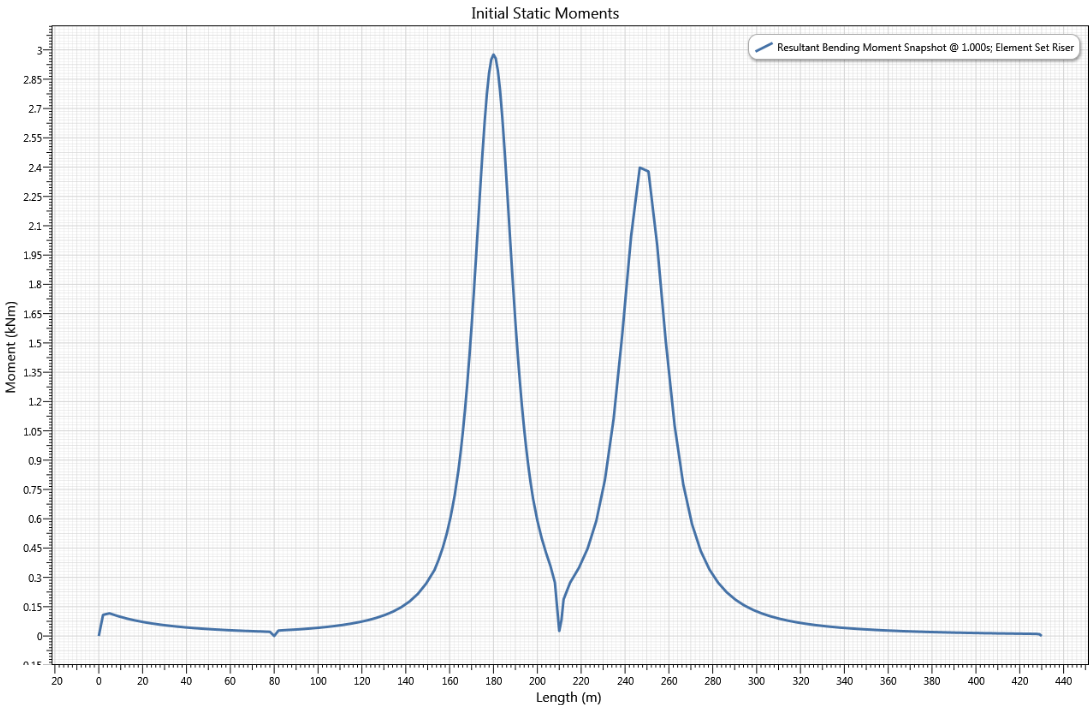
```
  !dec$ attributes reference :: damper

! Variable names
!     iter              : Current iteration
!     time              : Current timestep
!     ramp              : Ramp
!     nnode             : Number of nodes in the model
!     nncon             : Number of nodes with connected elements
!     ncord             : Number of coordinates
!     nndof             : Number of degrees of freedom per node (6)
!     nelmn             : Number of elements in the model
!     necon             : Number of elements connected
!     nedof             : Number of degrees of freedom per element (14)
!     ndamp             : Number of damper elements
!     elmcon            : Element connectivity array
!     nodcon            : Node connectivity array
!     intoun            : Internal node to user node numbering array
!     ietoue            : Internal element to user element numbering array
!     etype             : Element type - (1) Beam, (2) Spring, (3) Hinge, (4) Damper
!     edamp             : Damper element number array
!     cord              : Initial nodal co-ordinates
!     displacement      : Nodal displacements at previous iteration
!     velocity          : Nodal velocities at previous iteration
!     acceleration      : Nodal accelerations at previous iteration
!     trgb              : Rigid body rotation (local undeformed -> convected)
transformation matrix
!     tglu              : Global to local undeformed transformation matrix
!     disp_prev         : Nodal displacements at previous timestep
!     pos_prev          : Nodal positions at previous timestep
!     vel_prev          : Nodal velocities at previous timestep
!     acc_prev          : Nodal accelerations at previous timestep
!     axial_force       : Axial force in elements at previous timestep
!     y_shear           : Y Shear forces in elements at previous timestep
!     z_shear           : Z Shear forces in elements at previous timestep
!     torque            : Torque in elements at previous timestep
!     y_bending         : Y bending moments in elements at previous timestep
!     z_bending         : Z bending moments in elements at previous timestep
!     eff_tension       : Effective Tension in elements at previous timestep
!     y_curvature       : Y curvatures in elements at previous timestep
!     z_curvature       : Z curvatures in elements at previous timestep
!     length            : Element natural length
!     eiyy              : Element linear EIyy
!     eizz              : Element linear EIzz
!     gj                : Element linear GJ
!     ea                : Element linear EA
!     mass              : Element mass per unit length
!     polar             : Element polar inertia per unit length
!     dint              : Element internal diameters
!     ddrag             : Element drag diameters
!     dbuoy             : Element buoyancy diameters
!     dout              : Element outer diameters
!     dcont             : Element contact diameters
!     fluid             : Element fluid contents -
!                           (1) Top elevation (fluid head)
!                           (2) Density
!                           (3) Internal pressure
!                           (4) Velocity
!     damper            : Element contact diameters
!                           (1) C0
```

```
!                             (2) C1
!                             (3) C2
!                             (4) C0_Threshold


! Declare local variables...
  integer :: i

! Insert user code below this line...
  do i = 1, nelmn
    if( ietoue(i) == 853 )then
      dbuoy(i) = 2.0d0 + 0.02d0 * (time / 100.0d0)
    else if( ietoue(i) == 854 )then
      dbuoy(i) = 2.0d0 + 0.02d0 * (time / 100.0d0)
    else if( ietoue(i) == 855 )then
      dbuoy(i) = 2.0d0 + 0.02d0 * (time / 100.0d0)
    else if( ietoue(i) == 856 )then
      dbuoy(i) = 2.0d0 + 0.02d0 * (time / 100.0d0)
    end if
  end do

end subroutine user_defined_element
```

Results

INITIAL STATIC ANALYSIS

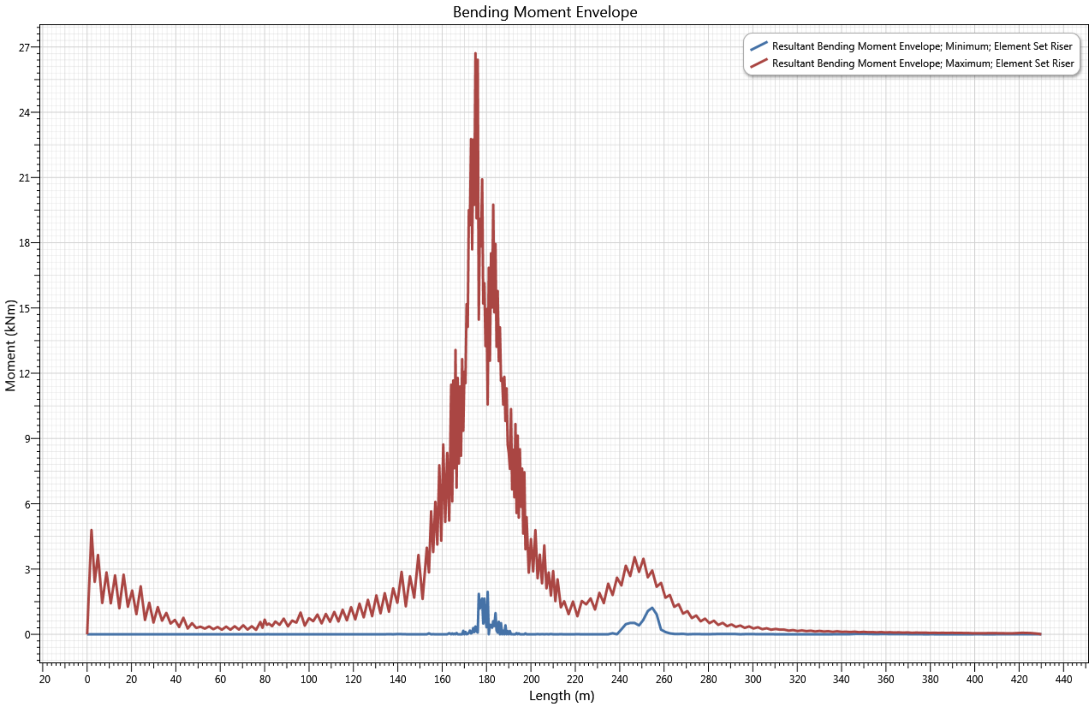
From the initial static analysis, the bending moment distribution is shown in the figure below.



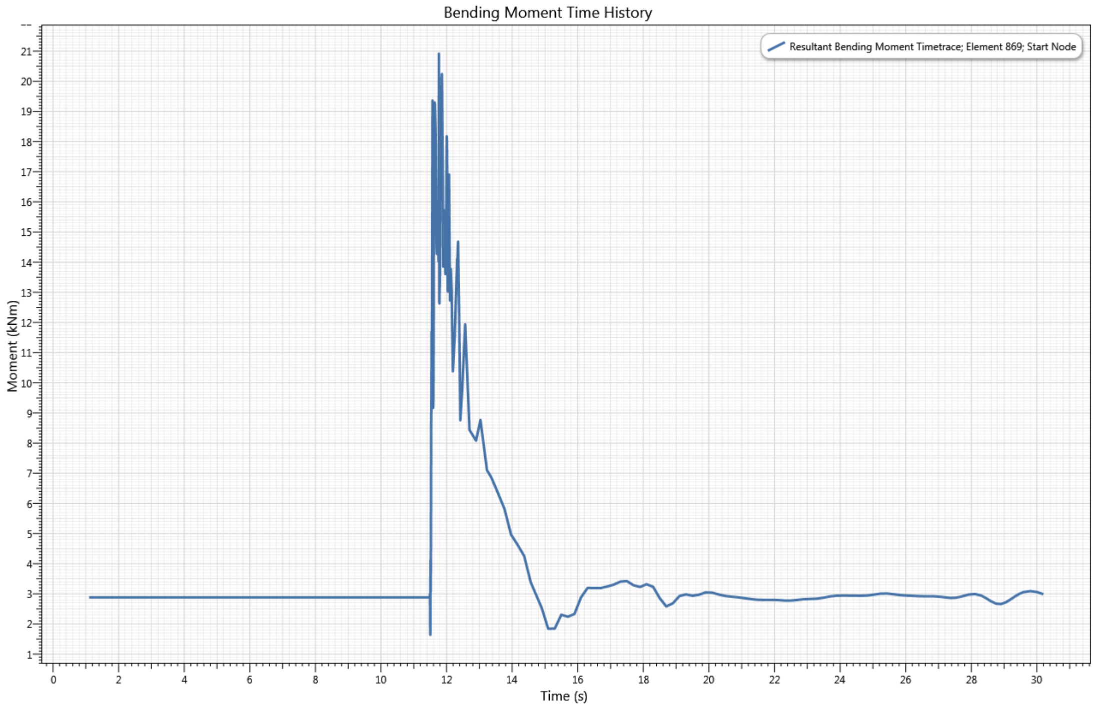
Initial Static Moments

DYNAMIC ANALYSIS

Sample dynamic results are given in the figures below. The bending moment envelope for the riser is shown in presented in the Bending Moment Envelope figure. The maximum moment naturally occurs in the contact region. The Bending Moment Time History figure below presents a time history of bending moment for location experiencing the greatest moment.

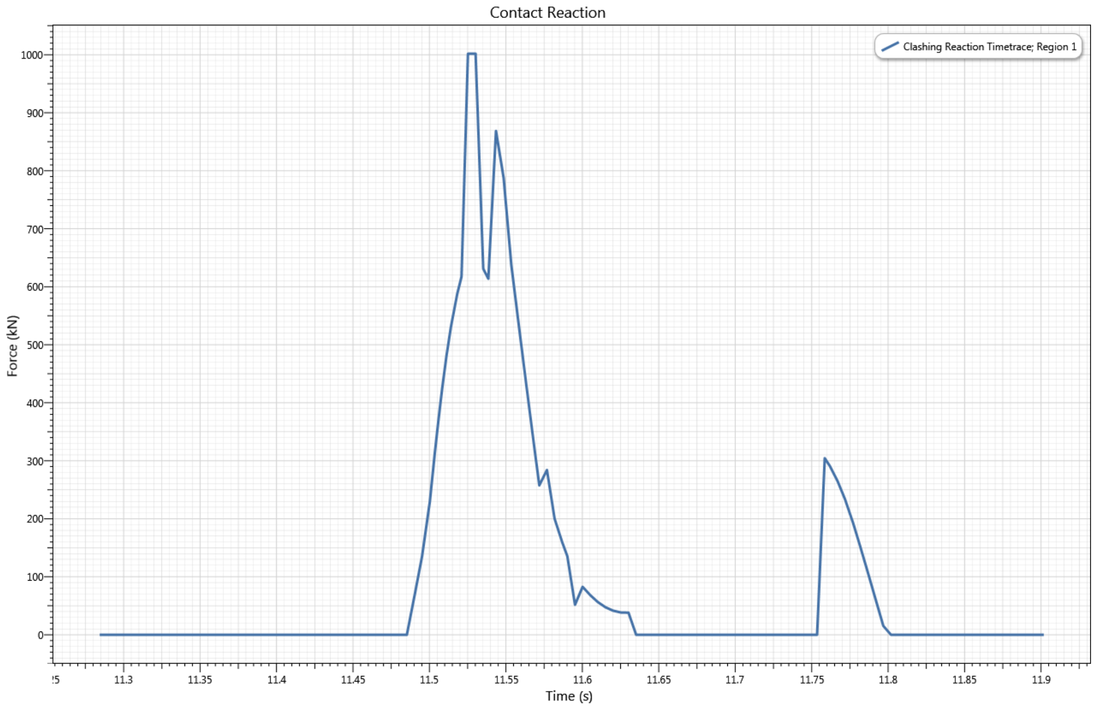


Bending Moment Envelope

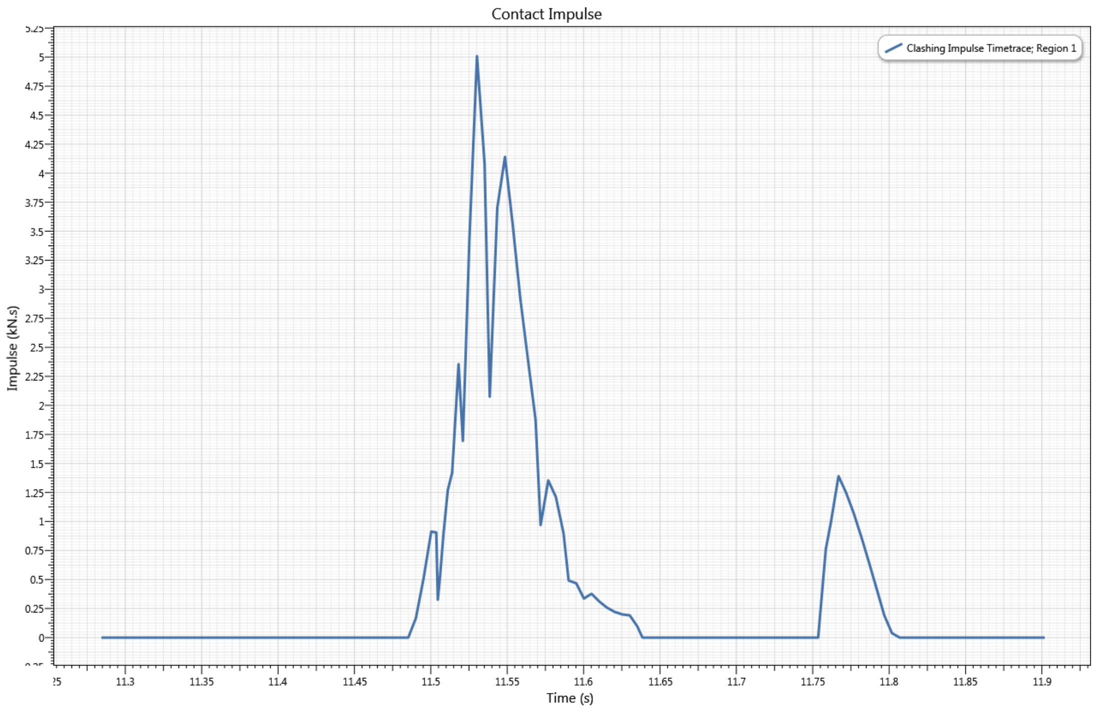


Bending Moment Time History

The two figures below present the total contact reaction and impulse, respectively, as a function of time. Note also that the clearance/interference analysis predicts an impact velocity of approximately 14.5m/s.



Contact Reaction



Contact Impulse

1.10.10.2 J02 - Advanced Database Postprocessing

OVERVIEW

This example illustrates the options a user has to access the database output files created by Flexcom. It is divided into the following sections:

- Database Access via Excel VBA describes how to set up a VBA macro in Microsoft Excel. It also contains some simple VBA macro code as an example and gives an overview of the VBA macro supplied with Flexcom.
- Database Access via Fortran DAR describes how to set up a similar Fortran program to access Flexcom database files.

FURTHER INFORMATION

- Excel Add-in provides details on every Database Access function available to the user within the workbook and VBA environments.
- Database Access Routines outlines each procedure available to another application (e.g. Fortran or C++ program) through the Database Access Library, together with details of the correct syntax and arguments.

Database Access via Excel VBA

Creating an Excel Visual Basic for Applications (VBA) macro is a powerful way of extracting data from a Flexcom results database. This method allows you to extract and process the data, do further calculations, or simply output it in a desired format. The use of conditional statements and loops is an efficient way of filtering and processing the data that is needed.

This section is divided into two articles:

- Node Positions Example describes, by way of a worked example, how to set up a simple VBA macro to access Flexcom database files.
- Element Statistics Example gives a brief overview of the methodology used to create a macro that calculates statistics such as mean and standard deviation for various Flexcom outputs.

Node Positions Example

OVERVIEW

This article describes the creation of a VBA macro in Microsoft Excel 2010. The procedure should be similar for other versions of Excel.

The objective of the macro is to write the position of every node in the model to a file. Since the database contains a number of solution times, it is further required that a new file be generated for each solution time. Each file will contain the node number and position of every node in the model for that particular solution time. It is written in a Comma Separated Value (CSV) format to facilitate subsequent parsing. Its name will convey the associated solution time (e.g. “Time=98.2s.csv”).

The database from the dynamic analysis of K01 - Worked Example - Simple will be used for demonstration. This example should be run in Flexcom before proceeding, in order to generate the required database files.

Refer to the VBA section of the Excel Add-in documentation for details on how to enable the Developer tab in Excel and set up a VBA macro.

Next, open a new Excel spreadsheet and proceed through the instructions below. Note that the completed Workbook is named “GetNodePositions.xlsm” in this example folder.

Cell C2 will contain the path to the database to be used and the macro will read this file name when it runs. This allows you to easily change the database to be processed without going in and editing the macro itself.

The code for this example is given below.

SAMPLE VBA MACRO CODE

```
Sub GeneratePositionFiles()

  Dim DB As New FlexcomDatabaseAccess.DatabaseAccess
  Dim DatabaseFile As String

  DatabaseFile = Range("C2").Value

  Dim error As Boolean
  If ((DB.IsValidDatabase(DatabaseFile))) Then
    Dim NumTimeSteps As Integer
    NumTimeSteps = DB.GetSolutionTimeCount(DatabaseFile)
```

```
            Dim NumNodes As Integer
            NumNodes = DB.GetNodeCount(DatabaseFile)

            For iTime = 1 To NumTimeSteps
              Dim SolutionTime As Single
              SolutionTime = DB.GetTime(DatabaseFile, iTime)

              Dim FilePath As String
              FilePath = Application.ActiveWorkbook.Path & "\Time="
              Open FilePath For Output As #1

              For iNode = 1 To NumNodes
                Dim UserNodeNumber As Integer
                UserNodeNumber = DB.GetUserNodeNumber(DatabaseFile,
                Dim x, y, z As Single
                x = DB.GetPosition(DatabaseFile, iTime, iNode, 1)
                y = DB.GetPosition(DatabaseFile, iTime, iNode, 2)
                z = DB.GetPosition(DatabaseFile, iTime, iNode, 3)

                Write #1, UserNodeNumber, x, y, z
              Next iNode

              Close #1

            Next iTime

          Else
          MsgBox ("Invalid Database")
          End If

        End Sub
```

CODE DETAILS

To specify the path of the database file which contains the necessary information, declare a new string parameter and use the *Range( ).Value* function as shown on line 6 of the sample VBA code.

An If-Then-Else statement is used to perform a validity check of the database file. If it is valid, the program proceeds with the calculations or else it exits the program and displays an error message. This is shown on lines 9, 39, 40 and 41.

For a valid database file, the number of solution times is extracted using the *DB.GetSolutionTimeCount( )* function and the number of nodes stored in the database is extracted using the *DB.GetNodeCount( )* function, shown on lines 10 to 14.

The program then loops over all the stored time steps. Within this loop, the current solution time is first found using the *DB.GetTime( )* function, given on lines 16 to 18.

Next, the path to the output file is defined as a string using the *Application.ActiveWorkbook.Path( )* function and it is opened to allow the program to write to the file, shown on lines 20 to 22.

The program then loops over each node stored in the database. The user-specified node number is obtained using the *DB.GetUserNodeNumber( )* function. This is given on lines 25 and 26. This step is performed as the user is able to arbitrarily assign numbers each node during the setup of the model. The output will thus be in agreement with the user node numbering used.

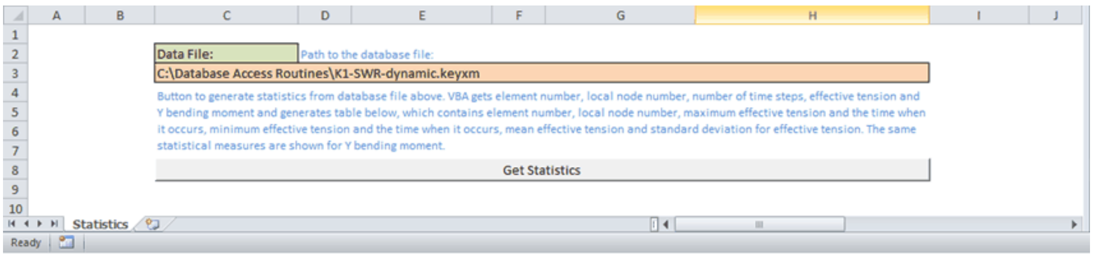
Next, the x, y and z coordinates of each node at the current time is found using the *DB.GetPosition( )* function shown on lines 27 to 30.

Finally, the current user specified node number and its coordinates are written to the output file on line 32, before moving to the next node number. Once the program has cycled over each node, the time is increased and the previous steps are repeated.

A full description of each function which may be used is given in the Excel Add-in section.

Element Statistics Example

This example shows how to generate a variety of statistical data in an Excel Worksheet ("VBA Database Access - Model Statistics.xlsm") for a Flexcom database file using Excel VBA macros. The input is the path to the database file and is written in cell "C3" of the Excel Spreadsheet.



Get Statistics button in Excel file

There are three macros associated with this example:

- The first VBA macro, “Main”, clears the charts in the current workbook and the data in the active spreadsheet and then calls the other two macros. Permission will be requested to delete each chart; select “Yes” each time.

- The second VBA macro, “Retrieve_Data”, reads the element number, local node number, number of time steps, effective tension and y bending moment from the model database. A table of statistical measures is then created. It contains the element number, local node number, maximum effective tension and the time when it occurs, minimum effective tension and the time when it occurs, mean effective tension and the standard deviation. The same statistical measures are also calculated for local-y bending moment.

- A third VBA macro, “Create_Charts”, creates two plots; minimum and maximum effective tension versus element number, and minimum and maximum y bending moment versus element number, in two new sheets.

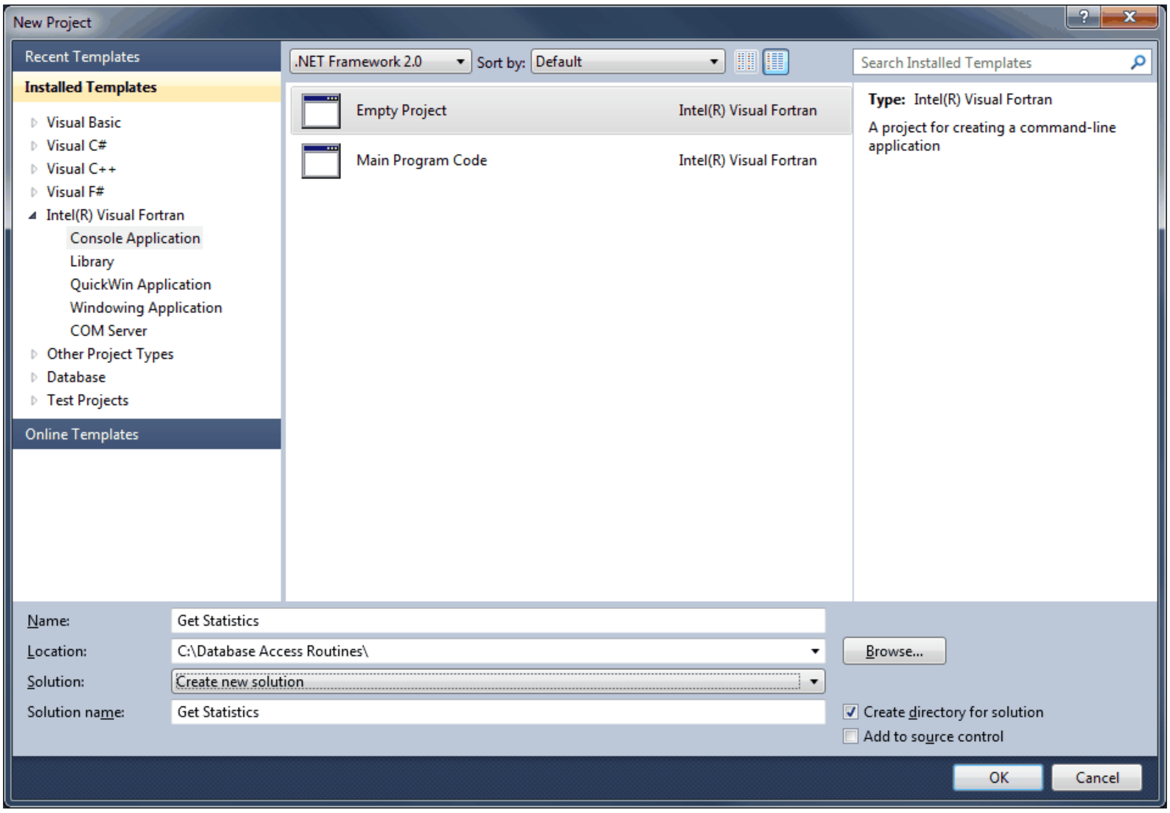
Database Access via Fortran DAR

OVERVIEW

As an alternative to using the VBA environment in Excel, a Fortran program can also be created to generate the same results as the previous Element Statistics Example and write them to a text file. As before, the database from the dynamic analysis of Flexcom Example K1 will be used for demonstration. This example should be run in Flexcom before proceeding, in order to generate the required database files.

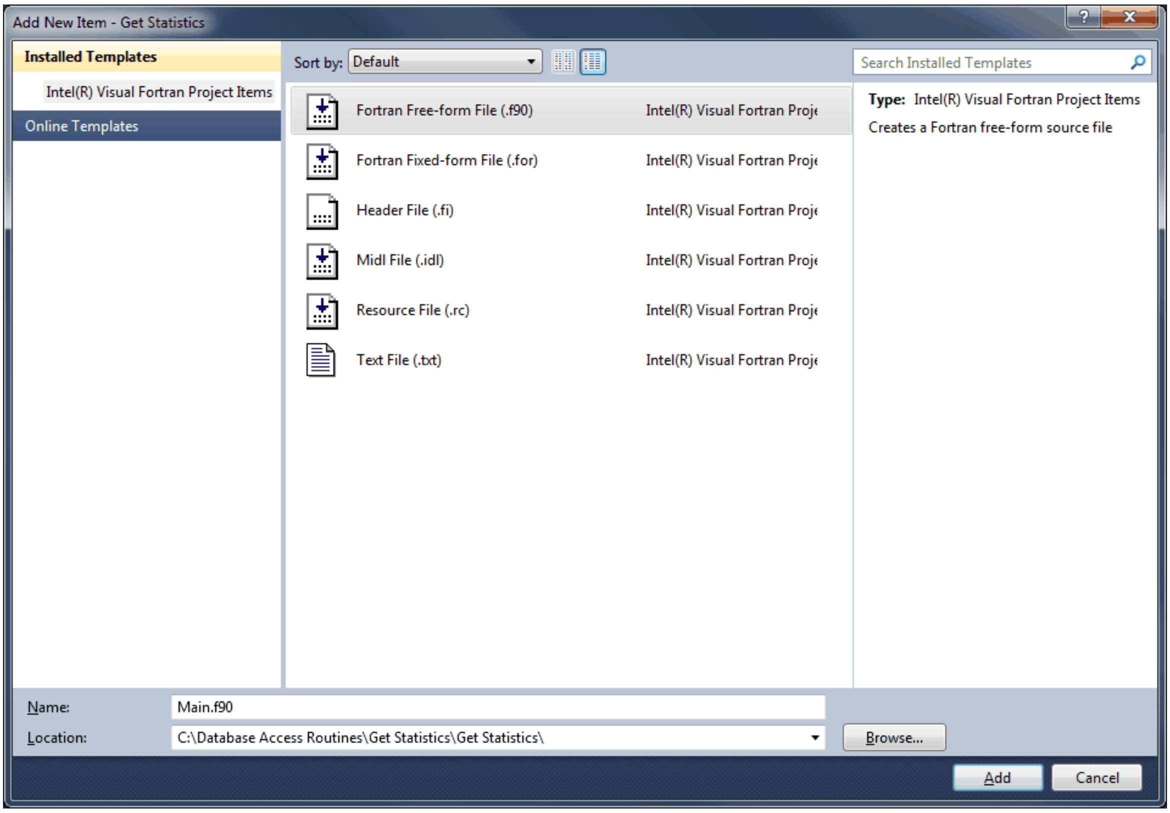
MICROSOFT VISUAL STUDIO SETUP

Firstly, create a new, empty Fortran project with a suitable name and location.



Creating an empty Fortran project

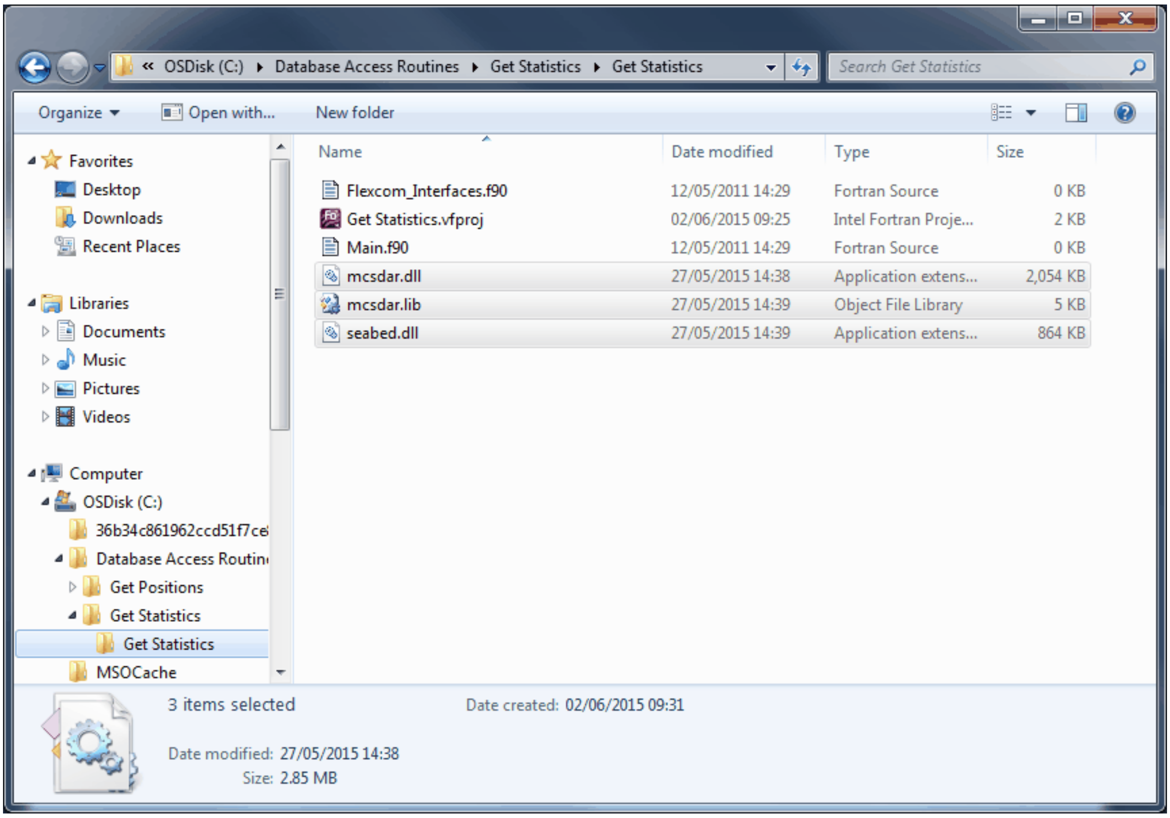
Next, add a new file by right-clicking on Source Files in the solution explorer and selecting Add and New Item. Create a Fortran Free-Form File (.f90) called *Main*, or otherwise, according to your own naming conventions.



Creating *Main* Fortran file

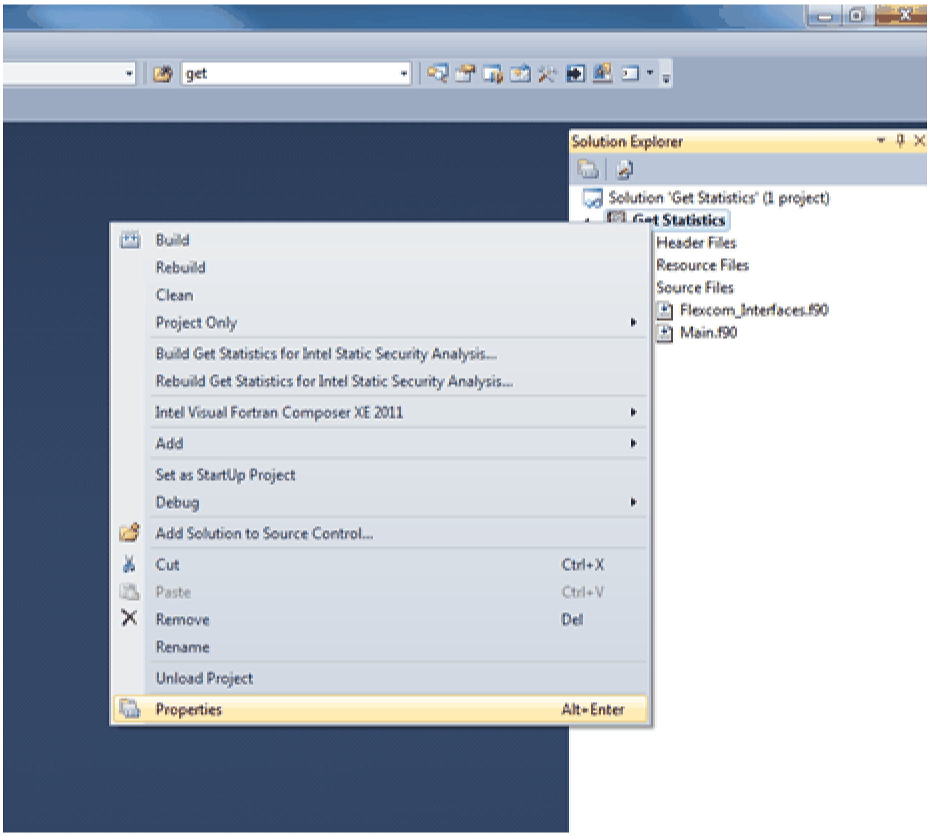
Similarly, add the *Flexcom_Interfaces.f90* file by choosing 'add an existing file' and navigating to the location of the file. This file contains the procedures used during Database Access Routines.

Next, copy and paste *mcsdar.lib, mcsdar.dll* and *seabed.dll* from the Flexcom bin folder into the same folder as the .vfproj file.



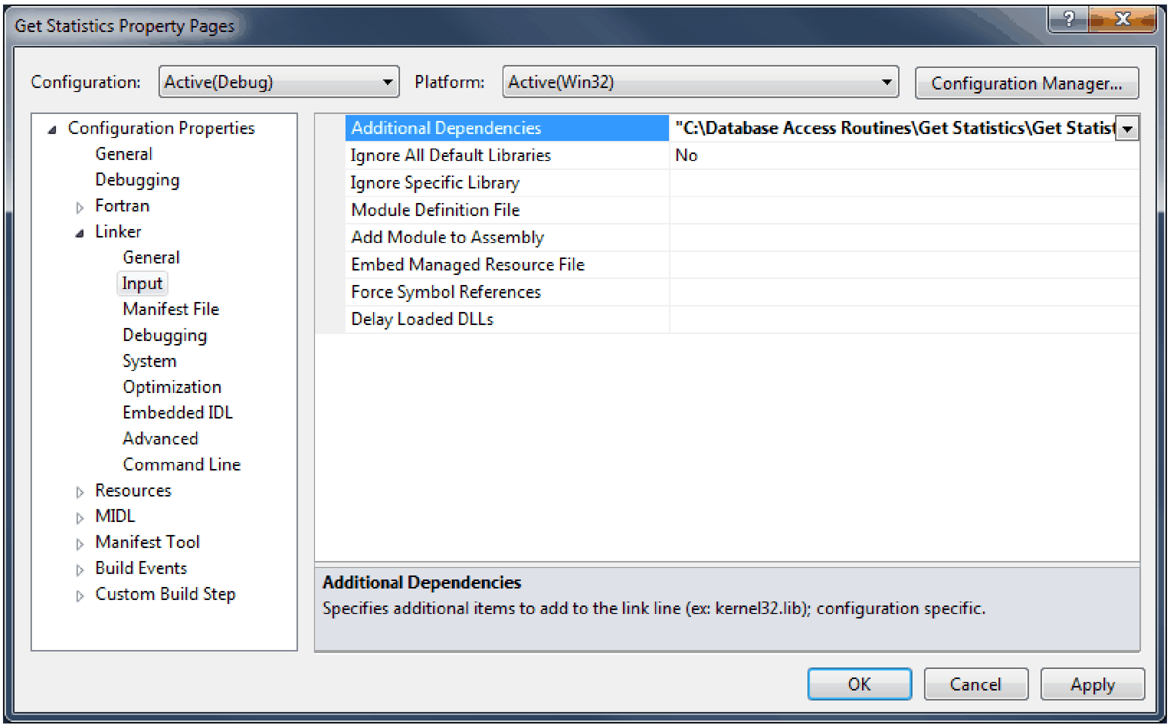
Adding the required libraries and DLLs

Finally, in Visual Studio, right-click on the project name and select properties



Accessing the project properties

Navigate to the Input section under the Linker heading and insert the path to the location of *mcsdar.lib* into the Additional Dependencies box. It may be necessary to enclose the path in a set of inverted commas (for example, "C:\Database Access Routines\Get Positions\Get Positions\mcsdar.lib").



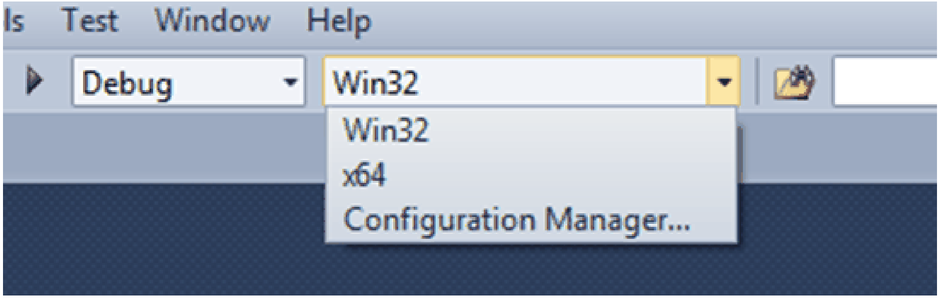
Including mcsdar.lib as an additional dependency in the project Linker

Within the program in Main.f90, it is necessary to include the text:

```
use Flexcom_interfaces
```

This will call the Database Access Routines located in Flexcom_Interfaces.f90 and will allow you to use the predefined Database Access procedures.

It is important to note when debugging the project using Visual Studio that the processor used corresponds to the equivalent DLLs and libraries. When debugging using the 32 bit processor, it is important to use the DLLs and libraries from the Bin folder of the 32 bit version of Flexcom. Similarly, DLLs and libraries from the 64 bit installation of Flexcom are used with the 64 bit debugging processor. The processor to be used can be selected from the drop-down on the main toolbar or in the Configuration Manager, accessed through the Build menu.



Choosing the correct debugging processor

SAMPLE FORTRAN CODE

Two sample Fortran solution files are located in the example folder. One, named Fortran (x32), should be used if you have installed the 32 bit version of Flexcom, while the second, named Fortran (x64), should be used if you have installed the 64 bit version of Flexcom. Each of the project files contain code with supplementary comments to help the reader understand the process used.

In brief, *main.f90* first uses the GetDatabaseInfo function to find the number of parameters in the model. This allows the number of elements and number of timesteps to be found.

Next the GetRequestedHistory function is used to find the output analysis times in the model.

The program then loops over each element and local node number. Within this loop, the *GetRequestedHistory* function is used again to get the Effective Tension and Local-Y Bending Moment of the particular node. The maximum, minimum and mean Effective Tension and Local-Y Bending Moment is then calculated.

Finally, the variance, and subsequently the standard deviation, is calculated and the above results are written to a text file.

1.10.10.3 J03 - Summary Postprocessing Collation

This example illustrates how to quickly simulate a large load case matrix using Flexcom's parameter variation feature. It also demonstrates the power and versatility of the summary collation feature, which allows you to quickly assemble a dashboard of pertinent information from the entire load case.

A flexible riser in steep wave configuration is used as a backdrop for this example. The model itself is relatively straightforward, for reasons of simplicity and run-time efficiency, as the real focus of this example is on the Keyword Parameterisation and Summary Postprocessing Collation features.

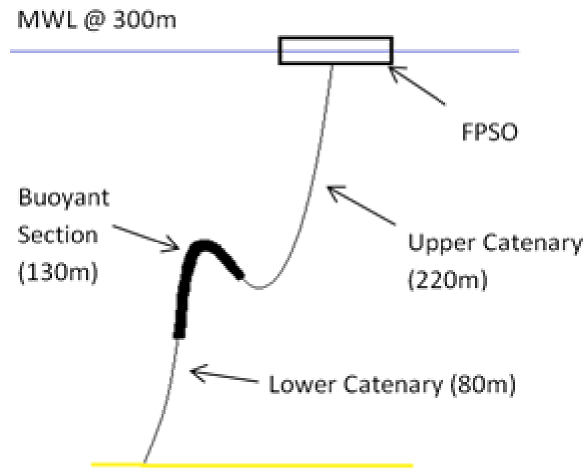
The overall layout of this example is as follows:

- Riser System provides a brief overview of the flexible riser system.
- Load Case Simulation describes how parameters and variations are used to quickly create all the keyword files necessary to simulate the entire load case matrix.
- Postprocessing outlines how the summary output data is requested from the individual simulations, and how each individual simulation within the load case matrix is assigned key parameters for identification purposes.
- Collation illustrates how to identify folders on the hard drive for inclusion in the collation process, and how to request the generation of 3D plots.
- Spreadsheet Output presents a sample list of collated parameters along with their extreme values.
- 3D Plots graphically presents the variation of some sample outputs against the key driving parameters of wave period and wave heading.

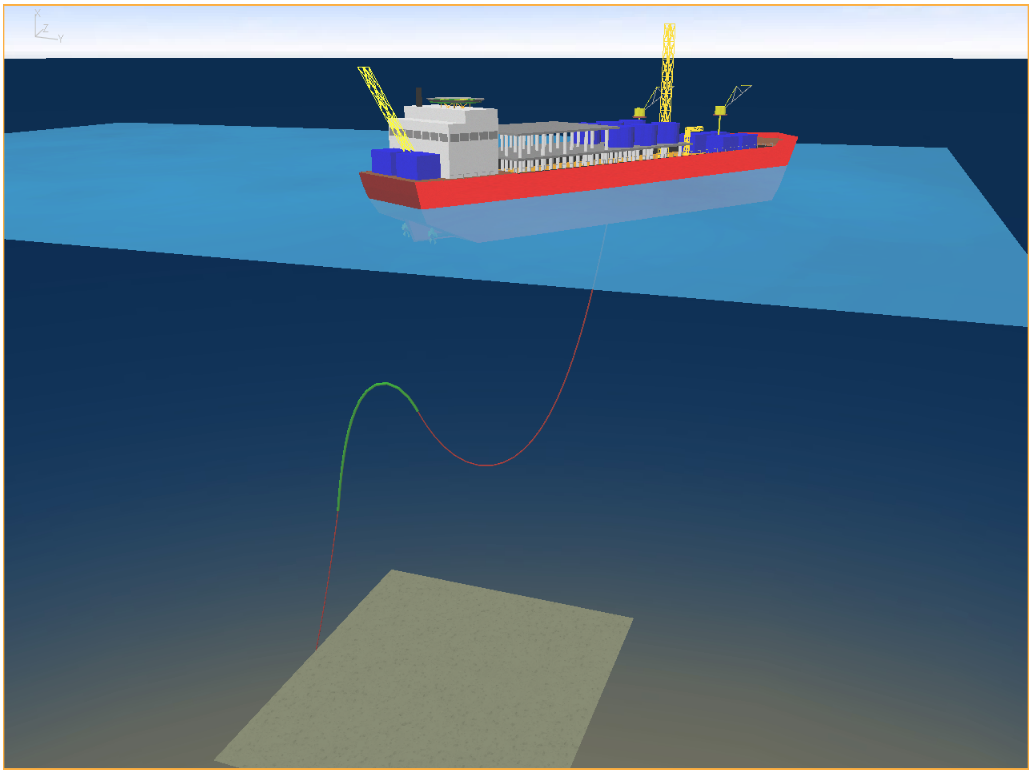
Riser System

This example considers the analysis of a flexible riser in steep wave configuration. The steep wave riser configuration is shown below. The riser in this example is 290m in length and is sited in 200m of water. It is filled with oil and attached to a production vessel.

The analysis is performed in two stages. The initial static analysis applies gravity and buoyancy forces. The dynamic analysis finds the response of the structure to vessel motions and wave loading.



Model Schematic



Riser in Model View

Load Case Simulation

LOAD CASE MATRIX

This example considers a large load case matrix. Specifically, wave periods ranging from 5s to 25s (at 1s intervals), are examined at incident wave headings which vary from 0 degress to 345 degrees (at 15 degree intervals). This results in 624 regular wave load cases in total. Using the Keyword Parameterisation allows you to create all the required input files to model every possible combination of wave period and incident heading.

PARAMETERS

Naturally two separate Parameters are defined to wave period and wave heading respectively. When the load case is expanded, these variables will assume values which have different numbers of digits. For example, wave directions of 0 (1 digit), 45 (2 digits) and 150 (3 digits). Given that the name of each individual keyword file will be based on the particular load case combination it represents, it is useful to define two further parameters which contain the same wave direction and heading information, but which are neatly formatted to ensure a consistent number of digits is always used (the FORMAT utility function is very helpful).

```
*PARAMETERS
 Direction, 0
 Period, 10
 DirectionStr, =[format(Direction, "000")]
 PeriodStr, =[format(Period, "00")]
```

Parameters

VARIATIONS

Keyword Based Variations are used to define wave periods which range from 5s to 25s, and wave headings which range from 0 degrees to 345 degrees. This facility is ideally suited to parameters which vary in fixed increments.

```
*VARIATION
 NAME=Periods
 PARA=Period, GEN=5, 25, 1
 NAME=Directions
 PARA=Direction, GEN=0, 345, 15
```

Variations

Although this example focuses on regular wave loading, a similar approach would also be used to simulate random seas, where seastate information is typically presented using a Scatter Diagram. Given that the driving variables (i.e. Hs, Tz etc.) are generally not round integers, the use of Spreadsheet Based Variations would be more appropriate in this case.

COMBINATIONS

The combinations input is used to control the names and locations of generated keyword files. In this case, a separate folder is established for every wave heading, and individual file names within each folder reflect the relevant wave period.

When you run the master template file (J3-SC-Regular Wave Template.keyxm), Flexcom automatically creates 624 separate keyword files for you, each one corresponding to a unique combination of wave period and heading.

```
*COMBINATIONS
 DESCRIPTION=Waves, SUBDIRECTORY="%DirectionStr% degrees", FILENAME="Wave %PeriodStr%s"
 VARIATIONS=ALL
```

Combinations

OTHER INPUT VARIABLES

Regular wave loading is defined in the usual way, referencing the predefined wave period and direction parameters.

Appropriate time variables are computed based on the regular wave period. Purely for efficiency reasons, the regular wave simulations are analysed for 3 periods only, whereas 5-10 periods would generally be considered recommended practice. The loads are ramped on over the first wave period. A fixed time step equal to 1/20th of the regular wave period is used.

Customised database storage options are specified, reducing the number of parameters that are output to the database files for disk and computational efficiency reasons.

RELEVANT KEYWORDS

- `*PARAMETERS` is used to define parameters whose names may be referenced subsequently in the definition of other input variables.
- `*VARIATION` is used to define parameter or keyword variations.

- *COMBINATIONS is used to control the names and locations of generated keyword files, along with the required analysis combinations.
- *EXCEL VARIATIONS is used to generate keyword files based on a parameter matrix contained in an Excel workbook.
- *WAVE-REGULAR is used to specify regular Airy wave loading.
- *TIME is used to define time parameters for an analysis.
- *DATABASE CONTENT is used to customise the contents of the database output files.

Postprocessing

INTRODUCTION

Flexcom provides a powerful facility for generating a Summary Output File. Here the maximum value, the minimum value, the range of values and the standard deviation of values is tabulated for a group of parameters you specify. The output is in a succinct format and is suitable for inserting directly into a study report.

OUTPUT REQUESTS

In this example, four separate types of output are requested.

- Effective tension at the top of the riser. The largest tension values occur at the hang-off point.
- Maximum bending moment in the riser. The largest moments occur in the more curved regions of the model, such as the peak of the buoyant section, or the trough at the lower end of the upper catenary.
- Maximum Von Mises stress in the riser.
- The angle between the riser and the vessel at the hang-off point.

```
$SUMMARY POSTPROCESSING
*PARA FORCE
 TITLE=Effective Tension at Riser Top
 SCALE=1.0, UNITS=kN
 PLOT=NO
 DATA1={Upper Catenary_Last}
 DATA2=3
 DATA3=7
*PARA FORCE ENVELOPE
 TITLE=Bending Moment in Riser
 SCALE=1.0, UNITS=kNm
 PLOT=NO
 DATA1=All
 DATA2=8
 TITLE=Von Mises Stress in Riser
 SCALE=1.0, UNITS=MPa
 PLOT=NO
 DATA1=All
 DATA2=13
*PARA ANGLE AXIS
 TITLE=Angle between Riser Top and Vessel
 SCALE=1.0, UNITS=degrees
 PLOT=NO
 DATA1={Upper Catenary_Last}
 DATA2=Vertical
 DATA3=1
*AXIS/VECTOR
 VECTOR=Vertical
 {Upper End}, 1, 0, 0
```

Summary Postprocessing Requests

After a simulation has completed, Flexcom produces a text-based Summary Output File for visual inspection, but more importantly it also creates a Summary Database File, which is effectively a binary version of the same data, for subsequent collation.

IDENTIFICATION PARAMETERS

It is also necessary to define some key parameters which uniquely identify each individual simulation within the load case matrix. In this example, wave period and wave direction are defining parameters in the regular wave load case matrix, and serve as ideal parameters for identification purposes.

```
$SUMMARY POSTPROCESSING
*COLLATE PLOT AXES
 Period, =[Period]
 Direction, =[Direction]
```

Identification Parameters

RELEVANT KEYWORDS

- `*PARA FORCE` is used to request summary output of element restoring forces.
- `*PARA FORCE ENVELOPE` is used to request summary output of statistics of element restoring forces.
- `*PARA ANGLE AXIS` is used to request summary output of the angle between an element and either a vector or an axis system.
- `*AXIS/VECTOR` is used to define axis systems and vectors for use in postprocessing.
- `*COLLATE PLOT AXES` is used to define a number of key parameters which uniquely identify a particular simulation within a load case matrix. This is a prerequisite for generating 3-dimensional Summary Collation Plots.

Collation

INTRODUCTION

If you are performing a series of analyses (for example to examine a large number of different load cases), the Summary Postprocessing Collation facility provides a useful means of assembling all the pertinent summary data across a range of load cases into a single Summary Collation Spreadsheet. Enhanced data visualisation is also provided by the ability to produce 3-dimensional Summary Collation Plots. You can plot the variation of any summary postprocessing output against any key driving parameters. For example, you can plot maximum effective tension as a function of both wave period and incident wave heading in a 3-dimensional space.

COLLATION INFORMATION

In this example, the following information is specified to govern the collation process.

- Four different parameters are explicitly requested for inclusion in the collation process. Strictly speaking, this is not absolutely necessary, as Flexcom will simply collate all available data in the absence of specific guidance. However, inclusion of this information allows you to assign custom units to the collated parameters.

- All sub-folders of the current working directory are designated for inclusion in the collation process.

- Four separate 3D plots are requested, corresponding to the output parameters which were already requested as part of the individual simulations.

```
$SUMMARY COLLATE
*COLLATE
 TITLE="Angle between Riser Top and Vessel", UNITS=degrees
 TITLE="Effective Tension at Riser Top", UNITS=kN
 TITLE="Bending Moment in Riser", UNITS=kNm
 TITLE="Von Mises Stress in Riser", UNITS=MPa

*IDENTIFY
 OPTION=SUBFOLDERS

*PLOT
 TITLE="Angle between Riser Top and Vessel", PRIMARY AXIS=Period, SECONDARY AXIS=Direction
 TITLE="Effective Tension at Riser Top", PRIMARY AXIS=Period, SECONDARY AXIS=Direction
 TITLE="Bending Moment in Riser", PRIMARY AXIS=Period, SECONDARY AXIS=Direction
 TITLE="Von Mises Stress in Riser", PRIMARY AXIS=Period, SECONDARY AXIS=Direction
```

Summary Collation Requests

RELEVANT KEYWORDS

- `*COLLATE` is used to specify the summary postprocessing data to be collated and any exclusion criteria. This keyword is optional, and if you do not explicitly designate certain parameters for collation purposes, Flexcom will attempt to collate all available data.

- `*IDENTIFY` is used to identify output files for inclusion in summary postprocessing collation.

- `*PLOT` is used to request the creation of a Summary Collation Plot which graphically presents the variation of any summary postprocessing output against key driving parameters.

Spreadsheet Output

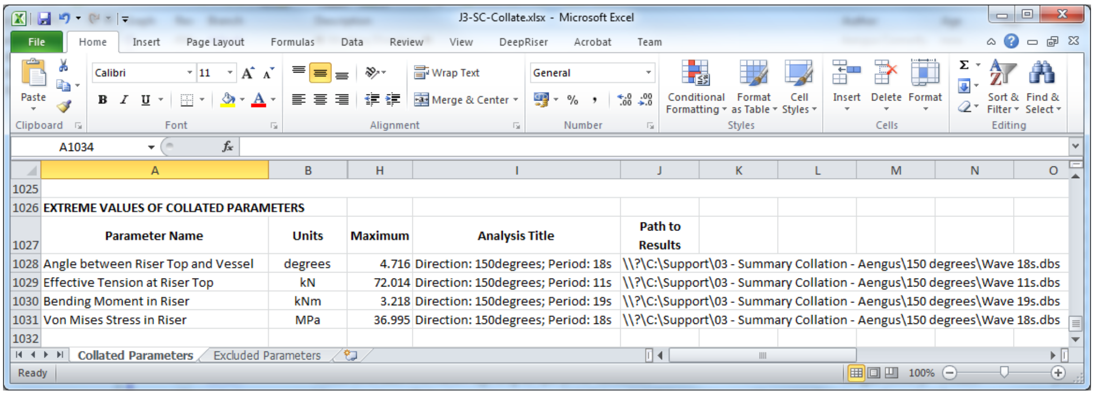
INTRODUCTION

The collation facility provides a useful means of assembling all the pertinent summary data across a range of load cases into a single Summary Collation Spreadsheet. This spreadsheet contains a table of data listing the parameters of interest from all of the individual simulations, including maximum, minimum and mean values, range and standard deviation.

Towards the end of the spreadsheet, a useful summary of extreme values is also presented. For every parameter which is collated, this section highlights the most critical analyses in the load case matrix, both in terms of greatest maxima and greatest minima attained.

RESULTS

The extreme values of the collated parameters across all the individual simulations are presented as shown below. The most critical wave direction is 150 degrees, as all the parameters considered attain their maximum values at this direction. The most critical wave period appears to be in the region of 18s-19s where both bending moment and von Mises stress reach maximum values.



EXTREME VALUES OF COLLATED PARAMETERS

Parameter Name	Units	Maximum	Analysis Title	Path to Results
Angle between Riser Top and Vessel	degrees	4.716	Direction: 150degrees; Period: 18s	\\?\C:\Support\03 - Summary Collation - Aengus\150 degrees\Wave 18s.dbs
Effective Tension at Riser Top	kN	72.014	Direction: 150degrees; Period: 11s	\\?\C:\Support\03 - Summary Collation - Aengus\150 degrees\Wave 11s.dbs
Bending Moment in Riser	kNm	3.218	Direction: 150degrees; Period: 19s	\\?\C:\Support\03 - Summary Collation - Aengus\150 degrees\Wave 19s.dbs
Von Mises Stress in Riser	MPa	36.995	Direction: 150degrees; Period: 18s	\\?\C:\Support\03 - Summary Collation - Aengus\150 degrees\Wave 18s.dbs

Summary of Extreme Values

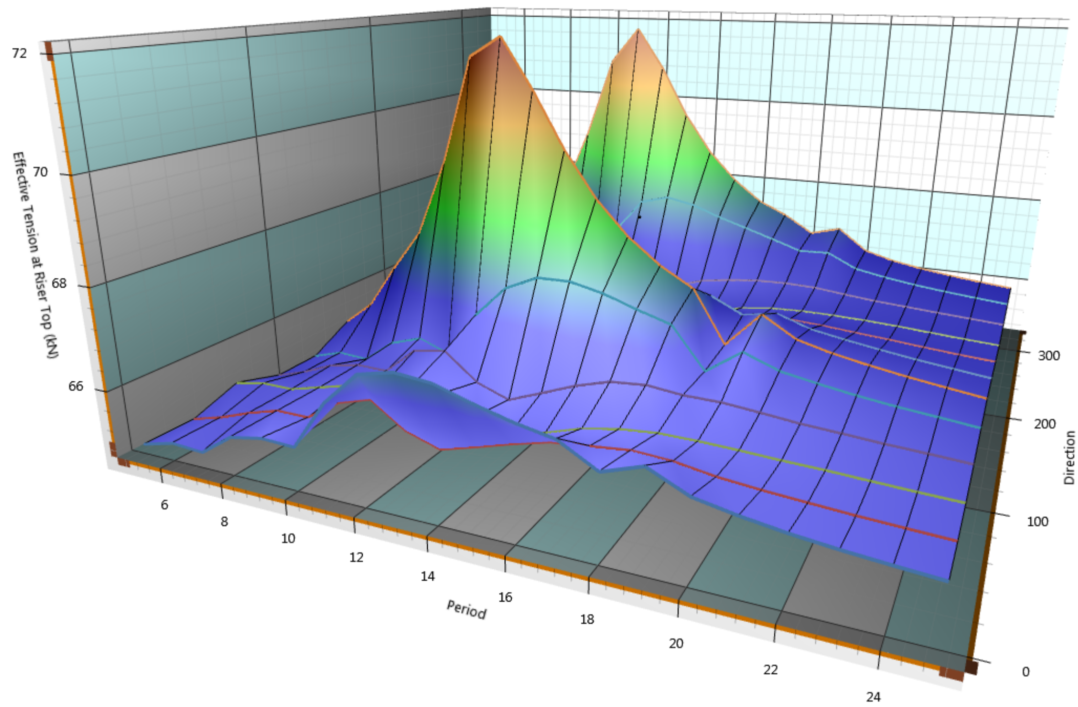
3D Plots

INTRODUCTION

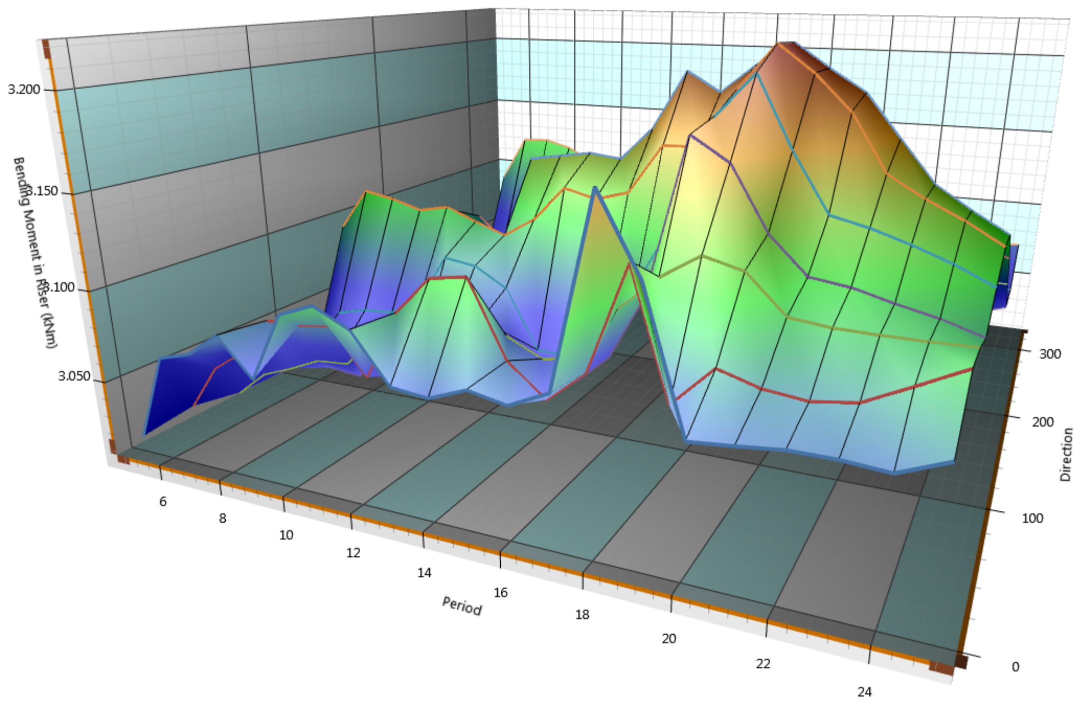
Once all the individual simulations have been completed, you can plot the variation of any output against key driving parameters using a 3-dimensional Summary Collation Plot.

RESULTS

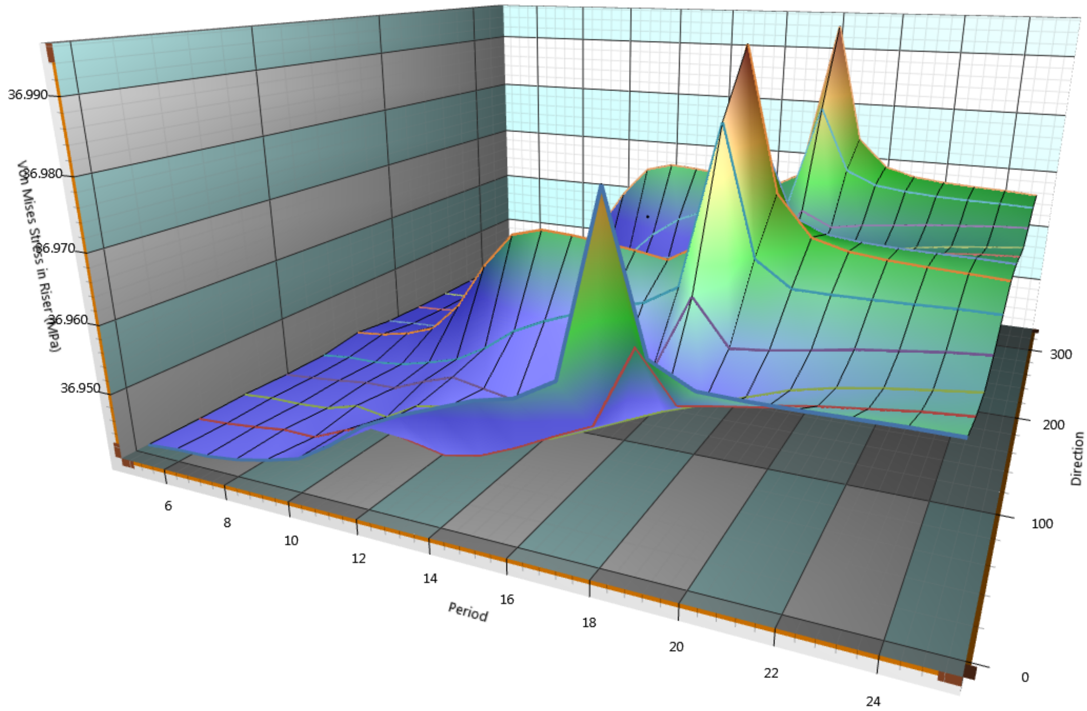
Four separate 3D plots are created in this example, corresponding to the output parameters which were already requested as part of the individual simulations. Wave period and wave direction are plotted on the horizontal axes.



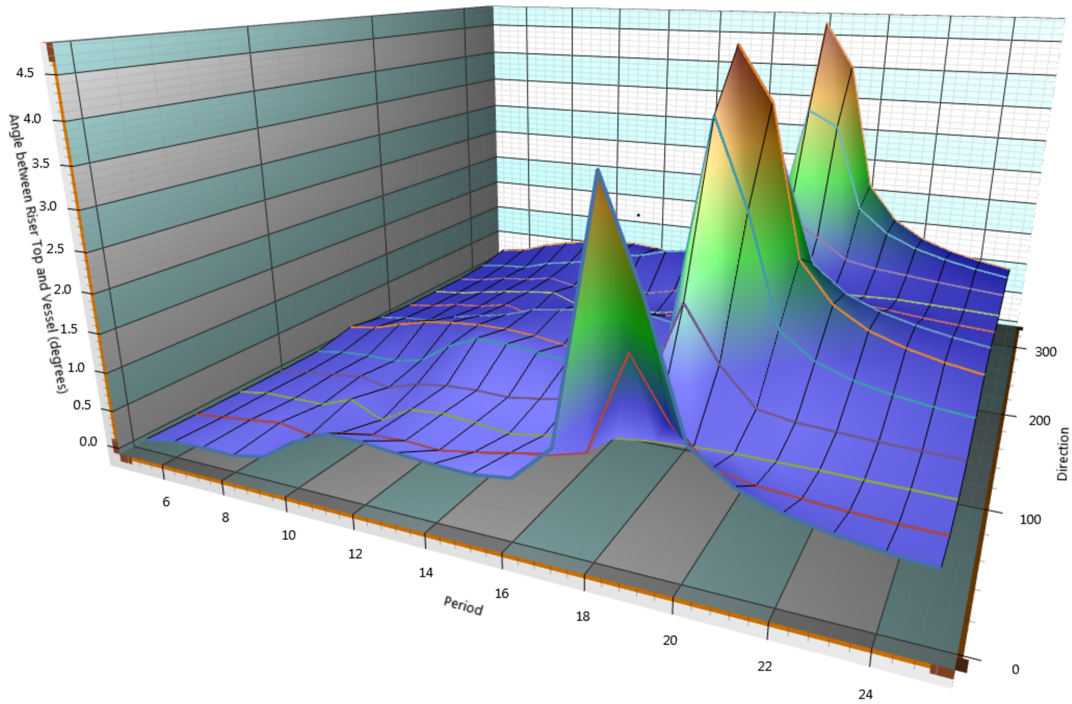
Maximum Effective Tension at Riser Top



Maximum Bending Moment in Riser



Maximum Von Mises stress in Riser



Maximum Hang-off Angle

1.10.10.4J04 - User Solver Variables

OVERVIEW

Flexcom now provides advanced user subroutines, denoted as User Solver Variables to distinguish from legacy features, which provide much greater flexibility and control. The user is empowered by the ability to access a host of solution variables including nodal kinematic variables (displacements, velocities, accelerations) and elemental restoring forces (effective tension, bending moment, curvature, torque etc.). Equipped with this information it is then possible to directly augment the global force vector to simulate an arbitrary time-varying load. It is even possible to directly modify the constitutive finite element matrices (i.e. stiffness and mass) should you have very specialised modelling requirements. Naturally this requires some programming expertise, but it provides complete generality for power users.

EXAMPLES

The following sections present some illustrative examples to help upskill users and demonstrate the potential of this powerful modelling feature.

- J04a - Follower Force illustrates the application of arbitrary loading. The direction of the applied force is dependent on the instantaneous position of the point of application. The use of custom code enables the definition of a force which can track/follow the direction of response, via continuous adjustment of the global force vector.
- J04b - Sphere Contact illustrates how to model contact with a spherical surface, a modelling feature which is not currently available as standard in Flexcom. This is a more complicated example which involves augmentation of both the global stiffness matrix and the global force vector.
- Mass matrix - although not part of a formal example, this section provides some sample source code which enables you to write the global mass matrix to the output file.

J04a - Follower Force

INTRODUCTION

This example illustrates the application of arbitrary loading via the User Solver Variables feature. The direction of the applied force is dependent on the instantaneous position of the point of application. The use of custom code enables the definition of a force which can track/follow the direction of response, via continuous adjustment of the global force vector.

MODEL

The model consists of a simple horizontal beam of length L, which is rigid and massless. It is constrained at one end in all degrees of freedom except for rotation in one degree of freedom, such that in plan view, it is free to rotate about a fixed pivot point. A point mass, m, is positioned at the free end of the beam. A force of magnitude F is applied to the free end in order to induce some rotation. Based on the instantaneous orientation of the beam, the direction of this force is dynamically updated such that it always remains perpendicular to the free end. The custom code uses an appropriate local to global transformation matrix to convert the local force terms into the global axis system before augmenting the global force vector. In order to verify correct model behaviour, results are compared with an analytical solution, and also an analogous model which applied a constant rotational moment at the fixed end.

THEORETICAL SOLUTION

The theoretical angular acceleration is defined as follows:

$$a = F/(m * L) \quad (1)$$

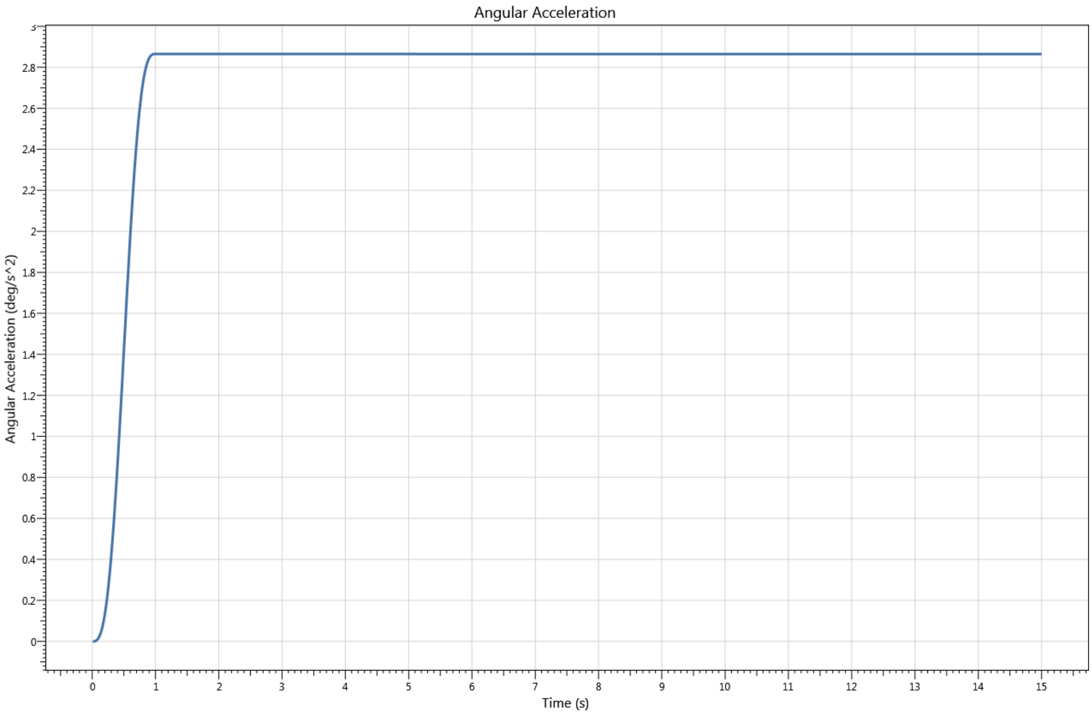
where

- a is acceleration in rads/s^2
- F is the applied force in N
- m is the point mass in kg
- L is the beam length

Based on the above inputs, the angular acceleration at the free end should be 2.86degs/s^2.

RESULTS

The rotational acceleration predicted by Flexcom for the customised model is exactly 2.86degs/s^2 as expected.



Angular Acceleration at Fixed End

SOURCE CODE

Pre-compiled DLLs for both 32-bit and 64-bit operating systems are provided with this example. The original Fortran source code is also provided should you wish to examine it.

```
subroutine
user_solver_variables(iter,time,ramp,nnode,nncon,ncord,nndof,nelmn,necon,nedof,
&
  size_force,size_mass,size_stiff,elmcon,nodcon,intoun,ietoue,cord,displacement,
           &
  velocity,acceleration,trgb,tglu,disp_prev,pos_prev,vel_prev,acc_prev,axial_force,y
_shear,    &
  z_shear,torque,y_bending,z_bending,eff_tension,y_curvature,z_curvature,force,mass,
stiff)
!dec$ attributes dllexport, stdcall, reference :: user_solver_variables
  implicit none

! Input variables - cannot be modified within this subroutine

  integer, intent(in)    :: iter
  !dec$ attributes value :: iter
  real(8), intent(in)    :: time
  !dec$ attributes value :: time
  real(8), intent(in)    :: ramp
  !dec$ attributes value :: ramp
  integer(4), intent(in) :: nnode
  !dec$ attributes value :: nnode
  integer(4), intent(in) :: nncon
  !dec$ attributes value :: nncon
  integer(4), intent(in) :: ncord
  !dec$ attributes value :: ncord
  integer(4), intent(in) :: nndof
  !dec$ attributes value :: nndof

  integer(4), intent(in) :: nelmn
  !dec$ attributes value :: nelmn
  integer(4), intent(in) :: necon
  !dec$ attributes value :: necon
  integer(4), intent(in) :: nedof
  !dec$ attributes value :: nedof

  integer(4), intent(in) :: size_force
  !dec$ attributes value :: size_force
  integer(4), intent(in) :: size_mass
  !dec$ attributes value :: size_mass
  integer(4), intent(in) :: size_stiff
  !dec$ attributes value :: size_stiff

  integer(4), intent(in), dimension(nncon, nelmn)              :: elmcon
  !dec$ attributes reference :: elmcon
  integer(4), intent(in), dimension(necon,nnode)               :: nodcon
  !dec$ attributes reference :: nodcon
  integer(4), intent(in), dimension(nnode)                     :: intoun
  !dec$ attributes reference :: intoun
  integer(4), intent(in), dimension(nelmn)                     :: ietoue
  !dec$ attributes reference :: ietoue
  real(8),  intent(in), dimension(ncord,nnode)                 :: cord
  !dec$ attributes reference :: cord
  real(8),  intent(in), dimension(nndof,nnode)                 :: displacement
  !dec$ attributes reference :: displacement
```

```
  real(8),  intent(in), dimension(nndof,nnode)                 :: velocity
  !dec$ attributes reference :: velocity
  real(8),  intent(in), dimension(nndof,nnode)                 :: acceleration
  !dec$ attributes reference :: acceleration
  real(8),  intent(in), dimension(3,3,nelmn)                   :: trgb
  !dec$ attributes reference :: trgb
  real(8),  intent(in), dimension(3,3,nelmn)                   :: tglu
  !dec$ attributes reference :: tglu
  real(8),  intent(in), dimension(nndof,nnode)                 :: disp_prev
  !dec$ attributes reference :: disp_prev
  real(8),  intent(in), dimension(nndof,nnode)                 :: pos_prev
  !dec$ attributes reference :: pos_prev
  real(8),  intent(in), dimension(nndof,nnode)                 :: vel_prev
  !dec$ attributes reference :: vel_prev
  real(8),  intent(in), dimension(nndof,nnode)                 :: acc_prev
  !dec$ attributes reference :: acc_prev
  real(8),  intent(in), dimension(nelmn,3)                     :: axial_force
  !dec$ attributes reference :: axial_force
  real(8),  intent(in), dimension(nelmn,3)                     :: y_shear
  !dec$ attributes reference :: y_shear
  real(8),  intent(in), dimension(nelmn,3)                     :: z_shear
  !dec$ attributes reference :: z_shear
  real(8),  intent(in), dimension(nelmn,3)                     :: torque
  !dec$ attributes reference :: torque
  real(8),  intent(in), dimension(nelmn,3)                     :: y_bending
  !dec$ attributes reference :: y_bending
  real(8),  intent(in), dimension(nelmn,3)                     :: z_bending
  !dec$ attributes reference :: z_bending
  real(8),  intent(in), dimension(nelmn,3)                     :: eff_tension
  !dec$ attributes reference :: eff_tension
  real(8),  intent(in), dimension(nelmn,3)                     :: y_curvature
  !dec$ attributes reference :: y_curvature
  real(8),  intent(in), dimension(nelmn,3)                     :: z_curvature
  !dec$ attributes reference :: z_curvature

! Output variables - can be modified within this subroutine if required
  real(8),  intent(inout), dimension(size_force,nedof)         :: force
  !dec$ attributes reference :: force
  real(8),  intent(inout), dimension(nedof,nedof,size_mass)    :: mass
  !dec$ attributes reference :: mass
  real(8),  intent(inout), dimension(nedof,nedof,size_stiff)   :: stiff
  !dec$ attributes reference :: stiff

! Variable names
!     iter                : Current iteration
!     time                : Current timestep
!     ramp                : Ramp
!     nnode               : Number of nodes in the model
!     nncon               : Number of nodes with connected elements
!     ncord               : Number of coordinates
!     nndof               : Number of degrees of freedom per node (6)
!     nelmn               : Number of elements in the model
!     necon               : Number of elements connected
!     nedof               : Number of degrees of freedom per element (14)
!     size_force          : Dimension of the global force vector
!     size_mass           : Dimension of the global mass matrix
!     size_stiff          : Dimension of the global stiffness matrix
!     elmcon              : Element connectivity array
!     nodcon              : Node connectivity array
```

```
!     intoun              : Internal node to user node numbering array
!     ietoue              : Internal element to user element numbering array
!     cord                : Initial nodal co-ordinates
!     displacement        : Nodal displacements at previous iteration
!     velocity            : Nodal velocities at previous iteration
!     acceleration        : Nodal accelerations at previous iteration
!     trgb                : Rigid body rotation (local undeformed -> convected)
transformation matrix
!     tglu                : Global to local undeformed transformation matrix
!     disp_prev           : Nodal displacements at previous timestep
!     pos_prev            : Nodal positions at previous timestep
!     vel_prev            : Nodal velocities at previous timestep
!     acc_prev            : Nodal accelerations at previous timestep
!     axial_force         : Axial force in elements at previous timestep
!     y_shear             : Y Shear forces in elements at previous timestep
!     z_shear             : Z Shear forces in elements at previous timestep
!     torque              : Torque in elements at previous timestep
!     y_bending           : Y bending moments in elements at previous timestep
!     z_bending           : Z bending moments in elements at previous timestep
!     eff_tension         : Effective Tension in elements at previous timestep
!     y_curvature         : Y curvatures in elements at previous timestep
!     z_curvature         : Z curvatures in elements at previous timestep
!     force               : Global force vector at previous iteration
!     mass                : Global mass matrix
!     stiff               : Global stiffness matrix

! Declare local variables.
  integer i,j, index, index1, index2
  integer :: int_elem_no, int_node_no
  integer :: node_on_elem_index
  integer :: int_DOF
  integer :: user_elem_no, user_node_no, user_dof
  real(8), dimension(3) :: user_applied_force
  real(8), dimension(3) :: global_user_applied_force
  real(8),dimension(3,3) :: trgb_int_elem
  real(8),dimension(3,3) :: tglu_int_elem
  real(8),dimension(3,3) :: transformation_matrix
  real(8),dimension(3,3) :: transpose_matrix
  real(8), dimension(size_force,nedof) :: force_array
  logical :: is_elem_OK, is_node_OK, is_node_on_element

  ! Follower Force Test

  ! User data
  user_elem_no = 10
  user_node_no = 11
  ! Local axes applied force
  user_applied_force(1:3) = [0.0d0, 0.0d0, 1000.0d0] ! Force applied in DOF3

  ! Internal works
  int_elem_no = 0
  int_node_no = 0
  transformation_matrix = 0.0d0
  is_elem_OK = .false.
  is_node_OK = .false.
  is_node_on_element = .false.

  ! Check if user element exists in Flexcom list of elements and set internal
element number
```

```
  ! if element exists.
  do i = 1,nelmn
    if( user_elem_no == ietoue(i) )then
      int_elem_no = i
      is_elem_OK = .true.
    end if
  end do

  ! Check if user node exists in Flexcom list of elements and set internal node
number
  ! if node exists.
  do i = 1, nnode
    if( user_node_no == intoun(i) )then
      int_node_no = i
      is_node_OK = .true.
    end if
  end do

  ! Check if internal node is on internal element
  do i=1,nncon
    if(int_node_no  == elmcon(i,int_elem_no))then
      is_node_on_element = .true.
      exit
    end if
  end do
  if( is_node_on_element == .false.) then
    write(*,*) "Node ", user_node_no, "is not included on any elements."
  end if

  ! Apply user force at internal node number and internal DOF
  if( is_elem_OK .and. is_node_OK .and. is_node_on_element ) then

    !Extract transformation matrices at internal element number
    trgb_int_elem(1:3,1:3) = trgb(1:3,1:3, int_elem_no)
    tglu_int_elem(1:3,1:3) = tglu(1:3,1:3, int_elem_no)

    ! Perform transformation from local to global axes
    transformation_matrix(1:3,1:3) =
matmul(trgb_int_elem(1:3,1:3),tglu_int_elem(1:3,1:3))

    ! Transformation matrix transpose
    transpose_matrix = transpose(transformation_matrix)

    ! Global user applied force
    global_user_applied_force(1:3) =
matmul(transpose_matrix(1:3,1:3),user_applied_force(1:3))

    ! Set node on element index
    node_on_elem_index = 0
    if ( int_node_no == elmcon(2,int_elem_no) )node_on_elem_index = 6

    ! Augment the force with global user applied force.
    ! DOF 1-3 for Start Node, DOF 7-9 for End Node
    do i = 1,3
      int_DOF = i + node_on_elem_index
      force(int_elem_no,int_DOF) = force(int_elem_no,int_DOF) + ramp *
global_user_applied_force(i)
    end do
```

```
    end if

  ! Do not alter the next line.
  end subroutine user_solver_variables
```

Source Code

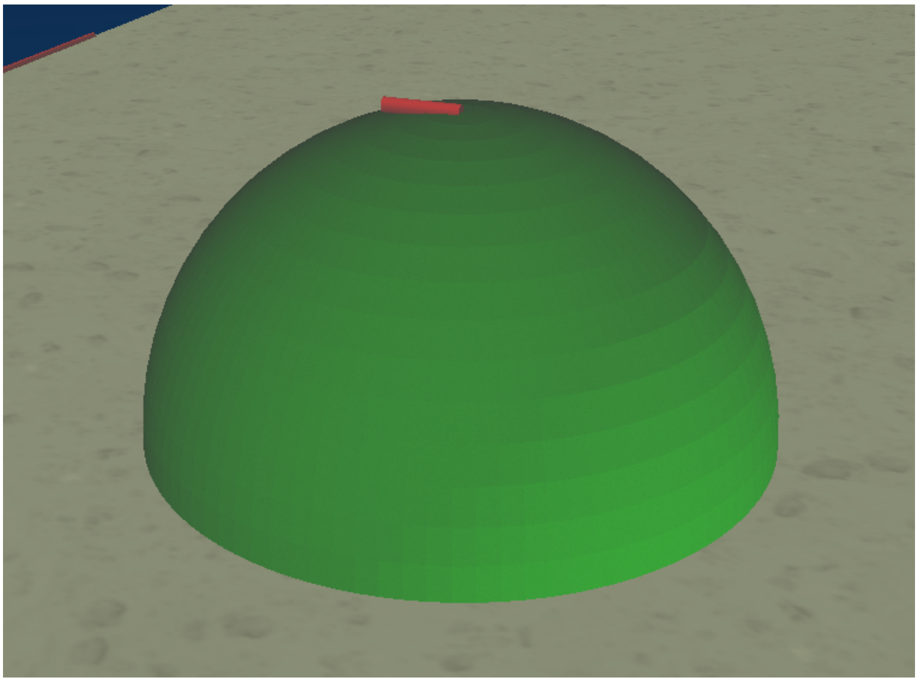
J04b - Sphere Contact

INTRODUCTION

This example illustrates how you can model an arbitrary contact surface via the User Solver Variables feature. A sphere is used in this example for simplicity but the feature is fully customisable, the use of custom code enables direct modification of the global stiffness matrix and global force vector.

MODEL

The model consists of a simple horizontal beam which is initially constrained above a spherical contact surface. For visual purposes, a spherical shape is added as an Auxiliary Body, but this does not affect the numerical simulation. During a restart stage, the beam is allowed to fall freely and make contact with the spherical body below it. In order to verify correct model behaviour, results are compared with an analogous model which uses a Cylindrical Guide Surface.

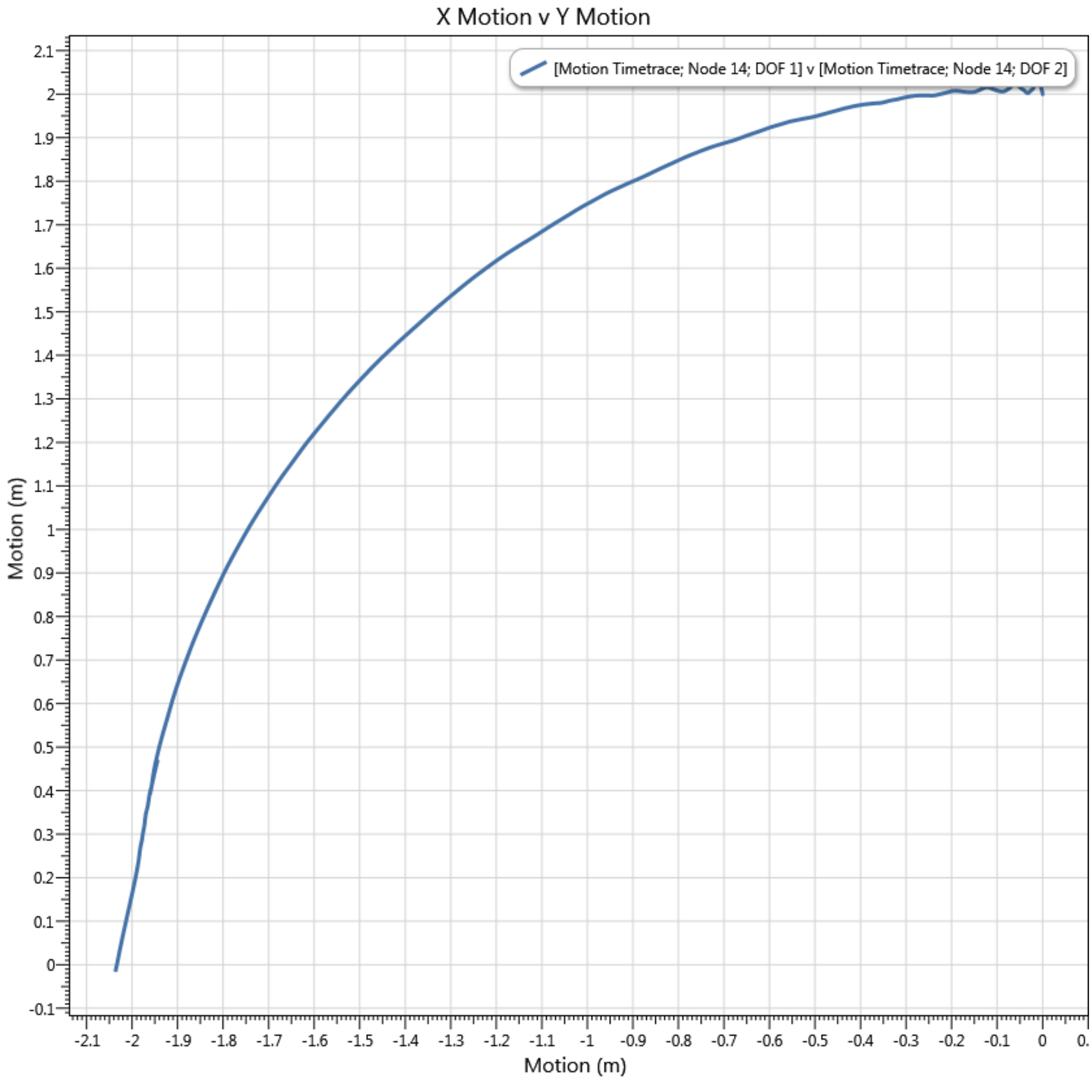


Sphere Contact Model

The positions of the finite element nodes are monitored for contact with the sphere. Once contact occurs, a local stiffness term is applied in a direction which is perpendicular to the sphere at the contact location. A force vector is also computed to account for the initial separation between the node and the contact surface (because Flexcom solves for position rather than displacement). The custom code uses appropriate local to global transformation matrices to convert the local components into the global axis system before augmenting the global constituent terms. The custom code is effectively replicating (a simplified version of) Flexcom's guide surface contact modelling algorithm, but using a different geometrical pattern to model the sphere.

RESULTS

The beam slides down along the side of the sphere as expected. Results show close correlation with the analogous cylindrical surface model.



