Riflex 4.21.0 User Guide

Table of Contents

1. Introduction

1.1. Purpose of Program

`RIFLEX` was developed as a tool for analysis of flexible marine riser systems, but is as well suited for any type of slender structure, such as mooring lines, umbilicals, and also for steel pipelines and conventional risers.

These slender structures may be characterized by:

- Small bending stiffness

- Large deflection

- Large upper end motion excitation
- Nonlinear cross section properties
- Complex cross section structure

Due to the complex cross sections typical found for flexible pipes, a global cross section model is applied in `RIFLEX`. This means that cross section properties such as axial-, bending-, and torsional stiffness must be specified as input. Furthermore, structural response is always computed as global deformations and stress resultants (axial force, moments). Hence, local strains and stresses in different cross section layers and materials are not considered.

Nonlinear cross section behaviour is modelled by introducing nonlinear relations between global deformation parameters and stress resultant, i.e. curvature and moment; relative elongation and tension.

The program computes static and dynamic characteristics of the structure.

Static analysis comprises:

- Equilibrium configuration
- Parameter variations of tension or position parameters, current velocity and direction

Dynamic analysis comprises:

- Eigenvalue analysis, natural frequencies and mode shapes
- Response to harmonic motion and wave excitation
- Response to irregular wave- and motion excitation

The program is based on a nonlinear finite element formulation. The following key features are included:

- Flexible modelling of simple as well as complex systems
- Nonlinear time domain simulation of riser motions and forces
- Nonlinear cross section properties
- Generalized Morison type of load model Simplified analysis options:
    - Static analyses, catenary approximations Linearized time domain simulation
    - Frequency domain analysis

1.2. Program Documentation

The program documentation comprises:

- Theory Manual containing a description of mathematical models used in the program
- User’s Manual containing description of input and output
- Release Notes for each release of the program

1.3. Structure of Computer Program

The program system consists of four programs or modules communicating via the file system as shown in the figure below.

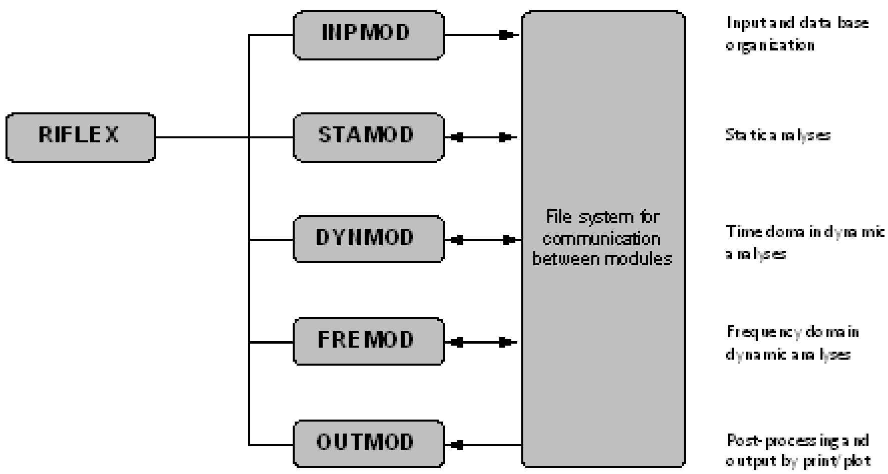

Figure 1. Structure of program system

A complete dynamic analysis must include a run of all modules. However, an efficient data base system simplifies the work during a complete study by storing input data and intermediate results. (I.e. problem description, static configurations, wave induced vessel motions and water particle velocities and accelerations).

Each module will be further detailed in the following.

1.3.1. INPMOD module

The `INPMOD` module reads most input data and organizes a data base for use during subsequent analyses. Once the `INPMOD` module has been run, several analyses can be performed by the other modules without rerun of `INPMOD`.

1.3.2. STAMOD module

The `STAMOD` module performs several types of static analyses. The results may be used directly in parameter studies etc., and are also used to define the initial configuration for a succeeding dynamic analysis. Element mesh, stressfree configuration and key data for finite element analysis are also generated by `STAMOD` based on system data given as input to `INPMOD`.

1.3.3. DYNMOD module

The `DYNMOD` module carries out time domain dynamic analyses based on the final static configuration, environment data and data to define motions applied as forced displacements in the analysis. It is possible to perform several dynamic analyses without rerun of `INPMOD` and `STAMOD`. Response time series are stored on file for further postprocessing by `OUTMOD`. In addition to dynamic response, natural frequencies and modeshapes can be calculated.

1.3.4. OUTMOD module

`OUTMOD` performs postprocessing of selected results generated by `STAMOD` and `DYNMOD`. It is also possible to export time series via a standardized file format for further postprocessing by general purpose statistical analysis program (`STARTIMES`).

1.4. Explanation of Files Used

In running the 5 different `RIFLEX` modules different kinds of files are needed. The files can be divided into the following categories:

1. Symbolic input/output files (i.e. readable ASCII files)
2. Binary files for internal communication between `RIFLEX` modules
3. Files for export of results for postprocessing

An overview of files used is given in the figure File system for communication between modules. A `RIFLEX` user will only need to specify input files for the `INPMOD`, `STAMOD`, `DYNMOD` and `OUTMOD` modules.

The internal file communication is organized via run command procedures and therefore hidden for the user. The file names and extensions may be adapted to the computers operating system and the actual run command procedures used. Description of the file name conventions used in the standard run command procedure supplied with a `RIFLEX` installation is given in How to use Standard Run Command Procedures.

Below is a brief description of the files used.

1.4.1. Symbolic input/output files

Each analysis module needs a symbolic data file to read input data from (extension `.inp`) and one symbolic file to print out major results (extension `.res`). These files are denoted:

- `xxxxxx.inp`: symbolic input file to module `xxxxxx`

xxxxxx.res: symbolic result file from module xxxxxx

xxxxxx here means either INPMOD, STAMOD, DYNMOD or OUTMOD, see the figure File system for communication between modules. Description of data needed in the input files are described in Chapters 5-9 of the User Manual.

1.4.2. Files for internal communication between modules

Files for internal communication are binary, direct access data files in either SAM-DMS format (extension .sam) or in FFILE format (extension .ffi). See RIFLEX maintenance manual for further file format description.

A short description of the files used:

- ifninp.sam: storage of all data given as input to the INPMOD module. System data read by STAMOD for generation of finite element model, wave and transfer function data read by DYNMOD
- ifnsys.sam: contains system finite element model generated by STAMOD
- ifndmp.sam: temporary storage of all system data. To be used in possible restart analysis in STAMOD
- ifnsta.ffi: storage of results from static analysis
- ifndyn.ffi: storage of results from dynamic analysis
- ifnirr.ffi: storage of wave kinematics data for irregular dynamic analysis
- ifnplo.ffi: storage of plot data generated by OUTMOD

1.4.3. Files for communication with external programs

The following files can optionally be applied to export results from RIFLEX for postprocessing by other programs:

- startimes.ts: Export of response time series from OUTMOD for postprocessing by the general purpose statistical analysis program STARTIMES. File format is standard STARTIMES format (binary, direct access file)
- ifrdyn.raf: File for communication with general purpose program for advanced graphical presentation

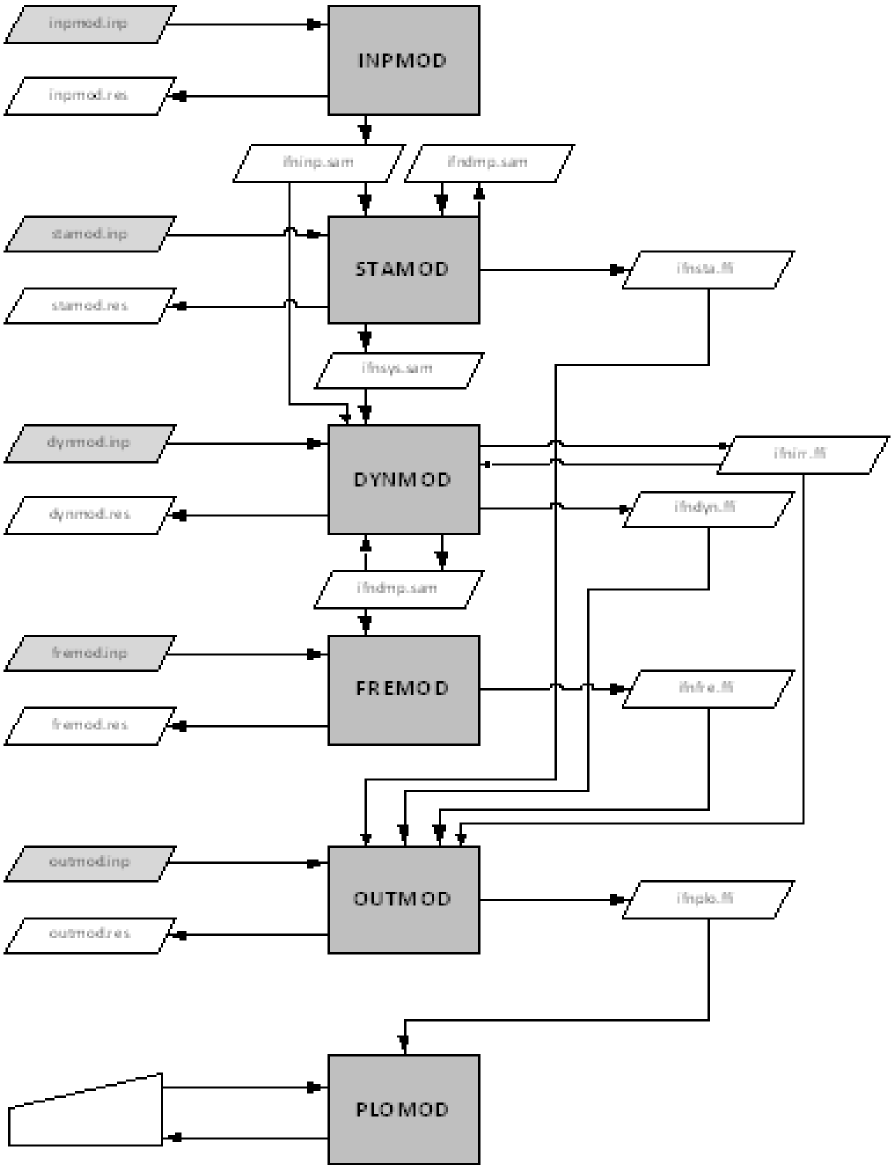

Figure 2. File system for communication between modules

1.5. Applied Units and Physical Constants

Throughout the theory description a *consistent* set of units is used.

In the program input the user is allowed to select mass as well as force units. This implies that the user also has to specify the gravitational constant as the ratio of force to mass units $[\mathrm{F}/\mathrm{M}]$. In order to allow inconsistent units, e.g. kN, kg, m, s, the acceleration in terms of F/M ratio will be different from acceleration in terms of the ratio length to squared time $[\mathrm{L}/\mathrm{T}^2]$. A constant `GCONS` is therefore introduced as a specification of the difference $\texttt{GCONS} = \frac{\mathrm{F}/\mathrm{M}}{\mathrm{L}/\mathrm{T}^2}$. In the example case, `GCONS = 0.001`.

Physical quantity **Modified SI** (kN)	**Symbol**	**Units, SI**	**Modified SI** (kN)
Basic			
Time	T	s	s
Length	L	m	m
Mass	M	kg	kg
Force	$\mathrm{F} = \mathrm{ML}/\mathrm{T}^2$	N	kN
Derived			
Pressure, stress	$\mathrm{P} = \mathrm{F}/\mathrm{L}^2$	N/m^2	kN/m^2
Velocity	$\mathrm{V} = \mathrm{L}/\mathrm{T}$	m/s	m/s
Physical constants			
Acceleration of gravity	`G` $[\mathrm{F}/\mathrm{M}]$	$9.81\,\mathrm{N}/\mathrm{kg}$	$0.00981\,\mathrm{kN}/\mathrm{kg}$
Acceleration of gravity	`G` $[\mathrm{L}/\mathrm{T}^2]$	$9.81\,\mathrm{m}/\mathrm{s}^2$	$9.81\,\mathrm{m}/\mathrm{s}^2$
Consistency of units	`GCONS`	`1`	`0.001`
Density of sea water	`WATDEN` $[\mathrm{M}/\mathrm{L}^3]$	$1025\,\mathrm{kg}/\mathrm{m}^3$	$1025\,\mathrm{kg}/\mathrm{m}^3$

This section will give an introduction to the terminology and principles for system modelling in RIFLEX.

The system definition starts with definition of the topology and proceeds in increasing detail to the line and component descriptions. It is possible to specify a system with general topology (Arbitrary Riser System, AR), but several alternatives are also available for simplified input of commonly used configurations with well defined standard topologies (e.g. standard system SA, SB, SD, SC). The line- and component specifications will, in most cases, be independent of the system topology (see the figure `System definition INPMOD' below). There are, however, a few component types available for the arbitrary system that can not be used in standard systems, see Component Description.

2.1.1. System Topology Description

The system topology is in general described in terms of branching points and terminal points. These points are denoted supernodes. Supernodes are connected by simple lines. This means that the system topology is uniquely determined by the connectivity between a number of defined supernodes and lines.

A general supernode/line connectivity can be specified for arbitrary systems while a restricted system specific connectivity is available for standard systems.

2.1.2. Boundary Condition Modelling at Supernodes

Supernodes are classified as free, fixed or prescribed depending on their boundary condition modelling. A supernode is denoted free if all degrees of freedom are free (i.e. nodal position and rotations are unknown prior to the analysis). In modelling of standard systems it is further distinguished between free branchings (TSNBRA) and free ends (TSNFRE) to ease system topology description.

Supernodes of type fixed are used for modelling supports at fixed structures, seafloor connection, etc. A supernode is denoted fixed if one or several degrees of freedom (dof’s) are fixed. For arbitrary systems it is possible to specify status code free/fixed for all degrees of freedom for each supernode of type fixed (i.e. status code specifications for global x, y, z translations and rotations). For standard systems, all dof’s of fixed supernodes are assumed fixed (rotation free support can still be specified using the ``connector'' component).

Prescribed supernodes are normally used for modelling supports with forced (prescribed) dynamic motions (e.g. connections to floating support vessels).

The interpretations/specifications are similar to fixed supernodes.

System definition, INPMOD

2.1.3. Specification of Supernode Positions for Stressfree and Final Configurations

The basis for calculation of structural forces and deformations in finite element analysis is a stressfree reference configuration defining the state of no structural forces/deformations. A stressfree configuration for structural parts with bending stiffness and no initial deformations will always be a straight line.

The stressfree configuration for arbitrary systems is therefore specified on system level by specification of stressfree positions (global x, y and z-coordinates) of all supernodes. Stressfree position of intermediate `FEM` nodes are then computed by the program assuming a straight line configuration between stressfree supernode positions. The stressfree configurations for standard systems are automatically generated by the program, see Standard Systems for a description. Output of generated stressfree configurations are optionally available in `STAMOD`.

Final static position of relevant dof's for fixed and prescribed supernodes are specified as a part of the system description.

2.1.4. Line and Segment Description

A line is a linear structural element between two supernodes which is identified by a line type number. This means that a line type can be referred to several times in the system topology description, which is convenient for modelling of systems with several identical lines (e.g. anchor systems).

2.1.4.1. The line is specified in terms of:

- Sequence of segments with homogeneous cross sectional properties. Cross sectional component type, length and number of elements to be used for finite element discretization are specified for each segment (see `System definition terms' (below)).
- Nodal components for modeling of clump weights, buoys, swivels and hinges etc. can be specified at segment intersections.
- Fluid component (`FLUID`) for description of possible internal fluid flow.

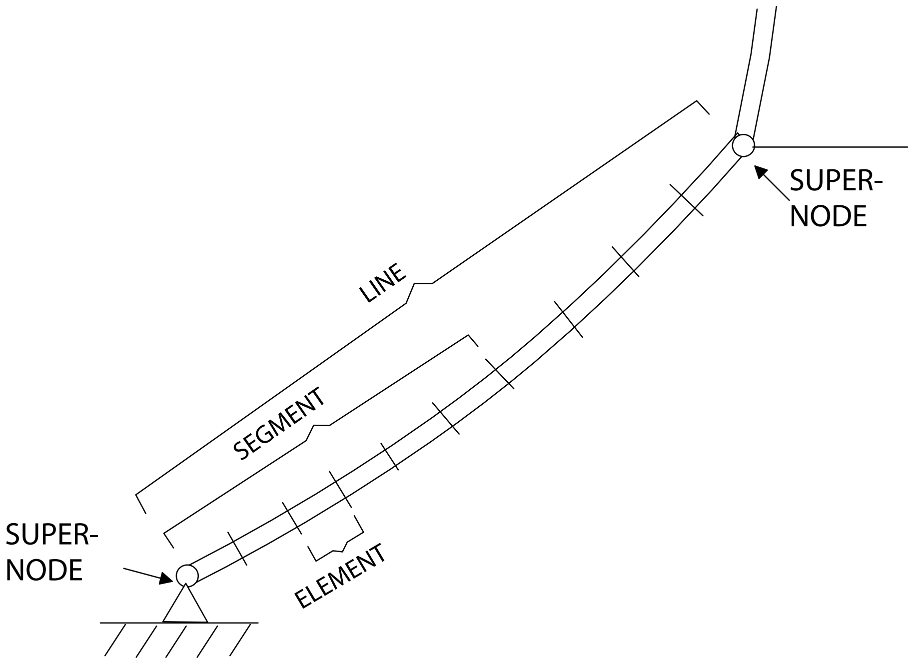

Figure 3. System definition terms

- `SUPERNODE`: Branching points or nodes with specified boundary conditions.
- `LINE`: Suspended structure between two supernodes.
- `SEGMENT`: (Part of) line with uniform cross section properties and element length.
- `ELEMENT`: Finite element unit.

2.1.5. Component Description

The components represents the elementary description of the mechanical properties. A component is identified by a numerical identifier called component type number.

The components available in present `RIFLEX` version are:

1. Cross sectional components
    - Pipe cross section (`CRS0`)
    - Axi-symmetric cross section (`CRS1`)
    - Bi-symmetric cross section (`CRS2`)
    - Cross section for advanced modelling of floating, partly submerged structures, either axi-symmetric or bi-symmetric (`CRS3`, `CRS4`, `CRS5`). Only available for ``arbitrary" systems.
    - General non-symmetric cross section (`CRS7`)

Cross sectional stiffness properties are specified in terms of axial and, optionally, bending and torsional stiffness. Elements specified with axial stiffness only are represented by 3D bar elements. Elements with specified bending and torsional stiffness are represented by 3D beam elements. Linear or nonlinear stiffness specifications can be applied for all cross sectional types.

Additional data that must be specified for all cross sectional types are external and internal area, mass and hydrodynamical coefficients.

A special component denoted external wrapping (`EXT1`) is also available for modelling additional distributed weight or buoyancy.

2. Nodal components

    - Body (`BODY`) for modelling of clump weight, submerged buoys etc.

    - Ball joint connector (`CONB`) for modelling of swivels, hinges etc.

Mass, volume and hydrodynamical coefficients must be specified for both component types.

3. Special components

    - Rollers for description of elastic contact forces between lines.

    - Tensioner component for modelling of tensioner mechanisms.

2.1.6. Element Mesh Generation

The element mesh is computed automatically based on the topology, line and component description. Constant element lengths are applied within segments. Connections between lines, segments and elements specified as input and nodal/element numbers used in the finite element analysis are available as output from `STAMOD`.

2.2. Standard Systems

2.2.1. Classification

In order to simplify the system topology definition for commonly used configurations, a selection of standard systems are provided:

- SA - Seafloor to surface vessel. One point seafloor contact. The Steep Wave, Steep S and Jumper flexible riser configurations are special cases of the SA system.

- SB - Seafloor to surface vessel. Seafloor tangent and/or additional seafloor attachment point. The Lazy Wave and Lazy S flexible riser configurations are special cases of the SB system. The SB system is also convenient for modelling of anchorlines with seafloor contact at lower end.

SC - Free lower end. Riser during installation etc.

- SD - Free upper end. Buoyed riser, loading system, etc.

The stressfree configurations are automatically generated for all standard systems. Definition of system topologies and stressfree configurations are further discussed in the remaining sections of this chapter (`` `SA'' Seafloor to Surface Vessel, One-Point Seafloor Contact' to `` `SD'' Free Upper End').

2.2.1.1. Global coordinate systems

The x-y plane of the global coordinate system is placed at the sea surface with the z-axis pointing upwards.

The following conventions are in addition adopted for the standard riser systems:

- Boundary conditions, i.e. terminal point coordinates are specified in x-z plane
- x-coordinate at lower end is zero for SA, SB and SD systems
- x-coordinate at upper end is zero for SC systems

The global coordinate systems for all standard systems are shown in figures presented in the remaining sections of this chapter (`` `SA'' Seafloor to Surface Vessel, One-Point Seafloor Contact' to `` `SD'' Free Upper End').

2.2.1.2. Special analysis features

An important feature available for standard systems is simplified static analysis based on catenary analysis. It is also possible to use the catenary solution as starting point for the static finite element analysis or to apply conventional finite element analysis starting from stressfree position.

For further details, see `Static Catenary Analysis' and `Static Finite Element Analysis' in the Theory Manual.

2.2.2. ``SA'' Seafloor to Surface Vessel, One-Point Seafloor Contact

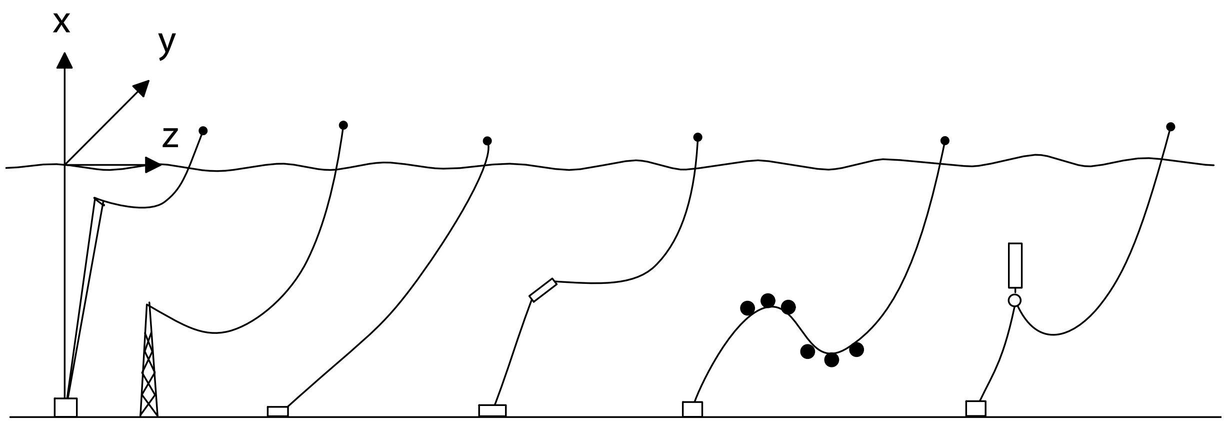

Figure 4. Examples of configurations covered by SA

2.2.2.1. System topology

The riser is suspended between two defined points. The lower end is fixed while upper end is connected to the support vessel. The only type of branching elements are slender buoyancy or weight elements suspended in one-point attachment. Only one branch is accepted per branch node. The branches are thus assumed to be vertical in a zero current condition.

2.2.2.2. Stressfree configuration

The stressfree configuration is placed horizontally at seafloor, branches are assumed vertical.

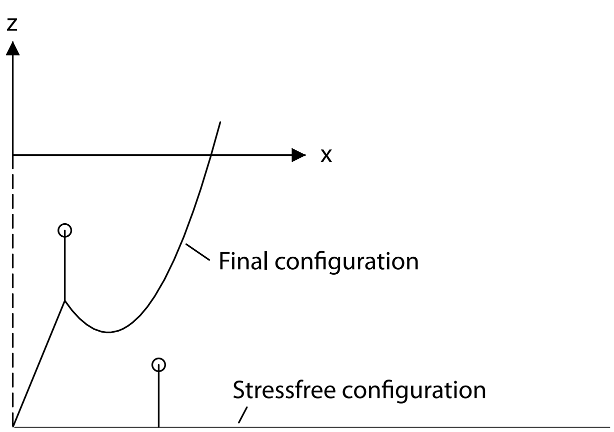

Figure 5. Example of stressfree configuration for SA system

2.2.3. ``SB'' Seafloor to Surface Vessel, Seafloor Tangent

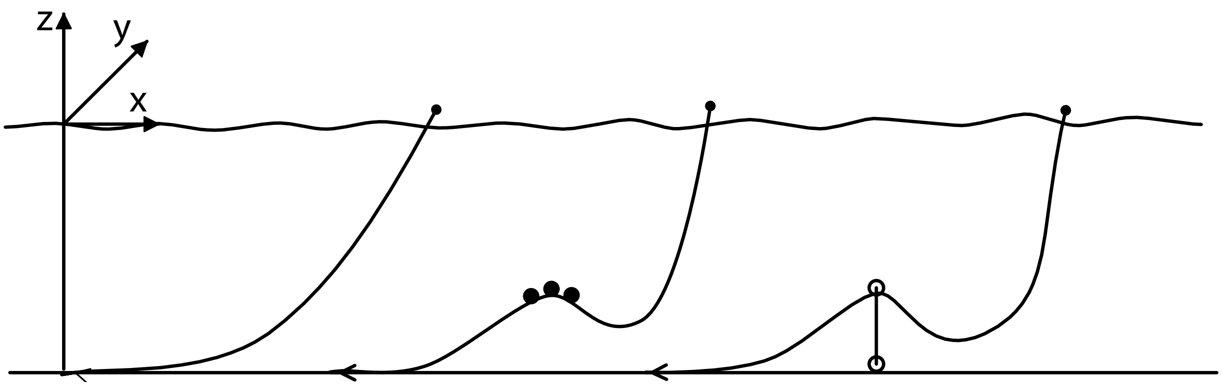

Figure 6. Examples of configurations covered by SB

2.2.3.1. System topology

Compared with the previous systems this system includes additional features:

- Seafloor tangent boundary condition
- Buoyancy guide at one point

The seafloor contact is modelled by bilinear stiffness. The stiffness is discretized and implemented as springs at the nodal points that may touch the seafloor.

2.2.3.2. Stressfree configuration

The stressfree configuration is placed horizontally. The vertical position is placed above the seafloor to avoid possible seafloor contact at the first steps in the incremental loading sequence applied in the static finite element analysis. Possible branches are assumed vertical.

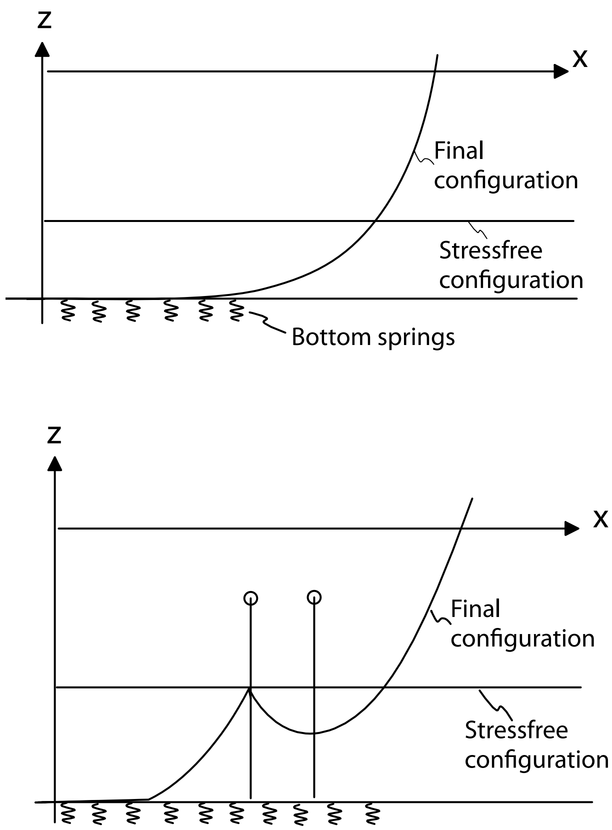

Figure 7. Examples of stressfree configuration for SB systems

2.2.4. ``SC'' Free Lower End, Suspended from Surface Vessel

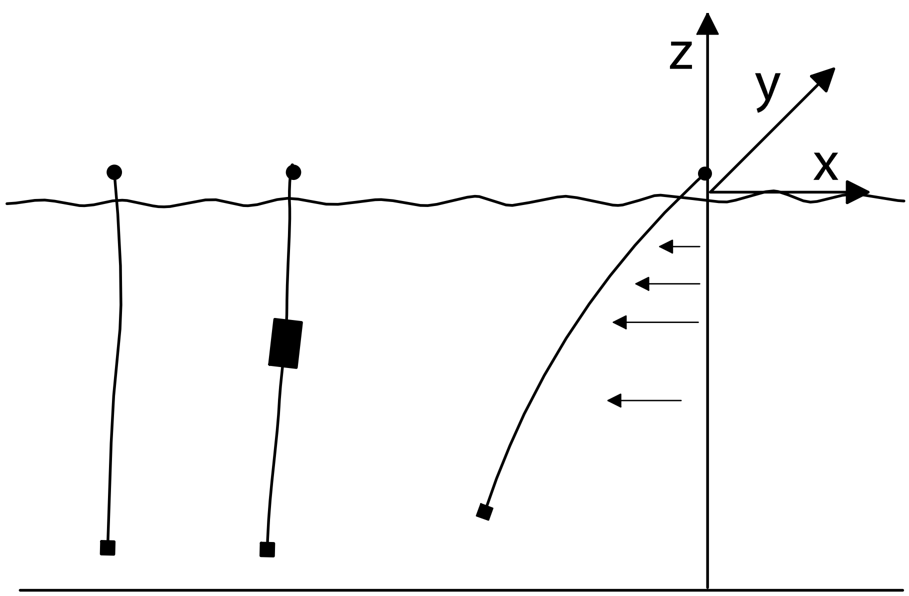

Figure 8. Examples of configurations covered by SC

2.2.4.1. System topology

This group is characterized by a free lower end, all degrees of freedom being specified at the upper end. This configuration represents typical installation phases, but as indicated in the figure, towing configurations can be analyzed as well.

2.2.4.2. Stressfree configuration

The stressfree configuration is assumed vertical with vertical position of upper end equal to final position.

2.2.5. ``SD'' Free Upper End

2.2.5.1. System topology

Single line system connected to seafloor at lower end and with free upper end.

2.2.5.2. Stressfree configuration

The stressfree configuration is assumed vertical with lower end in final position (e.g. at seafloor).

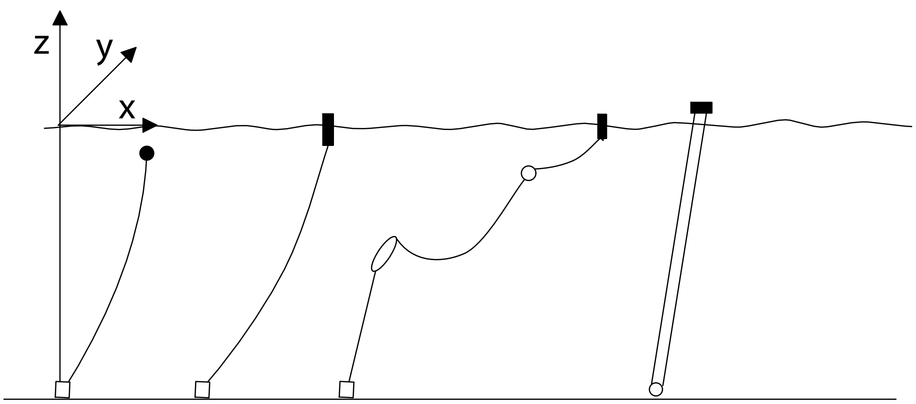

Figure 9. Examples of configurations covered by SD

With a free upper end the configuration is governed by hydrodynamic forces in the horizontal direction. If the buoyancy element is surface-piercing, it is assumed that it is a long, slender, spar-type buoy.

3. Main Types of Analysis

3.1. General

Four main types of analysis are available in `RIFLEX`: 1. Static analyses. 2. Static parameter variation analyses. 3. Dynamic time domain analysis including eigenvalue analysis. 4. Frequency domain analysis.

The method of analysis is based on finite element technique which has proved to be a powerful tool for several applications. The co-rotated finite element formulation applied in `RIFLEX` allows for unlimited translations and rotations in 3D-space.

The static analysis is based on a complete non-linear formulation. However, a pre-processor based on catenary theory is also implemented. The reason for this is to reduce computing time by giving the nonlinear iteration a good starting point, but also to analyse simple problems without use of the finite element method.

Time domain analysis is based on step by step numerical integration of the dynamic equilibrium equations. It is possible to apply a complete nonlinear method based on the incremental dynamic equilibrium equations, or alternatively, a linearized approach by linearization of mass, damping and stiffness matrices at the static equilibrium position. Nonlinear hydrodynamic loading is, however, included in the linearized time domain analysis.

The frequency domain analysis is based on the linearized dynamic equilibrium equation at static equilibrium position by application of stochastic linearization of the hydrodynamic loading.

All analyses are three-dimensional.

The mathematical models are described in detail in the Theory Manual.

3.2. Static Analysis

This is the elementary mode of analysis and is used for establishing pipe configuration for a specified set of conditions.

###The computations include: 1. Establishment of initial configurations based on catenary approximation. 2. Iteration for equilibrium position by incremental reduction of unbalanced forces. (Newton-Raphson iteration) by application of nonlinear finite element analysis.

Step 1 is optional and may be replaced by a zero-load initial configuration.

Snap-through behaviour and multiple equilibrium configurations can be discovered by incremental static analysis from different initial positions. The program will be able to discover the appearance of kinks. However, a detailed study of kinks including contact forces between pipe elements is not included in the analysis.

###Basic results are: - Nodal point coordinates - Curvature at nodal points - Axial force - Bending moment - Shear force - Torsion

The bending moments and the torsion moment are calculated about the area center and the shear center, respectively. Note that all results refer to the element coordinate system. The results are available as print (tables), and stored on file for post processing and graphic presentation.

3.3. Static Parameter Variation Analysis

The purpose of these analyses is to study the influence of varying key parameters in the system. Key problems: - Establish static stiffness characteristics in order to specify support vessel requirements with regard to position-keeping. - Clarify sensitivity to support vessel position, external force, or current variations.

For this purpose the following analyses are available: 1. Stepwise increment supernode position in any direction. 2. Stepwise increment of support vessel position. 3. Stepwise increment of current velocity or direction. 4. Stepwise increment force components.

Combinations of above basic cases are also possible.

3.3.1. Results

The same results as for the basic static analyses are available, but the main output consists of a set of key parameters to be presented as function of the varied parameter. E.g. tension or curvature at selected locations as function of position.

3.4. Time Domain Dynamic Response Analysis

The purpose of these analyses is to study the influence of support vessel motions as well as of direct wave induced loads on the system.

A static analysis to define equilibrium condition is assumed to be carried out before starting a dynamic analysis. The last step of a parameter variation analysis can also be used as starting point for a dynamic analysis.

The following types of dynamic analyses are included:

1. Eigenvalue analysis
2. Harmonic (periodic) excitation
    - Forced displacements (harmonic) at one or more specified nodes
    - Regular waves
3. Irregular excitation
    - Stochastic, stationary excitation due to support vessel motions and irregular waves
    - Transient excitation. Special options available to simulate release or rupture, slug flow, time dependent current and external force variations

The mathematical models used in these analyses are described in the Theory Manual.

3.4.1. Results

The basic results from the eigenvalue analysis will be the system's eigenfrequencies and eigenvectors. The basic result format from the dynamic excitation analysis will be as time series of a selected, limited number of response parameters: - Nodal point coordinates - Axial force - Shear force - Curvature - Bending moment - Torsion

The bending moments and the torsion moment are calculated about the area center and the shear center, respectively. Note that all results refer to the element coordinate system. The parameters may be given as total values or as difference from static equilibrium condition. In addition, the total system configuration may be stored for a limited number of time steps.

3.4.1.1. Wind Turbine Results

An additional output file is created for analyses which include a wind turbine. A columnwise description of the outputs is given in the witurb key file. Several wind-turbine-specific coordinate systems are defined in order to present the results. - Shaft system $(\mathrm{XS}, \mathrm{YS}, \mathrm{ZS})$: Follows the non-rotating shaft element. Wind output, azimuth, and out-of-plane (OoP) tip deflection follow this system. - Rotor system $(\mathrm{XR}, \mathrm{YR}, \mathrm{ZR})$: Follows the rotating shaft element and 1st blade. The blade tip in-plane (IP) deflection follows this system.

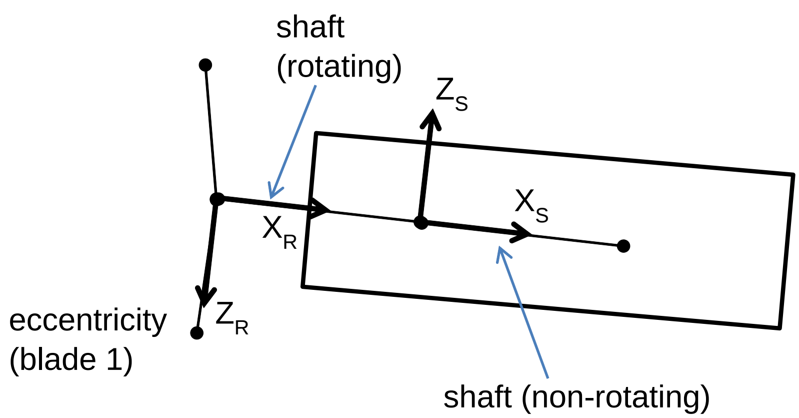

Figure 10. Rotor and shaft coordinate systems

Out-of-plane deflection

In-plane deflection

3.5. Result Post Processing and Graphic Presentation

Results from static and dynamic analyses are stored on file for subsequent post processing and graphic presentation. An overview of main types of output is given in the following:

1. Output from static analysis:

    - 2D and 3D plot of system geometry

    - 2D plot of projected line geometries

- Plot of force variation along lines
- Print of forces, coordinates and element projection angles, optionally element by element, segment by segment or line by line (direct output from static analysis)
- Calculation/graphic presentation of pipe wall force (i.e. axial force including hydrostatic pressure)

2. Output from static parameter variation analysis:
    - Print/plot of selected response quantities during parameter variation
    - Plot of system geometries during parameter variation
3. Output from dynamic time domain analysis:
    - Computation of time series derived from basic response quantities
        - calculation of curvature from nodal coordinates
        - calculation of support forces
        - wall force calculation (e.g. axial force including effects from internal and external hydrostatic pressure)
        - element angle calculation (e.g. angle between elements, vessel and element and global axis and elements)
        - calculation of distance time series (e.g. clearance between lines, vessel and lines, global axis and lines)
    - calculation of velocities and accelerations from wave and vessel motion time series
    - Plot/print of response time series
    - Statistical time series analysis (e.g. estimation of spectral densities, probabilistic distribution for maxima/minima, sample moments, spectral moment, etc.)
    - Animation of the dynamic behaviour of the complete system including support vessel and exciting waves
    - Graphic presentation of vessel motion transfer functions
    - Envelope curves for displacements, curvature and forces showing static value, mean value and response range

In addition to the post processing features available in `RIFLEX` it is also possible to export results via standardized file formats to general purpose statistical analysis programs (e.g. `STARTIMES`) and a advanced graphical presentation/animation tools (e.g. `GLVIEW`).

4. How to Run the Program

4.1. How to give Input - General Rules

4.1.1. Organization of Input Data

The input data are organized in groups, each group starting with a data group identifier, and terminating on the next data group identifier.

A data group identifier may consist of one, two or three words, which give a unique description of the data group. Each word may consist of several characters, but only the first four characters of each word are checked. The user may, of course, write the words completely to increase readability.

In this manual each input line is presented within a standard list, and each list represents either one single input line or a sequence of similar input lines. The input description list is presented in The Input Description List.

4.1.2. Formats

Nearly all input data are read by FREAD /1/, a FORTRAN free-format reader and decoder. This means that the data items may be written anywhere on the input line, as long as the specified order is satisfied. The data items must be separated by at least one blank (exceeding blanks are ignored). Note that since blank is exclusively interpreted as a delimiter, blank can not be used to specify a zero value.

The line length is currently limited to 260 characters. Note that a RIFLEX input line may be split into several lines on the input file, see Continuation of Input Statements.

4.1.2.1. Important:

All digits, letters and/or special symbols in a data item must be given consecutively without blanks.

4.1.2.2. Comments

Input lines with an apostrophe (`'`) in the first column are interpreted as comments, and simply ignored. Comments may be inserted anywhere in the input data stream in order to improve the readability of the input data.

4.1.2.2.1. Example:

```
'THIS IS A COMMENT
'NOTE! COMMENTS ARE IGNORED BY THE PROGRAM
```

4.1.2.3. Alphanumeric Data Items

In this manual, alphanumeric data items are denoted as character with a maxim length given in parentheses; e. g. character(8) which has a maximum length of 8. If the input item is longer than the maximum length, all characters in excess of this are simply ignored.

An alphanumeric data item may consist of one or more characters. The first character is always a letter (A-Z), while the remaining ones may be letters, digits or special symbols (except /, $, & and blank).

4.1.2.3.1. Example:

```
'WITH CHARACTER(4)
ENVIRONMENTAL DATA
'WILL BE INTERPRETED AS
ENVI DATA
'WITH CHARACTER(6)
ENVIRONMENTAL DATA
'WILL BE INTERPRETED AS
ENVIRO DATA
```

When file names are input, the maximum length of the file name is given in the manual.