Motion of Beam End Point

SOURCE CODE

Pre-compiled DLLs for both 32-bit and 64-bit operating systems are provided with this example. The original Fortran source code is also provided should you wish to examine it.

```
subroutine
user_solver_variables(iter,time,ramp,nnode,nncon,ncord,nndof,nelmn,necon,nedof,
&
  size_force,size_mass,size_stiff,elmcon,nodcon,intoun,ietoue,cord,displacement,
           &
  velocity,acceleration,trgb,tglu,disp_prev,pos_prev,vel_prev,acc_prev,axial_force,y
_shear,    &
  z_shear,torque,y_bending,z_bending,eff_tension,y_curvature,z_curvature,force,mass,
stiff)
!dec$ attributes dllexport, stdcall, reference :: user_solver_variables
  implicit none
```

```
! Input variables - cannot be modified within this subroutine

  integer, intent(in)    :: iter
  !dec$ attributes value :: iter
  real(8), intent(in)    :: time
  !dec$ attributes value :: time
  real(8), intent(in)    :: ramp
  !dec$ attributes value :: ramp
  integer(4), intent(in) :: nnode
  !dec$ attributes value :: nnode
  integer(4), intent(in) :: nncon
  !dec$ attributes value :: nncon
  integer(4), intent(in) :: ncord
  !dec$ attributes value :: ncord
  integer(4), intent(in) :: nndof
  !dec$ attributes value :: nndof

  integer(4), intent(in) :: nelmn
  !dec$ attributes value :: nelmn
  integer(4), intent(in) :: necon
  !dec$ attributes value :: necon
  integer(4), intent(in) :: nedof
  !dec$ attributes value :: nedof

  integer(4), intent(in) :: size_force
  !dec$ attributes value :: size_force
  integer(4), intent(in) :: size_mass
  !dec$ attributes value :: size_mass
  integer(4), intent(in) :: size_stiff
  !dec$ attributes value :: size_stiff

  integer(4), intent(in), dimension(nncon, nelmn)              :: elmcon
  !dec$ attributes reference :: elmcon
  integer(4), intent(in), dimension(necon,nnode)               :: nodcon
  !dec$ attributes reference :: nodcon
  integer(4), intent(in), dimension(nnode)                     :: intoun
  !dec$ attributes reference :: intoun
  integer(4), intent(in), dimension(nelmn)                     :: ietoue
  !dec$ attributes reference :: ietoue
  real(8),  intent(in), dimension(ncord,nnode)                 :: cord
  !dec$ attributes reference :: cord
  real(8),  intent(in), dimension(nndof,nnode)                 :: displacement
  !dec$ attributes reference :: displacement
  real(8),  intent(in), dimension(nndof,nnode)                 :: velocity
  !dec$ attributes reference :: velocity
  real(8),  intent(in), dimension(nndof,nnode)                 :: acceleration
  !dec$ attributes reference :: acceleration
  real(8),  intent(in), dimension(3,3,nelmn)                   :: trgb
  !dec$ attributes reference :: trgb
  real(8),  intent(in), dimension(3,3,nelmn)                   :: tglu
  !dec$ attributes reference :: tglu
  real(8),  intent(in), dimension(nndof,nnode)                 :: disp_prev
  !dec$ attributes reference :: disp_prev
  real(8),  intent(in), dimension(nndof,nnode)                 :: pos_prev
  !dec$ attributes reference :: pos_prev
  real(8),  intent(in), dimension(nndof,nnode)                 :: vel_prev
  !dec$ attributes reference :: vel_prev
  real(8),  intent(in), dimension(nndof,nnode)                 :: acc_prev
```

```
  !dec$ attributes reference :: acc_prev
  real(8),  intent(in), dimension(nelmn,3)                 :: axial_force
  !dec$ attributes reference :: axial_force
  real(8),  intent(in), dimension(nelmn,3)                 :: y_shear
  !dec$ attributes reference :: y_shear
  real(8),  intent(in), dimension(nelmn,3)                 :: z_shear
  !dec$ attributes reference :: z_shear
  real(8),  intent(in), dimension(nelmn,3)                 :: torque
  !dec$ attributes reference :: torque
  real(8),  intent(in), dimension(nelmn,3)                 :: y_bending
  !dec$ attributes reference :: y_bending
  real(8),  intent(in), dimension(nelmn,3)                 :: z_bending
  !dec$ attributes reference :: z_bending
  real(8),  intent(in), dimension(nelmn,3)                 :: eff_tension
  !dec$ attributes reference :: eff_tension
  real(8),  intent(in), dimension(nelmn,3)                 :: y_curvature
  !dec$ attributes reference :: y_curvature
  real(8),  intent(in), dimension(nelmn,3)                 :: z_curvature
  !dec$ attributes reference :: z_curvature

! Output variables - can be modified within this subroutine if required
  real(8),  intent(inout), dimension(size_force,nedof)       :: force
  !dec$ attributes reference :: force
  real(8),  intent(inout), dimension(nedof,nedof,size_mass)   :: mass
  !dec$ attributes reference :: mass
  real(8),  intent(inout), dimension(nedof,nedof,size_stiff)  :: stiff
  !dec$ attributes reference :: stiff

! Variable names
!     iter                 : Current iteration
!     time                 : Current timestep
!     ramp                 : Ramp
!     nnode                : Number of nodes in the model
!     nncon                : Number of nodes with connected elements
!     ncord                : Number of coordinates
!     nndof                : Number of degrees of freedom per node (6)
!     nelmn                : Number of elements in the model
!     necon                : Number of elements connected
!     nedof                : Number of degrees of freedom per element (14)
!     size_force           : Dimension of the global force vector
!     size_mass            : Dimension of the global mass matrix
!     size_stiff           : Dimension of the global stiffness matrix
!     elmcon               : Element connectivity array
!     nodcon               : Node connectivity array
!     intoun               : Internal node to user node numbering array
!     ietoue               : Internal element to user element numbering array
!     cord                 : Initial nodal co-ordinates
!     displacement         : Nodal displacements at previous iteration
!     velocity             : Nodal velocities at previous iteration
!     acceleration         : Nodal accelerations at previous iteration
!     trgb                 : Rigid body rotation (local undeformed -> convected)
transformation matrix
!     tglu                 : Global to local undeformed transformation matrix
!     disp_prev            : Nodal displacements at previous timestep
!     pos_prev             : Nodal positions at previous timestep
!     vel_prev             : Nodal velocities at previous timestep
!     acc_prev             : Nodal accelerations at previous timestep
!     axial_force          : Axial force in elements at previous timestep
!     y_shear              : Y Shear forces in elements at previous timestep
```

```
!     z_shear            : Z Shear forces in elements at previous timestep
!     torque             : Torque in elements at previous timestep
!     y_bending          : Y bending moments in elements at previous timestep
!     z_bending          : Z bending moments in elements at previous timestep
!     eff_tension        : Effective Tension in elements at previous timestep
!     y_curvature        : Y curvatures in elements at previous timestep
!     z_curvature        : Z curvatures in elements at previous timestep
!     force              : Global force vector at previous iteration
!     mass               : Global mass matrix
!     stiff              : Global stiffness matrix

  ! Declare local variables.
  integer :: i,j
  integer :: int_elem_no
  integer :: int_node_no
  integer :: local_node
  integer :: user_number_of_nodes
  integer :: allocate_error
  integer :: index
  integer, allocatable, dimension(:) :: user_node_no_array
  real(8) :: user_sphere_radius
  real(8) :: user_sphere_stiffness
  real(8) :: distance
  real(8) :: penetration
  real(8) :: alpha
  real(8) :: magnitude_local_unit_vector
  real(8) :: pi
  real(8) :: dot_prod
  real(8) :: orthogonal_magnitude
  real(8), dimension(3) :: user_sphere_origin
  real(8), dimension(3) :: node_co_ordinates
  real(8), dimension(3) :: global_contact_force_array
  real(8), dimension(3) :: local_unit_vector
  real(8), dimension(3) :: orthogonal_local_unit_vector
  real(8), dimension(3) :: position_array
  real(8), dimension(3) :: global_co_ordinates_unit_vector
  real(8), dimension(3,3) :: rotation_matrix
  real(8), dimension(3,3) :: transpose_rotation_matrix
  real(8), dimension(3,3) :: local_stiffness_matrix
  real(8), dimension(3,3) :: global_stiffness_matrix
  real(8), dimension(3,3) :: interm_stiffness_matrix
  real(8), dimension(nedof,nedof) :: elem_stiff
  logical :: is_node_ok
  logical :: start_node
  logical :: end_node

  ! Sphere Contact Test

  ! User data
  user_number_of_nodes = 2
  user_sphere_radius = 2.0d0
  user_sphere_stiffness = 1.0d3
  user_sphere_origin = [0.0d0, 0.0d0, 0.0d0]

  ! Allocate array of node numbers for which calculations are performed.
  do i = 1, 1
    allocate(user_node_no_array(user_number_of_nodes), stat=allocate_error )
    if( allocate_error /= 0 )exit
    user_node_no_array = [1,2]
```

```
end do

! Internal works

! Pi
pi = 4.0d0 * datan(1.0d0)

do index = 1, user_number_of_nodes
  int_elem_no = 0
  int_node_no = 0
  local_node  = 0
  is_node_ok = .false.
  start_node = .false.
  end_node = .false.
  alpha = 0.0d0
  distance = 0.0d0
  penetration = 0.0d0
  orthogonal_magnitude = 0.0d0
  dot_prod                         = 0.0d0
  magnitude_local_unit_vector      = 0.0d0
  node_co_ordinates                = 0.0d0
  local_unit_vector                = 0.0d0
  global_co_ordinates_unit_vector = 0.0d0
  orthogonal_local_unit_vector     = 0.0d0
  global_contact_force_array       = 0.0d0
  local_stiffness_matrix           = 0.0d0
  interm_stiffness_matrix          = 0.0d0
  global_stiffness_matrix          = 0.0d0
  rotation_matrix                  = 0.0d0
  transpose_rotation_matrix        = 0.0d0

  ! Check if user element exists in Flexcom list of elements and set internal
element number
  ! if element exists.
  do i = 1,nnode
    if( user_node_no_array(index) == intoun(i) )then
      int_node_no = i
      is_node_ok = .true.
    end if
  end do

  ! Start calculations
  if( is_node_ok ) then

    ! Get instantaneous co-ordinates of Local Node
    do i = 1,3
      node_co_ordinates(i) = cord(i,int_node_no) + displacement(i,int_node_no)
    end do

    ! Distance from origin of the sphere to End Node
    distance = norm2( node_co_ordinates - user_sphere_origin )

    ! Penetration
    penetration = user_sphere_radius - distance

    ! Proceed calculations for value of penetration greater than zero
    if ( penetration > 1.0d-10 )then
```

```
        ! Fill in local stiffness matrix
        local_stiffness_matrix(1,1) = user_sphere_stiffness

        ! Unit vector of the local axis system which gives the direction of contact
force
        local_unit_vector = ( node_co_ordinates - user_sphere_origin ) /
( distance )

        ! Global co-ordinates vector (global X axis)
        global_co_ordinates_unit_vector = [1.0d0, 0.0d0, 0.0d0]

        ! Vector orthogonal to the global X axis and local_unit_vector
        orthogonal_local_unit_vector(1) = global_co_ordinates_unit_vector(2)
*local_unit_vector(3) - global_co_ordinates_unit_vector(3)*local_unit_vector(2)
        orthogonal_local_unit_vector(2) = global_co_ordinates_unit_vector(3)
*local_unit_vector(1) - global_co_ordinates_unit_vector(1)*local_unit_vector(3)
        orthogonal_local_unit_vector(3) = global_co_ordinates_unit_vector(1)
*local_unit_vector(2) - global_co_ordinates_unit_vector(2)*local_unit_vector(1)

        orthogonal_magnitude = norm2( orthogonal_local_unit_vector )
        orthogonal_local_unit_vector = orthogonal_local_unit_vector /
orthogonal_magnitude

        ! Magnitude of Local unit vector
        magnitude_local_unit_vector = dsqrt( local_unit_vector(1)**2 +
local_unit_vector(2)**2 + local_unit_vector(3)**2 )

        ! Angle between global X axis and local_unit_vector
        dot_prod = global_co_ordinates_unit_vector(1)*local_unit_vector(1) +
global_co_ordinates_unit_vector(2)*local_unit_vector(2)
+global_co_ordinates_unit_vector(3)*local_unit_vector(3)

        if ( magnitude_local_unit_vector > 1.0d-10 )then
          alpha = dacos( dot_prod / magnitude_local_unit_vector )
        else
          alpha = 0.0d0
        end if

        ! Normalise angle to interval [-pi,pi]
        if( alpha < - pi )then
          do while (alpha < - pi)
            alpha = alpha + (2.d0 * pi)
          end do
        else if( alpha > pi )then
          do while (alpha > pi)
            alpha = alpha - (2.d0 * pi)
          end do
        end if

        ! get rotation matrix and transpose of roation matrix
        call calculate_rotation_matrix(alpha, orthogonal_local_unit_vector,
rotation_matrix, transpose_rotation_matrix)

        ! Get internal element number
        int_elem_no = nodcon(1,int_node_no)

        ! Set Local Node
        if ( int_node_no == elmcon(1,int_elem_no) )then
```

```
          local_node = 1
        else
          local_node = 2
        end if

        ! Premultiply local stiffness matrix with transpose of rotation matrix
        interm_stiffness_matrix = matmul( transpose_rotation_matrix,
local_stiffness_matrix )
        ! Get global stiffness matrix - postmultiply interm_stiffness_matrix with
rotation matrix
        global_stiffness_matrix =  matmul( interm_stiffness_matrix,
rotation_matrix )

        ! Store element stiffness matrix
        elem_stiff(:,:) = 0.0d0
        if ( local_node == 1 )then
          elem_stiff(1:3,1:3) = global_stiffness_matrix(:,:)
        else
          elem_stiff(7:9,7:9) = global_stiffness_matrix(:,:)
        end if

        ! Augment the global stiffness matrix
        stiff(:,:,int_elem_no) = stiff(:,:,int_elem_no) + elem_stiff(:,:)

        ! Position array
        position_array = [ user_sphere_radius, 0.0d0, 0.0d0]


        ! Premultiply local position array with transpose of rotation matrix
        position_array = matmul(transpose_rotation_matrix,position_array) +
user_sphere_origin

        ! Get global nodal load
        global_contact_force_array = matmul(global_stiffness_matrix,position_array)

        ! Augment the force with global contact force.
        ! DOF 1-3 for Start Node, DOF 7-9 for End Node
        if ( local_node == 1 )then
          force(int_elem_no,1:3) = force(int_elem_no,1:3) +
global_contact_force_array(:)
        else
          force(int_elem_no,7:9) = force(int_elem_no,7:9) +
global_contact_force_array(:)
        end if

      end if

    end if

  end do

  ! Deallocate array
  if(allocated(user_node_no_array)) deallocate(user_node_no_array)

! Do not alter the next line.
end subroutine user_solver_variables
```

```
!
********************************************************************************
***
!
********************************************************************************
***
!
********************************************************************************
***
!
********************************************************************************
***


! Subroutine to calculate rotation matrix for an array rotated by an angle alpha
about
! an axis defined by an orthogonal_local_unit_vector. Transpose of the rotation
matrix is
! calculated as well.
subroutine calculate_rotation_matrix(angle, vector, rotation_matrix,
transpose_rotation_matrix)
  implicit none

  ! Declare arguments
  real(8), intent(in) :: angle
  real(8), intent(in), dimension(3) :: vector
  real(8), intent(inout), dimension(3,3) :: rotation_matrix
  real(8), intent(inout), dimension(3,3) :: transpose_rotation_matrix

  ! Rotation matrix
  rotation_matrix(1,1) = dcos(angle) + vector(1) * vector(1) * ( 1 - dcos(angle) )
  rotation_matrix(2,1) = vector(1) * vector(2) * ( 1 - dcos(angle) ) - vector(3) *
dsin(angle)
  rotation_matrix(3,1) = vector(1) * vector(3) * ( 1 - dcos(angle) ) + vector(2) *
dsin(angle)

  rotation_matrix(1,2) = vector(2) * vector(1) * ( 1 - dcos(angle) ) + vector(3) *
dsin(angle)
  rotation_matrix(2,2) = dcos(angle) + vector(2)* vector(2) * ( 1 - dcos(angle) )
  rotation_matrix(3,2) = vector(2) * vector(3) * ( 1 - dcos(angle) ) - vector(1) *
dsin(angle)

  rotation_matrix(1,3) = vector(3) * vector(1) * ( 1 - dcos(angle) ) - vector(2) *
dsin(angle)
  rotation_matrix(2,3) = vector(3) * vector(2) * ( 1 - dcos(angle) ) + vector(1) *
dsin(angle)
  rotation_matrix(3,3) = dcos(angle) + vector(3)* vector(3) * ( 1  - dcos(angle) )

  ! Transpose rotation matrix
  transpose_rotation_matrix = transpose(rotation_matrix)

end subroutine calculate_rotation_matrix
```

Source Code

Mass Matrix Output

OVERVIEW

Although not part of a formal example, this section provides some sample source code which enables you to write the global mass matrix to the output file. If you would like to try it out...

- Create a new project workspace in your developer studio environment. If you are using a recognised IDE (integrated development environment), look for a DLL project template which you can begin working from. This will typically appear under New->Project->Templates->Library or something similar.
- Create a new free form file (with .f90 file extension) and add it to the project workspace.
- Copy the standard template for the user_solver_variables subroutine and insert it into the file.
- Copy the sample source code below and insert it into the subroutine at the appropriate location.
- Compile the DLL and link it to your Flexcom simulation via the `*USER SOLVER VARIABLES` keyword. Ensure that the bitness of your DLL matches that of your Flexcom installation - i.e. if you have a 64-bit version of Flexcom, you should compile a 64-bit DLL.

SOURCE CODE

```
! ------------------------------------------
! Sample code for writing out global matrices
! ------------------------------------------

! Declare local variables.
integer :: iout, ielem, user_elem_no, irow, icol

! Initialise local variables
iout = 31

! Write out header block
write(iout,fmt="(18x,a,/,a,18x,a)")&
 "*** GLOBAL MASS MATRIX ***", &
 "---------------------------"

! Write out current time and current iteration
write(iout,fmt="(/,2x,a,f9.3,a,/,2x,a,i0,/)")"Time: ",time,"s", "Iteration No: ",iter

! Loop over elements
element_loop: do ielem = 1,nelmn
```

```
    ! Write out element number
    user_elem_no = ietoue(ielem)
    write(iout,fmt="(/,2x,a,i0,/)")"Element No: ",user_elem_no

    ! Loop over the rows
    row_loop: do irow = 1,14
      write(iout,fmt="(14(2x,e11.4))")(mass(irow,icol,ielem),icol=1,14)
    end do row_loop
  end do element_loop
```

1.10.11 K - Software Tutorials

Section K provides some worked examples which serve as tutorials in the use of Flexcom, taking you step-by-step through sample model creation and simulation. They provide a practical overview of how to use the software, and ideally you should work through these examples completely before you start working with Flexcom.

Flexcom is accompanied by two separate worked examples as follows.

- K01 - Worked Example - Simple. This simple example considers a flexible riser in steep wave configuration. This is a very straightforward example, and illustrates basic model building principles. Once the structural configuration has been obtained statically, regular wave loading is applied to the system. This example introduces you to some fundamental concepts such as Lines, Geometric Properties, Hydrodynamic Properties, Vessels and Vessel Motions, Boundary Conditions, Static Analysis, Wave Loading, Time Domain Analysis and Database Postprocessing. Rather than following the written documentation, you may prefer to watch the Tutorial Video on YouTube.

- K02 - Worked Example - Complex. The complex example considers a free standing hybrid riser arrangement. This is a much more complex model, which includes a dual bore vertical rigid steel riser tensioned via a buoyancy tank, with a connection to an FPSO through two flexible jumpers. Once the structural configuration has been obtained statically, a load case matrix of various vessel offsets and wave periods are considered. In addition to the fundamental concepts introduced in the first example, this example also includes some more advanced topics, such as Pipe-in-Pipe, Keyword Parameterisation, Summary Postprocessing and Summary Postprocessing Collation, Code Checking and Video Creation. Rather than following the written documentation, you may prefer to watch the Tutorial Video on YouTube.

To derive maximum benefit from the worked examples, it is recommended that you actually sit at a computer and work through all of the instructions, rather than just reading through the documentation or watching the videos. Naturally, this will help you to remember the contents of the examples and should result in a more thorough understanding of Flexcom.

1.10.11.1 K01 - Worked Example - Simple

This worked example takes you through all of the stages in performing an analysis with Flexcom. It begins by describing the structure under consideration, and then details the steps in creating the finite element model. This worked example considers a flexible riser in steep wave configuration. It is a very straightforward example, and illustrates basic model building principles.

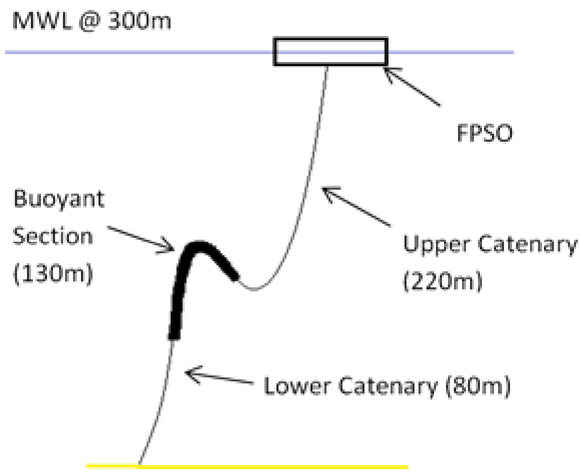
The example is now presented using the following sub-sections:

- Riser System summarises the flexible riser in steep wave configuration which forms the basis of the worked example.
- Project Structure introduces the concept of a highly project-focused integrated engineering environment.
- Model Building describes the model building process, focusing particularly on the lines feature, which essentially provides an automatic mesh creation facility to expedite the model creation process.
- Structural and Hydrodynamic Properties outlines the assignment of structural and hydrodynamic properties to the model.
- Environment outlines the ocean and seabed properties
- Vessel and RAO Data outlines the vessel definition and the application of vessel RAOs.
- Initial Static Analysis describes how to perform the initial static analysis, to extract results via Database Postprocessing, and to examine graphical output using the Plotting facility.
- Dynamic Analysis describes how to perform restart regular wave analyses.

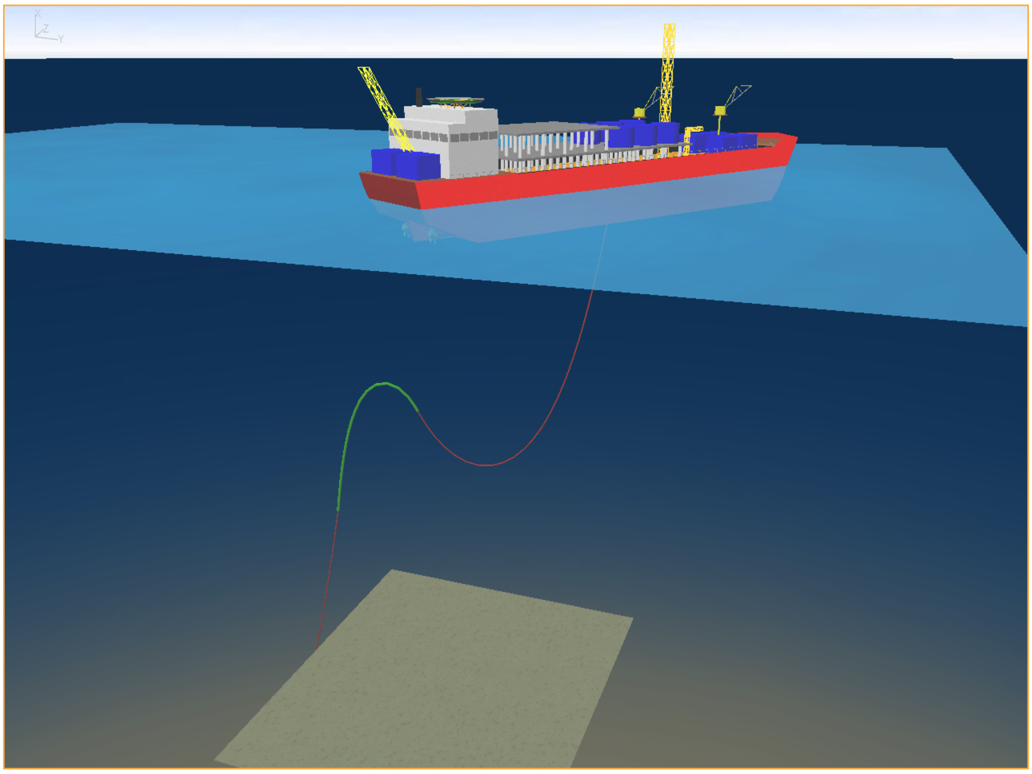
Riser System

This example considers the analysis of a flexible riser in steep wave configuration. The steep wave riser configuration is shown below. The riser in this example is 290m in length and is sited in 200m of water. It is filled with oil and attached to a production vessel.

The analysis is performed in two stages. The initial static analysis applies gravity and buoyancy forces. The dynamic analysis finds the response of the structure to vessel motions and wave loading.



Model Schematic



Riser in Model View

Project Structure

.If you are unfamiliar with the basic components of Flexcom's user interface, please refer to the Interface section first. In particular, this example assumes some level of familiarity with the Keyword Editor, Table Editor and Model View components.

In order to create a new project, select New Project from the file menu on the top menu bar. You will then be presented with options for defining a project name and a project folder. You can specify any project name you like – indeed the same is true for any of the keyword files which will be described in the following sections. It is advised that you use a project folder other than the Worked example folder in the Flexcom installation directory. This will ensure that you do not overwrite the original files provided with the program installation, and will allow you to use these files as a reference to verify your own files should you experience any problems running the example.

Once the project is created, you need to add the first keyword file to the project. To add a keyword file, select New File from the file menu on the top menu bar. You will be presented with options to select a template and name for the file.Select a blank Keyword file which uses the metric unit system and enter the name *K1-SWR-Static.keyxm* (or similar, depending on your naming convention) and create an empty keyword file. The file should open in the main document panel. You should also see the file added to the project in the file view.

Model Building

OVERVIEW OF LINES FEATURE

The lines functionality is a relatively recent addition to Flexcom, and essentially provides an automatic mesh creation facility whereby nodes, elements and cables can be generated quite easily. Using lines to build your model is a fundamentally different approach to working directly with nodes, elements and cables. While all the information is ultimately handled in the same fashion internally in the software, the use of lines expedites the model creation process considerably, as you need not concern yourself with individual node and element numbering. While the explicit numbering scheme (which was the only approach available in earlier versions of Flexcom) may appear obsolete by comparison, it is retained for complete generality, and also to maintain downward compatibility with previous versions. Refer to Lines for further information on this feature.

RISER CONSTRUCTION

The Riser, including the Lower Catenary, Buoyant Section and Upper Catenary are modelled using a single line running all the way from the seabed to the base of the ship.

The line is called Riser, and is assigned a length of 430m. Its start and end locations are called Lower End {0m, 0m} and Upper End {210m, 430m}. Obviously the structural properties of the structure vary along its length due to the different components, so it is necessary to define a number of subsections in terms of distances along the line. The subsections are called Lower Catenary (from 0m to 80m), Buoyant Section (from 80m to 210m) and Upper Catenary (from 210m to 430m). The definition of the Riser line in both table and keyword editor is shown in below.

Input data is organised into two main categories, model data and load case data. This distinction is reflected in both the table and keyword editors. Model data consists of information which must be specified at the start of a series of consecutive analyses. This data is carried through to all subsequent restart analyses and may not be changed. This category naturally includes the finite element discretisation, structural and hydrodynamic properties, plus any other inputs which characterise the initial model configuration (e.g. initial vessel position, seabed properties, ocean depth etc.). Load case data may be entered in any analysis of a series of runs, and which may be subsequently altered in restart analyses. This category naturally includes environmental loading such as current and waves, boundary conditions, solution variables etc. There is a clear division between model and load case data in the keyword editor, via the $MODEL and $LOAD CASE sections in the keyword file. Similarly, in the table editor, the sections are split onto different tabs named MODEL and LOAD CASE. Before you enter the first keyword (i.e. *LINES, to define the central structure), you should include a $MODEL section entry. You can also add a MODEL section through the table editor by right-clicking on the table editor and selecting Add Section from the context menu.

Line Sections

Line Name	Section Set Name	Start Distance	End Distance	Min. Element Length	Max. Element Length	Additional Sets for this Section
Riser	Lower Catenary	0	80			
	Buoyant Section	80	210			
	Upper Catenary	210	430			

Lines

Line Name	Non-Section Set Name	Length	Start Node Label	Start (X)	Start (Y)	Start (Z)	End Node Label	End (X)	End (Y)	End (Z)	Min. Element Length
Riser		430	Lower End	0	0	[0.0]	Upper End	290	120	[0.0]	1

```
*LINES
 LINE=Riser, 430
 START=Lower End, 0, 0
 END=Upper End, 290, 120
 1, 5
 SECTION=Lower Catenary,    0,   80
 SECTION=Buoyant Section,  80,  210
 SECTION=Upper Catenary,  210,  430
```

Central Structure Line

Refer to *LINES for further information on these data inputs.

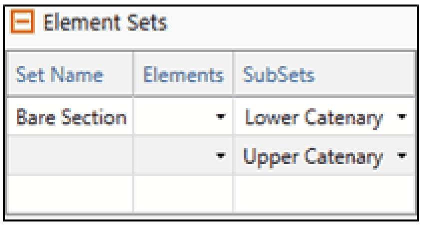
ALTERNATIVE MODEL CONSTRUCTION

The above construction method is fairly standard, where a smeared approach is adopted in terms of modelling riser buoyancy. In reality the central section of the riser is likely to be supported by a series of discrete buoyancy modules, rather than a continuous section of buoyancy foam. Modelling discrete buoyancy modules requires a little more effort, but assistance is provided by the `*LINE SECTION GROUPS` keyword which facilitates the creation of repeating sub-sections. A separate static input file is included in the example folder to illustrate this modelling approach. Although not used subsequently in terms of the dynamic simulation in this particular example, you may be interested to learn about this alternative modelling approach.

In this case, three separate lines are used to model the Lower Catenary, Middle Catenary and Upper Catenary. A line section group is assembled, comprising of a section of bare riser, a buoyancy module, and another bare riser section. The Middle Catenary line is then assembled using repeated occurrences of the line section group. Nodal equivalences are used to connect the lines together at the intersection points.

ELEMENT SETS

Flexcom uses the concept of element sets to define the physical properties of the finite element model. Groups of similar elements are logically combined into named element sets, properties are then assigned on an element set-by-set basis. When lines are utilised during model creation, relevant element sets are created automatically, based on the names of each line (and subsection of each line if appropriate) in the model.As the Lower Catenary and Upper Catenary have identical structural properties, a single element set called Bare Section is created to group both of these sections together.



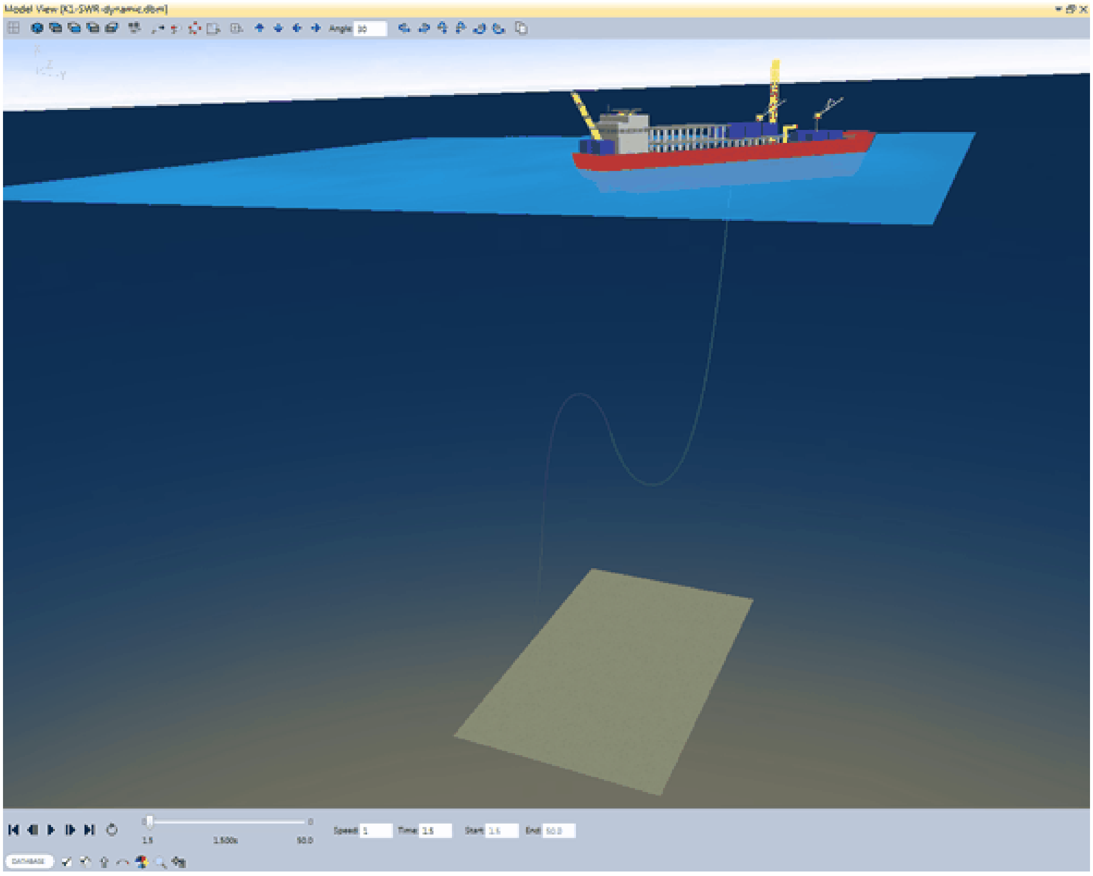

```
*ELEMENT SETS
 SET=Bare Section
 SUBSET=Lower Catenary
 SUBSET=Upper Catenary
```

Element sets

Refer to *ELEMENT SETS for further information on these data inputs.

MODEL VIEW

The *Model View* provides a “live” structure preview facility available during model building. It also allows you to view an animation of the structure response after an analysis has completed. The *Model View* allows you to rotate and pan the viewpoint, to zoom in and out, and has many other useful display features (feel free to experiment) such as node and element numbering, nodal coordinates, seabed topography, water surface profile etc. Once the geometric specification has been completed, the *Model View* provides a preview of the steep wave riser, as shown in the figure below. Refer to Model View for further information on this feature.



Model View

Structural and Hydronamic Properties

STRUCTURAL PROPERTIES

Flexcom provides two different formats for specifying structure geometric properties, although ultimately the information is the same and is used in the same way by the program. These are termed the *Flexible Riser* format and the *Rigid Riser* format. The *Flexible Riser* format is so called, naturally, because it is the format commonly used when defining flexible risers. Specifically, the data to be specified in this case comprises of bending stiffness about two axes, torsional stiffness, axial stiffness, mass per unit length etc. When you use the *Rigid Riser* format on the other hand, you input internal and external diameters, Young's modulus, shear modulus, mass density etc. – data for the analysis of a *Rigid Riser* (whether a top tensioned riser or an SCR) would normally be available in this format. Refer to Geometric Properties in Flexible Riser Format for further information on this feature.

In this example the steep wave riser's structural properties are specified in the Flexible Riser format, as shown below.

Flexible Riser Geometric Properties

Set Name	Bending Stiffness	EIyy	EIzz	GJ	EA	m	p	Di	Dd	Db
Bare Section	LINEAR	30000	30000	1.0E+06	5.0E+08	80	[0.0]	0.1	0.25	0.25
Buoyant Section	LINEAR	30000	30000	1.0E+06	5.0E+08	160	[0.0]	0.1	0.52	0.52

```
*GEOMETRIC SETS
 OPTION=FLEXIBLE
 SET=Bare Section
 30000, 30000, 1.0E+06, 5.0E+08, 80, , 0.1, 0.25, 0.25
 OPTION=FLEXIBLE
 SET=Buoyant Section
 30000, 30000, 1.0E+06, 5.0E+08, 160, , 0.1, 0.52, 0.52
```

Structural Properties – Flexible riser

Refer to `*GEOMETRIC SETS` for further information on these data inputs.

HYDRODYNAMIC PROPERTIES

The simplest hydrodynamic property specification in Flexcom is that of constant coefficients, which remain the same throughout an analysis. Five hydrodynamic coefficients are specified on an element set by set basis, to model drag, added mass and inertia loading. For simplicity, all elements in this example are assigned the same hydrodynamic coefficients as shown in the figure below. Refer to Constant Hydrodynamic Coefficients for further information on this feature.

Hydrodynamic Properties

Set Name	Normal Drag	Tangential Drag	Normal Inertia	Tangential Added Mass
All	1	0	2	0

```
*HYDRODYNAMIC SETS
 SET=All, TYPE=CONSTANT
 1, 0, 2, 0
```

Hydrodynamic Properties

Refer to `*HYDRODYNAMIC SETS` for further information on these data inputs.

BUOYANCY PROPERTIES

Flexcom requires an effective outside diameter in order to compute stresses such as bending or Von Mises. For properties assigned using the Flexible Riser format, the default outer diameter is assumed equal to the drag diameter. In this example, the default value is suitable for the bare section, but is incorrect for the buoyant section. So the `*PROPERTIES` keyword is used to explicitly assign the correct outer diameter for the purposes of stress computations.

Properties - Stress

Set Name	Do [m]	Di [m]	A [m^2]	Iyy [m^4]	Izz [m^4]	Min. Wall Thickness [m]
Buoyant Section	0.25					

```
*PROPERTIES
 SET=Buoyant Section
 0.25
```

Buoyancy Properties

Environment

OCEAN

Flexcom traditionally groups together four parameters required to provide the minimum environment specification, namely the analysis water depth, the density of seawater ρw, the gravitational constant g, and ν, the kinematic viscosity of seawater. The environmental parameters for this example are shown below. Refer to Environment Parameters for further information on this feature.

Ocean - Properties

Water Depth	300
Water Density	1025
Acceleration due to Gravity	9.81
Water Kinematic Viscosity	[0.0000013]

```
*OCEAN
 300, 1025, 9.81
```

Ocean Parameters

Refer to `*OCEAN` for further information on these data inputs.

SEABED

Flexcom has a range of options for modelling seabed interaction. Both rigid and elastic seabed models are provided, and seabeds of either type may be either flat or sloping, or an arbitrary bathymetry may be specified. Seabeds of either type may be smooth or have longitudinal and/or transverse friction included. For structures running through and below the mudline, soil-structure interaction may also be modelled by the specification of P-y curves. Refer to Seabed Interaction for further information on this feature.

Rigid Seabed Properties

Set Name	Longitudinal Friction Coefficient	Transverse Friction Coefficient
All	[0.0]	[0.0]

```
*SEABED PROPERTIES
 TYPE=RIGID
 SET=All
```

Seabed Parameters

Refer to *SEABED PROPERTIES for further information on these data inputs.

Vessel and RAO Data

Two sets of inputs are required to completely define the response of a floating vessel to the ambient wave field. The first set defines the vessel initial position, that is, the location of the vessel prior to the application of any vessel offset or dynamic motions. This information is defined in terms of the location of the vessel reference point (the point at which the RAOs are defined), and initial yaw orientation of the vessel.

The second set of inputs is the full vessel RAOs and phase angles in all vessel degrees of freedom. Given the vessel's initial position, offset and RAOs, in addition to the time history of wave elevation, the motion of the reference point in response to the wave field throughout an analysis can be calculated, and from this the motion of the point(s) on the structure attached to the vessel. As the RAO data may be quite extensive, it is typically stored in a separate file which is simply referenced by the main keyword file. The file with RAO data already exists in the *Worked Example - Basic* folder, and is named *K1-SWR-RAO.incx*. This should be copied into the folder for this project. Refer to High Frequency RAO Motions for further information on this feature.

You may also optionally associate a structural profile with the vessel. While this does not affect analysis results, it does add considerably to the visual appeal of the structural animation. A range of standard vessel profiles are provided; the FPSO profile being particularly suited to this worked example.

The vessel in this model is located at X=305m, Y=120m, Z=0m and has an initial yaw orientation of 50 degrees with respect to the global Y-axis. The completed specification of vessel data for the FPSO in this example is specified as shown in the figure below. Note that all vessel data (apart from vessel offsets) is specified in the $MODEL section of the file.

Vessel Integrated - Setup

Vessel	Focal Point X	Focal Point Y	Focal Point Z	Yaw	RAO File	Format	Units	Ref Pt Offset X	Ref Pt Offset Y
FPSO	305	120	0	50	K1-SWR-RAO.incx			[0.0]	[0.0]

Ref Pt Offset Z	Profile	Height	Length	Width	User Profile	COP Offset X	COP Offset Y	COP Offset Z	Angle Theory
[0.0]	FPSO	55	170	30		-7	0	0	[Large Angles]

```
*VESSEL,INTEGRATED
 VESSEL=FPSO
 INITIAL POSITION=305, 120, 0, 50
 RAO=K1-SWR-RAO.incx
 PROFILE=FPSO
 DIMENSIONS=55,170,30
 COP=-7, 0, 0
```

Vessel Data

Refer to *VESSEL INTEGRATED for further information on these data inputs.

Initial Static Analysis

ANALYSIS TYPE

The type of analysis to be performed must be specified in each $LOAD CASE section. In this example, a static analysis is firstly performed.

Analysis Type

Analysis Type	STATIC

```
*ANALYSIS TYPE
 TYPE=STATIC
```

Analysis Type

Refer to `*ANALYSIS TYPE` for further information on these data inputs.

BOUNDARY CONDITIONS

Flexcom presents a comprehensive range of options to fully describe the constraints which are applied to the finite element model. The various options include constant (time invariant constraints), vessel (to apply vessel motions), and several others including an arbitrary user-subroutine facility which provides for complete generality in terms of constraint application. The first two options are invoked in this example, with the base of the riser and vessel fixed in three of freedom. Note that the relevant node labels (i.e. *Lower End* and *Upper End*), are created automatically by the lines facility during model building, and referenced when defining boundary conditions. The specification of boundary condition data goes into the `$LOAD CASE` section of the file. Refer to Boundary Conditions for further information on this feature.

Boundary - Constant - Direct

Node	DOF	Displacement	Fixation
{Lower End}	1	0.0	ABSOLUTE
{Lower End}	2	0.0	ABSOLUTE
{Lower End}	3	0.0	ABSOLUTE

```
*BOUNDARY
 TYPE=CONSTANT
 {Lower End}, 1, 0.0
 {Lower End}, 2, 0.0
 {Lower End}, 3, 0.0
 TYPE=VESSEL, VESSEL=FPSO
 {Upper End}, 1, 0.0
 {Upper End}, 2, 0.0
 {Upper End}, 3, 0.0
```

Initial Static Analysis - Boundary Conditions

Refer to `*BOUNDARY` for further information on these data inputs.

INTERNAL FLUID

Flexcom provides a comprehensive internal fluid modelling capability. Stationary internal fluids, uniform steady state internal fluid flow and multi-phase slug flow may all be modelled. A stationary internal fluid in Flexcom is characterised by its mass density, pressure above hydrostatic and the level above the mudline to which it extends. In this example, the riser is filled with oil, as evident from its mass density. The level above mudline input is set equal to the riser elevation – Flexcom uses this input to determine whether elements are filled with fluid, and also to compute the hydrostatic pressure. Notice that internal fluid data is specified in a newly created `$LOAD CASE` section of the file. Refer to Internal Fluid for further information on this feature.

Internal Fluid

Set Name	Level Above Mudline	Mass Density	Internal Pressure	Velocity
All	290	880	18E6	[0.0]

```
*INTERNAL FLUID
 SET=All
 290, 880, 18E6
```

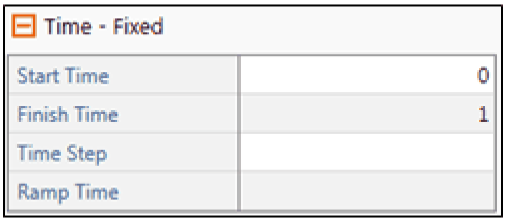
Internal Fluid

Refer to `*INTERNAL FLUID` for further information on these data inputs.

SIGNIFICANCE OF TIME VARIABLES IN STATIC ANALYSIS

Naturally, all time domain analyses require the specification of time variables. Since a static analysis is one which considers time invariant loading and structure response, time variables are effectively meaningless in this context, however it is required to specify time variables for a static analysis in Flexcom. This is purely a consequence of the historical evolution of the software.

An initial static analysis is typically run from t=0 to t=1 second using a single fixed time step, and this standard approach is adopted for this example, as shown below. Refer to Time Variables in Static Analysis for further information on this feature.




```
*TIME
 STEP=FIXED
 0, 1
```

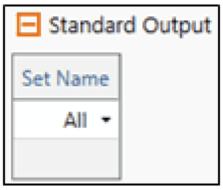
Initial Static Analysis Time Variables

Refer to `*TIME` for further information on these data inputs

DATABASE POSTPROCESSING

Database Postprocessing is the most powerful and widely used postprocessing facility in Flexcom. The database files provide a very detailed picture of the structure response. For large models, or long simulations with a large number of database outputs, the database files can become quite large. However, this is not a concern for static analyses, or indeed the dynamic analysis studied in this example. Refer to Database Postprocessing for further information on this feature.

In this example, the postprocessing requests are specified before the analysis begins, via the standard output option. This option allows you to quickly request a summary of pertinent information, without the inconvenience of explicitly requesting specific outputs. For every element set referenced, Flexcom produces outputs of effective tension, resultant bending moment and von Mises stress for that set.




```
$DATABASE POSTPROCESSING
*STANDARD OUTPUT
  SET=All
```

Database Postprocessing Requests

Refer to *STANDARD OUTPUT for further information on these data inputs.

RUNNING ANALYSIS

Specification of the model geometry, structural and hydrodynamic properties, environment and loading data, vessel and RAO data, boundary conditions and solution parameters is now complete, so you should save the keyword file and run the analysis.

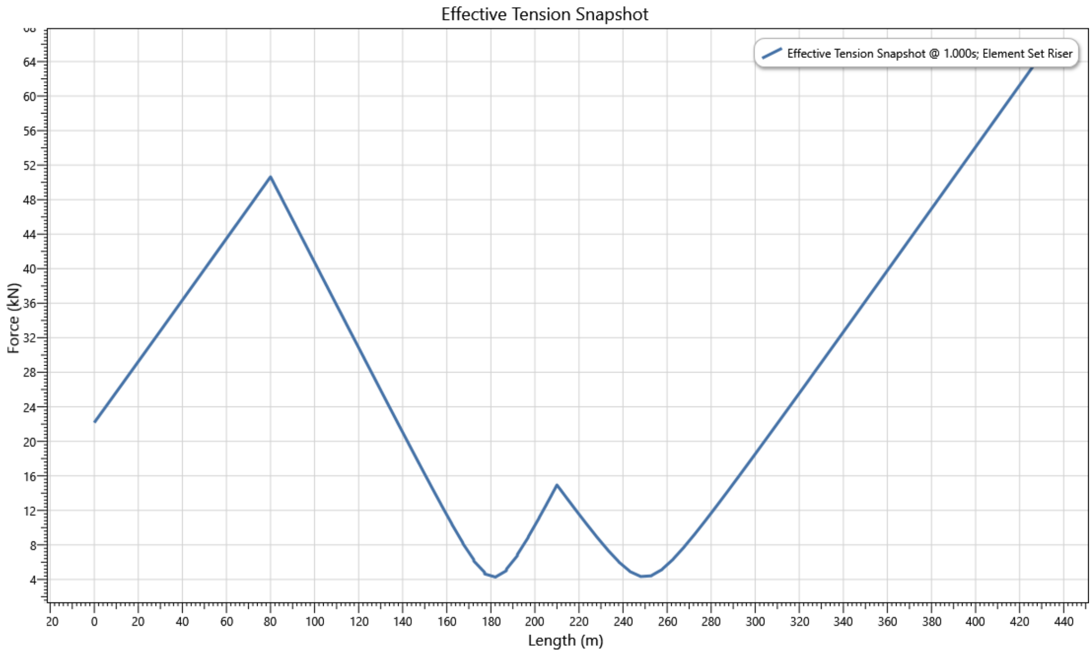
When a run is in progress, you can monitor its status via the Analysis Status View. The progress bar is naturally most beneficial for longer dynamic analyses (e.g. where it provides an approximate estimate of remaining CPU time), and sample progress information is shown below.

Analysis Progress Indicator

Another useful feature in the File View is that the program provides an indication of the status of the analysis. When the run has finished, the icon associated with the keyword file changes to , indicating that the analysis has completed successfully. You may have noticed the icon next to the keyword file name was previously , indicating the analysis had not yet been run. As the computation time for the static analysis is relatively short, you probably won’t have noticed that the icon briefly changed to , indicating that the analysis was currently in progress

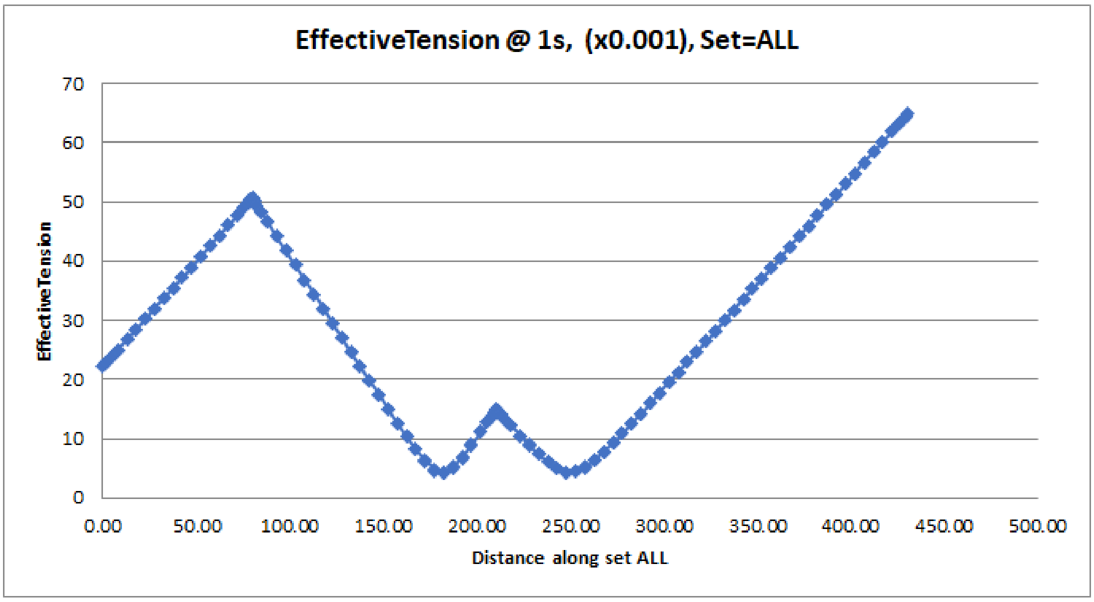
EXAMINING RESULTS

You can view the analysis results using the Plotting facility. Plot files have the file extension .MPLT (an abbreviation of Mcs PLoT). The figure below shows the static effective tension distribution in the riser (this corresponds to the file K1-SWR-static.S1.mplt, or similar, depending on your naming convention). You can examine the other plots created for the static analysis at your convenience. Refer to Plotting for further information on this feature



Static Effective Tension

Similar results can obtained using the [Excel Add-in](), as shown below. This plot was quickly created using the *Force Snapshot* template from the [Flexcom ribbon control]().



Static Effective Tension

Dynamic Analysis

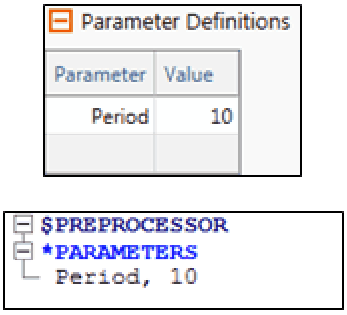
The presence of waves, whether regular or random, requires you to specify a dynamic analysis. As Flexcom has traditionally been a time domain analysis tool, existing users may instinctively associate “dynamic analysis” with “time domain”. The vast majority of time domain analyses are non-linear, and this option is selected by default. Refer to Time Domain Analysis for further information on this feature.

Create a new blank keyword file which uses the metric unit system and name it K1-SWR-dynamic.keyx, (or similar, depending on your naming convention).

KEYWORD PARAMETERISATION

You may define constant parameters within a keyword file, and reference these parameters when defining input variables. If you choose to define any such parameter, their definitions must appear at the beginning of the keyword file, under the `$PREPROCESSOR` section. When referencing a parameter, the variable definition must be preceded by the characters =[and followed by the] character. Specifically the variable definition will appear in the format =[Parameter], rather than just having an explicitly specified value. Refer to Parameters for further information on this feature.

For the purposes of this example, we will examine a regular wave load case, corresponding to wave periods of 10s, though in practice many more waves would also be examined. A new parameter call Period is defined, and assigned a value of 10 seconds as shown below.

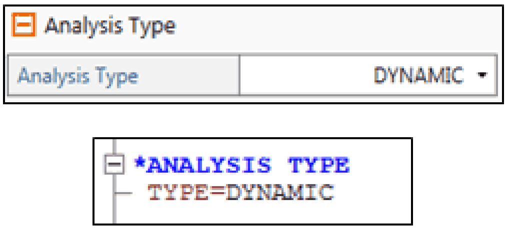


Period Parameter

Refer to `*PARAMETERS` for further information on these data inputs.

ANALYSIS TYPE

The type of analysis to be performed must be specified in each `$LOAD CASE` section. In this example, a dynamic analysis is required.

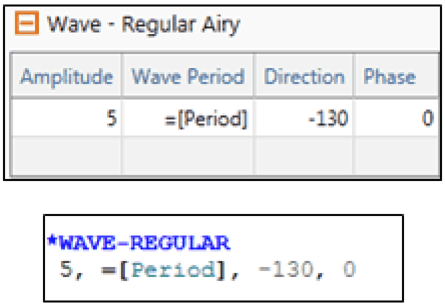


Analysis Type

Refer to *ANALYSIS TYPE for further information on these data inputs.

REGULAR WAVE

In this example, a regular wave of period of 10 seconds and amplitude of 5 meters is created. The wave direction is measured in degrees anti-clockwise from the global Y-direction and all waves in Flexcom emanate from the origin. The default direction is 0°, however for this example the direction is set at -130° aligned with the local vessel surge axis. Refer to Regular Airy Wave for further information on this feature.



Wave Properties

Refer to *WAVE-REGULAR for further information on these data inputs.

TIME STEPPING

In a variable step analysis, the choice of time step magnitude is made by the program based on a number of criteria. The time step is continuously monitored and varied as appropriate by the program within user-specified limits, to ensure a stable and convergent solution. A variable step is typically used in analyses where the structural response varies significantly during the course of the simulation. Appropriate time variables are computed based on the regular wave period. The dynamic analysis runs for 5 wave periods, with the loads ramped on over the first wave period. A fixed time step equal to 5% of the wave period is used. Refer to Variable Time Stepping and Choice of Time Step for further information on these features.

Time - Fixed	
Start Time	1
Finish Time	=[5*Period]
Time Step	=[Period*0.05]
Ramp Time	=[Period]

```
*TIME
 STEP=FIXED
 1, =[5*Period], =[Period*0.05], =[Period]
```

Dynamic Analysis Time Stepping

Refer to *TIME for further information on these data inputs.

RESTART

The Restart facility allows us to specify that a particular analysis is to be restarted from a previous run, to build up to the full dynamic solution in stages. In a restart, the structure configuration at the end of the preceding analysis becomes the starting configuration for the restart. Refer to Restart Analysis for further information on this feature.

In this case we want to restart the present run from *the K1-SWR-static* file.

Restart	
Type	[New Loads or BCs]
Restart File	"K1-SWR-static"
Append	

```
*RESTART
 LAST="K1-SWR-static"
```

Restart Analysis

Refer to `*RESTART` for further information on these data inputs.

DATABASE POSTPROCESSING

For this analysis the Statistics output category is used. Options are provided for plotting statistics of motions and forces for specified element sets or the whole structure. Outputs include maximum/minimum envelopes, mean values, standard deviations and extreme values. Refer to Database Postprocessing for further information on this feature.

In this example a plot of the maximum/minimum envelopes of effective tension and bending moment is requested. Variable values are assigned to different force statistics for ease of use. In this case the variable value of 7 corresponds to effective tension and the value of 8 corresponds to bending moment. The statistic output feature is calculated over the last two periods in this example. Flexcom excludes any values before this time.

Statistics - Force

Parameter	Variable	Start Time [s]	Scale	Title	Element Set	Location	Unit
Envelope	Effective tension	=[3*Period]	[1.0]	Effective Tension Envelope	[All]		kN
Envelope	Resultant bending moment	=[3*Period]	[1.0]	Bending Moment Envelope	[All]		kN.m

```
$DATABASE POSTPROCESSING
*STATISTICS
 TYPE=FORCE
 ENVELOPE, 7, =[3*Period], , , UNITS=kN
 TITLE=Effective Tension Envelope
 TYPE=FORCE
 ENVELOPE, 8, =[3*Period], , , UNITS=kN.m
 TITLE=Bending Moment Envelope
```

Database Postprocessing Requests

Refer to `*STATISTICS` for further information on these data inputs.

RUNNING THE ANALYSIS

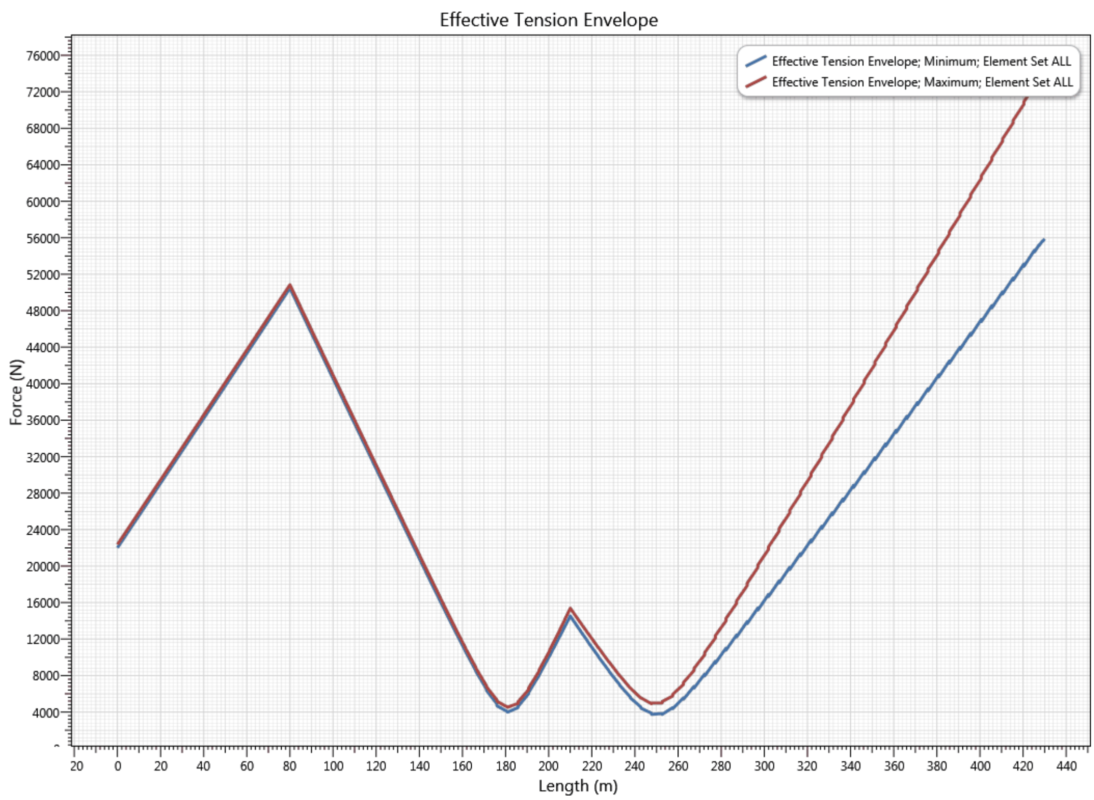
Once the specification of the dynamic data is complete, you should save the keyword file and run the analysis.

When a run is in progress, you can monitor its status via the Analysis Status area. The progress bar is naturally most beneficial for longer dynamic analyses (e.g. where it provides an approximate estimate of remaining CPU time), and sample progress information is shown below.

Another useful feature in the File View is that the program provides an indication of the status of the analysis. When the run has finished, the icon associated with the keyword file changes to , indicating that the analysis has completed successfully. You may have noticed the icon next to the keyword file name was previously , indicating the analysis had not yet been run. As the computation time for the static analysis is relatively short, you probably won't have noticed that the icon briefly changed to , indicating that the analysis was currently in progress

EXAMINING RESULTS

You can view the analysis results using the Plotting facility. Plot files have the file extension .MPLT (an abbreviation of Mcs PLoT). The plot below shows the static effective tension distribution in the riser (this corresponds to the file *K1-SWR-dynamic.D1.mplt*, or similar, depending on your naming convention). You can examine the other plots created for the static analysis at your convenience. Refer to Plotting for further information on this feature



Dynamic Analysis - Effective Tension Envelope

1.10.11.2 K02 - Worked Example - Complex

This worked example takes you through all of the stages in performing an analysis with Flexcom. It begins by describing the structure under consideration, and then details the steps in creating the finite element model. It then goes on to work through all the analysis stages, from finding the static configuration in an initial analysis, to creating and running a matrix of dynamic load cases. At each stage, appropriate analysis results are presented. Along the way, Flexcom utilities and tools are described as appropriate.

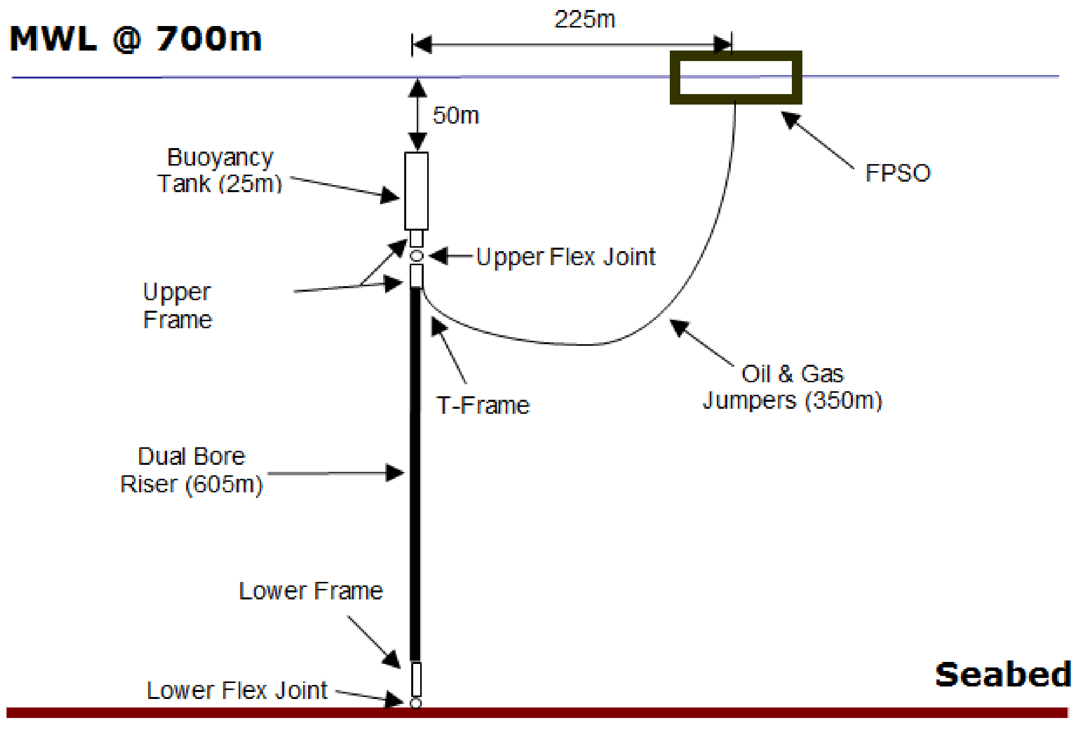
The example is now presented using the following sub-sections:

- Riser System summarises the hybrid riser system configuration which forms the basis of the worked example.

- Project Structure introduces the concept of a highly project-focused integrated engineering environment
- Model Building describes the model building process, focusing particularly on the lines feature, which essentially provides an automatic mesh creation facility to expedite the model creation process.
- Structural and Hydrodynamic Properties outlines the assignment of structural and hydrodynamic properties to the model.
- Environment and Loading outlines the environment (e.g. ocean depth etc.) and loading (e.g. internal fluid) specification.
- Vessel and RAO Data outlines the vessel definition and the application of vessel RAOs.
- Boundary Conditions outlines the constraints which are applied to the finite element model.
- Initial Static Analysis describes how to perform the initial static analysis, to extract results via Database Postprocessing, and to examine graphical output using the Plotting facility.
- Vessel Offset Analyses describes how to perform restart static vessel offset analyses, and illustrates the use of the keyword parameterisation facility to create load case variations.
- Regular Wave Analyses describes how to perform restart regular wave analyses, how to extract results via Summary Postprocessing, and illustrates the keyword files interdependencies in the project structure.
- Code Checking Post-Processing describes how to use the DNV and API code checking module for code compliance and unity checking.
- Results Collation introduces the Summary Postprocessing Collation facility, and illustrates the assembly of pertinent results across the range of regular wave load cases.
- Video Creation introduces the AVI Studio program, and describes a sample AVI video creation.

Riser System

This worked example considers a hybrid riser system – the model is a top tensioned riser concept, based on a free standing hybrid riser arrangement. The system consists of a dual bore vertical rigid steel riser, anchored to the seabed and tensioned by means of a buoyancy tank, with a connection to an FPSO through two flexible jumpers. One of the jumpers takes oil from the wellhead, while the other injects gas into the production stream. A T-frame connects the jumpers to the assembly at the upper end of the riser. The riser system is situated in 700m of water, with the top of the buoyancy tank positioned approximately 50m below the water surface. A schematic of the overall hybrid riser system configuration is shown in the below figure.



Hybrid Riser System Configuration

The central structure is comprised of the following components:

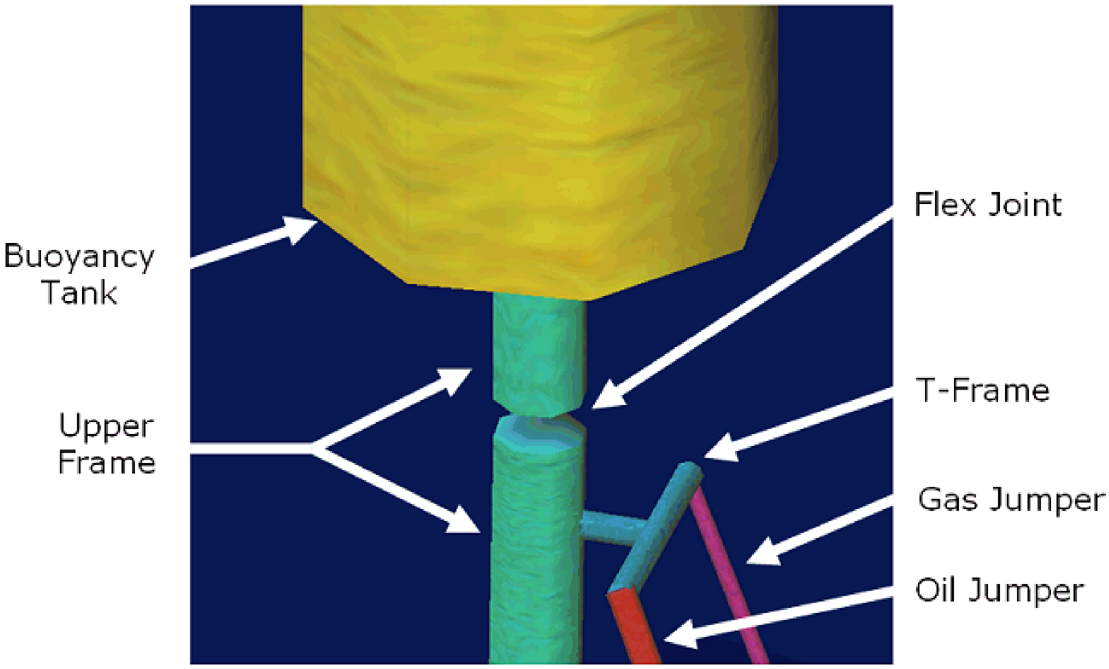
- The lower flex joint, which is 1m in length. The stiffness of the joint is 20kNm/degree.
- The lower frame, which is 9m in length. The frame is effectively rigid, and has a mass of 35 tonnes.

- The main riser, which is 605 m in length. The outer pipe has an external diameter of 364mm, and a wall thickness of 24mm. It is also coated with an additional 80mm wall thickness of insulation material (of density 900 kg/m^3). The inner pipe has an external diameter of 296mm, and a wall thickness of 21mm. The outer and inner risers are connected with centralisers at regular intervals (of 25m). The inner pipe is oil filled (of density 800 kg/m^3) while the annulus is gas filled (of density 200 kg/m^3).
- The upper frame, which is 9m in length. The frame is effectively rigid, and has a mass of 35 tonnes.
- The upper flex joint, which is 1m in length, is located 5m along the upper frame. The stiffness of the joint is 20kNm/degree.
- The buoyancy tank, which is 25m in length. The tank is effectively rigid, and has a mass of 225 tonnes. It has a considerable external diameter of 6m.

The jumpers have the following specifications:

- Both jumpers are 350m in length.
- The oil (of density 800 kg/m^3) jumper has an external diameter of 370mm, and a wall thickness of 58mm.
- The gas (of density 200 kg/m^3) jumper has an external diameter of 210mm, and a wall thickness of 65mm.

The T-Frame is effectively a rigid massless structure. The below figure shows a close-up of the intersection region between the jumpers and central structure.



Intersection Region

Project Structure

If you are unfamiliar with the basic components of Flexcom's user interface, please refer to the Interface section first. In particular, this example assumes some level of familiarity with the Keyword Editor, Table Editor and Model View components.

In order to create a new project, select New Project from the File menu on the top menu bar. You will then be presented with options for defining a Project Name and a Project Folder. You can specify any Project Name you like – indeed the same is true for any of the keyword files which will be described in the following sections. It is advised that you use a Project Folder other than the Worked example folder in the Flexcom installation directory. This will ensure that you do not overwrite the original files provided with the program installation, and will allow you to use these files as a reference to verify your own files should you experience any problems running the example.

Once the project is created, you need to add the first keyword file to the project. To add a keyword file, select New File from the File menu on the top menu bar. You will be presented with options to select a template and name for the file. Enter the name Static.keyx (or similar, depending on your naming convention) and create an empty keyword file. The file should open in the main document panel. You should also see the file added to the project in the File View.

Model Building

The model is largely created using lines. The majority of the central structure, including the upper and lower flex joints, the upper and lower frames, the outer pipe and the buoyancy tank, is modelled using a single line. A separate line is used to model the inner pipe. Another two lines are used to model the oil and gas jumpers. Finally, two further short lines are used to model the T-frame, which connects the jumpers to the central structure.

OVERVIEW OF LINES FEATURE

The lines functionality is a relatively recent addition to Flexcom, and essentially provides an automatic mesh creation facility whereby nodes, elements and cables can be generated quite easily. Using lines to build your model is a fundamentally different approach to working directly with nodes, elements and cables. While all the information is ultimately handled in the same fashion internally in the software, the use of lines expedites the model creation process considerably, as you need not concern yourself with individual node and element numbering. While the explicit numbering scheme (which was the only approach available in earlier versions of Flexcom) may appear obsolete by comparison, it is retained for complete generality, and also to maintain downward compatibility with previous versions. Refer to Lines for further information on this feature.

CENTRAL STRUCTURE

The majority of the central structure is modelled using a single line, running all the way from the seabed to the top of the buoyancy tank. This line includes the upper and lower flex joints, the upper and lower frames, the outer pipe and the buoyancy tank. For most vertical structures (e.g. drilling risers), a single line would be sufficient, but this model is slightly complicated by the fact that the riser is dual bore, so a separate line is required to model the inner pipe.

The first line is called Central Structure, and is assigned a length of 650m. Its start and end locations are called Seabed End {0m, 0m, 0m} and Buoyancy Tank Top {650m, 0m, 0m}. Obviously the structural properties of the structure vary along its length due to the different components, so it is necessary to define a number of subsections in terms of distances along the line. The subsections are called Lower Flex Joint (from 0m to 1m), Lower Frame (from 1m to 10m), Outer Pipe (from 10m to 615m), Upper Frame (from 615m to 620m, and from 621m to 625m), Upper Flex Joint (from 620m to 621m) and Buoyancy Tank (from 625m to 650m). In order to ensure that the automatic meshing algorithm positions a node on the upper frame at the correct elevation for connection to the T-frame, a line location is defined on the Central Structure called Central Structure T-Frame, at a distance of 619m along the line (i.e. 1m below the upper flex joint).

In terms of element lengths, it is desirable to model the upper and lower flex joints using a single element, so a suggested length of 1m is specified (which is equal to the joint length in both cases). A constant element length of 1m is also suggested for the lower and upper frames. Suggested minimum and maximum lengths of 1m and 5m, respectively, are suggested for the outer pipe. A constant element of 0.5m is suggested for the buoyancy tank. The definition of the Central Structure line in both the table and keyword editor is shown in the figure below.

Input data is organised into two main categories, model data and load case data. This distinction is reflected in both the table and keyword editors. Model data consists of information which must be specified in the very first of a series of consecutive analyses. This data is carried through to all subsequent restart analyses and may not be changed. This category naturally includes the finite element discretisation, structural and hydrodynamic properties, plus any other inputs which characterise the initial model configuration (e.g. initial vessel position, seabed properties, ocean depth etc.). Load case data may be entered in any analysis of a series of runs, and which may be subsequently altered in restart analyses. This category naturally includes environmental loading such as current and waves, boundary conditions, solution variables etc. There is a clear division between model and load case data in the keyword editor, via the $MODEL and $LOAD CASE sections in the keyword file. Similarly, in the table editor, the sections are split onto different tabs named MODEL and LOAD CASE. Before you enter the first keyword (i.e. `*LINES`, to define the central structure), you should include a $MODEL section entry. You can also add a MODEL section through the table editor by right-clicking on the table editor and selecting Add Section from the context menu.

Note:

The width of the columns in the table editor images that follow have been reduced due to formatting contraints in this document. Refer to the corresponding keyword editor excerpt to view the data values.

Lines

Line Name	Non-Se	Length	Start Node	Start (X)	Start (Y)	Start (Z)	End Node Label	End (X)	End (Y)	End (Z)	Min. Ele
Central Structure		650	Seabed End	0	0	0	Buoyancy Tank Top	650	0	0	2

Line Sections

Line Name	Section Set Name	Start Distance	End Distance	Min. Element Length	Max. Element Length
Central Structure	Lower Flex Joint	0	1	1	
	Lower Frame	1	10	1	
	Outer Pipe	10	615	1	5
	Upper Frame 1	615	620	1	
	Upper Flex Joint	620	621	1	
	Upper Frame 2	621	625	1	
	Buoyancy Tank	625	650	0.5	

Element Sets

Set Name	Elements	SubSets
Upper Frame		Upper Frame 1
		Upper Frame 2

Line Locations

Line Name	Location Name	Distance Along Line
Central Structure	Central Structure T-Frame	619

```
$MODEL
*LINES
 LINE=Central Structure, 650
 START=Seabed End, 0, 0, 0
 END=Buoyancy Tank Top, 650, 0, 0
 2
 SECTION=Lower Flex Joint, 0, 1
 1
 SECTION=Lower Frame, 1, 10
 1
 SECTION=Outer Pipe, 10, 615
 1,5
 SECTION=Upper Frame 1, 615, 620
 1
 SECTION=Upper Flex Joint, 620, 621
 1
 SECTION=Upper Frame 2, 621, 625
 1
 SECTION=Buoyancy Tank, 625, 650
 0.5

*ELEMENT SETS
 SET=Upper Frame
 SUBSET=Upper Frame 1
 SUBSET=Upper Frame 2

*LINE LOCATIONS
 LINE=Central Structure
 LABEL=Central Structure T-Frame, 619
```

Central Structure Line

Refer to the *LINES, *LINE LOCATIONS and *ELEMENT SETS keywords for further information on these data inputs.

A second line (aptly named Inner Pipe) is defined to model the inner pipe, and is assigned a length of 605m. Its start and end locations are called Riser Lower End {10m, 0m, 0m} and Riser Upper End {615m, 0m, 0m}. As the inner pipe is composed of homogenous material with constant properties along its entire length, it is not necessary to define any subsections of the line. For consistency with the outer pipe, suggested minimum and maximum lengths of 1m and 5m, respectively, are also suggested for the inner pipe. The definition of the Inner Pipe line is shown in the figure below. Note that that symbol indicates that the following keyword data is supplementary to the existing data under the same keyword as shown previously.

Lines

Line Name	Non-Se	Length	Start Node	Start (X)	Start (Y)	Start (Z)	End Node Label	End (X)	End (Y)	End (Z)	Min. Ele	Max. Ele
Central Structure		650	Seabed End	0	0	0	Buoyancy Tank Top	650	0	0	2	
Inner Pipe		605	Riser Lower	10	0	0	Riser Upper End	615	0	0	1	5

```
*LINES
:
LINE=Inner Pipe, 605
START=Riser Lower End, 10, 0, 0
END=Riser Upper End, 615, 0, 0
1,5
```

Inner Pipe Line

The upper and lower ends of the outer and inner pipes are rigidly connected using bulkheads (these are automatically modelled internally using equivalent nodes). The pipes are also connected at regular intervals (of 25m) using centralisers, and these are modelled using (linear) pipe-in-pipe connections of relatively large stiffness. Nonlinear “gap” connections are modelled at all nodal locations intermediate to the centralisers. The definition of the various pipe-in-pipe connections is shown in the figure below. Refer to Pipe-in-Pipe Configurations for further information on this feature.

Line - Pipe-in-Pipe Connections

Table Name	Type	Spacing	Start	End	Stiffness
PIPConTable1	Bulkhead	605	10		
	PIP Connection	25	10		1.0E+06
	PIP Connection	5	10		Gap

Line - Pipe-in-Pipe Sections

Primary Line Name	Secondary Line Name	Configuration Type	Connections Table	Offset For Secondary Line
Central Structure	Inner Pipe	Pipe-in-pipe	PIPConTable1	10

Pipe-in-Pipe Stiffness - Power Law

Curve Name	Contact Force	Exponent
Gap	1.0E+06	10

```
*LINES PIP
 PRIMARY=Central Structure, SECONDARY=Inner Pipe, TYPE=PIP, OFFSET=10
 BULKHEAD, 605, 10
 STIFFNESS=1.0E+06, 25, 10
 CURVE=Gap, 5, 10

*PIP STIFFNESS
 CURVE=Gap, TYPE=POWER LAW
 1.0E+06, 10
```

Pipe-in-Pipe Connections

Refer to the `*LINES PIP` and `*PIP STIFFNESS` keywords for further information on these data inputs.

JUMPERS

Two separate lines, Oil Jumper and Gas Jumper are used to model the jumpers, both having an assigned length of 350m. The start and end locations for the former are called Oil Jumper FPSO {695m, 225m, -1.5m} and Oil Jumper T-Frame {619m, 1.5m, -1.5m}, and Gas Jumper FPSO {695m, 225m, 1.5m} and Gas Jumper T-Frame {619m, 1.5m, 1.5m} for the latter. Suggested minimum and maximum lengths of 1m and 5m, respectively, are suggested for both jumpers. The definition of the Oil Jumper and Gas Jumper lines are shown in the figure below. Note that that ':' symbol indicates that the following keyword data is supplementary to the existing data under the same keyword as shown previously.

Lines

Line Name	Non-Se	Length	Start Node	Start (X)	Start (Y)	Start (Z)	End Node Label	End (X)	End (Y)	End (Z)	Min. Ele	Max. Ele
Central Structure		650	Seabed End	0	0	0	Buoyancy Tank Top	650	0	0	2	
Inner Pipe		605	Riser Lower	10	0	0	Riser Upper End	615	0	0	1	5
Oil Jumper		350	Oil Jumper F	695	225	-1.5	Oil Jumper T-Frame	619	1.5	-1.5	1	5
Gas Jumper		350	Gas Jumper	695	225	1.5	Gas Jumper T-Fram	619	1.5	1.5	1	5

```
*LINES
 ⋮
 LINE=Oil Jumper, 350
 START=Oil Jumper FPSO, 695, 225, -1.5
 END=Oil Jumper T-Frame, 619, 1.5, -1.5
 1, 5

 LINE=Gas Jumper, 350
 START=Gas Jumper FPSO, 695, 225, 1.5
 END=Gas Jumper T-Frame, 619, 1.5, 1.5
 1, 5
```

Refer to the *LINES keyword for further information on these data inputs.

T-FRAME

The T-Frame is modelled using two short lines. The first line, T-Frame-Top, is 3m long, and its start and end locations are T-Frame-1 {619m, 1.5m, -1.5m} and T-Frame-2 {619m, 1.5m, 1.5m}. The second line, T-Frame-Leg, is 1.5m long, and its start and end locations are T-Frame-4 {619m, 0m, 0m} and T-Frame-5 {619m, 1.5m, 0m}. In order to model this portion of the T-Frame using a single element, a suggested length of 1.5m is specified. The definition of the T-Frame lines is shown in the figure below. Note that that '⋮' symbol indicates that the following keyword data is supplementary to the existing data under the same keyword as shown previously.

Lines

Line Name	Non-Se	Length	Start Node	Start (X)	Start (Y)	Start (Z)	End Node Label	End (X)	End (Y)	End (Z)	Min. Ele	Max. Ele
Central Structure		650	Seabed End	0	0	0	Buoyancy Tank Top	650	0	0	2	
Inner Pipe		605	Riser Lower	10	0	0	Riser Upper End	615	0	0	1	5
Oil Jumper		350	Oil Jumper F	695	225	-1.5	Oil Jumper T-Frame	619	1.5	-1.5	1	5
Gas Jumper		350	Gas Jumper	695	225	1.5	Gas Jumper T-Fram	619	1.5	1.5	1	5
T-Frame-Top		3	T-Frame-1	619	1.5	-1.5	T-Frame-2	619	1.5	1.5	1.5	
T-Frame-Leg		1.5	T-Frame-4	619	0	0	T-Frame-5	619	1.5	0	1.5	

```
*LINES
 ⋮
 LINE=T-Frame-Top, 3
 START=T-Frame-1, 619, 1.5, -1.5
 END=T-Frame-2, 619, 1.5, 1.5
 1.5

 LINE=T-Frame-Leg, 1.5
 START=T-Frame-4, 619, 0, 0
 END=T-Frame-5, 619, 1.5, 0
 1.5
```

T-Frame Lines

Refer to the *LINES keyword for further information on these data inputs.

The intersection of the T-Frame is located at T-Frame-3, 1.5m along the line T-Frame-Top, as shown in the figure below.

Line Locations

Line Name	Location Name	Distance Along Line
Central Structure	Central Structure T-Frame	619
T-Frame-Top	T-Frame-3	1.5

```
*LINE LOCATIONS
 ⋮
 LINE=T-Frame-Top
 LABEL=T-Frame-3, 1.5
```

Line Locations on T-Frame

Refer to the *LINE LOCATIONS keyword for further information on these data inputs.

The top of the T-Frame is rigidly connected to its leg in reality, and the leg is rigidly connected to the central structure. Nodal equivalences are defined in the model in order to model these connections, as shown in the figure below. Refer to Connecting Lines for further information on this feature.

Nodes - Equivalent

First Node	Second Node
{T-Frame-3}	{T-Frame-5}
{T-Frame-4}	{Central Structure T-Frame}

```
*EQUIVALENT
 {T-Frame-3}, {T-Frame-5}
 {T-Frame-4}, {Central Structure T-Frame}
```

Nodal Equivalences on T-Frame

Refer to the *EQUIVALENT keyword for further information on these data inputs.

The top of the T-Frame is connected to the jumpers using free hinges in reality. Hinge elements are defined in the model in order to model these connections, as shown in the figure below. Refer to Hinges and Flex Joints for further information on this feature.

Elements - Define Directly

Element	First Node	Last Node	V1	V2	V3	W1	W2	W3
1	{T-Frame-1}	{Oil Jumper T-Frame}						
2	{T-Frame-2}	{Gas Jumper T-Frame}						

Hinges

Element Number
1
2

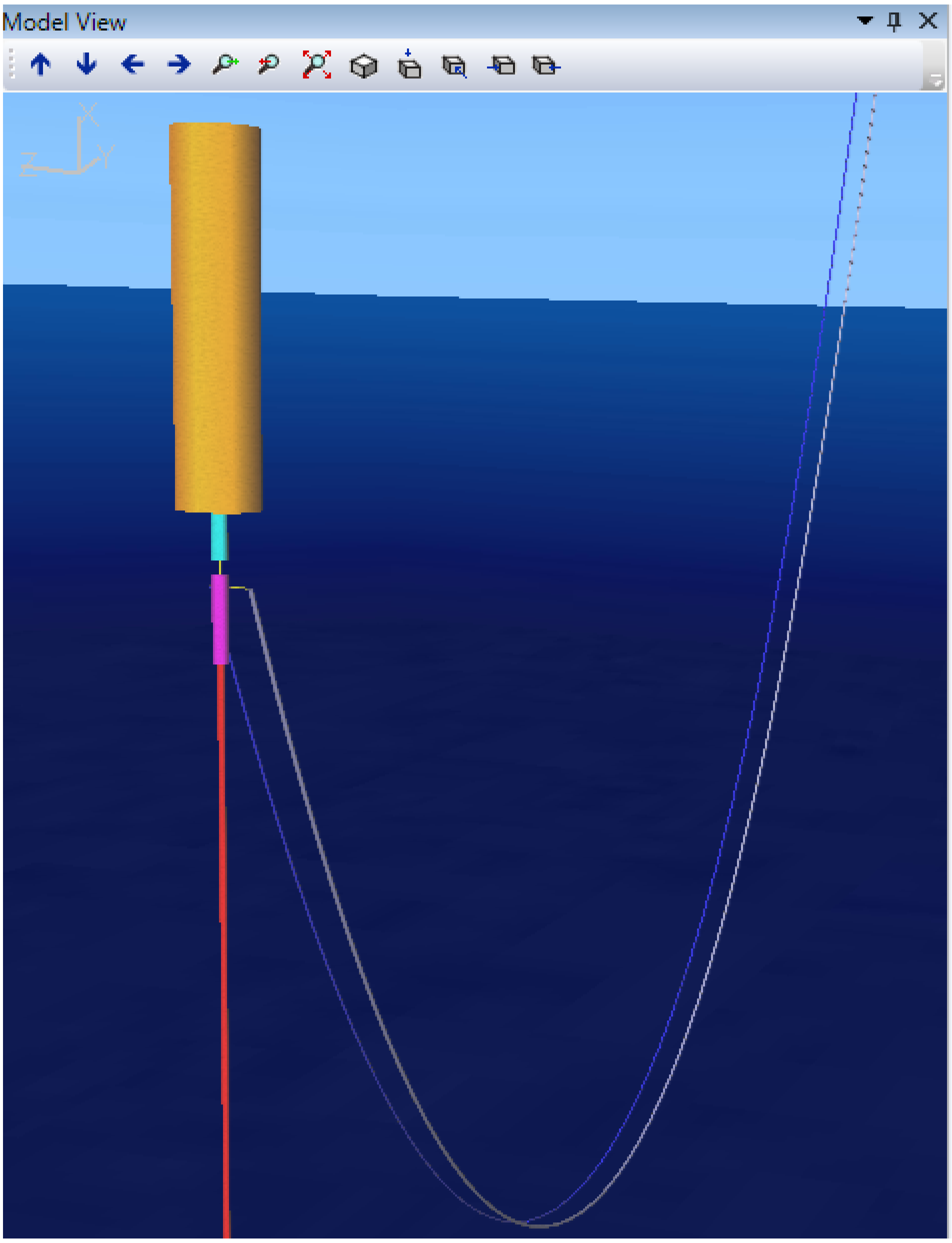
```
*ELEMENT
 1, {T-Frame-1}, {Oil Jumper T-Frame}
 2, {T-Frame-2}, {Gas Jumper T-Frame}

*HINGE
 1
 2
```

Hinge Elements

Refer to the *ELEMENT and *HINGE keywords for further information on these data inputs.

MODEL VIEW

The Model View provides a “live” structure preview facility available during model building. It also allows you to view an animation of the structure response after an analysis has completed. The Model View allows you to rotate and pan the viewpoint, to zoom in and out, and has many other useful display features (feel free to experiment) such as node and element numbering, nodal coordinates, seabed topography, water surface profile etc. Once the geometric specification has been completed, the Model View provides a preview of the hybrid riser system, as shown in the figure below. Refer to Model View for further information on this feature.



Model View

Structural and Hydrodynamic Properties

Flexcom uses the concept of element sets to define the physical properties of the finite element model. Groups of similar elements are logically combined into named element sets, properties are then assigned on an element set-by-set basis. When lines are utilised during model creation, relevant element sets are created automatically, based on the names of each line (and subsection of each line if appropriate) in the model. From the line definitions detailed previously, the relevant element sets in this context are Outer Pipe, Inner Pipe, Oil Jumper, Gas Jumper, Lower Frame, Upper Frame, Buoyancy Tank, T-Frame-Leg and T-Frame-Top. Sets are also created for the Lower Flex Joint and the Upper Flex Joint, although the assignment of properties to these is slightly different to the other sets.

Flexcom provides two different formats for specifying structure geometric properties, although ultimately the information is the same and is used in the same way by the program. These are termed the Flexible Riser Format and the Rigid Riser Format. The Flexible Riser format is so called, naturally, because it is the format commonly used when defining flexible risers. Specifically, the data to be specified in this case comprises bending stiffnesses about two axes, torsional stiffness, axial stiffness, mass per unit length etc. When you use the Rigid Riser format on the other hand, you input internal and external diameters, Young’s modulus, shear modulus, mass density etc. – data for the analysis of a rigid riser (whether a top tensioned riser or an SCR) would normally be available in this format.

Both the inner and outer pipes are made of steel, and their structural properties are specified in the Rigid Riser format, as shown in the figure below. Refer to Geometric Properties in Rigid Riser Format for further information on this feature.

Rigid Riser Geometric Properties

Set Name	E	G	Do	Di	rho	A	I	J	Dd	Db	Dc	Buoyancy
Inner Pipe	2.07E+11	8.00E+10	0.296	0.254	7850							[Default]
Outer Pipe	2.07E+11	8.00E+10	0.364	0.316	7850				0.524	0.524		[Default]

```
*GEOMETRIC SETS
 OPTION=RIGID
 SET=Inner Pipe
 2.07E+11, 8.00E+10, 0.296, 0.254, 7850
 OPTION=RIGID
 SET=Outer Pipe
 2.07E+11, 8.00E+10, 0.364, 0.316, 7850, , , , 0.524, 0.524
```

Structural Properties – Rigid Riser Format

Refer to the *GEOMETRIC SETS keyword for further information on these data inputs.

The outer pipe is also coated with insulation material, as shown in the figure below. Refer to Coatings for further information on this feature.

Coatings

Set Name	Type	Name	Thickness	Mass Density
Outer Pipe	External	Insulation	0.08	900

```
*COATINGS
 SET=Outer Pipe
 TYPE=EXTERNAL
 NAME=Insulation, THICKNESS=0.08, DENSITY=900
```

Application of Coatings

Refer to the *COATINGS keyword for further information on these data inputs.

Structural properties for the other element sets are specified in the Flexible Riser format, as shown in the figure below. Note that that symbol ':' indicates that the following keyword data is supplementary to the existing data under the same keyword as shown previously. Refer to Geometric Properties in Flexible Riser Format for further information on this feature.

Flexible Riser Geometric Properties

Set Name	Bending	EIyy	EIzz	GJ	EA	m	p	Di	Dd	Db	Do	Dc	Buoyancy
Oil Jumper	[Linear]	350000	350000	300000	7.00E+08	210	1	0.254	0.37	0.37			Default
Gas Jumper	[Linear]	33000	33000	45000	2.50E+08	70	1	0.08	0.21	0.21			Default
Lower Frame	[Linear]	1.00E+10	1.00E+10	1.00E+10	1.00E+10	3889	1	0.08	1.18	1.18			[Default]
Upper Frame	[Linear]	1.00E+10	1.00E+10	1.00E+10	1.00E+10	3889	1	0.08	1.18	1.18			[Default]
T-frame-Leg	[Linear]	1.00E+10	1.00E+10	1.00E+10	1.00E+10	1	1	0.08	0.2	0.2			Default
T-frame-Top	[Linear]	1.00E+10	1.00E+10	1.00E+10	1.00E+10	1	1	0.08	0.2	0.2			Default
Buoyancy Tank	[Linear]	1.00E+10	1.00E+10	1.00E+10	1.00E+10	9000	1	0.08	6	6			Default

```
*GEOMETRIC SETS
 :
 OPTION=FLEXIBLE
 SET=Oil Jumper,
 350000, 350000, 300000, 7.00E+08, 210, 1, 0.254, 0.37, 0.37
 OPTION=FLEXIBLE
 SET=Gas Jumper,
 33000, 33000, 45000, 2.50E+08, 70, 1, 0.08, 0.21, 0.21
 OPTION=FLEXIBLE
 SET=Lower Frame
 1.00E+10, 1.00E+10, 1.00E+10, 1.00E+10, 3889, 1, 0.08, 1.18, 1.18
 OPTION=FLEXIBLE
 SET=Upper Frame
 1.00E+10, 1.00E+10, 1.00E+10, 1.00E+10, 3889, 1, 0.08, 1.18, 1.18
 OPTION=FLEXIBLE
 SET=T-frame-Leg,
 1.00E+10, 1.00E+10, 1.00E+10, 1.00E+10, 1, 1, 0.08, 0.2, 0.2
 OPTION=FLEXIBLE
 SET=T-frame-Top,
 1.00E+10, 1.00E+10, 1.00E+10, 1.00E+10, 1, 1, 0.08, 0.2, 0.2
 OPTION=FLEXIBLE
 SET=Buoyancy Tank,
 1.00E+10, 1.00E+10, 1.00E+10, 1.00E+10, 9000, 1, 0.08, 6, 6
```

Structural Properties – Flexible Riser Format

Refer to the `*GEOMETRIC SETS` keyword for further information on these data inputs.

The structural properties of a flex joint are characterised in terms of a rotational stiffness, and it may optionally have associated weights in air and in water. The upper and lower flex joints for this example are assigned properties as shown in the figure below. Refer to Hinges and Flex Joints for further information on this feature.

Flex Joints

Element Set/Number	Weight in Water	Weight in Air	Rotational Stiffness
Lower Flex Joint	0.0	0.0	20000
Upper Flex Joint	0.0	0.0	20000

```
*FLEX JOINT
 SET=Lower Flex Joint
 0.0, 0.0
 TYPE=LINEAR
 20000
 SET=Upper Flex Joint
 0.0, 0.0
 TYPE=LINEAR
 20000
```

Flex Joint Properties

Refer to the *FLEX JOINT keyword for further information on these data inputs.