4.1.2.3.2. Example:

```
CHFTRA
```

- `CHFTRA: character(80)`: File name with transfer functions data

It is recommended that the combined length of the path and file name does not exceed 256 characters.

4.1.2.4. Integer Data Items (denoted integer)

All characters must be digits. The first digit may be preceded by a + or a - character.

4.1.2.4.1. Example:

```
0 1 -27 +66
```

4.1.2.5. Real Number Data Items (denoted real)

A real number data entry may consist of up to 3 components, i.e. an integer part i, a decimal part d, and exponent part e. The following 4 basic forms are accepted:

```
(+)i (+)i. (+)i.d (+).d
```

These may all be combined with exponent parts yielding the forms:

```
(+)iE(+)e (+)i.E(+)e (+)i.dE(+)e (+).dE(+)e
```

4.1.2.5.1. Example:

```
0 -1. -0.2E14 +17.E-3 1.78E+3
```

4.1.2.6. Default Data Items

Some of the input data have default values, that is, values to be inserted when no other values are given. There are two ways to signal that the program shall insert default values. One, that may always be used, is to type a slash (/) for the actual data item. Second, if the actual data item (or data items) is (are) the last one(s) on the input line, no slash(es) is (are) necessary. FREAD will insert the number of slashes needed. Note that the slash is also a data item, and must be separated from adjacent data items by one or more blanks.

4.1.2.6.1. Example:

```
'THE DEFAULT VALUE OF THE SECOND DATA ITEM IS 2. THEN
3 / 5
'WILL BE INTERPRETED AS
3 2 5
'6 DATA ITEMS ARE READ AND THE LAST 3 DATA ITEMS HAVE DEFAULT
VALUES 6, 7 AND 8, THEN
1 2 3
'WILL BE INTERPRETED AS
1 2 3 6 7 8
```

4.1.2.7. Continuation of Input Statements

An input statement may consist of one or several input lines. If an input statement consists of more than one input line, all its input lines except the last one must be terminated by the character &, which must be preceded by at least one blank. This implies that on such lines data cannot be specified beyond column 258.

4.1.2.7.1. Example:

```
1 2 &
3
'IS THE SAME AS GIVING
1 2 3
```

4.1.2.8. The Input Description List

```
<parameter names, in the order decoded>
```

`Parameter: type, default: Default value`: Description $[\mathrm{unit}]$

  - Optional additional information for the given parameter.

Comments, notes etc.

If there is no default value, that part of the line is skipped. The parameter types are integer, real and character.

If unit is not applicable, e.g. for integer options, that part of the line is skipped.

4.1.2.8.1. Example:

```
IKS DAMP
```

- `IKS: integer`: Stiffness code 1
  - 1 : Constant spring compression stiffness
  - N : Table with N pairs of pressure force - displacements to be specified
    - N > 2
- `DAMP: real, default: 0`: Dash pot damping coefficient $[\mathrm{FT/L}]$

4.1.2.9. Data group Identifier

As mentioned in Organization of Input Data, the input data are organized in groups.

A data group identifier is given on the first input line of each group.

```
IDW1 IDW2 IDW3
```

- `IDW1: character(4)`: First word of the identifier
- `IDW2: character(4)`: Second word of the identifier
- `IDW3: character(4)`: Third word of the identifier

Data group identifiers may consist of one, two or three words

In the input description, data group identifiers are not presented within the standard list, but the words of the identifiers are written completely in the code box above. With format character(4), only the first four characters are checked, and these parts of the words are marked.

4.1.2.9.1. Example:

```
ROTAtion STIFfness CHARacteristics
```

may be given as

```
ROTA STIF CHAR
```

4.2. How to run RIFLEX with the riflex.bat batch file

This section will give a description of how to run RIFLEX modules using the `riflex.bat` batch file delivered as a part of a `RIFLEX` installation. The `riflex.bat` file is used for running the batch oriented modules `INPMOD`, `STAMOD`, `DYNMOD` and `OUTMOD`.

The file name convention used consists of a prefix in addition to the basic file names described in Explanation of Files Used. The prefix and the basic file names are separated by an underscore.

A common prefix may be used for all the files or two prefixes may be used; prefix1 for identification of `INPMOD` and `STAMOD` files and prefix2 for identification of `DYNMOD` and `OUTMOD` files. If prefix2 is not given, it will be set to prefix1.

The necessary commands to run `RIFLEX` modules `INPMOD`, `STAMOD`, `DYNMOD` and `OUTMOD` with a common prefix are:

```
riflex inpmod prefix1
riflex stamod prefix1
riflex dynmod prefix1
riflex outmod prefix1
```

or alternatively with two prefixes: ~ riflex inpmod prefix1 riflex stamod prefix1 riflex dynmod prefix1 prefix2 riflex outmod prefix1 prefix2 ~

4.3. How to run RIFLEX with the bash script vrr

This section will give a description of how to run `RIFLEX` modules using the `vrr` bash script. Bash scripts may be run on Linux or Unix systems or in a bash shell under Windows; e.g. Git bash, Cygwin.

The `vrr` bash script is delivered as a part of a `RIFLEX` installation and may be used to run several modules in succession, stopping if errors are detected. The necessary command to run the `RIFLEX` modules `INPMOD`, `STAMOD`, `DYNMOD` and `OUTMOD` with a common prefix is:

```
/full/path/to/Riflex/bin/vrr isdo prefix1
```

4.4. RIFLEX files

The file names used by the different modules are summarized in the following. For description of the files used for internal communication, see Explanation of Files Used.

4.4.1. INPMOD files:

- `prefix1_inpmod.inp` - ASCII input file
- `prefix1_inpmod.res` - ASCII result file .
- `prefix1_ifninp.sam` - internal communication file

4.4.2. STAMOD files:

- `prefix1_stamod.inp` - ASCII input file
- `prefix1_stamod.res` - ASCII result file
- `prefix1_stamod.mpf` - ASCII file with key static results .
- `prefix1_ifnsys.sam` - internal communication file
- `prefix1_ifnsta.ffi` - internal communication file
- `prefix1_ifndmp.sam` - internal communication file

4.4.3. DYNMOD files:

- `prefix2_dynmod.inp` - ASCII input file
- `prefix2_dynmod.res` - ASCII result file
- `prefix2_dynmod.mpf` - ASCII file with key dynamic results .
- `prefix2_ifnirr.sam` - internal communication file
- `prefix2_ifndyn.ffi` - internal communication file
- `prefix2_ifndmp.sam` - internal communication file

4.4.4. OUTMOD files:

- `prefix2_outmod.inp` - ASCII input file
- `prefix2_outmod.res` - ASCII result file
- `prefix2_outmod.mpf` - ASCII file with some key `OUTMOD` results; e.g. stresses.
- `prefix2_ifnplo.ffi` - plot file, to be referred in `PLOMOD`

The user input is given on files with extension `.inp`, while key results can be found on files with extension `.res`.

4.5. References

1. Adam, J et al. (1987): FREAD: A FORTRAN free format reader and decoder. SINTEF, Trondheim, Norway.

5. Input to INPMOD

5.1. General Information

The purpose of the `INPMOD` module is to administer reading/interpretation of system and environmental data. The following data groups are included:

- General Control Data - Data Group A
- Single Riser Data - Data Group B
- Component Data - Data Group C
- Environmental Data - Data Group D
- Support Vessel Data - Data Group E

An `INPMOD` run may consist of a combination of the data groups, but a data group included must be complete. Data group A is mandatory, while data groups B-E normally will be given. All data groups except data group A can be given more than once. Note that running `INPMOD` always overwrites old information in the `INPFIL`. That means that all needed data on `INPFIL` must be generated in one `INPMOD` run.

Within each group, the first two data groups (identification and control data) must be given first, while the order of the remaning data groups, to some extent, is arbitrary. Data groups in one section cannot be mixed with data groups in another section. This means that when the program finds a data group identifier it will only check the data within the current section. If one or more errors are found, an error message is written and an error flag is written to the `INPFIL` file. A run with `STAMOD` or `DYNMOD` using this file will then be terminated. `INPMOD` will continue to read data until the `END` input line is found (see Termination of input data) regardless of the number of errors detected.

5.2. Data Group A: General Control Data

General inputs include the unit system.

5.2.1. INPMOD identification text

An identification text describing the `INPMOD` run may be given. If given, this text is printed on the front page of the `INPMOD` print file.

5.2.1.1. Data group identifier, one input line

Mandatory. ~ INPMod IDENtification TEXT CHVERS ~ - `CHVERS: character(8)`: `RIFLEX` input file version, e.g. 3.2

5.2.1.2. Identification text, three input lines

Mandatory. ~ Heading, line no 1 Heading, line no 2 Heading, line no 3 ~

Always three input lines, which may all be blank. Each line may contain up to 60 characters.

5.2.2. Selection of unit system physical constant

The program applies a set of consistent units. The user is free to select any combination of unit names for:

- time $[\mathrm{T}]$
- length $[\mathrm{L}]$
- mass $[\mathrm{M}]$
- force $[\mathrm{F}]$

5.2.2.1. Data group identifier, one input line

Mandatory. ~ UNIT NAME SPECification ~

5.2.2.2. Unit names and gravitational constant

Mandatory. ~ UT UL UM UF GRAV GCONS ~

- `UT: character(6), default: "s"`: Unit name for time
- `UL: character(6), default: "m"`: Unit name for length
- `UM: character(6), default: "kg"`: Unit name for mass
- `UF: character(6), default: "kN"`: Unit name for force
- `GRAV: real > 0, default: 9.81`: Numerical value of gravitational constant $[\mathrm{L}/\mathrm{T}^2]$
- `GCONS: real > 0, default: 0.001`: Consistency acceleration parameter.
  -

`GCONS` = ACC[UF/UM]/ACC - where ACC is the numerical value of acceleration with the force and mass units, and with the length and time units, respectively. - With a consistent set of units, `GCONS = 1.0`.

Example: For the default units, the acceleration of gravity is $9.81 \times 10^{-3}\,\mathrm{kN/kg}$ or $9.81\,\mathrm{m/s}^2$ and `GCONS` may be calculated as $9.81 \times 10^{-3}/9.81 = 0.001$.

In the case of `GCONS` $\neq 1$, the calculation of drag force coefficients must be multiplied by `GCONS`, for example `CDY = GCONS` $(\frac{1}{2}\rho C_D D)$, see also Hydrodynamic force coefficients (@ref inpmod_c_crs1_hydrodynamic).

The unit names are printed as an output heading as shown below.

```
+---------------------------------------------------------+
!     CONSISTENT UNITS USED THROUGHOUT THE ANALYSES       !
!---------------------------------------------------------!
!     NAME FOR TIME   :  T = s                            !
!     NAME FOR LENGTH :  L = m                            !
!     NAME FOR MASS   :  M = kg                           !
!     NAME FOR FORCE  :  F = kN                           !
!                                                         !
!     GRAVITATIONAL CONSTANT :                            !
!     G     =  9.810     L/T**2                           !
!     CONSISTENCY ACCELERATION PARAMETER :                !
!     GCONS = 0.1000E-02 (F/M)/(L/T**2)                   !
+---------------------------------------------------------+
```

5.2.3. Termination of input data

As mentioned an `INPMOD` run consist of some of the data groups A, B, C, D, E and F. No termination is to be provided between the data groups, but after the last data group is given, the following input line must be given to terminate the input stream:

```
END
```

5.3. Data Group B: Single Riser Data

The riser description consists of the data group Riser type specification (@ref inpmod_data_group_b_riser_type_specification) followed by one of Standard system SA (@ref inpmod_b_SA), Standard system SB, Standard system SC, Standard system SD or Arbitrary system AR with the topology and boundary conditions.

The data group Line and segment specification is used for the specification of the line types that are used in topology specification.

5.3.1. Riser type specification

5.3.1.1. Data group identifier, one input line

```
NEW SINGle RISEr
```

5.3.1.2. Selection of riser type and identifier

```
ATYPS IDRIS IDCON
```

- `ATYPS: character(6)`: Riser code type for single riser. Allowable codes are listed below: - SA: Next data in Standard system SA (@ref inpmod_b_SA) - SB: Next data in Standard system SB - SC: Next data in Standard system SC - SD: Next data in Standard system SD - AR: Next data in Arbitrary system AR
- `IDRIS: character(6)`: Riser identifier
- `IDCON: character(6), deault: "NONE"`: Normally not used!
    - =LAYFLX - Roller contact forces are dumped on data files for LAYFLEX post-processing.
    - Static roller contact forces will be output to the file `prefix_consta.asc`.
    - For nonlinear dynamic analysis, time series of roller contact results will be stored on the file prefix_contts.asc and key results will be stored on the file `prefix_condyn.asc`.

One or several riser specifications may be created and stored by one run of INPMOD. The identifier IDRIS is used by STAMOD to select one particular riser for analysis.

5.3.2. Standard system SA

One-point seafloor contact to (surface) vessel. The frequently used `steep wave'',` steep S" and ``jumper" configurations are special cases of the SA system. The initial configuration of this system is two-dimensional in X-Z plane.

5.3.2.1. Topology

The only variable topology feature is the option of including vertical buoyancy or weight elements in the form of branches.

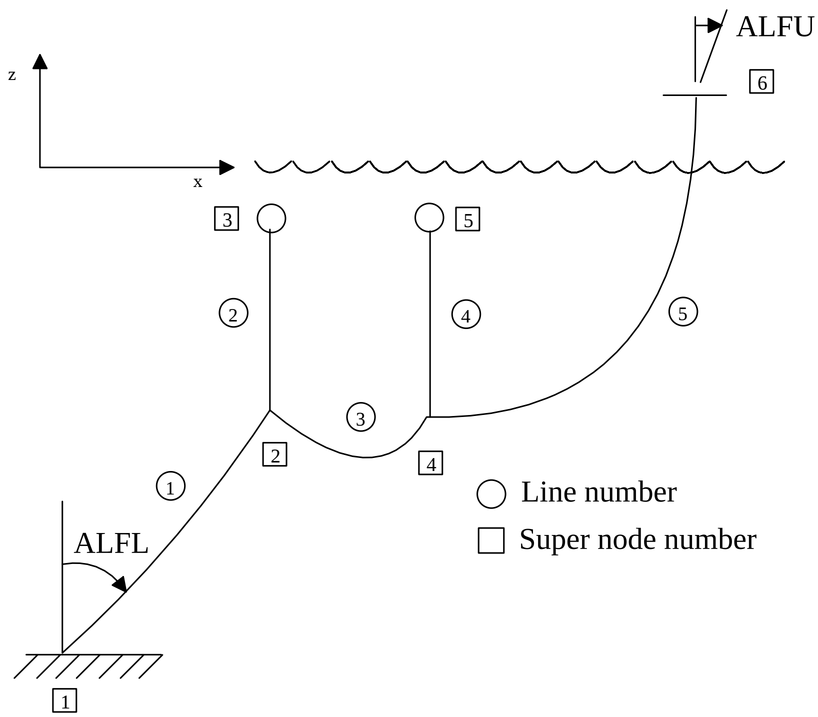

Figure 11. Topology of system SA

Introduction of branching points implies that more than one line has to be specified, see the figure `Topology of system SA' above. Branches are assumed to be vertical in a still water condition. Seafloor and surface contacts are not modelled for this system.

5.3.2.2. Data group identifier

```
SINGle RISEr SA
```

5.3.2.3. Topology

```
NSNOD
```

- `NSNOD: integer`: Number of supernodes

This implies that number of lines to be defined is `NLIN=NSNOD-1`, see the figure `Topology of system SA' above.

5.3.2.4. Line, line type and supernode connectivity

This data group defines the connectivity between lines and supernodes. If the line identifier is missing the line number implicitly defined by the order in which the lines are specified, will be used as the line identifier. References to line type IDs and supernode numbers are mandatory.

The lines must be specified in the order indicated in the figure `Topology of system SA' above. This means that the lines are given continuously from seafloor to upper riser end.

At each branching point the line defining a branch is specified before the next line in the main riser configuration. No ball joint components are accepted in branch lines.

`NLIN` input lines. ~ LINE-ID LINTYP-ID ISNOD1 ISNOD2 ~ - `LINE-ID: character(8)`: Line identifier - `LINTYP-ID: character(8)`: Reference to line type identifier - `ISNOD1: integer`: Reference to supernode number at end 1 - `ISNOD2: integer`: Reference to supernode number at end 2

If only 1 alphanumeric string and 2 integers are specified, the first string is taken as `LINTYP-ID`. ~ LINTYP-ID ISNOD1 ISNOD2 ~ The `LINE-ID` is taken as the line number as implicitly defined by the order in which the lines are given.

5.3.2.5. Boundary conditions, coordinates

`ZL XU ZU ALFL ALFU`

- `ZL: real`: Z coordinate of lower end $[\mathrm{L}]$
  - X and y coordinates of lower end are set equal to zero.
- `XU: real > 0`: X coordinate of upper end $[\mathrm{L}]$.
- `ZU: real`: Z coordinate of upper end $[\mathrm{L}]$.
  - Y coordinate of upper end is equal to zero
  - Z coordinate is positive upwards
  - `ZU=0.0` at still water level (see the figure `Topology of system SA' above).
- `ALFL: real, default: 0.0`: Angle of lower end from vertical $[\mathrm{deg}]$.
- `ALFU: real, default: 0.0`: Angle of upper end from vertical $[\mathrm{deg}]$.

If the lower/upper end later in the specification is allowed to rotate freely around the y-direction, ALFL/ALFU will be dummy.

5.3.2.6. Supernode types

`NSNOD-2` input lines. `ISNOD` must be given in increasing order from 2 to `NSNOD-2`. Only to be given if `NSNOD>2`. ~ ISNOD ITYPSN ~ - `ISNOD: integer`: Supernode ISNOD=2,3,….., NSNOD-1 - `ITYPSN: character(6)`: Type of supernode - `TSNBRA` - Branch point - `TSNFRE` - Free end

Specification of supernodes: Supernode at lower and upper end are not to be specified. Supernode number at lower end is automatically set to 1 and the supernode type is fixed

(`ITYPSN=TSNFIX`). Supernode at upper end is automatically set to `NSNOD` and the supernode type is specified position (`ITYPSN=TSNPOS`) indicating that the upper end is connected to the support vessel.

5.3.2.7. Support vessel reference

`IVES IDWFTR XG YG ZG DIRX`

- `IVES: integer, default: 1`: Vessel number (`IVES` = 1)
- `IDWTFR: character (6), default: 'NONE'`: Identifier for WF motion transfer function
  - `IDWFTR = 'NONE'` means no transfer function specified.
- `XG: real`:X position of vessel coordinate system referred in global system $[L]$
- `YG: real`:Y position of vessel coordinate system referred in global system $[L]$
- `ZG: real`:Z position of vessel coordinate system referred in global system $[L]$
  - Confer Data Group E: Support Vessel Data.
- `DIRX: real`: Direction of vessel X-axis.
  - See [Location of support vessel coordinate system] (@ref Location_of_support_vessel_coordinate_system).

Next data group is Line and segment specification.

5.3.3. Standard system SB

Multiple seafloor contact points are allowed, with upper end connected to the support vessel. The frequently used lazy wave, lazy S and free hanging configurations are special cases of the SB system. The initial configuration of this system is two-dimensional in X-Z plane.

5.3.3.1. Topology

In addition to the branching feature of system SA it is also allowed to specify seafloor tangent and intermediate seafloor anchor point.

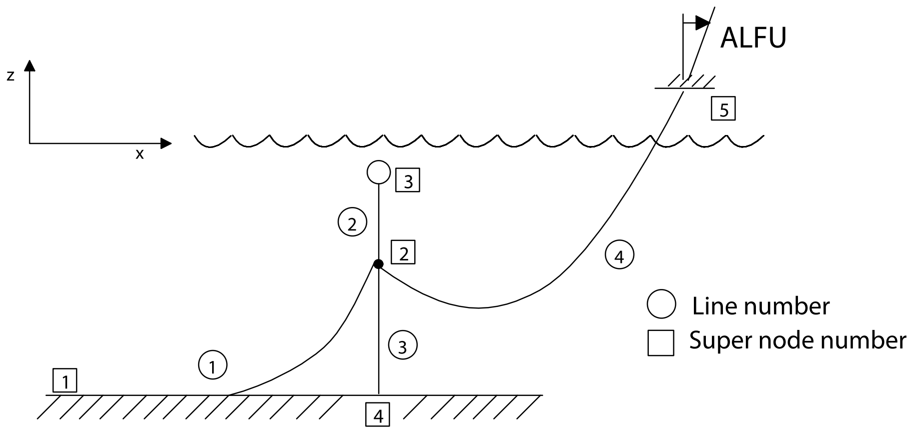

Figure 12. Topology of system SB

- Vertical branches with free ends are specified as for system SA
- One anchor point can be specified (in addition to supernode 1)
- Horizontal seafloor tangent can be specified

5.3.3.2. Data group identifier

```
SINGle RISEr SB
```

5.3.3.3. Definition of topology

```
NSNOD IBTANG
```

- `NSNOD: integer`: Number of supernodes
- `IBTANG: integer, default: 0`: Bottom tangent option
    - `IBTANG=0`: No seafloor contact
    - `IBTANG = 1`: Seafloor contact forces on all nodes that are below `Z < ZBOT + R_EXTCNT`. The modified 3D seafloor formulation is used. Friction contribution to torsional loading is possible.
    - `IBTANG = -1`: Equivalent with setting `IBTANG=1`.
    - `IBTANG = -9`: Seafloor contact forces on all nodes that are below `Z < ZBOT` using the original flat bottom formulation. If `IBOT3D = 1`, `IBTANG` is set to 1. This option is deprecated.

In the static `CAT`-analysis only contact in the vicinity of supernode 1 is included. Contact forces on all nodes that are below `ZL` are considered in static `FEM` analysis and dynamic analysis.

Note that flat bottom topology based on original FORTRAN code is planned to be removed and substituted by the general 3D seafloor contact formulation FORTRAN code. The old code will be kept for debugging purposes.

5.3.3.4. Line, line type and supernode connectivity

This data group defines the connectivity between lines and supernodes. If the line identifier is missing the line number implicitly defined by the order in which the lines are specified, will be used as the line identifier. References to line type IDs and supernode numbers are mandatory.

The lines must be specified in the order indicated in the figure `Topology of system SB' above. This means that the lines are given continuously from seafloor to upper riser end.

At each branching point the branching line(s) are specified before the next line in the main riser configuration. No ball joint components are accepted in branch lines.

`NLIN (=NSNOD-1)` input lines ~ LINE-ID LINTYP-ID ISNOD1 ISNOD2 ~ - `LINE-ID: character(8)`: Line identifier - `LINTYP-ID: character(8)`: Reference to line type identifier - `ISNOD1: integer`: Reference to supernode number at end 1 - `ISNOD2: integer`: Reference to supernode number at end 2

If only 1 alphanumeric string and 2 integers are specified, the string is interpreted as `LINTYP-ID`. ~ LINTYP-ID ISNOD1 ISNOD2 ~ The `LINE-ID` is taken as the line number as implicitly defined by the order in which the lines are given.

5.3.3.5. Boundary conditions

`ZL XU ZU ALFL ALFU ZA XA`

- `ZL: real`: Z-coordinate of lower end $[L]$
    - ZL will also be used as Z-coordinate of seafloor if `IBTANG>0`
- `XU: real > 0`: X-coordinate of upper end $[L]$
- `ZU: real`: Z-coordinate of upper end $[L]$
- `ALFL: real`: Angle of lower end from vertical $[deg]$
    - Dummy when seafloor contact is specified (i.e. `IBTANG/=0`)
- `ALFU: real`: Angle of upper end from vertical $[deg]$
- `ZA: real >= ZL`: Z-coordinate of anchor point $[L]$
    - Dummy if no anchor point is specified

`XA: real`: X-coordinate of anchor point $[\mathrm{L}]$

  - Required input when static `FEM` analysis is applied. The X-location of the anchor point is automatically computed so that the anchor line is vertical if the `CAT` or `CATFEM` analysis is used in `STAMOD` (i.e. `XA` is dummy). `XA` is also dummy when no anchor point is specified.

If the lower/upper end later in the specification is allowed to rotate freely around the Y-axis, `ALFL`/`ALFU` will be dummy.

5.3.3.6. Supernode types

`NSNOD-2` input lines. `ISNOD` must be given in increasing order from 2 to `NSNOD-2`. Only to be given if `NSNOD>2`.

`ISNOD ITYPSN`

- `ISNOD: integer`: Supernode no = 2,3,….., NSNOD-1
- `ITYPSN: character(6)`: Type of supernode
  - `TSNFIX` - Fixed
  - `TSNBRA` - Branch point
  - `TSNFRE` - Free end

Specification of supernodes: Supernodes at lower and upper end are not to be specified. The supernode number at lower end is automatically set to 1 and the supernode type is fixed (`ITYPSN=TSNFIX`). The supernode at upper end is automatically set to NSNOD and the supernode type is specified position (`ITYPSN=TSNPOS`) indicating the upper end is connected to the support vessel. A possible additional anchor point is defined by specification of a supernode of type `TSNFIX`. The additional anchor line must be connected to the first branching point along the main riser.

5.3.3.7. Seafloor support conditions

To be given only if `IBTANG`=1, -1 or -9. ~ STFBOT STFAXI STFLAT FRIAXI FRILAT DAMBOT DAMAXI DAMLAT ILTOR ~ - `STFBOT: real > 0`: Seafloor stiffness normal to the seafloor $[\mathrm{F}/\mathrm{L}^2]$ - `STFAXI: real >= 0, default: 0`: In-plane seafloor stiffness for friction in axial direction $[\mathrm{F}/\mathrm{L}^2]$ - `STFLAT: real >= 0, default: 0`: In-plane seafloor stiffness for friction in lateral direction $[\mathrm{F}/\mathrm{L}^2]$ - `FRIAXI: real >= 0, default: 0`: In-plane seafloor friction coefficient in axial direction [1] - `FRILAT: real >= 0, default: 0`: In-plane seafloor friction coefficient in lateral direction [1] - `DAMBOT: real >= 0, default: 0`: seafloor damping coefficient normal to the seafloor $[\mathrm{F} \times \mathrm{T}/\mathrm{L}^2]$ - Dummy for IBTANG=-9 - `DAMAXI: real >= 0, default: 0`: In-plane seafloor damping coefficient in axial direction $[\mathrm{F} \times \mathrm{T}/\mathrm{L}^2]$ - Dummy for

IBTANG=-9 - `DAMLAT: real >= 0, default: 0`: In-plane seafloor damping coefficient in lateral direction $[F \times T/L^2]$ - Dummy for IBTANG=-9 - `ILTOR: integer, default: 0`: Option for applying lateral contact forces at the external contact radius, giving a torsional moment - `= 0`: Lateral loads are applied at the node - `= 1`: Lateral loads are applied at the external contact radius if it is specified for the associated beam cross-section. - Dummy for IBTANG=-9

`STFBOT` is used for computing the spring stiffness normal to the seafloor, k_V , for seafloor contact. k_V = `STFBOT` $\times L$ where L is the element length.

Horizontal contact with the seafloor is modelled independently in the axial and lateral directions. Contact is initially modelled with linear springs. Sliding will occur when an axial or lateral spring force reaches the friction force value. Springs will be reinstated if the line starts sliding in the opposite direction, or if the friction force increases and is greater than the spring force. The spring stiffness is calculated as k_s = `STFxxx` $\times L_h$, where L_h is the length of the element's horizontal projection. The seafloor friction forces are calculated as F = `FRIxxx` $\times F_{vert}$ and are directed against the axial or lateral displacements.

5.3.3.8. Support vessel reference

Identification and location of support vessel. ~ IVES IDWFTR XG YG ZG DIRX ~ - `IVES: integer, default: 1`: Vessel number (`IVES` = 1) - `IDWTFR: character (6), default: 'NONE'`: Identifier for WF motion transfer function - IDWFTR = `NONE' means no transfer function specified - `XG: real`: X position of vessel coordinate system referred in global system $[L]$ - `YG: real`: Y position of vessel coordinate system referred in global system $[L]$ - `ZG: real`: Z position of vessel coordinate system referred in global system $[L]$ - Confer Data Group E: Support Vessel Data. - `DIRX: real`: Direction of vessel X-axis. - See [Location of support vessel coordinate system] (@ref Location_of_support_vessel_coordinate_system).

Next data group is Line and segment specification.

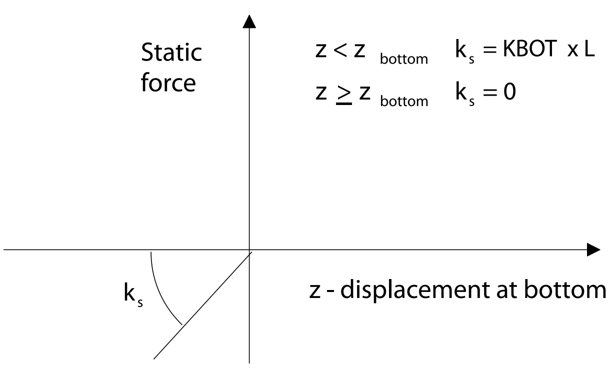

Figure 13. Vertical spring stiffness for seafloor contact

5.3.4. Standard system SC

Free lower end, suspended from surface vessel. This is a topologically simple system, with only two supernodes and one line.

5.3.4.1. Data group identifier

`SINGle RISEr SC`

5.3.4.2. Boundary conditions

`ZU ALFU`

- `ZU: real`: Z-coordinate of upper end $[L]$
- `ALFU: real`: Angle of upper end from vertical $[deg]$

If the upper end later in the specification is allowed to rotate freely around the Y-axis, ALFU will be dummy.

5.3.4.3. Line and line type

`LINE-ID LINTYP-ID`

- `LINE-ID: character(8)`: Line identifier
- `LINTYP-ID: character(8)`: Reference to line type identifier

If only 1 alphanumeric string is specified, the string is interpreted as `LINTYP-ID`. ~ LINTYP-ID ~ The `LINE-ID` is taken as the line number (=1)

Numbering of supernodes: - `ISNOD = 1` is lower end (free) - `ISNOD = 2` is upper end (specified position)

5.3.4.4. Support vessel reference

Identification and location of support vessel. ~ IVES IDWFTR XG YG ZG DIRX ~ - `IVES: integer, default: 1`: Vessel number (`IVES` = 1) - `IDWTFR: character (6), default: 'NONE'`: Identifier for WF motion transfer function - IDWFTR = `NONE' means no transfer function specified - `XG: real`: X position of vessel coordinate system referred in global system $[L]$ - `YG: real`: Y position of vessel coordinate system referred in global system $[L]$ - `ZG: real`: Z position of vessel coordinate system referred in global system $[L]$ - Confer Data Group E: Support Vessel Data. - `DIRX: real`: Direction of vessel X-axis. - See [Location of support vessel coordinate system] (@ref Location_of_support_vessel_coordinate_system).

Next data group is Line and segment specification.

5.3.5. Standard system SD

Free upper end.

5.3.5.1. Data group identifier

`SINGle RISEr SD`

5.3.5.2. Boundary condition

`ZL ALFL`

- `ZL: real`: Z-coordinate of lower end [L]
- `ALFU: real`: Angle of lower end from vertical [deg]

If the lower end later in the specification is allowed to rotate freely around the Y-axis, `ALFL` will be dummy.

5.3.5.3. Line and line type

`LINE-ID LINTYP-ID`

- `LINE-ID: character(8)`: Line identifier
- `LINTYP-ID: character(8)`: Reference to line type identifier

If only 1 alphanumeric string is specified, the string is interpreted as `LINTYP-ID`. ~ LINTYP-ID ~ The `LINE-ID` is taken as the line number (=1)

Numbering of supernodes: - `ISNOD = 1` is lower end, (fixed) - `ISNOD = 2` is upper end, (free)

Next data group is Line and segment specification.

5.3.6. Arbitrary system AR

This type of specification can be used to model:

- any of the previously specified systems
- systems with complex topology not covered by the standard systems
- systems for which boundary conditions are not covered by SA-SD systems

For this class of systems the topology must be described in detail. Boundary conditions corresponding to a 1) stress-free configuration, as well as 2) the wanted static configuration and parameters for a load incrementation procedure from 1) to 2) must be specified. The load incrementation procedure is specified as input to the `STAMOD` module.

5.3.6.1. Data group identifier

`ARBItrary SYSTem AR`

5.3.6.2. Topology

`NSNOD NLIN NSNFIX NVES NRICON NSPR NAKC`

- `NSNOD: integer`: Total number of supernodes
- `NLIN: integer, default: NSNOD-1`: Number of lines
- `NSNFIX: integer, default: 1`: Number of supernodes having one or more degrees of freedom that are fixed or prescribed
- `NVES: integer, default: 0`: Number of support vessels
- `NRICON: integer, default: 0`: Number of rigid supernode connections. Relevant for modelling of stiff connections between supernodes. The connection is described by specification of a master and a slave (dependent) node, see Rigid supernode connections
- `NSPR: integer, default: 0`: Number of global springs
- `NAKC: integer, default: 0`: Number of kill and choke lines. Relevant for modelling of tensioned risers only (Deprecated functionality)

5.3.6.3. Seafloor support conditions

5.3.6.3.1. Seafloor contact specification

`IBTANG ZBOT IBOT3D`

- `IBTANG: integer, default: 0`: Bottom tangent option.
  - `IBTANG = 0`: No seafloor contact
  - `IBTANG = 1`: Seafloor contact forces on all nodes that are below `Z < ZBOT + R_EXTCNT`. The modified 3D seafloor formulation is used. Friction contribution to torsional loading is possible.
  - `IBTANG = 2`: `CARISIMA` seafloor interaction model. Restricted option
  - `IBTANG = 3`: Seafloor contact elements will be added according to the specification given in the data group SEAFLOOR CONTACT SPECIFICATION.
  - `IBTANG = -1`: Equivalent with setting `IBTANG=1`.
  - `IBTANG = -9`: Seafloor contact forces on all nodes that are below `Z < ZBOT` using the original flat bottom formulation. If `IBOT3D = 1`, `IBTANG` is set to 1. This option is deprecated.

- `ZBOT: real`: Z-coordinate of seafloor (`ZBOT < 0`). $[\mathrm{L}]$
  - Dummy variable if `IBTANG = 0` or `IBOT3D = 1`.
- `IBOT3D: integer, default: 0`: Code for 3D bottom
  - `IBOT3D = 0`: flat bottom at depth `ZBOT`
  - `IBOT3D = 1`: 3D topology, file to be specified in input to STAMOD

Note that flat bottom topology based on original Fortran code is planned to be removed and substituted by the general 3D seafloor contact formulation FORTRAN code. The old code will be kept for debugging purposes.

5.3.6.3.2. Seafloor stiffness, friction and damping

The following input line must only be given if `IBTANG=1`, `IBTANG=-1` or `IBTANG=-9` ~ STFBOT STFAXI STFLAT FRIAXI FRILAT DAMBOT DAMAXI DAMLAT ILTOR ~ - `STFBOT: real > 0`: Seafloor stiffness normal to the seafloor $[\mathrm{F}/\mathrm{L}^2]$ - `STFAXI: real >= 0, default: 0`: In-plane seafloor stiffness for friction in axial direction $[\mathrm{F}/\mathrm{L}^2]$ - `STFLAT: real >= 0, default: 0`: In-plane seafloor stiffness for friction in lateral direction $[\mathrm{F}/\mathrm{L}^2]$ - `FRIAXI: real >= 0, default: 0`: In-plane seafloor friction coefficient in axial direction [1] - `FRILAT: real >= 0, default: 0`: In-plane seafloor friction coefficient in lateral direction [1] - `DAMBOT: real >= 0, default: 0`: seafloor damping coefficient normal to the seafloor $[\mathrm{F}\times\mathrm{T}/\mathrm{L}^2]$ - Dummy for IBTANG=-9 - `DAMAXI: real >= 0, default: 0`: In-plane seafloor damping coefficient in axial direction $[\mathrm{F}\times\mathrm{T}/\mathrm{L}^2]$ - Dummy for IBTANG=-9 - `DAMLAT: real >= 0, default: 0`: In-plane seafloor damping coefficient in lateral direction $[\mathrm{F}\times\mathrm{T}/\mathrm{L}^2]$ - Dummy for IBTANG=-9 - `ILTOR: integer, default: 0`: Option for applying lateral contact forces at the external contact radius, giving a torsional moment - `= 0`: Lateral loads are applied at the node - `= 1`: Lateral loads are applied at the external contact radius if it is specified for the associated beam cross-section. - Dummy for IBTANG=-9

`STFBOT` is used for computing the vertical spring stiffness, $\mathrm{k_V}$, for seafloor contact. $\mathrm{k_V}$ = `STFBOT` $\times\mathrm{L}$ where L is the element length.

Horizontal contact with the seafloor is modelled independently in the axial and lateral directions. Contact is initially modelled with linear springs. Sliding will occur when an axial or lateral spring force reaches the friction force value. Springs will be reinstated if the line starts sliding in the opposite direction, or if the friction force increases and is greater than the spring force. The spring stiffness is calculated as $k_h = \mathrm{Stalks} \times L_h$, where $\mathrm{L_h}$ is the length of the element's horizontal projection. The seafloor friction forces are calculated as $F = \mathrm{FRIxxx} \times F_{vert}$ and are directed against the axial or lateral displacements, where $\mathrm{F_{vert}}$ is the vertical contact force between the pipe and the bottom.

5.3.6.3.3. To be given only if `IBTANG = 2` (Restricted option) and `IBOT3D = 0`

`PHIG PHIS ZCUT X0 Y0`

- `PHIG: real, default: 0`: Angle anti-clockwise from X-axis to largest gradient $[\mathrm{deg}]$
- `PHIS: real, default: 0`: Slope of seafloor $[\mathrm{deg}]$
- `ZCUT: real, default: 0`: Elements above `ZCUT` ignored in contact analysis $[\mathrm{L}]$
- `X0: real, default: 0`: X position of seafloor origin with depth `ZBOT` $[\mathrm{L}]$
- `Y0: real, default: 0`: Y position of seafloor origin with depth `ZBOT` $[\mathrm{L}]$

Giving a low `ZCUT` increases efficiency, but should not be given lower than the highest position of any point actually in contact with the seafloor.

5.3.6.4. Line, line type and supernode connectivity

This data group defines the connectivity between lines and supernodes. If the line identifier is missing the line number implicitly defined by the order in which the lines are specified, will be used as the line identifier. References to line type IDs and supernode IDs are mandatory.

`NLIN` input lines.

`LINE-ID LINTYP-ID SNOD-ID1 SNOD_ID2`

- `LINE-ID: character(8)`: Line identifier
- `LINTYP-ID: character(8)`: Reference to line type identifier
- `SNOD-ID1: character(8)`: Reference to supernode identifier at end 1
- `SNOD-ID2: character(8)`: Reference to supernode identifier at end 2

If only 3 alphanumeric strings are specified, the first string is taken as `LINTYP-ID`, the second as `SNOD-ID1` and the third as `SNOD-ID2`: ~ LINTYP-ID SNOD-ID1 SNOD-ID2 ~ The `LINE-ID` is taken as the line number as implicitly defined by the order in which the lines are given.

The local element y- and z-axes are found from local x-axis and a reference vector.

The local element x-axis goes from end 1 to end 2 of the element.

The reference vector may be specified using the option LOCAL ELEMENT AXIS.

If the line is a blade in a wind turbine and `LOCAL ELEMENT AXIS` is not specified, the reference vector will be found as the cross product between the local X-axis of the shaft and the local X-axis of the blade element.

If the reference is not given by either of these two methods, the default method will be used. If the local x-axis is not parallel with the global z-axis, the reference vector is found as the cross product between the local x-axis and the global z-axis. If the local x-axis is parallel with the global z-axis, the global y-axis will be used as the reference vector if the local x-axis is completely vertical or tilted slightly in the positive global x-direction. Otherwise the reference vector will point in the negative global y-direction.

The local z-axis is found as the cross product between the local x-axis and the reference vector.

The local y-axis is then found as the cross product between the local z-axis and the local x-axis.

For beam elements, the element axes are found at the stress-free configuration and will subsequently follow the element.

5.3.6.5. Specification of boundary conditions, stressfree configuration and static equilibrium configuration

Coordinates of all supernodes must be specified to define the initial stressfree configuration as well as the final static configuration. If the distance between supernodes in stressfree configuration do not correspond to the line length as specified in Type and location of contact point, NCNODE input lines, the length of the last segment in the line is adjusted, and a warning is written.

5.3.6.5.1. Boundary conditions and coordinates for supernodes with at least one fixed or prescribed degree of freedom

The following two or three input lines must be given in blocks for each of the `NSNFIX` supernodes.