The simplest hydrodynamic property specification in Flexcom is that of constant coefficients, which remain the same throughout an analysis. Five hydrodynamic coefficients are specified on an element set by set basis. For simplicity, all elements in this example are assigned the same hydrodynamic coefficients as shown in the figure below. Refer to Constant Hydrodynamic Properties for further information on this feature.

Hydrodynamic Properties

Set Name	Normal Drag	Tangential Drag	Normal Inertia	Tangential Added Mass	Normal Added Mass	Drag Lift
All	1.2	0.0	2.0	0.0		

```
*HYDRODYNAMIC SETS
 SET=All, TYPE=CONSTANT
 1.2, 0.0, 2.0, 0.0
```

Hydrodynamic Properties

Refer to the *HYDRODYNAMIC SETS keyword for further information on these data inputs.

Environment and Loading

Flexcom traditionally groups together four parameters required to provide the minimum environment specification, namely the analysis water depth, the density of seawater ρ_w, the gravitational constant g, and ν, the kinematic viscosity of seawater. The environmental parameters for this example are shown in the figure below. Refer to Environmental Parameters for further information on this feature.

Ocean - Properties	
Water Depth	700
Water Density	1025
Acceleration due to Gravity	9.81
Water Kinematic Viscosity	1.3E-3

```
*OCEAN
 700, 1025, 9.81
```

Environmental Parameters

Refer to the *OCEAN keyword for further information on these data inputs.

Flexcom provides a comprehensive internal fluid modelling capability. Stationary internal fluids, uniform steady state internal fluid flow and multi-phase slug flow may all be modelled. A stationary internal fluid in Flexcom is characterised by its mass density, pressure above hydrostatic and the level above mudline to which it extends. The level above mudline input is used to determine whether elements are fully filled with fluid, partially filled or empty. In this example, both the inner pipe and the oil jumper are oil filled, while the annulus between the inner and outer pipes, and the gas jumper are gas filled, as shown in the figure below. Notice that internal fluid data is specified in a newly created $LOAD CASE section of the file. Refer to Internal Fluid for further information on this feature.

MODEL | LOAD CASE

Internal Fluid

Set Name	Level Al	Mass D	Internal	Velocity	Coriolis	Centrifu	Axial In	Lateral I	Dynami
Inner Pipe	615	800	[0.0]	[0.0]	[Include]	[Include]	[Include]	[Include]	[Include]
Outer Pipe	615	200	[0.0]	[0.0]	[Include]	[Include]	[Include]	[Include]	[Include]
Oil Jumper	695	800	[0.0]	[0.0]	[Include]	[Include]	[Include]	[Include]	[Include]
Gas Jumper	695	200	[0.0]	[0.0]	[Include]	[Include]	[Include]	[Include]	[Include]

```
$LOAD CASE
*INTERNAL FLUID
 SET=Inner Pipe
 615, 800
 SET=Outer Pipe
 615, 200
 SET=Oil Jumper
 695, 800
 SET=Gas Jumper
 695, 200
```

Internal Fluid

Refer to the *INTERNAL FLUID keyword for further information on these data inputs.

Vessel and RAO Data

Three sets of inputs are required to completely define the response of a floating vessel to the ambient wave field. The first set defines the vessel initial position, that is, the location of the vessel prior to the application of any vessel offset or dynamic motions. This information is defined in terms of the location of the vessel reference point (the point at which the RAOs are defined), and initial yaw orientation of the vessel. Refer to Basic Vessel Concepts for further information on this feature.

The second set of inputs is used to specify the offset of the vessel from its initial position and orientation. These entries are optional and are only specified if the vessel is offset from its initial location and orientation before the commencement of the dynamic analysis. Flexcom offers a full 6 degree of freedom offset capability, comprising three translations and three rotations. This facility will be illustrated later, in Vessel Offset Analyses. Refer to Vessel Offsets for further information on this feature.

The third set of inputs is the full vessel RAOs and phase angles in all vessel degrees of freedom. Given the vessel initial position, offset and RAOs, in addition to the time history of wave elevation, the motion of the reference point in response to the wave field throughout an analysis can be calculated, and from this the motion of the point(s) on the structure attached to the vessel. As the RAO data may be quite extensive, it is typically stored in a separate file which is simply referenced by the main keyword file. The file with RAO data already exists in the Worked example folder, and is named RAO.incx. This should be copied into the folder for this project. Refer to High Frequency RAO Motions for further information on this feature.

You may also optionally associate a structural profile with the vessel. While does not affect analysis results, it does add considerably to the visual appeal of the structural animation. A range of standard vessel profiles are provided; the FPSO profile being particularly suited to this worked example.

The completed specification of vessel data for the FPSO in this example is specified as shown in the figure below. Note that all vessel data (apart from vessel offsets) is specified in the `$MODEL` section of the file.

Vessel Integrated - Setup

Vessel	Focal Point X	Focal Point Y	Focal Point Z	Yaw	RAO File	Format	Units	Ref Pt Offset X	Ref Pt Offset Y	Ref Pt Offset Z	Profile	Height	Length	Width
FPSO	700	220	0	90	RAO.incx			[0.0]	[0.0]	[0.0]	FPSO	100	280	50

```
*VESSEL,INTEGRATED
 VESSEL=FPSO
 INITIAL POSITION=700, 220, 0, 90
 RAO=RAO.incx
 PROFILE=FPSO
 DIMENSIONS=100, 280, 50
```

Vessel Data

Refer to the `*VESSEL, INTEGRATED` keyword for further information on these data inputs.

Boundary Conditions

Flexcom presents a comprehensive range of options to fully describe the constraints which are applied to the finite element model. The various options include constant (time invariant constraints), vessel (to apply vessel motions), and several others including an arbitrary user-subroutine facility which provides for complete generality in terms of constraint application. The first two options are invoked in this example, with the base of the central structure fixed in all degrees of freedom, and the upper ends of the jumper hoses attached to the FPSO, as shown in the first and second figures below, respectively. Note that the relevant node labels (i.e. Seabed End, Oil Jumper FPSO and Gas Jumper FPSO), created automatically by the lines facility during model building, are referenced when defining boundary conditions. The specification of boundary condition data goes into the `$LOAD CASE` section of the file. Refer to Boundary Conditions for further information on this feature.

Boundary - Constant - Direct

Node	DOF	Displacement	Fixation
{Seabed End}	1	0.0	[Absolute]
{Seabed End}	2	0.0	[Absolute]
{Seabed End}	3	0.0	[Absolute]
{Seabed End}	4	0.0	[Absolute]
{Seabed End}	5	0.0	[Absolute]
{Seabed End}	6	0.0	[Absolute]

```
*BOUNDARY
 TYPE=CONSTANT
 {Seabed End}, 1, 0.0
 {Seabed End}, 2, 0.0
 {Seabed End}, 3, 0.0
 {Seabed End}, 4, 0.0
 {Seabed End}, 5, 0.0
 {Seabed End}, 6, 0.0
```

Constant Boundary Conditions

Note that that symbol ⋮ indicates that the following keyword data is supplementary to the existing data under the same keyword as shown previously.

Boundary - Vessel

Vessel	Node	DOF	Displacement
FPSO	{Oil Jumper FPSO}	1	0.0
	{Oil Jumper FPSO}	2	0.0
	{Oil Jumper FPSO}	3	0.0
	{Gas Jumper FPSO}	1	0.0
	{Gas Jumper FPSO}	2	0.0
	{Gas Jumper FPSO}	3	0.0

```
*BOUNDARY
 ⋮
 TYPE=VESSEL, VESSEL=FPSO
 {Oil Jumper FPSO}, 1, 0.0
 {Oil Jumper FPSO}, 2, 0.0
 {Oil Jumper FPSO}, 3, 0.0
 {Gas Jumper FPSO}, 1, 0.0
 {Gas Jumper FPSO}, 2, 0.0
 {Gas Jumper FPSO}, 3, 0.0
```

Vessel Boundary Conditions

Refer to the *BOUNDARY keyword for further information on these data inputs.

Initial Static Analysis

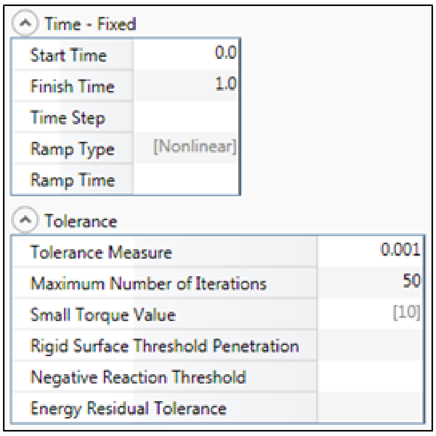
OVERVIEW OF RESTART FACILITY

The restart facility allows you to specify that a particular analysis is to be restarted from a previous run, to build up to the full dynamic solution in stages. In a restart, the structure configuration at the end of the preceding analysis becomes the starting configuration for the restart. Refer to Restart Analyses for further information on this feature.

SIGNIFICANCE OF TIME VARIABLES IN STATIC ANALYSIS

Naturally, all time domain analyses require the specification of time variables. Since a static analysis is one which considers time invariant loading and structure response, time variables are effectively meaningless in this context. However you are also required to specify time variables for a static analysis in Flexcom. This is purely a consequence of the historical evolution of the software. Refer to Time Variables in Static Analysis for further information on this feature.

An initial static analysis is typically run from t=0 to t=1 second using a single fixed time step, and this standard approach is adopted for this example, as shown in the figure below. Note also that the maximum number of iterations is increased to 50 to aid convergence in the initial static analysis.



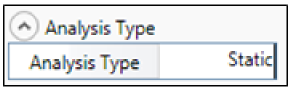

```
*TIME
 STEP=FIXED
 0.0, 1.0
*TOLERANCE
 ANALYSIS=STATIC/TIME
 0.001, 50
```

Initial Static Analysis Time Variables

Refer to the *TIME keyword for further information on these data inputs.

ANALYSIS TYPE

The type of the analysis to be performed must be specified in each $LOAD CASE section. In this case, a static analysis is required, as shown in the figure below.




```
*ANALYSIS TYPE
 TYPE=STATIC
```

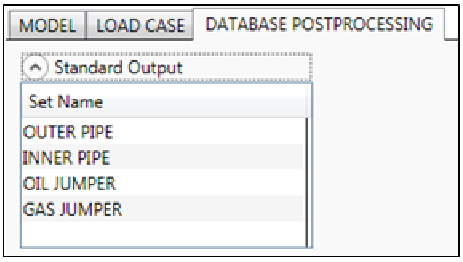
Analysis Type

Refer to the `*ANALYSIS TYPE` keyword for further information on these data inputs.

DATABASE POSTPROCESSING

Database Postprocessing is the most powerful and widely used postprocessing facility in Flexcom. The database files provide a very detailed picture of the structure response. For large models, or long simulations with a large number of database outputs, the database files can become quite large. However, this is not a concern for static analyses, or indeed the dynamic analysis studied in this example. Refer to Database Postprocessing for further information on this feature.

In this example, the postprocessing requests are specified before the analysis begins, via the standard output option. This option allows you to quickly request a summary of pertinent information, without the inconvenience of explicitly requesting specific outputs. For every element set referenced, Flexcom produces outputs of effective tension, resultant bending moment and von Mises stress for that set.




```
$DATABASE POSTPROCESSING
*STANDARD OUTPUT
 SET=OUTER PIPE
 SET=INNER PIPE
 SET=OIL JUMPER
 SET=GAS JUMPER
```

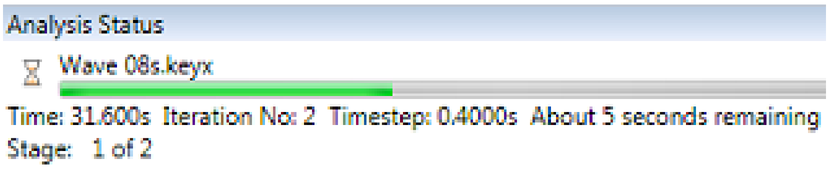
Database Postprocessing Requests

Refer to the `*STANDARD OUTPUT` keyword for further information on these data inputs.

RUNNING THE ANALYSIS

Specification of the model geometry, structural and hydrodynamic properties, environment and loading data, vessel and RAO data, boundary conditions and solution parameters is now complete, so you should save the keyword file and run the analysis.

When a run is in progress, you can monitor its status via the Analysis Status View. The progress bar is naturally most beneficial for longer dynamic analyses (e.g. where it provides an approximate estimate of remaining CPU time), and sample progress information is shown in the figure below.

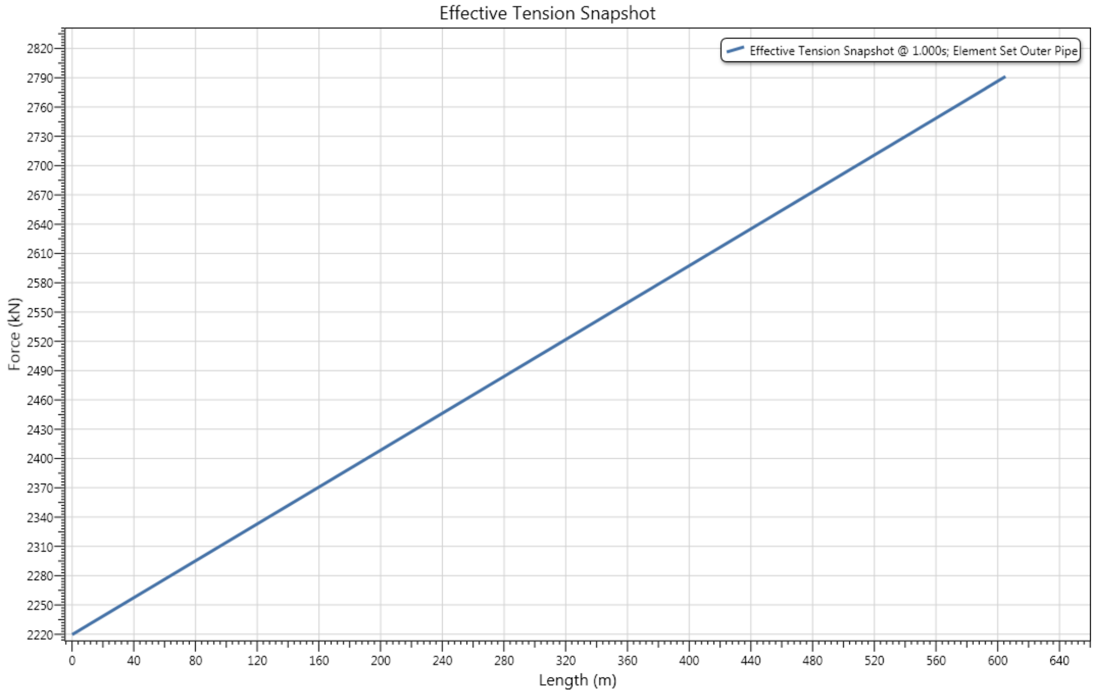


Analysis Progress Indicator

Another useful feature in the File View is that the program provides an indication of the status of the analysis. When the run has finished, the icon associated with the keyword file changes to , indicating that the analysis has completed successfully. You may have noticed the icon next to the keyword file name was previously , indicating the analysis had not yet been run. As the computation time for the static analysis is relatively short, you probably won't have noticed that the icon briefly changed to , indicating that the analysis was currently in progress.

EXAMINING RESULTS

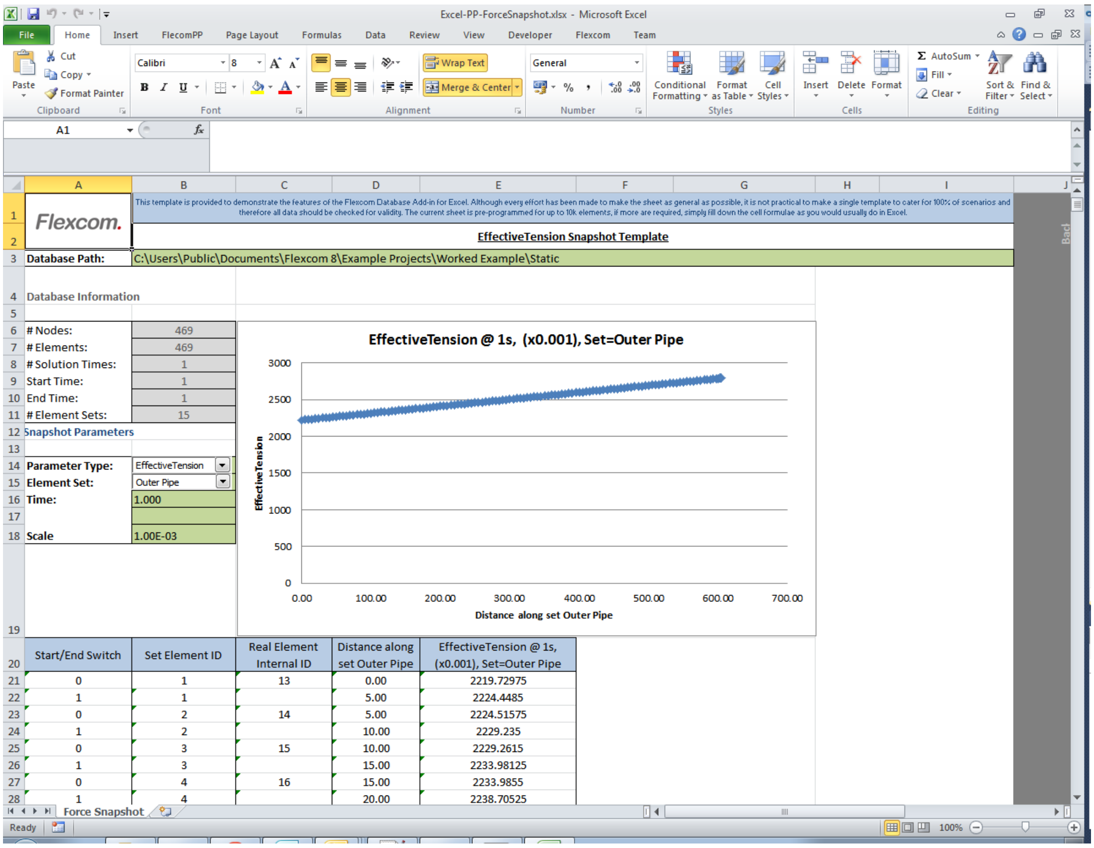
You can view the analysis results using the Plotting facility. Plot files have the file extension .MPLT (an abbreviation of Mcs PLoT). The figure below shows the static effective tension distribution in the outer pipe (this corresponds to the file Static.S1.mplt, or similar, depending on your naming convention). You can examine the other plots created for the static analysis at your convenience.



Static Effective Tension Distribution in Outer Pipe

Refer to Plotting for further information on this feature.

In addition to the standard postprocessing facilities available within Flexcom, the software is also accompanied by a dedicated Excel Add-In. This feature allows you to extract results from Flexcom database files directly into Excel. Its operation should be fairly self-explanatory, but sample spreadsheets are provided with the worked example for illustration purposes. Specifically, distributions of effective tension and bending moment are presented for the initial static analysis, while time dependent data is presented for one of the dynamic load cases.



Excel Add-In

Refer to Excel Add-in for further information on this feature.

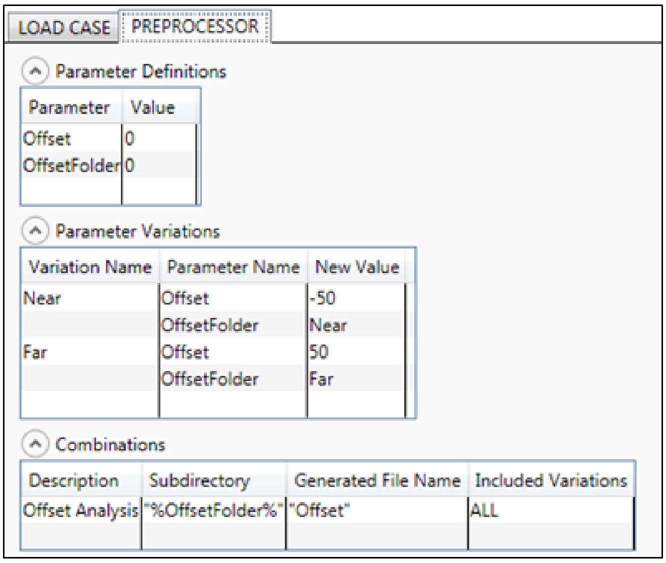
Vessel Offset Analyses

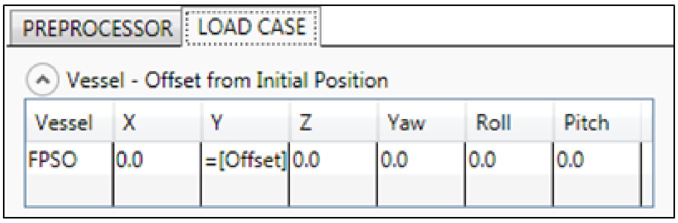
OVERVIEW OF KEYWORD PARAMETERISATION FACILITY

The keyword parameterisation facility means that a single keyword file is all that is required to model a series of (typically load case) variations about a base model. Flexcom will then automatically create all the keyword files required to examine each load case. For example, this is very useful if you have performed an initial static analysis, and you wish to examine the structural response to a series of regular wave load cases, each having a unique wave period. Refer to Keyword Parameterisation for further information on this feature.

NEAR AND FAR OFFSETS

For the purposes of this example, we will examine two vessel offset cases, near and far, at ±50m, though in practice several more might also be examined. In earlier versions of Flexcom, this would require the creation of separate keyword files for each offset, but the keyword parameterisation facility means that a single keyword file is all that is required to model each load case. So you include all the required data in a single base keyword file, restarting from the initial static analysis keyword file. Create a new keyword file based on the Empty Loadcase template, named Offset.keyx, (or similar, depending on your naming convention) and add the relevant input data as shown in the figure below. Note that that '↵' symbol indicates that the keyword data continues on the same line in the actual keyword file, even though it is necessary to wrap the text in this document due to formatting constraints.





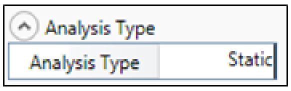

```
$PREPROCESSOR
*PARAMETERS
 Offset, -50
 OffsetFolder, "Near"
*VARIATION
 NAME=Near
 PARA=Offset, VALUE=-50
 PARA=OffsetFolder, VALUE=Near
 NAME=Far
 PARA=Offset, VALUE=50
 PARA=OffsetFolder, VALUE=Far
*COMBINATIONS
 DESCRIPTION=Offset Analysis, SUBDIRECTORY="%OffsetFolder%", ↵
FILENAME="Offset"
 VARIATIONS=ALL

$LOAD CASE
*OFFSET
 VESSEL=FPSO
 OFFSET=0.0, =[Offset], 0.0, 0.0, 0.0, 0.0
```

Near and Far Vessel Offset Variations

Refer to the *PARAMETERS, *VARIATION and *COMBINATIONS keywords for further information on these data inputs.

ANALYSIS TYPE

The type of the analysis to be performed must be specified in each $LOAD CASE section. In this case, a static analysis is required, as shown in the figure below.



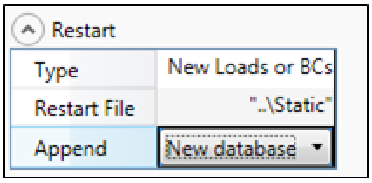

```
*ANALYSIS TYPE
 TYPE=STATIC
```

Analysis Type

Refer to the *ANALYSIS TYPE keyword for further information on these data inputs.

RESTART FACILITY

As mentioned already, the restart facility allows you to specify that a particular analysis is to be restarted from a previous run, to build up to the full dynamic solution in stages. In a restart, the structure configuration at the end of the preceding analysis becomes the starting configuration for the restart. Refer to Restart Analyses for further information on this feature. Here the offset analyses restart from the initial static analysis, as shown in the figure below.




```
*RESTART
 LAST="..\Static"
```

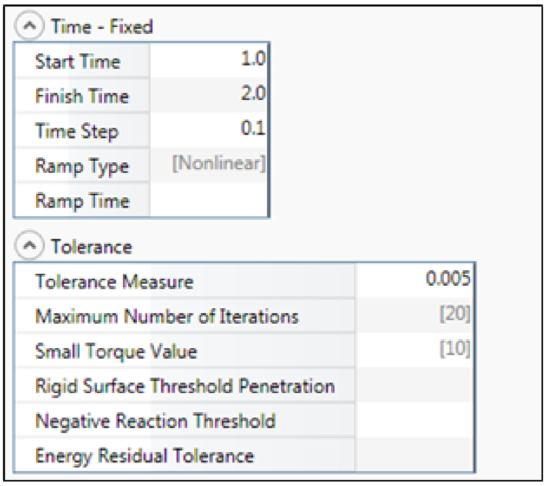
Restart Variables for Vessel Offset Analyses

Refer to the *RESTART keyword for further information on these data inputs.

OTHER SOLUTION VARIABLES

For the offset analyses, the start and end times are set equal to 1s and 2s, respectively, consistent with the end time of 1s chosen for the initial static analysis. A fixed time step of 0.1s is used to apply to offset over 10 solution increments. Refer to Time Variables in Static Analysis for further information on this feature.

Note also that the solution tolerance is relaxed slightly to aid convergence in the static offset analyses. Refer to Solution Convergence for further information on this feature.




```
*TIME
 STEP=FIXED
 1.0, 2.0, 0.1
*TOLERANCE
 ANALYSIS=STATIC/TIME
 0.005
```

Time Variables for Vessel Offset Analyses

Refer to the *TIME and *TOLERANCE keywords for further information on these data inputs.

RUNNING THE ANALYSIS

When a parameterised keyword file is “run”, Flexcom does not do any analysis per se, but rather generates other keyword files, which represent a variation about the base file.

Once you run the keyword file (Offset.keyx, or similar, depending on your naming convention), Flexcom will automatically create a separate keyword file corresponding to each vessel offset, and the *PARAMETERS keyword contains the relevant offset value. The names of the files are controlled by the *COMBINATIONS keyword. For the data shown in the Near and Far Vessel Offset Variations figure, you will obtain two separate keyword files, both entitled Offset.keyx. The newly created files are located in the two subdirectories, Near and Far.

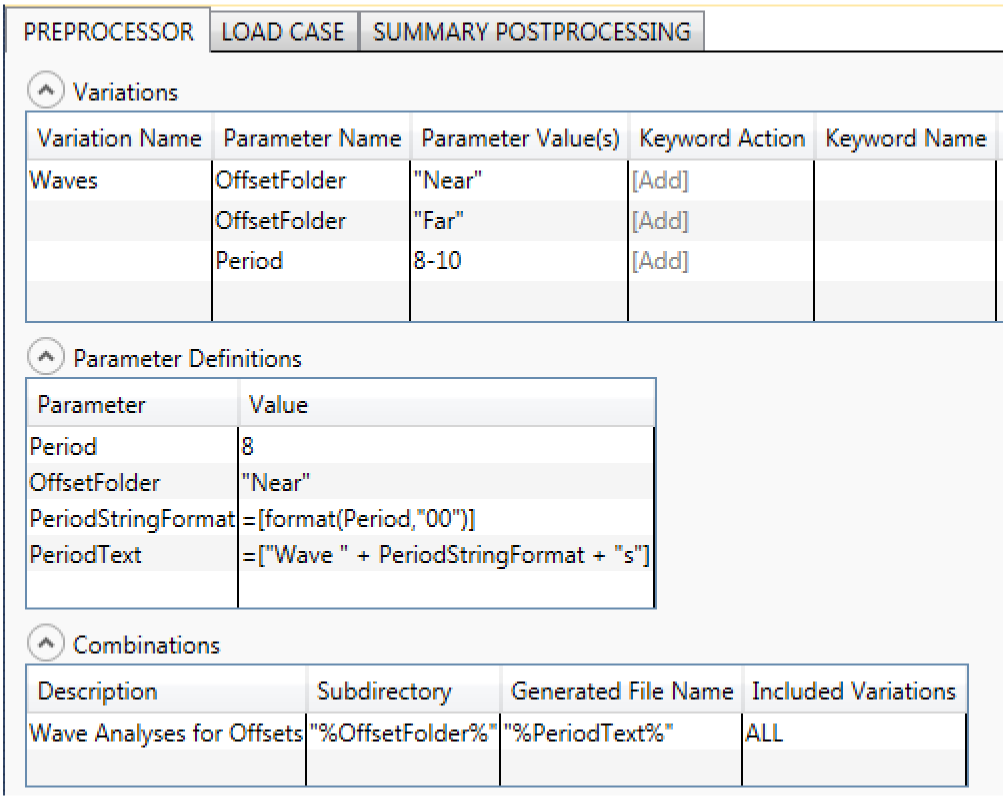
In order to perform the actual analyses themselves, you should run each keyword file individually. Alternatively, you can simply invoke the Run Branch option from the initial static keyword file, Static.keyx.

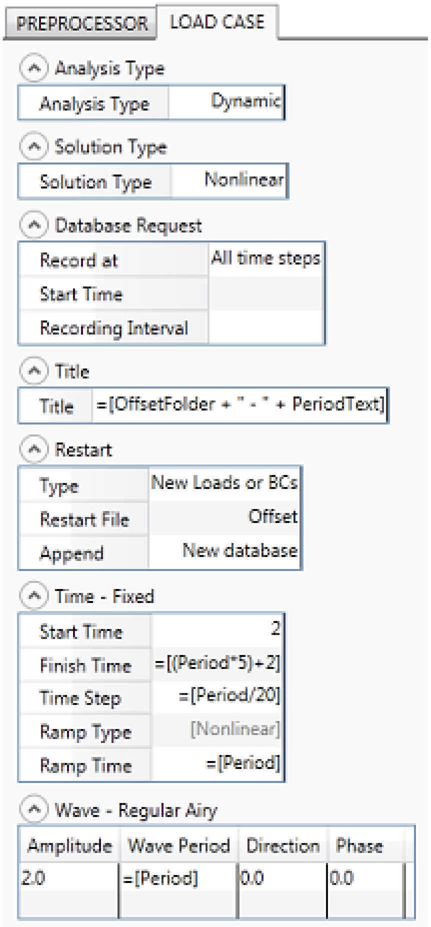
Regular Wave Analyses

LOAD CASE MATRIX

For the purposes of this example, we will examine three regular wave load cases, corresponding to wave periods of 8s, 9s, and 10s, respectively, though in practice many more would also be examined. Combined with the near and far vessel offset case, this will result in 6 regular wave keyword files in total. Using the keyword parameterisation facility means that a single keyword file is all that is required to model each combination, and although the load case matrix is relatively small in this instance, the benefits are obvious. Refer to Keyword Parameterisation for further information on this feature.

Create a new blank keyword file named Regular Wave.keyx, (or similar, depending on your naming convention) and include all the required data in this single base keyword file. The relevant input data is shown in the figure below. Note that that '↵' symbol indicates that the keyword data continues on the same line in the actual keyword file, even though it is necessary to show some data on a subsequent line here due to formatting constraints in this document.




PREPROCESSOR
LOAD CASE
Analysis Type
Analysis Type
Dynamic
Solution Type
Solution Type
Nonlinear
Database Request
Record at
All time steps
Start Time
Recording Interval
Title
Title
=[OffsetFolder + " - " + PeriodText]
Restart
Type
New Loads or BCs
Restart File
Offset
Append
New database
Time - Fixed
Start Time
2
Finish Time
=[(Period*5)+2]
Time Step
=[Period/20]
Ramp Type
[Nonlinear]
Ramp Time
=[Period]
Wave - Regular Airy
Amplitude
Wave Period
Direction
Phase
2.0
=[Period]
0.0
0.0

```
$PREPROCESSOR
*PARAMETERS
 Period, 8
 OffsetFolder, "Near"
 PeriodStringFormat, =[format(Period,"00")]
 PeriodText,   =["Wave " + PeriodStringFormat + "s"]
*VARIATION
 NAME=Waves
 PARA=OffsetFolder, VALUE="Near"
 PARA=OffsetFolder, VALUE="Far"
 PARA=Period, GEN=8, 10
*COMBINATIONS
 DESCRIPTION=Wave Analyses for Offsets, SUBDIRECTORY="%OffsetFolder%",↵
FILENAME="%PeriodText%"
 VARIATIONS=ALL

$LOAD CASE
*NAME
 =[OffsetFolder + " - " + PeriodText]
*ANALYSIS TYPE
 TYPE=DYNAMIC, SOLUTION=NONLINEAR
*WAVE-REGULAR
 2.0, =[Period], 0.0, 0.0
*TIME
 STEP=FIXED
 2, =[(Period*5)+2], =[Period/20], =[Period]
*RESTART
 LAST=Offset
*DATABASE
 TIME=ALL
```

Regular Wave Load Case Variations

A number of points are noteworthy:

- The load case matrix is automatically determined by Flexcom. In this case, there are two parameters changing, 3 different wave periods combined with 2 different offset directories, resulting in a load case matrix totalling 6 separate dynamic analyses. Refer to the `*PARAMETERS` and `*VARIATION` keywords for further information on these data inputs.
- The names of the files are based on the period parameter, such as Wave 08s.keyx, and so on. All the newly created files are located in either the Near or the Far directory. Refer to the `*COMBINATIONS` keyword for further information on these data inputs.
- Each dynamic analysis uses an appropriate regular wave period. A unit wave amplitude is used in all cases. Refer to the `*WAVE-REGULAR` keyword for further information on these data inputs.

- Appropriate time variables are computed based on the regular wave period. Each dynamic analysis runs for 5 wave periods, with the loads ramped on over the first wave period. A fixed time step equal to 1/20th of the regular wave period is used. Refer to the *TIME keyword for further information on these data inputs.

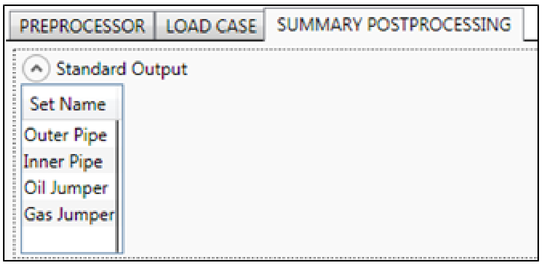
- The Analysis Type is set to Dynamic. Refer to the *ANALYSIS TYPE keyword for further information on these data inputs.

- Full database storage is requested for the regular wave analyses. Refer to the *DATABASE keyword for further information on these data inputs.

SUMMARY POSTPROCESSING

Flexcom provides a powerful facility for generating a Summary Output File. Here the maximum value, the minimum value, the range of values and the standard deviation of values is tabulated for a group of parameters you specify. The output is in a succinct format and could, for example, be pasted or inserted into a study report, or perhaps included as a report appendix. Refer to Summary Postprocessing for further information on this feature.

In this example, the postprocessing requests are specified before the analysis begins, via the standard output option. This option allows you to quickly request a summary of pertinent information, without the inconvenience of explicitly requesting specific outputs. For every element set referenced, Flexcom produces outputs of effective tension, resultant bending moment and von Mises stress for that set.




```
$SUMMARY POSTPROCESSING
*STANDARD OUTPUT
 SET=Outer Pipe
 SET=Inner Pipe
 SET=Oil Jumper
 SET=Gas Jumper
```

Summary Postprocessing Requests

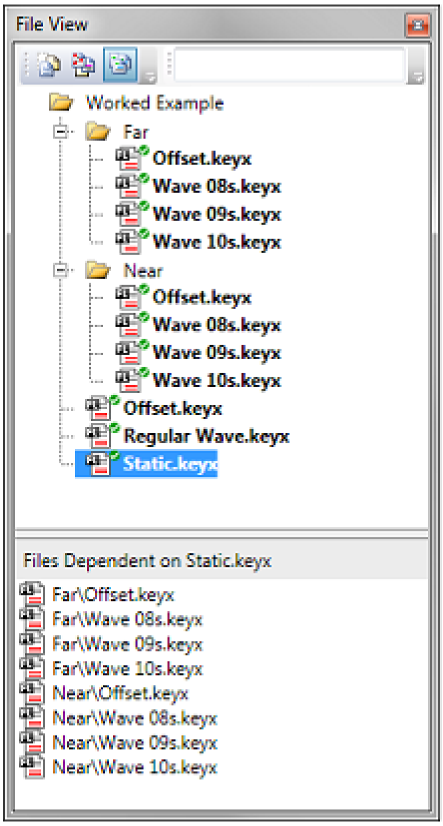
Refer to the *STANDARD OUTPUT keyword for further information on these data inputs.

RUNNING THE ANALYSIS

Once the specification of the regular wave data is complete, you should save the keyword file. Once you run the keyword file (Regular Wave.keyx, or similar, depending on your naming convention), Flexcom will automatically create a separate keyword file for you, corresponding to each combination of regular wave and preceding vessel offset directory. In order to perform the actual analyses themselves, you could run each analysis individually. However, a better approach is probably to invoke the Run Branch option from the vessel offset analyses.

KEYWORD FILE INTERDEPENDENCIES

Consistent with the concept of a highly project-focused integrated engineering environment introduced in Project Structure earlier, the keyword file interdependencies are readily evident at a glance in the File View. Specifically, at the top level lies the initial static analysis, which represents the starting point for all subsequent restart analyses. At the next level come the near and far vessel offset analyses, which restart directly from the initial static configuration. Next are the dynamic regular wave analyses, which are linked to the relevant offset analysis (i.e. near or far). So the entire keyword file hierarchy is clearly evident from the File View, as illustrated in the figure below.



Keyword File Interdependencies

EXAMINING RESULTS

Unlike the database postprocessing performed for the initial static analysis, no plot files have been requested or created in this instance. Instead, each regular wave analysis produces a tabular summary file (with the extension SUM). The figure below shows some sample summary output, corresponding to the 8s regular wave analysis, for the near vessel offset case (this corresponds to the file Wave 08s.sum, or similar, depending on your naming convention). You can examine the other tabular summary files created for the other load cases at your convenience.

```
                         F L E X C O M   Version 8.2.1

                      Time Domain Finite Element Analysis

                             (c) MCS Kenny 2012

Near - Wave 08s

Summary of results from analysis:  Wave 08s

Variable                                  Minimum     Maximum     Range     Standard     Scale
                                                                            Deviation

(1) Force Envelopes
-------------------
Effective Tension (Outer Pipe)           2203.490    2804.408    600.918       N/A     0.100E-02
Resultant Moment  (Outer Pipe)              0.395     179.251    178.856       N/A     0.100E-02
Von Mises Stress  (Outer Pipe)             95.458     177.374     81.916       N/A     0.100E-05
Effective Tension (Inner Pipe)            537.318    1643.397   1106.079       N/A     0.100E-02
Resultant Moment  (Inner Pipe)              0.026     168.783    168.756       N/A     0.100E-02
Von Mises Stress  (Inner Pipe)             44.051     177.068    133.017       N/A     0.100E-05
Effective Tension (Oil Jumper)             48.476     300.258    251.782       N/A     0.100E-02
Resultant Moment  (Oil Jumper)              0.000      10.097     10.097       N/A     0.100E-02
Von Mises Stress  (Oil Jumper)              2.403       5.310      2.907       N/A     0.100E-05
Effective Tension (Gas Jumper)             12.464      75.969     63.504       N/A     0.100E-02
Resultant Moment  (Gas Jumper)              0.000       0.958      0.958       N/A     0.100E-02
Von Mises Stress  (Gas Jumper)              2.134       3.178      1.043       N/A     0.100E-05

Notes:
------
(1) Parameters calculated over time interval      26.000 to      42.000
```

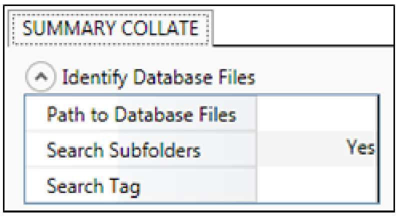
Sample Summary Output File

Results Collation

If you are performing a series of analyses (for example to examine a large number of different load cases), the Summary Postprocessing Collation facility provides a useful means of assembling all the pertinent summary output data (i.e. based on individual SUM files) across a range of load cases into a single document. Refer to Summary Postprocessing

Collation for further information on this feature.

Create a new blank keyword file named Collate.keyx (or similar, depending on your naming convention). For simplicity, we will collate all the available data from the regular wave analyses, as shown in the figure below (note, if you do not specify any descriptive titles for collation purposes, Flexcom will attempt to collate all available data).




```
$SUMMARY COLLATE
*IDENTIFY
 OPTION=SUBFOLDERS
```

Summary Collation Requests

Refer to the `*IDENTIFY` keyword for further information on these data inputs.

Once the specification of the summary collation parameters is complete, you should save the keyword file, and begin the postprocessing run.

The summary collation process produces a tabular summary file (with the extension OUT) and spreadsheet based output (with the extension XLSX). A sample extract from Collate.xlsx is shown in the figure below.

Flexcom.

Force Envelopes - Effective Tension (Outer Pipe)

Analysis Title	Minimum	Maximum	Time of Minimum Occurrence	Time of Maximum Occurrence	Mean	Range
Far - Wave 08s	2222.439	2794.918	26.4	30.4	N/A	572.478
Far - Wave 09s	2194.693	2821.242	47	42.5	N/A	626.549
Far - Wave 10s	2194.443	2822.61	32	47	N/A	628.167
Near - Wave 08s	2203.49	2804.408	40.8	28.8	N/A	600.918
Near - Wave 09s	2189.664	2817.078	45.65	41.15	N/A	627.414
Near - Wave 10s	2188.603	2819.103	40.5	46	N/A	630.5

Force Envelopes - Resultant Moment (Outer Pipe)

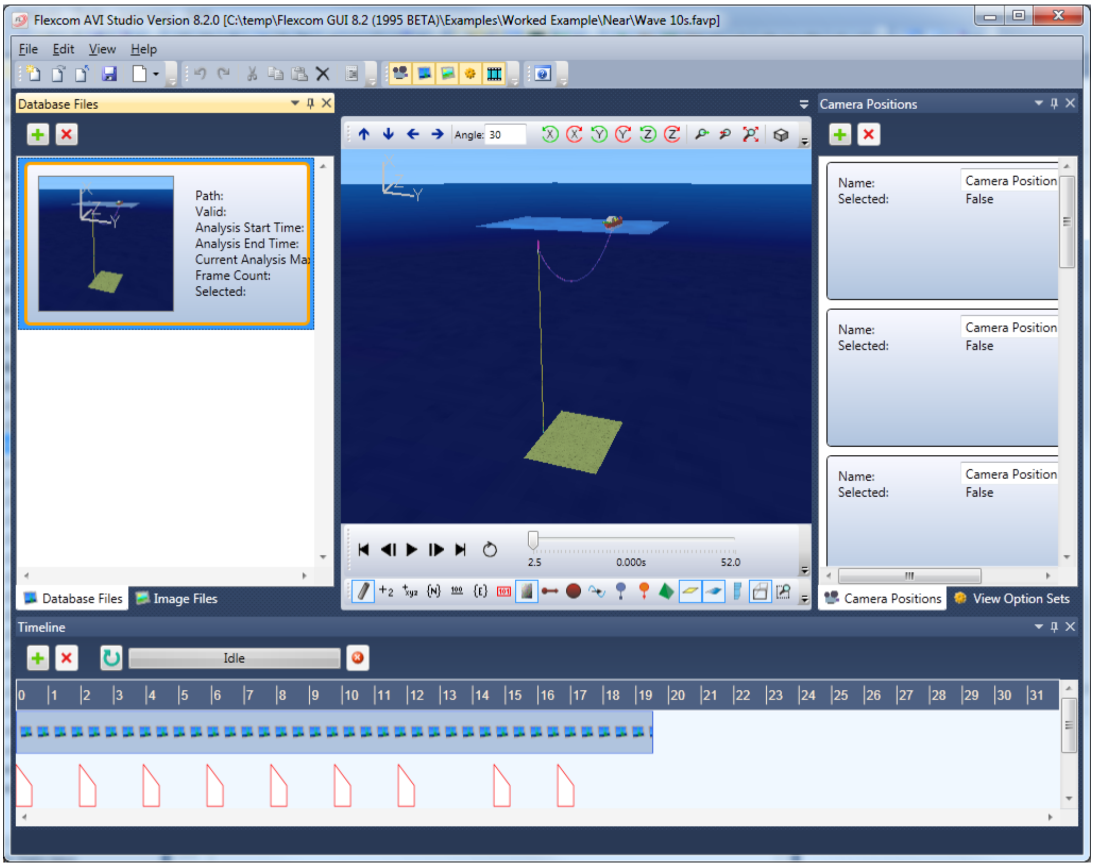
Analysis Title	Minimum	Maximum	Time of Minimum Occurrence	Time of Maximum Occurrence	Mean	Range
Far - Wave 08s	1.076	338.789	29.2	39.6	N/A	337.713
Far - Wave 09s	0.002	363.129	39.35	38.45	N/A	363.127
Far - Wave 10s	0.002	367.74	41.5	40.5	N/A	367.737
Near - Wave 08s	0.395	179.251	33.6	36.8	N/A	178.856
Near - Wave 09s	0.001	203.612	33.5	46.55	N/A	203.611
Near - Wave 10s	0.001	209.415	52	39.5	N/A	209.415

Summary Collation Output

Video Creation

The AVI Studio program is quite intuitive to use, and once you have mastered the basic functions, you should be able to create reasonably complex videos with relative ease. You are referred to the online help within the AVI Studio program for a full description of its functionality.

A sample layout for the AVI Studio is shown below. This includes the following windows (listed in order of appearance, from top left towards bottom right): Database Files, Preview, Camera Positions and Timeline. Also included but currently hidden are the Image Files and Option Sets windows.



For illustrative purposes, a sample AVI project file is included with this worked example, based on the combination of near vessel offset and 8s regular wave period.

CAMERA POSITIONS

For the purposes of this example, 14 camera positions were created to give an overall view of the complete structure and some close-up views of the various components.

- The first camera position specified is entitled ‘Full View’. The camera is positioned to give an overall view of the entire structure, as illustrated above.
- The next five camera positions, ‘Vessel 1’ to ‘Vessel 5’, provide various views around the vessel.
- ‘Under Vessel 1’ and ‘Under Vessel 2’ provide detailed views of the underside of the vessel, showing the vessel ends of the oil and gas jumpers.
- The ‘Half View’ camera position provides a view of the vessel and the buoy, similar to the ‘Full View’, but at a higher zoom level, and from a different viewpoint.

- The next camera position ‘Buoy 1’ gives a closer view of the buoy.
- ‘Buoy Detail 1’ provides a detailed view of the components beneath the buoy, namely the T-frame and buoy ends of the oil and gas jumpers.
- The camera position ‘Buoy Detail 2’ looks underneath the buoy.
- Camera positions ‘Buoy 2’ and ‘Buoy 3’ give some different viewing angles around the buoy.

OPTION SETS

Option Sets allow you to vary some display parameters at different stages during the video. For illustrative purposes, two option sets have been created in this example, namely ‘Show Node Labels’ and ‘Hide Node Labels’. These are used to switch on and off node lables at various stages during the video. For example, node labels are used to accompany the camera positions ‘Under Vessel 1’, ‘Under Vessel 2’ and ‘Buoy Detail 1’. This provides additional clarity to the video, highlighting pertinent components/connections within the model.

TIMELINE

The timeline is assembled as follows.

Time	Database	Camera	Option Set
0s	Wave 08s.dbm	Full View	Hide Node Labels
4s		Vessel 1	
8s		Vessel 2	
12s		Vessel 3	
15s		Vessel 4	
17s		Vessel 5	
19s		Under Vessel 1	Show Node Labels

22s		Under Vessel 2	
23s			Hide Node Labels
26s		Half View	
29s		Buoy 1	
33s		Buoy Detail 1	Show Node Labels
37s		Buoy Detail 1	Hide Node Labels
40s		Buoy Detail 2	
43s		Buoy 2	
46s		Full View	
48s		Full View	

ENCODING

While the AVI Studio project is provided with the installation, the AVI file itself must be generated locally before you can view the AVI video. You may create the video by pressing the  button in the Timeline view. The options are fairly self-explantory, but you may refer to the online help if you require any further information. In order to ensure the size of the resulting AVI file is reasonable, video compression is recommended, and you may select from a range of standard codecs.

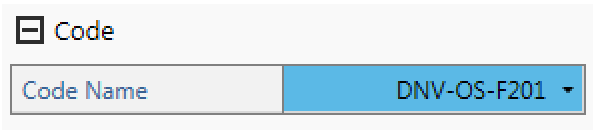
Code Checking Post-Processing

OVERVIEW OF CODE CHECKING MODULE

Flexcom allows for the post-processing of analysis results using three possible design codes including DNV-OS-F201 (Dynamic Risers), DNV-OS-F101 (Submarine Pipeline Systems), API STD 2RD (Dynamic Risers for Floating Productions Systems) and ISO-13628-7 (Completion/workover riser systems). While all the codes differ slightly in the how results are calculated and presented, the operation of each are very similar in nature. The code checking module (`$CODE CHECKING`) is comprised of four required keywords including `*CODE`, `*ENVIRONMENTAL`, `*MATERIAL`, and `*SECTION PROPERTIES`. This section will seek to explain how these keywords to perform a unity check in DNV-OS-F201.

CODE SELECTION

The first step in performing code checking post-processing is to select a code by which the result should be post-processed. Generally, risers are post-processed with DNV-OS-F201 or API STD 2RD, pipelines are checked against DNV-OS-F101 and completion/workover risers are checked against ISO-13628-7. While it is only possible to post-process using one design code at a time, API STD 2RD employs DNV-OS-F201's Load and Resistance Factor Design (LRFD) for calculation Method 3 making comparison between the codes readily available if desired. Specification of the design code is done using `*CODE`.




```
*CODE
 CODE=DNV-OS-F201
```

Code Selection

FUNCTIONAL AND ENVIRONMENTAL LOADS

DNV uses the notion of functional and environmental loads in the determination of Von Mises Stress (VMS), Working Stress Design (WSD) and Load and Resistance Factor Design (LRFD). Environmental loads are those loads due to environmental conditions. These include waves, current, wind, ice and earthquake. A functional load is defined as load caused by the existence of the structure and by operation and handling of the system excluding pressure. For further explanation of these loads, refer to DNV-OS-F201. In Flexcom, the environmental load is usually considered to be the dynamic analysis, and the functional load is typically a static analysis, either static or offset. Flexcom will assume the analysis preceding the environmental load case is the functional load unless otherwise specified. *ENVIRONMENTAL is used to define these environmental and functional loads along with the design factor (applicable only for API and ISO code checks), the collation option, and element set name.

Environmental

Environmental File Name	Functional File Name	API Design Factor (Fd)	Collate Option	Element Set Name	Start Time [s]	End Time [s]
".\Near\K2-HR-Near-Wave 08s"		[0.60]	[Yes]	Outer Pipe		
".\Near\K2-HR-Near-Wave 09s"		[0.60]	[Yes]	Outer Pipe		
".\Near\K2-HR-Near-Wave 10s"		[0.60]	[Yes]	Outer Pipe		
".\Far\K2-HR-Far-Wave 08s"		[0.60]	[Yes]	Outer Pipe		
".\Far\K2-HR-Far-Wave 09s"		[0.60]	[Yes]	Outer Pipe		
".\Far\K2-HR-Far-Wave 10s"		[0.60]	[Yes]	Outer Pipe		

```
*ENVIRONMENTAL
 ENVIRONMENTAL=".\Near\K2-HR-Near-Wave 08s
 SET=Outer Pipe
 ENVIRONMENTAL=".\Near\K2-HR-Near-Wave 09s
 SET=Outer Pipe
 ENVIRONMENTAL=".\Near\K2-HR-Near-Wave 10s
 SET=Outer Pipe
 ENVIRONMENTAL=".\Far\K2-HR-Far-Wave 08s"
 SET=Outer Pipe
 ENVIRONMENTAL=".\Far\K2-HR-Far-Wave 09s"
 SET=Outer Pipe
 ENVIRONMENTAL=".\Far\K2-HR-Far-Wave 10s"
 SET=Outer Pipe
```

Environmental Specification

MATERIAL PROPERTIES

DNV and API both employee a number of material specific empirical properties of the material in the calculation of the material utilisation. *MATERIAL is used to specify relevant structural material properties.

Material

Material Name	SMYS [N/m²]	SMTS [N/m²]	Young's Modulus [N/m²]	Fy Temp	Fu Temp	Alpha U	Poisson's Ratio
Carbon Steel	450E+06	535E+06	2.07E+11	3E+06	3E+06	[0.96]	[0.3]

```
*MATERIAL
 NAME=Carbon Steel
 450E+06, 535E+06, 2.07E+11, 3E+06, 3E+06
```

Material Property Specification

SECTION PROPERTIES

`*SECTION PROPERTIES` is used to specify which material corresponds to the which set. If a diameter or thickness is not specified, that will be assumed to be consistent with those specified geometeric sets in the static file.

Section Properties

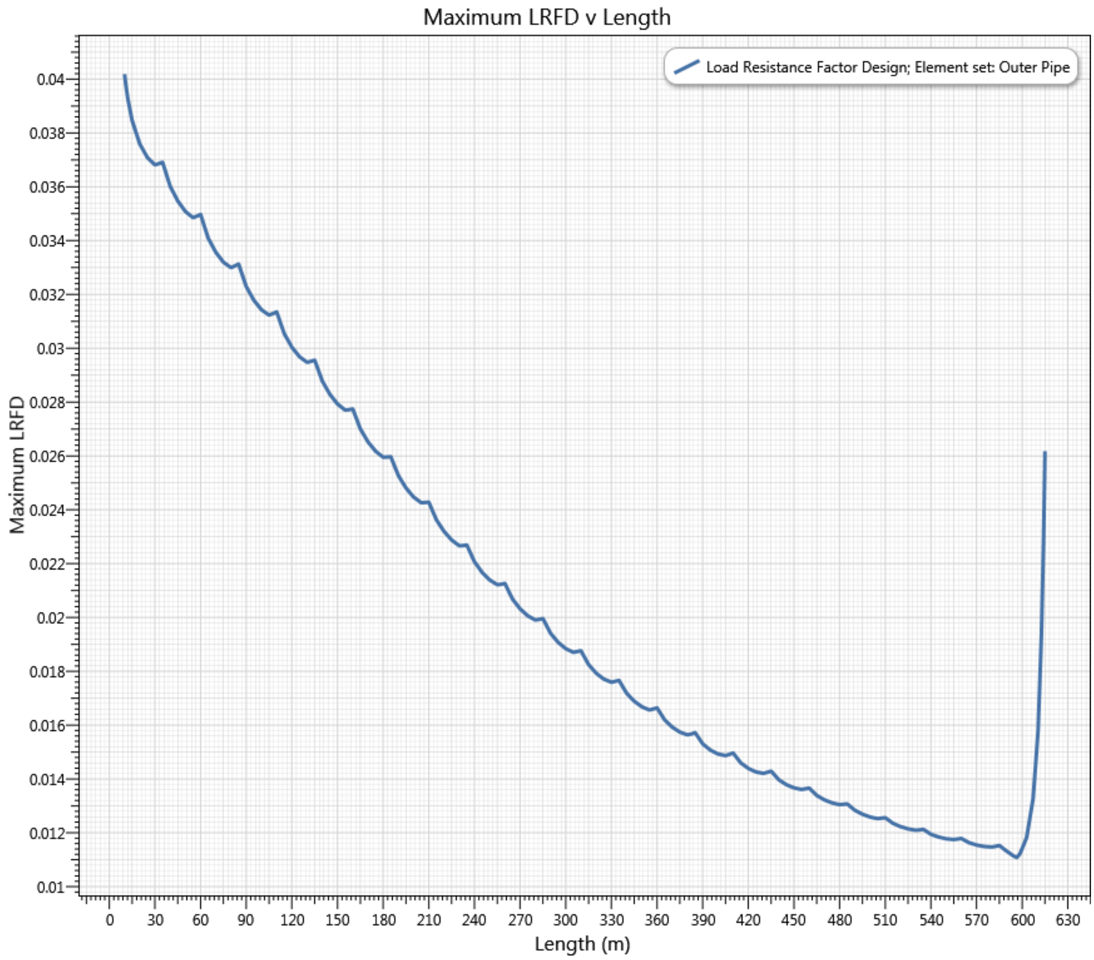
Set Name	Material Name	Alpha C	Diameter [m]	Thickness [m]	T corr [m]	Alpha Fab	Ovality	Moment Factor	Tension Factor
Outer Pipe	Carbon Steel	1.075			[0.0]	[0.85]	[0.005]	[1.0]	[1.0]

```
*SECTION PROPERTIES
 SET=Outer Pipe, MATERIAL=Carbon Steel
 1.075
```

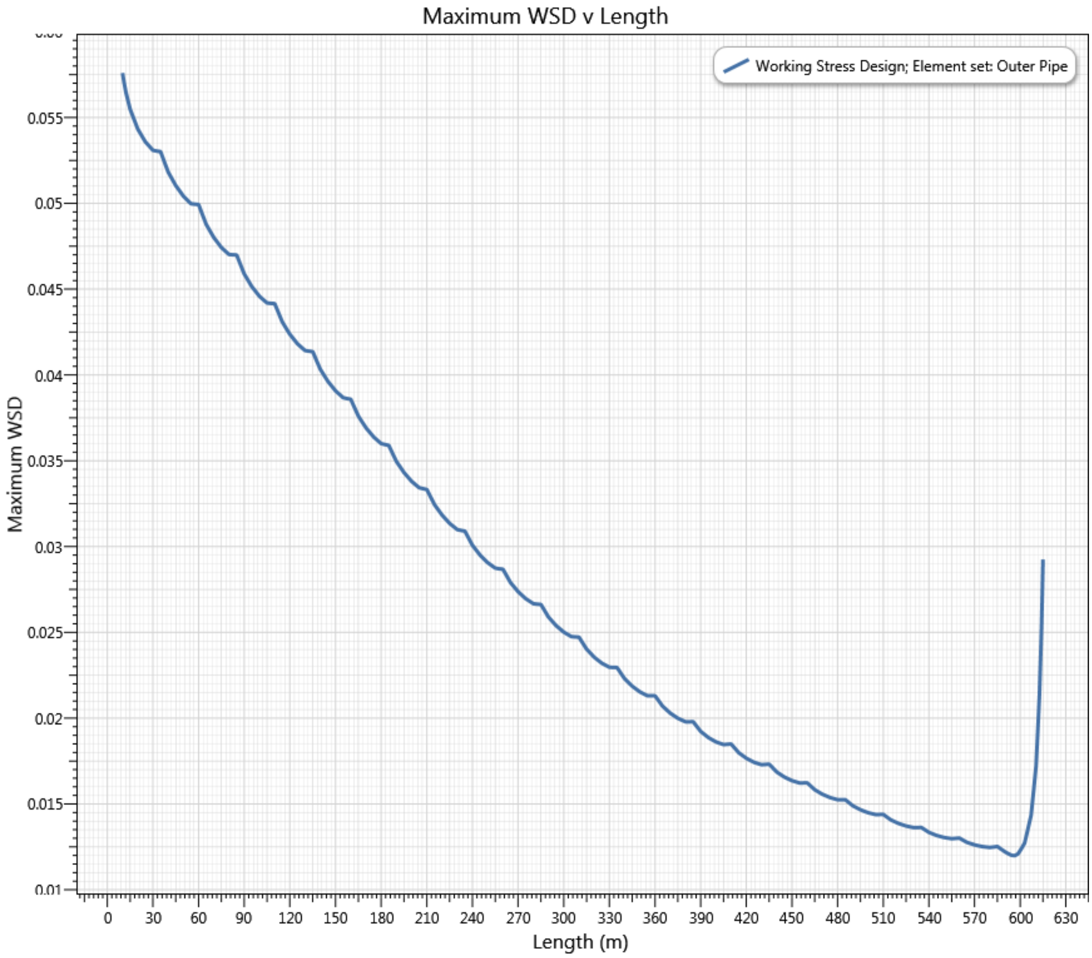
Section Property Specification

RESULTS

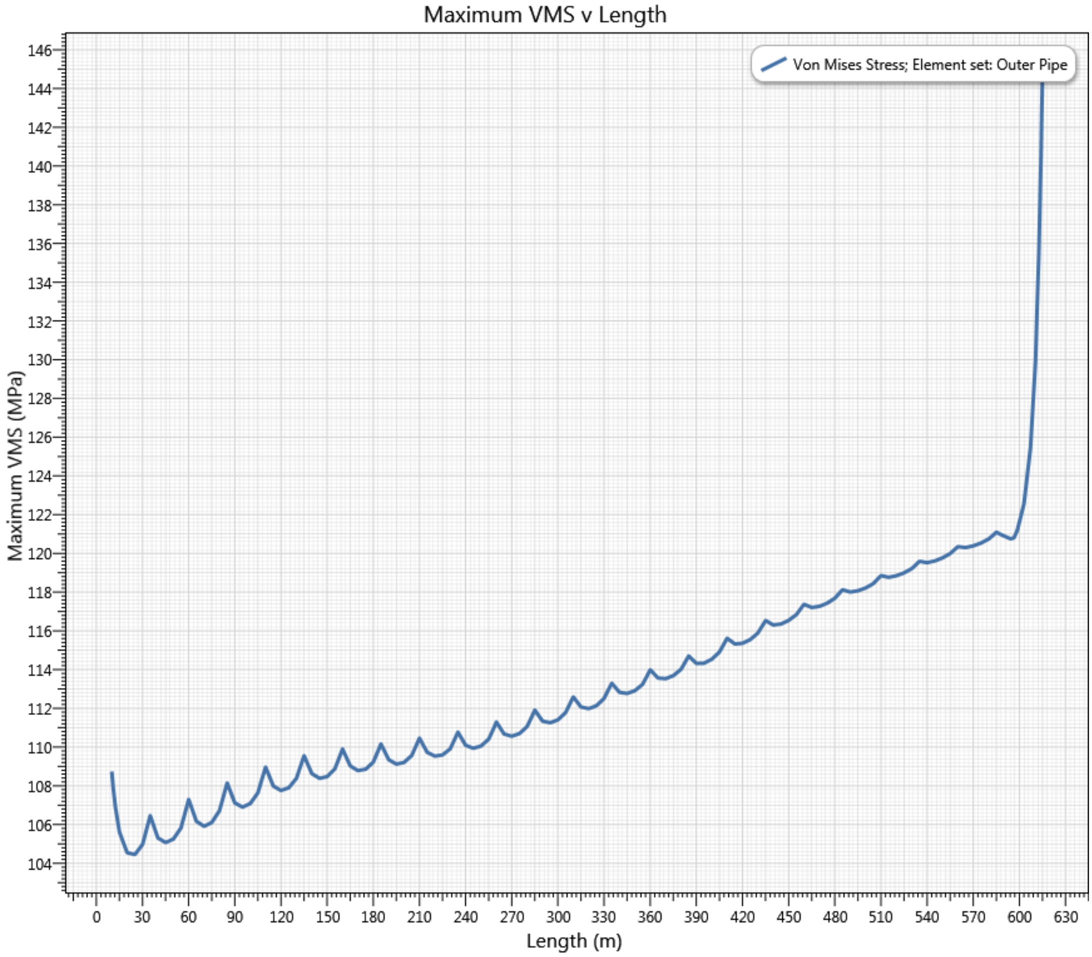
Results from the code checking post-processing are presentation in a number of ways. For every section, a plot is generated for LRFD, WSD, and von Mises stress. The LRFD and WSD values are both utilization values.



Load and Resistance Factor Design Utlisation



Working Stress Design Utilisation



von Mises Stress

Additionally, maximum utilisations and stresses are summarized in the output file.

Environmental #	Section #	Max LRFD	Max WSD	Max VMS
1	1	5.3762E-002	5.8497E-002	1.7333E+008
2	1	5.5655E-002	5.9132E-002	1.7498E+008
3	1	5.5817E-002	5.9086E-002	1.7512E+008
4	1	3.9580E-002	5.7009E-002	1.4333E+008
5	1	4.0177E-002	5.7585E-002	1.4462E+008
6	1	4.0189E-002	5.7596E-002	1.4488E+008

Collated Results

1.10.12 L - Wind Energy

Section L contains examples of offshore wind turbines:

- L01 - OC4 Semi-Submersible
- L02 - OC4 Semi-Submersible (Flexcom Wind)
- L03 - OC4 Jacket
- L04 - UMaine VolturnUS-S IEA15MW

1.10.12.1 L01 - OC4 Semi-Submersible

OC4 (Offshore Code Comparison Collaboration Continuation) is a code-to-code verification project operated under the IEA (International Energy Agency) Wind Task 30 and coordinated by NREL (National Renewable Energy Laboratory). In Phase II of OC4, participants used an assortment of simulation codes to model the coupled dynamic response of a 5-MW wind turbine installed on a floating semi-submersible in 200 m of water. Code predictions were compared from load case simulations selected to test different model features.

Although Flexcom was not officially represented in OC4, the software has been retrospectively benchmarked against OC4 results, as NREL and the IEA have kindly made all data from the project publicly available. The software validation is primarily focused on the OC4 semi-submersible platform (Connolly & O'Mahony, 2021), but comparisons are available for the OC4 jacket structure (Connolly & O'Mahony, 2021) also.

The overall layout of this example is as follows:

- Introduction gives an overview of the OC4 semi-submersible, the load cases simulated, and the project participants and software tools.
- Model Summary describes the Flexcom model in detail.

- Results from Flexcom are presented alongside results from a subset of the software tools used in OC4.

Introduction

OC4 PHASE II

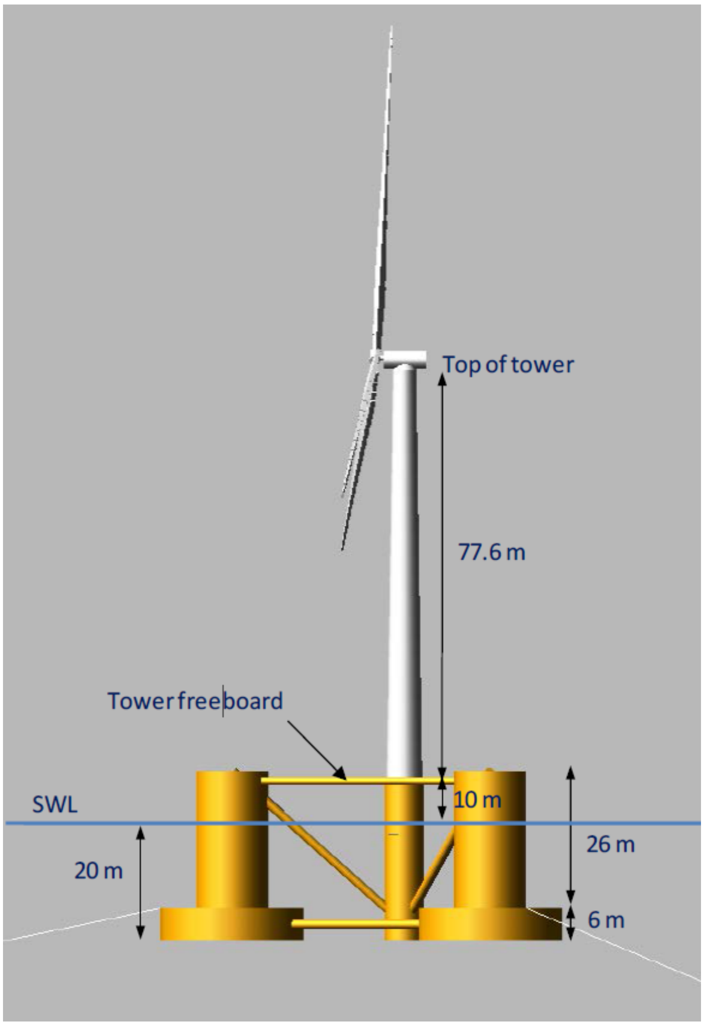
OC4 (Offshore Code Comparison Collaboration Continuation) is a code-to-code verification project operated under the IEA (International Energy Agency) Wind Task 30 and coordinated by NREL (National Renewable Energy Laboratory). In Phase II of OC4, participants used an assortment of simulation codes to model the coupled dynamic response of a 5-MW wind turbine installed on a floating semi-submersible in 200 m of water. Code predictions were compared from load case simulations selected to test different model features. According to (Robertson et al., 2014), "*the comparisons have resulted in a greater understanding of offshore floating wind turbine dynamics and modeling techniques, and better knowledge of the validity of various approximations. The lessons learned from this exercise have improved the participants' codes, thus improving the standard of offshore wind turbine modeling*". 21 different organizations from 11 different countries submitted results using 19 different simulation codes. Although Flexcom was not officially represented in OC4, the software has been retrospectively benchmarked against OC4 results, as NREL and the IEA have kindly made all data from the project publicly available.

Robertson et al. (2014) summarised the modelling capabilities of the various software tools under the headings of structural dynamics, aerodynamics, hydrodynamics and mooring model. Using the OC4 terminology, Flexcom would be categorised as follows:

- Structural dynamics: Tower – Finite Element (FE), Platform – Rigid
- Aerodynamics: Blade-Element Momentum (BEM) or Generalized Dynamic Wake (GDW), plus Dynamic Stall (DS)
- Hydrodynamics: Potential Flow (PF) plus Morison Equation (ME)
- Mooring model: Finite Element (FE) / Dynamic

SEMI-SUBMERSIBLE PLATFORM

Phase II of the OC4 project involved the modeling of a semi-submersible floating offshore wind system developed for the DeepCwind project (Goupee et al., 2012) as shown below. This concept was chosen for its increased hydrodynamic complexity compared to the only other floating system analyzed in the OC3 and OC4 projects, the OC3-Hywind spar buoy.



OC4 Semisub (Robertson et al., 2014)

A summary of the semi-sub's main characteristics is presented here, and further details are available in Robertson et al., 2014.

Property	Value
Depth of platform base below SWL (total draft)	20m
Elevation of main column (tower base) above SWL	10m

Property	Value
Elevation of offset columns above SWL	12m
Length of upper columns	26m
Length of base columns	6m
Depth to top of base columns below SWL	14 m
Diameter of main column	6.5 m
Diameter of offset (upper) columns	12 m
Diameter of base columns	24 m
Diameter of pontoons and cross braces	1.6 m
Platform mass, including ballast	1.3473E+7 kg
Platform centre of mass location below SWL	13.46 m
Number of mooring lines	3
Angle between adjacent lines	120 degrees
Depth to anchors below SWL (water depth)	200 m
Depth to fairleads below SWL	14 m
Radius to anchors from platform centerline	837.6 m
Radius to fairleads from platform centerline	40.868 m
Unstretched mooring line length	835.5 m
Mooring line diameter	0.0766 m

LOAD CASES

21 different load cases were performed in OC4 Phase II, encompassing varying levels of model complexity and a variety of ambient loading conditions. A subset of these cases were reproduced using Flexcom, and this forms the basis of this example. The load cases are ordered in increasing complexity, with three distinct groupings. The first group (1.X) are focused on fundamentals, including a static equilibrium simulation, a modal analysis, and a series of free-decay simulations. These simulations are run with still water, and have the generator locked (blade rotation is prevented via a brake). The second group (2.X) are focused on wave loading without wind - again the generator is locked. The third and final group (3.X) examines combined wind and wave excitation, including regular and irregular waves, and steady and turbulent wind.

Although not part of OC4 Phase II, one additional load case (OC4 P2 LC3.1 modified) was added here in order to verify the correct operation of the control system in Flexcom.

Load Case	Description	Wind	Wave
OC4 P2 LC1.1	Eigenanalysis	No wind	Still water
OC4 P2 LC1.2	Static equilibrium	No wind	Still water
OC4 P2 LC1.3a	Free decay, surge	No wind	Still water
OC4 P2 LC1.3b	Free decay, heave	No wind	Still water

Load Case	Description	Wind	Wave
OC4 P2 LC1.3c	Free decay, pitch	No wind	Still water
OC4 P2 LC1.3d	Free decay, yaw	No wind	Still water
OC4 P2 LC2.1	Regular waves	No wind	Regular airy: H = 6 m, T = 10 s
OC4 P2 LC2.2	Irregular waves	No wind	Irregular airy: Hs = 6 m, Tp = 10 s, γ=2.87, JONSWAP spectrum
OC4 P2 LC2.3	Current only	No wind	Surface = 0.5 m/s, 1/7th power law decrease with depth
OC4 P2 LC2.4	Current and regular waves	No wind	Regular airy: H = 6 m, T = 10 s; current at surface = 0.5 m/s, 1/7th power law
OC4 P2 LC2.5	50-year extreme wave	No wind	Irregular airy: Hs = 15.0 m, Tp = 19.2 s, γ=1.05, JONSWAP spectrum
OC4 P2 LC2.6	RAO estimation, no wind	No wind	Banded white noise, PSD =1 m2/Hz for 0.05-0.25 Hz

Load Case	Description	Wind	Wave
OC4 P2 LC3.1	Deterministic, below rated	Steady, uniform, no shear: Vhub = 8 m/s	Regular airy: H = 6 m, T = 10 s
OC4 P2 LC3.1 modified (Control System Test)*	Deterministic, below and above rated	Variable, uniform, shear: Vhub = 5/10/15/20 m/s	Regular airy: H = 6 m, T = 10 s
OC4 P2 LC3.2	Stochastic, at rated	Turbulent (Mann model): Vhub = Vr (11.4 m/s)	Irregular airy: Hs = 6 m, Tp = 10 s, γ=2.87, JONSWAP spectrum
OC4 P2 LC3.3	Stochastic, above rated	Turbulent (Mann model): Vhub = 18 m/s	Irregular airy: Hs = 6 m, Tp = 10 s, γ=2.87, JONSWAP spectrum
OC4 P2 LC3.4	Wind/wave/current	Steady, uniform, no shear: Vhub = 8 m/s	Regular airy: H = 6 m, T = 10 s; current at surface = 0.5 m/s, 1/7th power law
OC4 P2 LC3.5	50-year extreme wind/wave	Turbulent (Mann model): Vhub = 47.5 m/s	Irregular airy: Hs = 15.0 m, Tp = 19.2 s, γ=1.05, JONSWAP spectrum

Load Case	Description	Wind	Wave
OC4 P2 LC3.6	Wind/wave misalignment	Steady, uniform, no shear: Vhub = 8 m/s	Regular airy: H = 6 m, T = 10 s, direction = 30°
OC4 P2 LC3.7	RAO estimation, with wind	Steady, uniform, no shear: Vhub = 8 m/s	Banded white noise, PSD =1 m2/Hz for 0.05-0.25 Hz
OC4 P2 LC3.8	Mooring line loss	Steady, uniform, no shear: Vhub = 18 m/s	Regular airy: H = 6 m, T = 10 s
OC4 P2 LC3.9a	Flooded column	No wind	Still water
OC4 P2 LC3.9b	Flooded column	Turbulent (Mann model): Vhub = 18 m/s	Irregular airy: Hs = 6 m, Tp = 10 s, γ=2.87, JONSWAP spectrum

**Load case OC4 P2 LC3.1 modified) was not part of OC4 Phase II. It was added here in order to verify the correct operation of the control system in Flexcom*

SOFTWARE TOOLS AND PROJECT PARTICIPANTS

Results from Flexcom are presented alongside results from a subset of the software tools used in OC4 Phase II. This helps to reduce clutter on graph comparisons, whilst still including some of the most well-known software tools in industry. In some cases the same software tool was used by multiple OC4 participants, one of whom is included here.

Software	Developer	OC4 Participant
FAST[1]	NREL	NREL
Bladed[2]	Garrad Hassan	Garrad Hassan
OrcaFlex	Orcina	4Subsea
HAWC2	Technical University of Denmark (DTU)	Technical University of Denmark (DTU)
Riflex Coupled	Norwegian Marine Technology Research Institute (MARINTEK)	Norwegian Marine Technology Research Institute (MARINTEK)
DeepLinesWT[3]	Principia & IFP Energies Nouvelles	Principia
CHARM3D+FAST	Texas A&M University (TAMU) & NREL	American Bureau of Shipping (ABS)
Flexcom[4]	Wood	N/A

Notes

1. Data files for FAST are available from NREL for the majority of load cases. For a small number of load cases, results from NREL were not found in the IEA Wind Task 30 results library, so we sourced data for FAST from another participant instead. Where this is the case, the relevant data series in the comparisons presented here are labelled appropriately.

2. Results from the standard version of Bladed are used in the comparisons presented here. For a small number of load cases, results from Garrad Hassan for Bladed were not found in the IEA Wind Task 30 results library, so we used results for 'Bladed Advanced Hydro Beta' instead. If you would like further information about this version, we suggest that you contact Garrad Hassan directly.

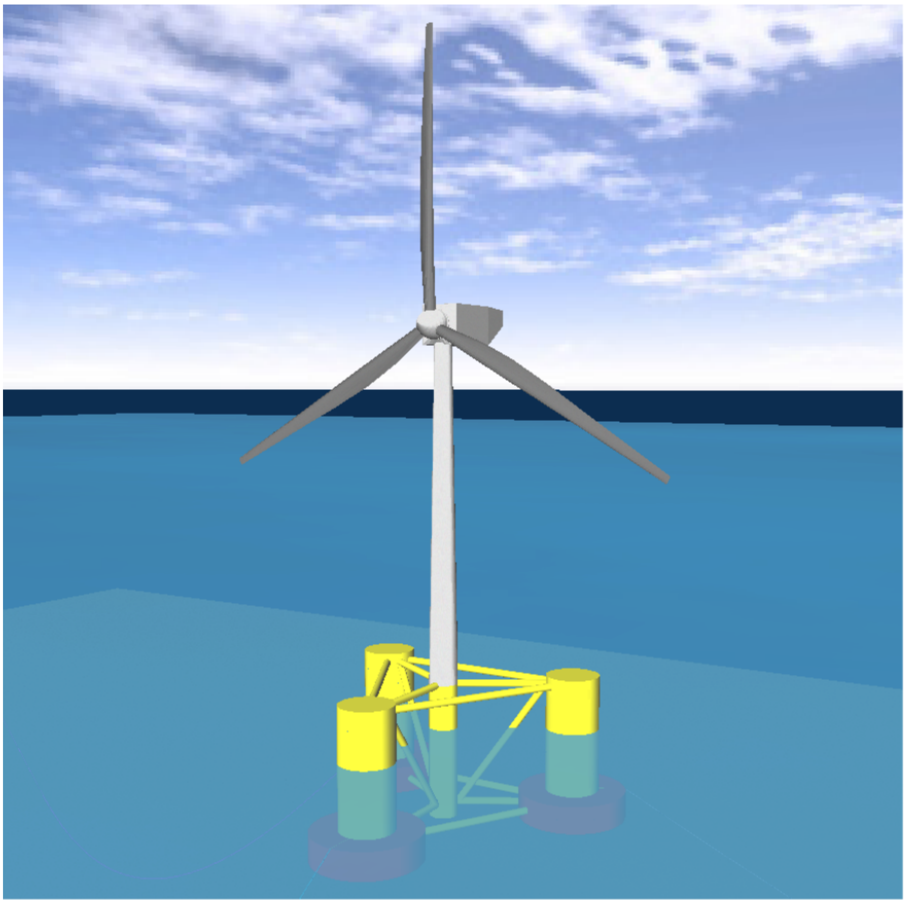
3. Principia provided two sets of results for data for DeepLines Wind, with the models differing in terms of how the hydrodynamic response of the platform was simulated. The first approach corresponds to potential flow theory, which Principia describe as *“hydrodynamic data calculated with Diodore software (potential flow + quadratic damping), rigid body (no strains)*”. The second approach corresponds to Morison’s equation, which Principia describe as “*FE model (Morison elements)*”. Most project participants provided one set of results only. In reality, many simulation tools adopt a combined approach, whereby the radiation and diffraction components are modelled using potential flow theory and the viscous drag terms are modelled using Morison’s equation. Given that the Morison approach ignores wave field disturbance, it is not generally used to model floating structures of significant size. So for the purposes of the comparisons presented here, results for DeepLines Wind are based on the potential flow modelling approach. However as this neglects viscous drag effects, the results presented here do not represent the full modelling capabilities of DeepLines Wind.

4. Flexcom was not part of OC4. It has been retrospectively benchmarked against OC4 results by Wood personnel.

Model Summary

INTRODUCTION

The Flexcom model of the OC4 semi-sub is shown below.



FLOATING PLATFORM

The floating platform is modelled using a series of discrete Lines to represent the 3 columns, the upper and lower pontoons, plus the various cross braces. These lines are connected up at appropriate points using a range of Equivalent Nodes to form a single coherent structure.

Strictly speaking it is not necessary to model the floating platform in such detail. A skeleton model which just includes key points of interest (such as the centres of gravity and buoyancy) would facilitate the application of concentrated loads. However, there are some advantages associated with creating a more detailed model - refer to Floating Body Modelling Detail if you are interested in further details.

Each line is assigned rigid Stiffness terms as the platform is assumed to act as a rigid body. All lines are assigned a Mass per Unit Length of zero (as the total mass is concentrated at the centre of mass) and a Buoyancy Diameter of zero (as the total buoyancy is concentrated at the centre of buoyancy). Each line is assigned a physical Drag Diameter, which allows Morison drag loads to be added to the standard radiation-diffraction excitation forces.

A Floating Body is defined which models the hydrodynamic characteristics of the floater. Hydrostatic Stiffness terms are used to simulate restoring forces and moments due to buoyancy. Added Mass, Radiation Damping and Force RAO coefficients are defined for the floating body over a range of discrete frequencies - these terms enable the computation of incident, diffracted and radiated (linear) wave potentials to be simulated. Note that these inputs are derived separately from a radiation-diffraction analysis. Relevant Rotational Inertia terms are specified at the floating body centre of gravity, while the body's mass is represented by a Point Mass.

TOWER

The tower is constructed using a single Line, with its lower end attached to the floating platform using an Equivalent Node. As the tower is tapered from base to top, it is modelled using several Line Sections of decreasing diameter. The finite element mesh density assigned to the tower ensures that each section is modelled using a single element. Note that a consistent mesh density is assigned to the aerodynamic model via the `*TOWER INFLUENCE` keyword. Realistic Stiffness and Mass per Unit Length terms are individually assigned to each tower section based on the material properties for steel and the relevant diameter and wall thickness at the section's mid-point. As the tower will not experience any hydrodynamic loading, the line is assigned zero values of Buoyancy Diameter and Drag Diameter.

TURBINE ASSEMBLY

Finite element Nodes are explicitly created at the centres of mass of the nacelle, hub and blades (centre of mass at initial position) respectively. These nodes are then connected to the top of the tower using finite Elements. The use of explicitly created nodes and elements is more convenient than using lines in such circumstances.

Point Mass terms are used to position appropriate masses at the nodes, hence the elements have zero values of Mass per Unit Length. All elements are assigned large Stiffness terms to simulate rigidity. As the turbine assembly will not experience any hydrodynamic loading, all elements are assigned zero values of Buoyancy Diameter and Drag Diameter.

An Auxiliary Profile is used to represent the rotating blades. While this has no structural function, it enhances the visual appeal of the model, and assists in the understanding of rotor and platform motions post-simulation.

AERODYNAMICS

All inputs which are required by AeroDyn to compute the aerodynamic loading on the blades and tower are are logically grouped together under the `$AERODYN` section, and specified in the dynamic simulation file. Fundamental inputs include Blade Geometries, Aerofoil Coefficients, miscellaneous Turbine Inputs (such as hub height, hub radius, overhang, shaft tilt, blade precone etc.) and Tower Influence (i.e. tower drag). These inputs should be intuitively familiar to engineers with some wind turbine modelling experience and you are referred to the keyword documentation should you require further information regarding the significance of any particular input.

CONTROL SYSTEM

The wind turbine control system is defined via the `*SERVODYN` keyword, which references the standard control DLL provided by NREL for the OC4 semi-submersible. At low wind speeds the turbine is operating below rated power, so the rotor is allowed to rotate freely without any control in order to maximise power extraction. At intermediate wind speeds the turbine is fully operational, and the generator torque is used to control rotor speed while maximising generated power. At higher wind speeds the available wind power is above the rated power of the turbine, so blade pitch control is used to feather the blades and shed excess power.

MOORING LINES

The mooring lines are created using 3 separate Lines. The upper end of each mooring line is attached to the relevant fairlead node on the floating platform using the Equivalent Nodes facility, while the lower ends are constrained using Fixed Boundary Conditions. Realistic Stiffness, Mass per Unit Length, Buoyancy Diameter and Drag Diameter terms are assigned to the mooring lines.

Results

You can use the hyperlinks below to browse through the results comparisons for the subset of OC4 Phase II load cases considered. Official results from OC4 were obtained from the IEA (International Energy Agency) Wind website, or more specifically, the IEA Wind Task 30 results library. Results from Flexcom are presented alongside results from a subset of the software tools used in OC4 Phase II. This helps to reduce clutter on graph comparisons, whilst still including some of the most well-known software tools in industry. For clarity, Flexcom results are overlaid as the last data series in each graph with the data series shown in dashed red to be distinctive.

We will be adding more test cases to the benchmark comparisons over time, once the relevant models have been completed.

Load Case	Description	Wind	Wave
OC4 P2 LC1.1	Eigenanalysis	No wind	Still water
OC4 P2 LC1.2	Static equilibrium	No wind	Still water
OC4 P2 LC1.3a	Free decay, surge	No wind	Still water
OC4 P2 LC1.3b	Free decay, heave	No wind	Still water
OC4 P2 LC1.3c	Free decay, pitch	No wind	Still water
OC4 P2 LC1.3d	Free decay, yaw	No wind	Still water
OC4 P2 LC2.1	Regular waves	No wind	Regular airy: H = 6 m, T = 10 s

Load Case	Description	Wind	Wave
OC4 P2 LC2.2	Irregular waves	No wind	Irregular airy: Hs = 6 m, Tp = 10 s, γ=2.87, JONSWAP spectrum
OC4 P2 LC2.3	Current only	No wind	Surface = 0.5 m/s, 1/7th power law decrease with depth
OC4 P2 LC2.4	Current and regular waves	No wind	Regular airy: H = 6 m, T = 10 s; current at surface = 0.5 m/s, 1/7th power law
OC4 P2 LC2.5	50-year extreme wave	No wind	Irregular airy: Hs = 15.0 m, Tp = 19.2 s, γ=1.05, JONSWAP spectrum
OC4 P2 LC2.6	RAO estimation, no wind	No wind	Banded white noise, PSD =1 m2/Hz for 0.05-0.25 Hz
OC4 P2 LC3.1	Deterministic, below rated	Steady, uniform, no shear: Vhub = 8 m/s	Regular airy: H = 6 m, T = 10 s

Load Case	Description	Wind	Wave
OC4 P2 LC3.1 modified (Control System Test)*	Deterministic, below and above rated	Variable, uniform, shear: Vhub = 5/10/15/20 m/s	Regular airy: H = 6 m, T = 10 s
OC4 P2 LC3.2	Stochastic, at rated	Turbulent (Mann model): Vhub = Vr (11.4 m/s)	Irregular airy: Hs = 6 m, Tp = 10 s, γ=2.87, JONSWAP spectrum
OC4 P2 LC3.3	Stochastic, above rated	Turbulent (Mann model): Vhub = 18 m/s	Irregular airy: Hs = 6 m, Tp = 10 s, γ=2.87, JONSWAP spectrum
OC4 P2 LC3.4	Wind/wave/current	Steady, uniform, no shear: Vhub = 8 m/s	Regular airy: H = 6 m, T = 10 s; current at surface = 0.5 m/s, 1/7th power law
OC4 P2 LC3.5	50-year extreme wind/wave	Turbulent (Mann model): Vhub = 47.5 m/s	Irregular airy: Hs = 15.0 m, Tp = 19.2 s, γ=1.05, JONSWAP spectrum
OC4 P2 LC3.6	Wind/wave misalignment	Steady, uniform, no shear: Vhub = 8 m/s	Regular airy: H = 6 m, T = 10 s, direction = 30°

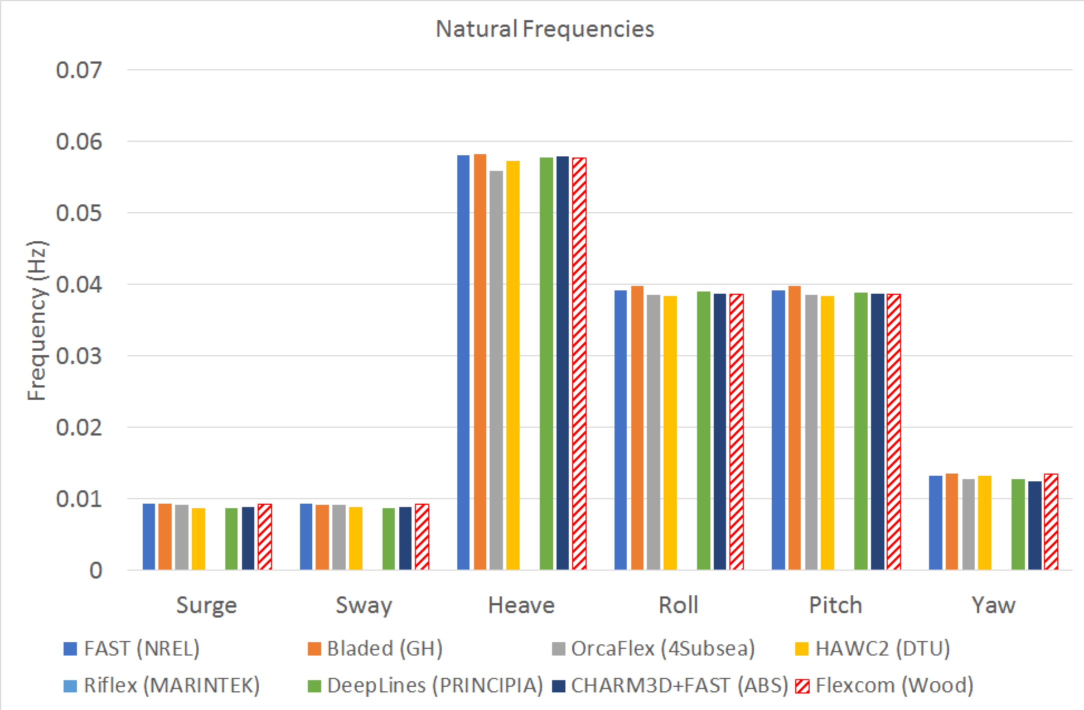
Load Case	Description	Wind	Wave
OC4 P2 LC3.7	RAO estimation, with wind	Steady, uniform, no shear: Vhub = 8 m/s	Banded white noise, PSD =1 m2/Hz for 0.05-0.25 Hz
OC4 P2 LC3.8	Mooring line loss	Steady, uniform, no shear: Vhub = 18 m/s	Regular airy: H = 6 m, T = 10 s
OC4 P2 LC3.9a	Flooded column	No wind	Still water
OC4 P2 LC3.9b	Flooded column	Turbulent (Mann model): Vhub = 18 m/s	Irregular airy: Hs = 6 m, Tp = 10 s, γ=2.87, JONSWAP spectrum

**not part of OC4 Phase II*

OC4 P2 LC1.1 Eigenanalysis

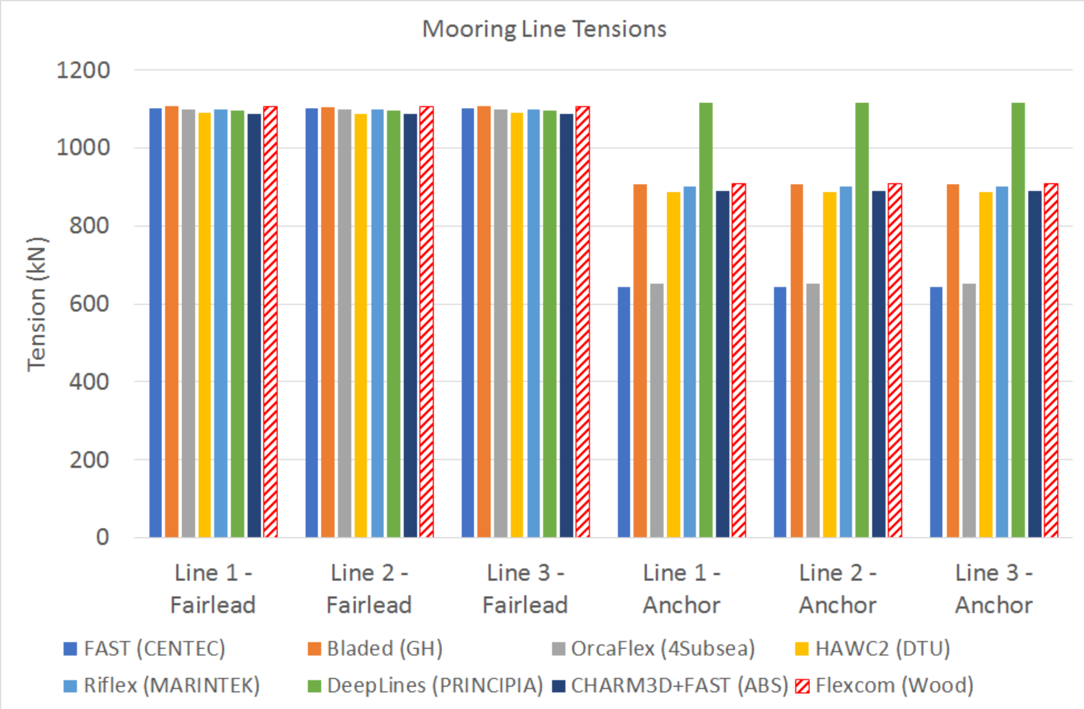
The natural frequencies predicted by Flexcom show close agreement with the other software tools.