Boundary conditions: ~ SNOD_ID IPOS IX IY IZ IRX IRY IRZ CHCOO CHUPRO ~

- `SNOD_ID: character(8)`: Supernode identifier
- `IPOS: integer, default: 0`: Boundary condition type
  - `IPOS = 0`: The supernode is not connected to a support vessel
  - `IPOS = IVES`: The supernode is connected to support vessel number `IVES`, `1 ⇐ IVES ⇐ NVES`.
- `IX: integer, default: 1`: Boundary condition code for translation in X-direction
  - `IX = 0`: Free
  - `IX = 1`: Fixed or prescribed
- `IY: integer, default: 1`: Boundary condition code for translation in Y-direction (same interpretation as for `IX`)

`IZ: integer, default: 1`: Boundary condition code for translation in Z-direction (same interpretation as for `IX`)

- `IRX: integer, default: 1`: Boundary condition code for rotation about X-axis (same interpretation as for `IX`)
- `IRY: integer, default: 1`: Boundary condition code for rotation about Y-axis (same interpretation as for `IX`)
- `IRZ: integer, default: 1`: Boundary condition code for rotation about Z-axis (same interpretation as for `IX`)
- `CHCOO: character(6)`: Identifier for boundary condition reference coordinate system
  - `CHCOO = GLOBAL`: Boundary conditions are referenced to global coordinate system.
  - `CHCOO = SKEW-G`: Boundary conditions are referenced to a skew coordinate system.
  - `CHCOO = VESSEL`: Boundary conditions are referenced to vessel coordinate system.
  - `CHCOO = SKEW-V`: Boundary conditions are referenced to a skew vessel coordinate system.
- `CHUPRO: character(3), default: NO`: Computational parameter. Boundaries rotate with specified rotation
  - `CHUPRO = YES`
  - `CHUPRO = NO`

A supernode with prescribed motions during dynamic analysis must have `IPOS>0`.

Possible hinges at riser ends connected to fixed supports or to a support vessel may be modelled by either choosing the correct boundary condition code (see above) or using ball-joint connectors. Be careful not to use both these modelling options at the same time for a given super-node. This will lead to program abortion.

Note that if some of the translations are not prescribed, rotation-induced translations may cause drift-off if used in combination with global boundary conditions at a node attached to a vessel.

Coordinates for stress free and static equilibrium position: ~ X0 Y0 Z0 X1 Y1 Z1 ROT DIR ~

- `X0: real`: Coordinates for stress free configuration specified so that the line between any two supernodes are straight and with zero tension. [L]
- `Y0: real`: As for `X0`
- `Z0: real`: As for `X0`

- `X1: real, default: X0`:
- `Y1: real, default: Y0`: Coordinates for static equilibrium position $[\mathrm{L}]$
- `Z1: real, default: Z0`
- `ROT: real, default: 0`: Specified rotation of supernode from stress free position to static equilibrium position $[\mathrm{deg}]$
- `DIR: real, default: 0`: Direction of axis for specified rotation $[\mathrm{deg}]$

`ROT` is the specified rotation in degrees from stress free position to equilibrium position and is analogous to `ALFL`/`ALFU` parameters used for the standard systems. The rotation `ROT` will be around the local `YREF`-axis as shown in the figure below. `DIR` is the rotation in degrees from global X-axis to `XREF`-axis. The local `ZREF`-axis is parallel to global Z-axis. `DIR=0` signifies that the rotation `ROT` will be around the global Y-axis. If the line end is allowed to rotate freely around the local `YREF`-axis, `ROT` will be dummy. Free rotation around global Y-axis is obtained with `IRY = 0` and `DIR = 0`.

Definition of rotation axis YREF versus global coordinate system, X, Y. The supernode is located in the origin

Definition of skew coordinate system One input line only if `CHCOO` = `SKEW-G' OR `SKEW-V' ~ XX XY XZ XP YP ZP ~

- `XX: real`: Components of the skew X-axis referred to the global system. $[\mathrm{L}]$. See figure below.
- `XY: real`: As for `XX`.
- `XZ: real`: As for `XX`.
- `XP: real`: Components of a reference vector from the supernode to a point in the skew XY-plane, referred to global system $[\mathrm{L}]$
- `YP: real`: As for `XP`
- `ZP: real`: As for `XP`

The skew Z-axis is found by the cross product between the skew X-axis and the reference vector. The skew Y-axis is found by the cross product between the skew Z-axis and the skew X-axis

Definition of skew boundary system.

5.3.6.5.2. Coordinates for completely free supernodes

This input group consists of `NSNFRE` input lines, where `NSNFRE=NSNOD-NSNFIX` gives the number of supernodes where all degrees of freedom are free. Skip this group if `NSNFRE=0`.

```
SNOD-ID X0 Y0 Z0
```

- `SNOD-ID: character(8)`: Supernode identifier
- `X0: real`: Nodal coordinate in stress free configuration $[\mathrm{L}]$
- `Y0: real`: Nodal coordinate in stress free configuration $[\mathrm{L}]$
- `Z0: real`: Nodal coordinate in stress free configuration $[\mathrm{L}]$

5.3.6.6. Support vessel reference

Identification and location of support vessel, `NVES` input lines

```
IVES IDWFTR XG YG ZG DIRX
```

- `IVES: integer, default: 1`: Vessel number, `1 ⇐ IVES ⇐ NVES`.
- `IDWFTR: character(6), default: 'NONE'`: Identifier for support vessel motion transfer function
  - `IDWFTR = 'NONE'` means no transfer function specified
- `XG: real`: X position of vessel coordinate system referred in global system $[\mathrm{L}]$
- `YG: real`: Y position of vessel coordinate system referred in global system $[\mathrm{L}]$
- `ZG: real`: Z position of vessel coordinate system referred in global system $[\mathrm{L}]$
  - Confer Data Group E: Support Vessel Data
- `DIRX: real`: Direction of vessel X-axis.
  - See Location of support vessel coordinate system.

5.3.6.7. Description of global springs

To be specified if `NSPR>0` The input lines below (`Spring localization and properties' and `Nonlinear spring stiffness') must be given in one block for each global spring.

5.3.6.7.1. Spring location and properties

```
LINE-ID ISEG INOD ILDOF STIFF/NPAIR DAMP A2
```

`LINE-ID: character (8)`: Line identifier

- `ISEG: integer`: Local segment number within line
- `INOD: integer`: Local node number within actual segment
- `ILDOF: integer`: Local degree of freedom
  - =1 global X-direction
  - =2 global Y-direction
  - =3 global Z-direction
  - =4 rotation around global X-axis
  - =5 rotation around global Y-axis
  - =6 rotation around global Z-axis
  - = 12 or 21 translation in global XY-plane
  - = 13 or 31 translation in global XZ-plane
  - = 23 or 32 translation in global YZ-plane
- `STIFF/NPAIR`
  - `STIFF: real >= 0`: Constant spring stiffness $[\mathrm{F}/\mathrm{L}]$ or $[\mathrm{FL}/\mathrm{deg}]$
  - `NPAIR: integer ⇐ -2`: `NPAIR` is number of force-displacement or moment-rotation relations in spring specification
- `DAMP: real, default: 0`: Linear damping coefficient $[\mathrm{FT}/\mathrm{L}]$ or $[\mathrm{FLT}/\mathrm{deg}]$
- `A2: real, default: 0`: Stiffness proportional damping factor

5.3.6.7.2. Nonlinear spring stiffness

The following input line is to be given if `NPAIR` >= 2

```
PON(1) DISPL(1)...............PON(NPAIR) DISPL(NPAIR)
```

- `PON(1): real`: Spring force $[\mathrm{F}]$ or moment $[\mathrm{FL}]$ corresponding to
- `DISPL(1): real`: Spring displacement $[\mathrm{L}]$ or $[\mathrm{deg}]$

All `NPAIR` pairs of `PON` and `DISPL` values are given on a single input line. The values of `PON` and `DISPL` must be monotonically increasing; `PON(1) < PON(2) < … < PON(NPAIR)` and `DISPL(1) < DISPL(2) < … < DISPL(NPAIR)`.

For `ILDOF > 6`, the first pair of values must both be zero; i.e. `PON(1) = DISPL(1) = 0.0`.

5.3.6.8. Description of kill and choke lines (deprecated functionality)

NAKC input lines.

```
IKCTYP DIAAKC MASAKC FLUAKC TENAKC NLINKC LINE-ID(1) ... LINE-ID(nlinkc)
```

- `IKCTYP: integer`: Type of kill and choke line
  - `IKCTYP = 1`: Internal line
  - `IKCTYP = 2`: External line
- `DIAAKC: real, default: 0`: Outer diameter of kill and choke line $[\mathrm{L}]$
- `MASAKC: real, default: 0`: Mass per unit length of kill and choke line excluding contents $[\mathrm{M}/\mathrm{L}]$
- `FLUAKC: real, default: 0`: Mass per unit length of fluid contents of kill and choke line $[\mathrm{M}/\mathrm{L}]$
- `TENAKC: real, default: 0`: Tension of kill and choke line $[\mathrm{F}]$
- `NLINKC: integer, default: 1`: Number of riser lines this kill and choke line is attached to
- `LINE-ID(1): character(8)`: Reference to line identifiers (Adjacent lines)
- .
- .
- .
- `LINE-ID(nlinkc)`:

LINE-IDs must be given in correct order from lower end to upper end. Tension will be applied at the supernode at the second end of line `LINE-ID(nlinkc)`

If tension is zero, an internal line will be fixed at the upper end.

5.3.6.9. Rigid supernode connections

In present version only to be applied for static and non-linear dynamic analysis.

This option enables the user to model rigid connections between supernodes. A rigid connection is modelled by specifying a master - and a slave node. Both the master and the slave have initially to be defined as free nodes. The theoretical formulation is a special application of linear constraints between degrees of freedom.

NRICON input lines. ~ CHMAST CHSLAV ~

- `CHMAST: character`: Reference to supernode identifier, `SNOD-ID`, specified as the master node
- `CHSLAV: character`: Reference to supernode identifier, `SNOD-ID`, specified as the slave node

Note that: - Both the master and the slave node have initially to be defined as free nodes. - A master node can not be slave of another master node. - A slave node can only be slave of one master node. - An arbitrary number of slave nodes can have the same master node. - The number of DOFs at the slave node must not exceed the number of DOFs at the master node.

5.3.7. Seafloor contact specification

This data group must be given for Arbitrary Systems AR-sytems with `IBTANG = 3`.

5.3.7.1. Data group identifier, one input line

```
SEAFloor CONTact SPECification
```

5.3.7.2. Seafloor contact specification

```
NSPEC
```

- `NSPEC: integer > 0`: Number of input lines to be given with seafloor contact specification

NSPEC input lines ~ CMPTYP-ID LINE-ID ISEG1 ISEG2 … ISEGn ~

- `CMPTYP-ID: character(8)`: Reference to a seafloor contact component identifier. Must be of type `SPRI` or `SOIL`.
- `LINE-ID: character(8)`: Reference to a line identifier
- `ISEG1: integer >= 0, default: 0`: Segment for which seafloor contact of type `CMPTYP-ID` is possible.
    - `ISEG1 = 0`: Seafloor contact is possible for all segments in line `LINE-ID`

`ISEG1 > 0`: First segment for which seafloor contact is possible.

- `ISEG2: integer != 0`: Segment for which seafloor contact of type `CMPTYP-ID` is possible.
  - `ISEG2 > 0`: Second segment for which seafloor contact is possible.
  - `ISEG2 < 0`: Seafloor contact is possible for all segments from ISEG1 to ABS(ISEG2).
- `ISEGn: integer !=0`: Last segment for which seafloor contact of type `CMPTYP-ID` is possible.
  - `ISEGn > 0`: Last segment for which seafloor contact is possible.
  - `ISEGn < 0`: Seafloor contact is possible for all segments from the previous specified segment to ABS(ISEGn).

Pairs of a positive and a negative segment number may be given anywhere in the sequence.

Note that a segment may only have one seafloor contact.

5.3.8. Elastic contact surface

This data group is optional and is available as additional information for Arbitrary Systems only. It enables the user to model contact effects between lines. For normal riser systems this data group should not be considered.

The main intention of this data group is to enable modelling of pipelines during laying operations. This includes contact forces between the pipe and rollers on the lay barge/stinger and applied tension from a tensioner.

Contact between roller and pipe is modelled by a bi-linear or non-linear spring and a bi-linear dash pot damper. The contact force acts normal to the pipe and the roller. It is treated as a discrete element load acting on the pipe, while the contact load acting on the roller is transferred as a nodal force to the stinger. The last includes possible torsional moment.

The term *contact surface* is introduced to cover stinger modelling. The stinger may be fixed or hinged to the vessel. Generally it is curved and may consist of a rigid part following the vessel motions and a flexible part.

The term *contact point* is defined as the location of rollers or tensioner on the stinger.

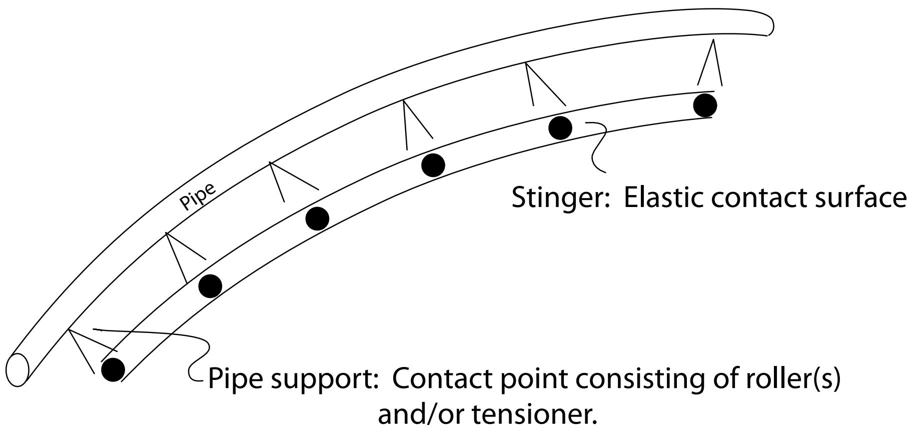

Figure 14. Elastic contact surface

5.3.8.1. Contact surface modelling

A complete model of an elastic contact surface includes the following information:

- Number of lines describing the surface The surface may consist of several adjacent lines. By introducing several adjacent lines it is possible to model a contact surface which has a curved stress-free initial configuration. In addition boundary conditions for the super-nodes at the line intersections can be specified. This is necessary to model prescribed displacements due to vessel motions.

- Type and location of contact points Contact points can be of roller and/or tensioner type and have to be located at ends of line segments.

- Identification of lines which may experience contact with the contact surface. The line identification is used to limit the number of elements that have to be checked for contact during program execution.

Supplementary information is specified in the following data groups: - System topology and boundary conditions - Line and segment specification - Contact point specification of roller type - Contact point specification of tensioner type - Tubular contact point specification

5.3.8.2. Data group identifier, one input line

```
ELAStic CONTact SURFace
```

5.3.8.3. Specification of elastic contact surface(s), one input line

```
NSURFE
```

- `NSURFE: integer > 0`: The number of elastic contact surfaces to be specified

5.3.8.4. Description of the contact surface(s) and specification of the line(s) which may experience contact with the surface(s).

The lines below, `Number of contact points…', `Type and location of contact point…' and `Specification of lines to be checked…', must be given in one block for each elastic contact surface.

5.3.8.4.1. Number of contact points along the current contact surface and number of lines which may experience contact with the surface, one input line.

```
NCNODE NCLINE
```

- `NCNODE: integer > 0`: Number of contact points located on the present contact surface
- `NCLINE: integer > 0`: Number of lines to be checked against contact with the present surface

5.3.8.4.2. Type and location of contact point, NCNODE input lines

```
ICNOD LINE-ID ISEG IEND CNTTY TENTY TUBTY
```

- `ICNOD: integer`: Contact point number
- `LINE-ID: character(8)`: Reference to line identifier
- `ISEG: integer`: Local segment number
- `IEND: integer`: Local segment end (1 or 2)
- `CNTTY: character(8)`: Reference to roller contact type identifier (`CMPTYP-ID`)
- `TENTY: character(8), default: 'NONE'`: Reference to tensioner contact type identifier (`CMPTYP-ID`)
- `TUBTY: character(8), default: 'NONE'`: Reference to tubular contact identifier (`CMPTYP-ID`)

1< `ICNOD` < `NCNODE`; icnod must be given in increasing order. The line number (`ILIN`) refers to the line number given implicit in the riser specification.

Note that: - `CNTTY`, `TENTY` and `TUBTY` refers to `CMPTYP-ID`: New component specification. - `CNTTY = 'NONE'` or `'0'` means no roller contact at the actual contact point. - `TENTY = 'NONE'` or `'0'` means no tensioner at the actual contact point. - `TUBTY = 'NONE'` or `'0'` means no tubular contact at the actual contact point.

Roller and tubular contact are activated by the `'ROLL'` command in the incremental static loading procedure. Tensioner type of contact is activated by the `'TENS'` command.

5.3.8.4.3. Specification of lines to be checked for contact with the current contact surface, NCLINE input lines.

```
LINCHK ISEGF IELF ISEGL IELL
```

- `LINCHK: character(8)`: Reference to line identifier (LINE-ID) to be checked for contact with the current contact surface.
- `ISEGF: integer, default: 1`: First local segment within the line to be checked for contact.
- `IELF: integer, default: 1`: First local element within the segment to be checked for contact.
- `ISEGL: integer, default: 0`: Last local segment within the line to be checked for contact.
- `IELL: integer, default: 0`: Last local element within the segment to be checked for contact.

`IELL = 0` means that the last element which will checked for contact is the last element within the segment `ISEGL`.

`ISEGL = 0` means that the last segment which will be checked for contact is the last segment within line `ILCHCK`.

5.3.9. Pipe-in-pipe contact

This data group is optional and is available for Arbitrary Systems only.

It enables the user to model pipe-in-pipe contact effects where each of the pipes is defined as a single line or as a main riser line. For normal riser systems this data group is normally not necessary.

A pipe-in-pipe pair consists of a master pipe and a slave pipe. Both the master and the slave pipe may consist of beam or bar types of elements and must be of cross section types CRS0, CRS1 or CRS3. The contact between the master pipe and the slave pipe will be applied between a node on the master pipe and an element in the slave pipe. This results in nodal loads along the master pipe and discrete element loads along the slave pipe. The discretization of the master pipe must therefore be fine enough to model the contact.

In the following the input parameters are described. A maximum of 400 pipe-in-pipe specifications may be given. The input lines in this group must be given in one block for each pipe-in-pipe pair.

See also @ref faq_modelling_pip

5.3.9.1. Data group identifier, one input line

```
PIPE IN PIPE
```

5.3.9.2. Pipe-in-pipe identifier, one input line

```
IDPIPE CHLOAD
```

- `IDPIPE: character(16)`: Pipe-in-pipe identifier
- `CHLOAD: character(4), default: EXPO`: Fluid loading on inner pipe
  - `= 'EXPO'`: Exposed. The inner pipe is exposed to external environmental loading; buoyancy and wave kinematics.
  - `= 'SHCL'`: Sheltered closed. The inner pipe is sheltered from external loading, but subjected to fluid loads from the inner fluid. The velocity and acceleration of the inner fluid follows the local transverse motion of the outer pipe. The inner fluid does not extend above the mean water level.

Note that: - If `CHLOAD = SHCL`, the master and slave pipes may not share supernodes.

5.3.9.3. Specification of master pipe, one input line

5.3.9.3.1. For master pipe defined by a single line

```
CHMAST LINE-ID ISEGF_M ISEGL_M
```

- `CHMAST : character(4)`: `= LINE`
- `LINE-ID: character(8)`: Reference to line identifier
- `ISEGF_M: integer, default: 1`: First local segment in line for master pipe
- `ISEGL_M: integer, default: NSEG`: Last local segment in line for master pipe

5.3.9.3.2. For master pipe defined by a main riser line

```
CHMAST = MRL  MRL-ID
```

- `CHMAST: character(4)`: `= MRL`
- `MRL-ID: character(8)`: Reference to main riser line identifier

5.3.9.4. Specification of slave pipe, one input line

5.3.9.4.1. For slave pipe defined by a single line

```
CHSLAV = LINE  LINE-ID ISEGF_S ISEGL_S
```

- `CHSLAV : character(4)`: `= LINE`

- `LINE-ID: character(8)`: Reference to line identifier
- `ISEGF_S: integer, default: 1`: First local segment in line for slave pipe
- `ISEGL_S: integer, default: NSEG`: Last local segment in line for slave pipe

Note that the slave pipe may not contain connectors.

5.3.9.4.2. For slave pipe defined by a main riser line

`CHSLAV = MRL  MRL-ID`

- `CHMAST: character(4): = MRL`
- `MRL-ID: character(8)`: Reference to main riser line identifier

Note that the slave pipe may not contain connectors.

5.3.9.5. Specification of contact force characteristics, one input line

`CHPOS IKS RELDAM DAMP STIFFR FRICST FRICDY CHAXI CHROT VELLIM`

- `CHPOS: character`: Position of master pipe
  - `= INNER`
  - `= OUTER`
- `IKS: integer`: Stiffness code for radial contact force
  - `= 1`: Constant contact compression stiffness
  - `= N`: Table with N pairs of contact force - displacement to be specified
- `RELDAM: real`: Desired relative damping level at estimated eigen period in pipe, pipe and contact spring system (see below) $[1]$. Damping is only applied in the radial direction. Not used in static analysis.
- `DAMP: real`: Dash pot damping coefficient per unit length of master pipe $[\mathrm{FT/L^2}]$. Damping is only applied in the radial direction. Not used in static analysis.
- `STIFFR: real`: Spring stiffness associated with the static friction coefficient `FRICST`, $[\mathrm{F/L^2}]$. The spring stiffness is applied in the ring and axial directions until the spring force exceeds the static friction force. Not used in static analysis. Dummy if `CHAXI = OFF`.
- `FRICST: real`: Static friction coefficient $[1]$. Not used in static analysis. Dummy if `CHAXI = OFF`.

`FRICDY: real`: Dynamic sliding friction coefficient $[1]$. `FRICDY` ⇐ `FRICST`. Not used in static analysis. Dummy if `CHAXI = OFF`.

- `CHAXI: character`: Control parameter for friction caused by translational displacements. Not used in static analysis.
  - `= ON`
  - `= OFF`
- `CHROT: character`: Control parameter for friction caused by rotation. Not used in static analysis.
  - `= ON`. Requires `CHAXI = ON`
  - `= OFF`
- `VELLIM: real`: Velocity limit for determining that sliding has stopped $[\mathrm{L/T}]$. If the relative sliding velocity between the pipes falls below `VELLIM`, the spring stiffness `STFFR` is applied. Should be small, but not zero. Not used in static analysis. Dummy if `CHAXI = OFF`.

Based on specified damping level the stiffness proportional damping coefficient is calculated by

$$a_2 = 2 \times \mathrm{RELDAM} \times \sqrt{\frac{\mathrm{AMS_M}+\mathrm{AMS_S}}{\mathrm{STIFF}}}$$

where $\mathrm{AMS_M}$ and $\mathrm{AMS_S}$ are structural mass per unit length of the master pipe and the slave pipe respectively and STIFF is contact spring stiffness per unit length.

5.3.9.5.1. Contact spring stiffness; IKS = 1, one input line

```
STIFF
```

- `STIFF: real`: Spring compression stiffness per unit length $[\mathrm{F/L^2}]$

5.3.9.5.2. Contact spring stiffness; IKS > 1, one input line

```
FS(1) ZS(1) ........ FS(N) ZS(N)
```

- `FS(1): real`: Contact force per unit length corresponding to the gap `ZS(1)` $[\mathrm{F/L}]$. A negative value is a force apposing penetration.
- `ZS(1): real`: Gap between the pipes $[\mathrm{L}]$, A negative values indicates contact.

`ZS(i)` must be given in increasing order

All `FS` and `ZS` values should be negative, as the `ZS` values are understood to be the gap between the two pipes. If no negative values are given, linear extrapolation will be done from the two smallest `ZS` values.

Please note that the sign convention of this input is planned changed in future versions!

5.3.10. Winch specification

This data group is optional and is available as additional information for Arbitrary Systems only.

It enables the user to model winch / winching. Boundary conditions previously defined for nodes attached to the winch will be substituted by the winch specification. For normal riser systems this data group should not be considered.

In the following the input parameters are described. Several winches may be specified. The lines in this group must be given in one block for each winch.

5.3.10.1. Data group identifier, one input line

```
WINCh SPECification
```

5.3.10.2. Specification of winch(es), one input line

```
NWINCH
```

- `NWINCH: integer`: Number of winches

5.3.10.2.1. Winch identifier, one input line

```
IDWINCH
```

- `IDWINCH: character(8)`: Winch identifier

5.3.10.2.2. Initial location of winch point, one input line

```
ILIN_W ISEG_WP RLSEG_WP IEND_W
```

- `ILIN_W: integer`: Line number attached to winch
- `ISEG_WP: integer`: Local segment number at winch point
- `RSEG_WP: real`: Relative segment length where segment is attached to winch point
- `IEND_W: integer`: End of segment (and line) attached to winch (1 or 2)

figurwinch

5.3.10.2.3. Final position of winch point, one input line

`XW YW ZW ROTW DIRW`

- `XW: real`: Coordinates for static equilibrium position $[\mathrm{L}]$
- `YW: real`: Coordinates for static equilibrium position $[\mathrm{L}]$
- `ZW: real`: Coordinates for static equilibrium position $[\mathrm{L}]$
- `ROTW: real`: Specified rotation of supernode from stress free position to static equilibrium position $[\mathrm{deg}]$
- `DIRW: real`: Direction of axis for specified rotation $[\mathrm{deg}]$

5.3.10.2.4. Boundary condition of winch, one input line

`CBOUND CIBODY`

- `CBOUND: character(6)`: Boundary condition for winch (All nodes attached to winch)
  - `= FIXED`: Fixed boundaries
  - `= VESSEL`: Winch attached to support vessel
  - `= FLOATE`: Winch attached to floater force model (`SIMO' body)
- `CIBODY`
  - `= IVES: integer`: Support vessel number (`CBOUND = VESSEL`)
  - `= CHBODY: character(8)`: Floater force model identifier (`CBOUND = FLOATE`)
  - (Dummy for `CBOUND=FIXED`)

5.3.10.2.5. Winch properties, one input line

`VELMAX TIMMAX LDROP RADIUS IZSIGN LELVAR`

- `VELMAX: real`: Maximum winch velocity $[\mathrm{L}/\mathrm{T}]$
- `TIMMAX: real`: Time to reach maximum velocity (from zero) $[\mathrm{T}]$
- `LDROP: integer`: Line release when no more line on winch (Dynamic analysis)
  - = 0 Not possible

= 1 Possible

- `RADIUS: real > 0`: Radius of winch $[\mathrm{L}]$
- `IZSIGN: integer`: = ± 1 Center of winch in positive or negative local Z-axis
- `LELVAR: integer`: Control parameter for adjusting the length of elements attached to winch.
    - = 0 No justification
    - = 1 Justification

The parameters `RADIUS`, `IZSIGN` and `LELVAR` specify how the elements attached to the winch are visualized.

For `LELVAR=0`: $\mathrm{RADIUS} >= (\mathrm{EL})/\sqrt{2}$, where EL is length of shortest element attached to the winch.

Note that Final position for winch is dummy for coupled analysis.

The RIFLEX winch formulation is intended for vertical winching and may be unstable when deviating from vertical in dynamic analysis.

It is recommended to apply some numerical damping when using the winch functionality, for example `` BETIN = 3.9`and `GAMMA = 0.505 ``, see @ref dynmod_e_method.

The Winch rotations follow the same conventions as prescribed rotations of nodes, see @ref inpmod_b_arbitrary_boundary_conditions. The coordinate system is rotated first using `DIRW`, followed by `ROTW`. For example: if Rotation direction `DIRW` is set to `0.0`, the winch will rotate `ROTW` degrees around the local y-axis. It is important to note that the coordinate system is rotated first relative to the global system using `DIRW`, the winch is then rotated `ROTW` degrees around the y-axis in the updated coordinate system.

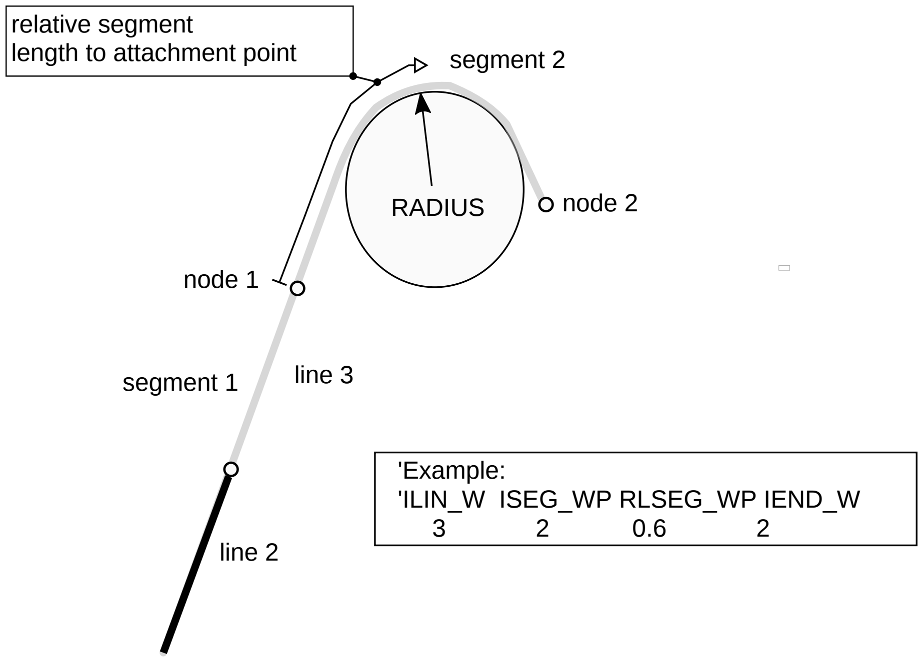

Figure 15. Sketch of a winch model

5.3.11. Wind turbine specification

This data group is optional and is only available for Arbitrary Systems. It enables the user to model wind turbines considering wind load acting on the blades and control system for blade pitch and electrical power extraction. For normal riser systems this data group should not be considered.

The wind turbine is modelled by a group of lines that constitute the blades and the shaft, see figure `Outline of a wind turbine model' below. In addition, references to the air foil library file and to control data for electrical torque and blade pitch must be given.

A shared supernode at the hub connects the blades and the shaft. Each blade must consist of two lines. The inner line represents eccentricity and the outer line the actual foil blade. A rigid supernode connection is used to represent connection between the two lines. The constant term, c0, in the linear constraint equation is subsequently used for blade pitch control.

The wind turbine blades have to be identical with regard to the element distribution, foil profile description and aerodynamic coefficients along the blades. The mass and stiffness distribution may be different.

The wind loads on the blades are computed based on the load coefficient description in the air foil library file and together with a blade element momentum (BEM) method. The applied BEM code includes dynamic inflow, i.e. a time delay on changes of induced velocity related to the time it takes to convect vorticity in the wake downstream, away from the rotor. With dynamic inflow, the BEM method will give correct time series of rotor and blade loads under conditions of changing blade pitch angle, wind speed and direction, and tower motion. The main features of the BEM theory are:

- Induced velocity is calculated assuming momentum balance for a ring-shaped control volume.
- Blade sections are treated as independent.
- Aerodynamic coefficients from wind tunnel tests are used for the blades.
- Empirical corrections are used for tip-vortices and cascade effects / lift amplification.

The BEM theory is a proven, simple and CPU efficient method to simulate rotor aerodynamics and the method represents the industry standard. Additional information regarding the coordinate systems for the wind turbine results can be found in Wind Turbine Results.

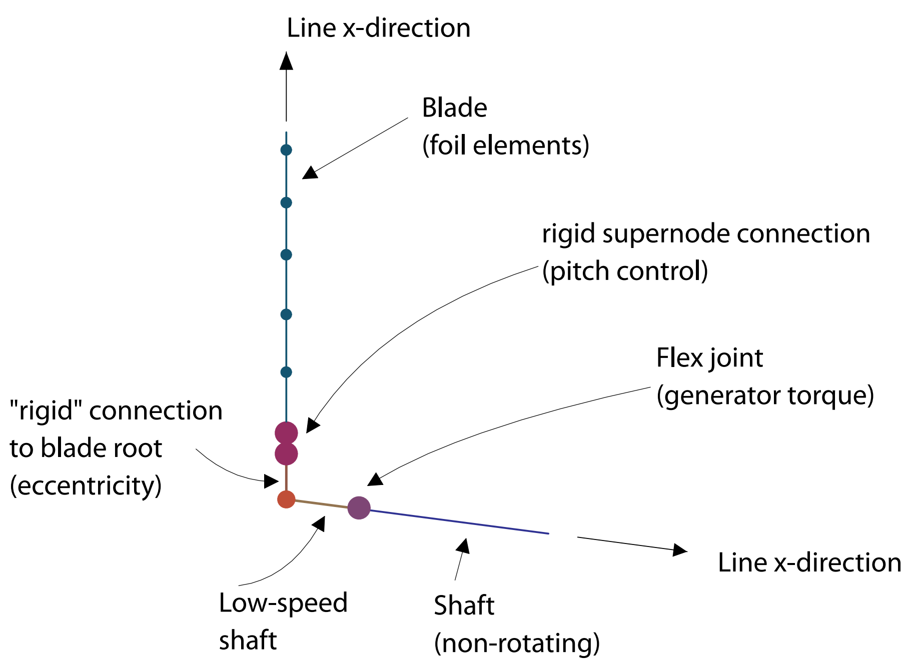

Figure 16. Outline of a wind turbine model

5.3.11.1. Data group identifier, one input line

```
WIND TURBine SPECification
```

5.3.11.2. Specification of windturbine(s), one input line

```
NWITURB
```

- `NWITURB: integer`: Number of wind turbines to be specified

In present version of RIFLEX only one wind turbine may be modelled.

5.3.11.3. Component type identifier

`FILE_NAME`

- `FILE_NAME: character`: Name of the airfoil coefficient library
    - `= 'NONE'`: No file

The next input lines are given in one block for each wind turbine.

5.3.11.4. Wind turbine identifier, one input line

`WIND-TURBINE-ID`

- `WIND-TURBINE-ID: character(8)`: Wind turbine identifier. The value `NONE' is not allowed.

5.3.11.5. Wind references for wind velocity, one input line

`WIND_REF AERO_DLL`

- `WIND_REF: character`: Coupled analysis: Reference to floater force model (`SIMO` body) for wind velocity. Must be given for coupled analysis, dummy if not coupled analysis.
- `AERO_DLL: character, default: 'NONE'`: Code for AERODYN DLL (Educational feature)
    - `AERO_DLL = NONE` : Official RIFLEX version
    - `AERO_DLL = EXTE` : Restricted option only available for selected academic users

For coupled analyses with `SIMO` wind type `IWITYP >= 10`, the incoming wind is calculated at the updated nodal coordinates of the nodes along the blades.

For coupled analyses with `SIMO` wind type `IWITYP < 10`, the wind at the `SIMO` body `WIND_REF` is reported. Note that this wind is calculated at the wind force coefficient height `ZCOEFF`. This wind is also used as the incoming wind along the blades .

5.3.11.6. Wind load options, one input line

`CLMOM    CUPWN   IWTADV`

- `CLMOM: character, default: 'ON'`: ON-OFF switch for inclusion of wind moment around longitudinal axis of the blades.
    - `CLMOM = ON`: Include wind moment
    - `CLMOM = OFF`: Exclude wind moment and disregard distance between foil center and pitch axis

`CUPWN: character, default: 'UPWN'`: switch for upwind or downwind turbine

  - `CUPWN = UPWN`: Upwind turbine
  - `CUPWN = DNWN`: Downwind turbine

- `IWTADV: integer, default: 0`: switch for advanced wind turbine aerodynamic options
  - `IWTADV = 0`: Default wind turbine aerodynamic options
  - `IWTADV = N`: Number of lines with advanced wind turbine aerodynamic options

5.3.11.7. Advanced wind turbine aerodynamic load options, IWTADV>0, IWTADV input lines

```
CHID    CHSW
```

- `CHID: character(4)` : identifier for switch to be given
- `CHSW: character(3)` : on-off switch

Legal combinations of `CHID` and `CHSW` include:

- `CHID == INDU`: induction calculation
  - `CHSW = OFF`: No induction calculation (parked turbine)
  - `CHSW = ON`: Induction calculation included (default)
- `CHID == PRTI`: Prandtl tip correction
  - `CHSW = OFF`: No Prandtl correction applied at the blade tip
  - `CHSW = ON`: Prandtl correction applied at the blade tip (default)
- `CHID == PRRO`: Prandtl root correction
  - `CHSW = OFF`: No Prandtl correction applied at the blade root (default)
  - `CHSW = ON`: Prandtl correction applied at the blade root
- `CHID == PRYA`: Prandtl factor correction for yaw
  - `CHSW = OFF`: Keep angle ϕ constant regardless of yawed inflow
  - `CHSW = ON`: Correct angle ϕ in the Prandtl factor for yawed inflow (default)

5.3.11.8. Specification of internal or external control system for blade pitch, electrical power and nacelle yaw, one input line

Specification of internal or external blade pitch and electrical power control system, [internal](@ref inpmod_b_wind_specification_internal) and external. Specification of internal or external nacelle yaw controller, Nacelle yaw controller.

`CHCODE  YAWCCODE`

- `CHCODE: character(4)`:
  - `CHCODE = INTC`: Internal control system
  - `CHCODE = EXTC`: External control system
- `YAWCCODE: character(4)`, default: `NONE`: Control system used for yaw
  - `YAWCCODE = NONE`: No yaw control system
  - `YAWCCODE = YCIN`: Internal yaw control system
  - `YAWCCODE = YCEX`: External yaw control system (not implemented)

5.3.11.8.1. Internal control system input for blade pitch and electrical power, CHCODE = INTC, one input line

`FILE_NAME`

Descriptions of the internal control system and file format are found in Specification of internal control system for blade pitch and electrical power.

5.3.11.8.2. External control system input for blade pitch and electrical power, CHCODE = EXTC, 3 input lines

`JarName`

`ClassName`

`Config`

Additional information about the interface for the external control system is given in Interface for external wind turbine control system.

5.3.11.8.3. Specification of nodes and elements for additional measurements, CHCODE = EXTC

`NNOD_MEAS  NEL_MEAS`

- `NNOD_MEAS: integer, default: 0`: Number of nodes for which additional measurements will be sent to the external control system. `NNOD_MEAS = 0` for the internal control system.
- `NEL_MEAS: integer, default: 0`: Number of elements for which additional measurements will be sent to the external control system. `NEL_MEAS = 0` for the internal control system.

======= Specification of nodes for additional measurements, NNOD_MEAS input lines

`LINE-ID ISEG INOD SYSTEM`

- `LINE-ID: character(8)`: Line identifier
- `ISEG: integer > 0`: Local segment number in specified line
- `INOD: integer > 0`: Local node number in specified segment
- `SYSTEM: character(6), default: 'GLOBAL'`: switch for reference system for the nodal measurements exported to the external controller.
  - `SYSTEM = GLOBAL`: Displacement, velocity and acceleration in the global coordinate system
  - `SYSTEM = SHAFT0`: Displacement, velocity and acceleration in the initial shaft system

======= Specification of elements for additional measurements, NEL_MEAS input lines

`LINE-ID ISEG IEL SYSTEM`

- `LINE-ID: character(8)`: Line identifier
- `ISEG: integer > 0`: Local segment number in specified line
- `IEL: integer > 0`: Local node number in specified segment
- `SYSTEM: character(6), default: 'LOCAL'`: switch for reference system for the element measurements exported to the external controller.
  - `SYSTEM = LOCAL`: Element forces and moments in the local element system

5.3.11.9. Shaft and tower specification, one input line

`LINE-IDSFT LINE-IDTWR  BAK  ICDCOR`

- `LINE-IDSFT: character(8)`: Reference to the line that is used for shaft modelling.
- `LINE-IDTWR: character(8), default: 'NONE'`: Reference to the line that is used for tower modelling.
  - If specified the incoming wind acting on the blades will be modified due to the presence of the tower.
- `BAK: real, default: 0`: Bak modification factor for tower shadow. May be applied to up- or down-wind turbines.

`ICDCOR: integer, default: 0`: Option for modification of the tower shadow based on the tower drag. May be applied to up- or down-wind turbines.

One and only one flex-joint has to be specified within the shaft line (line type). The flex-joint has to be fixed in the bending degrees of freedom and be given a specified stiffness in the torsional degree of freedom. The specified torsion stiffness will be used during static analysis. During dynamic analysis, the torsion will be controlled be the regulator.

5.3.11.10. Number of blades in the rotor, one input line

```
NBLADE
```

- `NBLADE: integer`: Number of blades

5.3.11.11. Blade identification, NBLADE input lines

```
LINE1-ID LINE2-ID
```

- `LINE1-ID: character(8)`: Reference to the inner line of the blade (eccentricity).
- `LINE2-ID: character(8)`: Reference to the outer line of the blade (foils).

The supernode at the eccentricity has to be master node and the supernode at the blade the slave node in the the rigid supernode connection between the eccentricity and the blade.

5.3.11.12. Specification of internal or external control system for nacelle yaw, one input line

5.3.11.12.1. Internal yaw control system input, YAWCCODE=YCIN, 1 input lines

```
FILE_NAME
```

Descriptions of the internal yaw control system and file format are found in ….

5.3.11.12.2. External yaw control system input, YAWCCODE=YCEX, 3 input lines

```
JarName
```

```
ClassName
```

```
Config
```

5.3.11.12.3. Yaw element identification, YAWCCODE=YCEX or YAWCCODE=YCIN, 1 input line

```
REFLINE-ID YAWLINE-ID
```

- `REFLINE-ID: character(8)`: Reference to the line that connected to the yaw element.

`YAWLINE-ID: character(8)`: Reference to the yaw element.

5.3.12. Airfoil coefficient library specification

This data group is optional and is only available for Arbitrary Systems. Only one airfoil coefficient library may be specified. This data group is not permitted for systems which include a wind turbine specification.

The airfoil lift, drag, and moment coefficients on elements that are not part of a wind turbine are applied via lookup table (no induction).