Note that results from Riflex (MARINTEK) were not found in the IEA Wind Task 30 results library, hence they appear blank in this comparison.



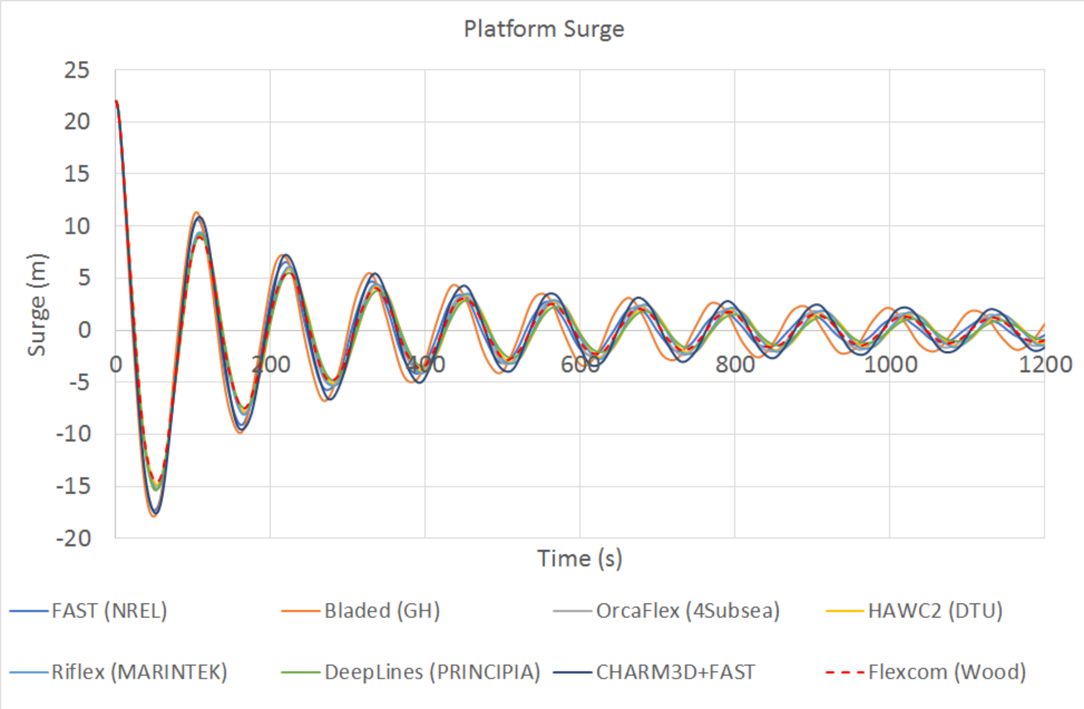
OC4 P2 LC1.2 Static equilibrium

The mooring line tensions predicted by Flexcom show close agreement with the other software tools. All tools predict similar fairlead tensions in all 3 mooring lines. Results from FAST (CENTEC) and Orcaflex (4Subsea) are lower than the others, possibly caused by tension dissipation due to seabed friction. It is possible to model this phenomenon in Flexcom also, but it was decided not to, for reasons of consistency with the other software tools. Results from DeepLinesWT (Principia) appear unrealistic as they are actually larger than the fairlead tensions, perhaps there was some mistake made during the uploading of results. Note that results from FAST (NREL) were not found in the IEA Wind Task 30 results library, so we have used results from FAST (CENTEC) instead in this comparison.



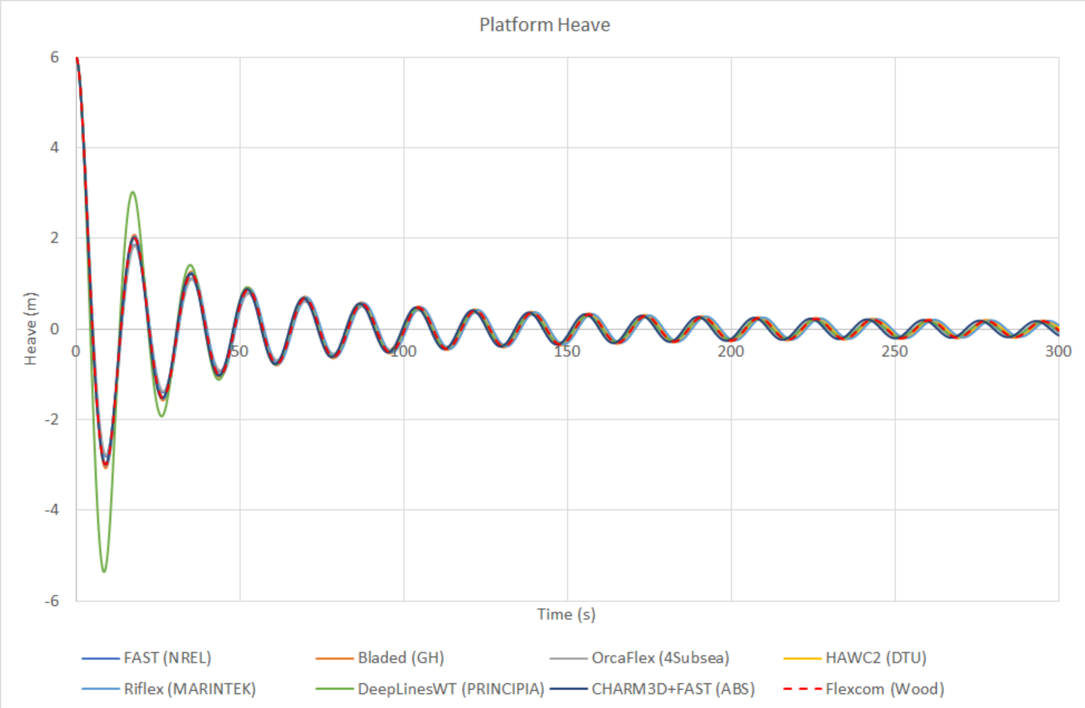
OC4 P2 LC1.3a Free decay, surge

Flexcom's surge response shows close agreement with the other software tools, in terms of response period and decay rate. Results from Bladed (GH) show a surge response which has a slightly shorter period than the other software tools.



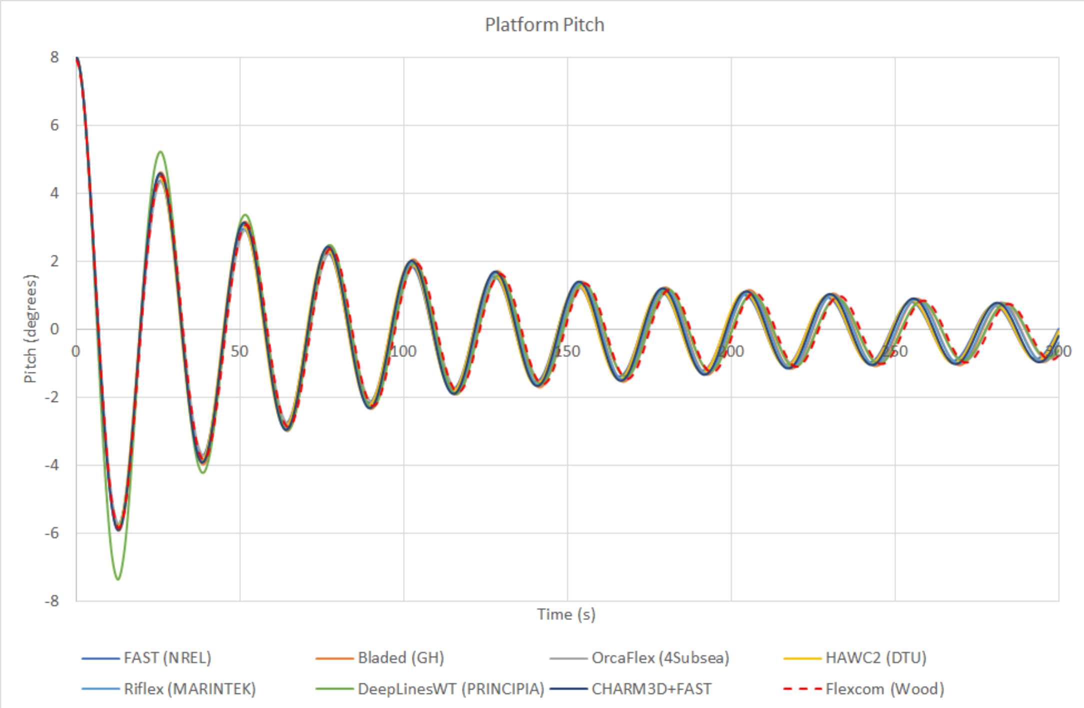
OC4 P2 LC1.3b Free decay, heave

Flexcom's heave response shows close agreement with the other software tools, in terms of response period and decay rate. Results from DeepLines WT (Principia) show a heave response which has a slightly larger amplitude initially than the other software tools.



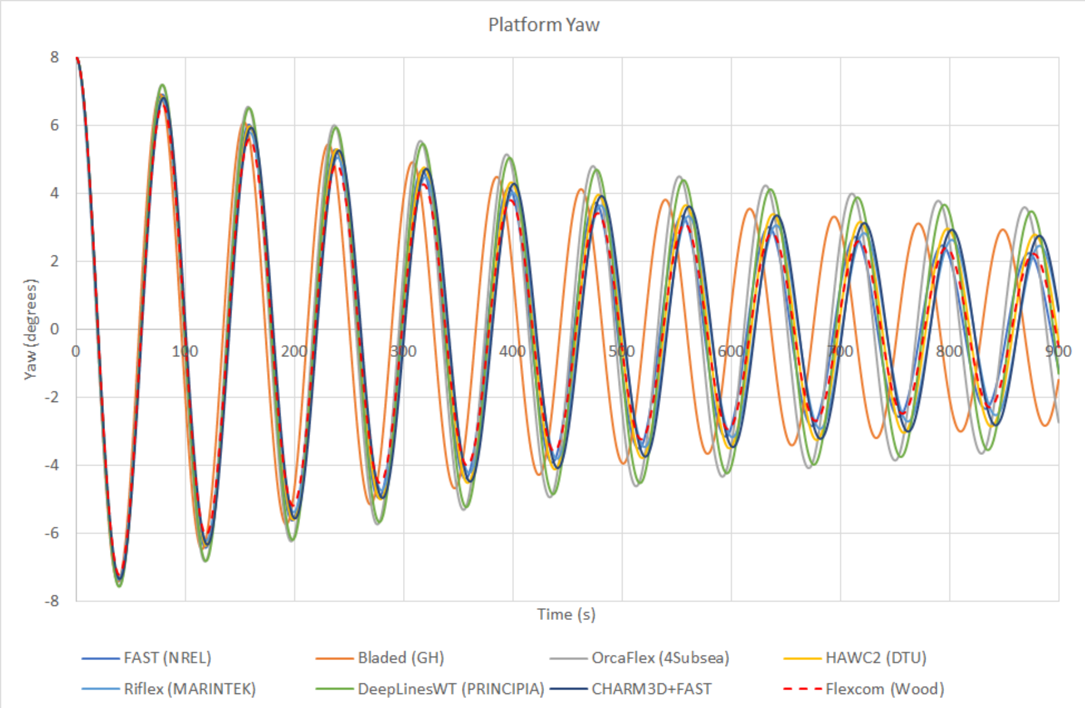
OC4 P2 LC1.3c Free decay, pitch

Flexcom's pitch response shows close agreement with the other software tools, in terms of response period and decay rate. Results from Bladed (GH) show a pitch response which has a slightly shorter period than the other software tools.



OC4 P2 LC1.3d Free decay, yaw

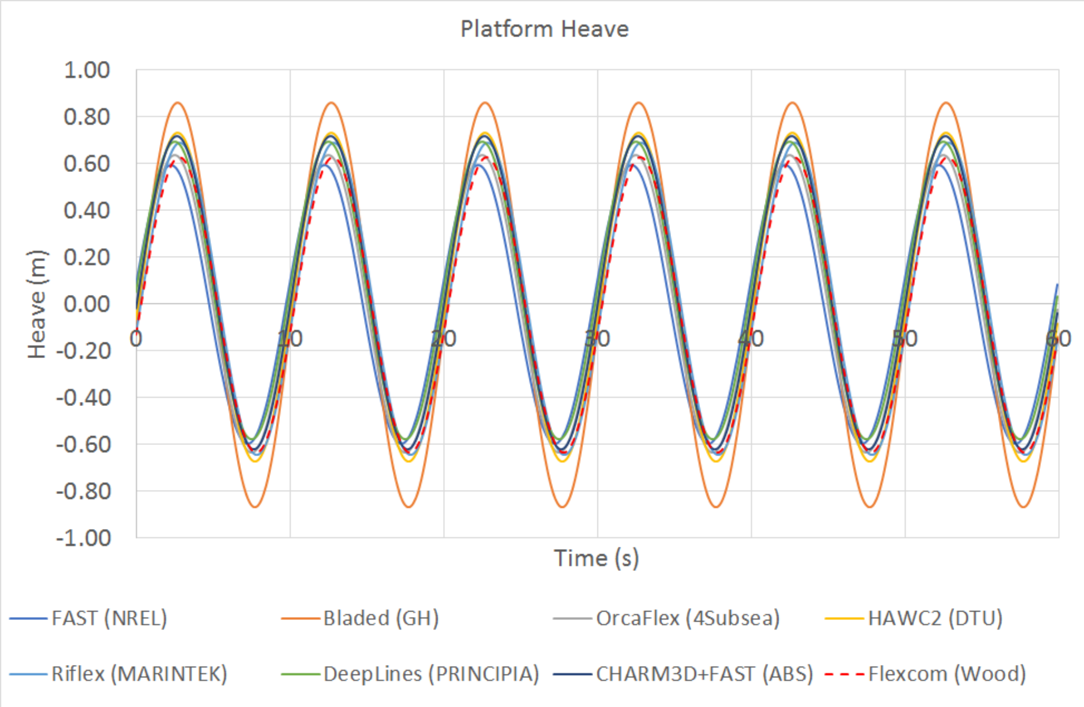
Flexcom's yaw response shows close agreement with the other software tools, in terms of response period and decay rate. All software predict a similar response period. There is some difference in terms of response amplitude, with Flexcom's yaw amplitude at the lower bound of the software tools.



OC4 P2 LC2.1 Regular waves

PLATFORM HEAVE

Flexcom's heave response shows close agreement with the other software tools, in terms of response amplitude, period and phase. Results from Bladed (GH) show a heave response which is slightly larger than the other software tools.



PLATFORM SURGE

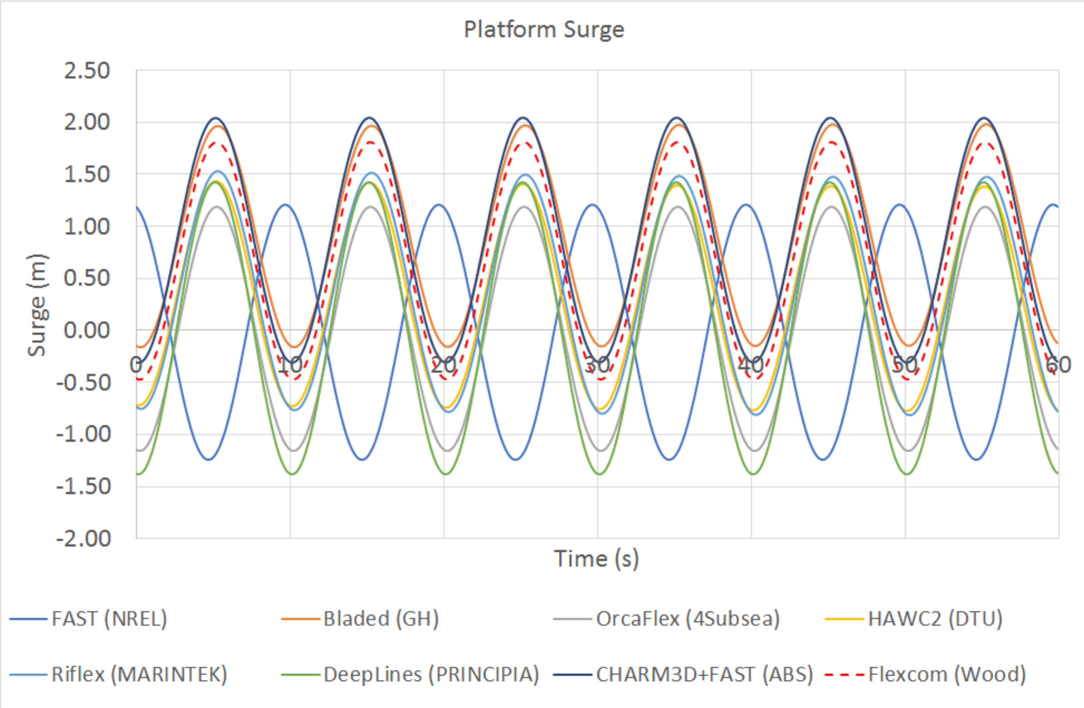
The surge response shows considerable variation across the different software products, depending on how drift forces are being modelled (if at all). Robertson et al. (2014) note that:

1. Simulations using first-order potential flow-theory and/or Morison's equation calculated at the undisplaced platform position (without any wave stretching assumption) will oscillate about a zero-mean position.

2. Simulations which incorporate one or more sources of non-linear drift will predict a non-zero mean value for the surge motion. Any of the following can induce non-linear drift...

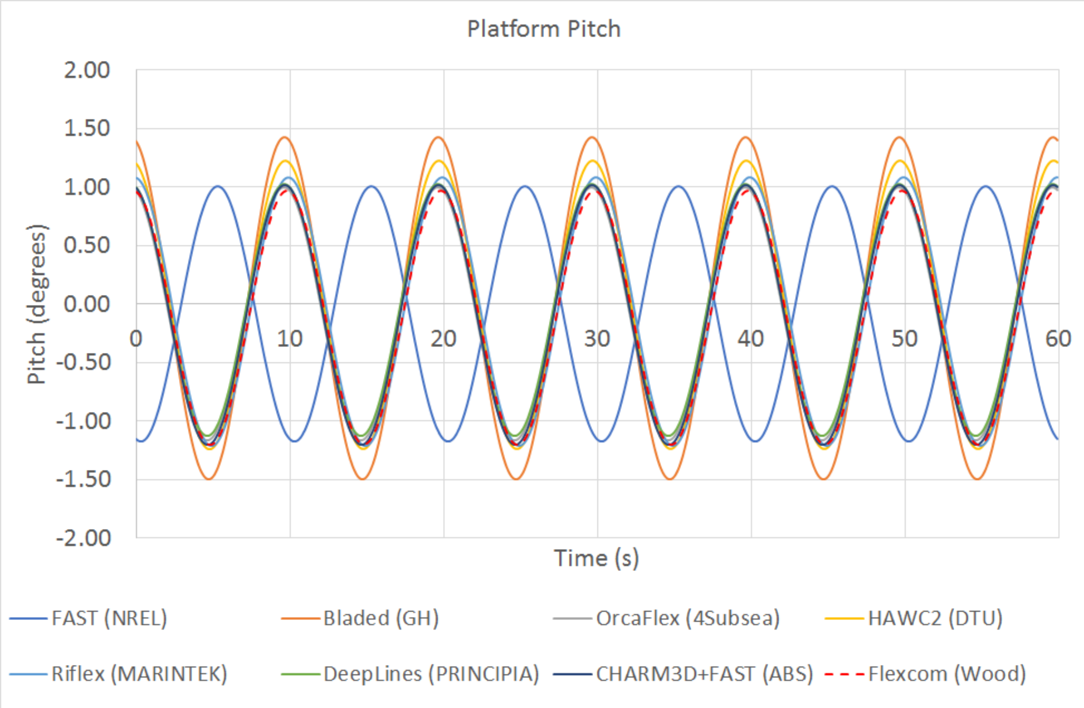
   - Inclusion of second-order terms in the potential-flow theory solution
   - Approximation of difference-frequency terms through Newman's approximation
   - Application of a mean-drift force derived from first-order potential-flow theory
   - Application of Morison's equation at the instantaneous platform position
   - Integration of Morison's equation up to the instantaneous wave elevation using a stretching technique such as Wheeler

The Flexcom model in this case computes the viscous drag forces at the instantaneous platform position which leads to a non-zero mean surge position. Results from Charm3D+FAST (ABS) and Bladed (GH) show the largest mean surge offsets, as these account for most of the drift modelling approaches noted above. Results from FAST (NREL) appear out of phase with the other tools, although this discrepancy may be caused by an inconsistency in presentation only.



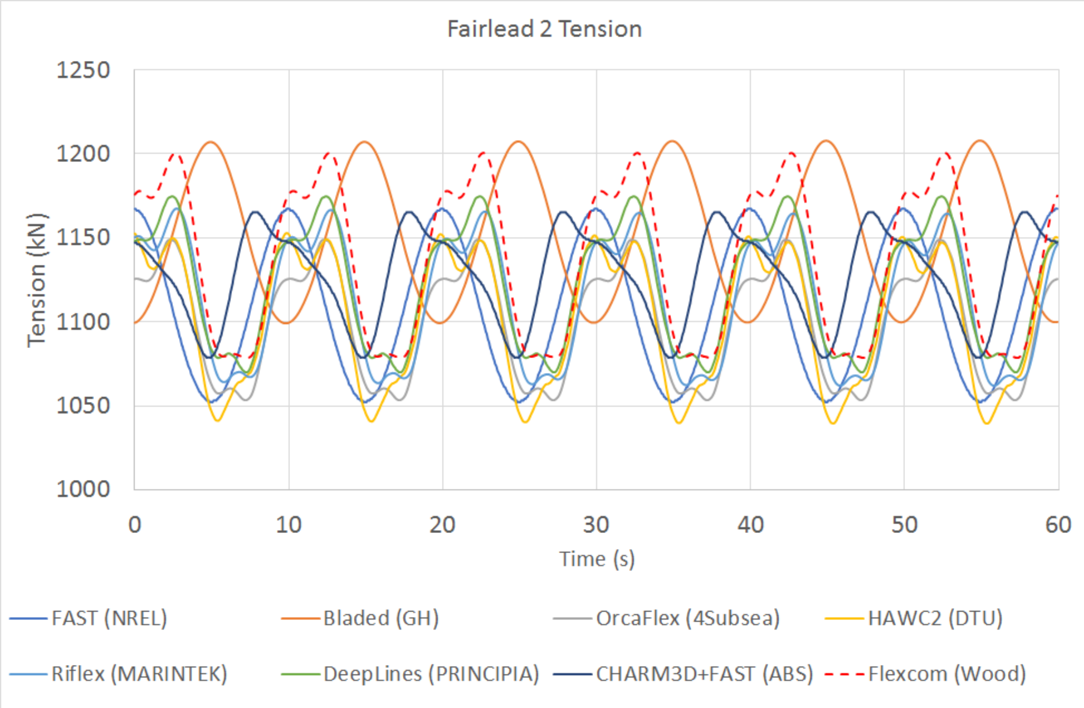
PLATFORM PITCH

Flexcom's pitch response shows close agreement with the other software tools, in terms of response amplitude, period and phase. Results from Bladed (GH) show a pitch response which is slightly larger than the other software tools. Results from FAST (NREL) appear out of phase with the other tools, although this discrepancy is most likely caused by an inconsistency in presentation only.



FAIRLEAD TENSION IN MOORING LINE 2

The fairlead tension predicted by Flexcom shows good agreement with the other software tools. In terms of the tension variation as a function of time, Flexcom's results show a multi-frequency style response. This characteristic is shared with most of the software tools, which typically consider the structural dynamics of the mooring lines in addition to the wave excitation forces. Some tools show a simpler response as these models are using a quasi-static approach for modelling the mooring lines. Robertson et al. (2014) note that "*although the mooring loads may differ vastly between these two approaches, the mean values are similar and tend to have no significant effect on the overall dynamic response of the structure*".



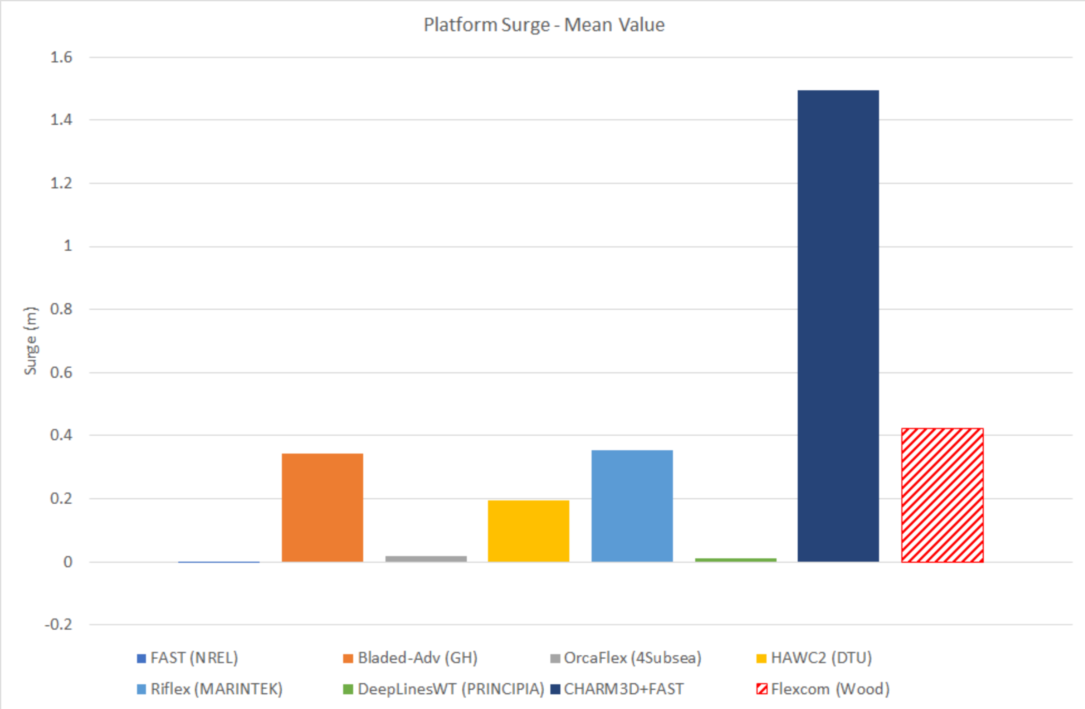
OC4 P2 LC2.2 Irregular waves

Results for irregular wave cases are presented in terms of mean and variance of key parameters rather than full time histories. To facilitate comparisons, Robertson et al. (2014) focus on the key outputs of platform surge, platform pitch and bending moment at the base of the tower. Generally speaking, Flexcom shows good agreement with the other software tools. Note that results from Bladed (GH) were not found in the IEA Wind Task 30 results library, so we have used results from Bladed Advanced Hydro Beta (GH) instead in this comparison.

MEAN RESPONSE

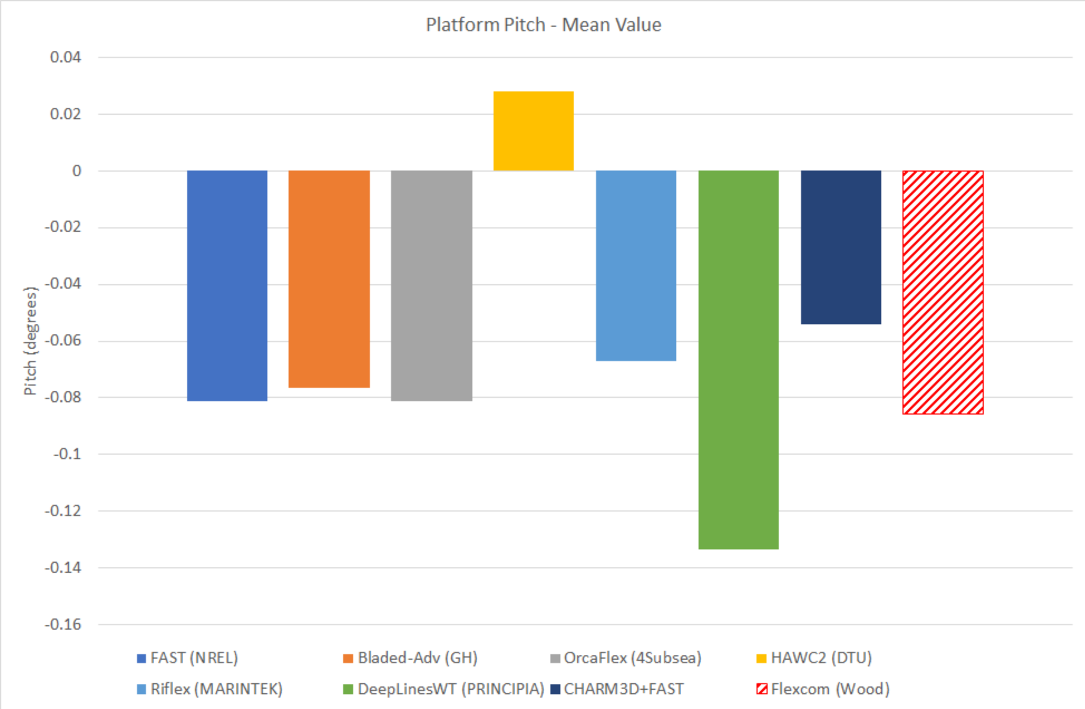
Platform Surge

The mean surge response shows considerable variation across the different software products. As noted by Robertson et al. (2014), it seems likely that the discrepancy is caused by different approaches to modelling drift forces (as mentioned in OC4 P2 LC2.1).



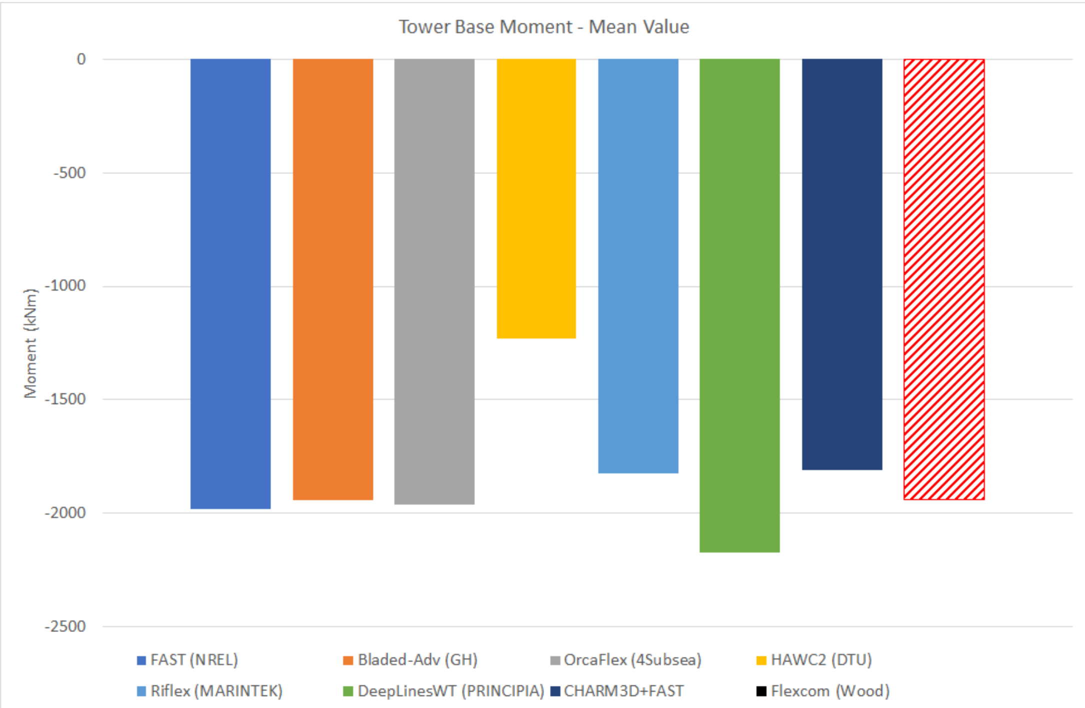
Platform Pitch

Flexcom's mean pitch response shows good agreement with the other software tools, particularly considering the variation across the different products. It is worth noting that the mean pitch is very small (<.0.1 degrees for most participants), so any discrepancies are extremely minor.



Tower Base Moment

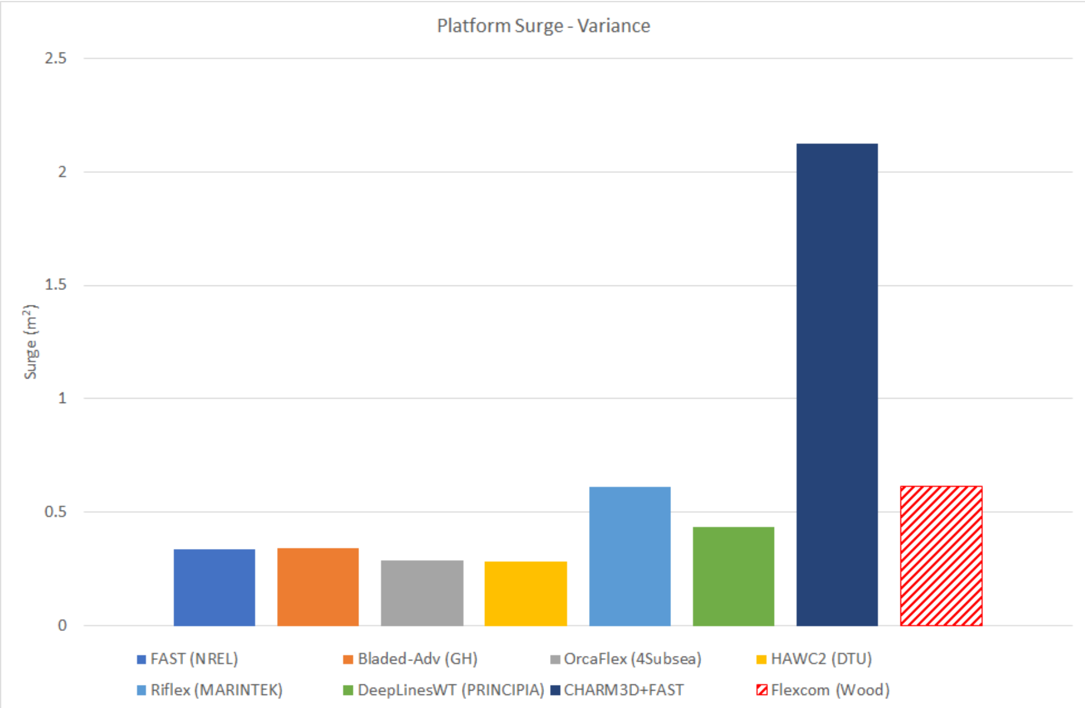
Flexcom's mean tower base moment shows good agreement with the other software tools.



VARIANCE OF RESPONSE

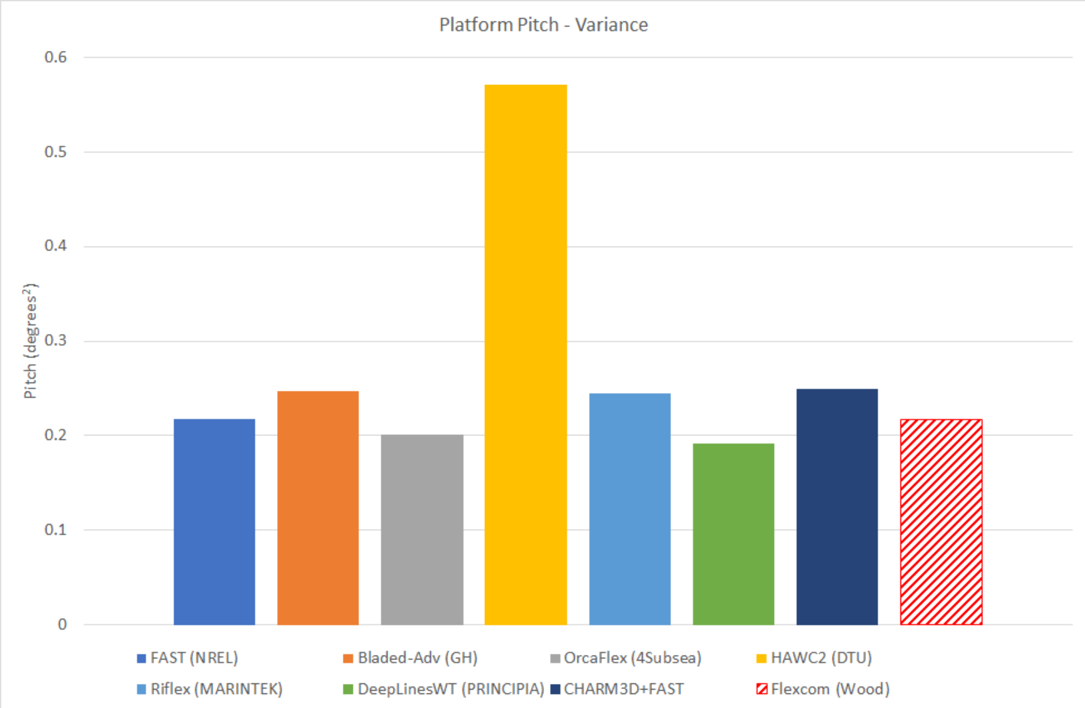
Platform Surge

The variance in surge shows some variation across the different software products, again possibly caused by different approaches to modelling drift forces. Results from Charm3D+FAST (ABS) show a surge variance which is larger than the other software tools.



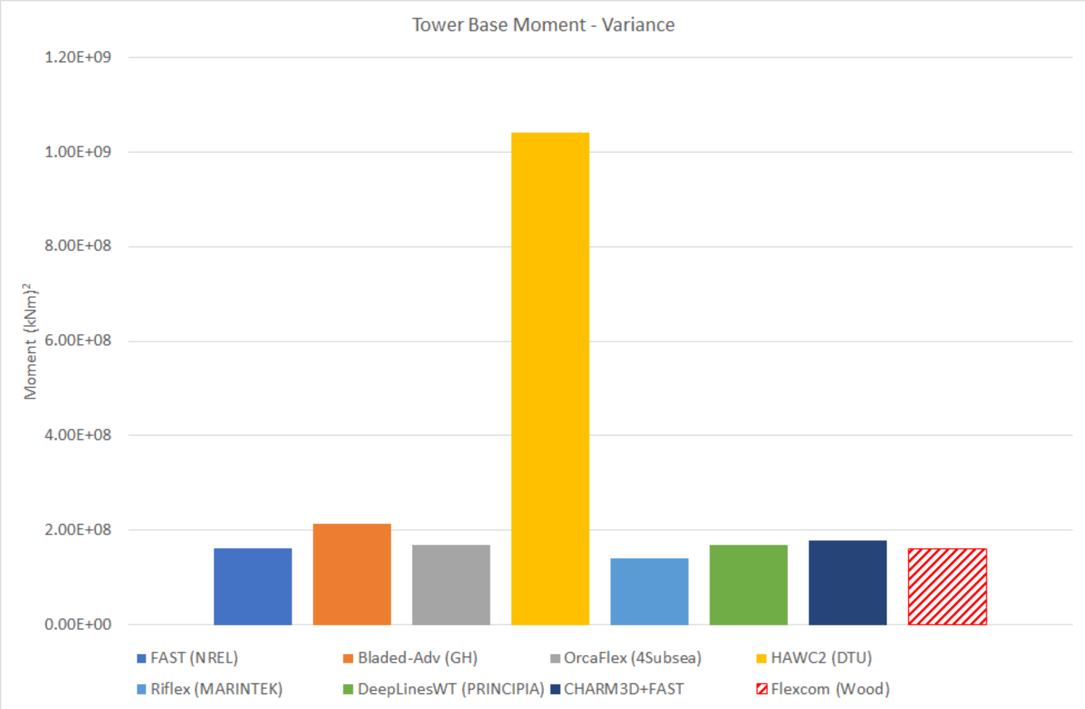
Platform Pitch

The variance in pitch shows reasonable agreement across the different software products, with the exception of HAWC2 (DTU) where the pitch variance appears larger than the others. Again it is worth noting that the pitch variance is quite small (<.0.25 degrees for most participants), so any discrepancies are extremely minor.



Tower Base Moment

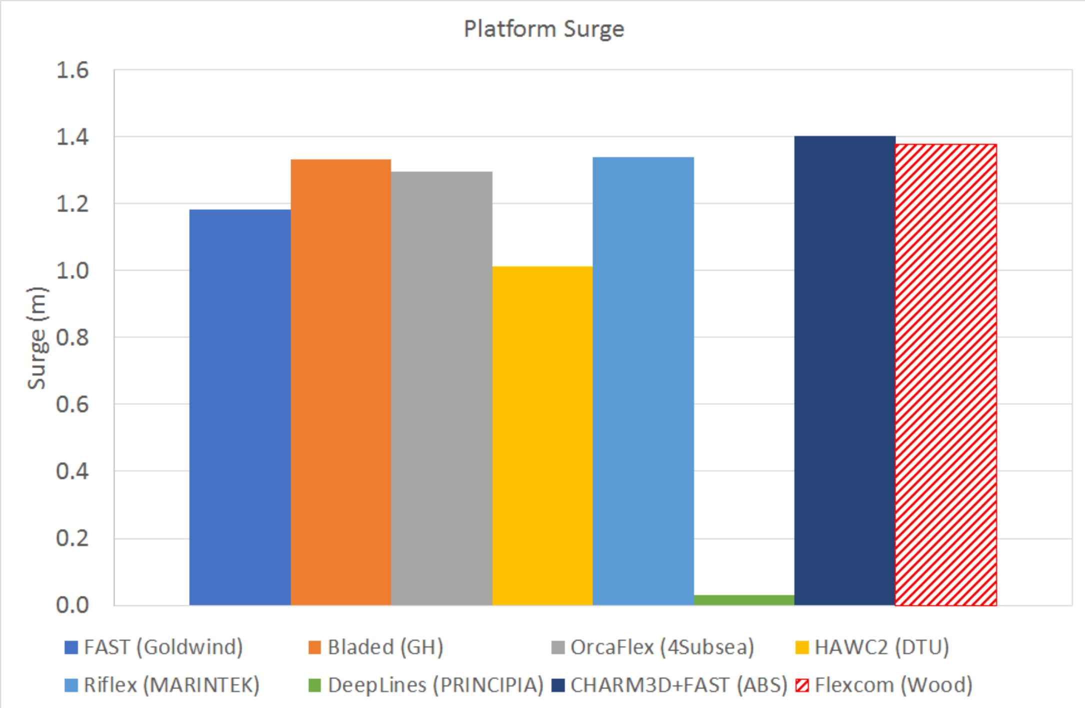
The variance in tower base moment shows reasonable agreement across the different software products, with the exception of HAWC2 (DTU) which appears to be inconsistent with the others. Robertson et al. (2014) suggest that this probably results from an incorrectly prescribed axis definition for the output.



OC4 P2 LC2.3 Current only

PLATFORM SURGE

Flexcom's surge response shows good agreement with the other software tools. Given that surge motions are dependent on (time-invariant) viscous drag loads, closer agreement might have been expected between all the software tools, as this is quite a simple test case. Differences in mooring line modelling approaches (as mentioned in OC4 P2 LC2.1) may be partially responsible. Note that results from DeepLinesWT (Principia) are based on potential flow theory only, and so are missing the viscous drag terms which are central to this test (see note on DeepLines Wind results in OC4). Note also that results from FAST (NREL) were not found in the IEA Wind Task 30 results library, so we have used results from FAST (Goldwind) instead in this comparison.

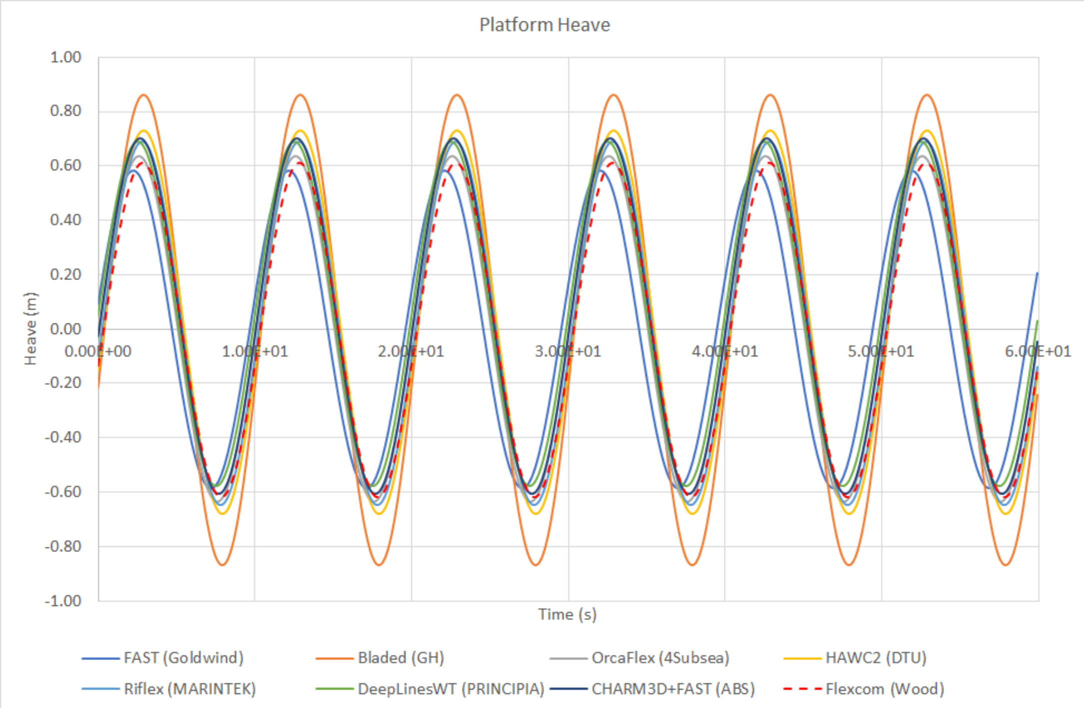


OC4 P2 LC2.4 Current and regular waves

This load case combines elements from OC4 P2 LC2.1 (which includes regular wave loading only) and OC4 P2 LC2.3 (which includes current loading only). Similar trends to that observed for these simpler load cases are also evident here. Note that results from FAST (NREL) were not found in the IEA Wind Task 30 results library, so we have used results from FAST (Goldwind) instead in this comparison.

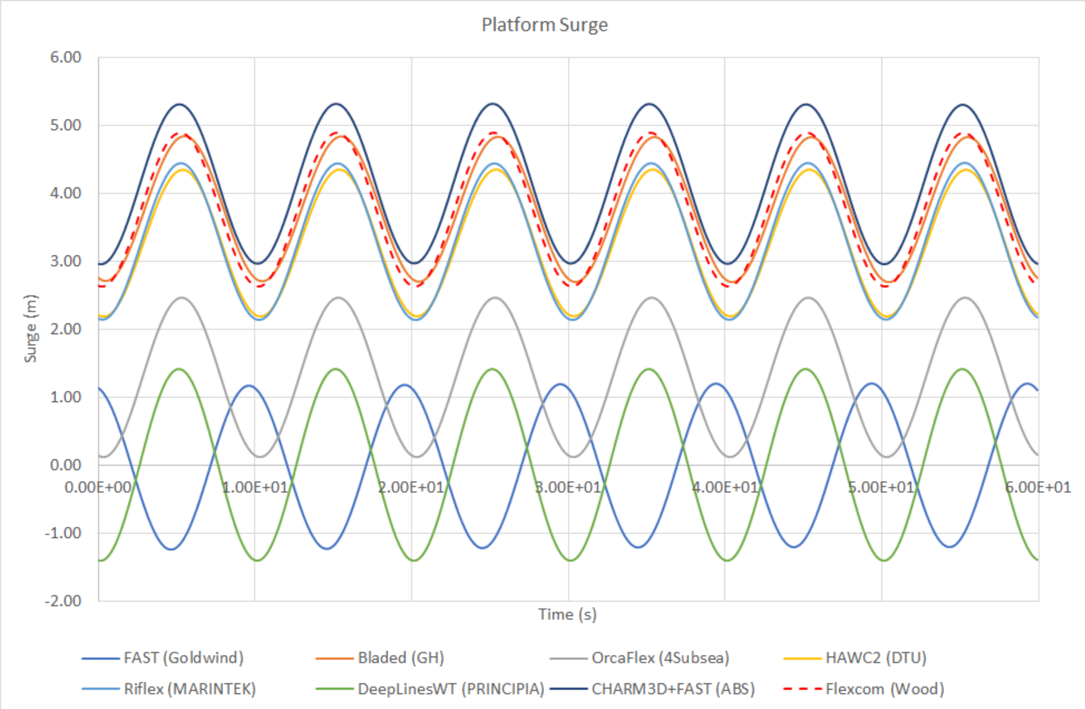
PLATFORM HEAVE

Flexcom's heave response shows close agreement with the other software tools, in terms of response amplitude, period and phase. Results from Bladed (GH) results show a heave response which is slightly larger than the other software tools.



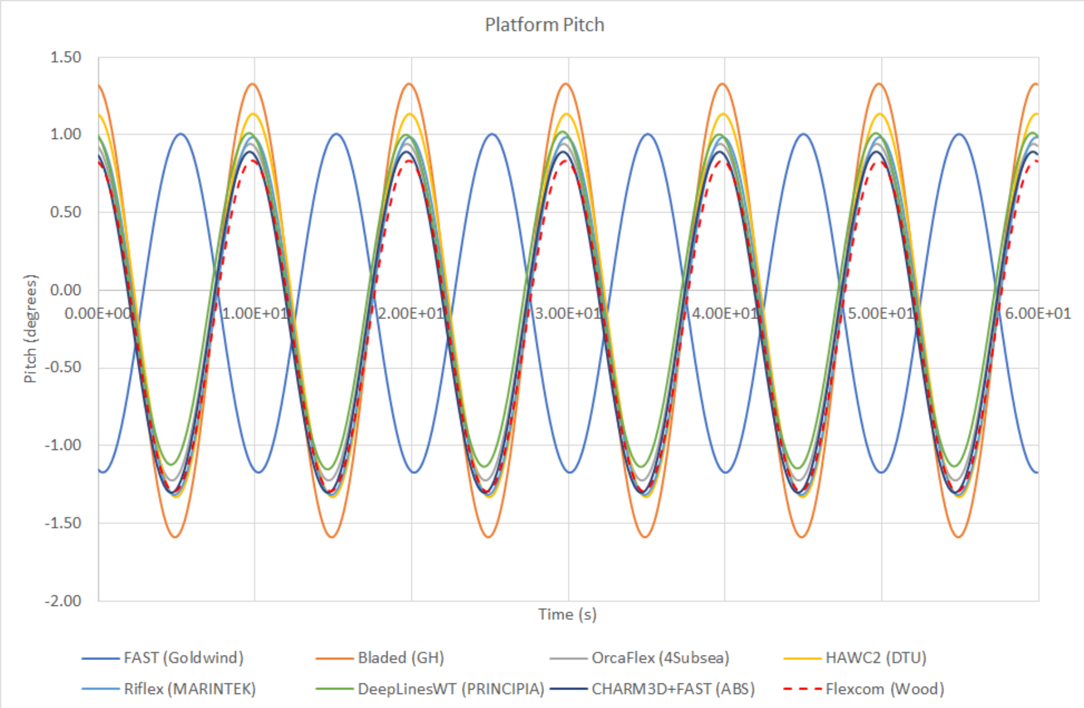
PLATFORM SURGE

The surge response shows considerable variation across the different software products. Some of this may be attributed to the modelling of drift forces (refer to OC4 P2 LC2.1 for further information). Note however that results from FAST (Goldwind), Orcaflex (4Subsea) and DeepLinesWT (Principia) show a mean surge position which is lower than the other software tools. So there is a clear discrepancy between two groups of software in how mean surge is computed when both wave and current loads are included, as broadly speaking, surge motions showed reasonable agreement for waves only (OC4 P2 LC2.1) and current only (OC4 P2 LC2.3). This issue is not discussed in Robertson et al. (2014) and it is unclear as to what is causing it.



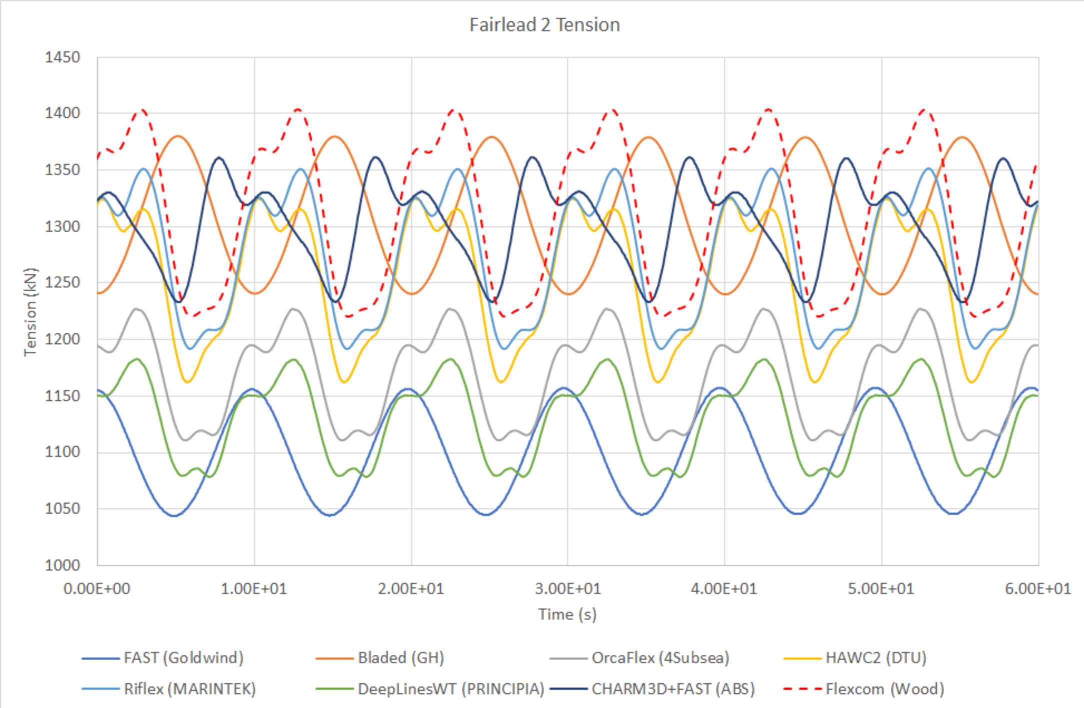
PLATFORM PITCH

Flexcom's pitch response shows close agreement with the other software tools, in terms of response amplitude, period and phase. Results from Bladed (GH) show a pitch response which is slightly larger than the other software tools. Results from FAST (NREL) appear out of phase with the other tools, although this discrepancy is most likely caused by an inconsistency in presentation only.



FAIRLEAD TENSION IN MOORING LINE 2

The fairlead tension shows considerable variation across the different software products, consistent with the discrepancies in surge discussed above. The fairlead tension predicted by Flexcom is at the upper bound of the software tools, a trend which is consistent with the surge response shown above. Differences in mooring line modelling approaches (as mentioned in OC4 P2 LC2.1) also contribute to tension variations.



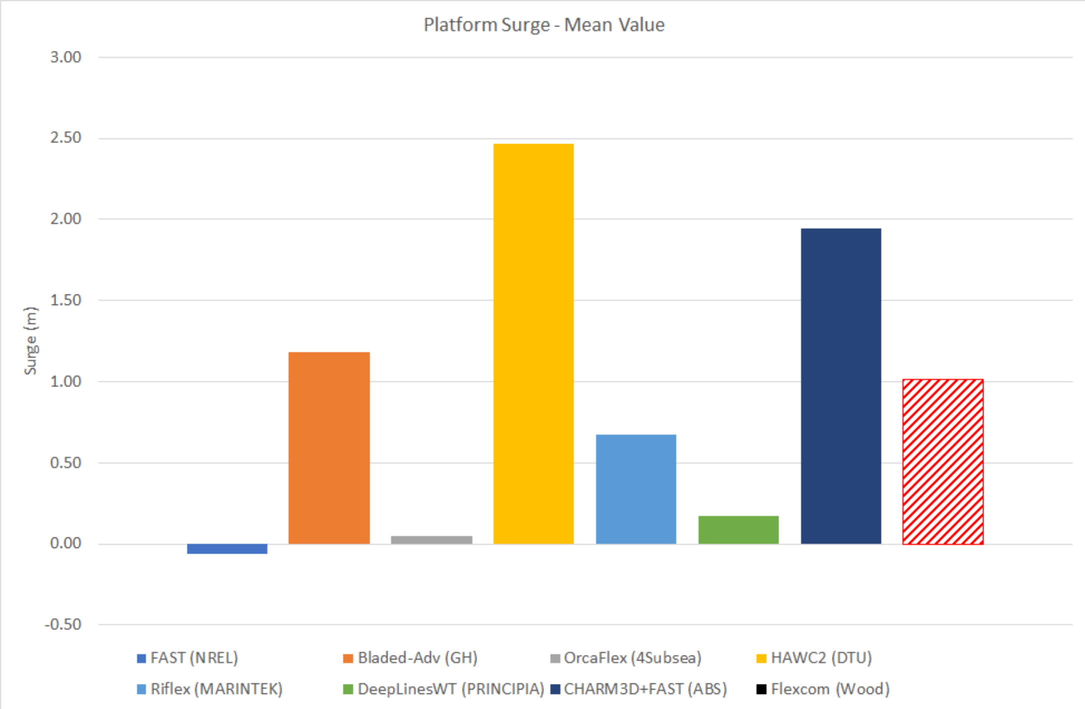
OC4 P2 LC2.5 50-year extreme wave

This load case is very similar to OC4 P2 LC2.2, but the wave loading is much more severe in this case (Hs=15m rather than 6m). It is not discussed in Robertson et al. (2014) but generally speaking, similar trends to that observed in OC4 P2 LC2.2 are also evident here. Results are presented below, via mean and variance, for platform surge, platform pitch and bending moment at the base of the tower. Note that results from Bladed (GH) were not found in the IEA Wind Task 30 results library, so we have used results from Bladed Advanced Hydro Beta (GH) instead in this comparison.

MEAN RESPONSE

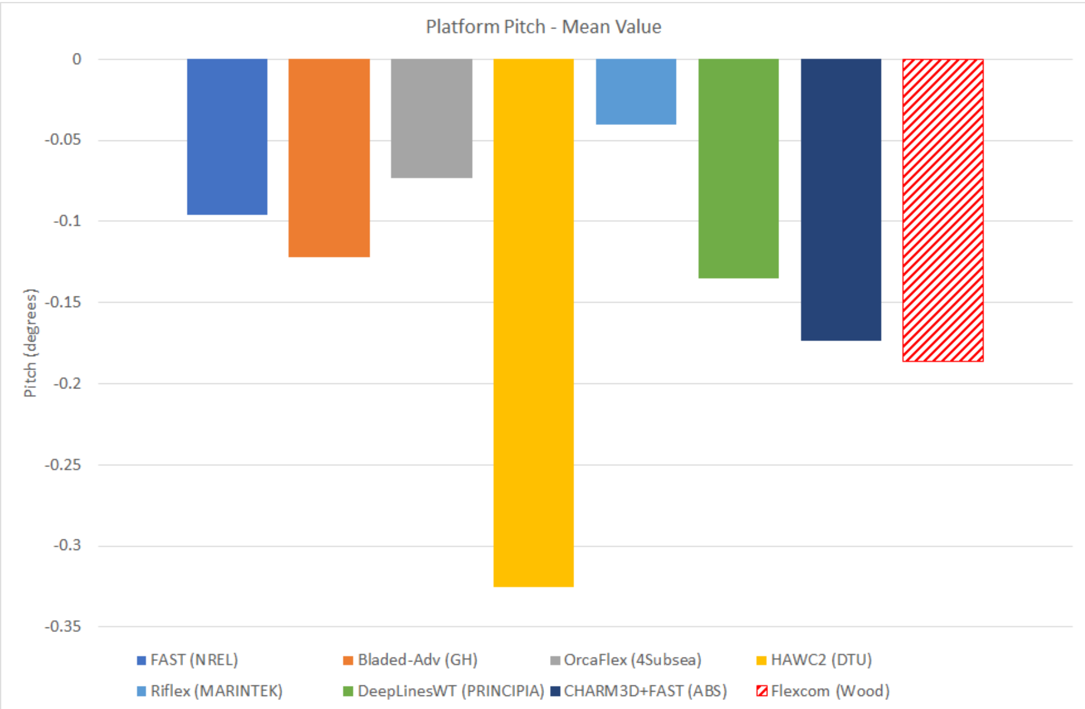
Platform Surge

The mean surge response shows considerable variation across the different software products, depending on how drift forces are being modelled (if at all). Refer to OC4 P2 LC2.1 for further information.



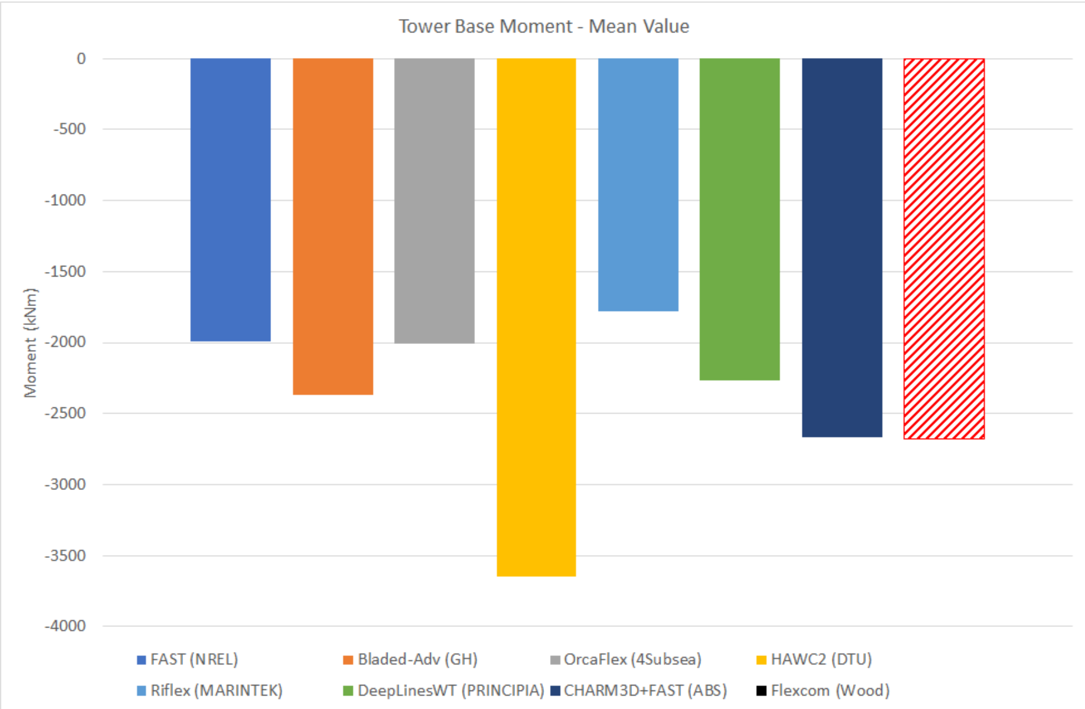
Platform Pitch

The mean pitch response shows some variation across the different software products, although the mean pitch is relatively minor.



Tower Base Moment

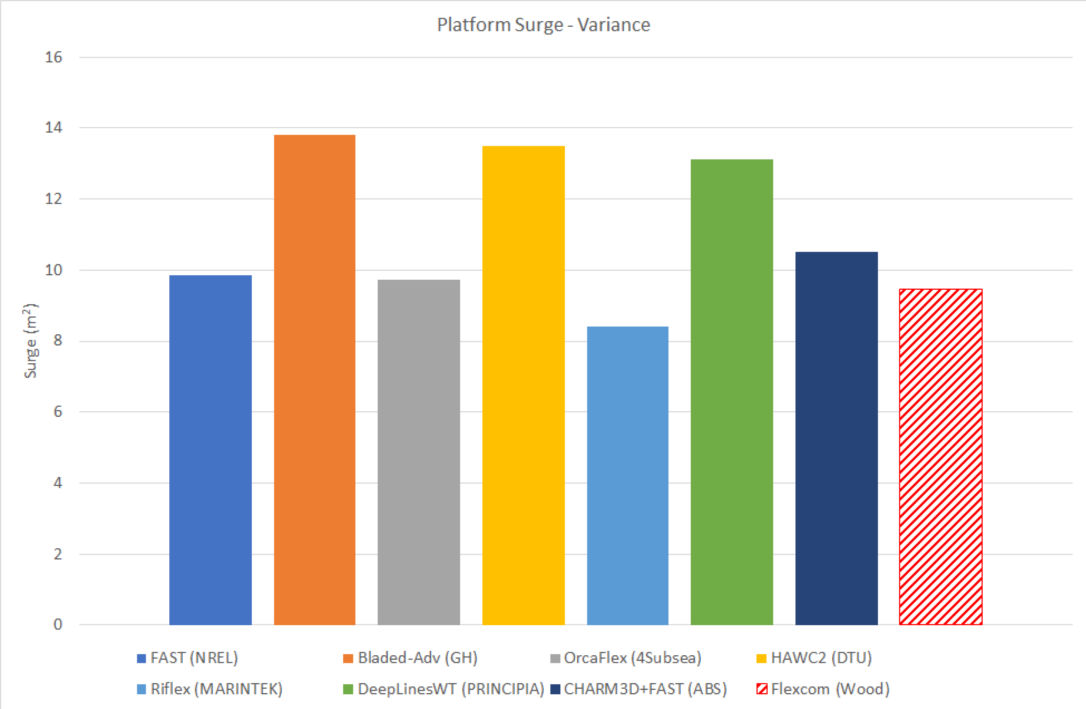
The mean tower base moment shows reasonable agreement across the different software products.



VARIANCE OF RESPONSE

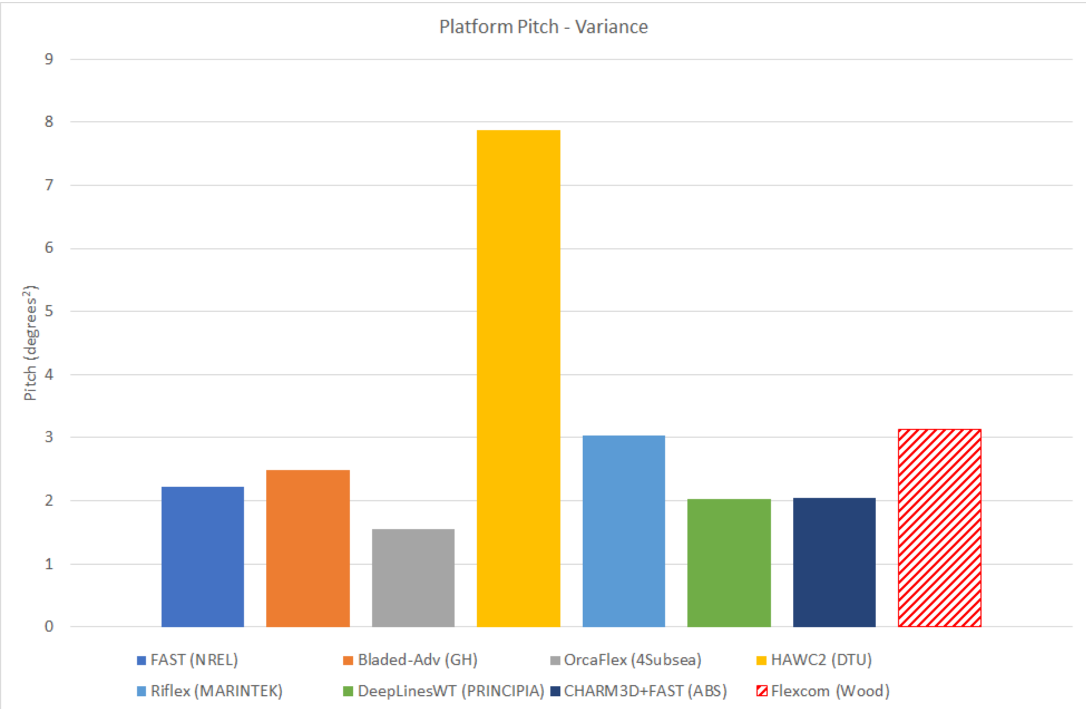
Platform Surge

The variance in surge shows reasonable agreement across the different software products.



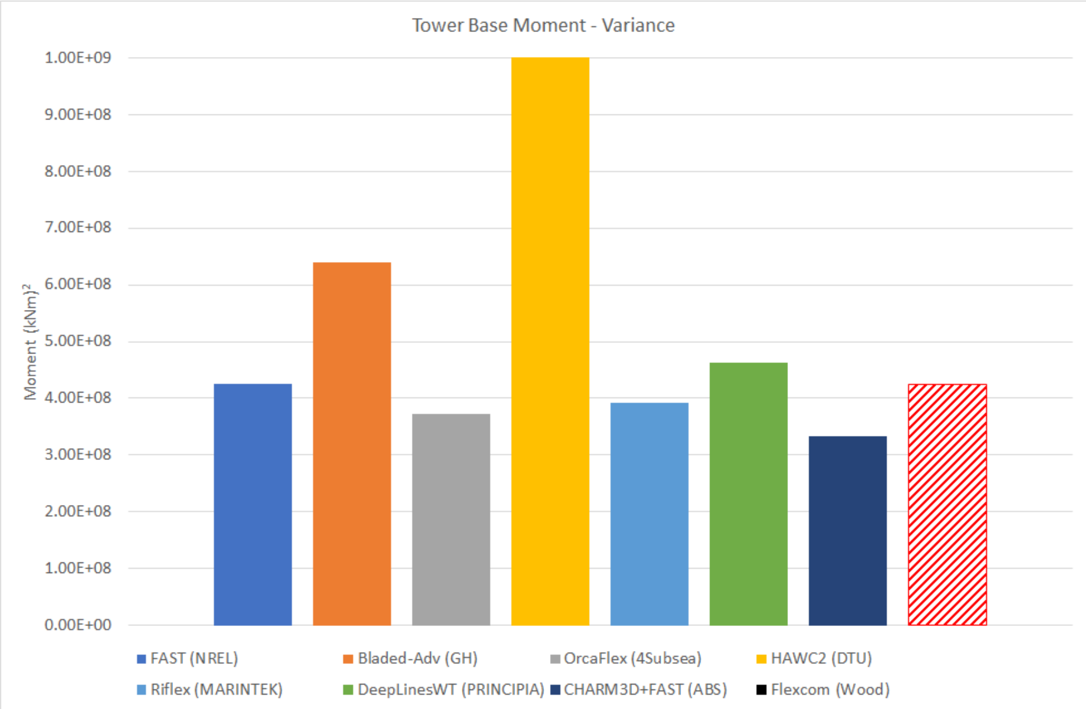
Platform Pitch

The variance in pitch shows reasonable agreement across the different software products, with the exception of HAWC2 (DTU) which appears to be inconsistent with the others.



Tower Base Moment

The variance in tower base moment shows reasonable agreement across the different software products, with the exception of HAWC2 (DTU) which appears to be inconsistent with the others. Although the full extent is not shown below, this is an order of magnitude larger than the others, and based on OC4 P2 LC2.2, it probably results from an incorrectly prescribed axis definition for the output.



OC4 P2 LC2.6 RAO estimation, no wind

Flexcom input files are available for this load case and simulations results are available in the form of random time histories. These need to be converted to RAOs (response amplitude operators) or PSDs (power-spectral densities) before results from Flexcom may be compared to official results from OC4. Once this process is complete, RAO/PSD comparisons will be uploaded here, including:

- Wave height

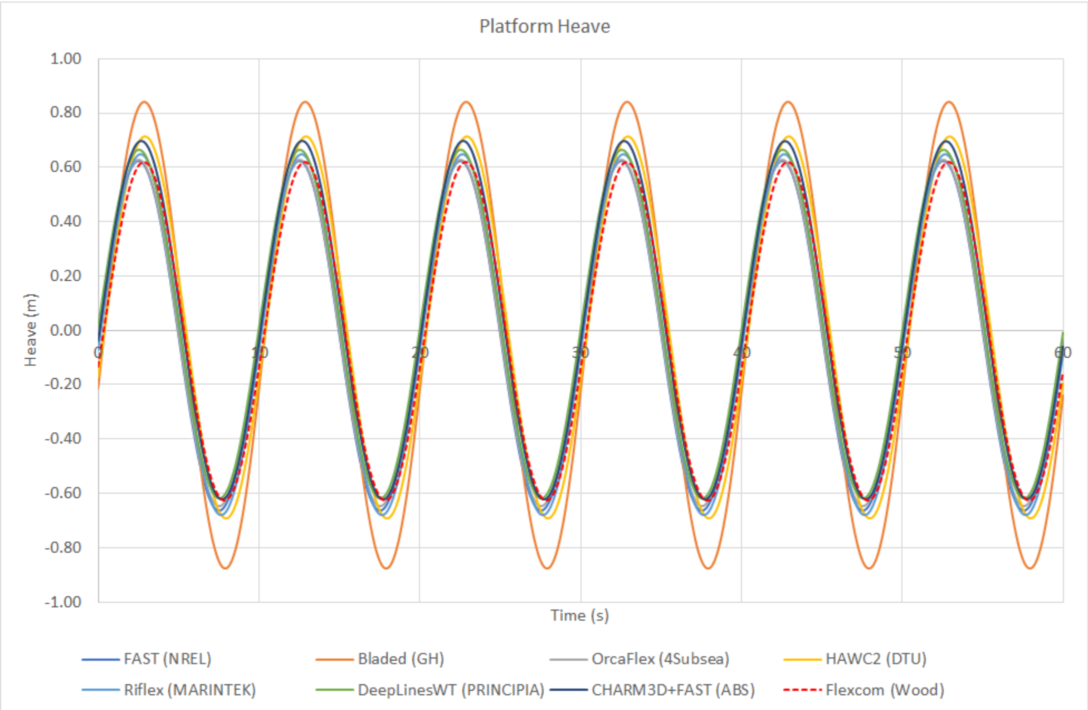
- Tower Base Moment
- Fairlead Tension in Mooring Line 2
- Platform surge
- Platform heave
- Platform pitch

OC4 P2 LC3.1 Deterministic, below rated

This load case is similar to OC4 P2 LC2.1, which does not include wind loading. Similar trends to that observed for OC4 P2 LC2.1 are also evident here.

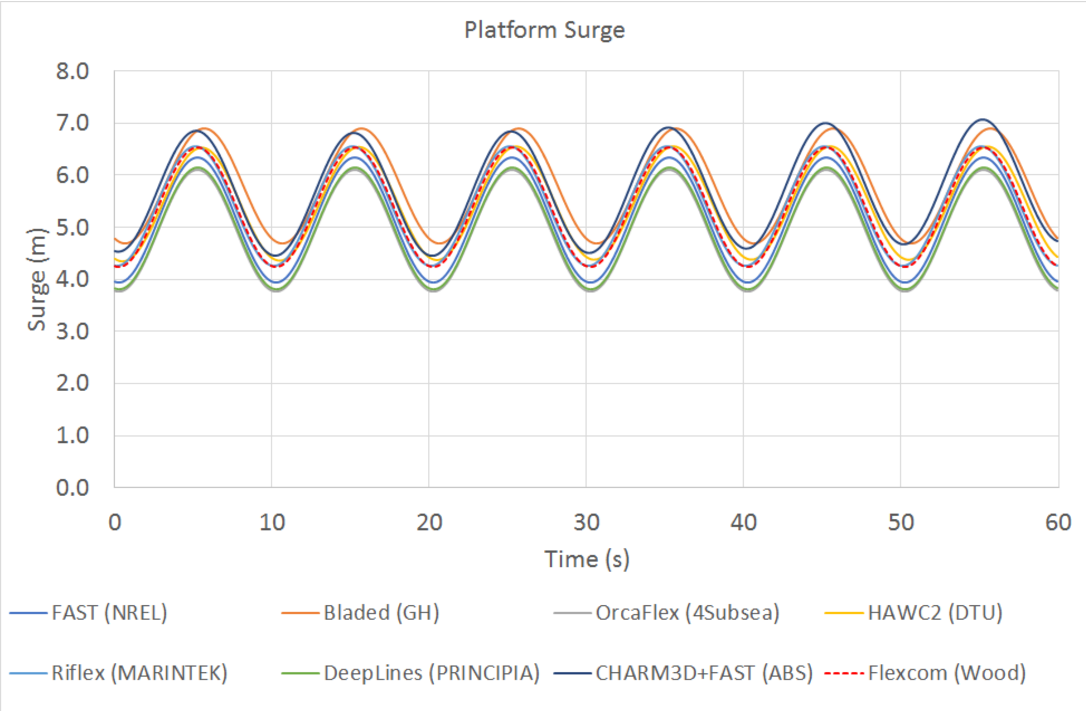
PLATFORM HEAVE

Flexcom's heave response shows close agreement with the other software tools, in terms of response amplitude, period and phase. Results from Bladed (GH) show a heave response which is slightly larger than the other software tools.



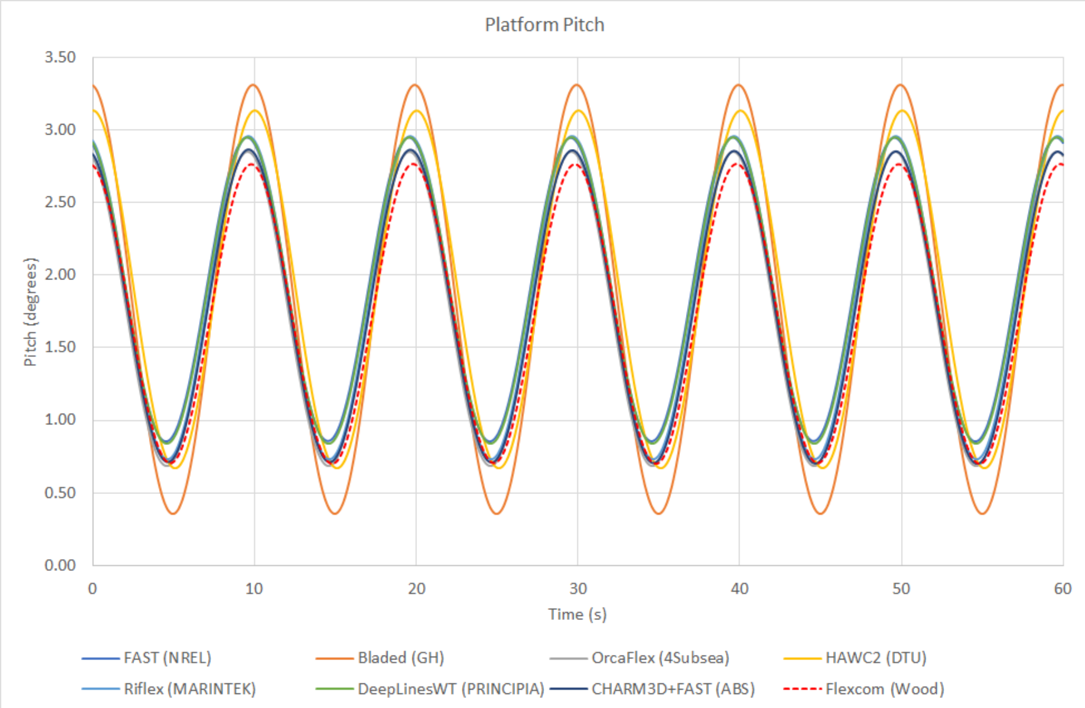
PLATFORM SURGE

Flexcom's surge response shows close agreement with the other software tools, in terms of response amplitude, period and phase. Although there is some variation across the different software products (depending on how drift forces are being modelled - refer to OC4 P2 LC2.1 for further information), the inclusion of wind tends to improve the correlation in surge results. This is because the wind thrust force is much larger than the hydrodynamic drift force, and hence the mean surge offset is significantly greater when wind is included.



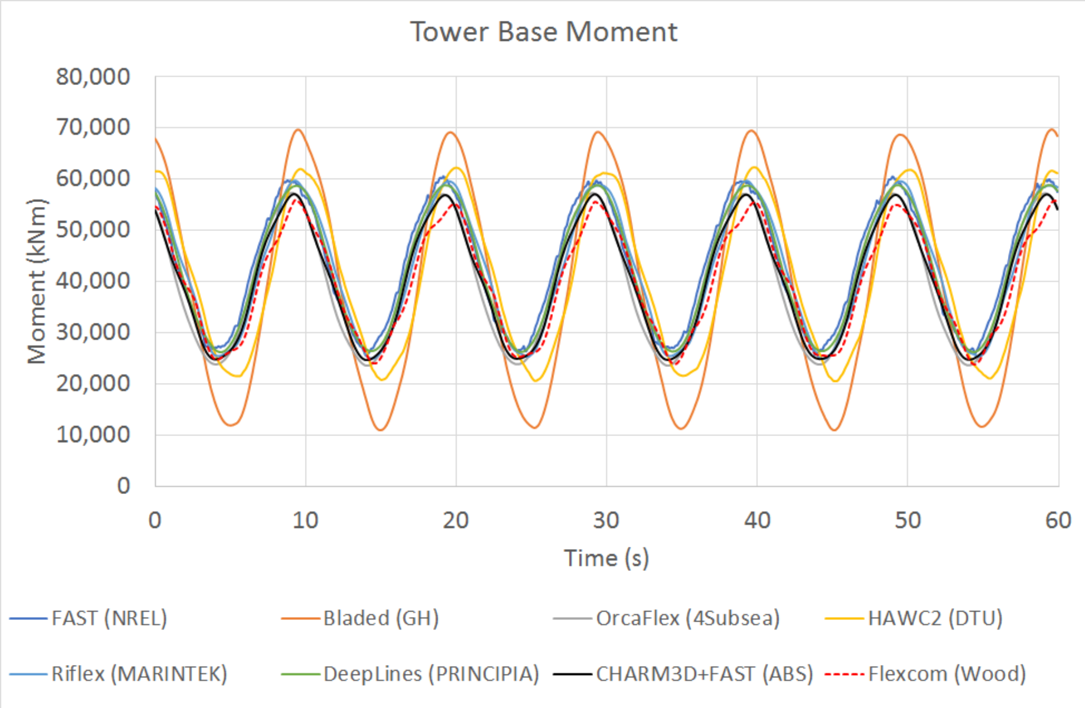
PLATFORM PITCH

Flexcom's pitch response shows good agreement with the other software tools, although the mean pitch is slightly lower than many of the other tools. Results from Bladed (GH) show a pitch response which is slightly larger than the other software tools.



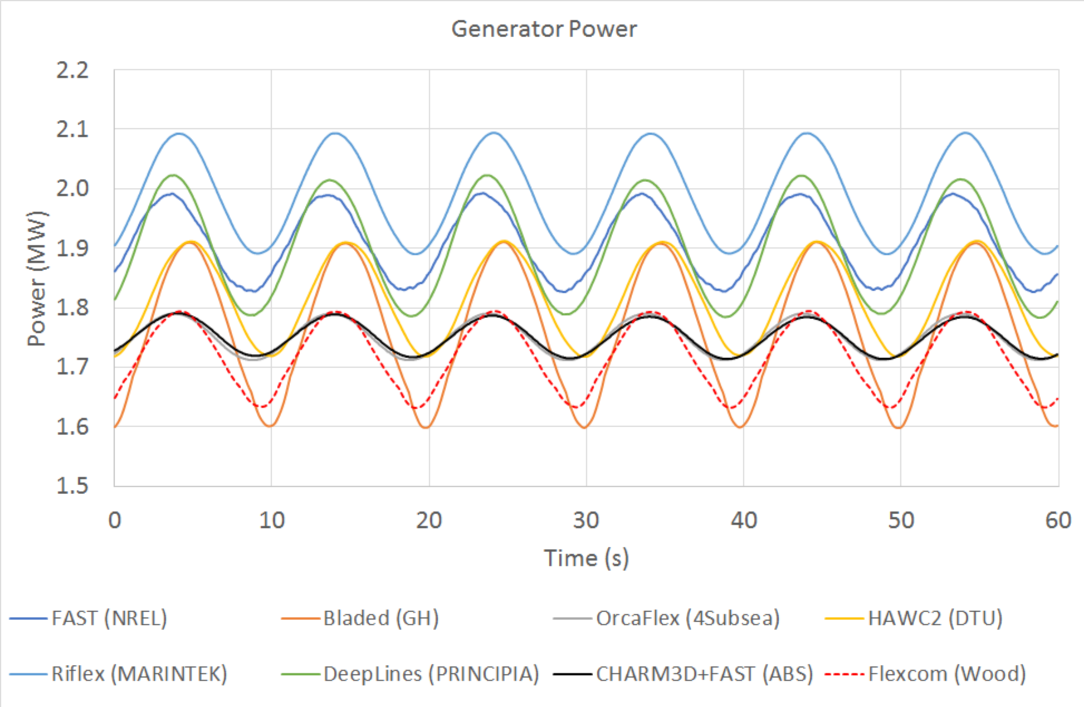
TOWER BASE MOMENT

Results from Flexcom show good agreement with the other software tools for bending moment at the base of the tower. The moments predicted by Flexcom are slightly lower than the other tools - it is believed this is related to the generator power discussed in the next paragraph. Results from Bladed (GH) show moment variations which are larger than the other software tools.

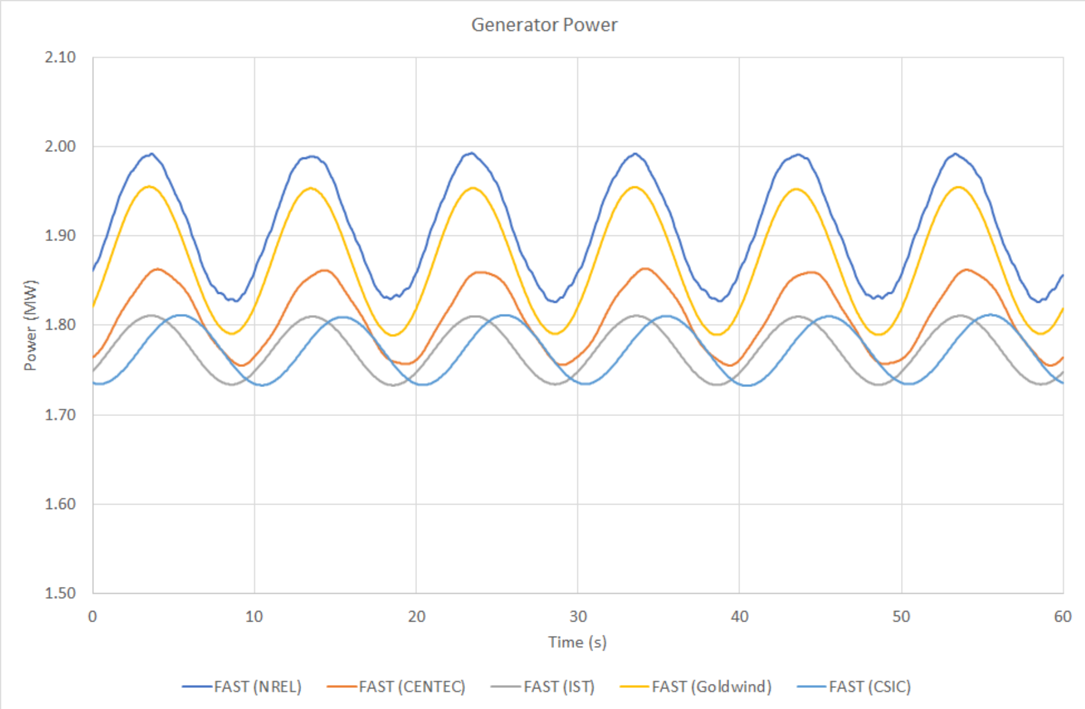


GENERATOR POWER, GENERATOR TORQUE AND ROTOR SPEED

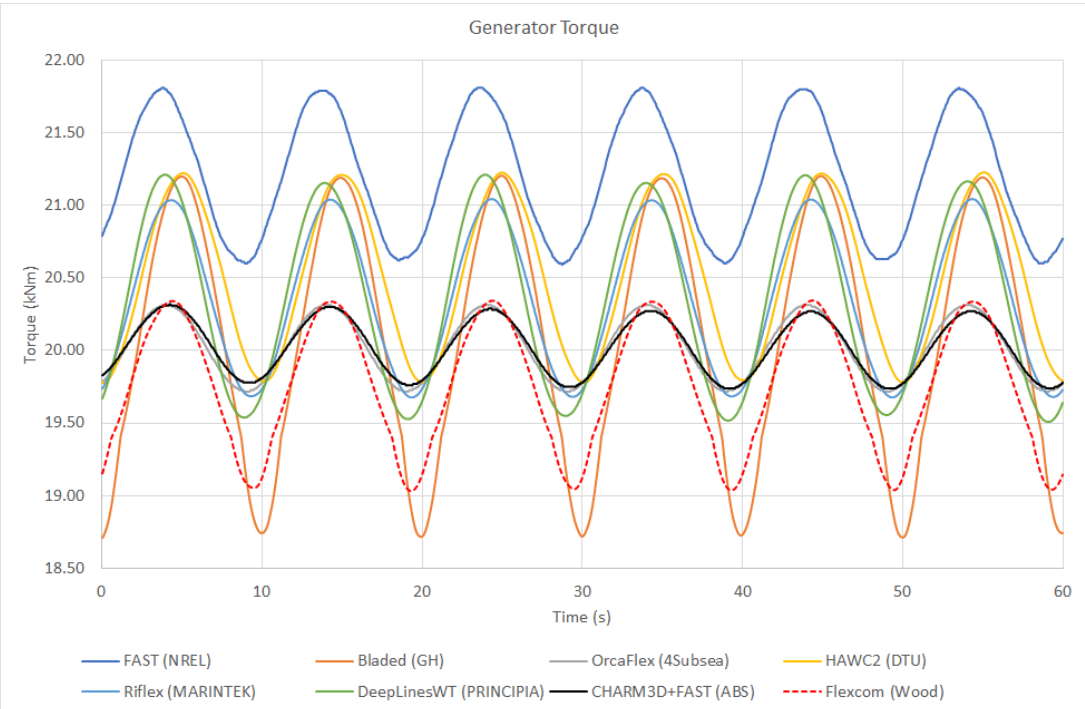
The generator power shows considerable variation across the different software products, with the mean power of the OC4 software tools considered ranging from 1.75MW (OrcaFlex/4Subsea, CHARM3D+FAST/ABS) to 1.99MW (Riflex/MARINTEK). The generator power predicted by Flexcom is slightly lower than most of the other software tools - Flexcom's mean power is 1.72MW, 6.6% below the average.

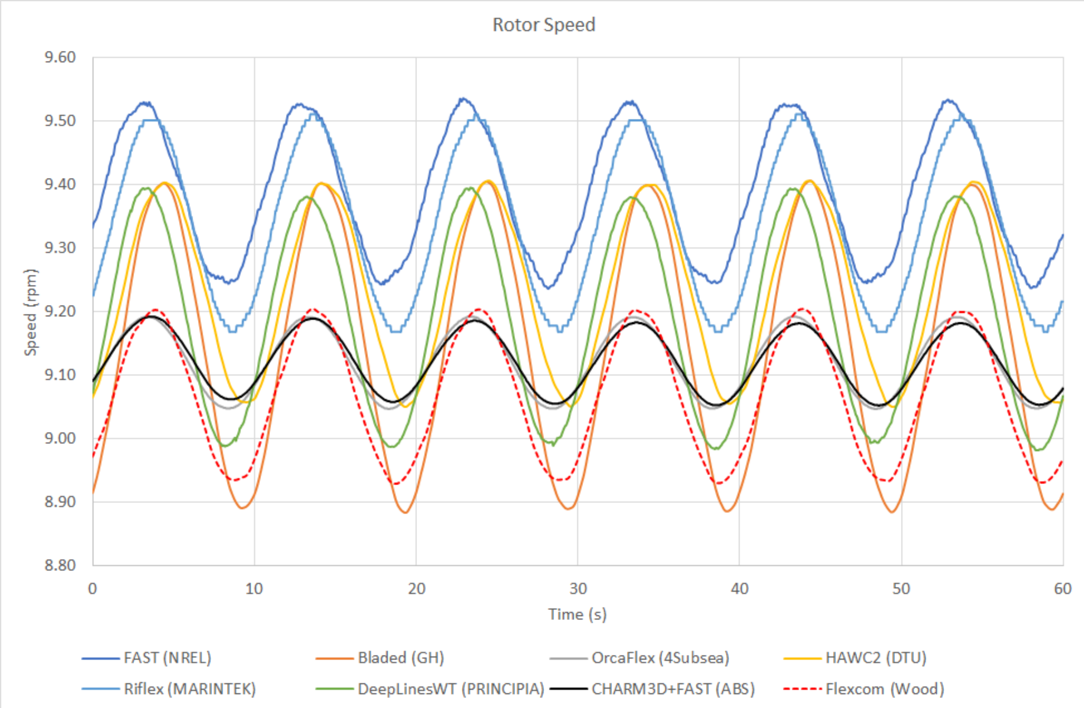


Generator power is not discussed in Robertson et al. (2014) but given that this is such a fundamental output for a wind turbine, closer agreement might have been expected, particularly given that the same control system (originally developed for the OC3-Hywind spar) was provided to all participants in form of a DLL which could be linked into the individual software products. For example, even amongst the participants who use FAST, the average power varies from 1.77MW (IST), 1.77MW (CSIC), 1.81MW (CENTEC), 1.87MW (Goldwind) to 1.91MW (NREL).