5.3.12.1. Data group identifier, one input line

```
AIRFoil COEFficient LIBRary
```

5.3.12.2. File name specification, one input line

```
FILE_NAME
```

- `FILE_NAME: character(256)`: Path to the airfoil coefficient library

5.3.13. Potential flow library specification

This functionality is currently under testing.

This data group is optional. Only one potential flow library may be specified.

For cross-sections with POTN loading, this file provides the frequency-dependent radiation and diffraction data. If cross-sections with POTN loading are used in the analysis, this input is required (checked in stamod).

5.3.13.1. Data group identifier, one input line

```
POTEntial FLOW LIBRary
```

5.3.13.2. File name specification, one input line

```
FILE_NAME
```

- `FILE_NAME: character(256)`: Path to the potential flow library

5.3.13.3. Connection between FE-elements and data in the pontential flow library

5.3.13.3.1. Number of elements with POTN loading, one input line

Note that the number of elements with POTN loading and wich are located below sea level has to be equal to the number of sections in the potential flow library!

```
NOPOTELEM
```

- `NOPOTELEM: integer`: Number of elements with POTN loading

5.3.13.3.2. Connections between elements and sections in the potential flow library, NOPOTELEM input lines

Note that the sequence of element specifications has to coincide with the sequence of sections in the library.

```
LINE-ID ISEG IEL
```

- `LINE-ID: character(8)`: Line identifier
- `ISEG: integer`: Local segment number within the line `LINE-ID`
- `IEL: integer`: Local element number within the segment `ISEG`

5.3.14. Main riser line specification

5.3.14.1. Data group identifier, one input line

The main riser line (MRL) may be used to define continuous contents of several lines.

This data group may be given for AR-systems only. Only one riser system may be defined on the INPMOD input file.

```
MAIN RISEr LINE
```

5.3.14.1.1. Main riser line definition, one input line

```
MRL-ID LINE-ID1 ... LINE-IDn
```

- `MRL-ID: character(8)`: Main riser line identifier
- `LINE-ID1: character(8)`: Reference to line identifier
  - `LINE-ID1` = First line in MRL
  - By default the MRL is defined from end 1 to end 2 of this line
  - If the line ID is given subsequent to a minus sign, `-LINE-ID1`, the MRL is defined from end 2 to end 1 of this line
- …

`LINE-IDn: integer`:

A MRL may consist of up to 200 lines. A maximum of 5 MRLs may be defined.

5.3.14.1.2. Main riser line flow, one input line

`RHOI IEND PRESSI DPRESS VVELI`

- `RHOI: real, default: 0`: Density of contents $[\mathrm{M}/\mathrm{L}^3]$
- `IEND: integer`:
  - `IEND=1`: Pressure specified at end 1
  - `IEND=2`: Pressure specified at end 2
- `PRESSI: real, default: 0`: Pressure at end `IEND` $[\mathrm{F}/\mathrm{L}^2]$
- `DPRESS: real, default: 0`: Pressure drop $[\mathrm{F}/\mathrm{L}^3]$
- `VVELI: real, default: 0`: Fluid velocity $[\mathrm{L}^3/\mathrm{T}]$
  - Dummy in present version

The pressure drop is assumed to be uniform over the line length starting at end `IEND`. The values given here will replace values given in the `FLUID` specification.

An illustration is shown the figure `Main riser line concept' below with corresponding input data.

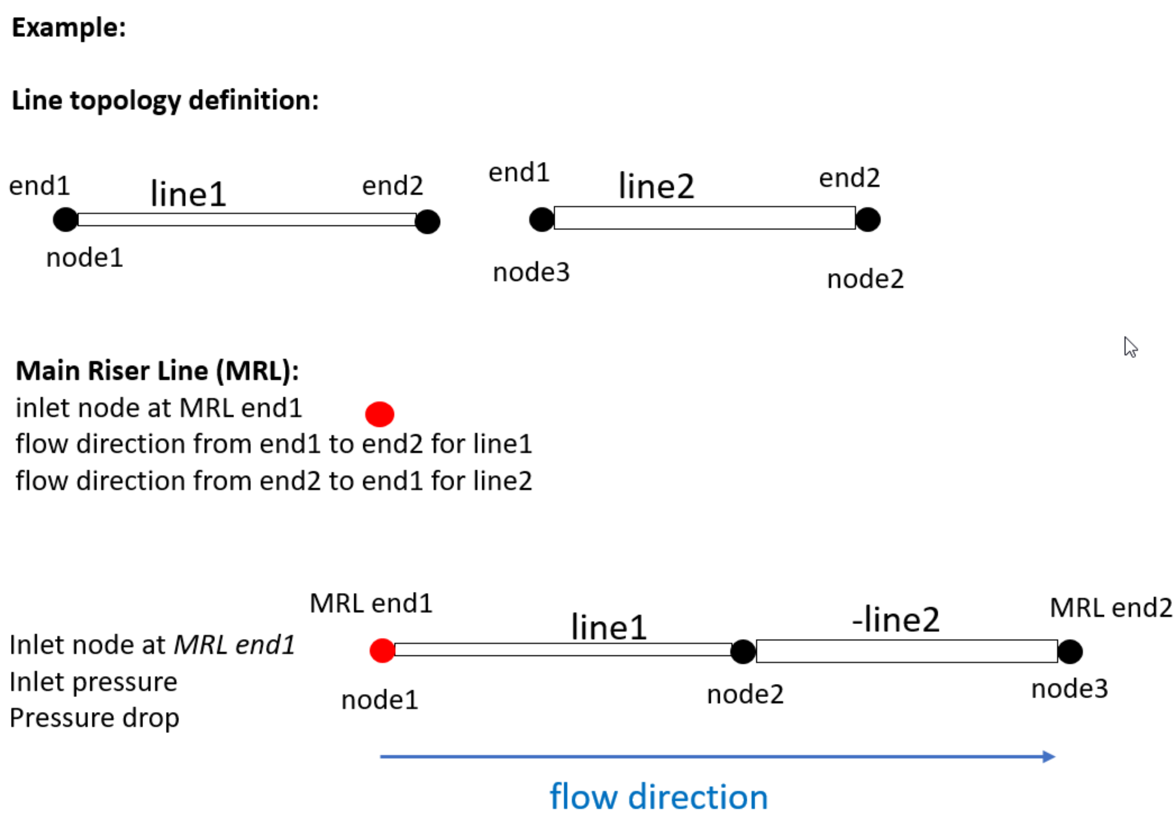

Figure 17. Main riser line concept

%# Main riser line input \{#inpmod_mrl_example}

Input using the main riser line concept is shown below.

5.3.15. Geotechnical spring specification

Additional to the Arbitrary system data group.

This data group is used to specify geotechnical springs in the global X-Y plane suitable for modeling cyclic geotechnical data with degradation. The stiffness of the spring is degraded based on the secant stiffness to ultimate resultant displacement from initial position.

The applied damping calculates damping force based on the following model

$$\mathrm{F} = \mathrm{C}(\mathrm{d}) \times |\dot{\mathrm{d}}|^{\mathrm{P}} \times \mathrm{sign}(\dot{\mathrm{d}})$$

where - F: damping force - C: damping coefficient (strain / displacement dependent) - d: spring displacement - $\dot{\mathrm{d}}$: velocity of attached node - P: exponent for strain velocity (P >= 1)

Note that the degradation of the geotechnical spring will dissipate some energy in the cycles where the spring is updated.

5.3.15.1. Data group identifier, one input line

`GEO SPRINg SPECification`

5.3.15.2. Number of geotechnical springs, one input line

`NGEOSPR`

- `NGEOSPR: integer > 0`: Number of geotechnical springs

The following input lines are repeared `NGEOSPR` times.

5.3.15.3. Spring ID

`SPRING-ID`

- `SPRING-ID: character(8)`: Spring ID. Must be unique.

5.3.15.4. Spring localization and properties

4 or 5 input lines, depending on the damping model selected,

`LINE-ID ISEG INOD`

- `LINE-ID: character(8)`: Line identifier
- `ISEG: integer`: Local segment number within line
- `INOD: integer`: Local node number within segment

`RLEN`

- `RLEN: real, default: 0`: Relative length, for scaling of results only

`FORCE1 DISP1 .... FORCEn DISPn`

- `FORCE1: real`: Force corresponding to DISP1. First value must be zwero. $[\mathrm{F}]$
- `DISP1: real`: Displacement correcponding to FORCE1. First value must be zwero. $[\mathrm{L}]$
- `FORCEi: real > 0`: Force corresponding to DISPi. $[\mathrm{F}]$
- `DISPi: real > DISP_i-1`: Displacement correcponding to FORCEi. $[\mathrm{L}]$

`IDMP EXPDMP`

- `IDMP: integer >= 0`: Damping coefficient code

- `= 1`: Constant damping coefficient
  - `= N`: Table with N pairs of damping coefficient / Displacement to be specified.
- `EXPDMP: real >= 0`: Exponent for displacement velocity (Z = 1). Dummy if IDMP = 0.

No more input in this data group if `IDMP = 0`

1 line of input if `IDSP = 1`:

```
DAMPGEO
```

- `DMPGEO: real > 0`: Displacement independent damping coefficient

1 line of input with IDMP pairs of damping values if `IDMP > 1`:

```
DAMPGEO1 DISP1 ... DAMPGEOidmp DISPidmp
```

- `DMPGEO1: real`: Damping coefficient corresponding to DISPL1
- `DISPL1: real`: Displacement corresponding to damping coefficient DMPGEO1. [L]
- `DMPGEOi: real`: Damping coefficient corresponding to DISPLi
- `DISPLi: real`: Displacement corresponding to damping coefficient DMPGEOi. [L]

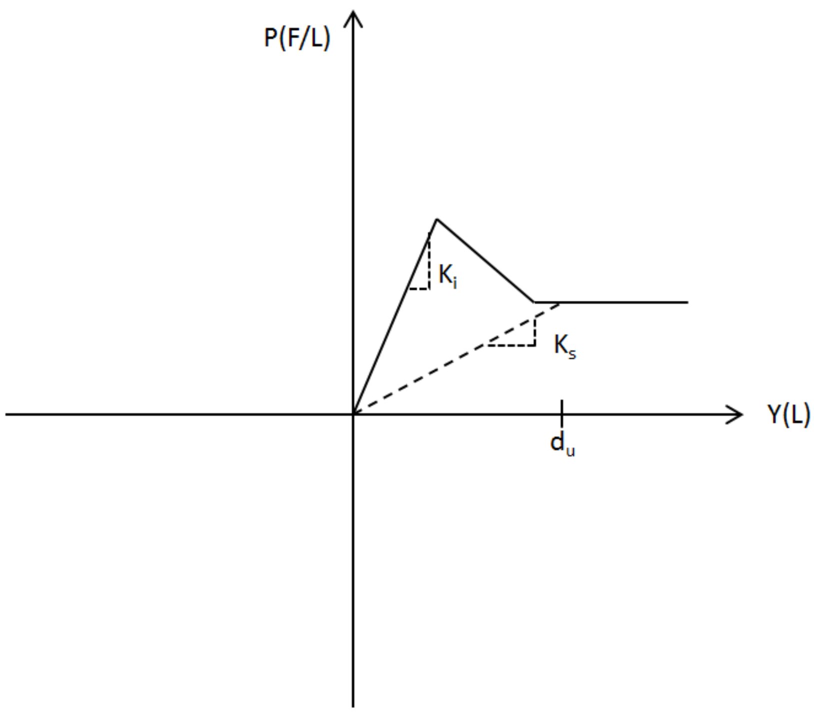

Figure 18. Spring degradation model

The figure shows the initial stiffness K_i and updated secant stiffness K_s after displacement d_u in Geotechnical spring.

5.3.16. Line and segment specification

5.3.16.1. Data group identifier, one input line

Lines which represent stress joints are specified in data group Stress joint line specification.

The total number of line types and stress joints has to be less or equal to 500 in the present version.

```
NEW LINE DATA
```

5.3.16.2. Line type specification, one input line

```
LINTYP-ID NSEG NCMPTY2 FLUTYP IADDTWI IADDBEND
```

- `LINTYP-ID: character(8)`: Line type identifier.
- `NSEG: integer`: The number of segments the line type consists of.
- `NCMPTY2: character/integer, default: 0`: Reference to nodal component type, `CMPTYP-ID`, attached to end 2 of segment `NSEG`.
    - Must be type `BODY`, `CONB`, `FLEX` or `DRAG`.
    - `NCMPTY = 'NONE'` or `'0'` means that no nodal component is attached at end 2 of segment NSEG.
- `FLUTYP: character/integer, default: 0`: Reference to internal fluid component type identifier. Must be of type `FLUID`.
    - `FLUTYP = 'NONE'` or `'0'` means no fluid in the line.
- `IADDTWI: integer, default: 0`: Indicator for twist information (Relative rotation around the line type length axis).
    - `IADDTWI = 0`: No extra specification to be given.
    - `IADDTWI = 1`: Extra specification to be given.
- `IADDBEND: integer, default: 0`: Indicator for pre-curved line type, i.e. offsets transverse to the straight line between the line ends.
    - `IADDBEND = 0`: No extra specification to be given.

`IADDBEND = 1`: Extra specification to be given at the second end of all but the last segment within the line type

- `IADDBEND = 2`: Extra specification to be given at the second end of all segments

5.3.16.3. Segment specification. NSEG input lines.

`CRSTYP NCMPTY1 EXWTYP NELSEG SLGTH NSTRPS NSTRPD SLGTH0 SOITYP`

- `CRSTYP: character`: Reference to cross-sectional component type identifier `CMPTYP-ID`
  - Must be of type `CRS0` … `CRS7` or `FIBR`
- `NCMPTY1: character/integer, default: 0`: Reference to nodal component type identifier, `CMPTYP-ID`, attached to end 1 of segment.
  - Must be type `BODY`, `CONB`, `FLEX` or `DRAG`.
  - `NCMPTY1 = 'NONE'` or `'0'` means that no nodal component is attached at end 1 of segment.
- `EXWTYP: character/integer`: Reference to external wrapping (distributed weight or buoyancy) component type identifier `CMPTYP-ID`.
  - Must be of type `EXT1`.
  - `EXWTYP = 'NONE'` or `'0'` means no external wrapping.
- `NELSEG: integer`: Number of elements for FEM analysis
- `SLGTH: real > 0`: Segment length $[\mathrm{L}]$
- `NSTRPS: integer, default: 3`: Number of sections each element is divided into for hydro-dynamic calculation; static analysis.
- `NSTRPD: integer, default: 5`: Number of sections each element is divided into for hydro-dynamic load calculation; dynamic analysis.
- `SLGTH0: real, default: SLGTH`: Actual stress free segment length
- `SOITYP: character/integer, default: '0'`: Reference to seafloor contact component type identifier `CMPTYP-ID`.
  - Must be a `SEAF` component of type `CARI` (restricted option).
  - Only to be specified if `IBTANG = 2`
  - `SOITYP = 'NONE'` or `'0'` means that the segment has no seafloor contact.

- Restricted option.

`NSTRPS` and `NSTRPD` are used only for cross-section types: `CRS3`, `CRS4`, `CRS5` and `CRS6`. Dummy for `CRS0`, `CRS1`, `CRS2` and `CRS7`.

If the stress free length of the line (sum of the segment `SLGTH`) is not equal to the distance between the stress free coordinates of the supernodes that the line is attached to, the stress free line length will be modified according to the following rules:

Length modification will always be done on the last segment within the line. Therefore the first task is to check if it is possible to modify this segment to obtain a line length equal the to the distance. If this is not possible the program will terminate with error message. Then the difference between the length and the distance is within the preset length tolerance: - Difference larger than 1% gives error termination. - Difference between 0.1% and 1% gives modification and written warning. - Difference less than 0.1% gives modification but no warning.

Specifying `SLGTH0` $\neq$ `SLGTH` enables the user to use initially stressed segments at the start of the static analysis. This feature is useful for modelling elastic springs between element nodes. `SLGTH` will be interpreted as stressed segment length by the program.

5.3.16.4. Relative twist specification

The following NSEG input lines is to be given if `IADDTWI = 1`.

`TWEND1 TWEND2`

- `TWEND1: real`: Relative twist segment end 1 $[\mathrm{deg}]$
- `TWEND2: real, default: TWEND1`: Relative twist segment end 2 $[\mathrm{deg}]$

The relative twist for elements within the segment is calculated by use of linear interpolation. The twist angle is constant over the element length. The twist angle of the actual line will relate to the line local Y-axis.

The local Y-axis may be set using the data group LOCAL ELEMENT AXIS, be determined by shaft and blade orientation for a wind turbine blade or the default procedure may be used. See Line, line type and supernode connectivity for details.

5.3.16.5. Transverse offset specification

If `IADDBEND = 1` the following NSEG-1 input lines are to be given. Offsets at the last node of the last segment are set to zero.

If `IADDBEND = 2` the following NSEG input lines are to be given.

`DY DZ`

DY: real, default: 0: Offset in line local Y-axis segment end 2 $[L]$

- DZ: real, default: 0: Offset in line local Z-axis segment end 2 $[L]$

This feature enables the user to model a curved stress-free configuration of a line type by specifying transverse offsets at end 2 of all line segments, i.e. transverse offsets from the straight line between the line ends. The offsets DY and DZ refer to the initial local line Y and Z axes, disregarding any specified relative twist.

Note! If only NSEG-1 input lines are given, the offsets are set to zero for the second end of the last segment.

Specified twist will be applied around the updated local element X-axis after the offsets have been accounted for.

The stressfree segment lengths SLGTH will be modified according to the specified offsets. If a segment consist of more than one element, the intermediate nodes will be placed along the straight line between the segment nodes.

The cross-section properties will not be changed, e.g. the specified mass per unit length will be used together with the modified segment length.

Note! If non-zero offset is specified at the second end of the last segment, no other line may be connected to this supernode

The local Y-axis may be set using the data group LOCAL ELEMENT AXIS, be determined by shaft and blade orientation for a wind turbine blade or the default procedure may be used. See Line, line type and supernode connectivity for details.

5.3.17. Stress joint line specification

5.3.17.1. Data group identifier, one input line

The stress joint lines are labelled with an unique line type identifier, LINTYP-ID. The total number of line types and stress joints has to be less or equal to 500 in the present version.

```
STREss JOINt DATA
```

5.3.17.2. Stress joint specification, one input line

```
LINTYP-ID CDSJ CMASJ NSJSEC FLUTYP
```

- LINTYP-ID: character(8): Line type identifier
- CDSJ: real: Non-dimensional quadratic drag coefficient in normal direction
- CMASJ: real: Non-dimensional added mass coefficient in normal direction

- `NSJSEC: integer`: Number of conical sections in stress joint
- `FLUTYP: character/integer`: Reference to internal fluid component type identifier, `CMPTYP-ID`.
  - Must be of type `FLUID`.
  - `FLUTYP = 'NONE'` or `'0'` means no fluid in the line.

5.3.17.3. Initial cross-section parameters, one input line

```
DESJS THSJS
```

- `DESJS: real`: External diameter at first end of first conical section in stress joint $[\mathrm{L}]$
- `THSJS: real`: Wall thickness at first end of first conical section in stress joint $[\mathrm{L}]$

5.3.17.4. Parameters to define the conical stress joint sections, NSJSEC input lines

```
NSJS DESJ THSJ SJSL NELSJ EMOD RHO
```

- `NSJS: integer`: Stress joint section number. To be given in increasing order starting with #1
- `DESJ: real`: External diameter at second end of the section $[\mathrm{L}]$
- `THSJ: real`: Wall thickness at second end of the section $[\mathrm{L}]$
- `SJSL: real`: Length of the section $[\mathrm{L}]$
- `NELSJ: integer`: Number of segments within the section
- `EMOD: real`: Young’s modulus of elasticity $[\mathrm{F}/\mathrm{L}^2]$
- `RHO: real`: Density of pipe material $[\mathrm{M}/\mathrm{L}^3]$

Each segment will consist of one element. `CRS0` cross-sections will be generated automatically for each segment in the stress joint.

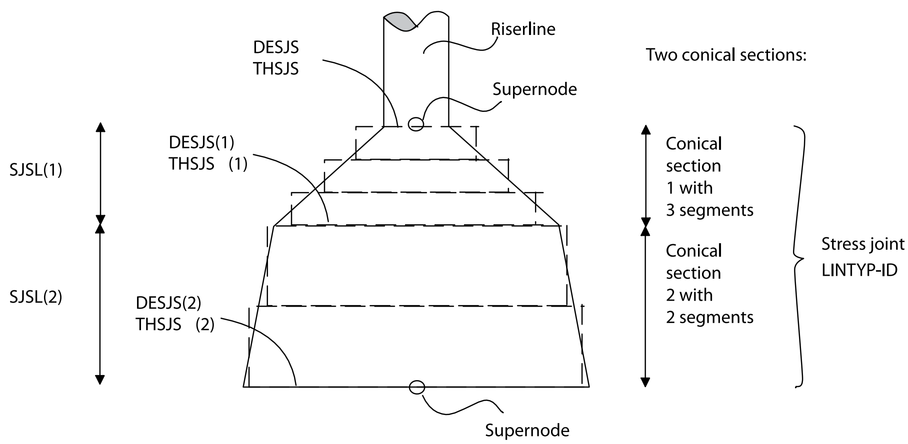

Figure 19. Stress joint description

5.4. Data Group C: Component Data

This section includes specification of all elementary components to be used for the riser modelling. It is possible to specify more components than are actually used.