The variations in generator power are also evident in the primary variables of generator torque and rotor speed.



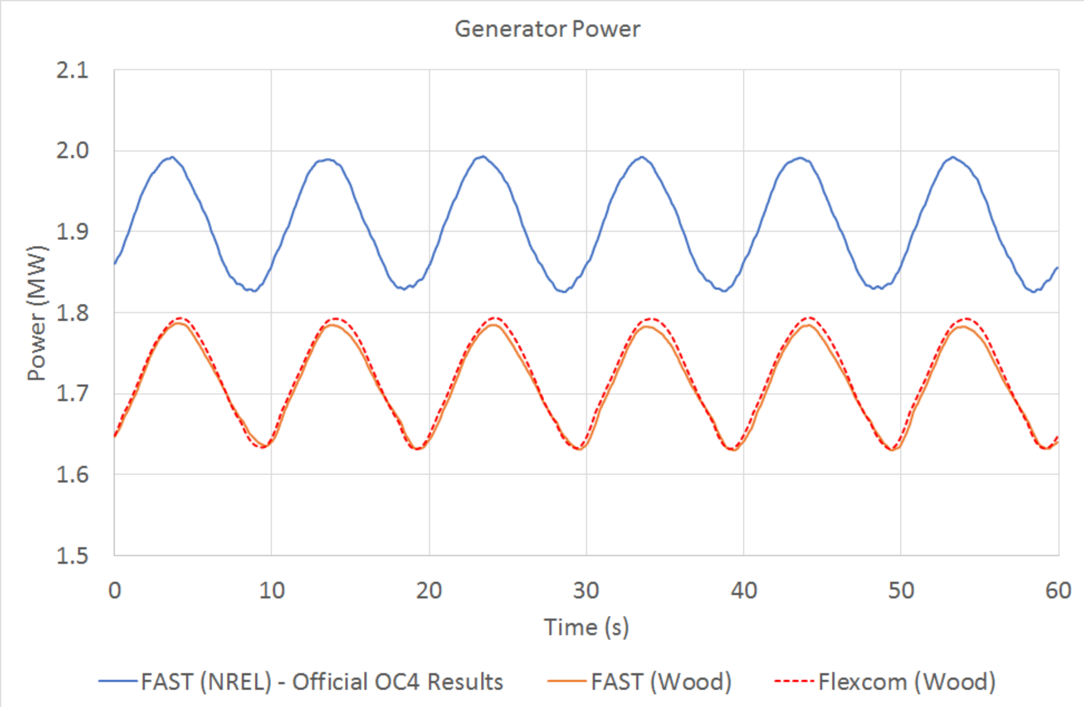


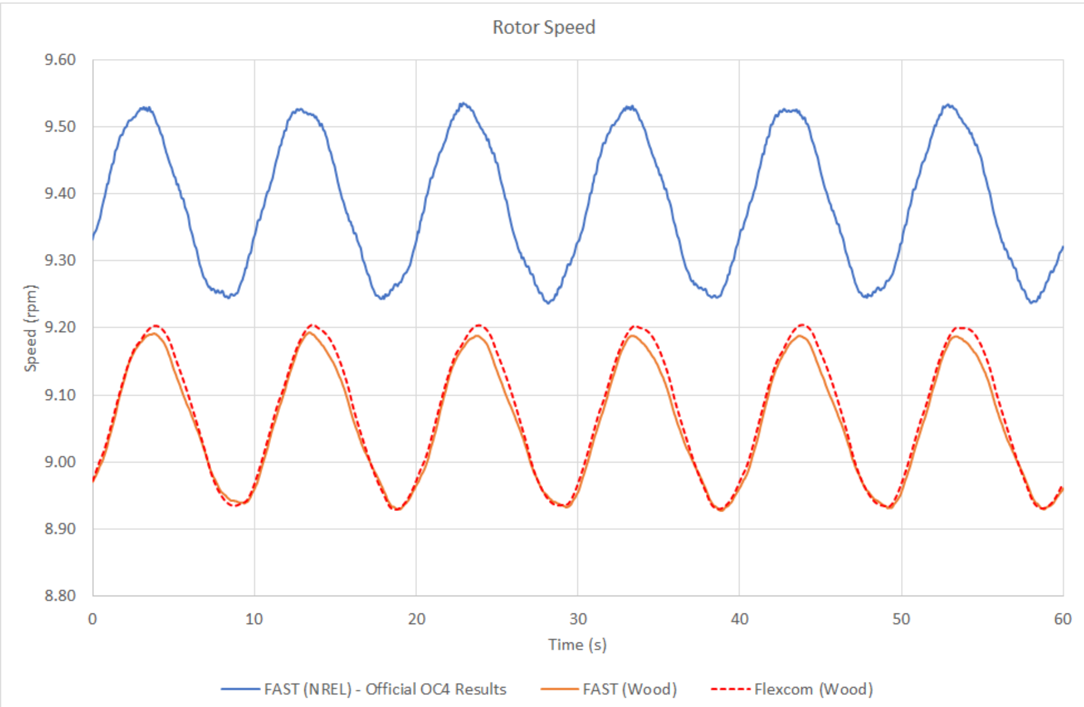
Discrepancies in generator power may be caused by several factors, including...

1. Variations in the aerodynamic theories underpinning each software product
2. Inconsistencies in user selection of aerodynamic modelling options within the software
3. Participants using different versions of the same software product
4. Discrepancies in computed platform motions which influence aerodynamic performance

Given that Flexcom uses the AeroDyn and ServoDyn modules directly from FAST, any difference in generator power is most likely to stem from the last two sources.

In an effort to better understand the discrepancies, we attempted to recreate the official OC4 results presented by NREL for OC4 P2 LC3.1. We downloaded an official release of FAST, Version 8.16, which appeared to be the latest version available at the time of writing. Although the sample input files provided by NREL corresponded to OC4 P2 LC3.7 (these were the only input files published by NREL), it was a simple task to recreate the input files required for OC4 P2 LC3.1, by simply adjusting the wave loading from a white noise spectrum to a regular Airy wave. We found that key output parameters, such as mean generator power and rotor speed, predicted by FAST V8.16 to be considerably lower than those presented in Robertson et al. (2014). When these discrepancies were queried with NREL via their online user forum, NREL advised that "*the CertTest model* (i.e. the more recent version) *uses AeroDyn15 with BEMT* (i.e. blade element momentum theory)*, while the original OC4 simulation used AeroDyn14 with DYNIN* (i.e. dynamic inflow)", and that "*for the original OC4 simulation the mooring system was modeled through a quasi-static approach and the current Test25 model* (i.e. the more recent version) *uses a dynamic mooring line model*", and consequently that "*due to these different modelling approaches it is expected to observe some differences in predicted system response between the two models*". As the Flexcom results show much closer agreement with FAST V8.16, it is concluded that the much of the difference between Flexcom and the official OC4 results presented by NREL stem from different versions of the aerodynamic modelling software. The remaining discrepancy may be attributed to slightly different platform motions being predicted by Flexcom and FAST. Furthermore, results obtained by Flexcom appear to be consistent with expectations as the mean rotor speed is closer to 9rpm, which is consistent with the official load case definition.


Generator Power
Power (MW)
2.1
2.0
1.9
1.8
1.7
1.6
1.5
0
10
20
30
40
50
60
Time (s)
FAST (NREL) - Official OC4 Results
FAST (Wood)
Flexcom (Wood)

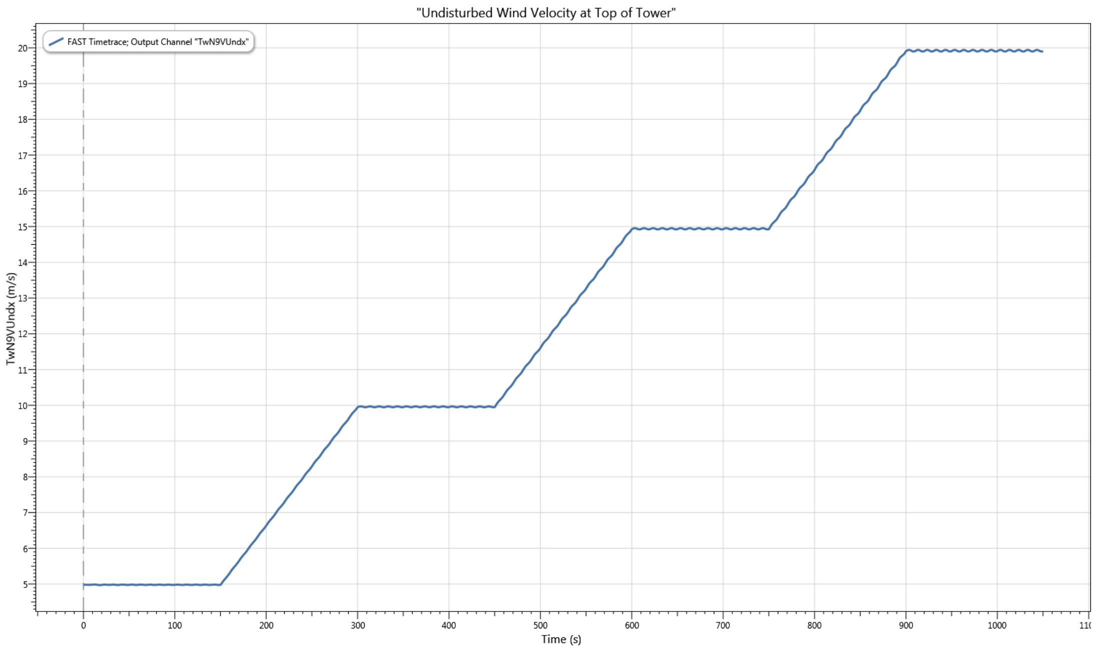

Rotor Speed
Speed (rpm)
9.60
9.50
9.40
9.30
9.20
9.10
9.00
8.90
8.80
0
10
20
30
40
50
60
Time (s)
FAST (NREL) - Official OC4 Results
FAST (Wood)
Flexcom (Wood)

OC4 P2 LC3.1 Modified (control system test)

This load case is not part of OC4 Phase II. It was added here in order to verify the correct operation of the control system in Flexcom. Fundamentally it is very similar to OC4 P2 LC3.1, but the wind speed in this load case varies over the course of the simulation, allowing us to examine the behaviour of the turbine. For comparison purposes, results predicted by an equivalent simulation performed with FAST are also included.

WIND CONDITIONS

A variable wind speed definition is used in order to test the control system. The wind speed at the hub increases from 5m/s, to 10m/s, to 15m/s, and finally to to 20m/s.



CONTROL SYSTEM

The following figures show the rotor speed, generator torque, blade pitch angle, and generator power. Very close agreement is demonstrated between Flexcom and FAST for all these key parameters.

At the beginning of the simulation, a low wind speed of 5m/s is applied, and the rotor speed quickly increases from its initial value of 0rpm. At approximately 60 seconds, the control system increases generator torque from 0kNm up to about 5.5kNm, the rotor speed reaches a steady state value of about 7.5rpm, and a small amount of generator power - about 0.4MW - is generated. These wind conditions (18kph/11mph) represent the cut-in phase of the turbine.

Between 150 seconds and 300 seconds, the wind speed is steadily increased from 5m/s to 10m/s. During the same period, the control system steadily increases generator torque to about 30.5kNm, the rotor speed steadily increases to about 11.3rpm, and the generator power increases to about 3.4MW.

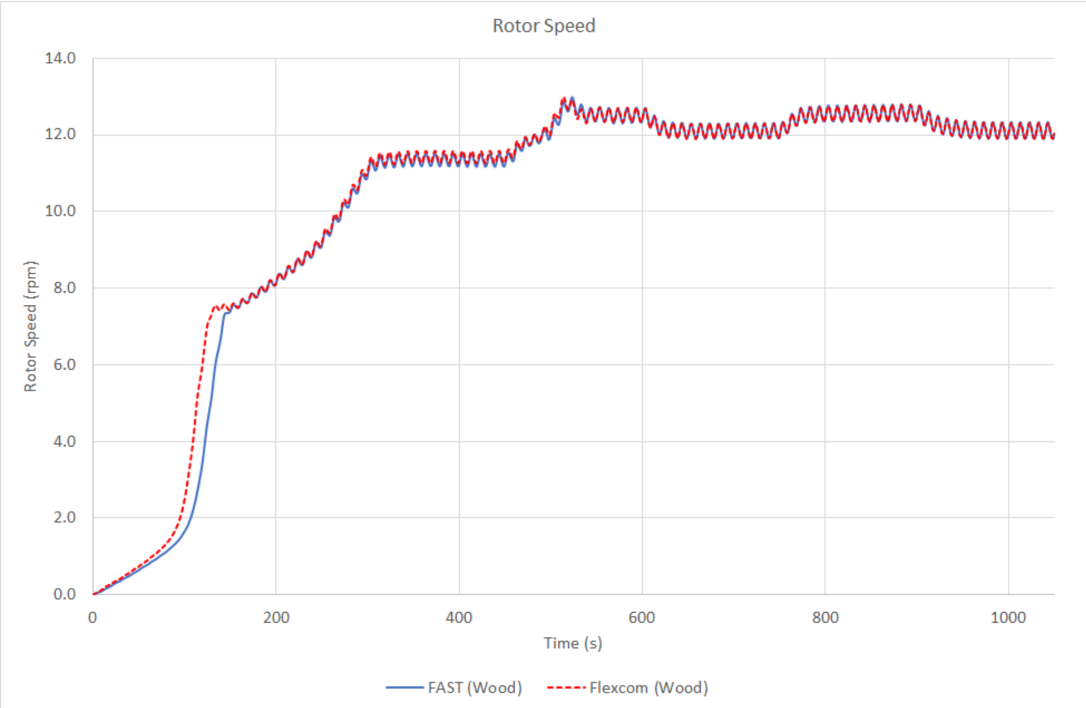
Between 300 seconds and 450 seconds, the wind speed is held constant at 10m/s, with all parameters hovering around mean steady-state values (torque ~30.5kNm, rotor speed ~11.3rpm, generator power ~3.4MW). Some minor fluctuations are caused by the wave induced motions of the platform, which change the tower pitch angle periodically, and affect the aerodynamic loading on the turbine. These wind conditions (36kph/22mph) keep the turbine operating below rated power.

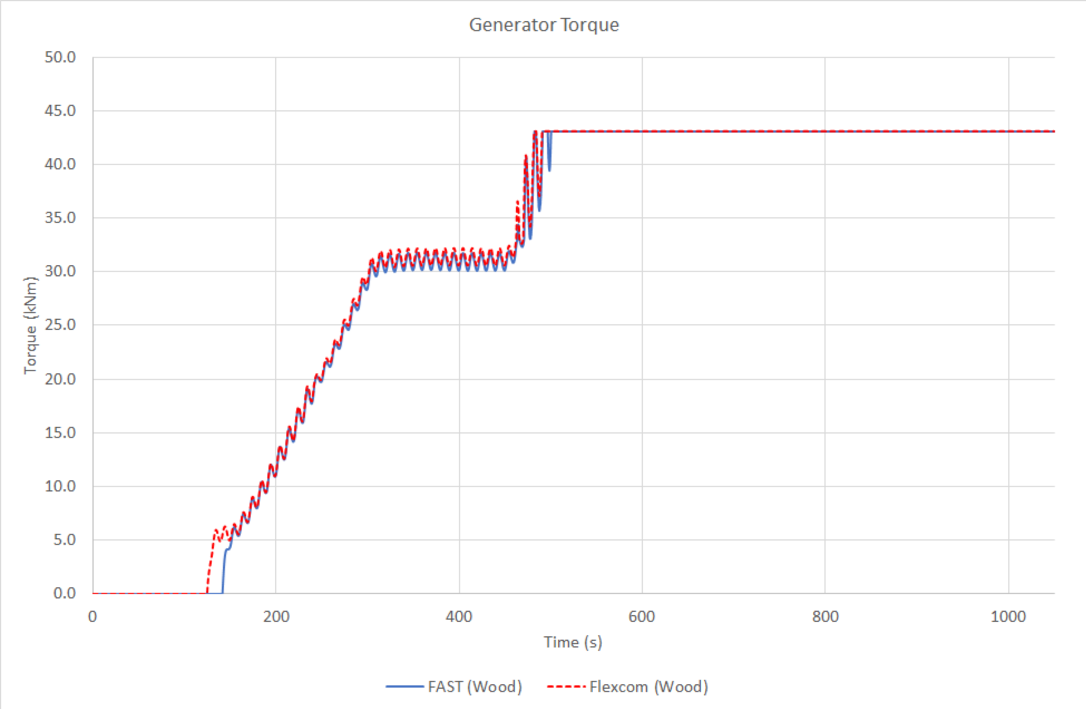
Between 450 seconds and 600 seconds, the wind speed is steadily increased from 10m/s to 15m/s. During this transition, the control system initially increases generator torque to about 43kNm, and the rotor speed increases to about 12.5rpm. At this point, the generator power increases to about 5.2MW, which is above the rated power of the turbine, so the control system activates the pitch actuators. Between 500 seconds and 600 seconds, the blade pitch angle increases from 0 degrees to about 10 degrees. This helps to reduce the rotor speed to about 12.1rpm and limits the generated power to 5MW.

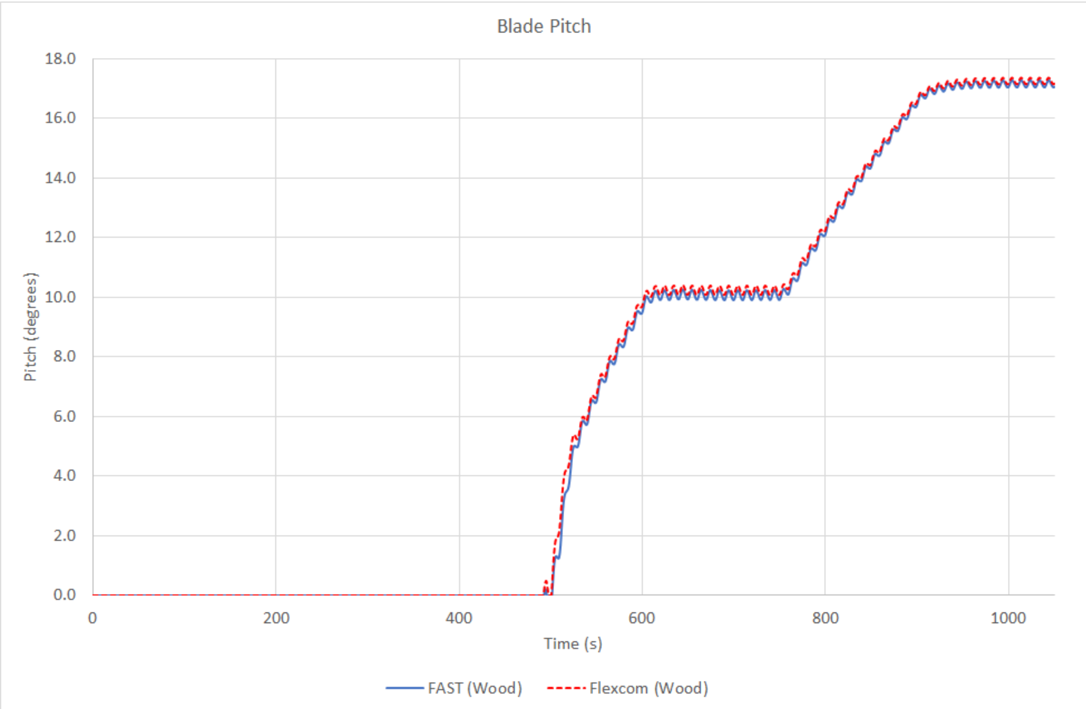
Between 600 seconds and 750 seconds, the wind speed is held constant at 15m/s, with all parameters hovering around mean steady-state values (torque ~43kNm, rotor speed ~12.1rpm, blade pitch ~10 degrees, generator power ~5MW). Again some minor fluctuations are caused by the wave induced motions of the platform. The control system keeps the turbine operating at rated power under these wind conditions (54kph/33mph).

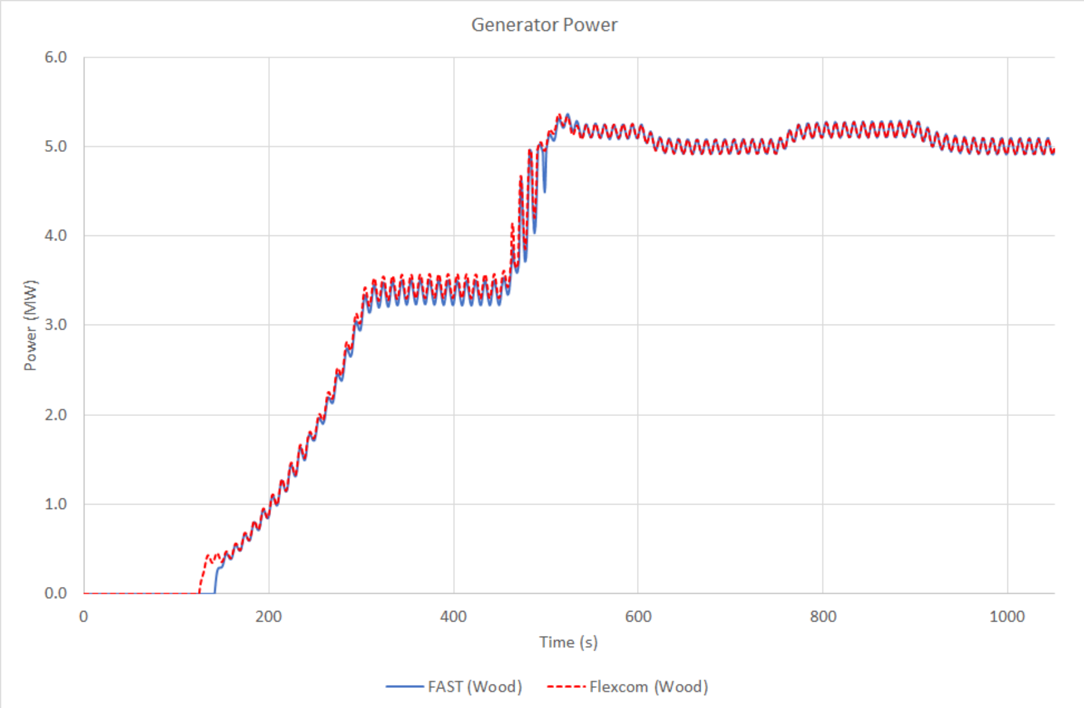
Between 750 seconds and 900 seconds, the wind speed is steadily increased from 15m/s to 20m/s. As the turbine is already operating at rated power, the generator torque remains at 43kNm, while the control system further pitches the blades to about 17 degrees. The rotor speed increases slightly to about 12.5rpm, and the generator power increases to about 5.2MW (slightly above rated), as there is naturally a slight time delay between the control and the response. It should be noted also that the wind time history used in this example is intended for illustrative purposes only - an increase in mean wind speed of 36kph over just 2.5 minutes would not be experienced in reality.

Between 900 seconds and 1050 seconds, the wind speed is held constant at 20m/s, with all parameters hovering around mean steady-state values (torque ~43kNm, rotor speed ~12.1rpm, blade pitch ~17 degrees, generator power ~5MW). Again some minor fluctuations are caused by the wave induced motions of the platform. The control system keeps the turbine operating at rated power under these wind conditions (72kph/44mph).



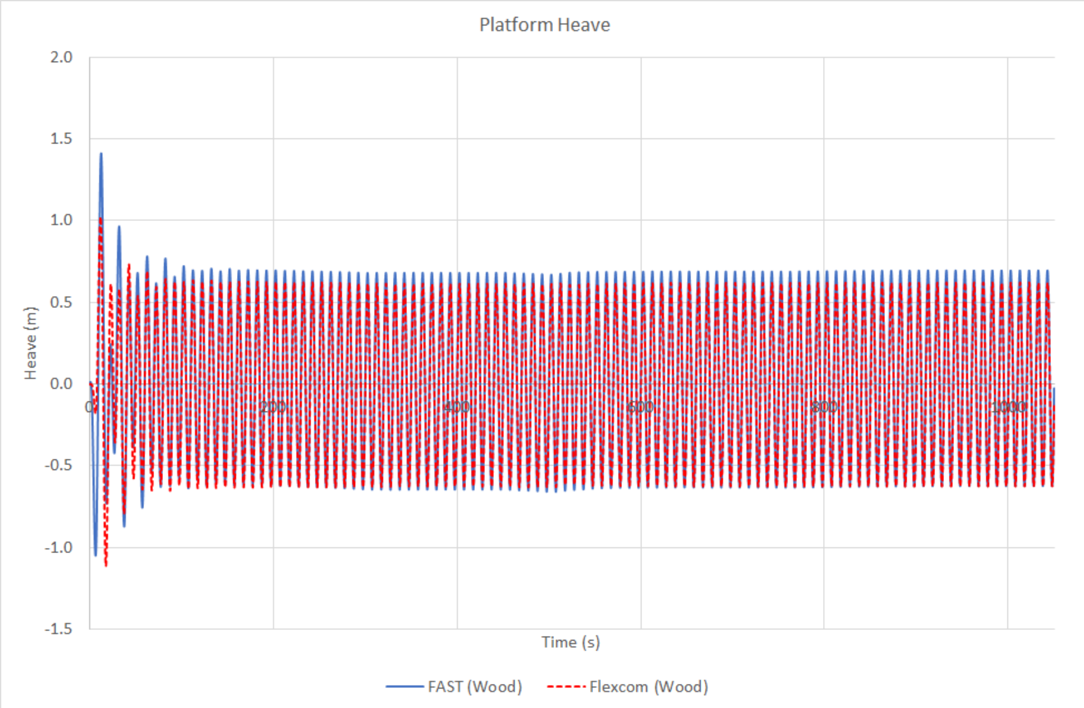

Generator Torque
Torque (kNm)
50.0
45.0
40.0
35.0
30.0
25.0
20.0
15.0
10.0
5.0
0.0
0
200
400
600
800
1000
Time (s)
FAST (Wood)
Flexcom (Wood)

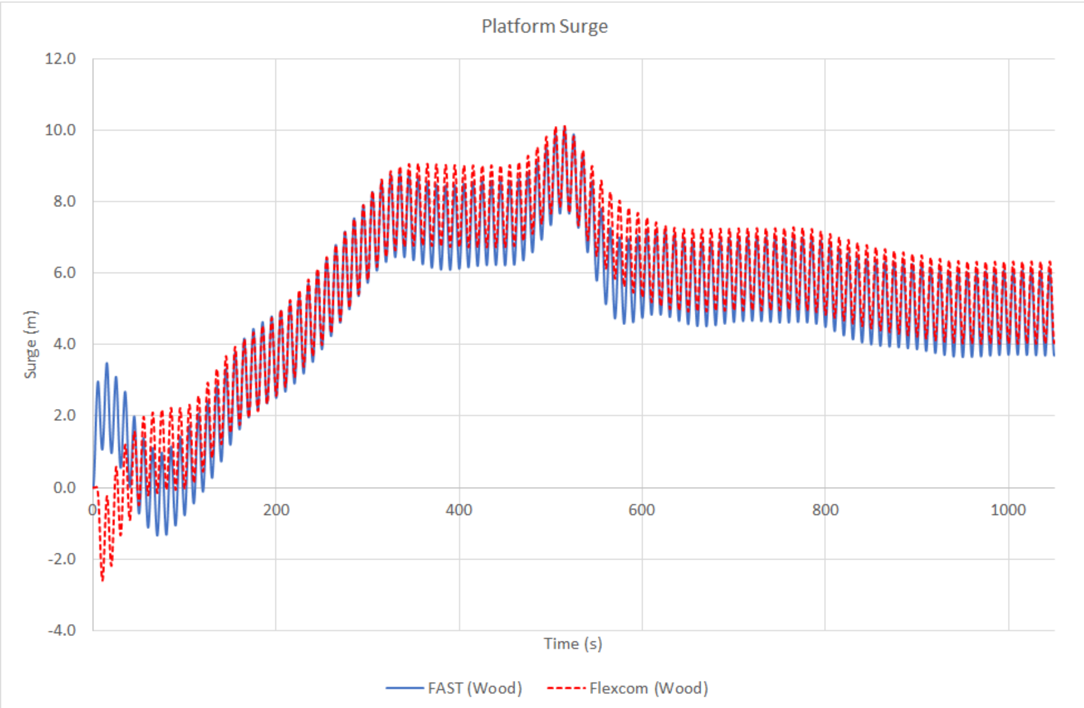

Blade Pitch
Pitch (degrees)
18.0
16.0
14.0
12.0
10.0
8.0
6.0
4.0
2.0
0.0
0
200
400
600
800
1000
Time (s)
FAST (Wood)
Flexcom (Wood)

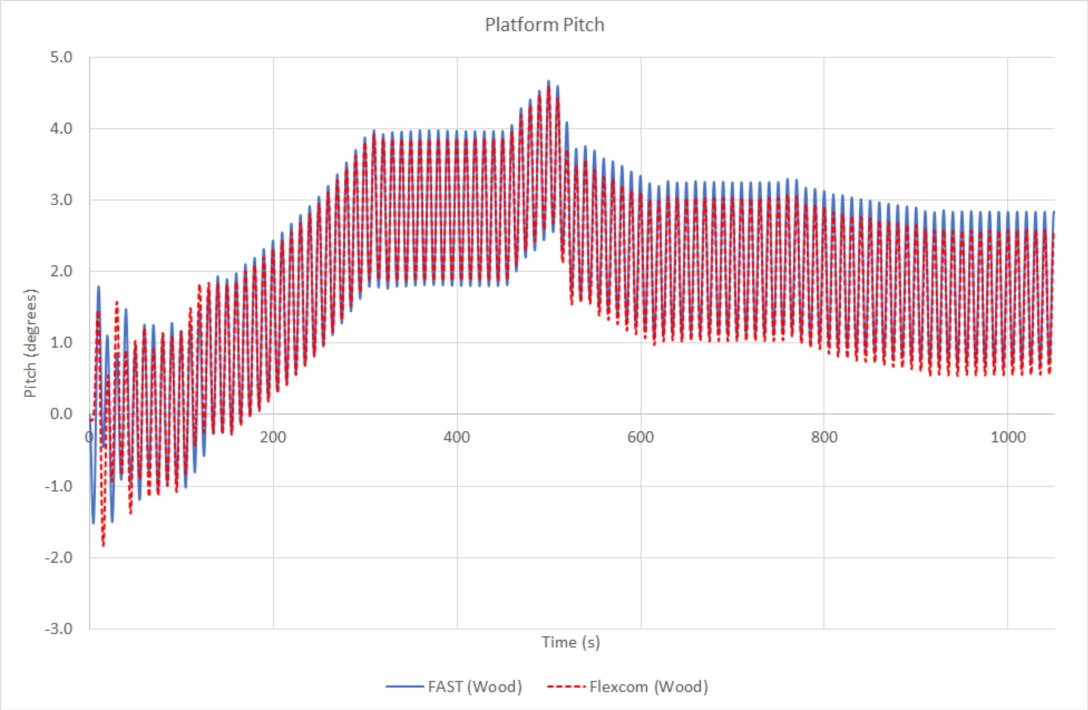


PLATFORM

The following figures show the platform responses in heave, surge and pitch. Close agreement is demonstrated between Flexcom and FAST.

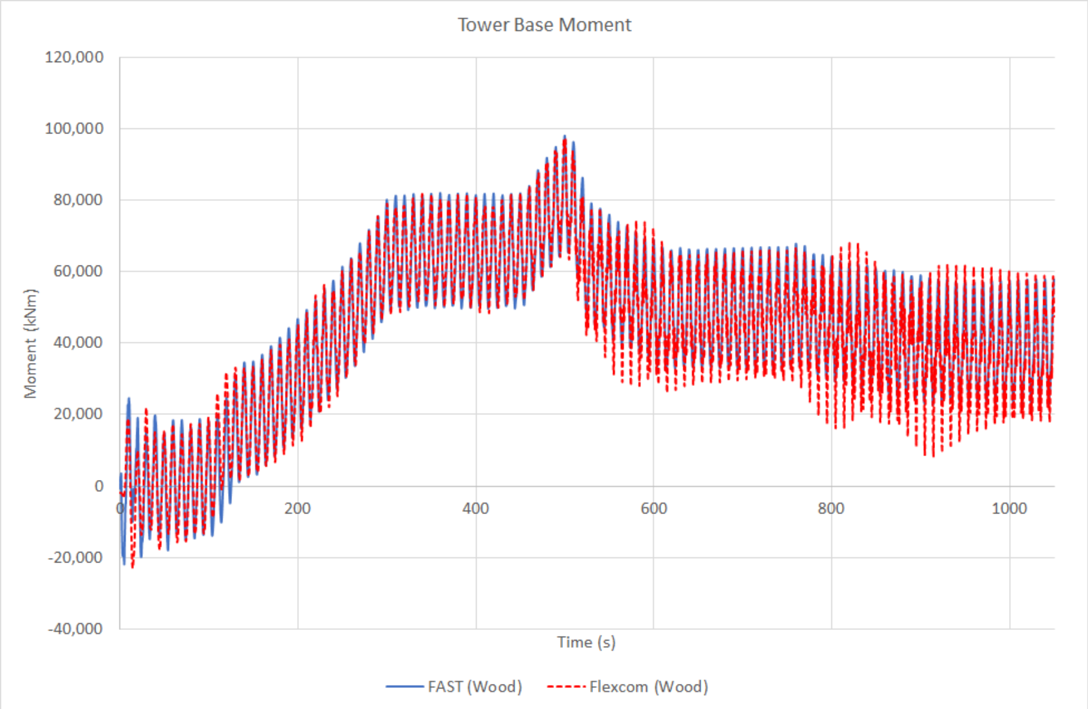

Platform Heave
2.0
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
Heave (m)
0
200
400
600
800
1000
Time (s)
FAST (Wood)
Flexcom (Wood)


Platform Surge
12.0
10.0
8.0
6.0
4.0
2.0
0.0
-2.0
-4.0
Surge (m)
0
200
400
600
800
1000
Time (s)
FAST (Wood)
Flexcom (Wood)



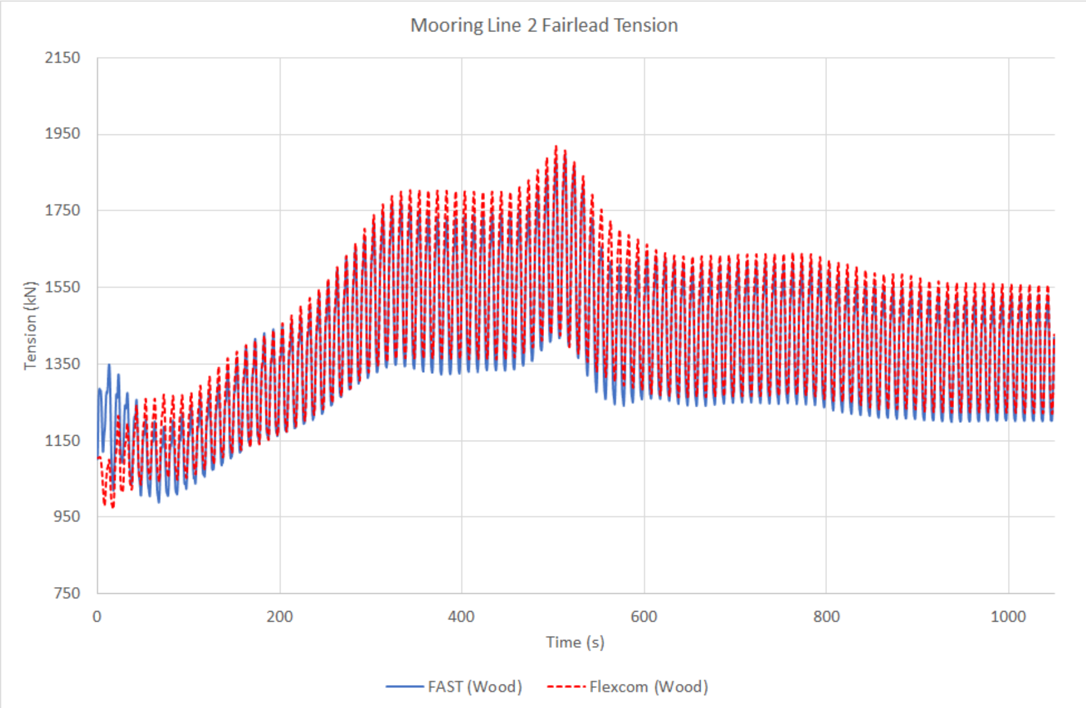
TOWER

Reasonable agreement is demonstrated between Flexcom and FAST for bending moment at the base of the tower.



MOORING LINES

Close agreement is demonstrated between Flexcom and FAST for the fairlead tension in mooring line no.2 (the mooring which is aligned with the wind and wave loading). While the mean value is slightly different, due to a difference in mean surge, the dynamic variations about the means are very similar.



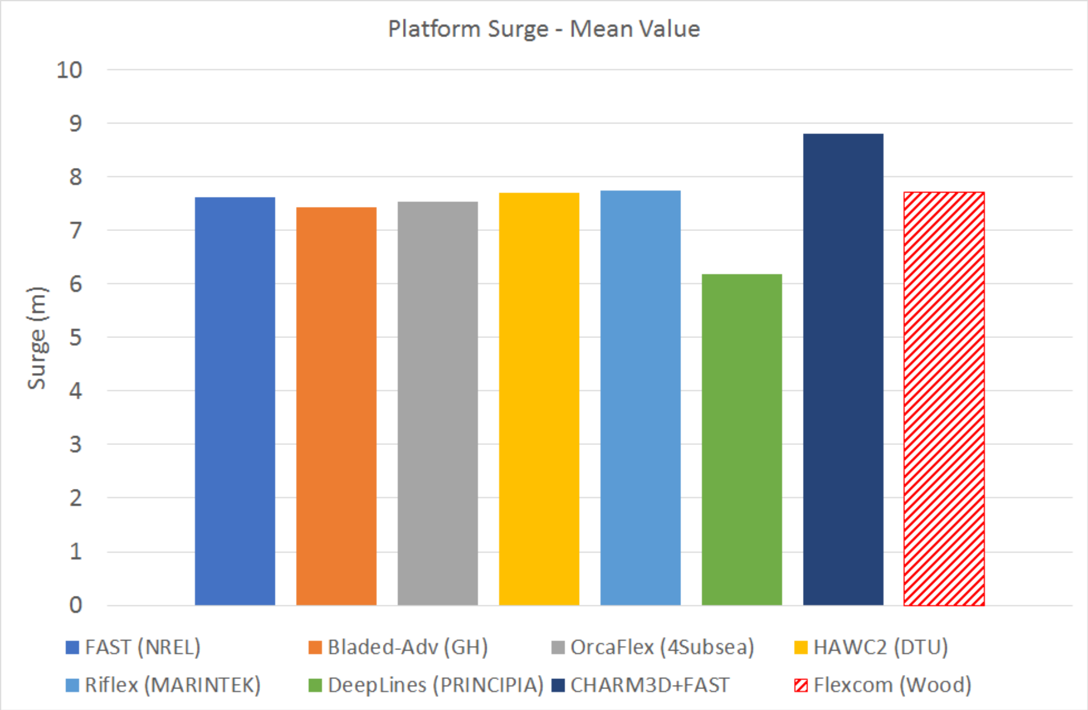
OC4 P2 LC3.2 Stochastic, at rated

This load case is similar to OC4 P2 LC2.2, which does not include wind loading. Generally speaking, the comparisons presented here are very similar also. Results for irregular wave cases are presented in terms of mean and variance of key parameters rather than full time histories. To facilitate comparisons, Robertson et al. (2014) focus on the key outputs of platform surge, platform pitch and bending moment at the base of the tower. Generator power is also considered here, building on the results presented for the earlier load cases. Flexcom shows good agreement with the other software tools. Note that results from Bladed (GH) were not found in the IEA Wind Task 30 results library, so we have used results from Bladed Advanced Hydro Beta (GH) instead in this comparison.

MEAN RESPONSE

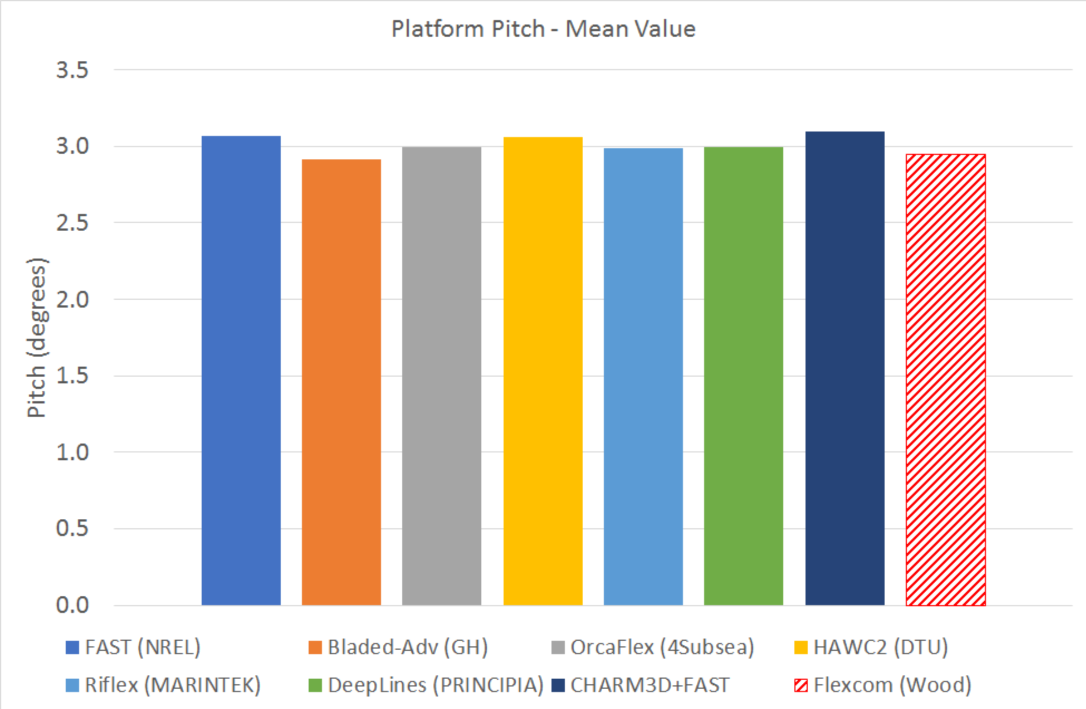
Platform Surge

Good agreement is shown between all software tools, including Flexcom, for the mean surge response. Some variation is expected depending on how drift forces are being modelled (refer to OC4 P2 LC2.1 for further information). The inclusion of wind tends to improve the correlation in surge results, because the wind thrust force is much larger than the hydrodynamic drift force, and hence the mean surge offset is significantly greater when wind is included.



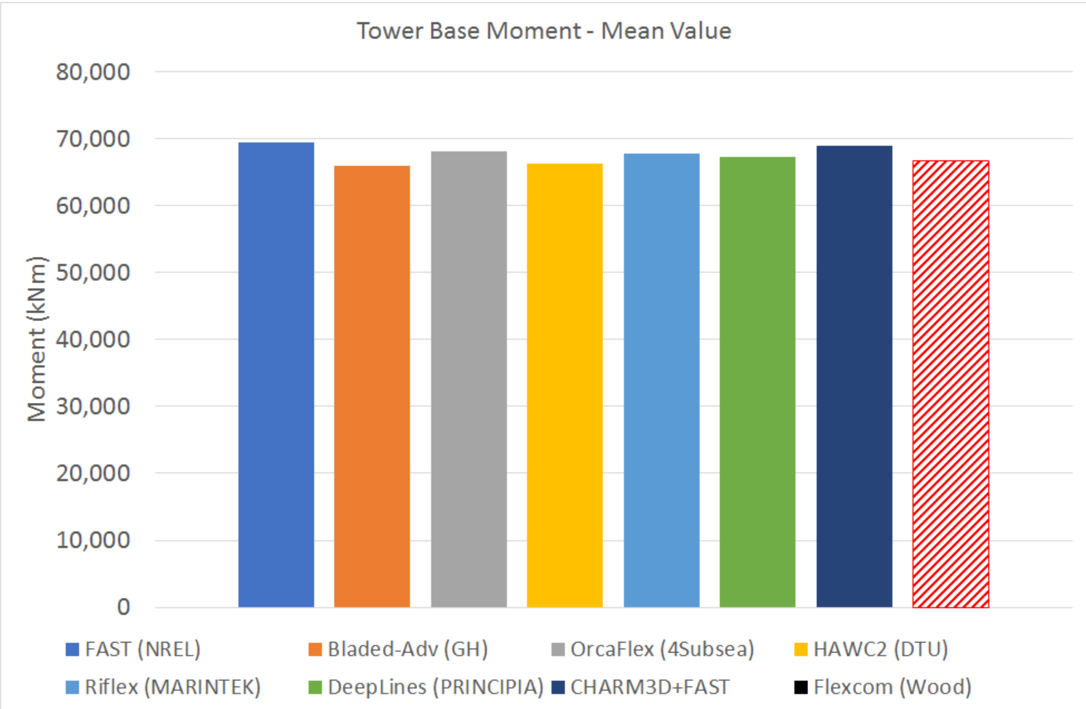
Platform Pitch

Close agreement is shown between all software tools, including Flexcom, for the mean pitch response. The inclusion of wind tends to improve the correlation in pitch results, because the wind thrust force is much larger than the hydrodynamic drift force, and hence the mean pitch offset is significantly greater when wind is included.



Tower Base Moment

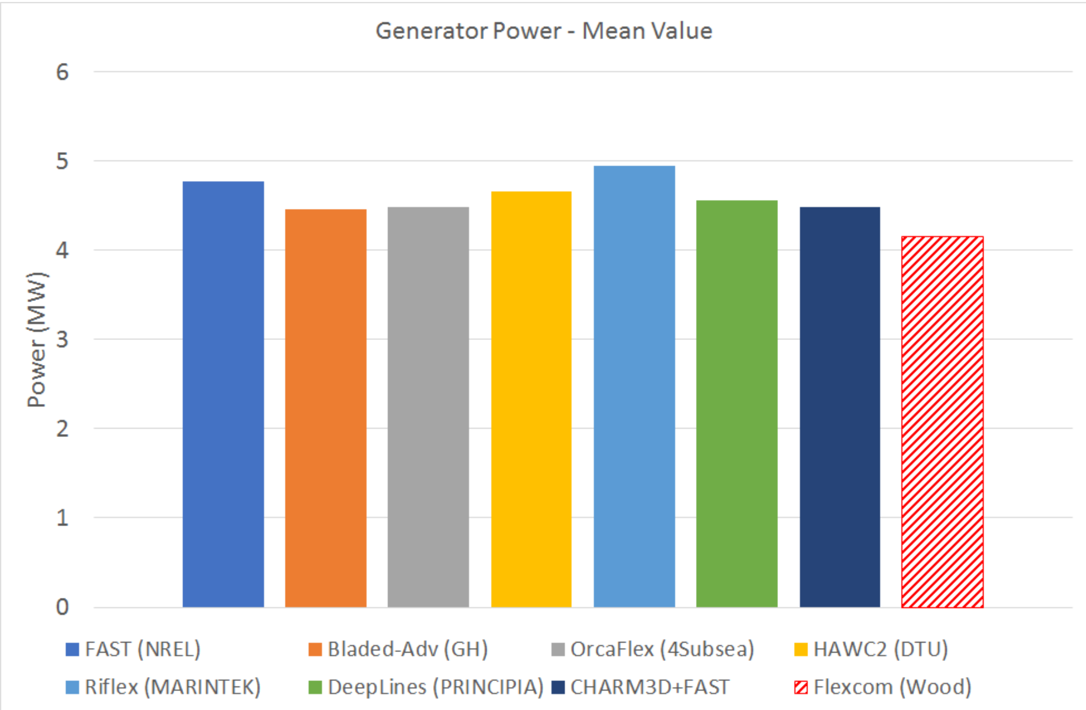
Close agreement is shown between all software tools, including Flexcom, for the mean tower base moment.



Generator Power

Good agreement is shown between all software tools for the mean generator power. The generator power predicted by Flexcom is slightly lower than most of the other software tools - Flexcom's mean power is 4.16MW, 10.1% below the average. This may sound quite significant, but it is worth noting that there is a 10.6% variation across the other software tools also.

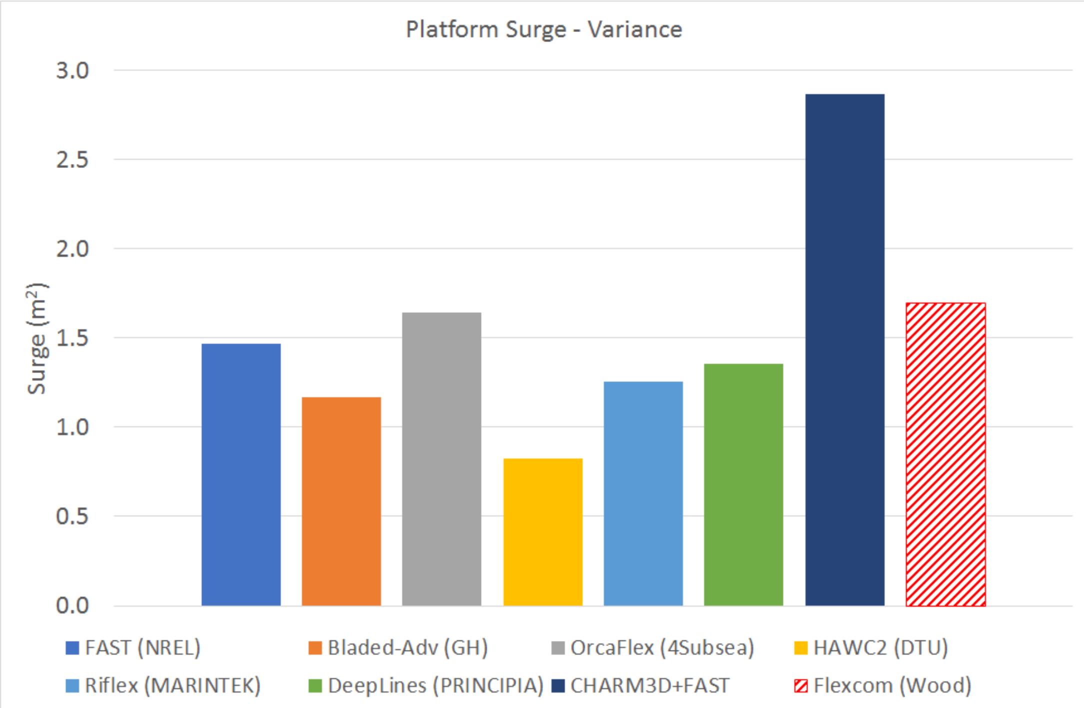
It is worth noting that the Flexcom results were produced using Flexcom 8.12.1 which is coupled with AeroDyn V15.02.04 (14-Apr-2016). The OC4 Phase II results were published in 2014 at which point the latest version of AeroDyn was Version 14. Improvements to the aerodynamic modelling software primarily account for the differences in generator power predicted by Flexcom and FAST (NREL). Slightly different platform motions predicted by each software tool also influences generator power. Variations in the aerodynamic theories underpinning each software product may also be a contributing factor. You are referred to OC4 P2 LC3.1 if you are interested in a more detailed discussion.



VARIANCE OF RESPONSE

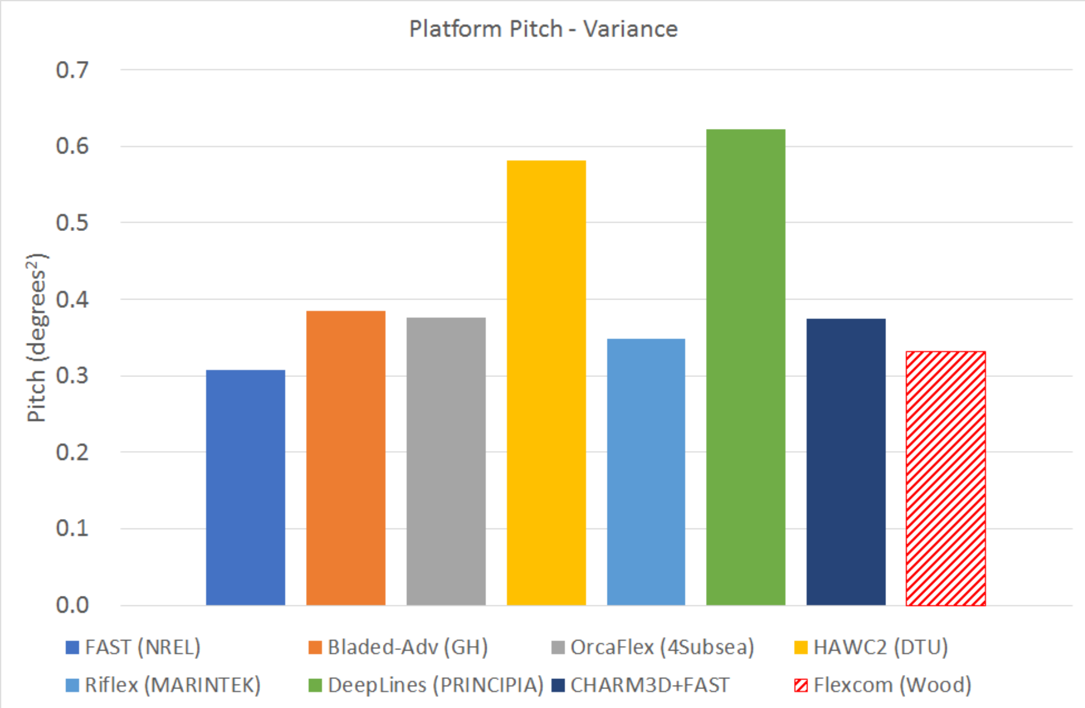
Platform Surge

The variance in surge shows some variation across the different software products, again possibly caused by different approaches to modelling drift forces. Results from Charm3D+FAST (ABS) show a surge variance which is larger than the other software tools.



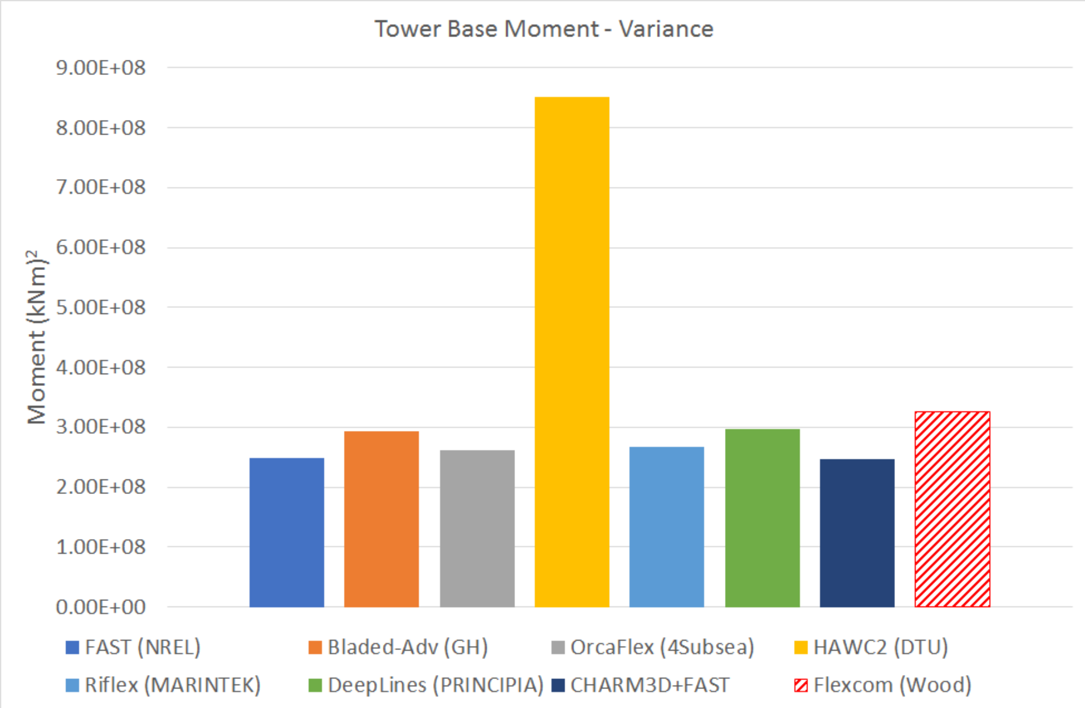
Platform Pitch

The variance in pitch shows some variation across the different software products. However the response predicted by Flexcom shows good general agreement with the average value.



Tower Base Moment

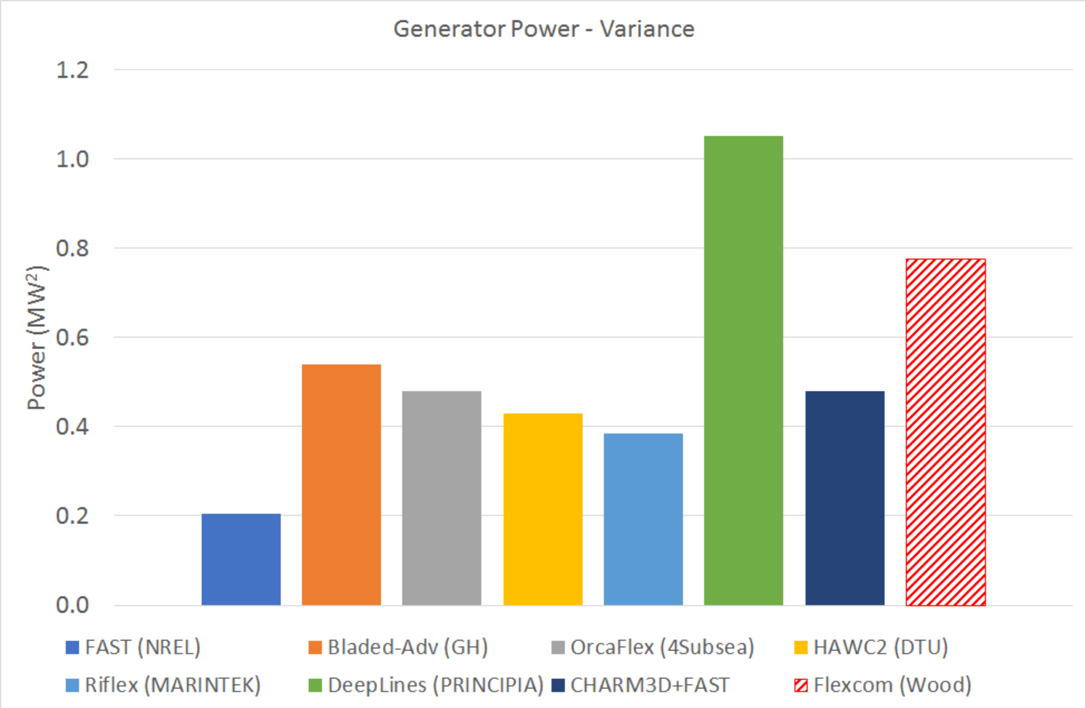
The variance in tower base moment shows reasonable agreement across the different software products, with the exception of HAWC2 (DTU) which appears to be inconsistent with the others. Robertson et al. (2014) suggest that this probably results from an incorrectly prescribed axis definition for the output.



Generator Power

The variance in generator power shows considerable variation across the different software products.

It is worth noting that the Flexcom results were produced using Flexcom 8.12.1 which is coupled with AeroDyn V15.02.04 (14-Apr-2016). The OC4 Phase II results were published in 2014 at which point the latest version of AeroDyn was Version 14. Improvements to the aerodynamic modelling software primarily account for the differences in generator power predicted by Flexcom and FAST (NREL). Slightly different platform motions predicted by each software tool also influences generator power. Variations in the aerodynamic theories underpinning each software product may also be a contributing factor. You are referred to OC4 P2 LC3.1 if you are interested in a more detailed discussion.



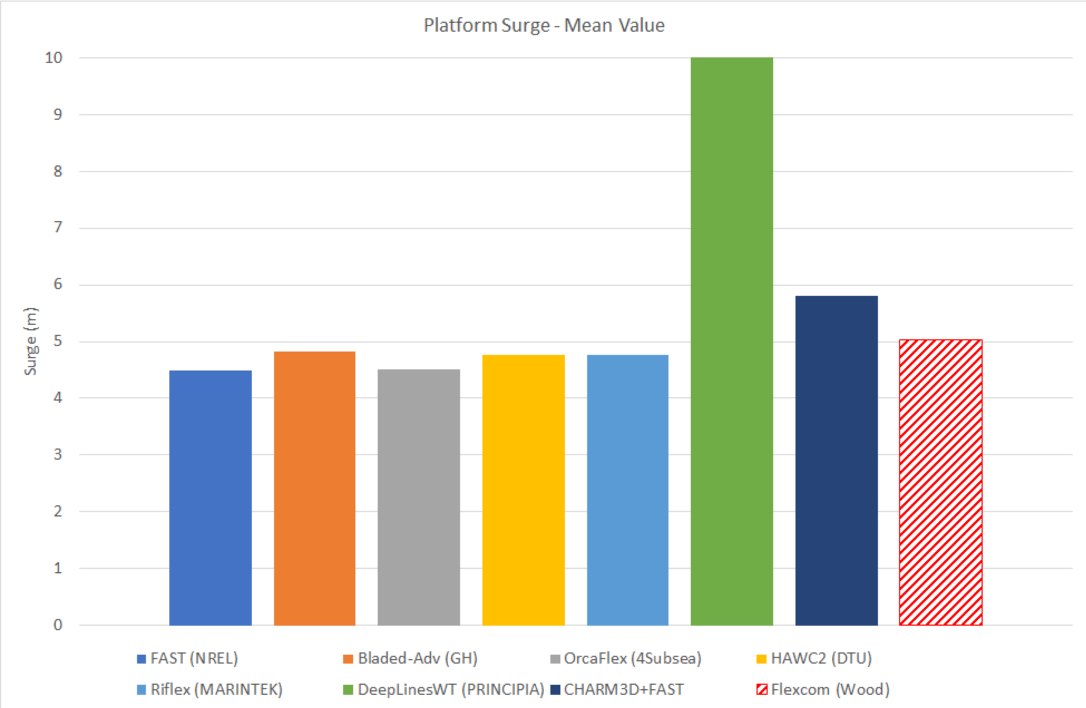
OC4 P2 LC3.3 Stochastic, above rated

This load case is very similar to OC4 P2 LC3.2, but the mean wind speed is higher in this case (Vhub=18.0m/s rather than 11.4m/s). Although this load case is not discussed by Robertson et al. (2014), comparisons are presented here in a similar manner to the preceding load case. Specifically, the mean and variance of platform surge, platform pitch, bending moment at the base of the tower, and generator power are considered. Generally speaking, Flexcom shows good agreement with the other software tools. Note that results from Bladed (GH) were not found in the IEA Wind Task 30 results library, so we have used results from Bladed Advanced Hydro Beta (GH) instead in this comparison.

MEAN RESPONSE

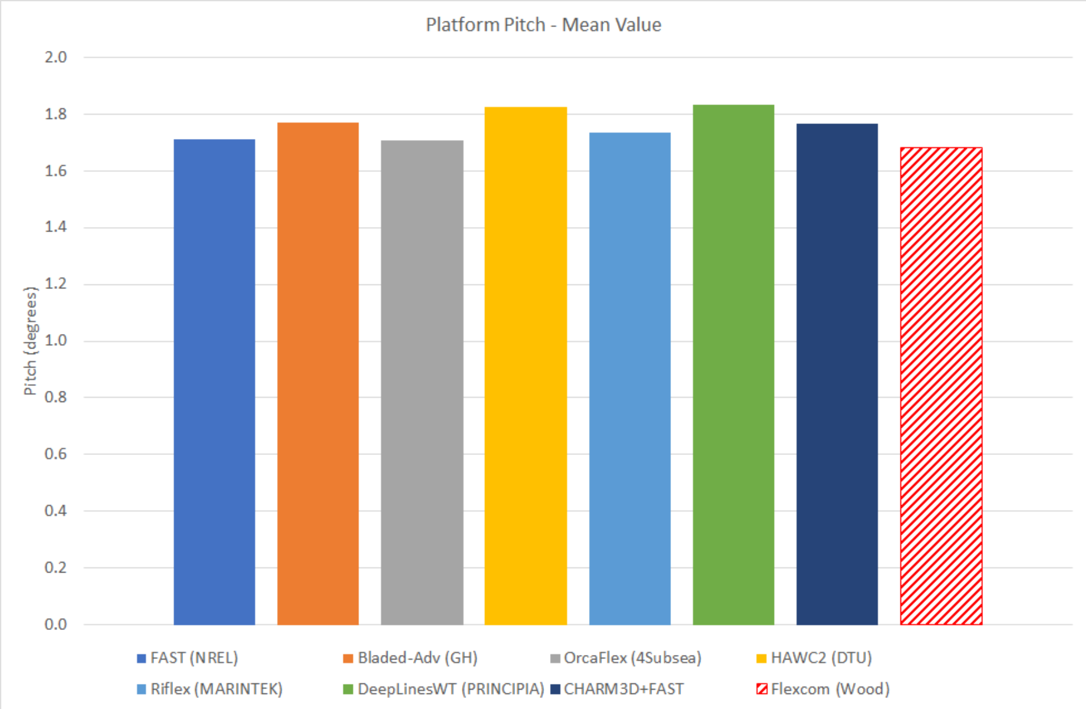
Platform Surge

The mean surge shows reasonable agreement across the different software products, with the exception of DeepLinesWT (Principia) which appears to be inconsistent with the others. Although the full extent is not shown below, this is about 5 times the average of the others. Without comment from Robertson et al. (2014), the reasons for this are not clear, but it could be something as simple as a mistake in the presentation of results. Regardless, some variation is expected depending on how drift forces are being modelled (refer to OC4 P2 LC2.1 for further information). The inclusion of wind tends to improve the correlation in surge results, because the wind thrust force is much larger than the hydrodynamic drift force, and hence the mean surge offset is significantly greater when wind is included.



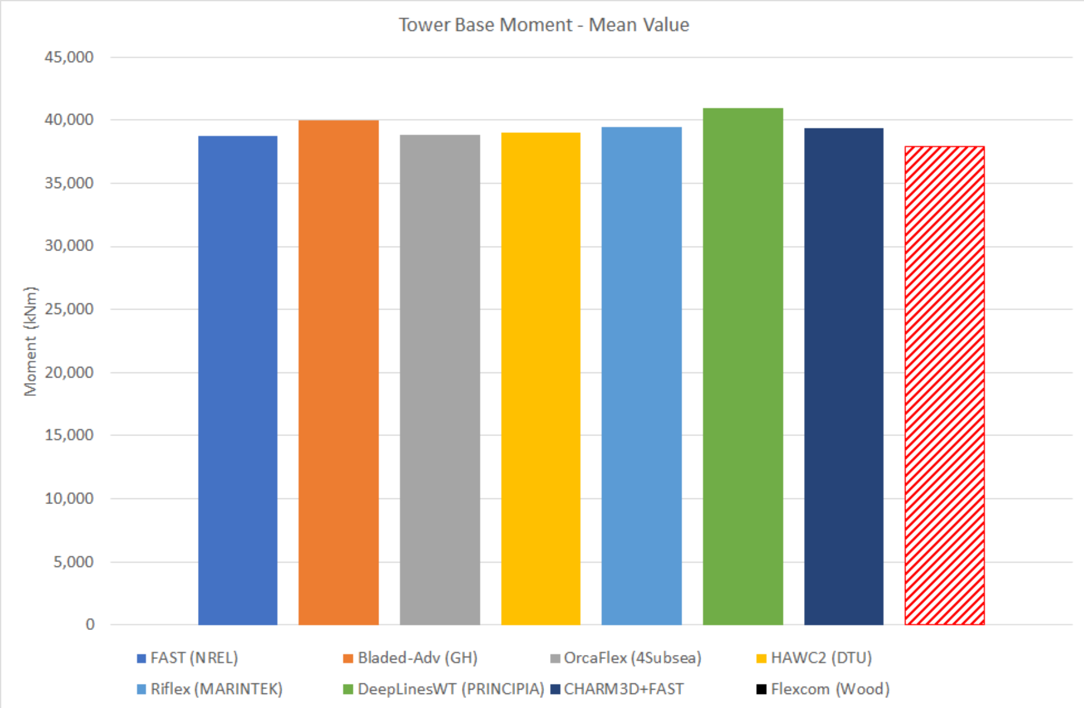
Platform Pitch

Close agreement is shown between all software tools, including Flexcom, for the mean pitch response. The inclusion of wind tends to improve the correlation in pitch results, because the wind thrust force is much larger than the hydrodynamic drift force, and hence the mean pitch offset is significantly greater when wind is included.



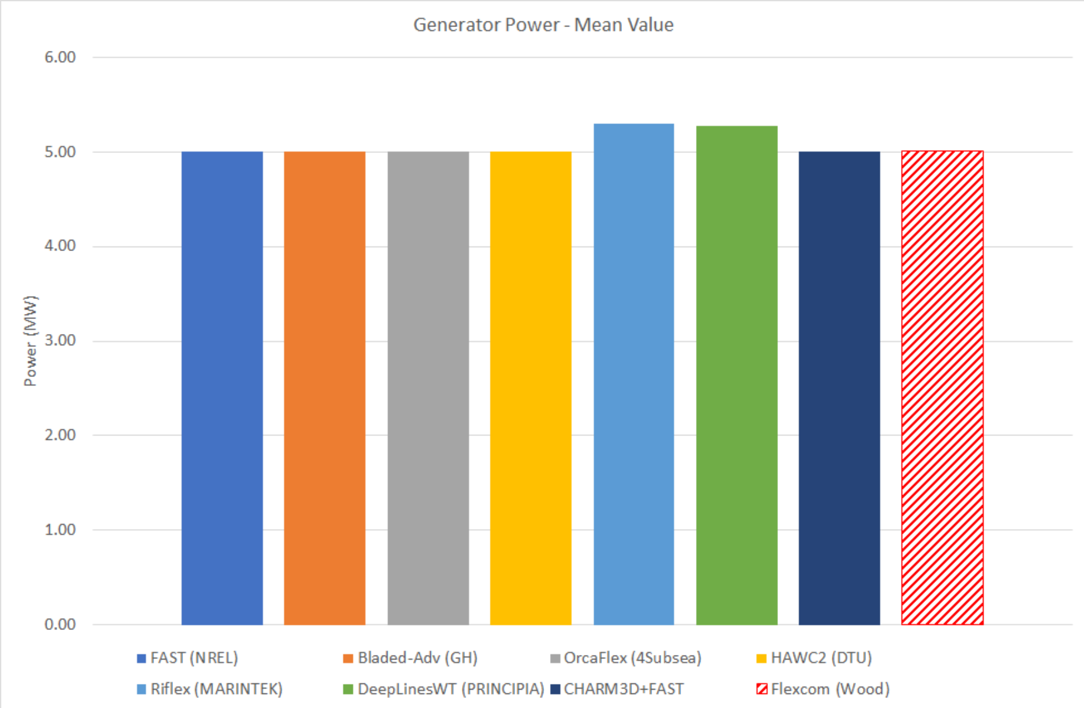
Tower Base Moment

Close agreement is shown between all software tools, including Flexcom, for the mean tower base moment.



Generator Power

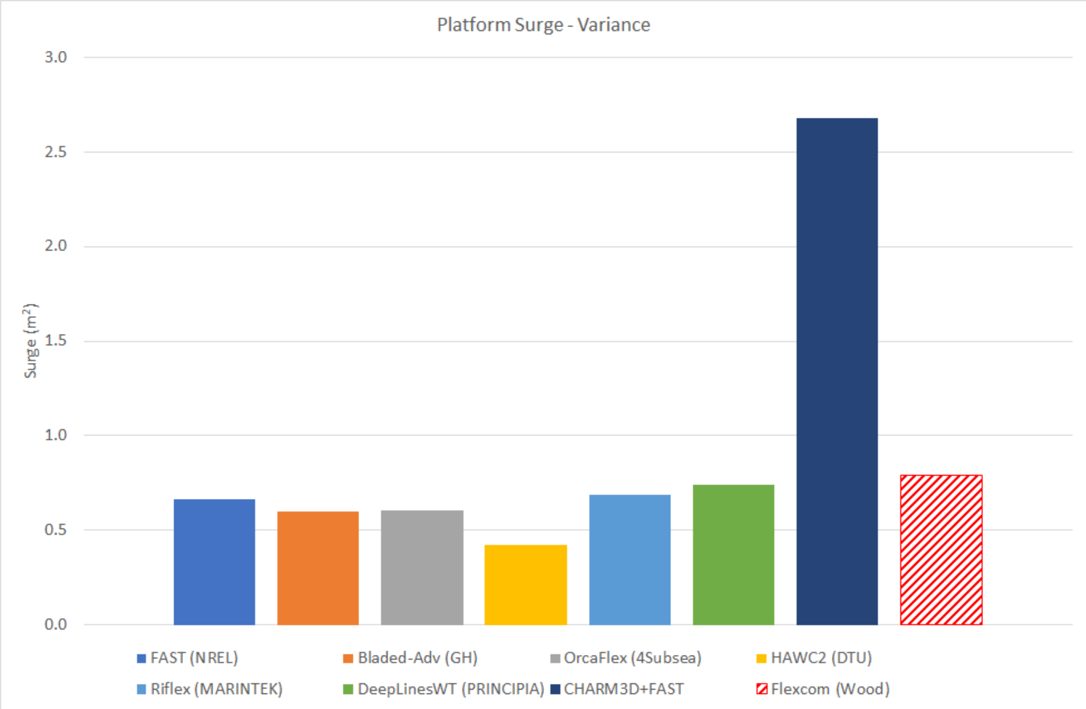
Very close agreement is shown between all software tools for the mean generator power. Unlike OC4 P2 LC3.2, where the mean wind speed is lower (Vhub=18.0m/s rather than 11.4m/s), the wind speed is this case is sufficiently high to keep the turbine operating at its rated power. Hence many of the software tools, including Flexcom, predict exactly 5MW.



VARIANCE OF RESPONSE

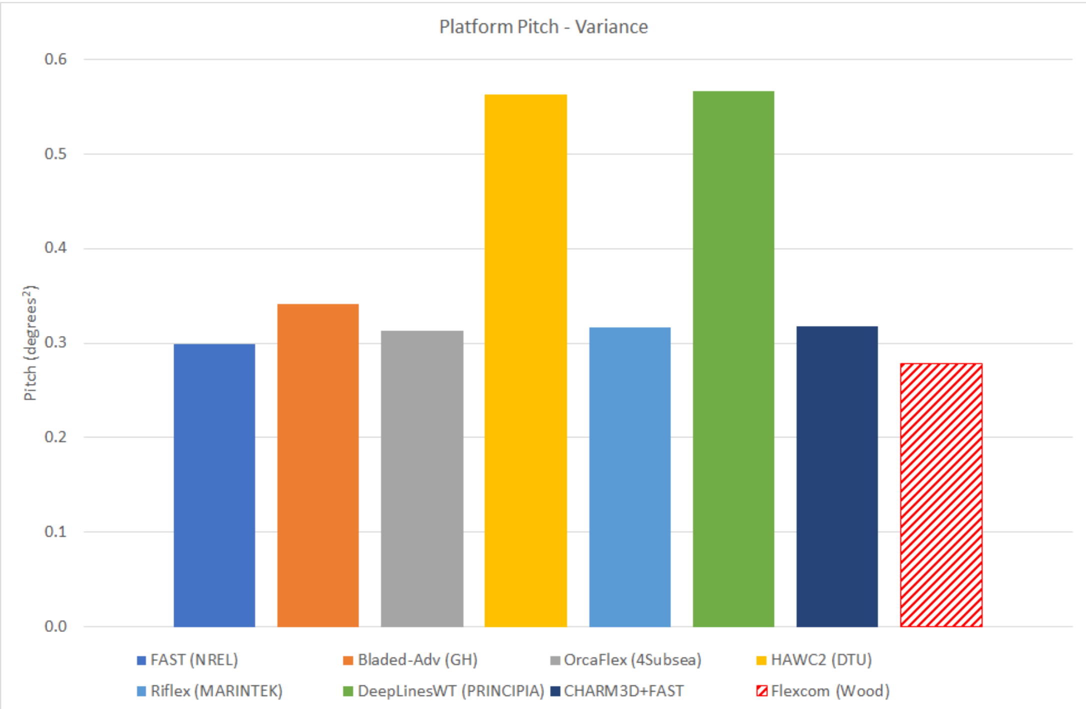
Platform Surge

The variance in surge shows some variation across the different software products, again possibly caused by different approaches to modelling drift forces. Results from Charm3D+FAST (ABS) show a surge variance which is larger than the other software tools.



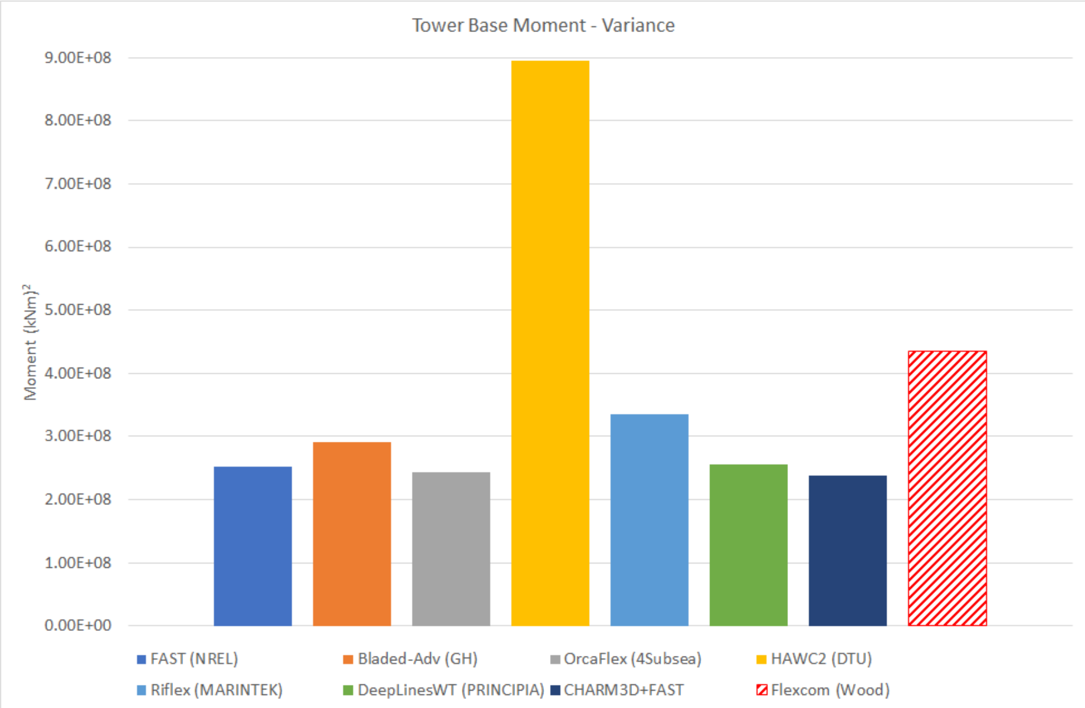
Platform Pitch

The variance in pitch shows some variation across the different software products. However the response predicted by Flexcom shows good general agreement with the average value.



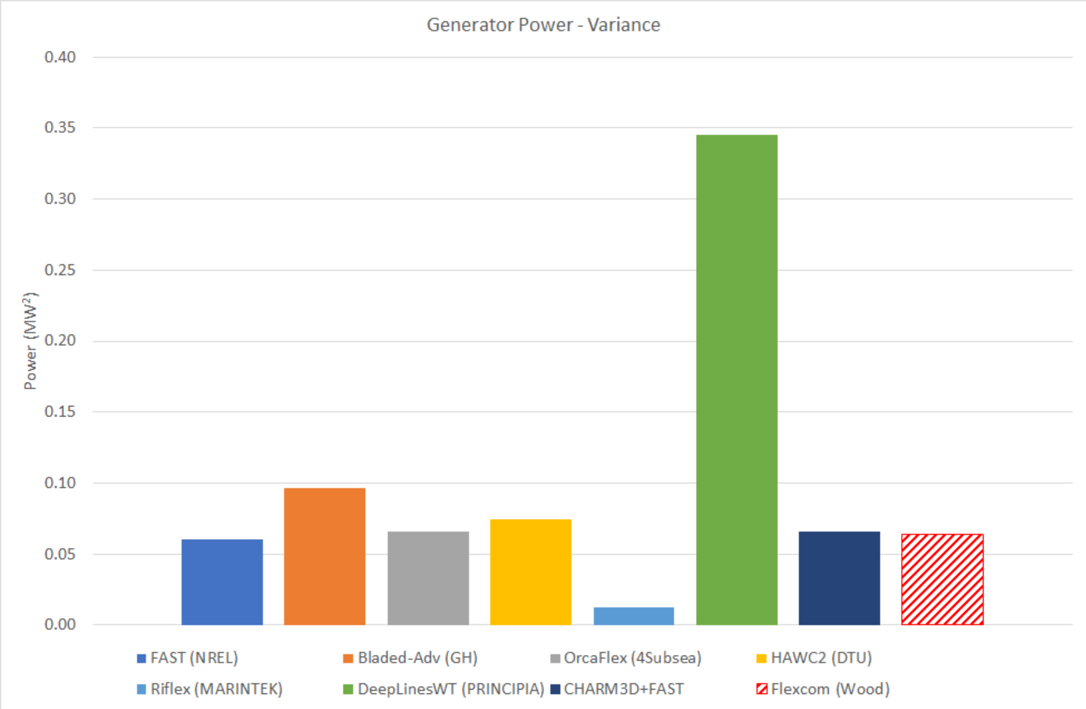
Tower Base Moment

The variance in tower base moment shows reasonable agreement across the different software products, with the exception of HAWC2 (DTU) which appears to be inconsistent with the others. Based on OC4 P2 LC3.2, it seems likely that the discrepancy probably results from an incorrectly prescribed axis definition for the output.



Generator Power

The variance in generator power is extremely small. Unlike OC4 P2 LC3.2, where the mean wind speed is lower (Vhub=18.0m/s rather than 11.4m/s), the wind speed is this case is sufficiently high to keep the turbine operating at its rated power. The average variance in generator power here is about 5 times lower than the corresponding value for OC4 P2 LC3.2.

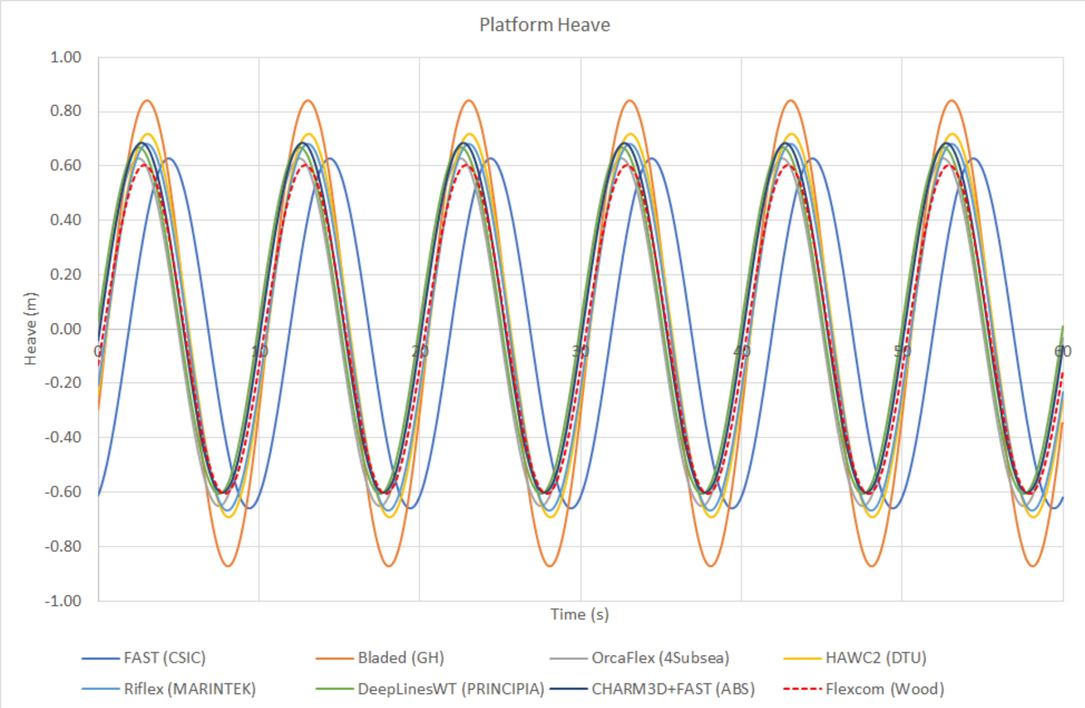


OC4 P2 LC3.4 Wind, wave and current

This load case combines elements from OC4 P2 LC3.1 (which includes regular wave and steady wind loading) and OC4 P2 LC2.4 (which regular wave and current loading). Similar trends to that observed for these simpler load cases are also evident here. Note that results from FAST (NREL) were not found in the IEA Wind Task 30 results library, so we have used results from FAST (CSIC) instead in this comparison.

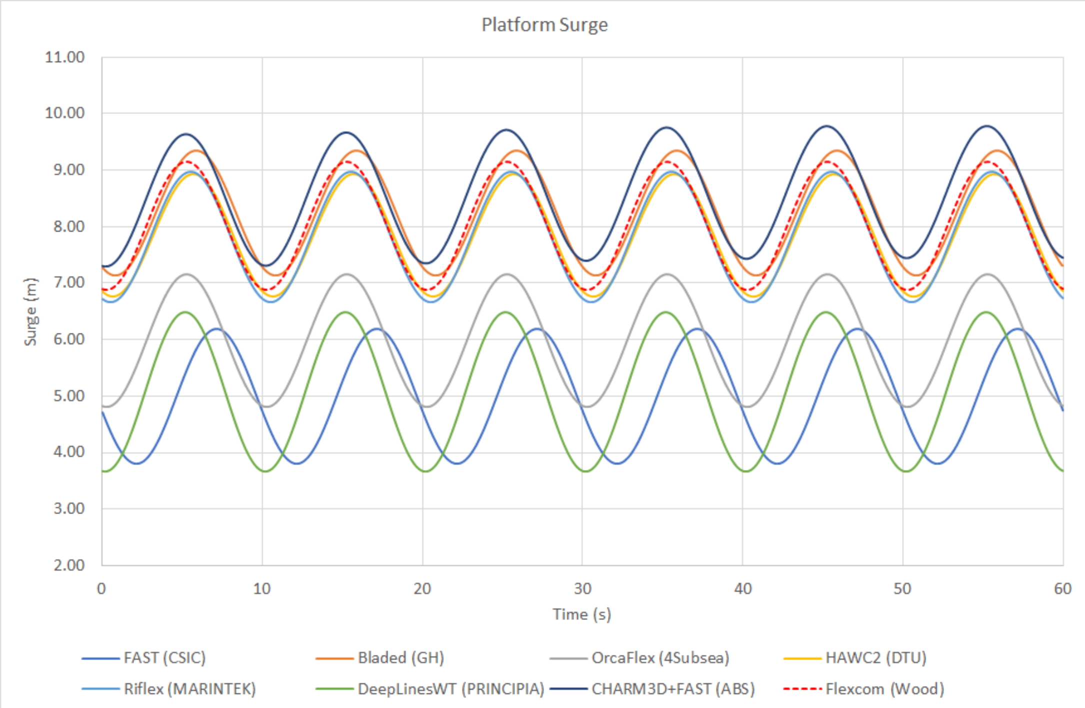
PLATFORM HEAVE

Flexcom's heave response shows close agreement with the other software tools, in terms of response amplitude, period and phase. Results from Bladed (GH) show a heave response which is slightly larger than the other software tools.



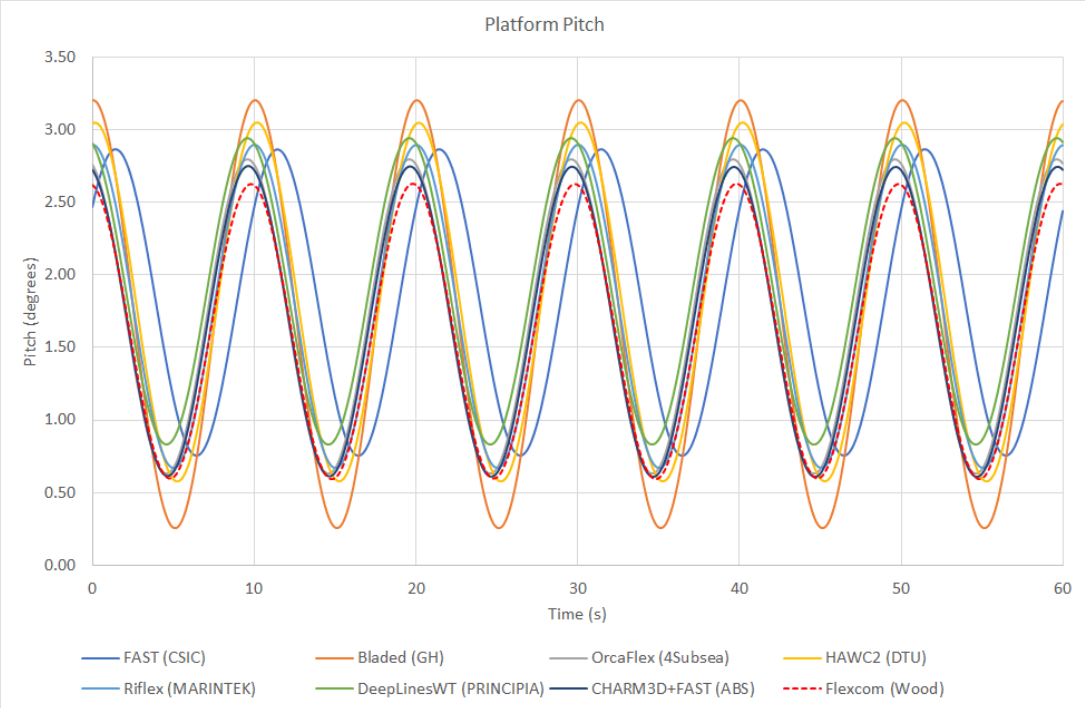
PLATFORM SURGE

The surge response shows considerable variation across the different software products. Some of this may be attributed to the modelling of drift forces (refer to OC4 P2 LC2.1 for further information). Note however that results from FAST (Goldwind), Orcaflex (4Subsea) and DeepLinesWT (Principia) show a mean surge position which is lower than the other software tools. So there is a clear discrepancy between two groups of software in how mean surge is computed when both wave and current loads are included, as broadly speaking, surge motions showed reasonable agreement for wind and waves (OC4 P2 LC3.1) and current only (OC4 P2 LC2.3). This issue is not discussed in Robertson et al. (2014) and it is unclear as to what is causing it.



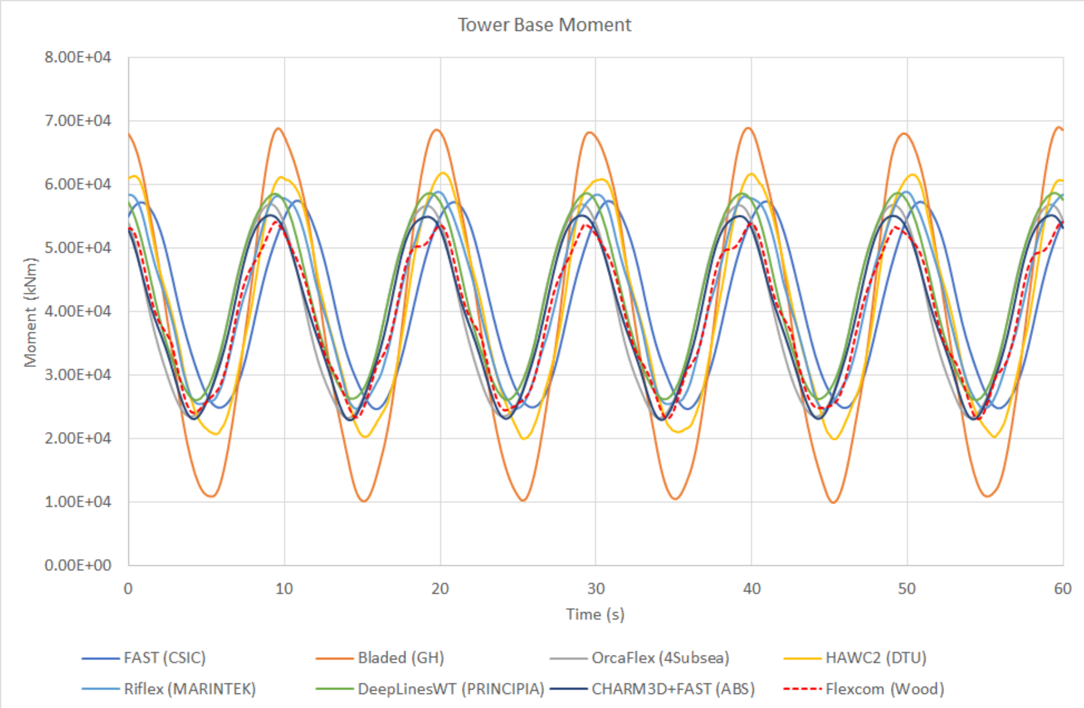
PLATFORM PITCH

Flexcom's pitch response shows reasonable agreement with the other software tools, although the mean pitch is slightly lower than many of the other tools.



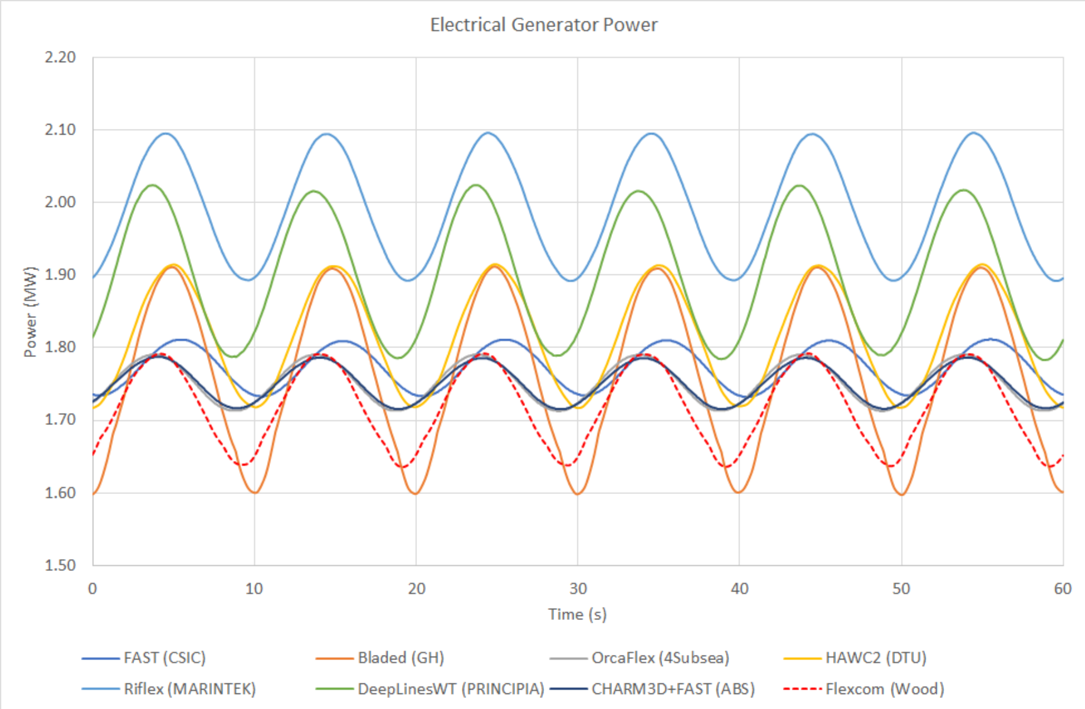
TOWER BASE MOMENT

Results from Flexcom show reasonable agreement with the other software tools for bending moment at the base of the tower. The moments predicted by Flexcom are slightly lower than the other tools - it is believed this is related to the generator power discussed in the next paragraph. Results from Bladed (GH) show moment variations which are larger than the other software tools.

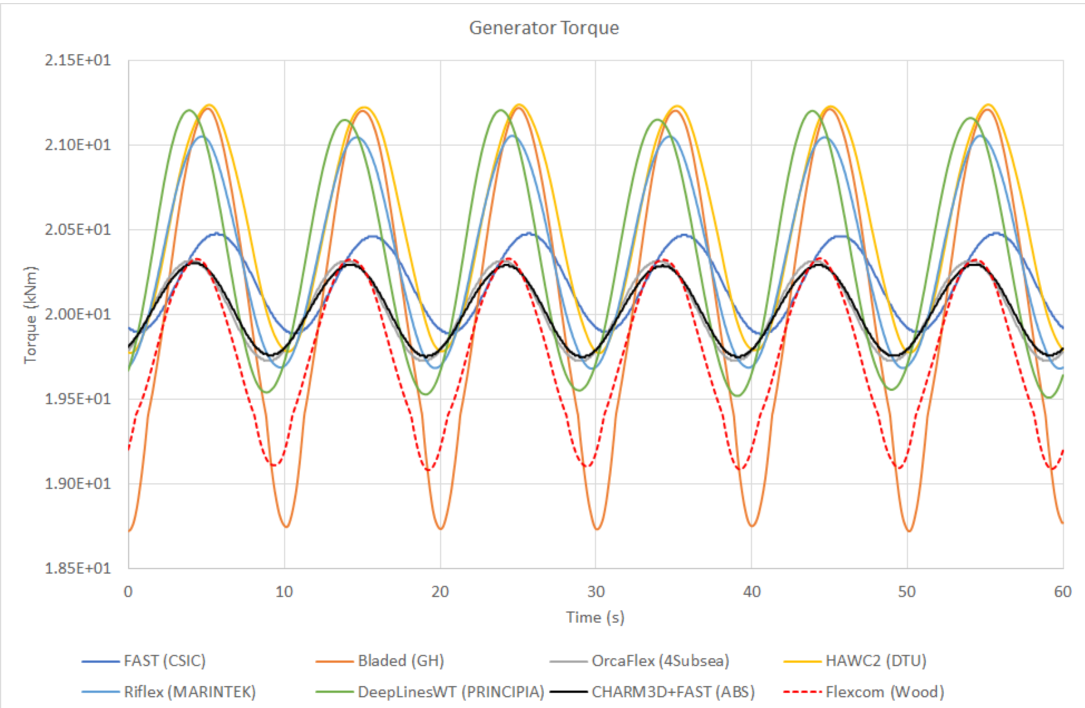


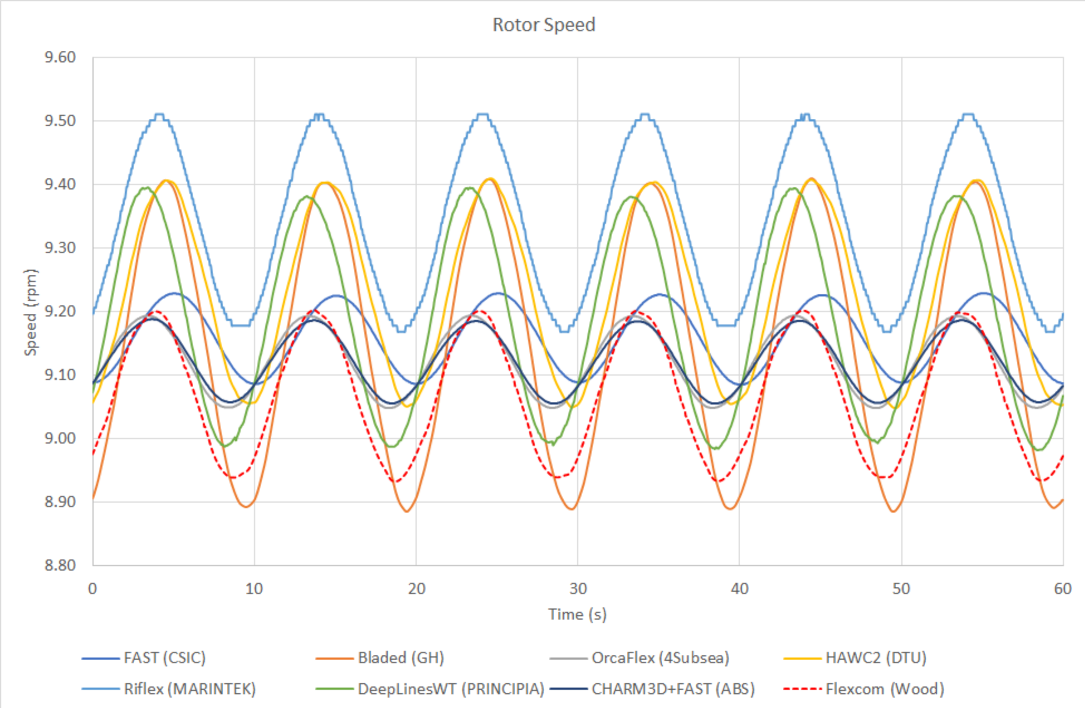
GENERATOR POWER, GENERATOR TORQUE AND ROTOR SPEED

The generator power shows considerable variation across the different software products, with the mean power of the OC4 software tools considered ranging from 1.75MW (OrcaFlex/4Subsea, CHARM3D+FAST/ABS) to 1.99MW (Riflex/MARINTEK). The generator power predicted by Flexcom is slightly lower than most of the other software tools - Flexcom's mean power is 1.72MW, 5.6% below the average.



The variations in generator power are also evident in the primary variables of generator torque and rotor speed.





It is worth noting that the Flexcom results were produced using Flexcom 8.12.1 which is coupled with AeroDyn V15.02.04 (14-Apr-2016). The OC4 Phase II results were published in 2014 at which point the latest version of AeroDyn was Version 14. Improvements to the aerodynamic modelling software primarily account for the differences in generator power predicted by Flexcom and CSIC/FAST. Slightly different platform motions predicted by each software tool also influences generator power. Variations in the aerodynamic theories underpinning each software product may also be a contributing factor. You are referred to OC4 P2 LC3.1 if you are interested in a more detailed discussion.

OC4 P2 LC3.5 50-year extreme wind/wave

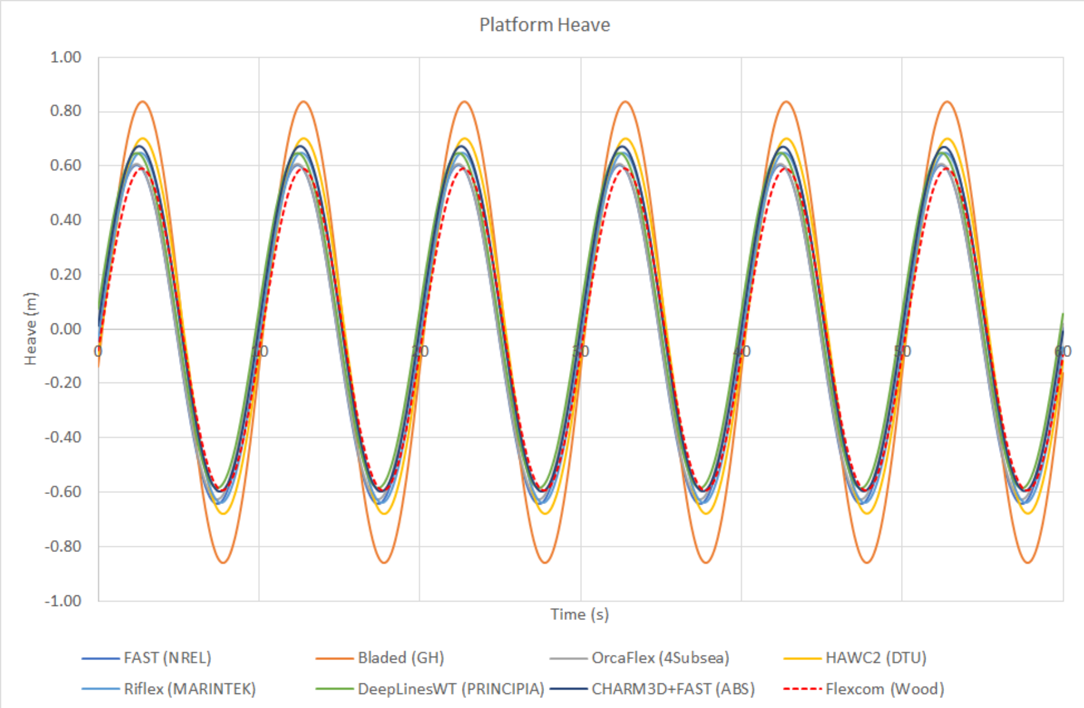
This load case is similar to OC4 P2 LC3.3, but the wind and wave loading is much more severe in this case (Vhub=47.5.0m/s rather than 18.0m/s, and Hs=15m rather than 6m). The simulation actually failed to complete in Flexcom due to an error in the InflowWind module. The turbulent wind field input data files were originally created by DTU (Technical University of Denmark) and kindly passed on to us by NREL. The vertical extents of the wind field definition are 17.46m and 162.54m respectively above still water. Given the severity of the applied loading, the induced platform motions causes one of the blade nodes to pass below the lower boundary of the wind field definition at a certain point during the simulation. InflowWind is unable to compute the wind velocity at this point so the software issues an error message and terminates: "*FF wind array boundaries violated. Grid too small in Z direction (height (Z=17.401 m) is below the grid and no tower points are defined*". It would be possible to manually regenerate the turbulent wind field with larger boundaries using a third-party wind turbulence simulator, but we wanted to use the exact wind field definition used by the OC4 participants to ensure consistency of the Flexcom model with the others. We are not sure if any of the other participants experienced a similar problem or not, as the load case is not discussed by Robertson et al. (2014).

OC4 P2 LC3.6 Wind/wave misalignment

This load case is similar to OC4 P2 LC3.1 (in which the wave direction is zero degrees rather than 30 degrees in this load case). Generally speaking, the comparisons presented here are naturally quite similar also.

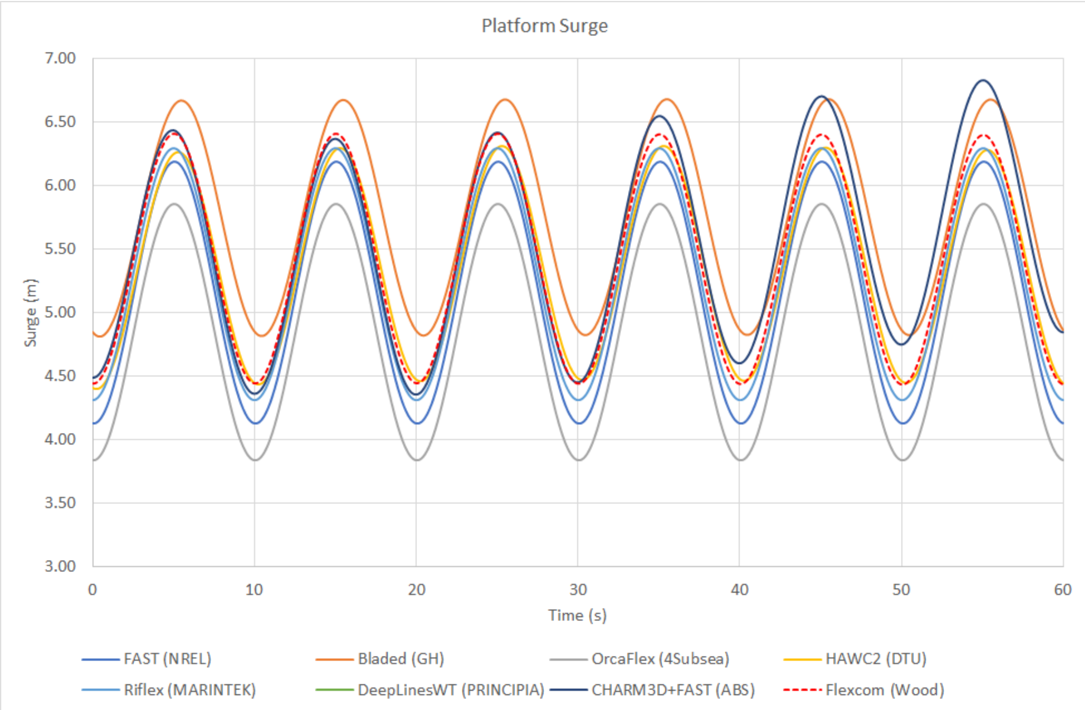
PLATFORM HEAVE

Flexcom's heave response shows close agreement with the other software tools, in terms of response amplitude, period and phase. Results from Bladed (GH) show a heave response which is slightly larger than the other software tools.



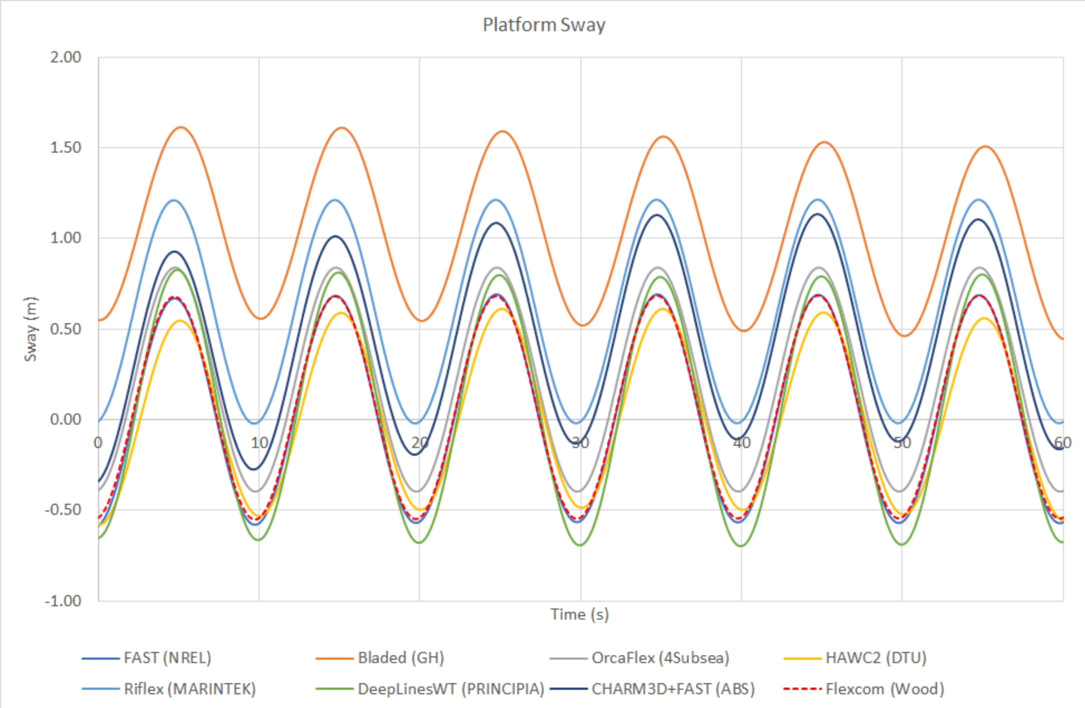
PLATFORM SURGE

Although the surge response shows some variation across the different software products (depending on how drift forces are being modelled - refer to OC4 P2 LC2.1 for further information), the inclusion of wind tends to improve the correlation in surge results. This is because the wind thrust force is much larger than the hydrodynamic drift force, and hence the mean surge offset is significantly greater when wind is included.



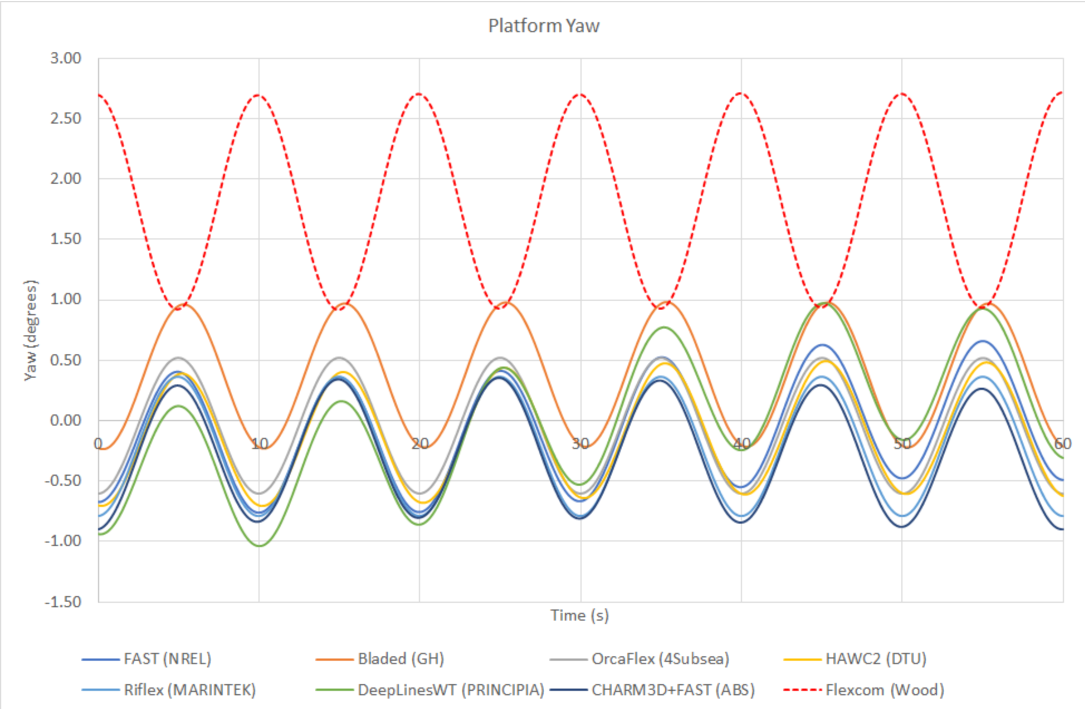
PLATFORM SWAY

The sway response shows some variation across the different software products (depending on how drift forces are being modelled - refer to OC4 P2 LC2.1 for further information).



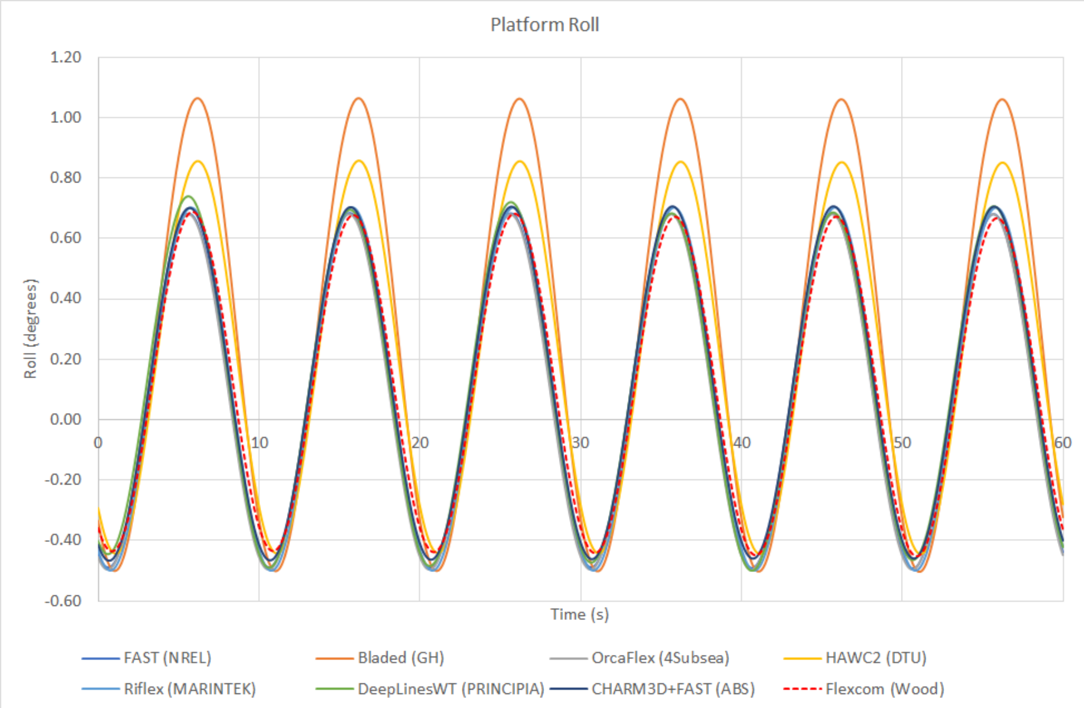
PLATFORM YAW

Flexcom's yaw results appear inconsistent with the other tools. The mean yaw is greater, and the periodic variation in yaw appears to be out of phase. This requires further investigation on our part, but it could be due to a difference in convention only.



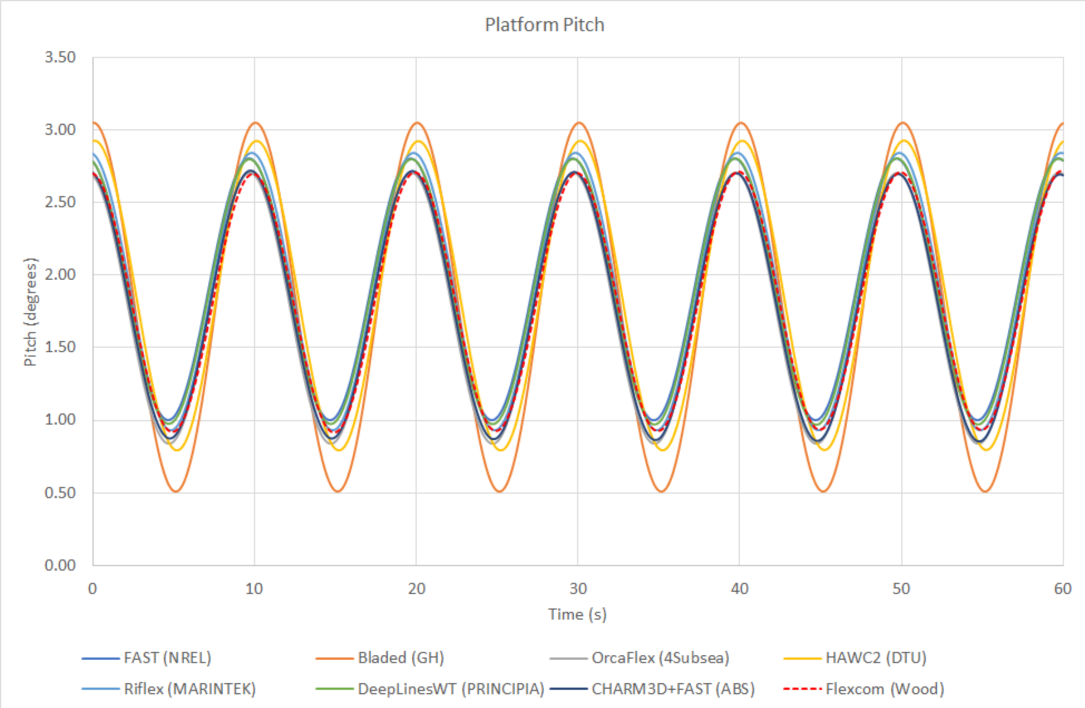
PLATFORM ROLL

Flexcom's roll response shows close agreement with the other software tools, in terms of response amplitude, period and phase. Results from Bladed (GH) show a roll response which is slightly larger than the other software tools.



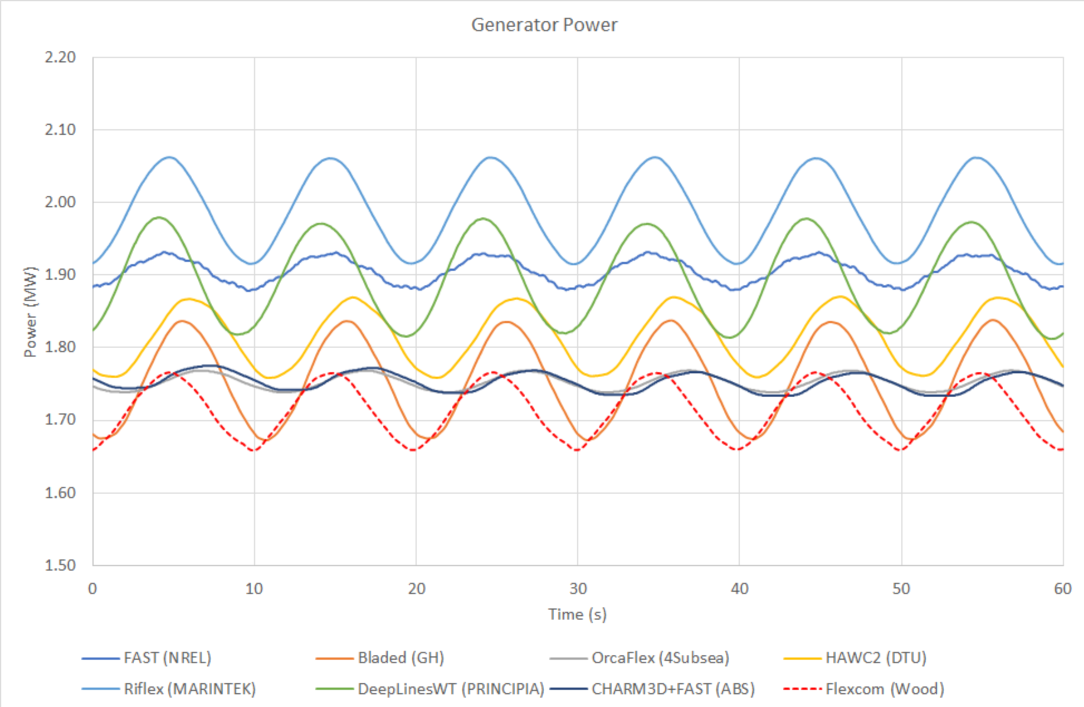
PLATFORM PITCH

Flexcom's pitch response shows reasonable agreement with the other software tools, in terms of response amplitude, period and phase.

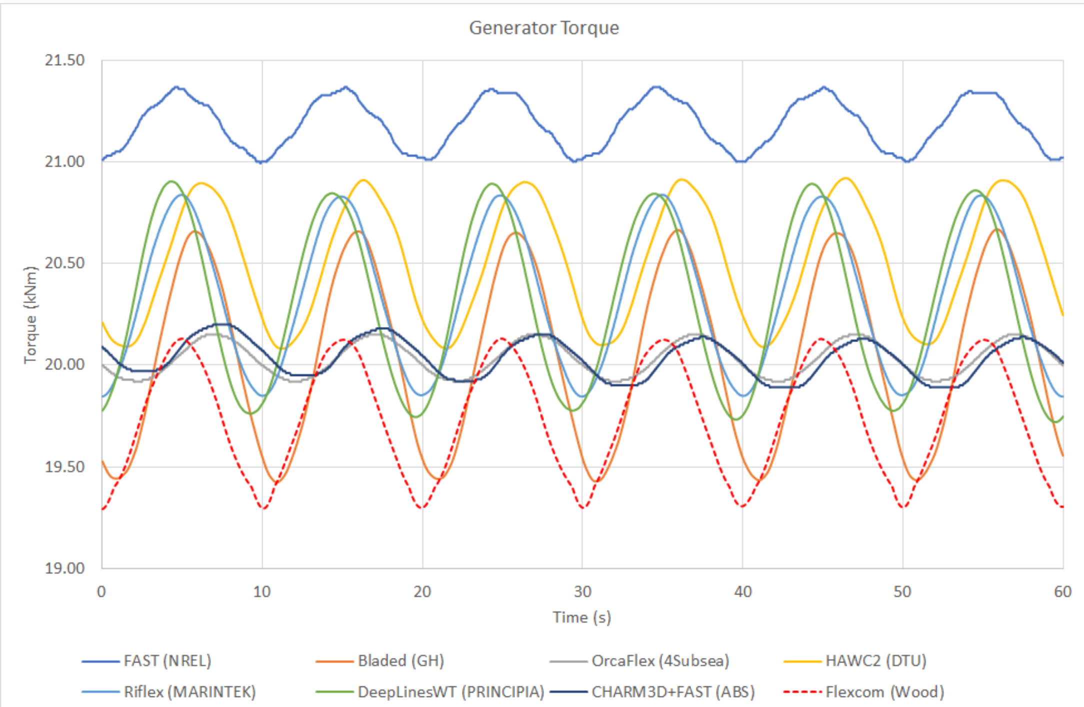


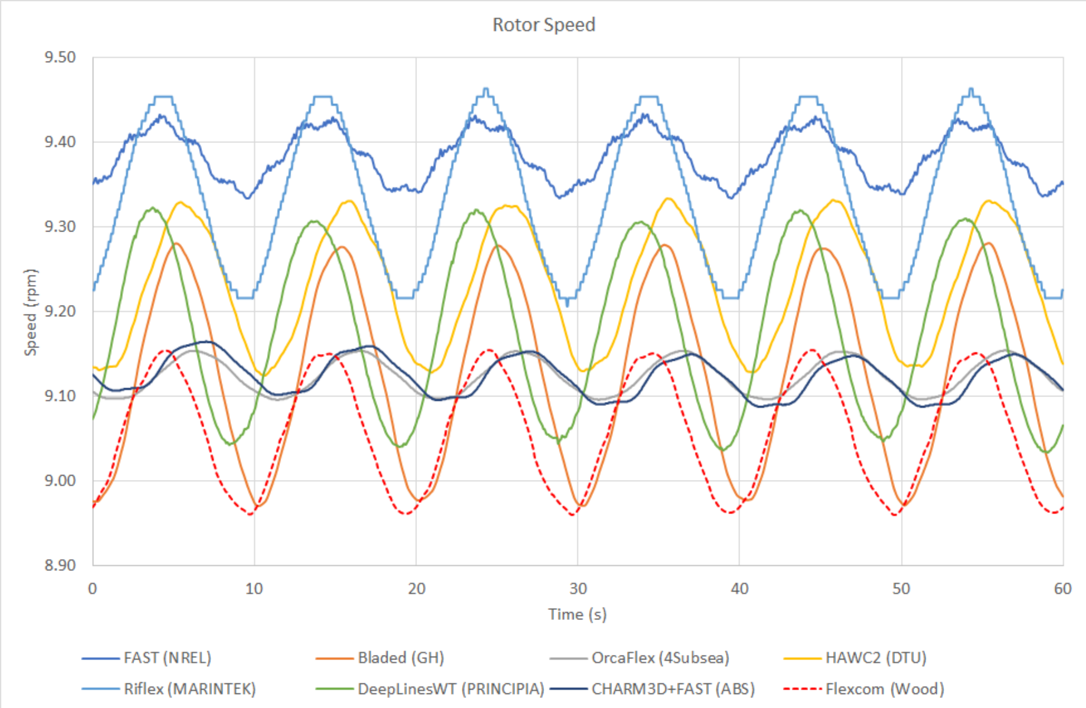
GENERATOR TORQUE, ROTOR SPEED AND GENERATOR POWER

The generator power shows considerable variation across the different software products, with the mean power of the OC4 software tools considered ranging from 1.75MW (Bladed/GH, OrcaFlex/4Subsea, CHARM3D+FAST/ABS) to 1.99MW (Riflex/MARINTEK). The generator power predicted by Flexcom is slightly lower than most of the other software tools - Flexcom's mean power is 1.71MW, 6.7% below the average.



The variations in generator power are also evident in the primary variables of generator torque and rotor speed.





It is worth noting that the Flexcom results were produced using Flexcom 8.12.1 which is coupled with AeroDyn V15.02.04 (14-Apr-2016). The OC4 Phase II results were published in 2014 at which point the latest version of AeroDyn was Version 14. Improvements to the aerodynamic modelling software primarily account for the differences in generator power predicted by Flexcom and NREL/FAST. Slightly different platform motions predicted by each software tool also influences generator power. Variations in the aerodynamic theories underpinning each software product may also be a contributing factor. You are referred to OC4 P2 LC3.1 if you are interested in a more detailed discussion.

OC4 P2 LC3.7 RAO estimation, with wind

This load case is similar to OC4 P2 LC2.6, which does not include wind loading. Generally speaking, the comparisons presented here are very similar also.

Flexcom input files are available for this load case and simulations results are available in the form of random time histories. These need to be converted to RAOs (response amplitude operators) or PSDs (power-spectral densities) before results from Flexcom may be compared to official results from OC4. Once this process is complete, RAO/PSD comparisons will be uploaded here, including:

- Wave height
- Tower Base Moment
- Fairlead Tension in Mooring Line 2
- Platform surge
- Platform heave
- Platform pitch

1.10.12.2 L02 - OC4 Semi-Submersible (Flexcom Wind)

This example is very similar to Example L01 - OC4 Semi-Submersible so you are referred to the documentation on that example for further details.

The key differences between Example L02 and Example L01 are:

- Example L02 is built using Flexcom Wind rather than the main Flexcom environment.
- Example L01 considers 21 different load cases, encompassing varying levels of model complexity and a variety of ambient loading conditions. Example L02 considers just one load case, corresponding to OC4 P2 LC3.1 modified (Control System Test).

1.10.12.3 L03 - OC4 Jacket

OC4 (Offshore Code Comparison Collaboration Continuation) is a code-to-code verification project operated under the IEA (International Energy Agency) Wind Task 30 and coordinated by NREL (National Renewable Energy Laboratory). In Phase I of OC4, participants used an assortment of simulation codes to perform a coupled simulation of a 5-MW wind turbine installed on a jacket support structure in 50 m of water. Code predictions were compared from load case simulations selected to test different model features.

Although Flexcom was not officially represented in OC4, the software has been retrospectively benchmarked against OC4 results, as NREL and the IEA have kindly made all data from the project publicly available. The software validation is primarily focused on the OC4 semi-submersible platform (Connolly & O'Mahony, 2021), but comparisons are available for the OC4 jacket structure (Connolly & O'Mahony, 2021) also.

The overall layout of this example is as follows:

- Introduction gives an overview of the OC4 jacket, the load cases simulated, and the project participants and software tools.
- Model Summary describes the Flexcom model in detail.
- Results from Flexcom are presented alongside results from the software tools used in OC4.

Introduction

OC4 PHASE I

OC4 (Offshore Code Comparison Collaboration Continuation) is a code-to-code verification project operated under the IEA (International Energy Agency) Wind Task 30 and coordinated by NREL (National Renewable Energy Laboratory). In Phase I of OC4, participants used an assortment of simulation codes to perform a coupled simulation of a 5-MW wind turbine installed on a jacket support structure in 50 m of water. Code predictions were compared from load case simulations selected to test different model features. Although Flexcom was not officially represented in OC4, the software has been retrospectively benchmarked against OC4 results, as NREL and the IEA have kindly made all data from the project publicly available.

Popko et al. (2012) summarise the modelling capabilities of the various software tools under the headings of aero-hydro-servo-elastic simulation capabilities. Using the OC4 terminology, Flexcom would be categorised as follows:

- Aerodynamics (aero): Blade-Element Momentum (BEM) or Generalised Dynamic Wake (GDW), plus Dynamic Stall (DS)
- Hydrodynamics (hydro): Airy theory with stretching method (Airy^{str}) or User-defined subroutine (UD) or Dean's stream function (Stream) + Morison's Formula (ME)

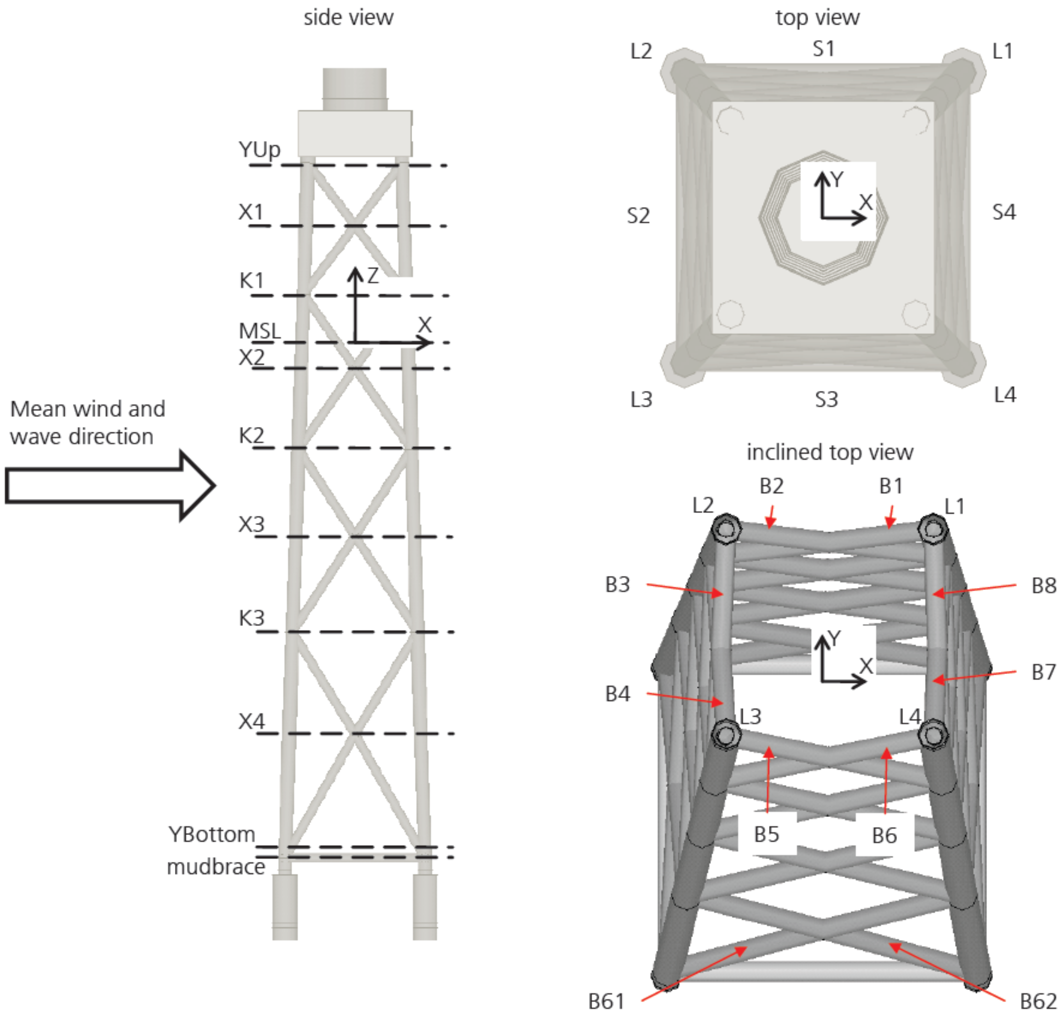
- Control (servo): External dynamic link library (DLL)
- Structural dynamics: Finite Element Method (FEM)

JACKET

In OC4 Phase 1, the NREL 5MW reference turbine is supported by the UpWind reference jacket model described by Vorpahl et al. (2011). The supporting platform consists of a jacket substructure, a transition piece and a tower. The jacket legs are supported by piles, which are modelled as being clamped at the seabed. The legs are inclined from the vertical position and stiffened by four levels of cross braces. Additionally, mudbraces are placed just above the mudline to minimise the bending moment at the foundation piles. The jacket and tower are connected through a transition piece. The elevation of the entire support structure is 88.15 m with a hub height of 90.55 m.

For simplification reasons, it was decided not to include appurtenances on the jacket structure such as boat landings, J-tubes, anodes, cables, ladders etc. and joint cans are not taken into account. At joints, the connecting nodes of elements are defined at the intersection points of the members' centerlines, which leads to overlap of elements in the analysed jacket. Hence the mass of the jacket is overestimated by about 9.7%, though Popko et al. (2012) state there is only a marginal influence coming from overlapping parts on eigenfrequencies and simulated loading. Additional masses such as hydrodynamic added mass, water in flooded legs and marine growth, have a strong influence on the dynamic response of the structure and so are included in the model. Marine growth mass and hydrodynamic added mass are overestimated by about 9.2% and 4.6%, respectively, due to the overlapping members.

As shown below, jacket legs are numbered L1 to L4, jacket sides are numbered S1 to S4, four levels of K-joints are numbered K1 to K4, and four levels of X-joints are numbered X1 to X4. The Flexcom model is built according to this numbering scheme. For example, the line named L1_K3toK2 represents jacket leg no. 1 from K-joint no. 3 to K-joint no. 2. The 64 braces are numbered top-down, from Leg 1 to Leg 4 with increasing leg number (counter clockwise in plan view) as shown below.



OC4 Jacket schematic (Popko et al., 2012)

LOAD CASES

17 different load cases were performed in OC4 Phase I, encompassing varying levels of model complexity and a variety of ambient loading conditions. A subset of these cases were reproduced using Flexcom, and this forms the basis of this example. The load cases are divided in five distinct sections.

0. Mass comparisons - not an official load case section, but verification of correct structural and additional masses is important, given the direct impact on structural dynamics

1. Eigenanalysis - examination of modal properties of flexible structure

2. Rigid offshore wind turbine - rigid structure excited by non-combined wind and wave loads

3. Land based turbine - jacket support structure replaced with tubular tower, designed to verify aerodynamics and control

4. Flexible offshore structure - flexible support structure with no aerodynamics examined under different wave loadings

5. Fully-flexible offshore wind turbine - fully flexible structure examined under the combined action of wind and wave loads

Load Case	Description	Wind	Wave
OC4 P1 LC0.0	Comparison of structural masses (jacket, transition piece, tower and RNA) and additional masses (marine growth, water in flooded legs and hydrodynamic added mass imposed by water surrounding the structure)	No air	No water
OC4 P1 LC1.0a	Eigenanalysis, no gravity or damping, natural frequencies and mode shapes	No air	No water

Load Case	Description	Wind	Wave
OC4 P1 LC1.0b	Eigenanalysis, no gravity or damping, natural frequencies and mode shapes	No air	No water
OC4 P1 LC1.0c	Eigenanalysis, gravity and structural damping included, damped frequencies and mode shapes	No air	No water
OC4 P1 LC1.0d	Eigenanalysis, gravity and structural damping included, damped frequencies and mode shapes	No air	No water
OC4 P1 LC2.1	Static simulation including gravity and buoyancy to MSL	No air	Still water
OC4 P1 LC2.2	Periodic time-series solution	Steady, uniform, no shear, Vhub = 8 m/s	No water
OC4 P1 LC2.3a	Periodic time-series solution	No air	Regular Airy: H = 6 m, T = 10 s

Load Case	Description	Wind	Wave
OC4 P1 LC2.3b	Periodic time-series solution	No air	Regular stream function (Dean, 9th): H = 8 m, T = 10 s
OC4 P1 LC2.4a	PDF, DEL, power spectra	NTM (Kaimal): Vhub =Vr = 11.4 m/s, sx = 1.68 m/s, sy = 1.34 m/s, sz = 0.84 m/s, Lk;x = 340.20m, Lk;y = 113.40m, Lk;z = 27.72m, Lc = 340.20m, Wind shear: a = 0.14	No water
OC4 P1 LC2.4b	PDF, DEL, power spectra	NTM (Kaimal): Vhub = 18 m/s, sx = 2.45 m/s, sy = 1.96 m/s, sz = 1.23 m/s, Lk;x = 340.20m, Lk;y = 113.40m, Lk;z = 27.72m, Lc = 340.20m, Wind shear: a = 0.14	No water
OC4 P1 LC2.5	PDF, DEL, power spectra	No air	Irregular Airy: Hs = 6 m, Tp = 10 s, Pierson-Moskowitz wave spectrum

Load Case	Description	Wind	Wave
OC4 P1 LC3.2	Periodic time-series solution	Steady, uniform, no shear: Vhub = 8 m/s	No water
OC4 P1 LC3.4a	PDF, DEL, power spectra	NTM (Kaimal): Vhub =Vr = 11.4 m/s, sx = 1.68 m/s, sy = 1.34 m/s, sz = 0.84 m/s, Lk;x = 340.20m, Lk;y = 113.40m, Lk;z = 27.72m, Lc = 340.20m, Wind shear: a = 0.14	No water
OC4 P1 LC4.3b	Periodic time-series solution	No air	Regular stream function (Dean, 9th): H = 8 m, T = 10 s
OC4 P1 LC4.5	PDF, DEL, power spectra	No air	Irregular Airy: Hs = 6 m, Tp = 10 s, Pierson-Moskowitz wave spectrum

Load Case	Description	Wind	Wave
OC4 P1 LC5.6	Periodic time-series solution	Steady, uniform, no shear:, Vhub = 8 m/s	Regular stream function (Dean, 9th): H = 8 m, T = 10 s
OC4 P1 LC5.7	PDF, DEL, power spectra	NTM (Kaimal): Vhub = 18 m/s, sx = 2.45 m/s, sy = 1.96 m/s, sz = 1.23 m/s, Lk;x = 340.20m, Lk;y = 113.40m, Lk;z = 27.72m, Lc = 340.20m, Wind shear: a = 0.14	Irregular Airy: Hs = 6 m, Tp = 10 s, Pierson-Moskowitz wave spectrum

SOFTWARE TOOLS AND PROJECT PARTICIPANTS

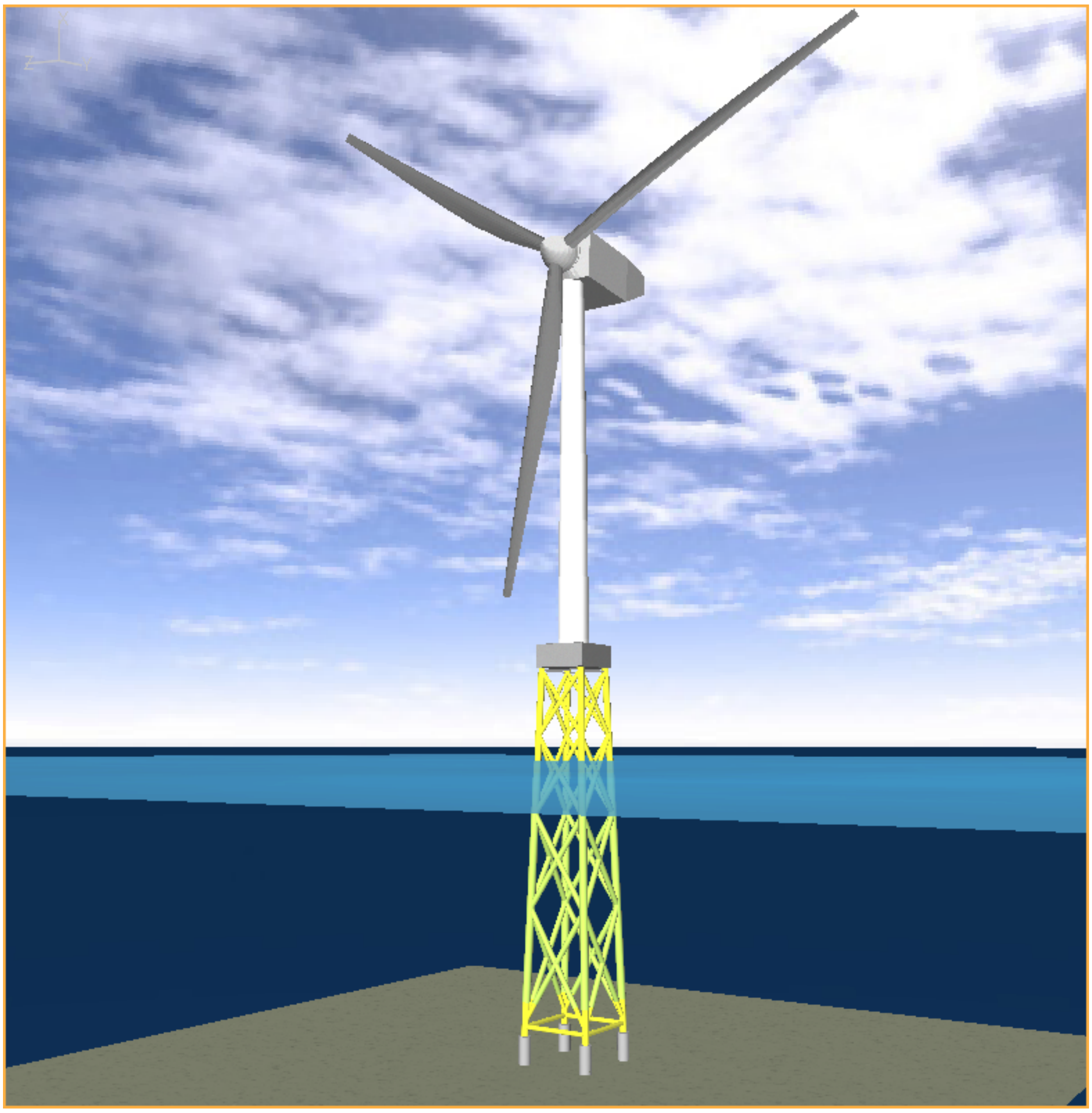
A set of state-of-the-art simulation codes for offshore wind turbine modeling are represented in OC4 Phase I, including: 3DFloat, ADAMS + AeroDyn, ADCoS-Offshore, ASHES, Bladed V3.8X, Bladed V4 Multibody, FAST-ANSYS, FEDEM WindPower, Flex-ASAS, Flex5-Poseidon, GAST, HAWC2, OneWind, Phatas-WMCfem and USFOS-vpOne.

A number of academic and industrial project partners from 10 countries participated in OC4 Phase I, including: Fraunhofer Institute for Wind Energy and Energy System Technology IWES (Germany), the National Renewable Energy Laboratory (NREL) (USA), Technical University of Denmark, Department of Wind Energy, campus Risø, Roskilde, Denmark (Risø DTU) (Denmark), Fedem Technology AS (Norway), Garrad Hassan & Partners Ltd. (UK), Institute for Energy Technology (IFE) (Norway), Pohang University of Science and Technology (POSTECH) (Korea), Centre for Ships and Ocean Structures (CeSOS) at the Norwegian University of Science and Technology (NTNU) (Norway), National Technical University of Athens (NTUA) (Greece), Institute of Steel Construction at the Leibniz Universität Hannover (LUH) (Germany), the Endowed Chair of Wind Energy at the Institute of Aircraft Design at Universität Stuttgart (SWE) (Germany), Norwegian University of Science and Technology (NTNU) (Norway), Knowledge Centre WMC (The Netherlands), Energy Research Centre of the Netherlands (ECN) (The Netherlands), American Bureau of Shipping (ABS) (USA), REpower Systems SE (Germany) and China General Certification (CGC) (China).

Model Summary

INTRODUCTION

The Flexcom model of the OC4 jacket is shown below.



Flexcom model of OC4 Jacket

JACKET

The jacket is modelled using a series of discrete Lines to represent the following items. Refer to the jacket schematic for an illustration of the various components. So in total 100 lines are used to model the jacket alone. This makes model construction somewhat tedious, but as a consistent naming convention is followed, you should not have any difficulty in interpreting this model, or indeed in building your own jacket model if required.

- Legs from mudline to top of piles (4 lines)
- Legs from top of piles to mud braces (4 lines)

- Legs from mud braces to level Y bottom (4 lines)
- Legs from level Y bottom to level K3 (4 lines)
- Legs from level K3 to level K2 (4 lines)
- Legs from level K2 to level K1 (4 lines)
- Legs from level K1 to level Yupper (4 lines)
- Legs from level Yupper to tower base (4 lines)
- Braces from level Yupper to level X1 (8 lines, named B1 to B8)
- Braces from level X1 to level K1 (8 lines, named B9 to B16)
- Braces from level K1 to level X2 (8 lines, named B17 to B24)
- Braces from level X2 to level K2 (8 lines, named B25 to B32)
- Braces from level K2 to level X3 (8 lines, named B33 to B40)
- Braces from level X3 to level K3 (8 lines, named B41 to B48)
- Braces from level K3 to level X4 (8 lines, named B49 to B56)
- Braces from level X4 to level Y bottom (8 lines, named B57 to B64)
- Mud Braces (4 lines, named MB1 to MB4)

Note that these lines are connected up at the intersection points using Equivalent Nodes to form a single coherent structure. This is an essential part of the model definition in Flexcom.

Each line is assigned appropriate Stiffness and Mass Density terms, plus a Buoyancy Diameter (to allow computation of buoyancy forces) and a Drag Diameter (to facilitate computation of Morison drag loads).

TOWER

The tower is constructed using a series of vertical Lines, corresponding to tower sections described in Vorpahl et al. (2011). Each section has uniform diameter and wall thickness (as opposed to tapering sections). The tower sections are assembled together, and connected to the transition piece at the bottom and RNA at the top, using Equivalent Nodes. Note that a consistent mesh density is assigned to the aerodynamic model via the `*TOWER INFLUENCE` keyword. Realistic Stiffness and Mass Density terms are assigned to each tower section based on the material properties for steel and the relevant diameter and wall thickness at the section's mid-point. Flexcom's traditional Buoyancy Diameter and Drag Diameter are unused, but a relevant drag diameter is assigned to each tower aerodynamic node via the *TOWER INFLUENCE keyword. Point Mass terms are placed at the top, midpoint and bottom elevations of the tower to represent flanges, bolts and equipment installed on the tower.

TRANSITION PIECE

The transition piece is a rigid concrete block in reality. It's weight is modelled as a Point Mass in the Flexcom model, and its rigidity is simulated by a series of rigid massless Elements which connect the upper ends of the jacket legs to the base of the tower. The use of explicitly created elements is more convenient than using lines in such circumstances. These elements have zero values of Mass per Unit Length but high Stiffness terms to simulate rigidity. An Auxiliary Profile is used to display the concrete block.

ROTOR-NACELLE-ASSEMBLY (RNA)

Finite element Nodes are explicitly created at the centres of mass of the nacelle, hub and blades (centre of mass at initial position) respectively. These nodes are then connected to the top of the tower using finite Elements (again the use of explicitly created nodes and elements is more convenient than using lines here). Point Mass terms are used to position appropriate masses at the nodes, hence the elements have zero values of Mass per Unit Length. All elements are assigned large Stiffness terms to simulate rigidity. An Auxiliary Profile is used to represent the rotating blades. While this has no structural function, it enhances the visual appeal of the model, and assists in the understanding of rotor and platform motions post-simulation.

AERODYNAMICS

All inputs which are required by AeroDyn to compute the aerodynamic loading on the blades and tower are are logically grouped together under the `$AERODYN` section, and specified in the dynamic simulation file. Fundamental inputs include Blade Geometries, Aerofoil Coefficients, miscellaneous Turbine Inputs (such as hub height, hub radius, overhang, shaft tilt, blade precone etc.) and Tower Influence (i.e. tower drag). These inputs should be intuitively familiar to engineers with some wind turbine modelling experience and you are referred to the keyword documentation should you require further information regarding the significance of any particular input.

CONTROL SYSTEM

The wind turbine control system is defined via the `*SERVODYN` keyword, which references the standard control DLL provided by NREL for the OC4 jacket. At low wind speeds the turbine is operating below rated power, so the rotor is allowed to rotate freely without any control in order to maximise power extraction. At intermediate wind speeds the turbine is fully operational, and the generator torque is used to control rotor speed while maximising generated power. At higher wind speeds the available wind power is above the rated power of the turbine, so blade pitch control is used to feather the blades and shed excess power.

Results

You can use the hyperlinks below to browse through the results comparisons for the subset of OC4 Phase I load cases considered. Official results from OC4 were obtained from the IEA (International Energy Agency) Wind website, or more specifically, the IEA Wind Task 30 results library. Results from Flexcom are presented alongside results from software tools used in OC4 Phase I. For clarity, Flexcom results are overlaid as the last data series in each graph with the data series shown in dashed red to be distinctive.

Load Case	Description	Wind	Wave
OC4 P1 LC0.0	Comparison of structural masses (jacket, transition piece, tower and RNA) and additional masses (marine growth, water in flooded legs and hydrodynamic added mass imposed by water surrounding the structure)	No air	No water
OC4 P1 LC1.0	Eigenanalysis	No air	No water
OC4 P1 LC2.1	Static simulation including gravity and buoyancy to MSL	No air	Still water
OC4 P1 LC2.2	Periodic time-series solution	Steady, uniform, no shear, Vhub = 8 m/s	No water

Load Case	Description	Wind	Wave
OC4 P1 LC2.3a	Periodic time-series solution	No air	Regular Airy: H = 6 m, T = 10 s
OC4 P1 LC2.3b	Periodic time-series solution	No air	Regular stream function (Dean, 9th): H = 8 m, T = 10 s
OC4 P1 LC2.4a	PDF, DEL, power spectra	NTM (Kaimal): Vhub =Vr = 11.4 m/s, sx = 1.68 m/s, sy = 1.34 m/s, sz = 0.84 m/s, Lk;x = 340.20m, Lk;y = 113.40m, Lk;z = 27.72m, Lc = 340.20m, Wind shear: a = 0.14	No water
OC4 P1 LC2.4b	PDF, DEL, power spectra	NTM (Kaimal): Vhub = 18 m/s, sx = 2.45 m/s, sy = 1.96 m/s, sz = 1.23 m/s, Lk;x = 340.20m, Lk;y = 113.40m, Lk;z = 27.72m, Lc = 340.20m, Wind shear: a = 0.14	No water

Load Case	Description	Wind	Wave
OC4 P1 LC2.5	PDF, DEL, power spectra	No air	Irregular Airy: Hs = 6 m, Tp = 10 s, Pierson-Moskowitz wave spectrum
OC4 P1 LC3.2	Periodic time-series solution	Steady, uniform, no shear: Vhub = 8 m/s	No water
OC4 P1 LC3.4a	PDF, DEL, power spectra	NTM (Kaimal): Vhub =Vr = 11.4 m/s, sx = 1.68 m/s, sy = 1.34 m/s, sz = 0.84 m/s, Lk;x = 340.20m, Lk;y = 113.40m, Lk;z = 27.72m, Lc = 340.20m, Wind shear: a = 0.14	No water
OC4 P1 LC4.3b	Periodic time-series solution	No air	Regular stream function (Dean, 9th): H = 8 m, T = 10 s

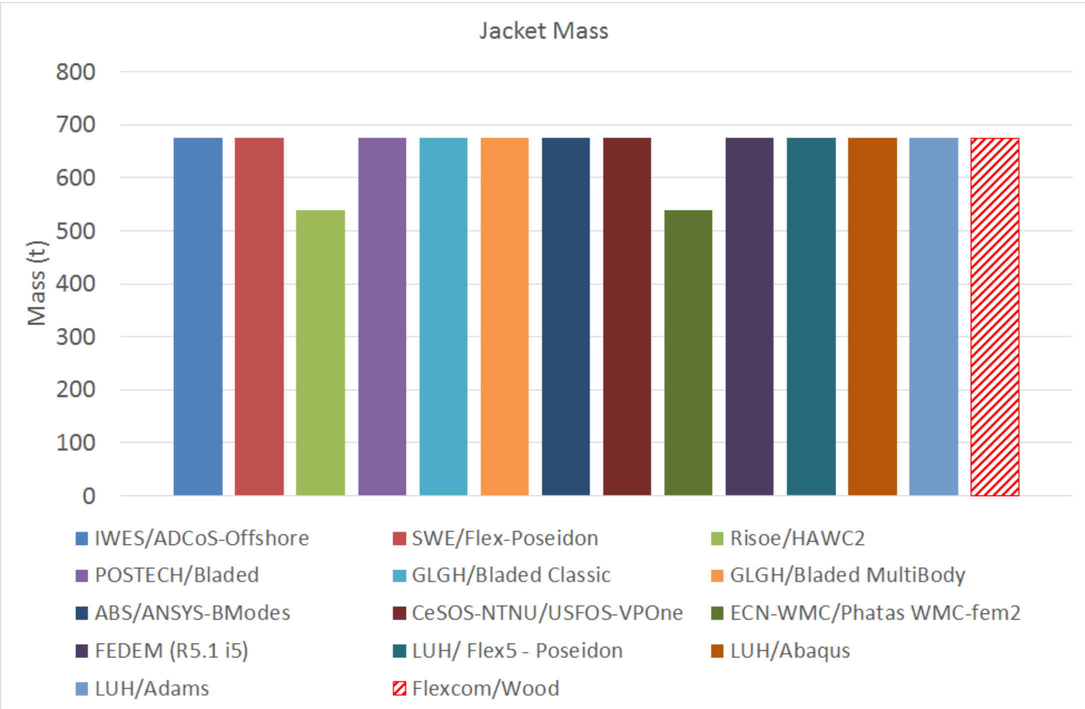
Load Case	Description	Wind	Wave
OC4 P1 LC4.5	PDF, DEL, power spectra	No air	Irregular Airy: Hs = 6 m, Tp = 10 s, Pierson-Moskowitz wave spectrum
OC4 P1 LC5.6	Periodic time-series solution	Steady, uniform, no shear:, Vhub = 8 m/s	Regular stream function (Dean, 9th): H = 8 m, T = 10 s
OC4 P1 LC5.7	PDF, DEL, power spectra	NTM (Kaimal): Vhub = 18 m/s, sx = 2.45 m/s, sy = 1.96 m/s, sz = 1.23 m/s, Lk;x = 340.20m, Lk;y = 113.40m, Lk;z = 27.72m, Lc = 340.20m, Wind shear: a = 0.14	Irregular Airy: Hs = 6 m, Tp = 10 s, Pierson-Moskowitz wave spectrum

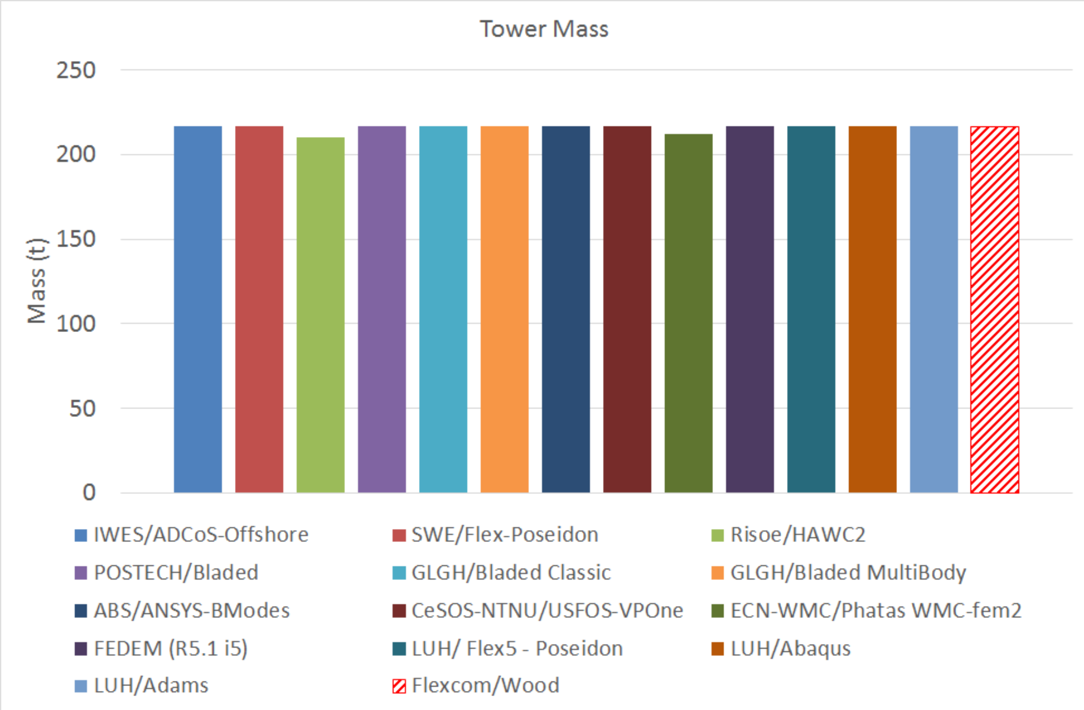
OC4 P1 LC0.0 Mass comparisons

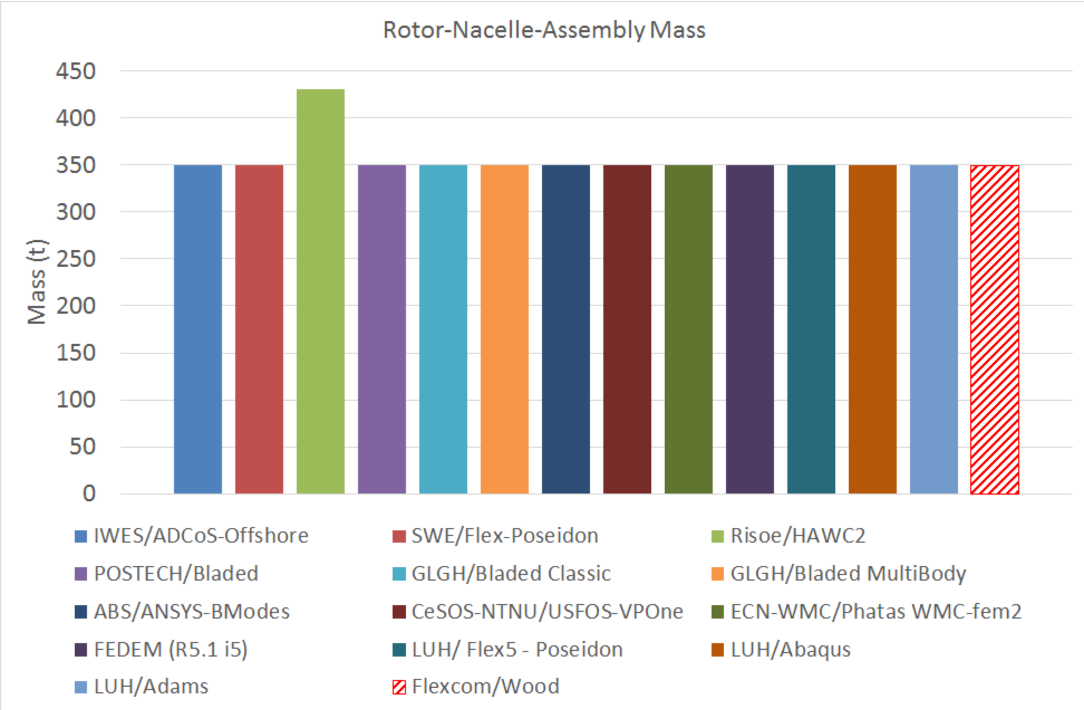
Broadly speaking, very close good agreement is demonstrated between all the software tools. Popko et al. (2012) mention that some minor discrepancies are expected for various reasons, including:

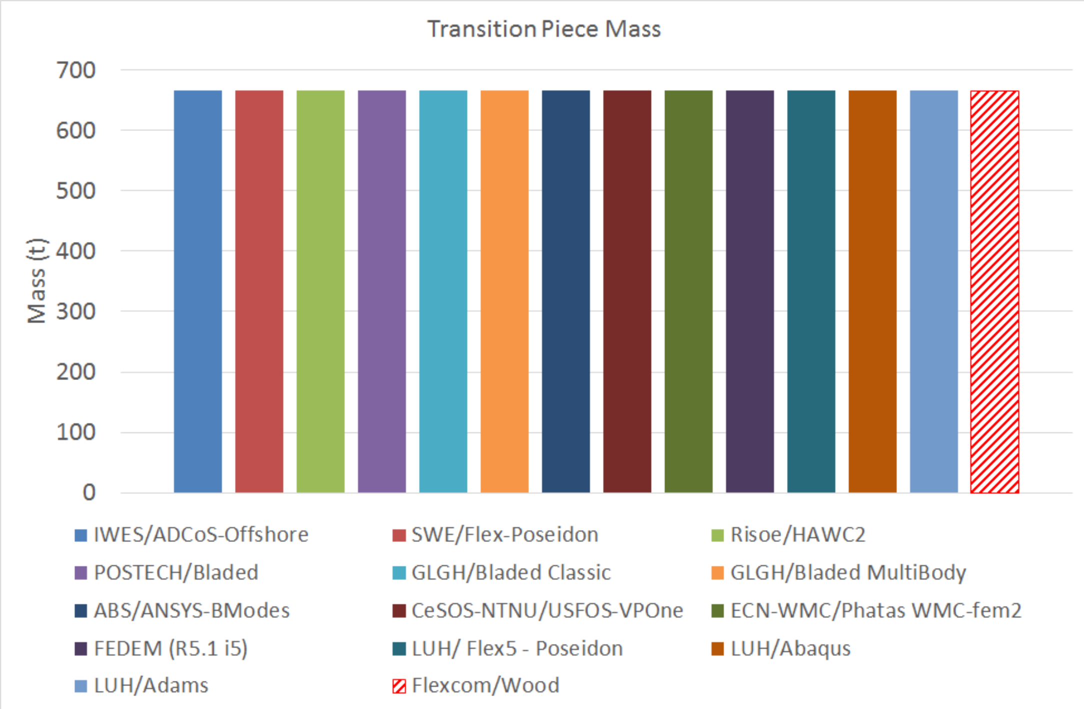
- Differences in RNA masses caused by discretisation of the blade and its mass integration
- Differences in tower masses caused by stepped versus conical elements
- Differences in jacket mass caused by slightly different modelling of the structure across diverse tools
- Differences in hydrodynamic added mass, water mass in free-flooded legs and marine growth caused by discretisation of the threshold regions at critical depths

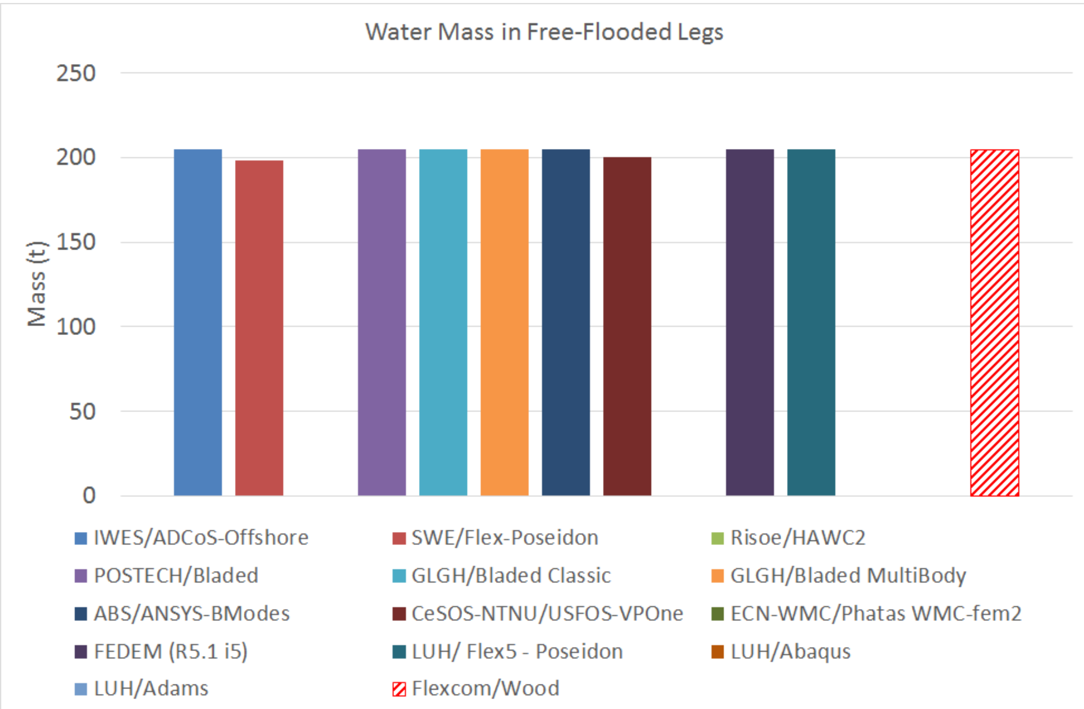
The Flexcom model shows very close correlation with the other software tools for all the component mass terms.

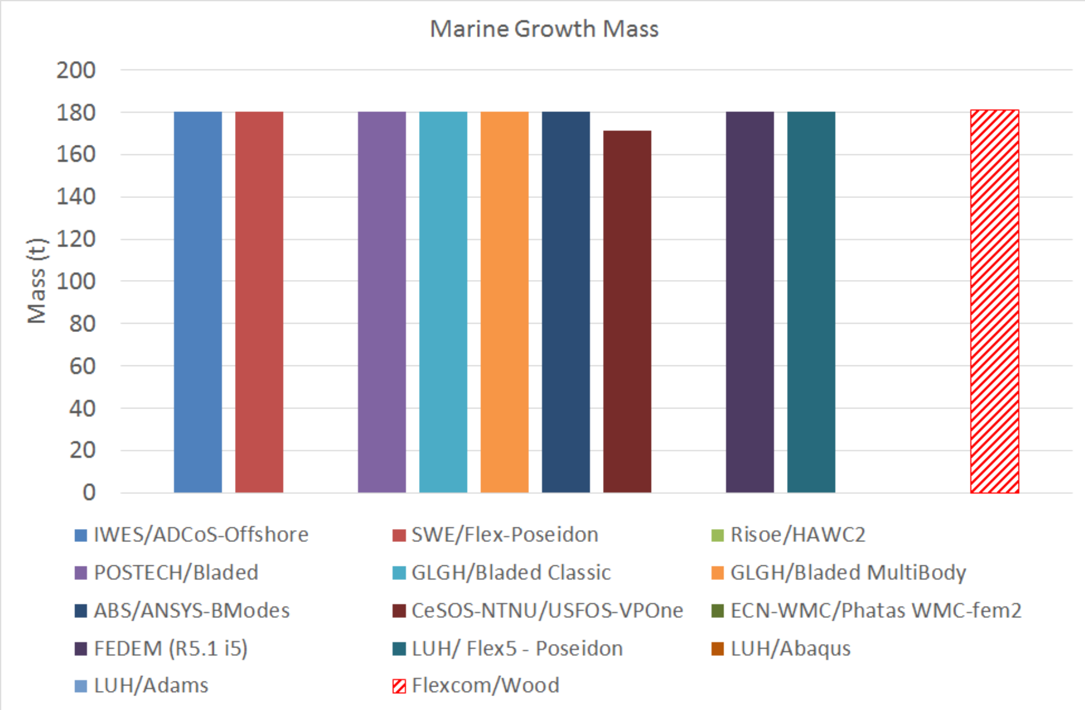



Tower Mass
250
200
150
100
50
0
Mass (t)
IWES/ADCoS-Offshore
SWE/Flex-Poseidon
Risoe/HAWC2
POSTECH/Bladed
GLGH/Bladed Classic
GLGH/Bladed MultiBody
ABS/ANSYS-BModes
CeSOS-NTNU/USFOS-VPOne
ECN-WMC/Phatas WMC-fem2
FEDEM (R5.1 i5)
LUH/ Flex5 - Poseidon
LUH/Abaqus
LUH/Adams
Flexcom/Wood


Rotor-Nacelle-Assembly Mass
450
400
350
300
250
200
150
100
50
0
Mass (t)
IWES/ADCoS-Offshore
SWE/Flex-Poseidon
Risoe/HAWC2
POSTECH/Bladed
GLGH/Bladed Classic
GLGH/Bladed MultiBody
ABS/ANSYS-BModes
CeSOS-NTNU/USFOS-VPOne
ECN-WMC/Phatas WMC-fem2
FEDEM (R5.1 i5)
LUH/ Flex5 - Poseidon
LUH/Abaqus
LUH/Adams
Flexcom/Wood


Transition Piece Mass
Mass (t)
0
100
200
300
400
500
600
700
IWES/ADCoS-Offshore
SWE/Flex-Poseidon
Risoe/HAWC2
POSTECH/Bladed
GLGH/Bladed Classic
GLGH/Bladed MultiBody
ABS/ANSYS-BModes
CeSOS-NTNU/USFOS-VPOne
ECN-WMC/Phatas WMC-fem2
FEDEM (R5.1 i5)
LUH/ Flex5 - Poseidon
LUH/Abaqus
LUH/Adams
Flexcom/Wood


Water Mass in Free-Flooded Legs
Mass (t)
0
50
100
150
200
250
IWES/ADCoS-Offshore
SWE/Flex-Poseidon
Risoe/HAWC2
POSTECH/Bladed
GLGH/Bladed Classic
GLGH/Bladed MultiBody
ABS/ANSYS-BModes
CeSOS-NTNU/USFOS-VPOne
ECN-WMC/Phatas WMC-fem2
FEDEM (R5.1 i5)
LUH/ Flex5 - Poseidon
LUH/Abaqus
LUH/Adams
Flexcom/Wood



OC4 P1 LC1.0 Eigenanalysis

OVERVIEW

Four separate eigenanalyses were studied in OC4 Phase I.

- OC4 P1 LC1.0a: Flexible tower and jacket, rigid RNA. No gravity or damping, natural frequencies and mode shapes extracted.
- OC4 P1 LC1.0b: Flexible tower and jacket, flexible RNA. No gravity or damping, natural frequencies and mode shapes extracted.
- OC4 P1 LC1.0c: Flexible tower and jacket, rigid RNA. Gravity and structural damping included, damped frequencies and mode shapes extracted.
- OC4 P1 LC1.0d: Flexible tower and jacket, flexible RNA. Gravity and structural damping included, damped frequencies and mode shapes extracted.

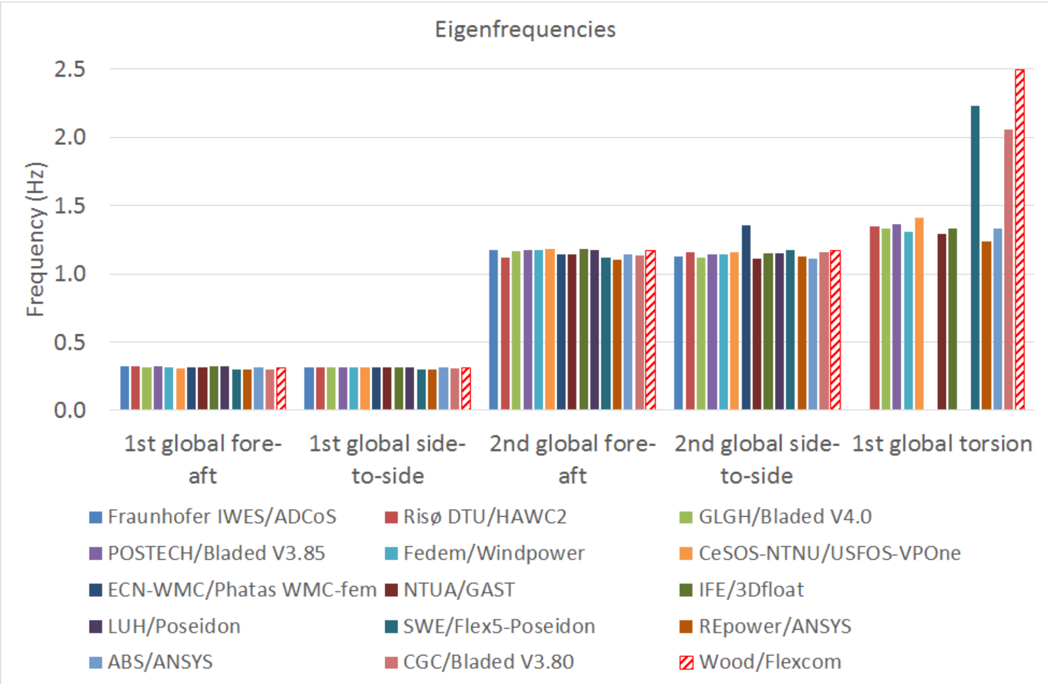
For the purposes of this study, only OC4 P1 LC1.0a is considered.

- The rotating blades are not explicitly modelled in Flexcom 8.12.1 - refer to Software Modelling Limitations for further details). So it is not possible to simulate a fully flexible RNA. So this rules out OC4 P1 LC1.0b and OC4 P1 LC1.0d.

- Flexcom's modal analysis solver computes natural frequencies and associated mode shapes (refer to Modal Analysis for further details). It does not consider damped frequencies and associated mode shapes. So this rules out OC4 P1 LC1.0c.

OC4 P1 LC1.0A

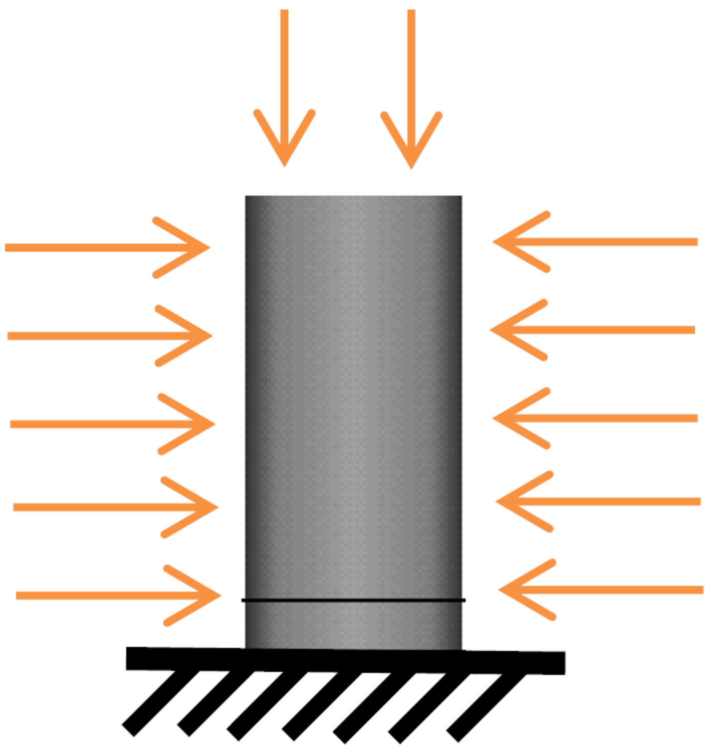
The eigenfrequencies predicted by Flexcom for the entire structure (jacket plus tower) show close agreement with the other software tools. There are no eigenfrequencies associated with the blades as this load case considers a rigid RNA.

- 

OC4 P1 LC2.1 Static Equilibrium

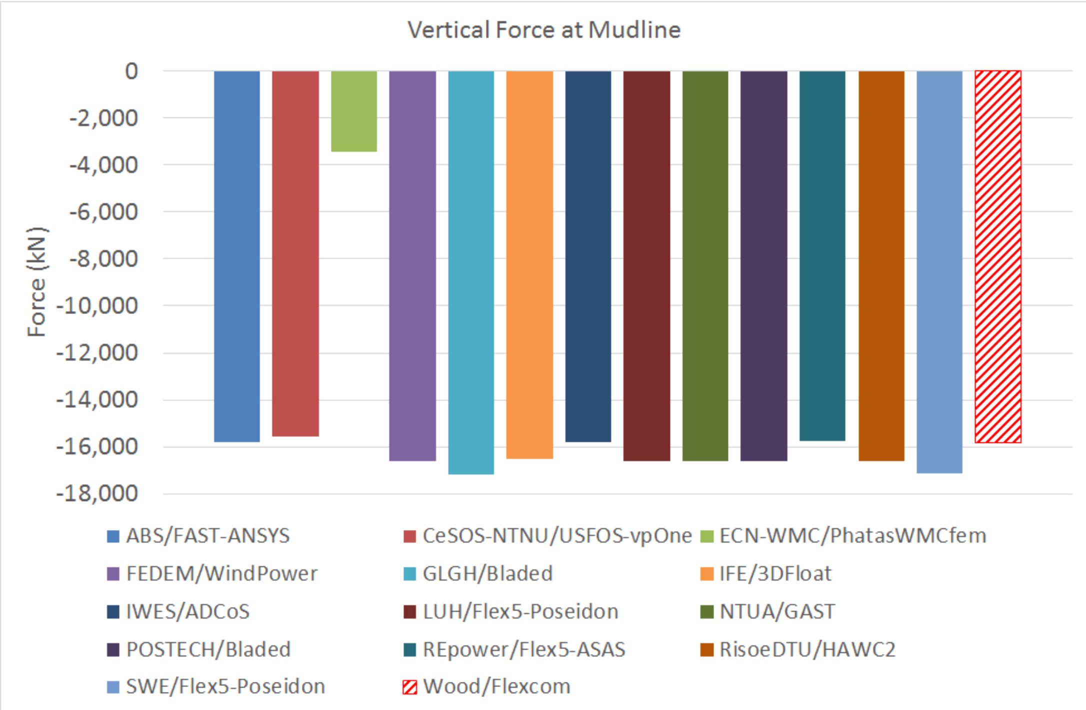
According to Popko et al. (2012), there was a wide discussion within OC4 concerning the modelling strategy for buoyancy and its physical correctness. Buoyancy can be accounted for based on the displaced volume method (which estimates the weight of water displaced by submerged elements) or the pressure integration method (which integrates the external pressure acting on the structure, accounting for all pressure forces imposed on individual members). The latter technique, which is used by Flexcom, is considered to provide a more accurate estimation of buoyancy and should be used in the analysis of jackets according to Popko et al.

The jacket structure modelled in OC4 is cut and clamped fixed at the mudline, so there is no upward buoyant force acting on the cross-sectional area of a pile that is in contact with the seabed, as described in Clauss et al. (2014). Hence Popko advises that no upward pressure should be included in the model at this point.



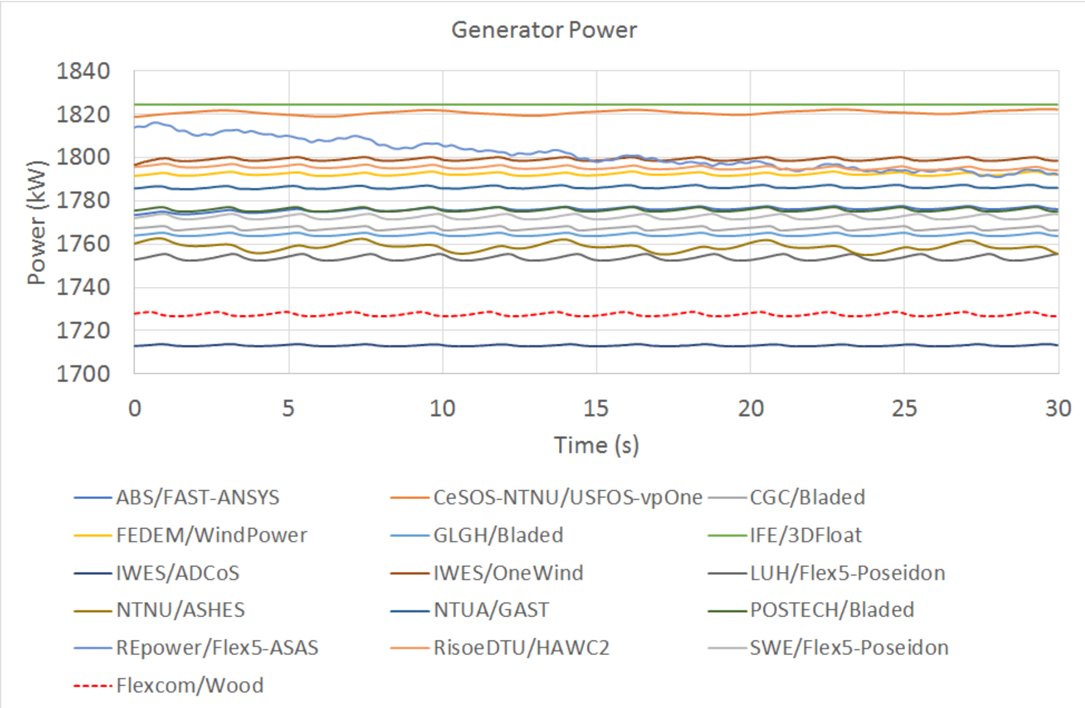
Pressure Forces acting on Pile (Popko et al., 2012)

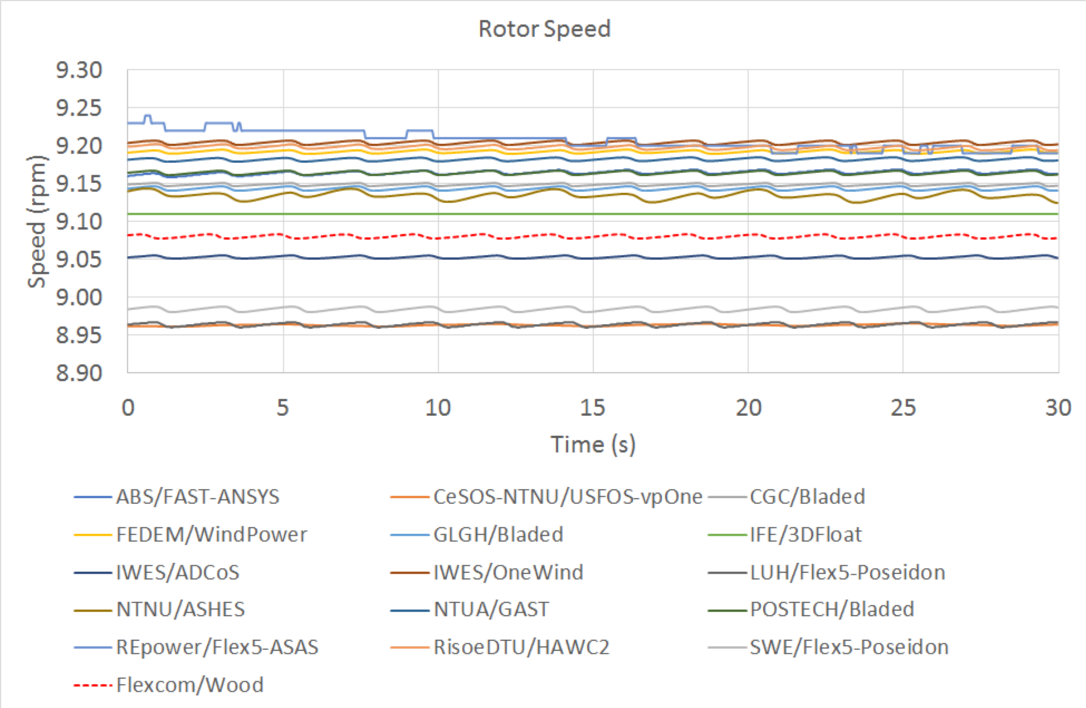
There are two distinct groupings in terms of the vertical force acting at the mudline. The project participants which omitted the buoyancy force at the mudline predict a vertical force at about -16600 kN, while others (including Flexcom) which included this term predict a vertical force of about -15800 kN. In Bladed V4.0, an additional pressure force applied on the top of the grouted piles led to a force of about -17150 kN. For SWE/Flex5-Poseidon, buoyancy of legs was ignored, so pressure integration was only applied on the surface of sealed braces, leading to a force of about -17100 kN. Flexcom applies a pressure force at the mudline based on the local cross-sectional area of each jacket leg, apparently contradicting advice from the OC4 coordinators. However, because the pressure force is applied at restrained nodes, it has no effect on the tension in the jacket legs or on the overall structural response. Hence Flexcom's structural model remains accurate, and any discrepancy in reaction forces is simply due to the computational method adopted for this parameter alone.



OC4 P1 LC2.2 Steady Wind

Good agreement is shown between all software tools for the mean generator power and rotor speed. The tower blockage effect is captured by most tools, including Flexcom, as small fluctuations of the generator power and rotor speed. The generator power predicted by Flexcom is slightly lower than most of the other software tools (apart from ADCoS which had an acknowledged miscalculation of the flow stagnation effect in front of the tower during OC4). Flexcom's mean power is 1727.6 kW, just 2.5% below the average (1780.1 kW). It is worth noting that there is a 6.2% variation across the other software tools also. Differences in mean rotor speed is less than 2.7%. Flexcom's results are very close to the median of OC4 data.



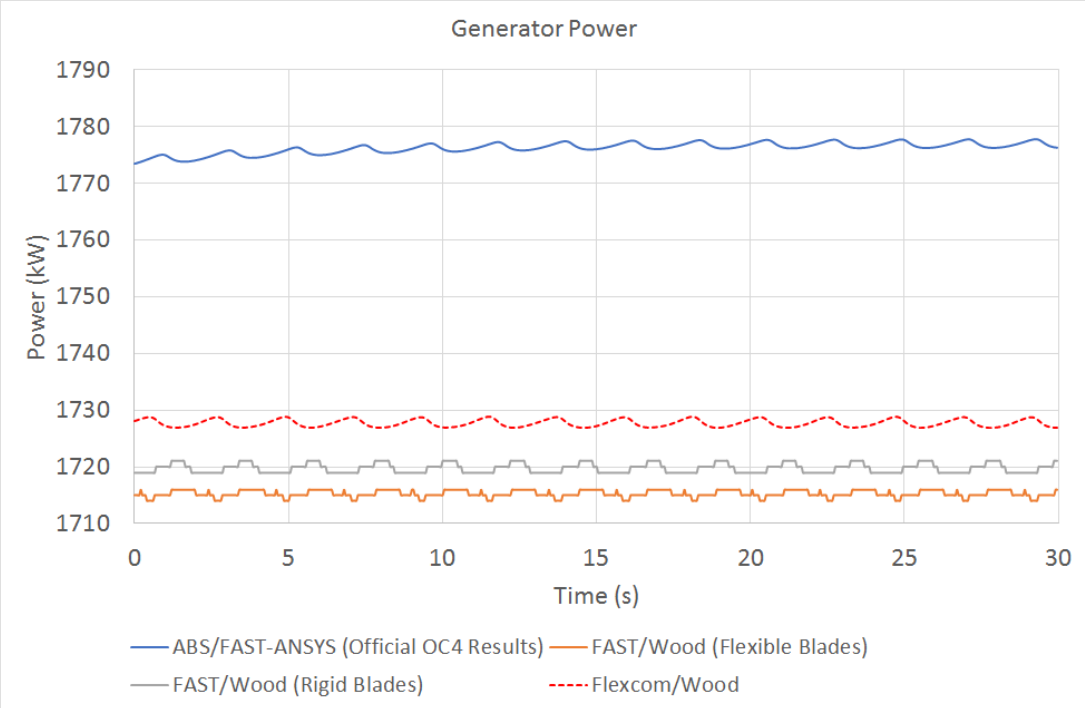


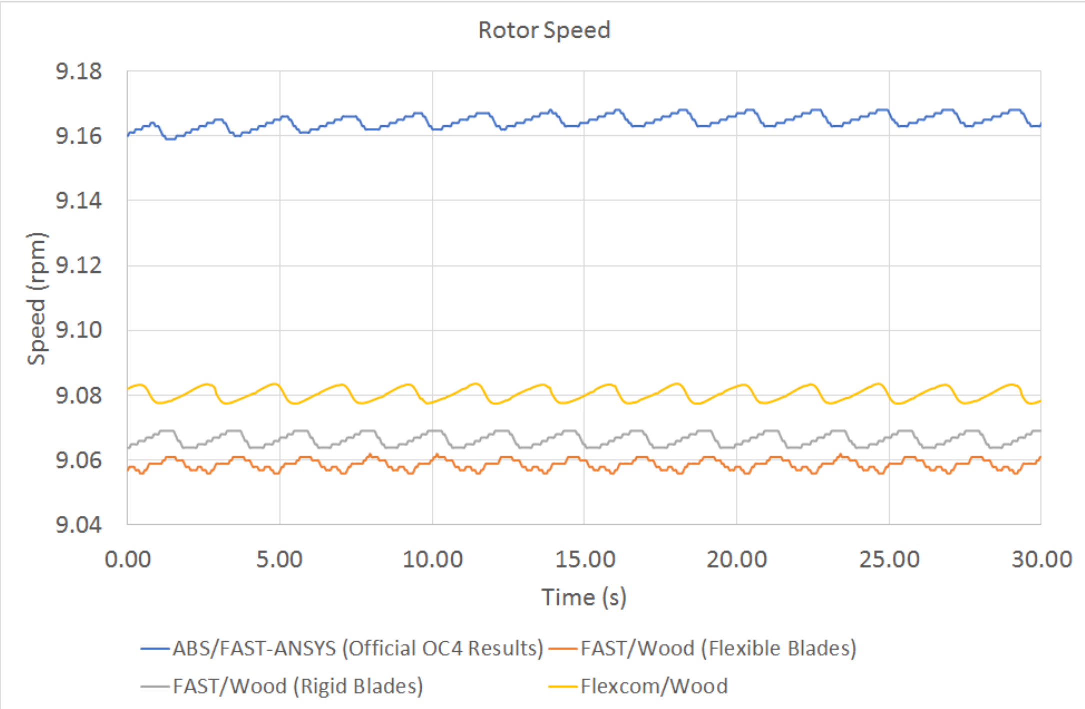
Given that generator power is such a fundamental output for a wind turbine, this issue merits further discussion. Discrepancies in power may be caused by several factors, including...

1. Variations in the aerodynamic theories underpinning each software product
2. Inconsistencies in user selection of aerodynamic modelling options within the software
3. Participants using different versions of the same software product
4. Slight discrepancies in computed structural deformations which influence aerodynamic performance

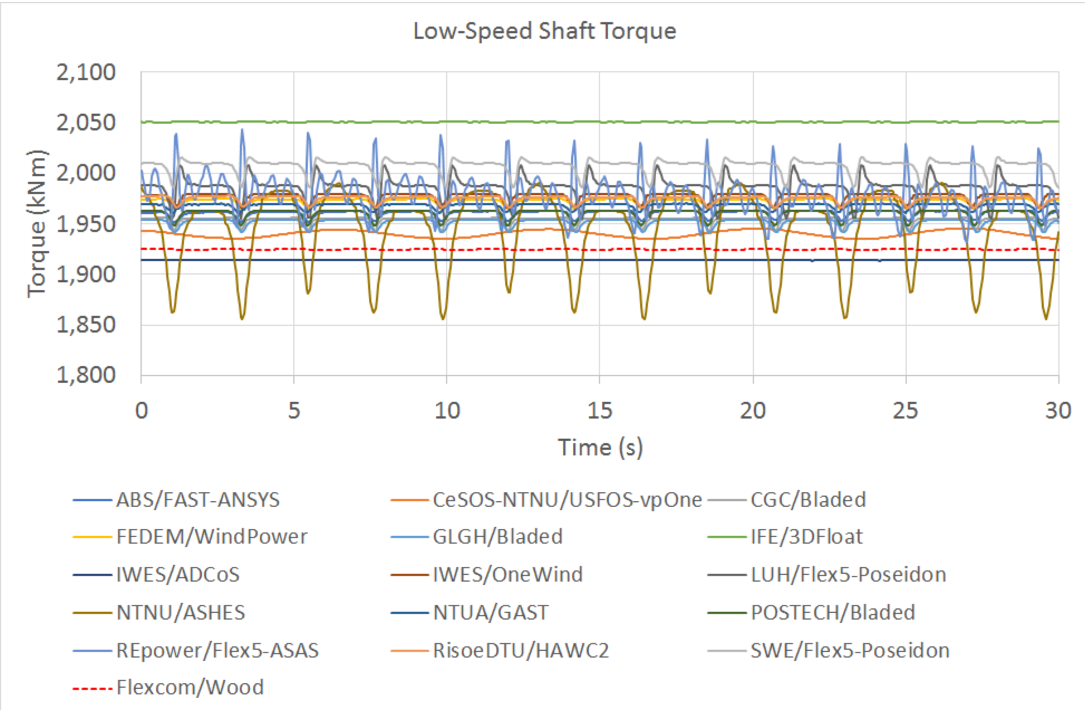
To better understand the discrepancies, this load case was also examined with NREL's FAST software. An official release of FAST was downloaded, Version 8.16, which appeared to be the latest version available at the time of writing. Although the sample input files provided by NREL corresponded to OC4 P1 LC5.7 (these were the only input files published by NREL), it was a simple task to recreate the input files required for OC4 P1 LC2.2, by removing the wave loading, and making the jacket and tower rigid by disabling their respective degrees of freedom. Key output parameters, such as mean generator power and rotor speed, predicted by FAST V8.16 were found to be lower than those presented in Popko et al. (2012) for ABS who used a combination of FAST and the structural modelling software ANSYS. When discrepancies in results were queried with NREL via their online user forum, NREL advised that "*the CertTest model (i.e. the more recent version) uses AeroDyn15 with BEMT (i.e. blade element momentum theory), while the original OC4 simulation used AeroDyn14 with DYNIN (i.e. dynamic inflow)*", and consequently that "*due to these different modelling approaches it is expected to observe some differences in predicted system response between the two models*". Although the query and response relate to the semi-submersible platform of OC4 Phase II (see OC4 P2 LC3.1 Deterministic, below rated for a similar discussion), differences in aerodynamic results are relevant to the jacket structure also.

Flexcom's mean generator power is 2.7% lower than the original ABS results. However, when the simulation is run with FAST V8.16, Flexcom's mean power agrees to within 0.7%, and this discrepancy reduces even further to 0.5% when rigid blades are used in the FAST model. Hence it is concluded that much of the difference between Flexcom and the official OC4 results presented by ABS stem from different versions of the aerodynamic modelling software AeroDyn. The remaining discrepancy may be attributed to the simplified RNA model used in the Flexcom simulations. Furthermore, Flexcom's mean rotor speed is closer to 9rpm, which is consistent with the official load case definition.

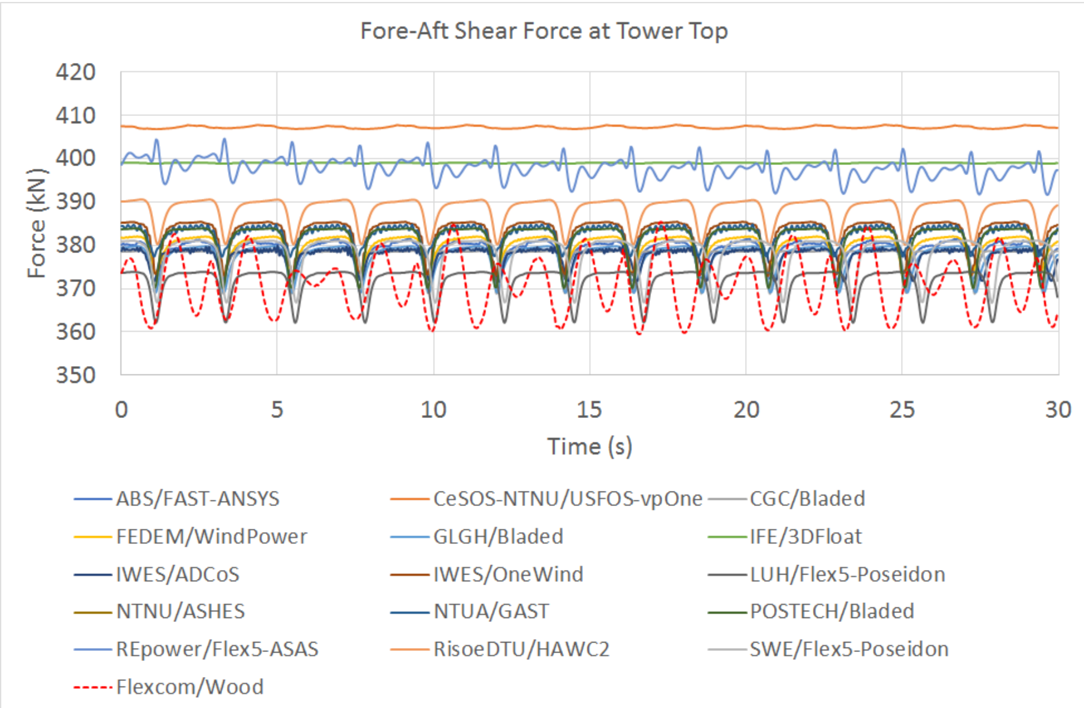

Generator Power
1790
1780
1770
1760
1750
1740
1730
1720
1710
Power (kW)
0
5
10
15
20
25
30
Time (s)
ABS/FAST-ANSYS (Official OC4 Results)
FAST/Wood (Flexible Blades)
FAST/Wood (Rigid Blades)
Flexcom/Wood


Rotor Speed
9.18
9.16
9.14
9.12
9.10
9.08
9.06
9.04
Speed (rpm)
0.00
5.00
10.00
15.00
20.00
25.00
30.00
Time (s)
ABS/FAST-ANSYS (Official OC4 Results)
FAST/Wood (Flexible Blades)
FAST/Wood (Rigid Blades)
Flexcom/Wood

Similar to generator power, the generator torque shows close agreement between all tools. It is accurately predicted by Flexcom as this information comes directly from ServoDyn, the control and electrical drive dynamics module of FAST. The low speed shaft torque shows good agreement between many of the software tools, with most codes showing downward peaks corresponding to the blade passing (3P) frequency. These peaks are not demonstrated by the Flexcom signal, given that a simplified RNA model is used in Flexcom 8.12.1. Specifically, the shaft is not modelled explicitly, and is instead represented by a rigid element which connects the hub location to the top of the tower. This element is non-rotational and its sole purpose is to transfer the aerodynamics loads from the hub to the tower. As the low-speed shaft torque is unknown in the current Flexcom model, it is derived from the generator torque and the gearbox ratio, and this leads to a smoother signal than most of the other modelling tools. It is interesting to note that the low-speed shaft torque is calculated indirectly for the IWES/ADCoS results also, and these results show close agreement with Flexcom.

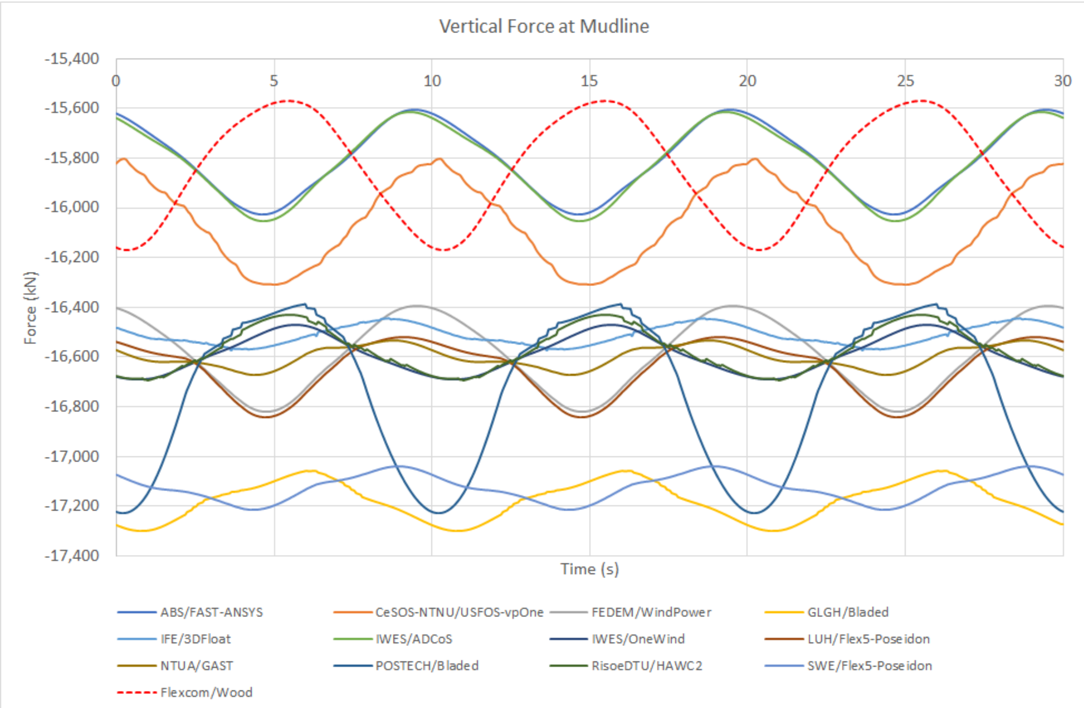


Like the shaft torque, the fore-aft shear force at the top of the tower shows good agreement between most of the modelling tools. The time history also shows the characteristic downward peaks corresponding to the 3P frequency. Although the mean force predicted by Flexcom is similar to the others, just 3.5% below the average, the variation over time is not accurately captured due to the lack of modelling detail for the RNA. The simplified RNA model in Flexcom 8.12 considers that blade deformations under applied loading are negligible, hence the blade geometries are approximated as rigid profiles. Consequently, blade rotational inertia effects are not included in the simulation, which accounts for the differences between the Flexcom solution and several of the other tools that employ more detailed modelling of the RNA. Work is presently under way to develop a more detailed RNA model that explicitly models the blades with finite elements, thereby allowing blade deformations and rotational inertia to be accurately captured.

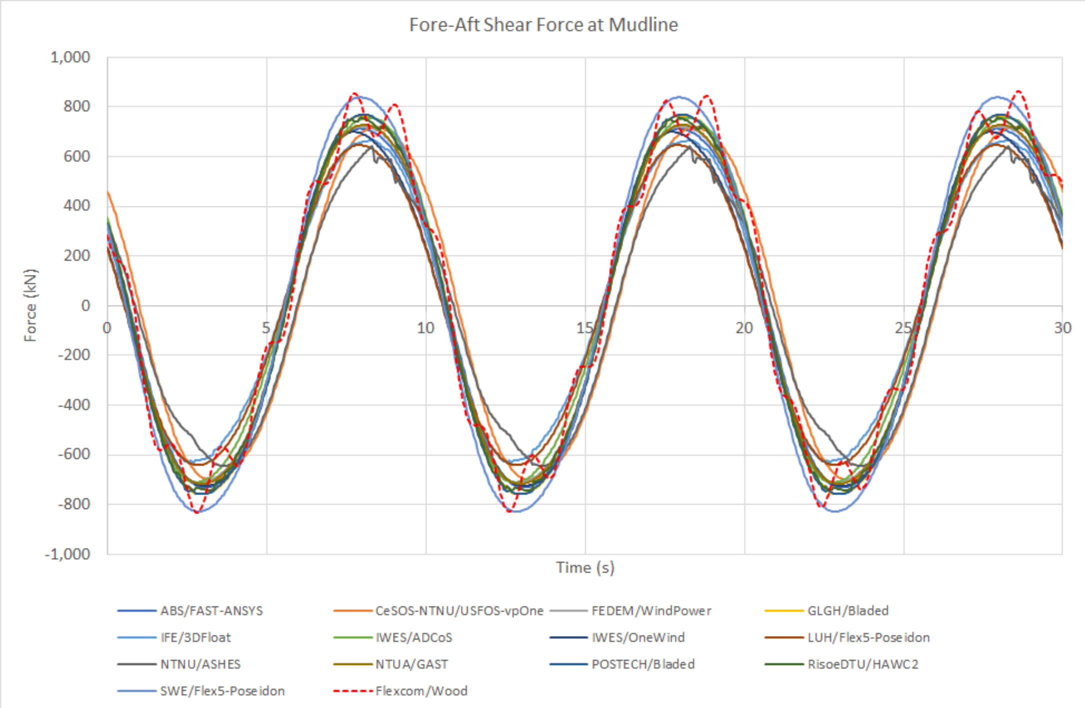


OC4 P1 LC2.3a Airy Wave

The vertical force at the mudline predicted by Flexcom shows good agreement with the other software tools. As outlined in OC4 P1 LC2.1, the project participants which omitted the buoyancy force at the mudline predict a mean vertical force at about -16600 kN, while others (including Flexcom) which included this term predict a mean vertical force of about -15800 kN. According to Popko et al. (2012), differences in peak-to-peak amplitude of vertical force are mainly due to the buoyancy calculation method, whether displaced volume or pressure integration. Additionally, different masses for the jacket substructure and marine growth contributed to discrepancies.



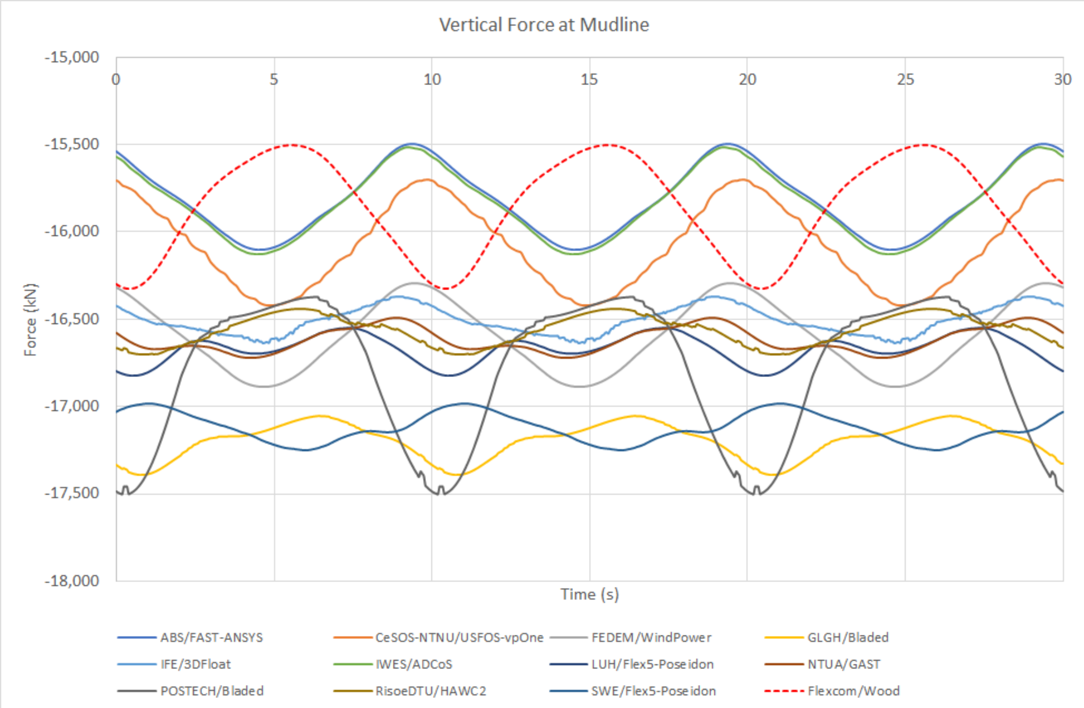
Very close agreement is shown between all software tools for the fore-aft shear force at the base of the jacket. Popko et al. mention that minor discrepancies in peak-to-peak amplitude may be caused by dissimilarities in implementations of wave kinematics, or slight differences originating in the modelling of the support structure.



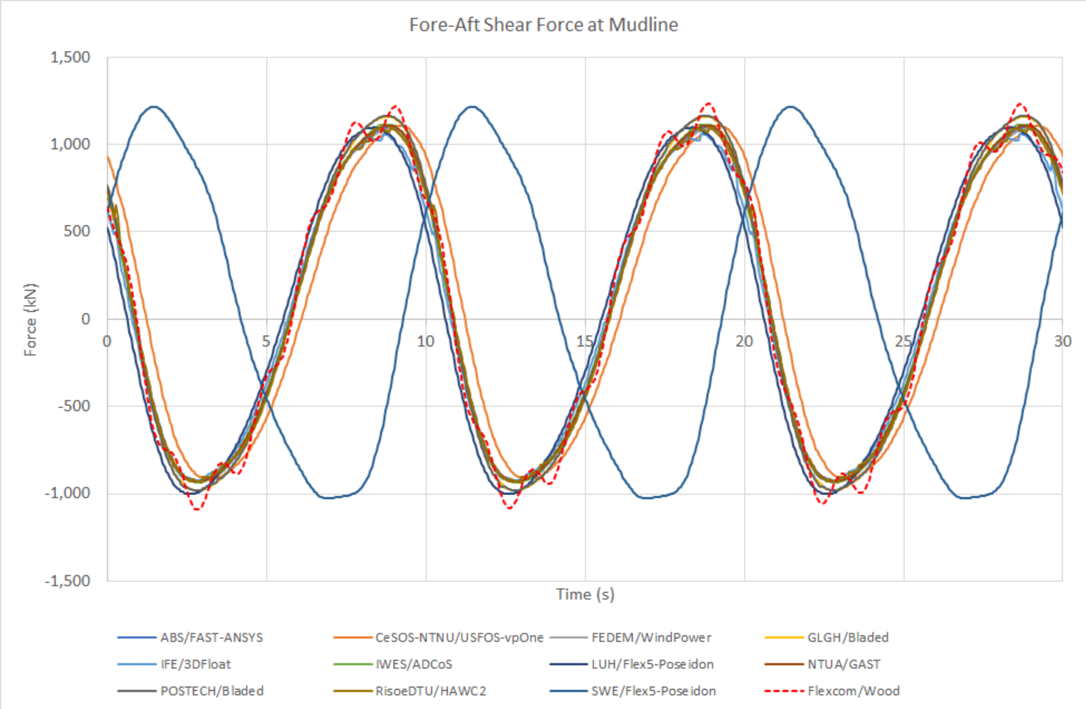
OC4 P1 LC2.3b Deans Wave

Although this load case is not discussed by Popko et al. (2012), results exhibited here show similar trends to the previous load case, OC4 P1 LC2.3a, which considers Airy wave loading rather than a Dean's stream wave.

The vertical force at the mudline predicted by Flexcom shows good agreement with the other software tools. As outlined in OC4 P1 LC2.1, the project participants which omitted the buoyancy force at the mudline predict a mean vertical force at about -16600 kN, while others (including Flexcom) which included this term predict a mean vertical force of about -15800 kN. Differences in peak-to-peak amplitude of vertical force are mainly due to the buoyancy calculation method, whether displaced volume or pressure integration. Additionally, different masses for the jacket substructure and marine growth contributed to discrepancies.



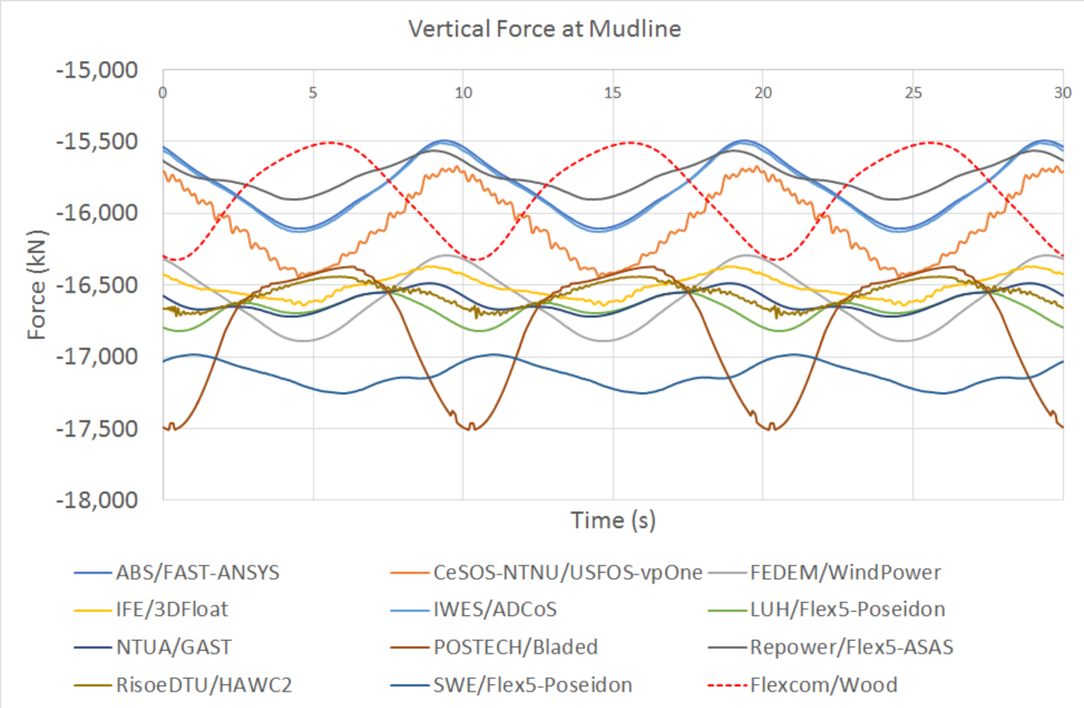
Very close agreement is shown between all software tools for the fore-aft shear force at the base of the jacket. Minor discrepancies in peak-to-peak amplitude may be caused by dissimilarities in implementations of wave kinematics, or slight differences originating in the modelling of the support structure. Results from SWE/Flex5-Poseidon appear out of phase with the other tools - a minor fault in the software at that time caused an unexpected frequency shift in the output.



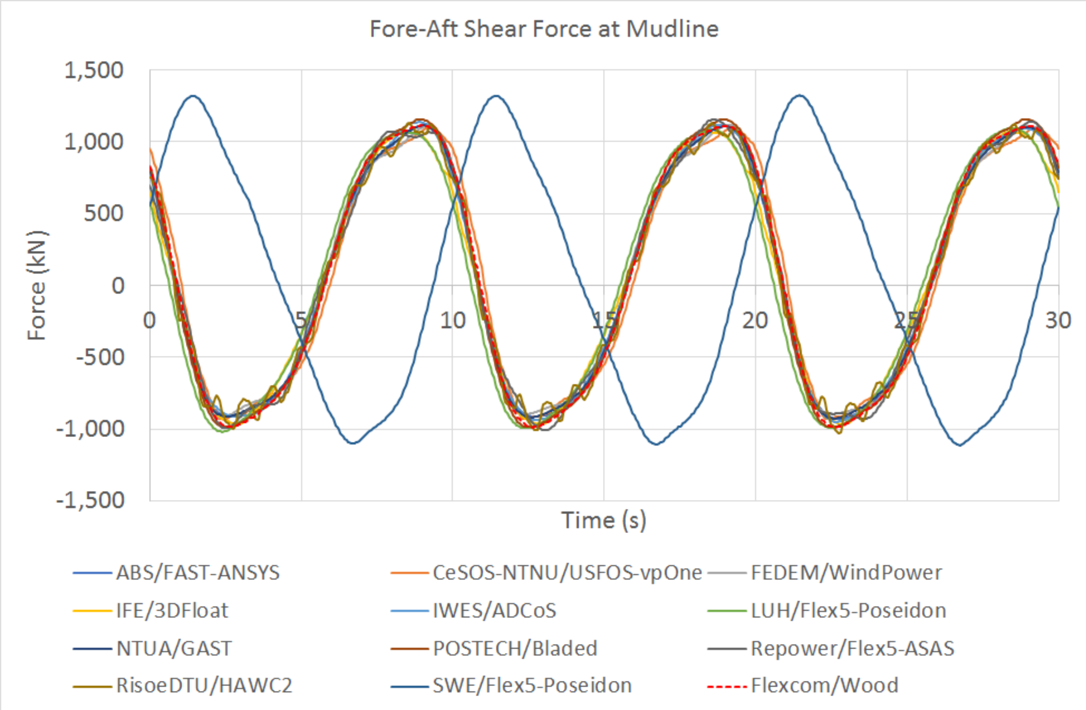
OC4 P1 LC4.3b Deans Wave

This load case is very similar to OC4 P1 LC2.3b, which considers a rigid offshore wind turbine rather than a flexible offshore structure. Results are naturally very similar also.

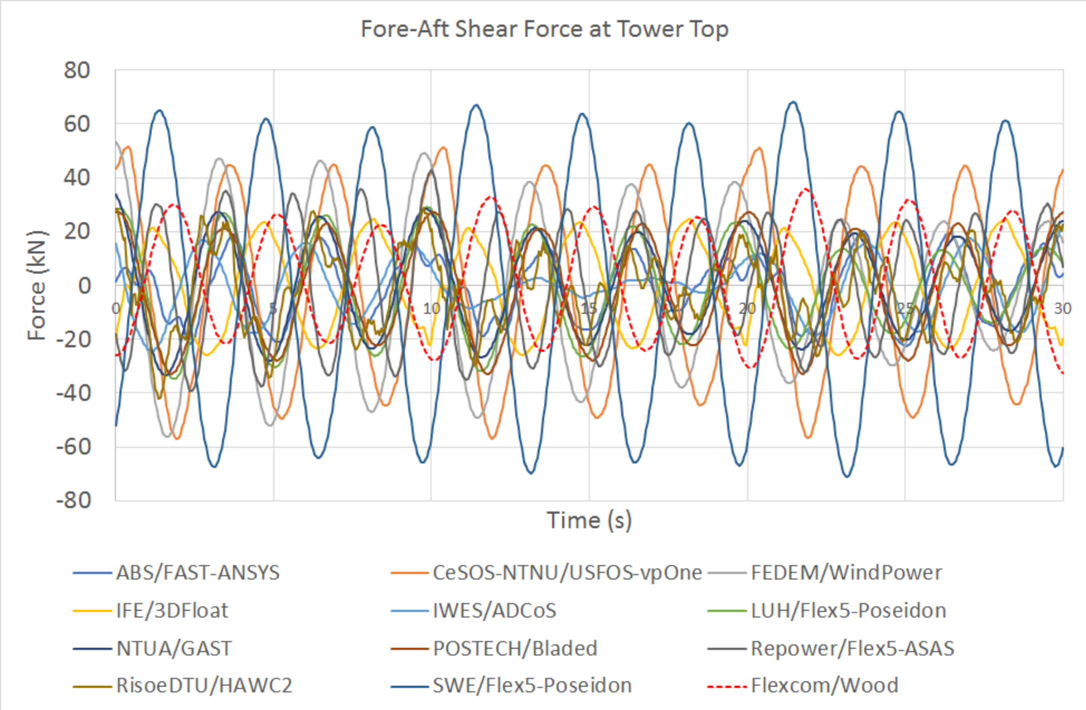
The vertical force at the mudline predicted by Flexcom shows good agreement with the other software tools. As outlined in OC4 P1 LC2.1, the project participants which omitted the buoyancy force at the mudline predict a mean vertical force at about -16600 kN, while others (including Flexcom) which included this term predict a mean vertical force of about -15800 kN. Differences in peak-to-peak amplitude of vertical force are mainly due to the buoyancy calculation method, whether displaced volume or pressure integration. Additionally, different masses for the jacket substructure and marine growth contributed to discrepancies.



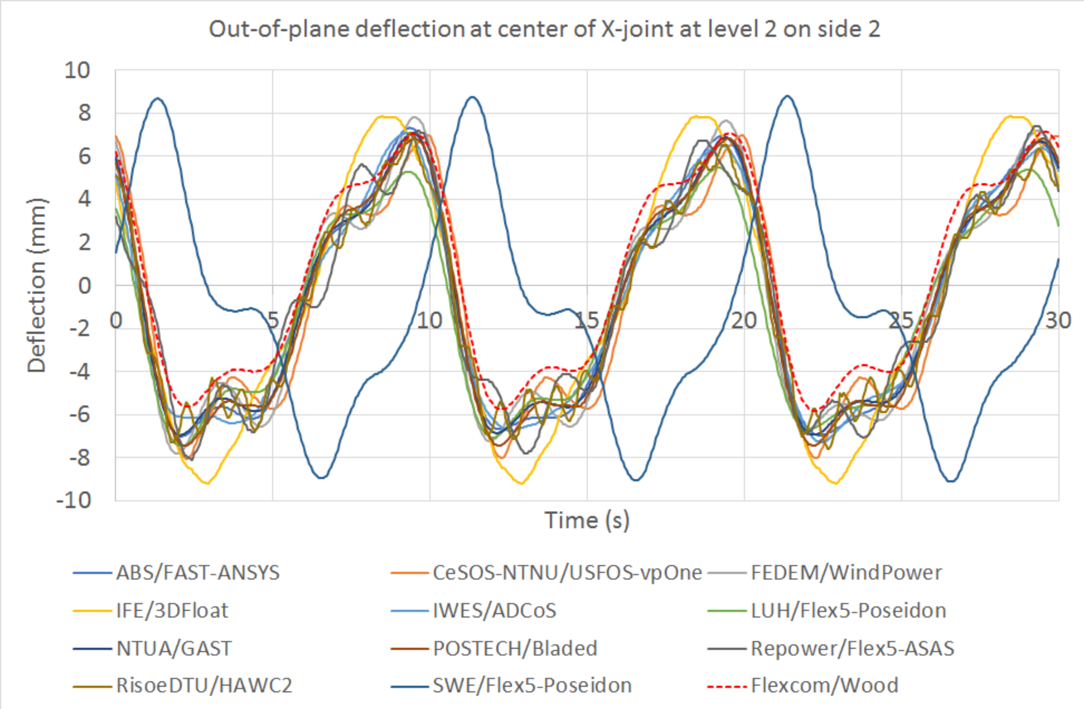
Very close agreement is shown between all software tools for the fore-aft shear force at the base of the jacket. Minor discrepancies in peak-to-peak amplitude may be caused by dissimilarities in implementations of wave kinematics, or slight differences originating in the modelling of the support structure. Results from SWE/Flex5-Poseidon appear out of phase with the other tools - a minor fault in the software at that time caused an unexpected phase shift in the output.



The fore-aft shear force at the top of the tower shows considerable variation across the different software products. However the response predicted by Flexcom shows good general agreement with many of the other tools in terms of response amplitude and period.



Popko et al. (2012) note that the out-of-plane displacement in the fore-aft direction, of the central joint of the cross-brace at level 2 on side 2 (see OC4 Jacket schematic for further details), shows a remarkably good agreement across the different software products. Close agreement is also demonstrated by Flexcom. As with the fore-aft shear force at the mudline, results from SWE/Flex5-Poseidon are out of phase with the other tools due to a minor fault in the software that caused an unexpected phase shift in the output.

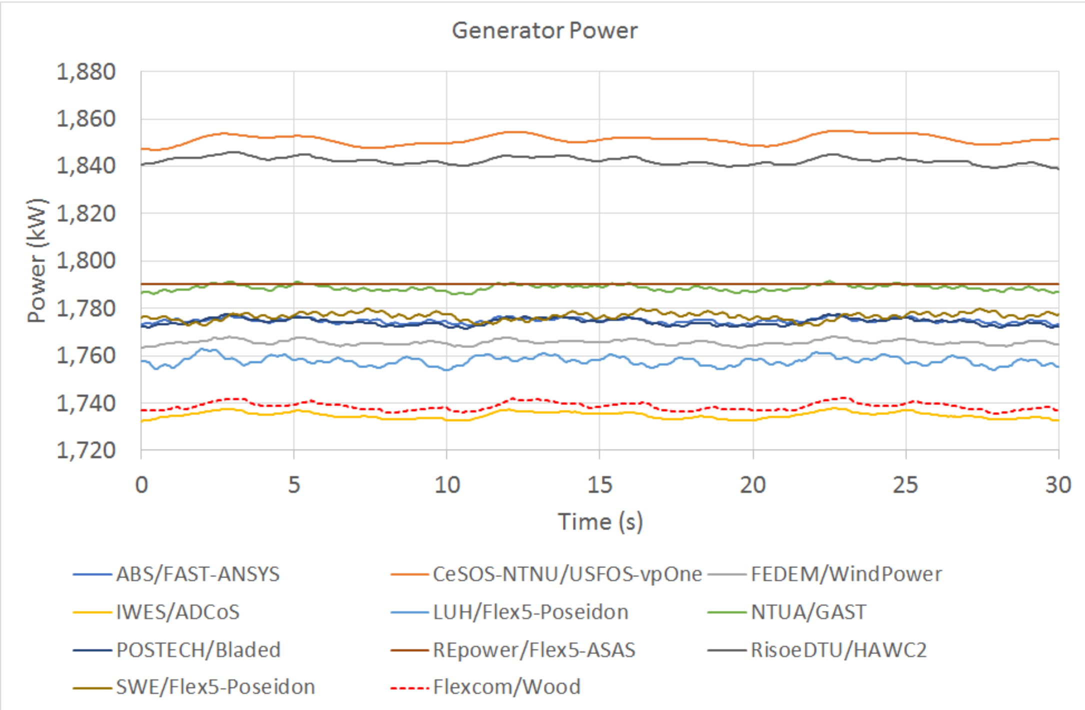


OC4 P1 LC5.6 Steady Wind Deans Wave

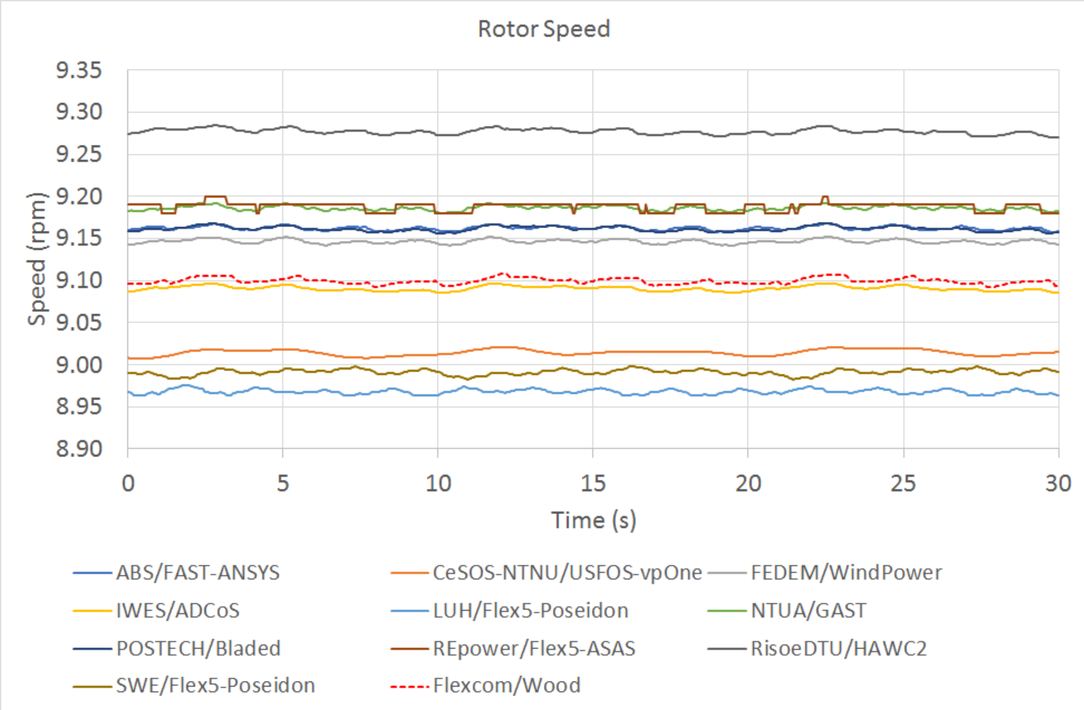
This load case combines elements from OC4 P1 LC2.2 (which considers a rigid offshore wind turbine subjected to steady wind loading) and OC4 P1 LC4.3b (which considers a flexible offshore structure subjected to Dean's stream wave loading). Results are naturally very similar to those presented for these simpler load cases. Although this load case is not discussed by Popko et al. (2012), a broad comparison is presented here, building on the results presented for the earlier load cases.

Good agreement is shown between all software tools for the mean generator power. The generator power predicted by Flexcom is slightly lower than most of the other software tools (apart from ADCoS which had an acknowledged miscalculation of the flow stagnation effect in front of the tower during OC4). Flexcom's mean power is 1738.6 kW, just 2.6% below the average (1785.6 kW). It is worth noting that there is a 6.5% variation across the other software tools also.

It is worth noting that the Flexcom results were produced using Flexcom 8.12.1 which is coupled with AeroDyn V15.02.04 (14-Apr-2016). The OC4 Phase I results were published in 2012 at which point the latest version of AeroDyn was Version 14. Improvements to the aerodynamic modelling software primarily account for the differences in generator power predicted by Flexcom and ABS/FAST-ANSYS. Variations in the aerodynamic theories underpinning each software product may also be a contributing factor for differences in generator power. Similar trends were observed for the OC4 semi-submersible and you are referred to OC4 P2 LC3.1 if you are interested in a more detailed discussion.

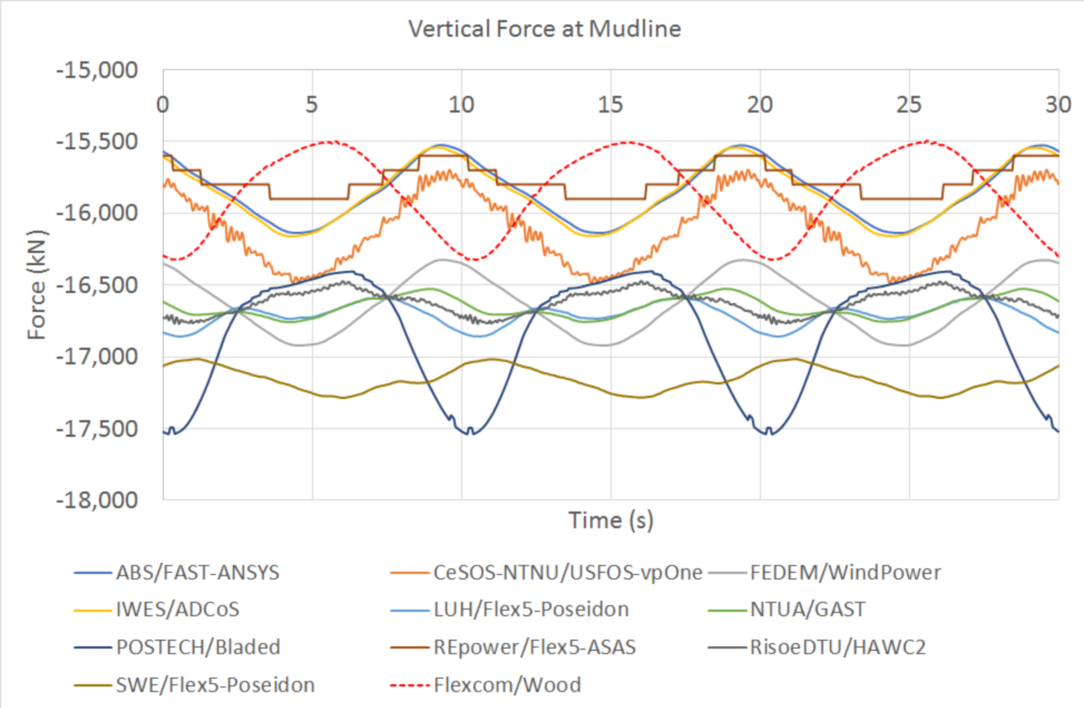


Close agreement is shown between all software tools for the rotor speed. Difference in mean rotor speed is less than 3.4%. Flexcom's results are very close to the median of OC4 data.

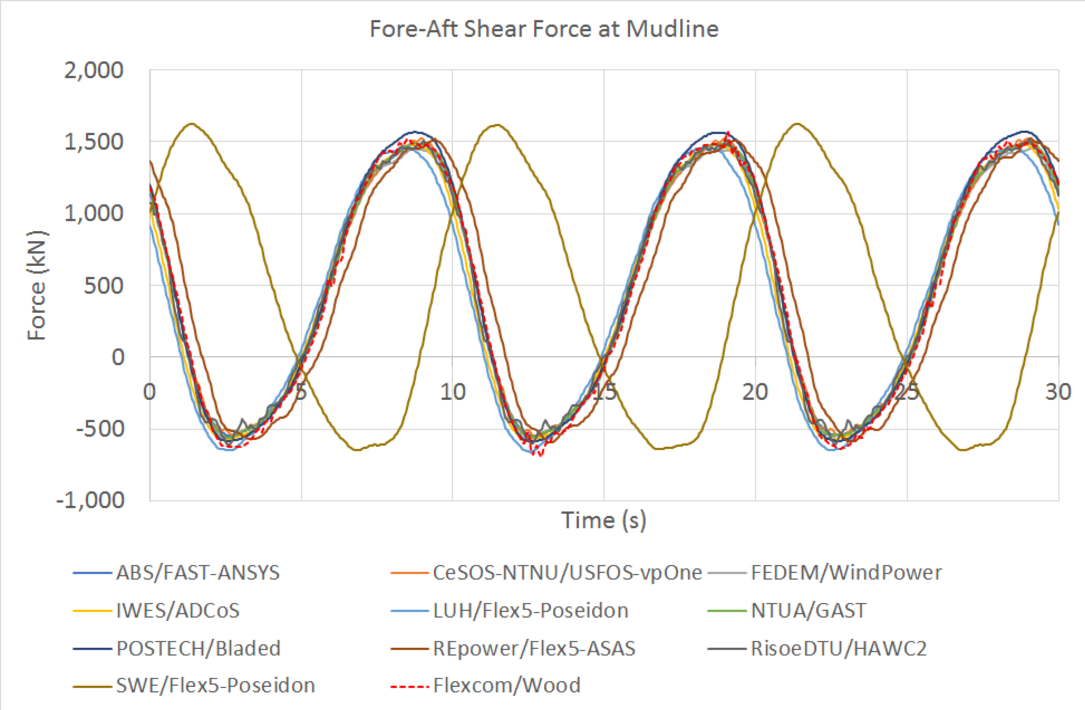


The low speed shaft torque shows good agreement between many of the software tools, with most codes showing peaks corresponding the blade passing (3P) frequency. These are not demonstrated by the Flexcom signal, given that a simplified RNA model is used in Flexcom 8.12.1. Instead the low-speed shaft torque is derived from the generator torque and the gearbox ratio, which leads to a smoother signal than most of the other software tools (note that the low-speed shaft torque is also calculated indirectly for the IWES/ADCoS results, which show close agreement with Flexcom).

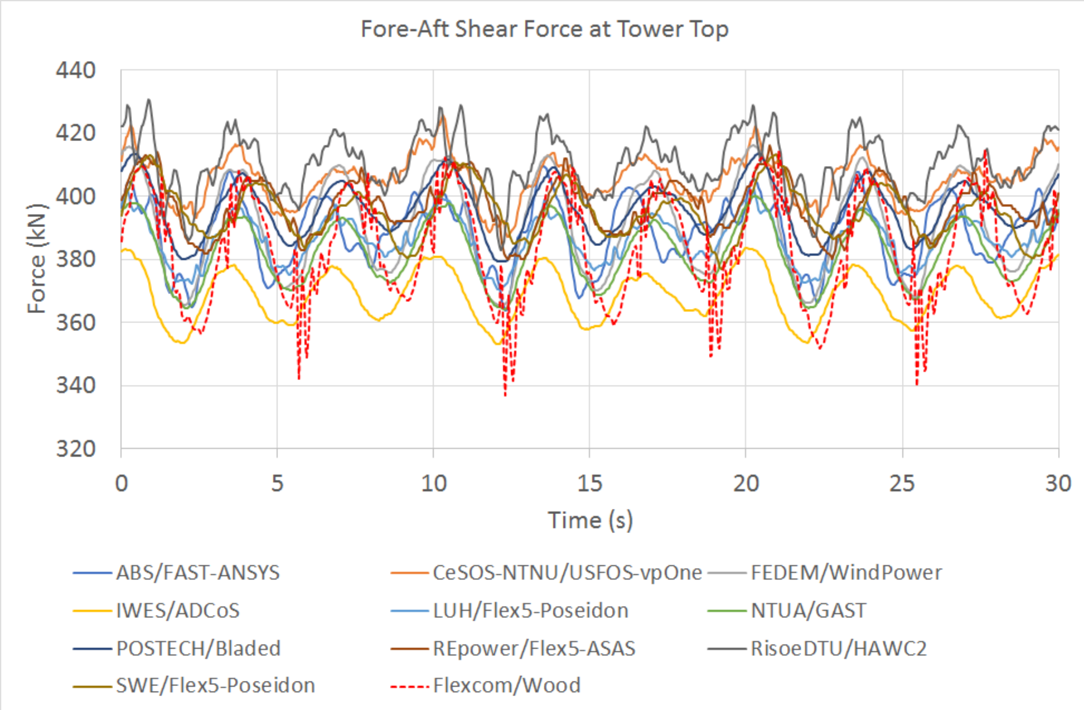
The vertical force at the mudline predicted by Flexcom shows good agreement with the other software tools. As outlined in OC4 P1 LC2.1, the project participants which omitted the buoyancy force at the mudline predict a mean vertical force at about -16600 kN, while others (including Flexcom) which included this term predict a mean vertical force of about -15800 kN. Differences in peak-to-peak amplitude of vertical force are mainly due to the buoyancy calculation method, whether displaced volume or pressure integration. Additionally, different masses for the jacket substructure and marine growth contributed to discrepancies.



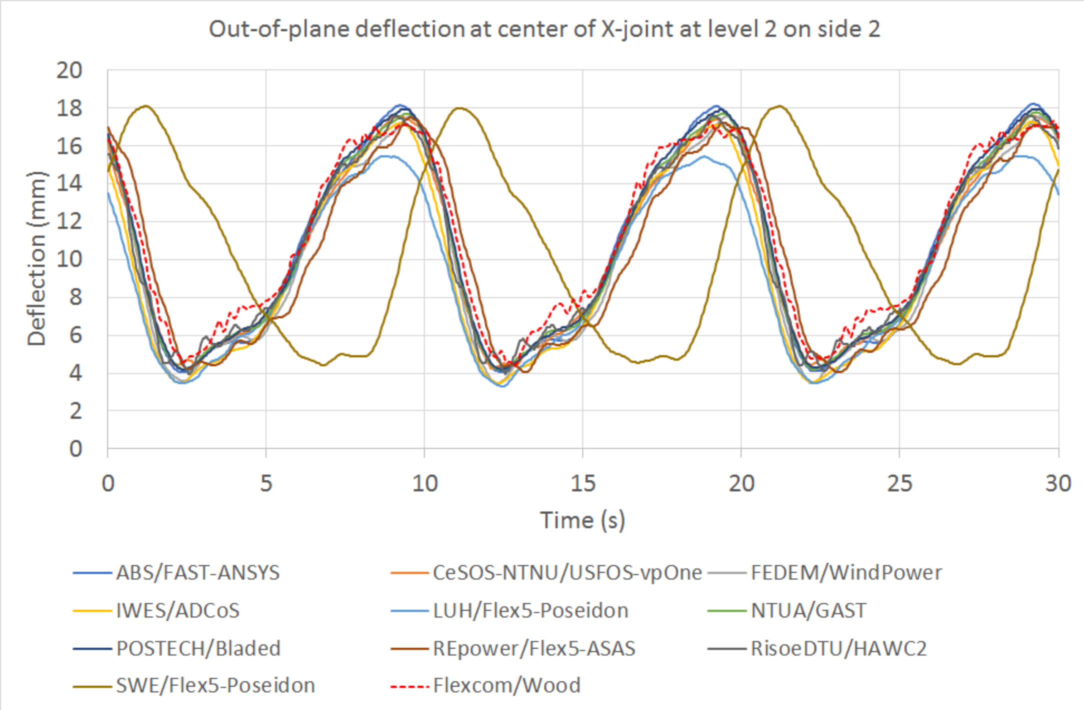
Very close agreement is shown between all software tools for the fore-aft shear force at the base of the jacket. Minor discrepancies in peak-to-peak amplitude may be caused by dissimilarities in implementations of wave kinematics, or slight differences originating in the modelling of the support structure. Results from SWE/Flex5-Poseidon appear out of phase with the other tools - a minor fault in the software at that time caused an unexpected phase shift in the output.



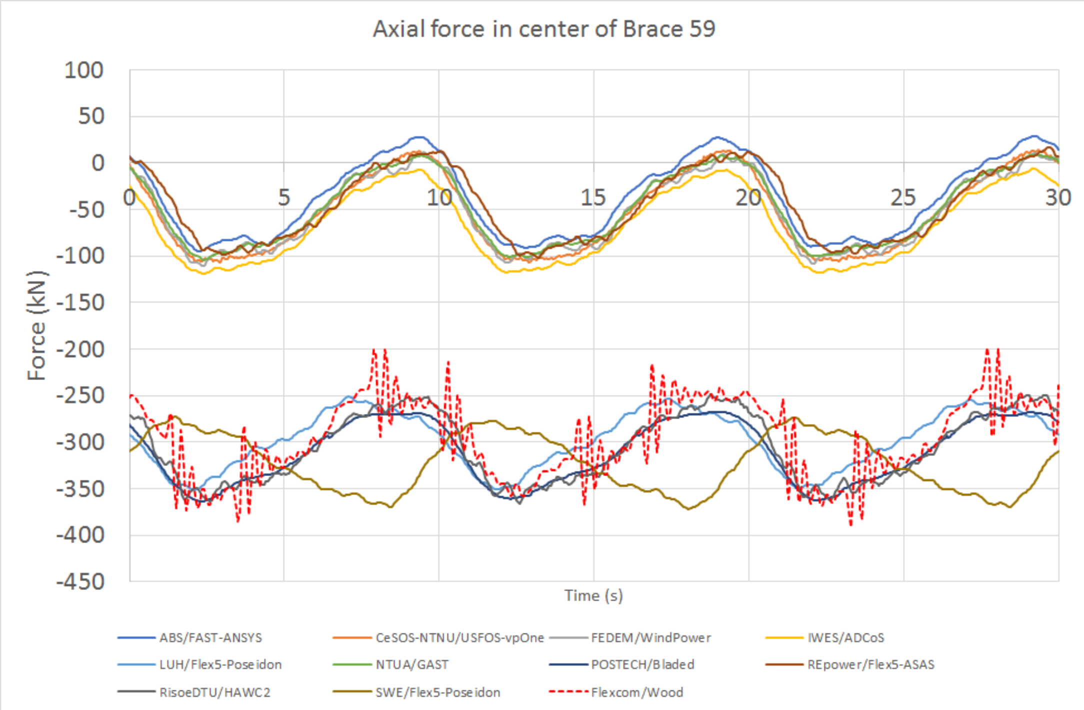
The fore-aft shear force at the top of the tower shows some variation across the different software products, but broadly speaking, good general agreement is observed. Tower shear is heavily influenced by wind rather than wave loading, so this plot shows similarities with the corresponding result from OC4 P1 LC2.2. It is interesting to note that the inclusion of wave loading, and the introduction of flexibility into the support structure, has negated the downward peaks at the blade passing (3P) frequency which were so clearly evident in OC4 P1 LC2.2. The Flexcom model displays some high frequency response at regular intervals corresponding to the 3P frequency. Further work is required to ascertain the precise source of this response; however it is believed to be related to the rigid blade assumption inherent in the RNA model in Flexcom 8.12.1. It is intended to verify this using the more advanced RNA model presently under development. Notwithstanding this, the Flexcom model matches the other tools for all of the key response parameters.



The out-of-plane displacement in the fore-aft direction, of the central joint of the cross-brace at level 2 on side 2 (see OC4 Jacket schematic for further details), shows a remarkably good agreement across the different software products, including Flexcom. Results from SWE/Flex5-Poseidon appear out of phase with the other tools - a minor fault in the software at that time caused an unexpected phase shift in the output.



The axial force in brace B59 at the lowest X-brace (see OC4 Jacket schematic for further details) shows some variation across the software products in terms of its mean value. However, variations about the mean show good agreement between the tools, including Flexcom. Differences in mean values may be largely attributed to differences in buoyancy modelling and discretisation of the brace (leading to a different measurement location). Like the tower-top shear force, Flexcom shows some higher frequency noise in the brace axial force also, so these issues are inter-related.



OC4 P1 LC5.7 Turbulent Wind Irregular Wave

The final load case in OC4 Phase I is the most complex, applying both turbulent wind and irregular wave loading to the fully flexible offshore wind turbine. The mean wind speed is 18m/s, and the turbulence description is based on the Kaimal spectrum and the normal turbulence model (NTM). The turbulent wind field definition is supplied by Risø DTU (Technical University of Denmark) in the form of HAWC2 format binary files. The ambient seastate is described by a Pierson-Moskowitz wave spectrum with Hs=6m and Tp=10s.

Popko et al. (2012) present results graphically in terms of probability density functions (PDF), power spectral densities (PSD) or damage-equivalent loads (DEL), for load cases with stochastic inputs such as this one. For conciseness here, a range of results are neatly summarised in tabular format instead. For each output parameter, the table presents:

- Mean value of the parameter predicted by Flexcom (e.g. mean generator power)
- Standard deviation of the parameter predicted by Flexcom (e.g. to quantify power fluctuation)

- Percentage difference between Flexcom's mean value and the average mean of the other modelling tools
- Percentage difference between Flexcom's standard deviation value and the average deviation of the other modelling tools

Wind speed and wave elevation are included in order to check consistency of inputs between Flexcom and the other models. The mean wind speed and the deviation in wind speed (which is a measure of the turbulence) are both consistent between all modelling tools. Likewise, the statistics of wave elevation are also consistent.

The mean values of generator power, rotor speed and lower speed shaft torque estimated by Flexcom all show very close agreement with the other modelling tools. However, Flexcom is underpredicting the deviations about the mean for each of these three parameters. These trends are consistent with OC4 P1 LC5.6, a more straightforward load case which considered steady wind and regular wave loading only. For reasons discussed previously, the simplified RNA model used in Flexcom 8.12.1 is unable to properly capture the characteristic fluctuations in loading and response at the 3P frequency. It is planned to address these limitations using a more advanced RNA model in the next version of the software.

	Flexcom Result		**Percentage difference from OC4 average**	
Parameter	**Mean**	**Deviation**	**Mean**	**Deviation**
Wind Speed (m/s)	18.0	2.2	-0.2%	1.2%
Wave Height (m)	0.0	1.5	N/A	-1.0%
Generator Power (kW)	5000.0	17.5	-2.1%	-75.1%
Rotor Speed (rpm)	12.1	0.2	0.0%	-50.0%
Low Speed Shaft Torque (kNm)	4180.6	54.2	-2.3%	-82.7%

	Flexcom Result		Percentage difference from OC4 average	
Parameter	**Mean**	**Deviation**	**Mean**	**Deviation**
Fore-Aft Shear Force at Tower Top (kN)	377.6	68.3	5.6%	-14.9%
Fore-Aft Shear Force at Mudline (kN)	423.7	394.7	14.3%	-14.5%
Out-of-Plane Deflection at Joint S2-L2 (mm)	11.3	3.4	38.7%	-8.9%

1.10.12.4 L04 - UMaine VolturnUS-S IEA15MW

The University of Maine (UMaine) VolturnUS-S reference semi-submersible (Allen et al., 2020) is a generic steel version of the UMaine patented concrete floating foundation technology developed in collaboration with the U.S. Department of Energy (Viselli et al., 2016). According to UMaine, "*the reference design aims to serve the wind industry by providing a publicly available design benchmark to explore new technologies and design methodologies while facilitating collaboration between industry and researchers*".

The overall layout of this example is as follows:

- Introduction gives an overview of the UMaine semisub, the IEA 15MW turbine and ROSCO controller, and the load cases simulated.
- Model Summary describes the Flexcom model in detail.
- Results from Flexcom are presented, alongside results from the UMaine report where possible.

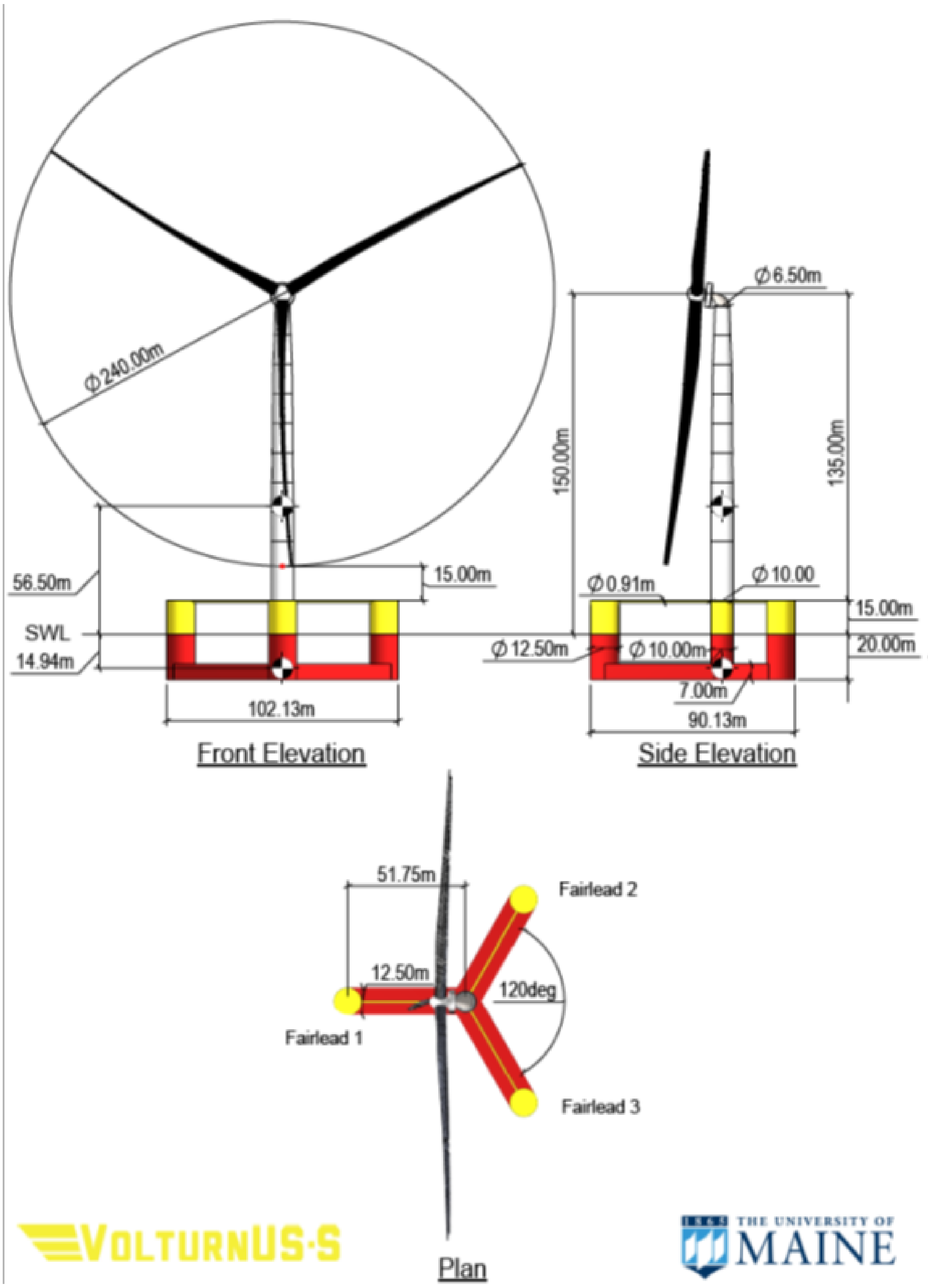
Introduction

SEMI-SUBMERSIBLE PLATFORM

The University of Maine (UMaine) VolturnUS-S reference semi-submersible (Allen et al., 2020) is a generic steel version of the UMaine patented concrete floating foundation technology developed in collaboration with the U.S. Department of Energy (Viselli et al., 2016). According to UMaine, "*the reference design aims to serve the wind industry by providing a publicly available design benchmark to explore new technologies and design methodologies while facilitating collaboration between industry and researchers*".

The hull comprises three 12.5m-diameter buoyant columns radially spaced with centres that are 51.75m from the tower's vertical axis. The platform-tower interface is atop a fourth buoyant column located at the centre of the platform in the surge-sway plane. This central column is connected to the outer columns via three 12.5m-wide-by-7.0m-high rectangular bottom pontoons and three 0.9m-diameter radial struts attached to the bottom and top of the buoyant columns, respectively. When on station, the total mass of the platform is 17,854t, of which 3,914t is structural steel, 2,540t is fixed iron-ore-concrete ballast, divided equally and placed at the base of the three radial columns, 11,300t is seawater ballast that floods the majority of the three submerged pontoons, and a 100t tower interface connection

The mooring system consists of three 850m-long chain catenary lines. Each line is connected at the fairlead to one of the platform's three outer columns at a depth of 14m below the SWL. The lines span radially to anchors spaced equally at 120 degrees in the surge-sway plane, which are located at a depth of 200m and spaced radially 837.6m from the tower's centreline. All lines use a studless R3 chain with a nominal (bar) diameter of 185mm.



Schematic of UMaine semisub (Allen et al., 2020)

A summary of the semi-sub's main characteristics is presented here, and further details are available in Allen et al., 2020.

Parameter	Value	Parameter	Value
Column length	35m	Platform mass	17,854 t
Central column diameter	10m	Hull steel mass	3,914 t
Offset column diameter	12.5m	Ballast mass (Fixed, Fluid)	2,540t, 11,300t
Radial spacing to offset columns	51.75m	Tower interface mass	100t
Pontoon width	12.5m	Roll inertia	1.251E+10 kg.m2
Pontoon height	7.0m	Pitch inertia	1.251E+10 kg.m2
Freeboard	15m	Yaw inertia	2.367E+10 kg.m2
Draft	20m		
Overall size (length, width, height)	88.0m, 103.0m, 35.0m		
Strut diameter	0.9m		
CoG (from SWL)	-14.94m		
CoB (from SWL)	-13.63m		

WIND TURBINE GENERATOR

The IEA Wind 15 MW wind turbine was jointly developed by the National Renewable Energy Laboratory (NREL) and the Technical University of Denmark (DTU). This reference wind turbine is a Class IB direct-drive machine, with a rotor diameter of 240m and a hub height of 150m. Full details of the turbine, supported by a fixed-bottom monopile, are presented by Gaertner et al. 2020, and a summary of key parameters are presented here. The wind turbine uses a direct-drive layout with a permanent-magnet, synchronous, radial flux outer-rotor generator in a simple and compact nacelle layout.

Parameter	Value	Parameter	Value
Power rating	15 MW	Maximum tip speed	95 m/s
Turbine class	IEC Class 1B	Rotor diameter	240 m
Specific rating	332 W/m2	Airfoil series	FFA-W3
Rotor orientation	Upwind	Hub height	150 m
Number of blades	3	Hub diameter	7.94 m
Control	Variable speed Collective pitch	Hub overhang	11.35 m
Cut-in wind speed	3 m/s	Rotor precone angle	-4.0 deg
Rated wind speed	10.59 m/s	Blade prebend	4 deg
Cut-out wind speed	25 m/s	Blade mass	65 t
Design tip-speed ratio (TSR)	9.0	Drivetrain	Direct drive
Minimum rotor speed	5.0 rpm	Shaft tilt angle	6 deg

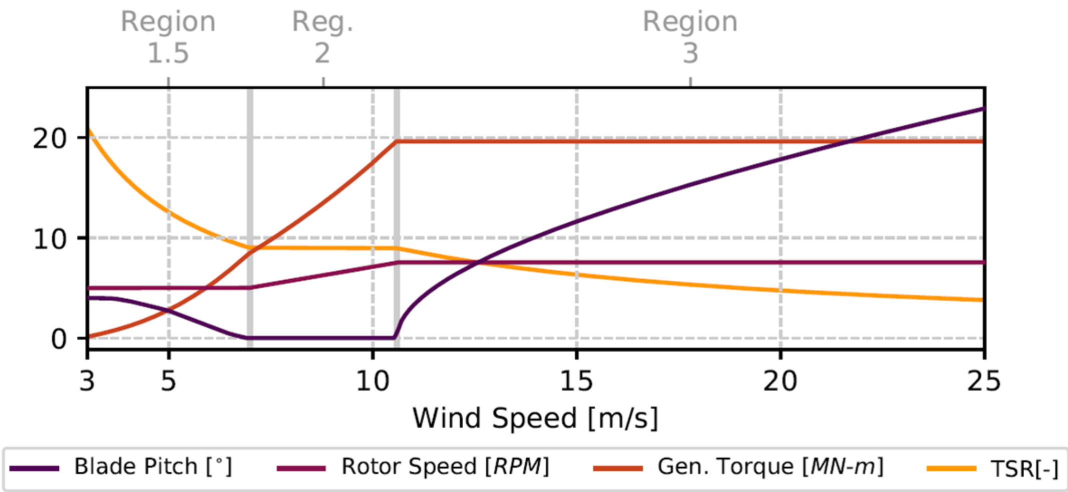
Parameter	Value	Parameter	Value
Maximum rotor speed	7.56 rpm	Rotor nacelle assembly (RNA) mass	1,017 t

CONTROLLER

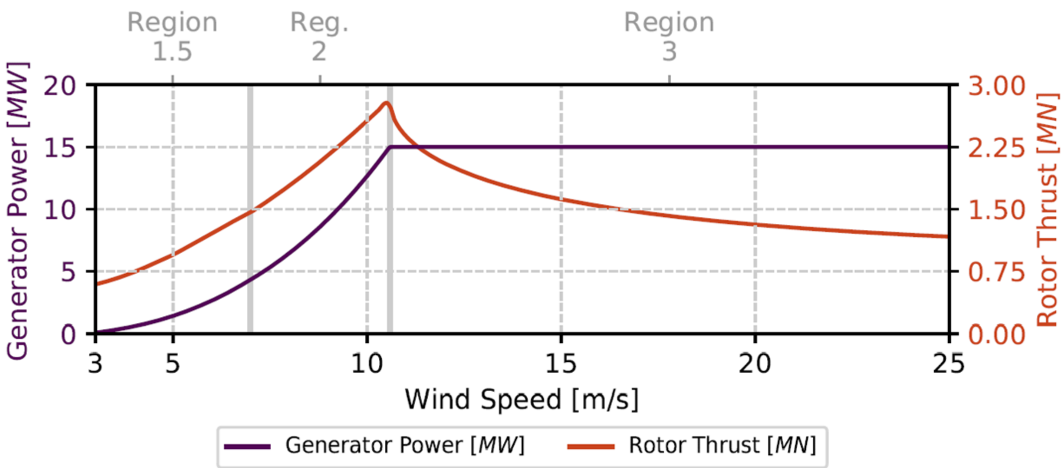
The NREL Reference OpenSource Controller (ROSCO) is used to control the turbine. Two active proportional integral (PI) controllers are implemented for the generator torque and blade pitch angles. The controller operation can be distinguished by three regions:

- 3 m/s <= U <= 6.98 m/s; minimum rotor speed. A PI controller on the generator torque is used to regulate the turbine to the turbine's minimum rotor speed, 5 rpm. The minimum blade pitch angle is defined based on a wind speed estimate, such that CP is maximized.
- 6.98 m/s <= U <= 10.59 m/s; optimal TSR. In below-rated wind speeds, the rotor speed is regulated to operate at the turbine's optimal TSR with a PI controller on the generator torque.
- 10.59 m/s <= U <= 25 m/s; rated power. In above-rated wind speeds, the rotor speed is regulated to its rated value, 7.55 rpm, via a PI controller on the blade pitch angle.

The steady-state performance of the rotor as a function of wind speed is shown below, along with the power and thrust curves for the turbine. In Region 1.5, minimum rotor speed constraints result in significantly higher, suboptimal tip-speed ratios. The blade pitch controller imposes a minimum pitch constraint based on the wind speed estimate, so that CP is maximized. This results in positive blade pitch angles of up to 4° at low wind speeds. Concurrently, the generator torque controller attempts to set the rotor to the minimum rotor speed. In Region 2, the torque controller tracks the set point tip-speed ratio, which is set near or at maximum CP. The design point for this blade is TSR = 9.0 and blade pitch = 0°. Finally, in Region 3, the torque controller is saturated at rated torque and the blade pitch PI controller pitches to feather to maintain rated rotor speed.



Controller Regulation (Gaertner et al. 2020)



Power & Thrust Curves (Gaertner et al. 2020)

DESIGN LOAD CASES

A subset of IEC design load case conditions was selected by UMaine to gauge the performance of the reference semisub in representative normal and extreme design conditions (based on industry standard IEC 61400-1, 2020). According to UMaine, the simulations were selected to represent governing conditions of various critical components of the floating offshore wind turbine based on design experience of similar systems. All conditions considered an aligned wind and wave heading of 0 degrees (head seas). The environmental conditions associated with the design cases are representative of the U.S. East Coast.

The design load cases considered may be summarised as follows:

DLC	Turbine	Wind	Wave
1.1	Power production	Normal Turbulence Model (NTM) Vin < Vhub < Vout	Normal Sea State Hs = E[Hs|Vhub]
1.3	Power production	Extreme Turbulence Model (ETM) Vin < Vhub < Vout	Normal Sea State Hs = E[Hs|Vhub]
1.4	Power production	Extreme Coherent Gust with Direction Change (ECD) Vhub = Vr–2m/s, Vr, Vr+2m/s	Normal Sea State Hs = E[Hs|Vhub]
1.6	Power production	Normal Turbulence Model (NTM) Vin < Vhub < Vout	Severe Sea State Hs = Hs,SSS
6.1	Parked	Extreme Wind Model Vhub = V10min,50-yr	Extreme Sea State Hs = Hs,50-yr
6.3	Parked	Extreme Wind Model Vhub = V10min,1-yr	Extreme Sea State Hs = Hs,1-yr

Design load cases 1.1, 1.3 & 1.6 are reproduced in Example L04, but you are free to simulate other cases in your own time. In order to achieve this, you should examine the load case spreadsheet (L4-DLCs.xlsx) and expand this appropriately. You can then modify the *EXCEL VARIATIONS keyword in the master template files (L4-DLCs.keyxm). This will allow you to produce all the required keyword files. Note also that for efficiency reasons, this example considers only one value of wind and wave seed while the UMaine simulations typically consider 6 seeds, so again you will need to expand the spreadsheet if you wish to overcome this limitation.

Further details regarding the construction of the design load cases are provided by Allen et al., 2020 and reproduced here.

DLC	Wind Condi-tion	Hub Height Wind Speed (m/s)	Wind Headings (°)	Significant Wave Height (m)	Peak Period (s)	Gamma Shape Factor (-)	Wave Headings (°)	Settings	# of Seeds	Total # of Sims.
1.1	NTM	4.00	0.00	1.10	8.52	1.00	0.00	-	6	6
		6.00	0.00	1.18	8.31	1.00	0.00	-	6	6
		8.00	0.00	1.32	8.01	1.00	0.00	-	6	6
		10.00	0.00	1.54	7.65	1.00	0.00	-	6	6
		12.00	0.00	1.84	7.44	1.00	0.00	-	6	6
		14.00	0.00	2.19	7.46	1.00	0.00	-	6	6
		16.00	0.00	2.60	7.64	1.35	0.00	-	6	6
		18.00	0.00	3.06	8.05	1.59	0.00	-	6	6
		20.00	0.00	3.62	8.52	1.82	0.00	-	6	6
		22.00	0.00	4.03	8.99	1.82	0.00	-	6	6
		24.00	0.00	4.52	9.45	1.89	0.00	-	6	6
1.3	ETM	4.00	0.00	1.10	8.52	1.00	0.00	-	6	6
		6.00	0.00	1.18	8.31	1.00	0.00	-	6	6
		8.00	0.00	1.32	8.01	1.00	0.00	-	6	6
		10.00	0.00	1.54	7.65	1.00	0.00	-	6	6
		12.00	0.00	1.84	7.44	1.00	0.00	-	6	6
		14.00	0.00	2.19	7.46	1.00	0.00	-	6	6
		16.00	0.00	2.60	7.64	1.35	0.00	-	6	6
		18.00	0.00	3.06	8.05	1.59	0.00	-	6	6
		20.00	0.00	3.62	8.52	1.82	0.00	-	6	6
		22.00	0.00	4.03	8.99	1.82	0.00	-	6	6
		24.00	0.00	4.52	9.45	1.89	0.00	-	6	6
1.4	ECD+/-R-2.0	8.00	0.00	1.32	8.01	1.00	0.00	+/- Dir. Change	1	2
	ECD+/-R	10.00	0.00	1.54	7.65	1.00	0.00	+/- Dir. Change	1	2
	ECD+/-R+2.0	12.00	0.00	1.84	7.44	1.00	0.00	+/- Dir. Change	1	2
1.6	NTM	4.00	0.00	6.30	11.50	2.75	0.00	-	6	6
		6.00	0.00	8.00	12.70	2.75	0.00	-	6	6
		8.00	0.00	8.00	12.70	2.75	0.00	-	6	6
		10.00	0.00	8.10	12.80	2.75	0.00	-	6	6
		12.00	0.00	8.50	13.10	2.75	0.00	-	6	6
		14.00	0.00	8.50	13.10	2.75	0.00	-	6	6
		16.00	0.00	9.80	14.10	2.75	0.00	-	6	6
		18.00	0.00	9.80	14.10	2.75	0.00	-	6	6
		20.00	0.00	9.80	14.10	2.75	0.00	-	6	6
		22.00	0.00	9.80	14.10	2.75	0.00	-	6	6
		24.00	0.00	9.80	14.10	2.75	0.00	-	6	6
6.1	EWM 50 yr	47.50	0.00	10.70	14.20	2.75	0.00	Yaw +/- 8°	6	12
6.3	EWM 1 yr	38.00	0.00	6.98	11.70	2.75	0.00	Yaw +/- 20°	6	12

NTM	normal turbulence model
ETM	extreme turbulence model
ECD	extreme coherent gust with direction change
EWM	extreme wind speed model

IEC Design Load Case Matrix (Allen et al., 2020)

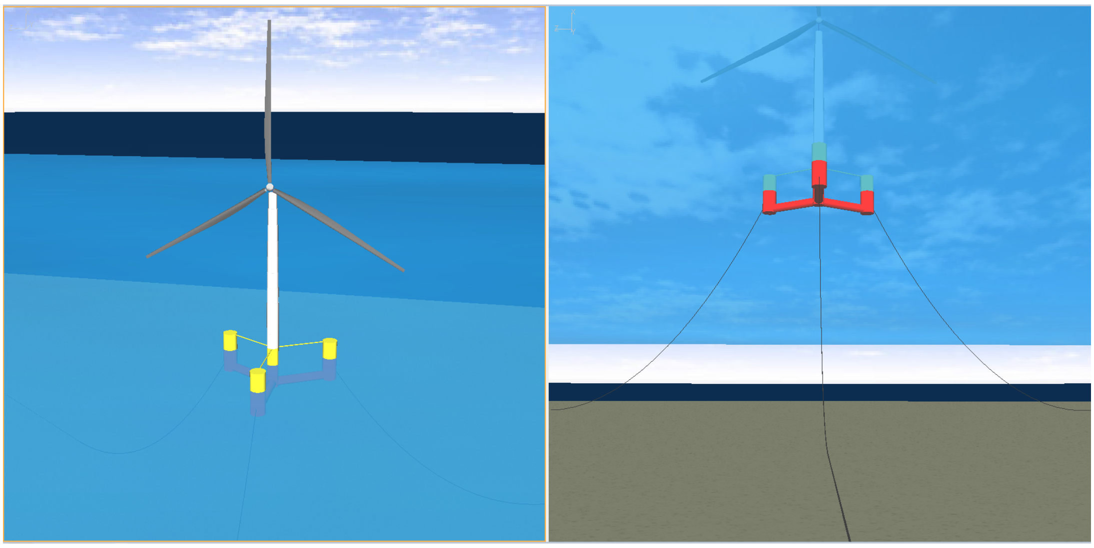
WIND INPUT FILES

The required wind data files for Example L04 are not shipped with Flexcom. The binary files tend to be very large, so it is not practical to include these in the installation package. Instead, you can readily generate the files yourself with the Wind Field Generator app. You will find a sample project file for the app in the Data->Wind subfolder of the project, it is called L4-DLCs-Wind.wfproj. It is already pre-populated with the relevant wind speeds and turbulence models etc., so all you need to do is select the *File->Generate Input Files and Run TurbSim* command.

Model Summary

INTRODUCTION

The Flexcom model of the UMaine semi-sub is shown below.



UMaine semisub modelled in Flexcom

FLOATING PLATFORM

The floating platform is modelled using a series of discrete Lines to represent the 3 columns, pontoons and struts. These lines are connected up at appropriate points using a range of Equivalent Nodes to form a single coherent structure.

Strictly speaking it is not necessary to model the floating platform in such detail. A skeleton model which just includes key points of interest (such as the centres of gravity and buoyancy) would facilitate the application of concentrated loads. However, there are some advantages associated with creating a more detailed model - refer to Floating Body Modelling Detail if you are interested in further details.

Each line is assigned rigid Stiffness terms as the platform is assumed to act as a rigid body. All lines are assigned a Mass per Unit Length of zero (as the total mass is concentrated at the centre of mass) and a Buoyancy Diameter of zero (as the total buoyancy is concentrated at the centre of buoyancy). Each line is assigned a physical Drag Diameter, which allows Morison drag loads to be added to the standard radiation-diffraction excitation forces.

A Floating Body is defined which models the hydrodynamic characteristics of the floater. Hydrostatic Stiffness terms are used to simulate restoring forces and moments due to buoyancy. Added Mass, Radiation Damping and Force RAO coefficients are defined for the floating body over a range of discrete frequencies - these terms enable the computation of incident, diffracted and radiated (linear) wave potentials to be simulated. Note that these inputs are derived separately from a radiation-diffraction analysis. Relevant Rotational Inertia terms are specified at the floating body centre of gravity, while the body's mass is represented by a Point Mass.

TOWER

The tower is constructed using a single Line, with its lower end attached to the floating platform using an Equivalent Node. As the tower is tapered from base to top, it is modelled using several Line Sections of decreasing diameter. The finite element mesh density assigned to the tower ensures that each section is modelled using a single element. Note that a consistent mesh density is assigned to the aerodynamic model via the `*TOWER INFLUENCE` keyword. Realistic Stiffness and Mass per Unit Length terms are individually assigned to each tower section based on the material properties for steel and the relevant diameter and wall thickness at the section's mid-point. As the tower will not experience any hydrodynamic loading, the line is assigned zero values of Buoyancy Diameter and Drag Diameter.

TURBINE ASSEMBLY

Finite element Nodes are explicitly created at the centres of mass of the nacelle, hub and blades (centre of mass at initial position) respectively. These nodes are then connected to the top of the tower using finite Elements. The use of explicitly created nodes and elements is more convenient than using lines in such circumstances.

Point Mass terms are used to position appropriate masses at the nodes, hence the elements have zero values of Mass per Unit Length. All elements are assigned large Stiffness terms to simulate rigidity. As the turbine assembly will not experience any hydrodynamic loading, all elements are assigned zero values of Buoyancy Diameter and Drag Diameter.

An Auxiliary Profile is used to represent the rotating blades. While this has no structural function, it enhances the visual appeal of the model, and assists in the understanding of rotor and platform motions post-simulation.

AERODYNAMICS

All inputs which are required by AeroDyn to compute the aerodynamic loading on the blades and tower are are logically grouped together under the `$AERODYN` section, and specified in the dynamic simulation file. Fundamental inputs include Blade Geometries, Aerofoil Coefficients, miscellaneous Turbine Inputs (such as hub height, hub radius, overhang, shaft tilt, blade precone etc.) and Tower Influence (i.e. tower drag). These inputs should be intuitively familiar to engineers with some wind turbine modelling experience and you are referred to the keyword documentation should you require further information regarding the significance of any particular input.

CONTROL SYSTEM

The wind turbine control system is defined via the `*SERVODYN` keyword, which references the standard control DLL (ROSCO) provided by NREL for the UMaine semi-submersible. At low wind speeds the turbine is operating below rated power, so the rotor is allowed to rotate freely without any control in order to maximise power extraction. At intermediate wind speeds the turbine is fully operational, and the generator torque is used to control rotor speed while maximising generated power. At higher wind speeds the available wind power is above the rated power of the turbine, so blade pitch control is used to feather the blades and shed excess power.

MOORING LINES

The mooring lines are created using 3 separate Lines. The upper end of each mooring line is attached to the relevant fairlead node on the floating platform using the Equivalent Nodes facility, while the lower ends are constrained using Fixed Boundary Conditions. The lines are modelled using the Truss Element feature, which is ideally suited to modelling chains and wires. Realistic Stiffness, Mass per Unit Length, Buoyancy Diameter and Drag Diameter terms are assigned to the lines.

Results

The University of Maine report presents a variety of results, a subset of which were reproduced using the Flexcom model.

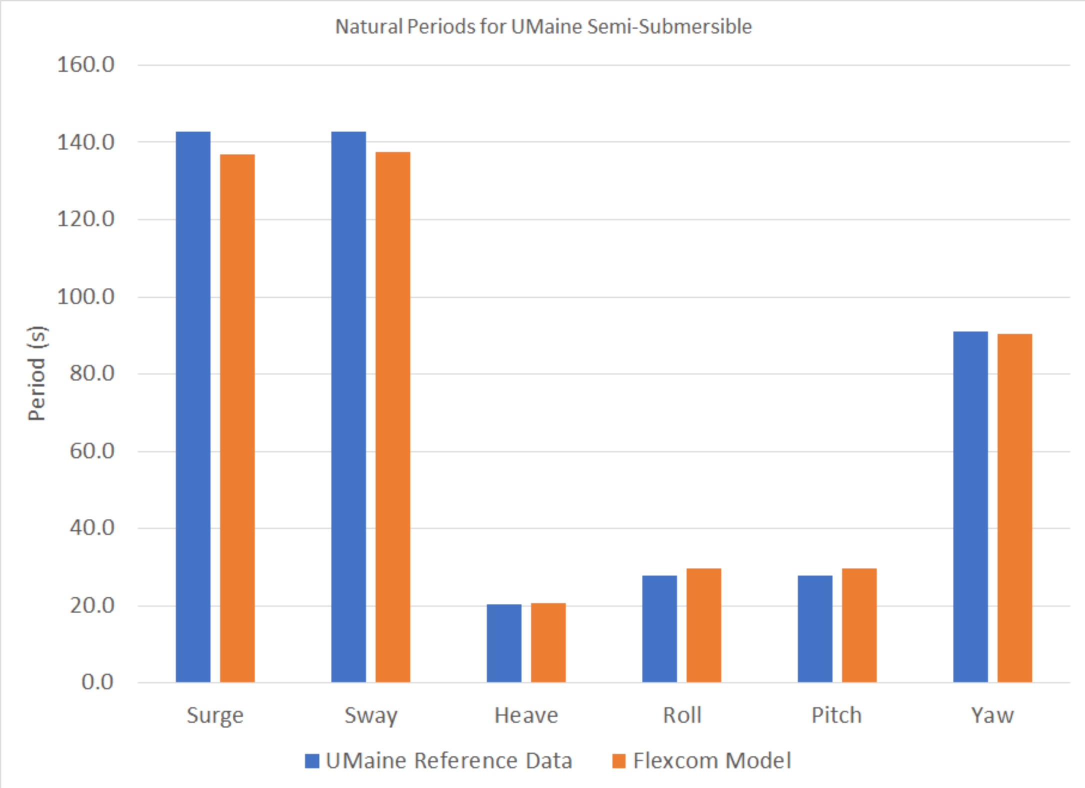
- Static Surge-Sway Offsets - to examine fairlead and anchor tension values resulting from static offsets of the system in the surge-sway plane.
- Free Decay Tests - to examine natural response frequencies of the system in each of the 6 degrees of freedom.

- Wave-induced motion RAOs - to examine the system response to wave excitation only.
- Design Load Cases - to gauge the performance of the reference semisub in normal and extreme design conditions which are representative of the U.S. East Coast.

Free Decay Tests

UMaine conducted free-decay simulations in OpenFAST for each of the rigid-body DOFs. The simulations considered a still wind environment and though aerodynamic drag was considered, its effects were minimised by orienting the blades such that they produced minimal drag. The decay simulations considered all DOFs pertaining to the platform, tower, and blades. A similar procedure was adopted using Flexcom, with the exception that the aerodynamic effects were switched off completely, by omitting the coupling between Flexcom and OpenFAST. The natural frequencies of the rigid-body modes of the system were calculated by taking the average period of oscillation over several decay periods. Results are presented in tabular and graphical form below. Close agreement is observed between the UMaine published report and the Flexcom model as expected.

Degree of Freedom	Frequency: UMaine	Frequency: Flexcom	Period: UMaine	Period: Flexcom
Surge	0.007 Hz	0.007 Hz	142.9 s	137.0 s
Sway	0.007 Hz	0.007 Hz	142.9 s	137.7 s
Heave	0.049 Hz	0.048 Hz	20.4 s	20.6 s
Roll	0.036 Hz	0.034 Hz	27.8 s	29.5 s
Pitch	0.036 Hz	0.034 Hz	27.8 s	29.5 s
Yaw	0.011 Hz	0.011 Hz	90.9 s	90.3 s

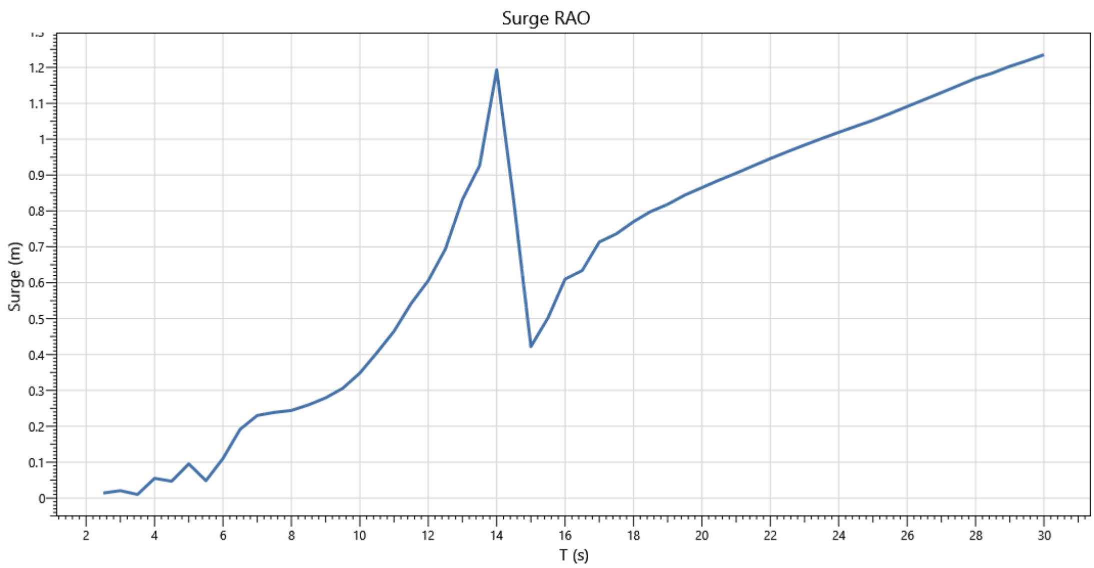


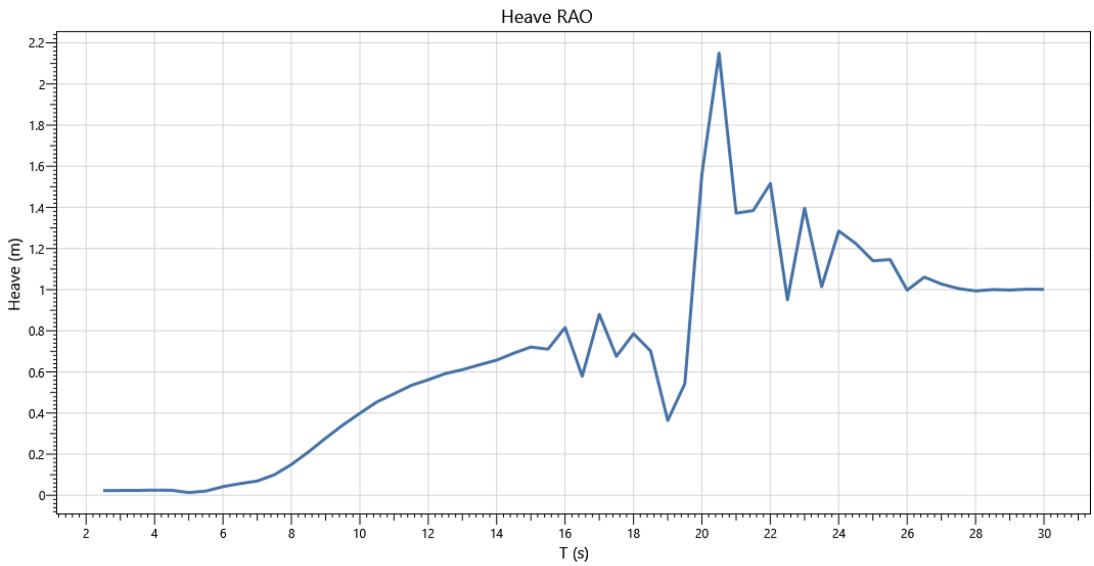
Wave Induced Motion RAOs

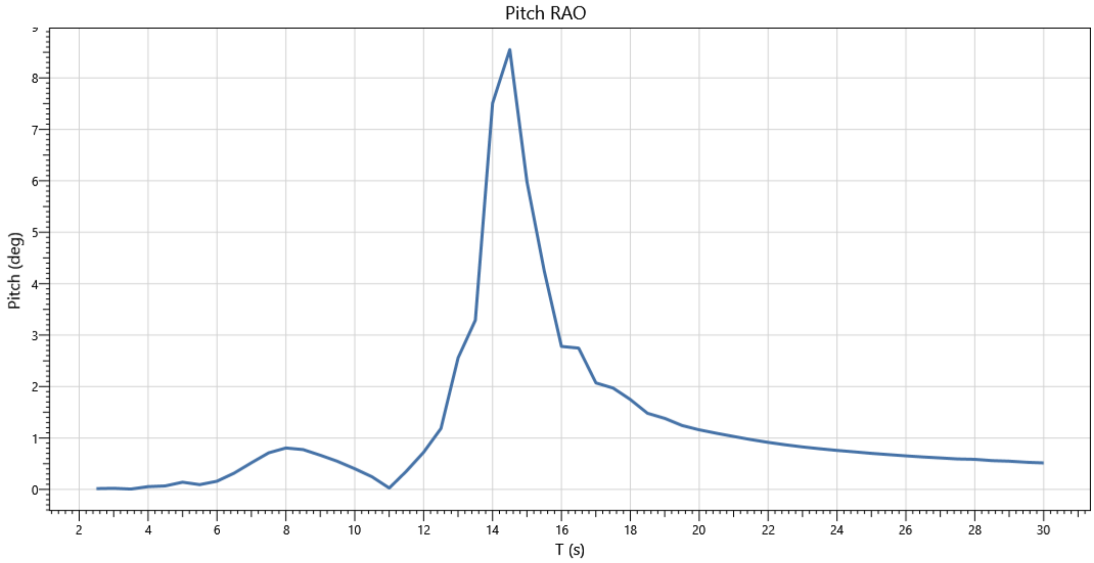
Wave induced motion RAOs are computed using a series of regular wave simulations at various periods, in this case going from 2.5s to 30.0s in 0.5s increments. The AeroDyn module is disabled so that the effects of aerodynamic loading are not considered. The assessment focuses one heading only (head seas), so only the RAOs relating to surge, heave, and pitch are relevant.

The published report on the UMaine semisub presents RAOs in graph form only. The underlying data is not supplied which makes comparisons between UMaine and Flexcom difficult. Hence only the Flexcom results are presented here.

The Flexcom results are based on a free-floating body only (RNA, tower and mooring system are excluded from the model), so the RAOs tend to exhibit peaks at the natural response periods, and can appear undamped in general.


Surge RAO
Surge (m)
1.2
1.1
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
T (s)


Heave RAO
Heave (m)
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
T (s)



Design Load Cases

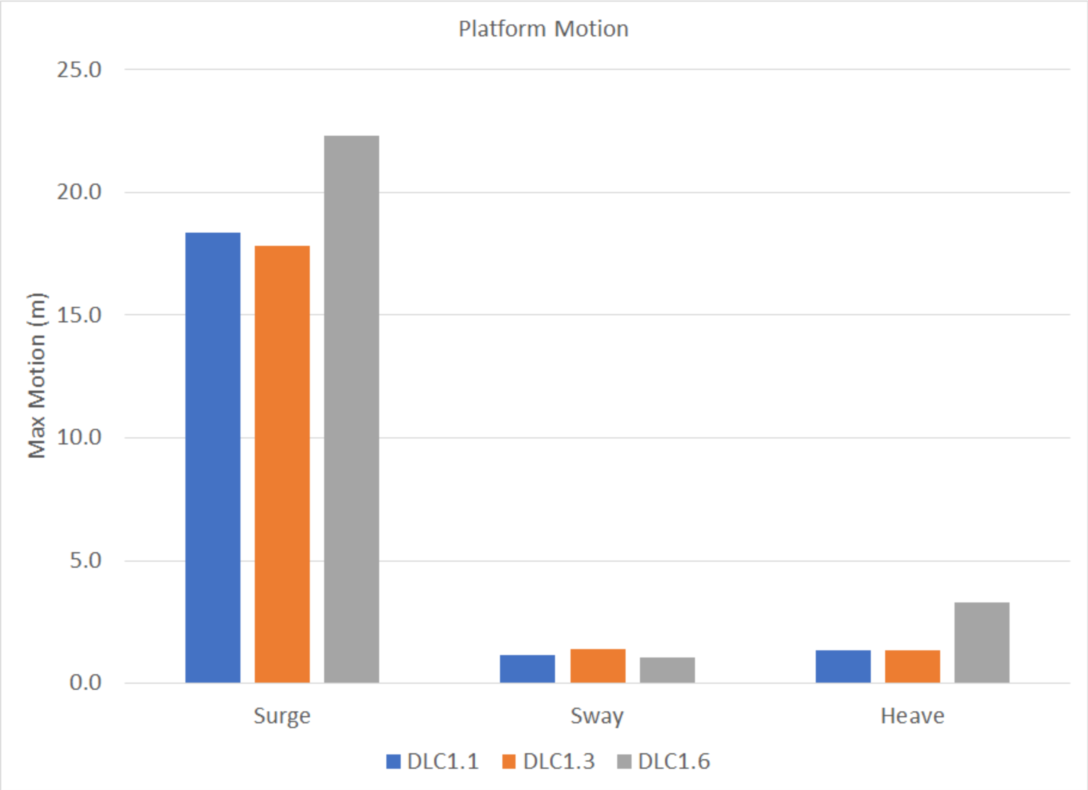
A subset of IEC design load case conditions was selected by UMaine to gauge the performance of the reference semisub in normal and extreme design conditions which are representative of the U.S. East Coast.

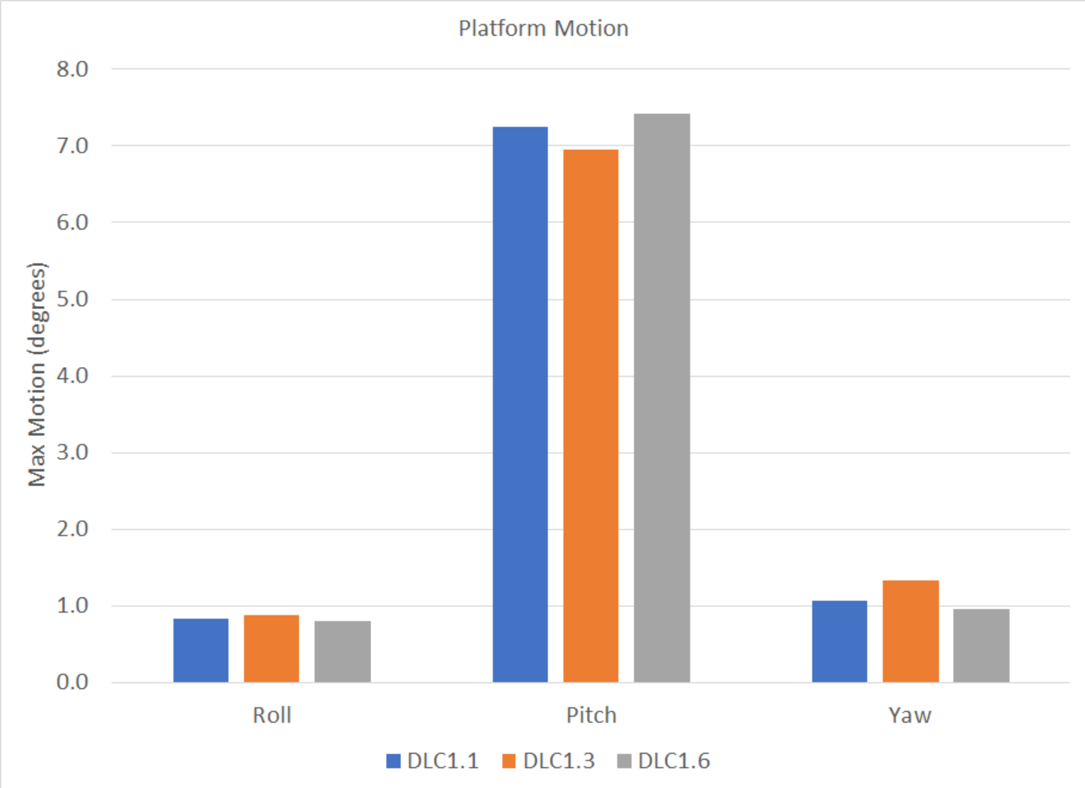
The floating offshore wind turbine’s performance under these design conditons is summarised via statistics for various outputs, including:

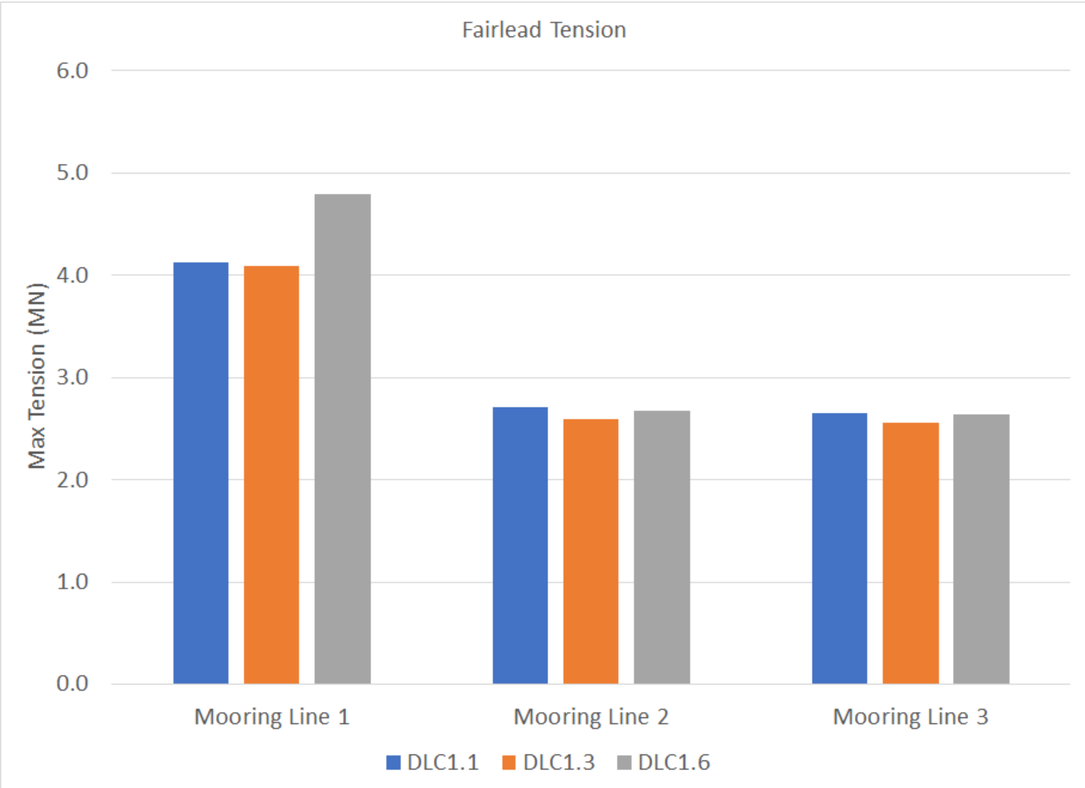
- Platform motion in 6 degrees of freedom
- Mooring line tension
- Tower base moment

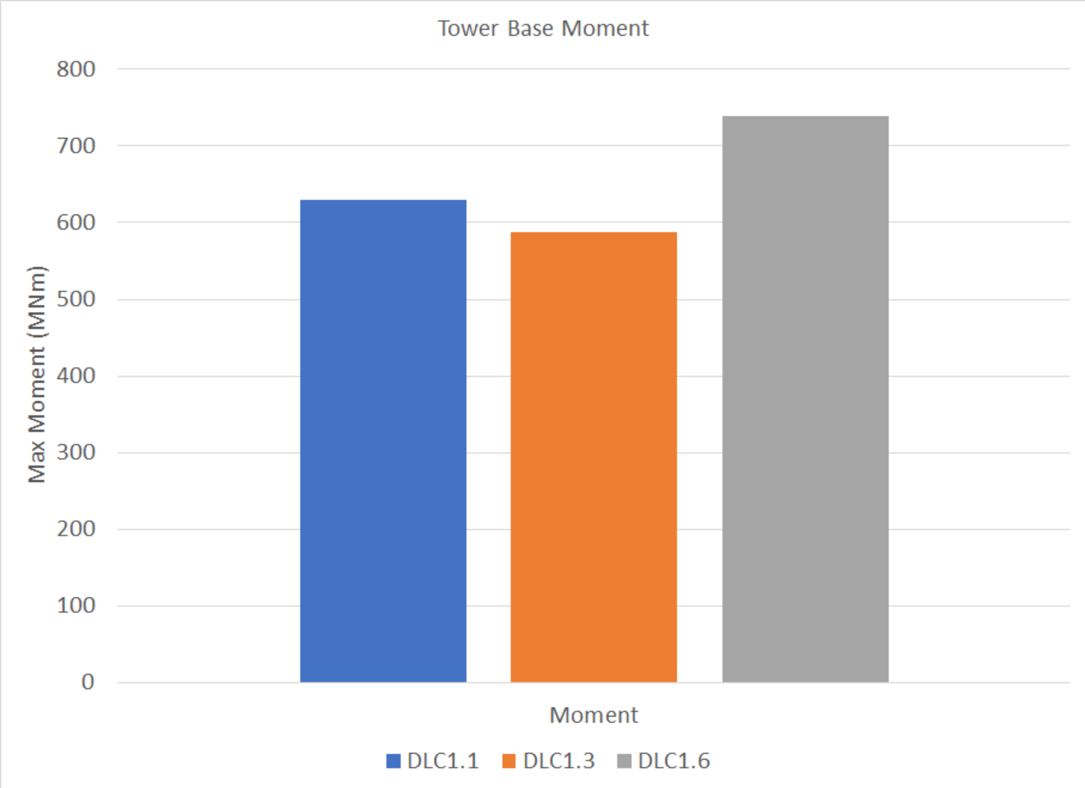
- Nacelle acceleration

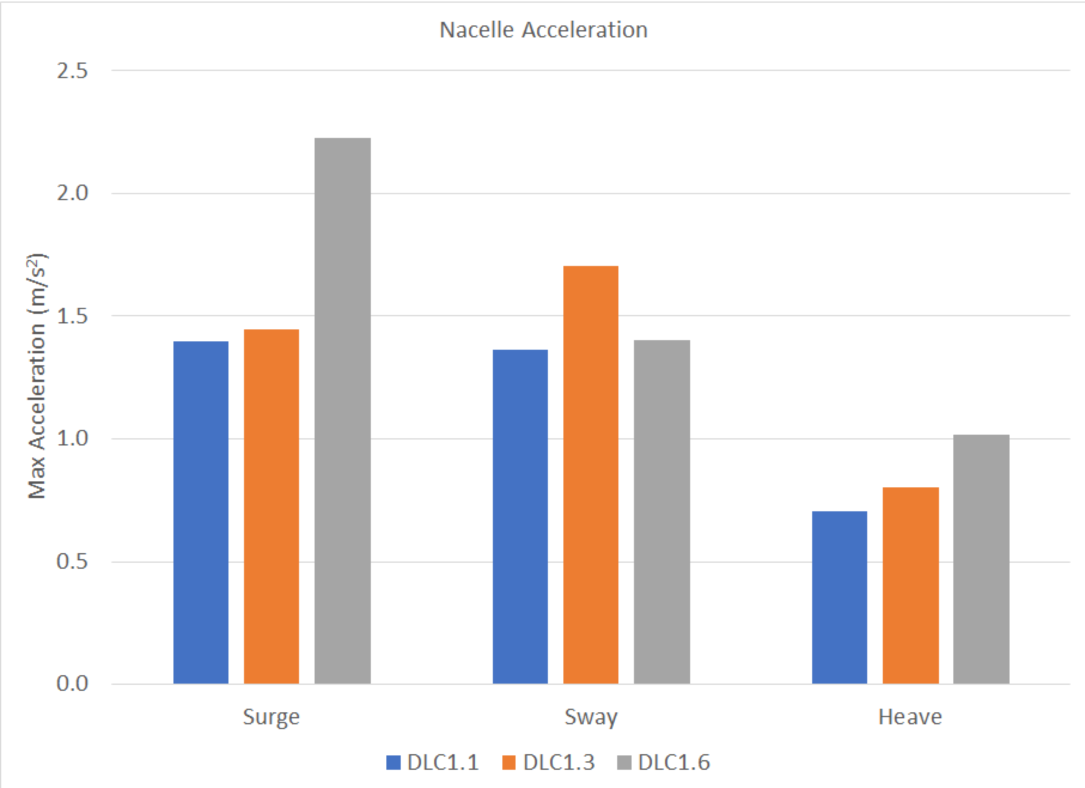
The published report on the UMaine semisub presents overall DLC results in graph form only. The underlying data is not supplied which makes comparisons between UMaine and Flexcom difficult. Hence only the Flexcom results are presented here. However a visual inspection of the published report suggests that the Flexcom results are comparable to the published data. It should be note that for efficiency reasons, the Flexcom simulations considered only one value of wind and wave seed while the UMaine simulations typically considered 6 seeds.




Platform Motion
Max Motion (degrees)
8.0
7.0
6.0
5.0
4.0
3.0
2.0
1.0
0.0
Roll
Pitch
Yaw
DLC1.1
DLC1.3
DLC1.6


Fairlead Tension
Max Tension (MN)
6.0
5.0
4.0
3.0
2.0
1.0
0.0
Mooring Line 1
Mooring Line 2
Mooring Line 3
DLC1.1
DLC1.3
DLC1.6


Tower Base Moment
800
700
600
500
400
300
200
100
0
Max Moment (MNm)
Moment
DLC1.1
DLC1.3
DLC1.6


Nacelle Acceleration
2.5
2.0
1.5
1.0
0.5
0.0
Max Acceleration (m/s²)
Surge
Sway
Heave
DLC1.1
DLC1.3
DLC1.6

1.10.13 M - Wave Energy

Section M contains some examples of wave energy converters:

- M01 - Floating Dual-Body Point Absorber
- M02 - Submerged Tether-Moored Point Absorber
- M03 - Measurement Buoy

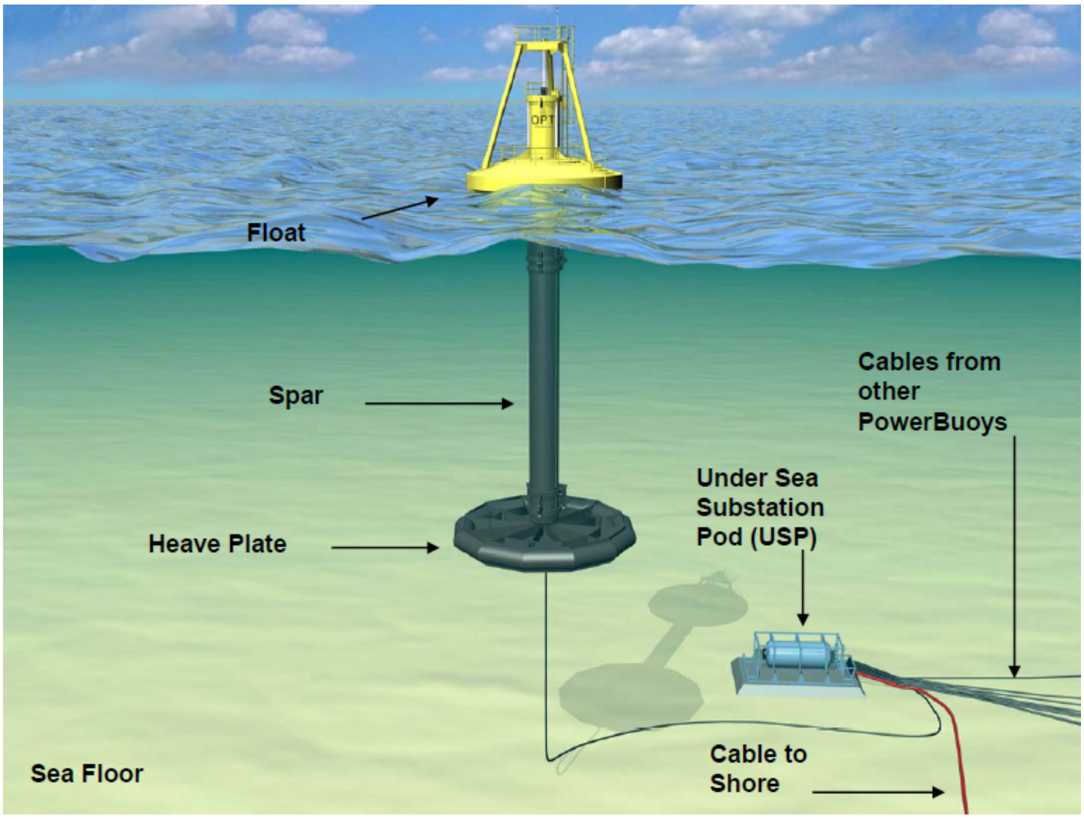
1.10.13.1 M01 - Floating Dual-Body Point Absorber

This example considers a two-body floating point absorber design which was designed by the U.S. Department of Energy as part of their reference model project. The overall layout of this example is as follows:

- Introduction gives an overview of the wave energy converter.
- Model Summary describes the Flexcom model in detail.
- Environment presents the ambient wave conditions, and explains how the load case matrix is simulated in Flexcom.
- Results presents pertinent results from the simulations.

Introduction

This wave energy converter was designed by the U.S. Department of Energy as part of their reference model project. The device concept was inspired by a device known as *PowerBuoy* (Ocean Power Technologies, 2017), which is a two-body floating point absorber design. The device consists of a surface float which moves in response to wave motion, relative to a vertical column spar buoy which is attached to a large reaction plate submerged at a considerable depth below the mean water line. Generation of electrical power occurs predominately by harnessing oscillations of the surface float in the heave direction (the device is designed to accommodate relative heave motions of up to 4 metres along the shaft). Stability of the device is ensured via a spread mooring configuration, and it is designed to operate in water depths of between 40 and 100 metres.

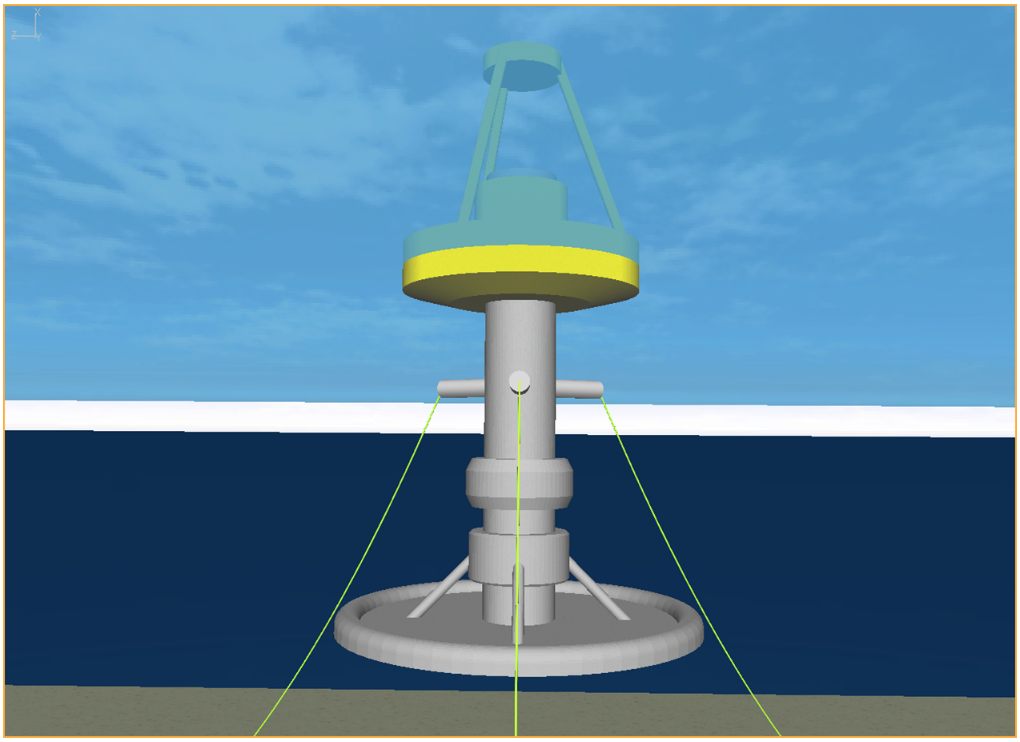


Floating Dual-Body Point Absorber (Ocean Power Technologies, 2017)

Model Summary

INTRODUCTION

The Flexcom model of the wave energy converter is shown below.



Flexcom model of Floating Point Absorber

SURFACE FLOAT & SPAR BUOY

Both the surface float and the spar buoy are modelled using a straight Line along their respective central axes. Line Locations are used to position finite element nodes at the respective centres of gravity. On the spar buoy, a line location point is also defined to identify the vertical elevation of the mooring connections. Each line is assigned rigid Stiffness terms as the individual components are assumed to act as rigid bodies. Each line is assigned a Mass per Unit Length of zero (as the total mass is concentrated at the centre of mass), a Buoyancy Diameter of zero (as the total buoyancy is concentrated at the centre of buoyancy), and a Drag Diameter of zero (to suppress the application of Morison drag loads).

An individual Floating Body is defined for both the surface float and the spar buoy, to model its hydrodynamic characteristics. Inertia terms are specified at the floating body centre of gravity. Hydrostatic Stiffness terms are used to simulate restoring forces and moments due to buoyancy. Added Mass, Radiation Damping and Force RAO coefficients are defined over a range of discrete frequencies - these terms enable the computation of incident, diffracted and radiated (linear) wave potentials to be simulated. Note that these inputs are derived separately from a radiation-diffraction analysis.

Auxiliary Profiles are used to represent the physical profiles of the surface float and the spar buoy. While this has no structural function, it enhances the visual appeal of the model, and assists in the understanding of relative motions post-simulation.

MOORING LINES

3 short Lines are used to create fairlead connections for the mooring lines, providing a radial separation of 3m (corresponding to the spar diameter) between the spar elements and the mooring connection points. The fairlead connections are assigned large Stiffness terms to simulate rigidity, and a Mass per Unit Length of zero. They are not intended to induce any hydrodynamic loading, so they are are assigned zero values of Buoyancy Diameter and Drag Diameter. Equivalent Nodes are used to attach the fairlead elements to the spar elements.

The mooring lines are created using 3 separate Lines. The upper end of each mooring line is attached to the relevant fairlead node on the spar buoy using a Hinge Element, while the lower ends are constrained using Fixed Boundary Conditions. Realistic Stiffness, Mass per Unit Length, Buoyancy Diameter and Drag Diameter terms are assigned to the mooring lines.

POWER TAKE-OFF

The power take-off mechanism is modelled using a Damper Element and a Spring Element in parallel. Both elements run between the respective centres of gravity of the surface float and the spar buoy. The damper provides a linear resistance which is proportional to the rate of expansion or contraction of the power take-off. In regions of free movement the spring element provides a low restoring force. If the mechanism approaches its end stops, the spring stiffness is increased to provide a high level of resistance to any further motion.

Environment

WAVE ENERGY RESOURCE

The wave energy converter was designed for a reference site located offshore of Eureka in Humboldt County, California. This particular location was identified by the U.S. Department of Energy as it has a wave climate representative of the US’s west coast, and moreover a wide range of high-fidelity oceanographic data sets is readily available for this area. A wave scatter diagram for the reference site is presented below (Neary et al., 2014).

								Te									
		4.5s	5.5s	6.5s	7.5s	8.5s	9.5s	10.5s	11.5s	12.5s	13.5s	14.5s	15.5s	16.5s	17.5s	18.5s	19.5s
	0.25m	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	0.75m	0.0%	0.0%	0.6%	0.8%	0.5%	0.5%	0.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	1.25m	0.0%	1.0%	2.7%	3.7%	4.1%	2.9%	1.5%	0.4%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	1.75m	0.0%	1.0%	4.4%	4.3%	4.1%	3.4%	2.0%	1.1%	0.6%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	2.25m	0.0%	0.2%	3.5%	4.2%	3.6%	4.1%	3.1%	1.5%	1.2%	0.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	2.75m	0.0%	0.0%	1.5%	2.5%	1.9%	3.2%	3.3%	1.8%	1.1%	0.4%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%
	3.25m	0.0%	0.0%	0.1%	0.9%	0.9%	2.0%	2.4%	1.4%	0.8%	0.4%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%
	3.75m	0.0%	0.0%	0.0%	0.1%	0.2%	1.0%	1.9%	1.5%	0.5%	0.3%	0.2%	0.1%	0.0%	0.0%	0.0%	0.0%
	4.25m	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%	1.0%	1.3%	0.5%	0.3%	0.2%	0.1%	0.0%	0.0%	0.0%	0.0%
Hs	4.75m	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.3%	0.4%	0.4%	0.2%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%
	5.25m	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.2%	0.3%	0.2%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%
	5.75m	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%	0.1%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%
	6.25m	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	6.75m	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	7.25m	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	7.75m	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	8.25m	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	8.75m	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	9.25m	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	9.75m	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Wave Scatter Diagram for Eureka Site (Neary et al., 2014)

SOLUTION TECHNIQUE

Numerical simulation of an entire scatter diagram in the time domain can be quite computationally expensive, so the Summary Wave Scatter feature offers an economical extrapolation technique with a view to estimating the full results matrix based on a selection of ‘reference seastates’ within the scatter diagram. The scatter diagram is first sub-divided into ‘blocks’, where similar seastates are grouped together, before a single seastate is nominated as being representative for each block. Based on the numerical simulation results for the reference seastate within each block, the program fabricates a Summary Database File for each of the remaining cells within the block. This allows Summary Postprocessing Collation to be used as normal to collate results from all seastates.

Wave Height - Hs (m) \ Wave Period - Te (s)	0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5	20.5	21.5	22.5	23.5	24.5
0.25																									
0.75							0.6	0.8	0.5	0.5	0.2														
1.25						1	2.7	3.7	4.1	2.9	1.5	0.4	0.1												
1.75						1	4.4	4.3	4.1	3.4	2	1.1	0.6	0.1											
2.25						0.2	3.5	4.2	3.6	4.1	3.1	1.5	1.2	0.3											
2.75							1.5	2.5	1.9	3.2	3.3	1.8	1.1	0.4	0.1	0.1									
3.25							0.1	0.9	0.9	2	2.4	1.4	0.8	0.4	0.1										
3.75								0.1	0.2	1	1.9	1.5	0.5	0.3	0.2	0.1									
4.25										0.2	1	1.3	0.5	0.3	0.2	0.1									
4.75											0.3	0.4	0.4	0.2	0.1	0.1									
5.25											0.1	0.2	0.3	0.2	0.1										
5.75												0.2	0.1	0.1	0.1										
6.25												0.1	0.1	0.1											
6.75																									
7.25																									
7.75																									
8.25																									
8.75																									
9.25																									

Nomination of Seastate Blocks and Reference Seastates

DYNAMIC SIMULATIONS

The wave scatter diagram contains a total of 84 different seastates with non-zero percentage occurrences, spanning across 11 different periods and 12 different wave heights. For computational efficiency, the scatter diagram is sub-divided into 11 blocks as shown above, with one block for each different wave period, and a sample wave height selected as the reference seastate within each block.

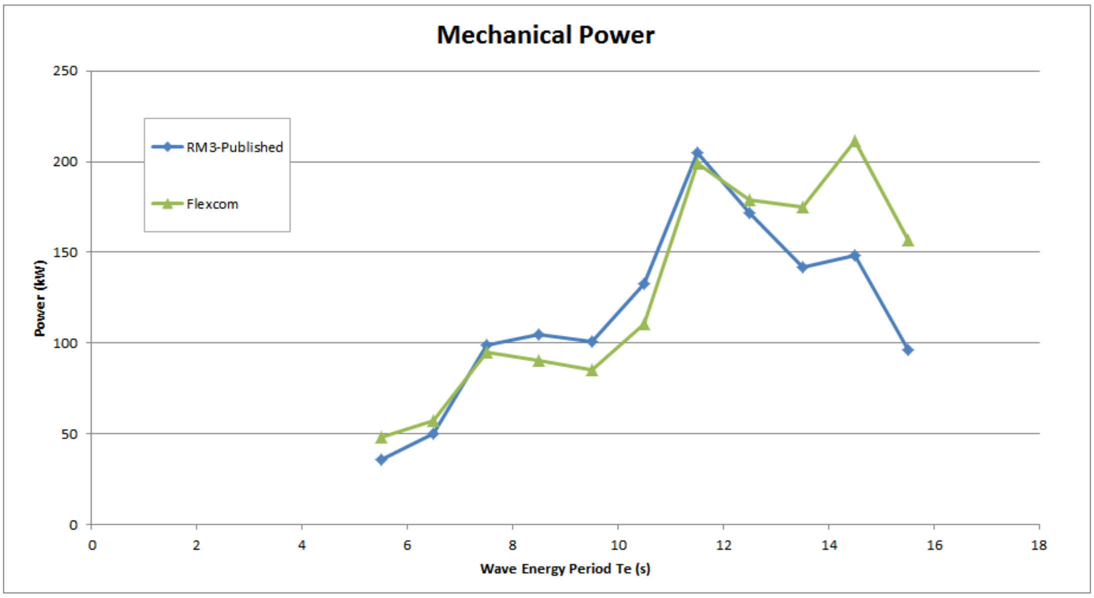
Numerical Parameters are defined to represent significant wave height (H_s) and wave energy period (T_e), and Keyword Based Variations are used to define values of H_s and T_e appropriate to each individual simulation. The `*COMBINATIONS` keyword is used to generate the required input files based on the master template file, and neatly names each file based on the H_s and T_e values. Given that Flexcom has traditionally modelled wave spectra in terms of either T_z or T_p, the T_e values are converted to T_p before being used to define the Jonswap spectrum in Flexcom. Refer to Wave Energy Period for further information on the conversion process.

POSTPROCESSING

After the dynamic simulations for the 11 reference seastates have been completed, the Summary Wave Scatter feature is used to fabricate Summary Database Files for all the non-reference seastates in the wave scatter diagram. The Summary Postprocessing Collation feature is then used to create 3D plots of electrical power, mooring fairlead tensions and mooring anchor loads.

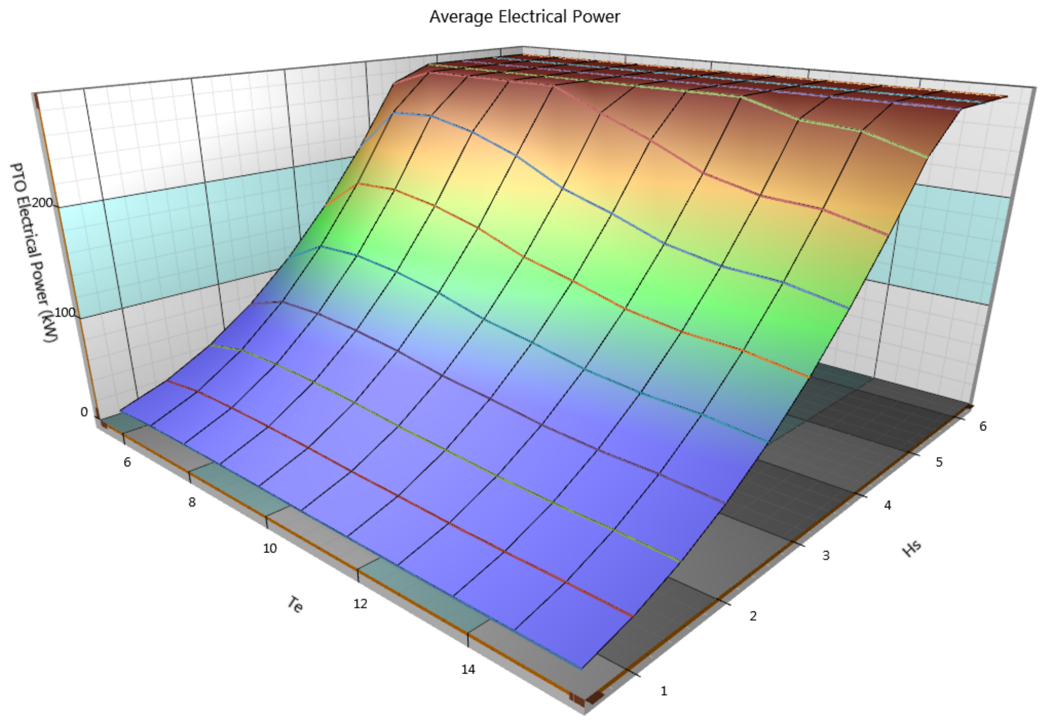
Results

Generated electrical power for the 11 reference seastates is shown below. For comparison purposes, results published on behalf of the U.S. Department of Energy (Neary et al., 2014) are also included. Very close agreement of power output is demonstrated for low to medium wave periods. Some discrepancies begin become to become apparent for the upper wave periods, where Flexcom appears to overestimate the mechanical power from the device. A qualitative assessment of the overall trend suggests that the Flexcom results are consistent with expectations as both data sets are generally well aligned.



Electric Power for Reference Seastates

The full electrical power matrix predicted by Flexcom is shown below.



Electrical Power Matrix

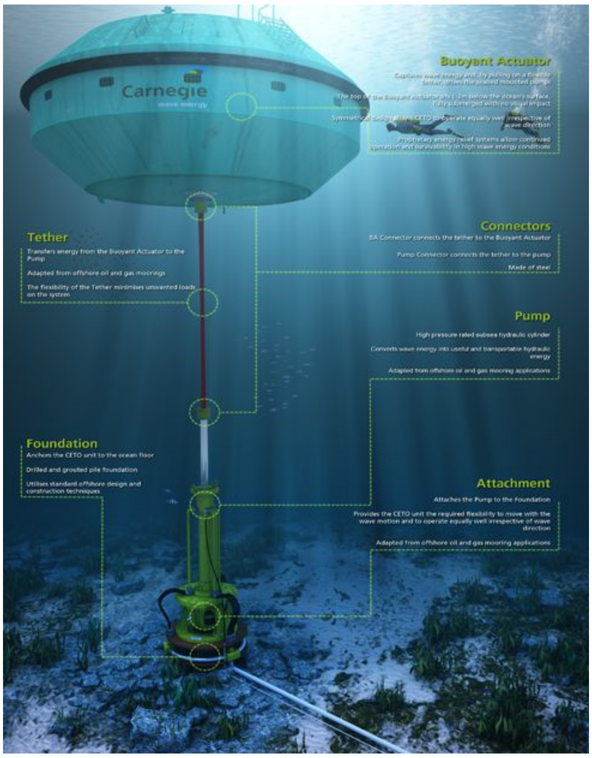
1.10.13.2 M02 - Submerged Tether-Moored Point Absorber

This example considers a bottom-referenced point absorber design. The overall layout of this example is as follows:

- Introduction gives an overview of the wave energy converter.
- Model Summary describes the Flexcom model in detail.
- Environment presents the ambient wave conditions, and explains how the load case matrix is simulated in Flexcom.
- Results presents pertinent results from the simulations.

Introduction

This device concept was inspired by a device known as *CETO5* (Carnegie Clean Energy, 2017), which is a bottom-referenced point absorber design. It consists of a cylindrical buoy which is submerged a couple of metres below the still water line. The device is tethered to a hydraulic pump-based power take-off mechanism which is located on the seabed.



Submerged Tether-Moored Point Absorber (Carnegie Clean Energy, 2017)

Model Summary

INTRODUCTION

The Flexcom model of the wave energy converter is shown below.

Flexcom model of Submerged Point Absorber

BUOY

The buoy is modelled using a straight Line along its central axis. A Line Location is used to position a finite element node at the centre of gravity. The line is assigned rigid Stiffness terms as the buoy is assumed to act as a rigid body. The line is assigned a Mass per Unit Length of zero (as the total mass is concentrated at the centre of mass), a Buoyancy Diameter of zero (as the total buoyancy is concentrated at the centre of buoyancy), and a Drag Diameter of zero (to suppress the application of Morison drag loads).

A Floating Body is defined for the buoy to model its hydrodynamic characteristics. Inertia terms are specified at the centre of gravity. Hydrostatic Stiffness terms are used to simulate restoring forces and moments due to buoyancy. Added Mass, Radiation Damping and Force RAO coefficients are defined over a range of discrete frequencies - these terms enable the computation of incident, diffracted and radiated (linear) wave potentials to be simulated. Note that these inputs are derived separately from a radiation-diffraction analysis.

An Auxiliary Profile is used to represent the physical profile of the buoy. While this has no structural function, it enhances the visual appeal of the model, and assists in the understanding of motions post-simulation.

In order to maintain the buoy in a submerged position, a vertical Point Load is applied to counteract the mean/static uplift force experienced by the buoy.

TETHER

The tether is also modelled using a straight Line, and attached to the base of the buoy using an Equivalent Node. Realistic Stiffness, Mass per Unit Length, Buoyancy Diameter and Drag Diameter terms are assigned to the tether.

POWER TAKE-OFF

The power take-off mechanism is modelled using a Damper Element and a Spring Element in parallel. Both elements run between the seabed anchor point (constrained using Fixed Boundary Conditions) and the base of the tether (connected via an Equivalent Node). The damper provides a linear resistance which is proportional to the rate of expansion or contraction of the power take-off. In regions of free movement the spring element provides a low restoring force. If the mechanism approaches its end stops, the spring stiffness is increased to provide a high level of resistance to any further motion.

Environment

WAVE ENERGY RESOURCE

The wave energy converter was developed and tested offshore Western Australia. At the time of writing, a wave scatter diagram for this location had not been obtained, so a generic scatter diagram was assumed for illustrative purposes.

Wave Height - Hs (m) \ Wave Period - Tp (s)	0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5
0.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
1.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
1.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
2.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
2.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
3.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
3.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
4.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
4.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
5.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
5.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
6.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
6.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
7.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
7.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

Note that as the latest version of Flexcom predicts an electrical power matrix (e.g. kW), rather than estimating total annual energy production (e.g. MWh), the number of occurrence entries in the scatter diagram are actually immaterial as they are not used by the software.

SOLUTION TECHNIQUE

Numerical simulation of an entire scatter diagram in the time domain can be quite computationally expensive, so the Summary Wave Scatter feature offers an economical extrapolation technique with a view to estimating the full results matrix based on a selection of 'reference seastates' within the scatter diagram. The scatter diagram is first sub-divided into 'blocks', where similar seastates are grouped together, before a single seastate is nominated as being representative for each block. Based on the numerical simulation results for the reference seastate within each block, the program fabricates a Summary Database File for each of the remaining cells within the block. This allows Summary Postprocessing Collation to be used as normal to collate results from all seastates.

Wave Height - Hs (m) \ Wave Period - Tp (s)	0.5	1.5	2.5	3.5	4.5	5.5	6.5	7.5	8.5	9.5	10.5	11.5	12.5	13.5	14.5	15.5	16.5	17.5	18.5	19.5
0.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
1.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
1.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
2.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
2.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
3.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
3.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
4.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
4.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
5.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
5.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
6.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
6.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
7.25			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
7.75			1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

Nomination of Seastate Blocks and Reference Seastates

DYNAMIC SIMULATIONS

The wave scatter diagram contains a total of 256 different seastates with non-zero percentage occurrences, spanning across 16 different periods and 16 different wave heights. For computational efficiency, the scatter diagram is sub-divided into 16 blocks as shown above, with one block for each different wave period, and a sample wave height selected as the reference seastate within each block.

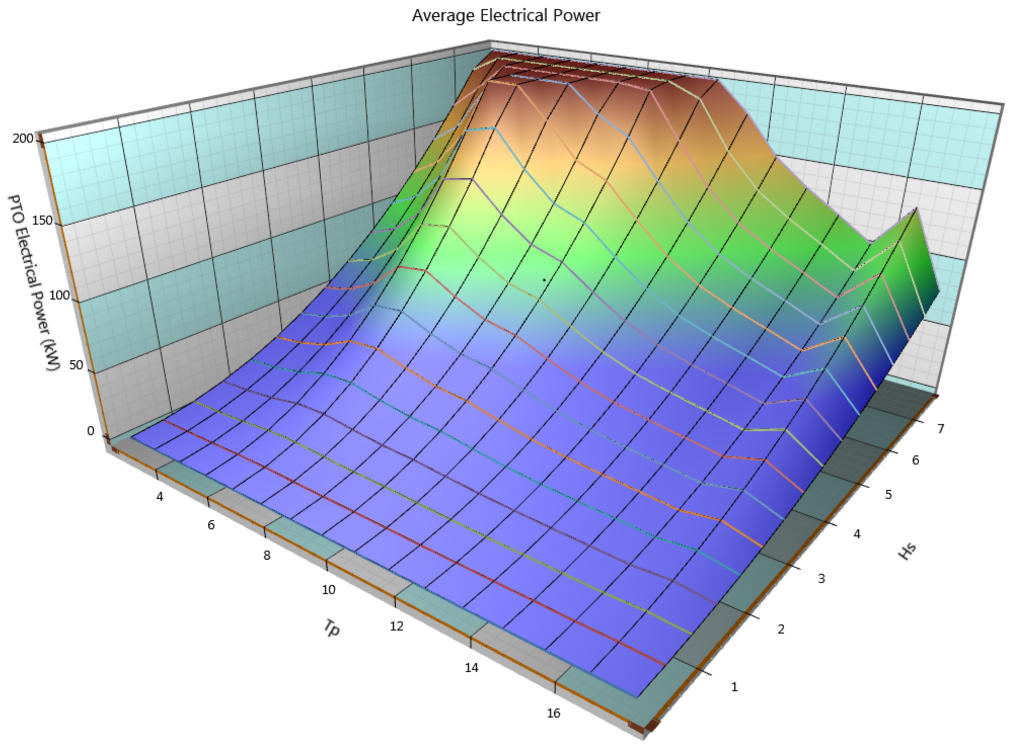
Numerical Parameters are defined to represent significant wave height (H_s) and wave peak period (T_p), and Keyword Based Variations are used to define values of H_s and T_p appropriate to each individual simulation. The `*COMBINATIONS` keyword is used to generate the required input files based on the master template file, and neatly names each file based on the H_s and T_p values.

POSTPROCESSING

After the dynamic simulations for the 16 reference seastates have been completed, the Summary Wave Scatter feature is used to fabricate Summary Database Files for all the non-reference seastates in the wave scatter diagram. The Summary Postprocessing Collation feature is then used to create 3D plots of electrical power and tether fairlead tensions.

Results

The full electrical power matrix predicted by Flexcom is shown below.



Electrical Power Matrix

1.10.13.3 M03 - Measurement Buoy

This example considers a wave measurement buoy. The overall layout of this example is as follows:

- Model Summary gives a general overview of the buoy and describes the Flexcom model in detail.
- Environment presents the ambient wave conditions, and explains how the load case matrix is simulated in Flexcom.

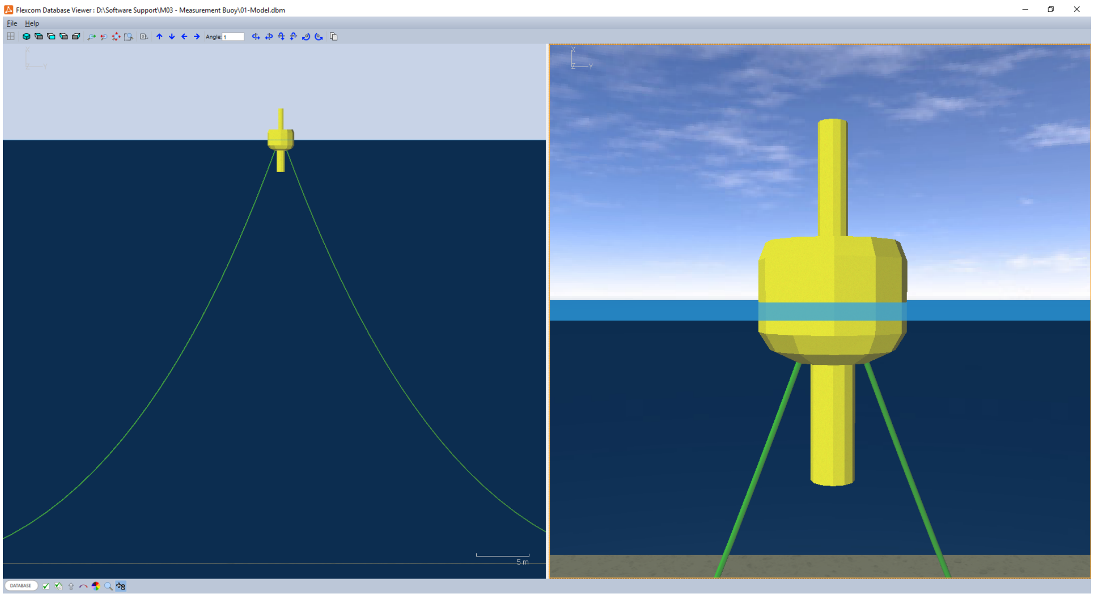
- Results presents pertinent results from the simulations.

Model Summary

INTRODUCTION

Wave measurement buoys are deployed to measure the water surface elevation as a function of time. This data can be logged for subsequent processing, providing statistical information such as significant wave height (Hs), mean up-crossing period (Tz), and wave directionality. The buoy considered in this example has a steel core with buoyancy material attached to it. The float is 2.5m in diameter and 2.0m high. It is situated in 40m water depth and is held on station using two mooring ropes.

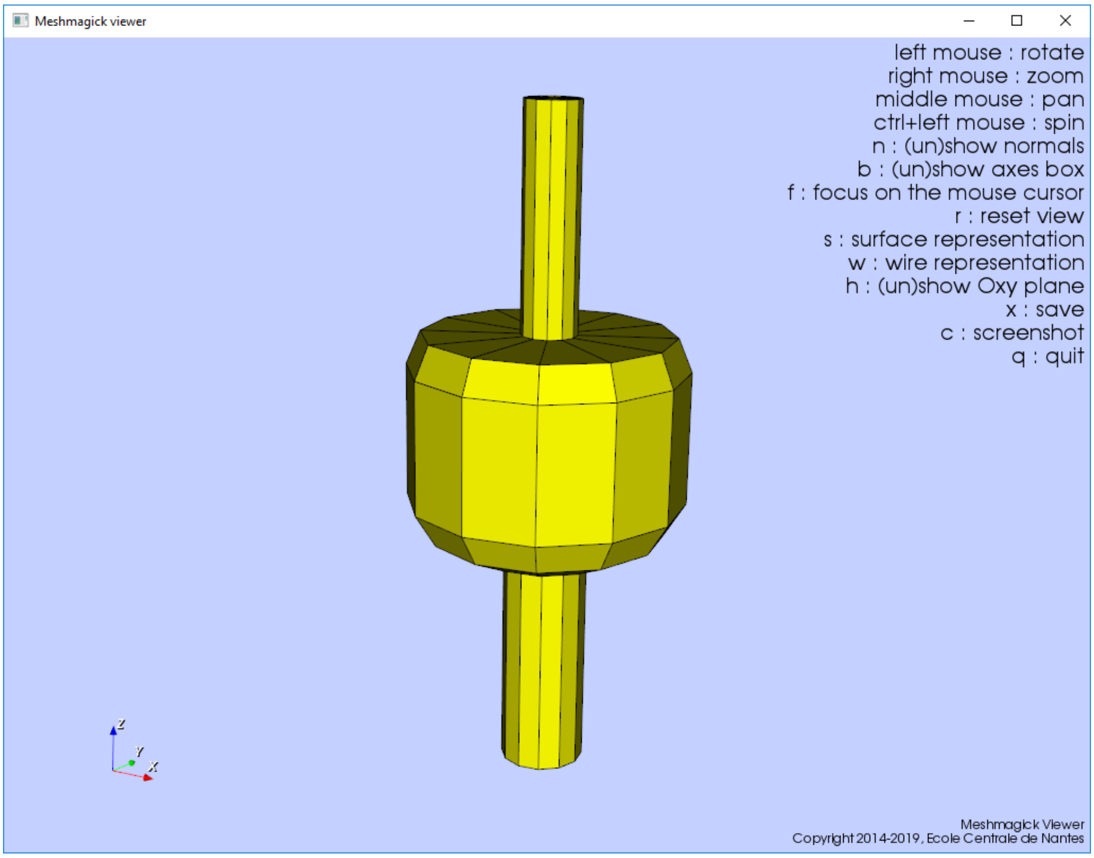
This example is based on a wave measurement buoy which was deployed at the SmartBay test site, Ireland's national observation and validation facility for the ocean energy sector.



Wave Measurement Buoy

HYDRODYNAMIC ANALYSIS

A hydrodynamic analysis of the buoy is performed in NEMOH, in order to compute its hydrodynamic characteristics. This data is then passed into Flexcom where it is combined with the structural model to examine the global response of the system. The image below shows the mesh discretisation used in the hydrodynamic simulation (the mesh visualisation is provided by Meshmagick).



Hydrodynamic Mesh illustrated by Meshmagick

The buoy profile is created approximately using a spreadsheet (see 'M03 - Measurement Buoy.xlsx'). It is sub-divided into a series of horizontal sections along its length, corresponding to the changes in diameter. Each section is modelled using 12 points which are equally spaced around the circumference at 30 degree intervals. Hydrodynamic meshes are typically derived from detailed specification drawings (e.g. AutoCAD) but in the case of this relatively simple shape, it was feasible to create it in a spreadsheet. It is not necessary to include any part of the body which lies above the water line, but the full body is modelled, as no further mesh creation is required if the body's draft is altered subsequently (e.g. due to additional weight imparted by mooring lines). The hydrodynamic analysis is performed by running NEMOH, and the hydrodynamic characteristics are converted into Flexcom format using the Hydrodynamic Data Importer.

MEASUREMENT BUOY

The buoy is modelled using a straight Line along its central axis. Line Locations are used to position finite element nodes at the centres of gravity and buoyancy. The line is assigned rigid Stiffness terms as it is assumed to act as a rigid body. The line is assigned a Mass per Unit Length of zero (as the total mass is concentrated at the centre of mass), a Buoyancy Diameter of zero (as buoyancy effects are modelled at the centre of buoyancy). Hydrodynamic drag is set to zero in order to suppress the application of viscous drag loads via Morison's equation (although realistic Drag Diameters are specified should you wish to add viscous drag yourself subsequently).

An individual Floating Body is defined to model the buoy's hydrodynamic response. Inertia terms are specified at its centre of gravity. Hydrostatic Stiffness terms are used to simulate restoring forces and moments due to buoyancy. Added Mass, Radiation Damping and Force RAO coefficients are obtained from the NEMOH simulation - these terms enable the computation of incident, diffracted and radiated wave loading.

An auxiliary profile is used to represent the physical profile of the buoy. While this has no structural function, it enhances the visual appeal of the model, and assists in the understanding of motions post-simulation. The profile is created from the 'Description_Full.tec' file which is created by NEMOH's meshing algorithm.

MOORING LINES

One short line is used to create a connecting frame to link the buoy with its mooring lines. The frame is assigned large Stiffness terms to simulate rigidity, and a Mass per Unit Length of zero for simplicity. It is not intended to induce any hydrodynamic loading, so it is assigned zero values of Buoyancy Diameter and Drag Diameter. The Equivalent Nodes feature is used to attach the frame to the buoy.

The mooring lines are created using 2 separate Lines. The upper end of each mooring line is attached to the connecting frame using a Hinge Element, while the lower ends are constrained using Fixed Boundary Conditions. Realistic Stiffness, Mass per Unit Length, Buoyancy Diameter and Drag Diameter terms are assigned to the mooring lines.

Environment

INTRODUCTION

The measurement buoy is subjected to a series of regular waves to examine its response to wave loading across a range of wave periods.

Before the wave loading is applied, a series of static analyses are performed in order to establish a certain tension distribution in the mooring lines. The actual tension distribution in reality will be a function of the lay process, including any pre-tension, and this will be different for every installation. In the majority of the standard examples provided with Flexcom which include catenary sections (whether riser or mooring line), the upper and lower ends of the catenary section are fixed, and a single initial static analysis is performed. In these cases the effective tension distribution remains constant along the seabed, as no frictional forces are mobilised during the computation of the static configuration. This modelling approach is adopted (i) for simplicity reasons and (ii) because the exact tension distribution is dependent on the lay process (which is not being modelled explicitly), as friction is a path-dependent phenomenon. A more complex approach is adopted in this example in order to illustrate how to model tension dissipation along the seabed due to frictional effects.

STATIC ANALYSIS STAGES

A series of static analysis stages are performed as follows:

- 01-Model.keyxm. Initial static analysis to determine the overall configuration due to gravity and buoyancy loads. The anchor points are fixed in all translational degrees of freedom. The buoy is restrained also to prevent any movement in the vertical direction. Node springs are inserted to help restrict horizontal movement of the anchor points in Stage 3.
- 02-FixTDP-ReleaseAnchor.keyxm. Restart analysis in which the touchdown points are fixed in the horizontal direction while the anchor points are free to move horizontally. Seabed friction effects are temporarily disabled for this stage.

- 03-ReleaseTDP.keyxm. Restart analysis in which the touchdown points are free to move horizontally. As the anchor points are also free, the mooring line can slide horizontally along the seabed subject to frictional resistance. The node springs at the anchor points offer a low but non-zero resistance which allows the mooring line to move sufficiently to generate frictional forces along the seabed. The anchor point moves about 1.2m, which is sufficient to ensure that the friction mobilisation length is exceeded at all seabed contact nodes, while only reducing the fairlead tension slightly.

- 04-FixAnchor.keyxm. Restart analysis in which the anchor points are re-fixed at their respective locations following the previous analysis stage.

- 05-QuasiStatic.keyxm. Restart analysis in which the buoy is free to move. The weight of the mooring lines is counteracted by the buoyancy stiffness and the buoy settles to its equilibrium position. Strictly speaking, the hydrodynamic analysis should be re-run again in NEMOH as the buoy draft has changed slightly, however this additional stage was omitted in this example for simplicity.

REGULAR WAVE ANALYSES

Numerical Parameters are defined to represent wave amplitude and wave energy period, and Keyword Based Variations are used to vary the wave period from 2s to 25s in increments of 0.5s. The `*COMBINATIONS` keyword is used to generate the required input files based on the master template file, and neatly names each file based on wave period.

POSTPROCESSING

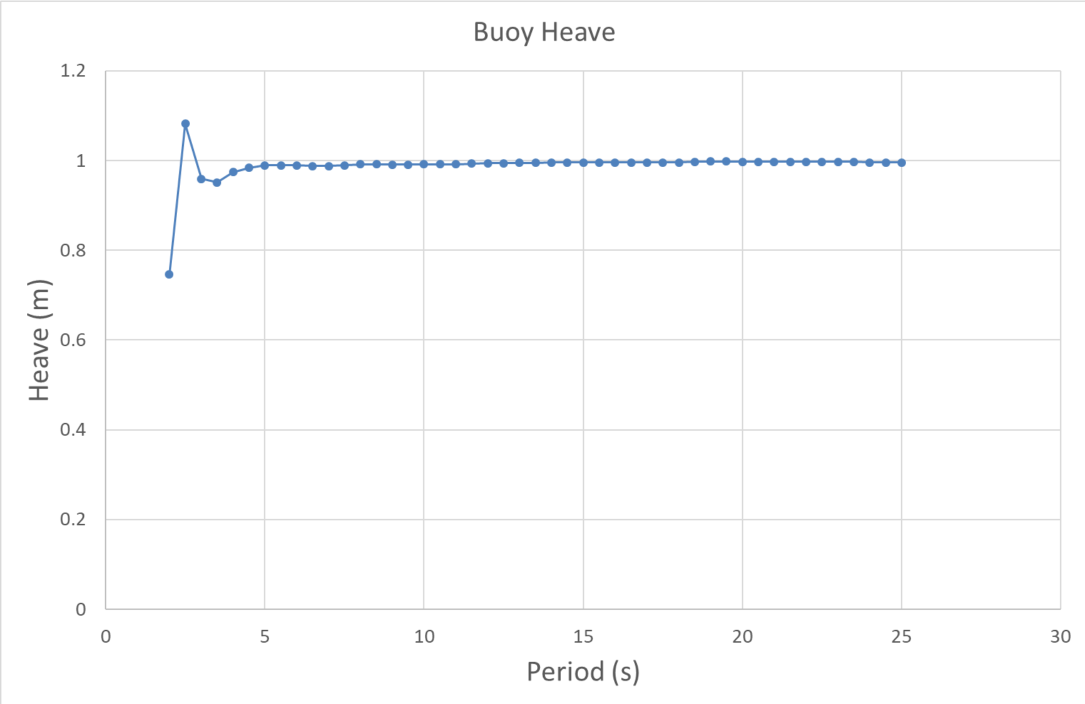
Summary Postprocessing is used to extract the buoy motions from each simulation, and the Summary Collation feature is then used to collate all the data into a spreadsheet.

Results

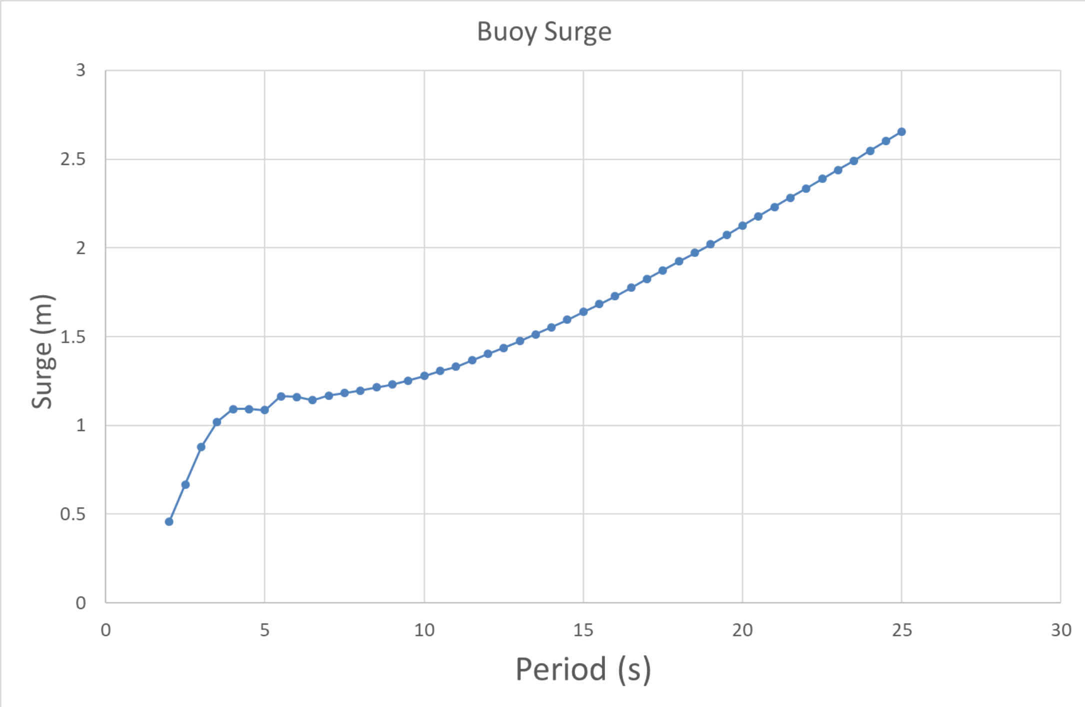
BUOY RESPONSE

The following figures illustrate the buoy response in heave, surge and pitch. As the amplitude of the regular waves is set to 0.5m, the response ranges quoted in the summary output file, and the summary collation spreadsheet subsequently, effectively represent the buoy RAOs. Ideally, smaller wave amplitudes should have been used with the shorter wave periods, but for simplicity a uniform amplitude was used across all periods in this example.

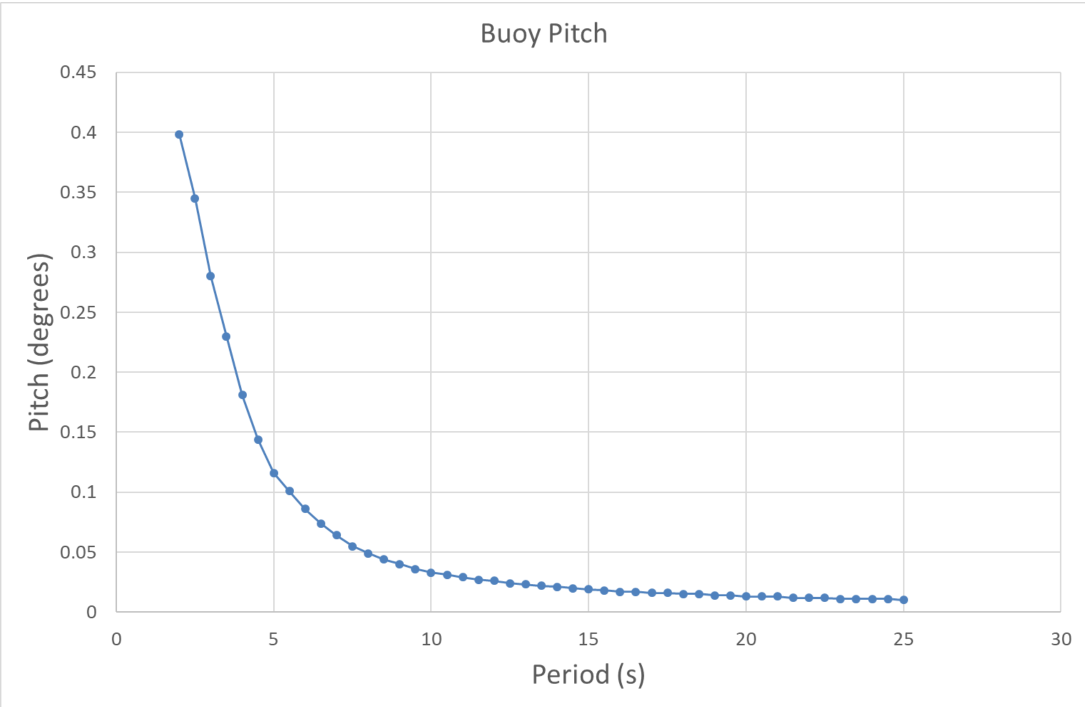
The heave response is close to unity at most wave periods, with the exception of some of the shortest ones. This is consistent with expectations as a measurement buoy is expected to ride the waves. Surge motions steadily increase with increasing period, with limited horizontal resistance being offered by the mooring ropes. Movement could be limited by the use of heavier chains on the seabed if required. Buoy pitch is very limited, with the largest pitch response being demonstrated at the lower wave periods.



Buoy Heave



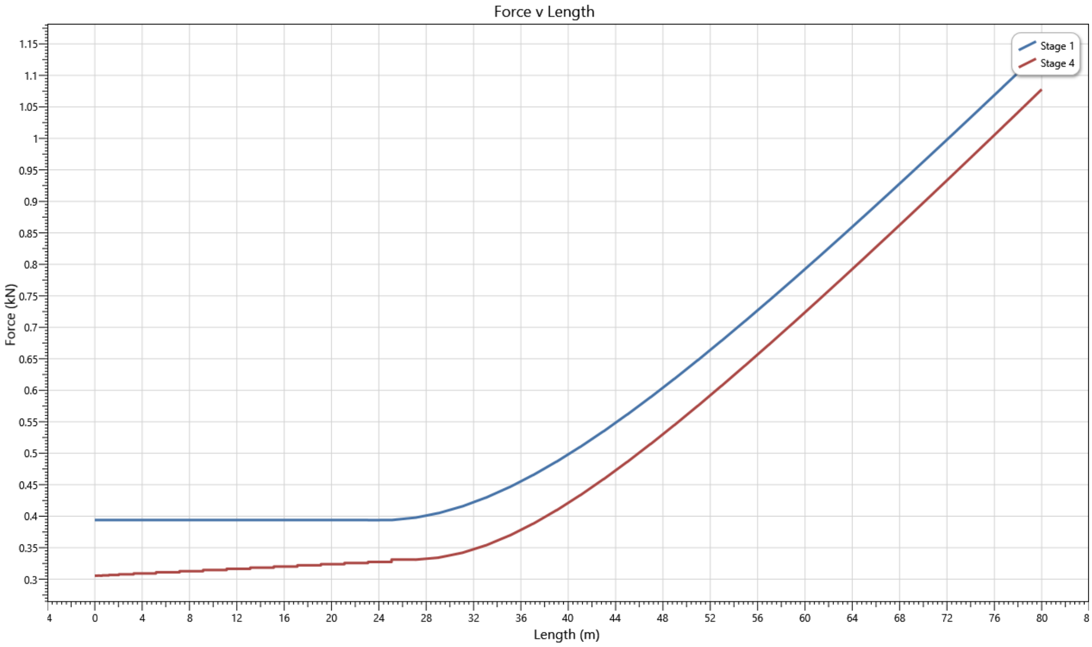
Buoy Surge



Buoy Pitch

MOORING LINE TENSION

The next figure shows the static mooring line tension, illustrating the mobilisation of frictional forces along the seabed via the initiation stages. In the initial static analysis (Stage 1), the effective tension distribution remains constant along the seabed, as no frictional forces are mobilised during the computation of the static configuration. After the release and re-fixing of the anchor point (Stage 4), the tension gradually reduces between the touchdown point and the anchor.



Mooring Line Tension

1.11 Troubleshooting Guide

SIMULATIONS FAILING TO START

- Job Execution Service & Troubleshooting describes the intermediary entity which manages the interaction between the user interface and the finite element engine programs, and provides some useful advice to overcome any communication problems should they arise.

- NetHASP License Troubleshooting describes steps that are useful for troubleshooting any issues with network licensing which uses the NetHASP system (physical red dongle).

SIMULATIONS FAILING TO COMPLETE SUCCESSFULLY

- Troubleshooting Simulation Failures presents some general guidelines regarding the use of Flexcom, and also gives some advice on what to do if an analysis fails to complete successfully.

USER INTERFACE ISSUES

- The Model View Troubleshooting guide should be useful if you experience problems with the Model View (e.g. persistently blank Model View).

FURTHER ASSISTANCE

- Our on-line Software Knowledge Base is a web based resource which contains answers to frequently asked questions. It essentially provides best practice advice on how to use the software and troubleshoot issues commonly encountered by software users. First time users will be asked to register with their company email address and individual password. New users are approved as soon as possible after registration.

- Contact Wood if you require further assistance from the Flexcom Software Support Team.

1.12 References

This section lists all the technical publications referenced throughout the help system

1. Allen, C., Viscelli, A., Dagher, H., Goupee, A., Gaertner, E., Abbas, N., Hall, M. and Barter, G., 2020. Definition of the UMaine VolturnUS-S Reference Platform Developed for the IEA Wind 15-Megawatt Offshore Reference Wind Turbine (No. NREL/TP-5000-76773). National Renewable Energy Lab. (NREL), Golden, CO (United States). Online at [https://www.nrel.gov/docs/fy20osti/76773.pdf].

2. Ang, Alfredo H-S., and Tang, Wilson H., "Probability Concepts in Engineering Planning and Design – Vol. 2, Decision, Risk and Reliability", John Wiley & Sons, New York, 1984.

3. ASTM Standard E1049-85, 1985 (2005), “Standard Practices for Cycle Counting in Fatigue Analysis”, ASTM International, 2005, DOI: 10.1520/E1049-85R05, www.astm.org.

4. Bathe, K. J., “Finite Element Procedures in Engineering Analysis”, Prentice-Hall, 1982.

5. Bathe, K. J. and Wilson, E. L., "Numerical Methods in Finite Element Analysis", Prentice-Hall, 1976.

6. Bergan, P.G. and Mollestad, E., “An Automatic Time-Stepping Algorithm for Dynamic Problems”, Computer Methods in Applied Mechanics and Engineering, Vol. 49, 1985, pp. 299-318.

7. Blevins, Robert D., “Forces on and Stability of a Cylinder in a Wake”, Journal of Offshore Mechanics and. Arctic Engineering. Volume 127, 2005, pp. 39-45.

8. Cahill, B. and Lewis, T. (2014). “Wave period ratios and the calculation of wave power”. Proceedings of the 2nd Marine Energy Technology Symposium, METS2014, April 15-18, 2014, Seattle, WA.

9. Carnegie Clean Energy (2017), “What is CETO”. [Online at https://www.carnegiece.com/wave/what-is-ceto/].

10. Chakrabarty, J., "Theory of Plasticity", Third Edition, Elsevier Butterworth-Heinemann, 2006

11. Chakrabarti, S. K., Hydrodynamics of Offshore Structures, Computational Mechanics Publications/Springer-Verlag, 1987.

12. Chaudhuri, J., McNamara, J.F., and O'Brien, P.J. “Non-Linear Dynamic Analysis of Hybrid Riser System for Deep Water Application”, Proceedings of the 19th Annual Offshore Technology Conference, Houston, Texas, April 27-30th, 1987, Paper No. OTC-5466, pp. 405-416.

13. Chung, J. and Hulbert, G. M., "A Time Algorithm for Structural Dynamics with Improved Numerical Dissipation: The Generalised-α Method", Journal of Applied Mechanics, June 1993, pp. 371 – 375.

14. Clauss, G., Lehmann, E. and Östergaard, C., 2014. Offshore structures: volume I: conceptual design and hydromechanics. Springer.

15. Connaire, A.D., Lang, D.W. and Galvin, C.D., “Closely-Moored Floating Bodies in a Production and Offloading Facility – Requirement for and Application of a Coupled Analysis Capability”, OTC 15378, 2003.

16. Connolly, A. J. and Brewster, P. Application of Coupled Numerical Simulation to Design Wave Energy Converters. In The 27th International Ocean and Polar Engineering Conference. International Society of Offshore and Polar Engineers, 2017.

17. Connolly, A.J. and O'Mahony, G., 2021, June. Validation of a Novel Wind Turbine Simulation Tool via Benchmarking: Case Study of Jacket Structure. In The 31st International Ocean and Polar Engineering Conference. OnePetro.

18. Connolly, A.J. and O'Mahony, G., 2021, September. Validation of a Novel Floating Wind Turbine Simulation Tool via Benchmarking: Case Study of a Semi-Submersible Platform. In SPE Offshore Europe Conference & Exhibition. OnePetro. https://doi.org/10.2118/205415-MS.

19. Connolly, A., O’Mahony, G. and O’Connor, A. Computationally Efficient Simulation of Floating Dual-Body Point Absorber. In the 7th International Conference on Ocean Energy (ICOE), 2018. [Online at https://www.icoe-conference.com/publication/computationally-efficient-simulation-of-floating-dual-body-point-absorber/]

20. Connolly, A., Guyot, M., Le Boulluec, M., Héry, L. and O’Connor, A., 2018, November. Fully Coupled Aero-Hydro-Structural Simulation of New Floating Wind Turbine Concept. In ASME 2018 1st International Offshore Wind Technical Conference. American Society of Mechanical Engineers Digital Collection.

21. Cummins W.E., ‘The Impulse Response Function and Ship Motions’, Schiffstechnik, vol. 9 (47), 1962, pp. 101-109.

22. Fylling, I, J., Bech, A., Sødahl, N., “The Effects of Slug Flow on the Dynamic Response of Flexible Risers and Submerged Hoses”, 4th International Conference on Floating Production Systems, IBC Technical Services, London, 1988.

23. Gaertner, E., Rinker, J., Sethuraman, L., Zahle, F., Anderson, B., Barter, G., Abbas, N., Meng, F., Bortolotti, P., Skrzypinski, W. and Scott, G., 2020. Definition of the IEA 15-Megawatt offshore reference wind turbine. Online at [https://www.nrel.gov/docs/fy20osti/75698.pdf].

24. Goda, Y., “A Review on Statistical Interpretation of Wave Data”, Report of the Port and Harbour Research Institute, March 1979, Vol. 18, No. 1, pp. 5-32.

25. Goupee, A.J., Koo, B., Lambrakos, K. and Kimball, R., 2012, April. Model tests for three floating wind turbine concepts. In Offshore technology conference. Offshore Technology Conference.

26. Guyot, M., De Mourgues, C., Le Bihan, G., Parenthoine, P., Templai, J., Connolly, A. and Le Boulluec, M., 2019, December. Experimental Offshore Floating Wind Turbine Prototype and Numerical Analysis During Harsh and Production Events. In ASME 2019 2nd International Offshore Wind Technical Conference. American Society of Mechanical Engineers Digital Collection.

27. Hamilton, J, "Three Dimensional Fourier Analysis of Drag Force for Compliant Offshore Structures", Applied Ocean Research, Vol. 2, No. 4, 1980, pp. 63-69.

28. Hasselmann K., T.P. Barnett, E. Bouws, H. Carlson, D.E. Cartwright, K. Enke, J.A. Ewing, H. Gienapp, D.E. Hasselmann, P. Kruseman, A. Meerburg, P. Mller, D.J. Olbers, K. Richter, W. Sell, and H. Walden. Measurements of wind-wave growth and swell decay during the Joint North Sea Wave Project (JONSWAP)' Ergnzungsheft zur Deutschen Hydrographischen Zeitschrift Reihe, A(8) (Nr. 12), p.95, 1973.

29. Hilber, H.M., Hughes, T.J.R. and Taylor, R.L., “Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics”. Earthquake Engineering and Structural Dynamics, Vol. 5, 1977, pp. 283-291.

30. Houmb, O.G. and Overvik, T., “Parameterisation of Wave Spectra and Long Term Joint Distribution of Wave Height and Period”, Proceedings of Conference on Behaviour of Offshore Structures (BOSS), Trondheim, 1976, Vol. 1.

31. Huse, E., “Interaction in Deep-Sea Riser Arrays”, Offshore Technology Conference, OTC 7237, 1993.

32. Isherwood, R.M., “Technical Note: A Revised Parameterisation of the JONSWAP Spectrum”, Applied Ocean Research, Vol. 9, No. 1, 1987, pp. 47-50.

33. Jonkman, J. M., "The new modularization framework for the FAST wind turbine CAE tool." 51st AIAA Aerospace Sciences Meeting and 31st ASME Wind Energy Symposium, Grapevine, Texas. 2013.

34. Karve, S., O'Brien, P.J., and McNamara, J.F., “Comparison of Dynamic Response of Alternate Flexible Riser Products”, Proceedings of the 20th Annual Offshore Technology Conference, Paper No. OTC-5796, Houston, Texas, May 1988, pp. 459-466.

35. Kavanagh, W. K., Connaire, A., Payne, J., Connolly, A., & McLoughlin, J. (2013). Slug Response in Subsea Piping: Advancements in Analytical Methods. In Offshore Technology Conference. Offshore Technology Conference.

36. Lang, D.W., Connolly A., Lane. M., Connaire A.D., “Advances in frequency and time domain coupled analysis for floating production and offloading system”, 24th International Conference on Offshore and Artic Engineering, 12-17 June 2005.

37. Langley, R. S., "The Linearisation of Three Dimensional Drag Force in Random Seas with Current", Applied Ocean Research, Vol. 6, No. 3, 1984, pp. 126-131.

38. Liu, H. Y., Wang, K., & Yiu, F., Analyses of SCR with Pull-Tube Using ABAQUS and Flexcom. In Offshore Technology Conference. Offshore Technology Conference, 2013.

39. Matlock, H. (1970). Correlations for design of laterally loaded piles in soft clay. Offshore Technology in Civil Engineering’s Hall of Fame Papers from the Early Years, 77-94.

40. McCann, D., Smith, F. and O’Brien, P., “Guidelines for Compression Modeling in Flexible Risers for Deepwater Applications”, OTC 15168, 2003.

41. McNamara, J. F. and Lane, M., "Computation Tools for the Time and Frequency Domain Analysis of TLP Tethers and Risers", Paper No. OMAE 91-351, 10th International Conference on Offshore Mechanics and Arctic Engineering, OMAE 1991, Stavanger, June 1991.

42. McNamara, J.F. and O'Brien, P.J., “Response of Flexible Pipelines to Ocean Wave Loading”, Proceedings of the 4th Conference of the Irish Durability and Fracture Committee, Queen's University, Belfast, September, 1986, edited by F.R. Montgomery, ISBN 0853892822, pp. 306-316.

43. McNamara, J.F., O'Brien, P.J., and Gilroy, J.P., “Nonlinear Analysis of Flexible Risers Using Hybrid Finite Elements”, Proceedings of the 5th International Offshore Mechanics and Arctic Engineering Symposium, ASME, Tokyo, April 1986, Vol. 3, pp. 371-377. (Later published in the Journal of Offshore Mechanics and Arctic Engineering, ASME, Vol. 110, No. 3, August 1988, pp. 197-204).

44. Neary, V. S., Lawson, M., Previsic, M., Copping, A., Hallett, K. C., LaBonte, A., ... & Murray, D. (2014). “Methodology for design and economic analysis of marine energy conversion (MEC) technologies”. SANDIA REPORT, SAND2014-9040. [Online at http://energy.sandia.gov/download/23111/].

45. Newman, J. N., "Second-order, slowly-varying forces on vessels in irregular waves." (1974).

46. NREL. 2020. Reference OpenSource Controller (ROSCO) for wind turbine applications. Online at [https://github.com/NREL/rosco].

47. O’Brien, P.J., Lane, M. and McNamara, J.F., “Improvement to the Convected Co-ordinates Method for Predicting Large Deflection Extreme Riser Response”, Proceedings of the 21st International Conference on Offshore Mechanics and Arctic Engineering, June, 2002, Oslo, Norway.

48. O'Brien, P.J., and McNamara, J.F., “Analysis of Flexible Riser Systems Subject to Three-Dimensional Seastate Loading”, Proceedings of the International Conference on Behaviour of Offshore Structures, (BOSS '88), Trondheim, Norway, June 1988, Vol. 3, pp. 1373-1388.

49. O'Brien, P.J., and McNamara, J.F., “Benchmark Solutions for Nonlinear Two and Three Dimensional Rigid and Flexible Marine Riser Problems”, Proceedings of the ABAQUS User's Conference, June 1-3rd, Newport, U.S.A., 1988.

50. O'Brien, P.J., and McNamara, J.F., “Computations of Flexible Pipeline Systems in Subsea Engineering”, Progress in Subsea Engineering, Published by BHRA, The Fluid Engineering Centre, Cranfield, UK, 1989, pp. 1-18.

51. O'Brien, P.J., McNamara, J.F., and Dunne, F.P.E., “Three-Dimensional Nonlinear Motions of Risers and Offshore Loading Towers”, Proceedings of the 6th International Offshore Mechanics and Arctic Engineering Symposium, ASME, Houston, Texas, March 1987, pp. 171-176. (Later published in the Journal of Offshore Mechanics and Arctic Engineering, ASME, Vol. 110, No. 3, August 1988, pp. 232-237).

52. O'Brien, P.J., McNamara, J.F., and Grealish, F., “Extreme Bending and Torsional Responses of Flexible Pipelines”, Proceedings of the 11th International Offshore Mechanics and Arctic Engineering Symposium, ASME, Calgary, Canada, June 1991, pp. 319-324

53. O'Brien, P.J., McNamara, J.F. and Lane, M., "Three Dimensional Finite Displacements and Rotations of Flexible Beams Including Non-equal Bending Stiffnesses", Proceedings of the 22nd International Conference on Offshore Mechanics and Arctic Engineering, June, 2003, Cancun, Mexico.

54. O'Neill, M. W., & Murchison, J. M. (1983). An evaluation of py relationships in sands. University of Houston.

55. Ocean Power Technologies (2017), "PB3 PowerBuoy". [Online at http://www.oceanpowertechnologies.com].

56. Ochi, Michael K., "Ocean Waves – The Stochastic Approach", Cambridge University Press, 1998.

57. Ochi, M.K. and E.N. Hubble, "Six-Parameter Wave Spectra", Proceedings of the Fifteenth International Conference on Coastal Engineering, Honolulu, Hawaii, ASCE, 1976, pp.301-328.

58. Patel, M.H., and Seyed, F.B., Engineering Structures, Volume 11, Issue 4, October 1989, "Internal Flow-Induced Behaviour of Flexible Risers" Pages 266-280.

59. Patrikalakis, N. M., International Offshore Mechanics and Arctic Engineering (OMAE) Symposium, Tokyo, 13-18 April 1986, "Three-Dimensional Compliant Riser Analysis" pp. 458-465.

60. Pierson, Willard J., Jr. and Moskowitz, Lionel A. Proposed Spectral Form for Fully Developed Wind Seas Based on the Similarity Theory of S. A. Kitaigorodskii, Journal of Geophysical Research, Vol. 69, p.5181-5190, 1964.

61. Platt, A., Jonkmann, B., Jonkmann, J. InflowWind Users Guide, National Wind Technology Center, July 2016. [Online at https://wind.nrel.gov/nwtc/docs/InflowWind_Manual.pdf]

62. Popko, W., Vorpahl, F., Zuga, A., Kohlmeier, M., Jonkman, J., Robertson, A., Larsen, T.J., Yde, A., Saetertro, K., Okstad, K.M. and Nichols, J., 2012. Offshore code comparison collaboration continuation (OC4), Phase I-results of coupled simulations of an offshore wind turbine with jacket support structure (No. NREL/CP-5000-54124). National Renewable Energy Lab.(NREL), Golden, CO (United States).

63. Reese, L. C., Cox, W. R., & Koop, F. D. (1975, January). Field testing and analysis of laterally loaded piles in stiff clay. In Offshore Technology Conference. Offshore Technology Conference.

64. Robertson, A.N., Gueydon, S., Bachynski, E., Wang, L., Jonkman, J., Alarcón, D., Amet, E., Beardsell, A., Bonnet, P., Boudet, B. and Brun, C., 2020, September. OC6 Phase I: Investigating the underprediction of low-frequency hydrodynamic loads and responses of a floating wind turbine. In Journal of Physics: Conference Series (Vol. 1618, No. 3, p. 032033). IOP Publishing.

65. Robertson, A., Jonkman, J., Vorpahl, F., Popko, W., Qvist, J., Frøyd, L., Chen, X., Azcona, J., Uzunoglu, E., Guedes Soares, C. and Luan, C., 2014, June. Offshore code comparison collaboration continuation within IEA wind task 30: phase II results regarding a floating semisubmersible wind system. In International Conference on Offshore Mechanics and Arctic Engineering (Vol. 45547, p. V09BT09A012). American Society of Mechanical Engineers

66. Smith R., Carr, T., and Lane, M., "Computational Tool for the Dynamic Analysis of Flexible Riser Incorporating Bending Hysteresis", Paper accepted for presentation at the 26th International Conference on Offshore Mechanics and Arctic Engineering (OMAE), San Diego, USA, June 2007.

67. Torsethaugen K. and Haver, S., "Simplified double peak spectral model for ocean waves", Proceedings of the Fourteenth International Offshore and Polar Engineering Conference, Toulon, France, May 2004.

68. Viselli, A.M., Goupee, A.J., Dagher, H.J. and Allen, C.K., 2016. Design and model confirmation of the intermediate scale VolturnUS floating wind turbine subjected to its extreme design conditions offshore Maine. Wind Energy, 19(6), pp.1161-1177.

69. Vorpahl, F., Popko, W. and Kaufer, D., 2011. Description of a basic model of the" UpWind reference jacket" for code comparison in the OC4 project under IEA Wind Annex XXX. Fraunhofer Institute for Wind Energy and Energy System Technology (IWES), Germany.

70. Williams, D. and Kenny, F., 2017, June. Calculation of VIV Fatigue of Multi-Pipe Systems. In ASME 2017 36th International Conference on Ocean, Offshore and Arctic Engineering (pp. V05BT04A036-V05BT04A036). American Society of Mechanical Engineers.

71. Wood, W. L., Bossak, M., and Zienkiewicz, O. C, 1981, “An Alpha Modification of Newmark’s Method”, International Journal for Numerical Methods in Engineering, Vol. f5, pp. 1562-1566.

1.13 3rd Party Software & External Libraries

Flexcom 2022.1.1 (August 2022) uses third-party software components:

- Intel® Math Kernel library (computing math library of highly optimised and extensively parallelised routines)
- OpenFAST source code and documentation (open-source wind turbine simulation tool)
- TurbSim software (open-source wind field turbulence modelling tool)
- EPPlus library (Excel spreadsheet library for .NET Framework/Core)

Index

- * -

- H -

- N -

- S -