The components are labelled with an identifier called “component type identifier: `CMPTYP-ID`. The maximum number of component types is 500 in the present version.

For each component a list of data ``attributes'' have to be specified. This list depends on the TYPE CODE given in the data group identifier.

The following component types are included:

- `CRS0`: Thin-walled pipe cross section
- `CRS1`: Axisymmetric cross section
- `CRS2`: Double symmetric cross section
- `BODY`: Body attached at one point. Mass point
- `CONB`: Connector. Ball-joint type
- `FLEX`: Flex joint (supressed translations)
- `FLUID`: Internal fluid flow
- `EXT1`: External wrapping, rotation symmetry
- `CRS3`: Partly submerged axisymmetric cross section
- `CRS4`: Partly submerged double symmetric cross section
- `CRS5`: Partly submerged general shaped cross section

- `CONT`: Contact point of roller type
- `TENS`: Tensioner
- `TUBU`: Tubular contact point
- `SEAF`: Seafloor contact
- `DRAG`: Drag chain element
- `FIBR`: Fibre rope cross section
- `GROW`: Marine growth
- `CRS6`: Fish net cross section
- `CRS7`: General cross section

Practical aspects of modelling:

- Bending stiffeners are assumed to be modelled by one or more segments with average mass, drag and stiffness properties from the riser and bending stiffener within each segment.
- External buoyancy of weight elements that are clamped to the pipe are specified as external wrapping.
- The mass of `EXT1` type component is added to the line properties. Drag and mass coefficients are added to those of the line.
- Body and external wrapping can not be specified for segments consisting of the cross section types: `CRS3`, `CRS4`, `CRS5`, or `CRS6`

5.4.1. CRS0 - Thin-walled pipe cross section

This cross-section allows for simplified input of circular, homogenous cross-sections. The input format is convenient for metallic pipe cross sections.

%# Example input \{#inpmod_c_crs0_example}

A thin-walled pipe cross section example is shown below. Subsequent sections give details and further options.

5.4.1.1. Data group identifier

```
NEW COMPonent CRS0
```

5.4.1.2. Component type identifier

`CMPTYP-ID TEMP ALPHA BETA`

- `CMPTYP-ID: character(8)`: Component type identifier
- `TEMP: real, default: 0`: Temperature at which the specification applies $[\mathrm{Temp}]$
- `ALPHA: character/real, default: 0`: Thermal expansion coefficient $[\mathrm{Temp}^{-1}]$
  - `= STEE`: The value 1.2×10^{-5} is used
  - `= TI23`: The value 9.0×10^{-6} is used
  - These values are applicable for temperatures in Celcius or Kelvin
- `BETA: character/real, default: 0`: Pressure expansion coefficient $[1/(\mathrm{F}/\mathrm{L}^2)]$
  - `BETA` gives the expansion of an element with zero effective tension from the difference between the internal and the external pressure.
  - `= PIPE`: thin walled pipe assumption. `BETA` is calculated from the parameters given in Cross-section parameters (below) as: $\frac{\mathrm{DIAST}(1-2\nu)}{4\mathrm{THST}\times\mathrm{EMOD}}$ where $\nu = \frac{\mathrm{EMOD}}{2\mathrm{GMOD}} - 1$

Axis symmetric pipe cross section

image::../figures/um_ii_fig56.svg [Axis symmetric pipe cross section,456]

5.4.1.3. Cross-section parameters

`DIAST THST DENSST THEX DENSEX R_EXTCNT R_INTCNT`

- `DIAST: real`: Diameter of pipe $[\mathrm{L}]$
  - `DIAST > 0`: Outer diameter of pipe
  - `DIAST < 0`: Inner diameter of pipe
- `THST: real`: Thickness of pipe $[\mathrm{L}]$
- `DENSST: real`: Density of pipe material $[\mathrm{M}/\mathrm{L}^3]$
- `THEX: real, default: 0`: Thickness of external coating $[\mathrm{L}]$
- `DENSEX: real, default: 0`: Density of external coating $[\mathrm{M}/\mathrm{L}^3]$
- `R_EXTCNT: real, default: 0`: Outer contact radius $[\mathrm{L}]$
- `R_INTCNT: real, default: 0`: Inner contact radius $[\mathrm{L}]$

Buoyancy is calculated from the total external diameter $\mathrm{DIAST} + 2 \times \mathrm{THEX}$ (For `DIAST > 0`) or $|\mathrm{DIAST}| + 2 \times \mathrm{THST} + 2 \times \mathrm{THEX}$ (For `DIAST < 0`).

The outer and inner contact radii of the cross section, `R_EXTCNT` and `R_INTCNT`, are used for - seafloor contact (unless `IBTANG = -9`; original flat bottom formulation) - pipe-in-pipe contact - tubular component contact

The default values of `R_EXTCNT` and `R_INTCNT` are zero in the present version.

5.4.1.4. Material properties

5.4.1.4.1. Material constants

`MATKIND EMOD GMOD SIGY EMODY/NPAIR HARPAR NCIRC`

- `MATKIND: integer`: Type of material model
    - `MATKIND = 1`: linear material
    - `MATKIND = 2`: elastic-plastic
    - `MATKIND = 3`: strain-stress curve
    - `MATKIND = 4`: linear material including shear deformation
- `EMOD: real > 0`: Modulus of elasticity $[\mathrm{F}/\mathrm{L}^2]$
- `GMOD: real > 0`: Shear modulus $[\mathrm{F}/\mathrm{L}^2]$
- `SIGY: real`: Yield stress $[\mathrm{F}/\mathrm{L}^2]$
- `EMODY/NPAIR: real/integer`:
    - `MATKIND = 2`: Slope of strain-stress curve for plastic region $[\mathrm{F}/\mathrm{L}^2]$.
        - `EMODY` < `EMOD`
    - `MATKIND = 3`: Number of user specified strain-stress relations
        - `2` ⇐ `NPAIR` ⇐ `99`
- `HARPAR: real, default: 1`: Hardening parameter for material
    - `0` ⇐ `HARPAR` ⇐ `1`
    - `HARPAR = 1`: Kinematic hardening
    - `HARPAR = 0`: Isotropic hardening

- `NCIRC: integer >= 8, default: 16`: Number of integration points along circumference

For `MATKIND = 1` or `4`: Only `EMOD` and `GMOD` are used

For `MATKIND = 4`: The shear stiffness is calculated as: $\mathrm{GMOD}\frac{\pi(\mathrm{D}_\mathrm{e}^2-\mathrm{D}_\mathrm{i}^2)}{4}0.5$

For `MATKIND = 3`: `NPAIR` input lines of the strain-stress curve must be given (@ref inpmod_c_crs0_strain).

5.4.1.4.2. Strain-stress curve (NPAIR input lines to be specified for MATKIND=3)

```
EPS(I) SIG(I)
```

- `EPS(i): real`: Strain for point i on strain-stress curve $[1]$
- `SIG(i): real`: Stress for point i on strain-stress curve $[\mathrm{F}/\mathrm{L}^2]$

The first point in the stress-strain curve is automatically deduced: `EPS(0) = SIGY/EMOD`, `SIG(0) = SIGY`. This point is taken as the proportionality limit of the material, at which the yield/hardening process starts. `EPS(i)` and `SIG(i)` are to be given in increasing order. The gradient of the curve must decrease with increasing strain.

5.4.1.5. Bending-torsion geometric coupling specification for `MATKIND = 1` or `4`

This data group is optional, and can only be applied for `MATKIND = 1` or `4`.

```
BTGC
```

- `BTGC: character(4)`: bending-torsion coupling identifier.

If the `BTGC` identifier is present, geometric coupling between torsion and bending is accounted for.

5.4.1.6. Damping specification

Identical to input for cross-section type CRS1 except that the local axial friction model, `AXFRC`, is illegal for CRS0, see Damping specification.

5.4.1.7. Hydrodynamic force coefficients

Identical to input for cross-section type CRS1, see Hydrodynamic load type.

5.4.1.8. Aerodynamic force coefficients

Identical to input for cross-section type CRS1, see Aerodynamic load type identification.

5.4.1.9. Capacity parameter

Identical to input for cross-section type CRS1, see Capacity parameter.

5.4.2. CRS1 - Axisymmetric cross section

%# Example input \{#inpmod_c_crs1_example}

The following is a CRS1 cross section example. Subsequent sections provide details and further options.

5.4.2.1. Data group identifier

```
NEW COMPonent CRS1
```

5.4.2.2. Component type identifier

```
CMPTYP-ID TEMP ALPHA BETA
```

- `CMPTYP-ID: character(8)`: Component type identifier
- `TEMP: real, default: 0`: Temperature at which the specification applies $[\mathrm{Temp}]$
- `ALPHA: real, default: 0`: Thermal expansion coefficient $[\mathrm{Temp}^{-1}]$
- `BETA: real, default: 0`: Pressure expansion coefficient $[1/(\mathrm{F}/\mathrm{L}^2)]$
  - `BETA` gives the expansion of an element with zero effective tension from the difference between the internal and the external pressure.

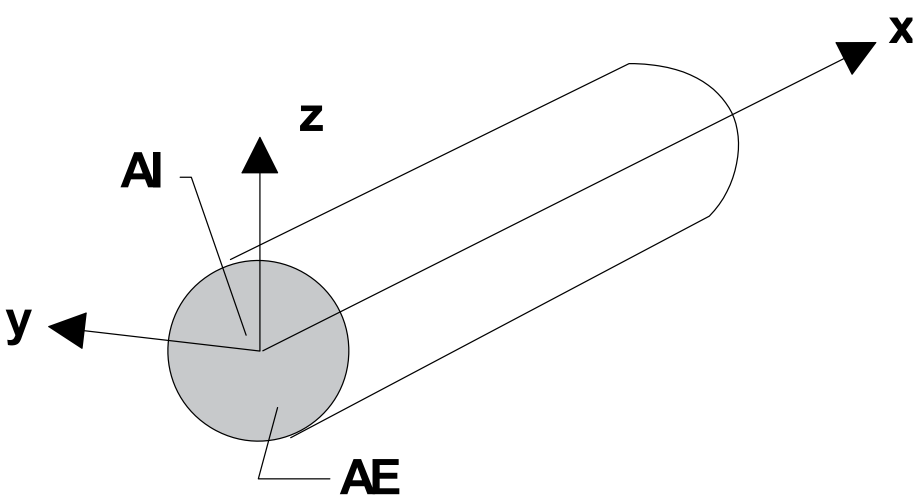

Figure 20. Axis symmetric cross section

5.4.2.3. Mass and volume

`AMS AE AI RGYR AST WST DST THST R_EXTCNT R_INTCNT`

- `AMS: real`: Mass/unit length $[\mathrm{M}/\mathrm{L}]$
- `AE: real`: External cross-sectional area $[\mathrm{L}^2]$
- `AI: real`: Internal cross-sectional area $[\mathrm{L}^2]$
- `RGYR: real`: Radius of gyration about local x-axis $[\mathrm{L}]$
- `AST: real`: Cross-section area for stress calculations $[\mathrm{L}^2]$
  - The default value is calculated as seen below
- `WST: real`: Cross-section modulus for stress calculations $[\mathrm{L}^3]$
  - The default value is calculated as seen below
- `DST: real`: Diameter for stress calculations $[\mathrm{L}]$
  - The default value is calculated as seen below
- `THST: real`: Thickness for stress calculations $[\mathrm{L}]$
  - The default value is calculated as seen below
- `R_EXTCNT: real, default: 0`: External contact radius $[\mathrm{L}]$
- `R_INTCNT: real, default: 0`: Inner contact radius $[\mathrm{L}]$

`AI` is used to calculate additional mass of internal fluid if present. Otherwise `AI` is dummy or see below.

Default values of the stress calculation parameters will be calculated from `AE` and `AI` if `AE > AI`. A homogenous cylinder shaped cross-section is assumed: - `AST` $= \mathrm{AE} - \mathrm{AI}$ - `WST` $= \pi(\mathrm{D_e^4} - \mathrm{D_i^4})/(32\mathrm{D_e})$ - `DST` $= \mathrm{D_e}$ - `THST` $= (\mathrm{D_e} - \mathrm{D_i})/2$ - where $\mathrm{D_e} = \sqrt{\frac{4\mathrm{AE}}{\pi}}$ and $\mathrm{D_i} = \sqrt{\frac{4\mathrm{AI}}{\pi}}$

The outer and inner contact radii of the cross section, `R_EXTCNT` and `R_INTCNT`, are used for - seafloor contact (unless `IBTANG = -9`; original flat bottom formulation) - pipe-in-pipe contact - tubular component contact

The default values of `R_EXTCNT` and `R_INTCNT` are zero in the present version.

5.4.2.4. Stiffness properties classification

`IEA IEJ IGT IPRESS IMF HARPAR`

- `IEA: integer, default: 1`: Axial stiffness code
  - 1 - constant stiffness
  - N - table with N pairs of tension-elongation to be specified
  - N >= 2
- `IEJ: integer, default: 0`:
  - `0` - zero bending stiffness
  - 1 - constant stiffness
  - N - table with N pairs of bending moment - curvature to be specified
  - N >= 2
- `IGT: integer, default: 0`: Torsion stiffness code
  - `0` - zero torsional stiffness
  - 1 - constant stiffness
  - -1- non-symmetric ``constant'' stiffness
  - N - symmetric, (N positive) pairs specified
  - -N- general torsion/relation (non-symmetric) N pairs specified
  - N >= 2
- `IPRESS: integer, default: 0`: Pressure dependency parameter related to bending moment
  - `0` - no pressure dependency
  - 1 - linear dependency (not implemented)
  - NP - NP sets of stiffness properties to be given, corresponding to a table of NP pressure values (not implemented)
    - 2 ⇐ NP ⇐ 10
- `IMF: integer, default: 0`: Hysteresis option in bending moment/curvature relation

- `0` - no hysteresis
- 1 - hysteresis generated by an internal friction moment at reversed curvature

- `HARPAR: real, default: 0`: Hardening parameter for kinematic/isotropic hardening
  - `0 ⇐ HARPAR ⇐ 1`
  - Only to be given if `IEJ > 1` and `IMF = 1`

`IEJ` and `IGT` must both be zero or both greater than zero to assure stability in the `FEM` analysis.

Note that: - `IPRESS=0` in this version. - `IMF=0`, `IMF=1` is implemented in present version. - `IMF` $\neq$ `0` should be used with care as the analysis can become unstable.

5.4.2.5. Bending-torsion geometric coupling specification

This data group is optional, and will only be applied for `IEJ=1`, `IGT=1`, and `IMF=0`.

```
BTGC
```

- `BTGC: character(4)`: bending-torsion coupling identifier.

If the `BTGC` identifier is present, geometric coupling between torsion and bending is accounted for.

5.4.2.6. Axial stiffness. Case 1, IEA=1

```
EA
```

- `EA: real > 0`: Axial stiffness $[\mathrm{F}]$

5.4.2.7. Axial stiffness. Case 2, IEA=N

```
EAF(1) ELONG(1) . . . EAF(N) ELONG(N)
```

- `EAF(1): real`: Axial force corresponding to relative elongation `ELONG(1)` $[\mathrm{F}]$
- `ELONG(1): real`: Relative elongation ()
- .
- .
- .

The pairs of `EAF` and `ELONG` must be given in increasing order on a single input line.

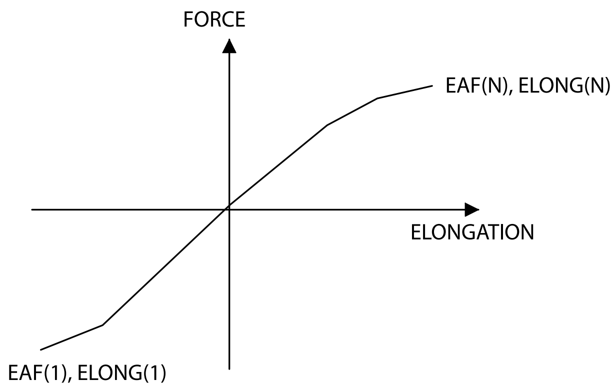

Figure 21. Axial force corresponding to relative elongation

5.4.2.8. Bending stiffness properties

The amount of input depends upon the parameters `IEJ`, `IPRESS` and `IMF` according to the table below: - Case: 0, `IEJ`: 0, `IPRESS`: 0, Allowed IMF-values: 0, Data required: None. - Case: 1a, `IEJ`: 1, `IPRESS`: 0, Allowed IMF-values: 0, Data required: `EI, GAs`. - Case: 1b, `IEJ`: 1, `IPRESS`: 0, Allowed IMF-values: 1, Data required: `EI, MF`. - Case: 2, `IEJ`: 1, `IPRESS`: 1, Allowed IMF-values: 0, Data required: Not implemented. - Case: 3, `IEJ`: N, `IPRESS`: 0, Allowed IMF-values: 0, 1, Data required: `CURV(I): I=1,N. BMOM(I): I=1,N`. - Case: 4, `IEJ`: N1, `IPRESS`: N2, Allowed IMF-values: 0, Data required: Not implemented.

5.4.2.9. Bending stiffness. Case 1a, IEJ=1 IPRESS=0 IMF=0

```
EI GAs
```

- `EI: real > 0`: Bending stiffness $[\mathrm{FL}^2]$
- `GAs: real`: Shear stiffness $[\mathrm{F}]$

The shear stiffness, `GAs`, is an optional input parameter. Specified `GAs > 0` will include shear deformation. This requires that all stiffness properties are constant, i.e. `IEA = 1`, `IEJ = 1`, `IGT = 1`

5.4.2.10. Bending stiffness. Case 1b, IEJ=1 IPRESS=0 IMF=1

```
EI MF SF
```

- `EI: real`: Bending stiffness $[\mathrm{FL}^2]$
- `MF: real`: Internal friction moment, see figure below. $[\mathrm{FL}]$
- `SF: real, default: 10.`: Internal friction moment stiffness factor. $[1]$

The default value of `SF` corresponds to the earlier fixed value of 10.0.

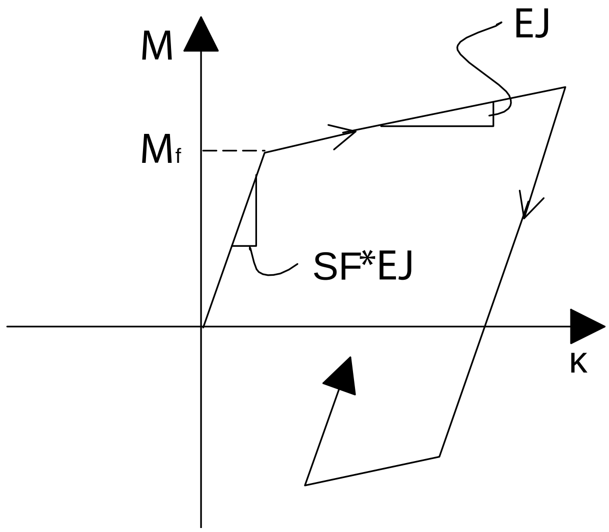

Figure 22. Internal friction moment description

5.4.2.11. Bending stiffness. Case 2, IEJ=1 IPRESS=1 (Not implemented)

`EI(1) PRESS(1) EI(2) PRESS(2) MF(1) MF(2)`

- `EI(1): real`: Bending stiffness $[\mathrm{FL}^2]$
- `PRESS(1): real`: Pressure at which the above values apply $[\mathrm{F/L}^2]$
- `EI(2): real`: See description above
- `PRESS(2):`
- `MF(1): real`: Internal friction moment for pressure PRESS(1)
- `MF(2): real`: Internal friction moment for pressure PRESS(2)

`PRESS(1) < PRESS(2)`

`MF(1) and MF(2) dummy for IMF = 0`

Bending stiffness around y-axis as function of pressure

Values at other pressure levels than `PRESS(1)` and `PRESS(2)` are obtained by linear interpolation/ extrapolation.

5.4.2.12. Bending stiffness description. Case 3 IEJ=N IPRESS=0

Tabulated curvature/bending moment relation. This specification consists of two different input lines. For `IMF` $\neq$ `0` cfr. Bending stiffness. Case 4…

5.4.2.12.1. Curvature

`CURV(1) ... CURV(N)`

- `CURV(1): real`: Curvature values for which bending moment is specified $[1/\mathrm{L}]$
- .
- .
- .
- `CURV(N)`: To be specified in increasing order

`CURV=1/CURVATURE RADIUS`

5.4.2.12.2. Bending moment, y-axis

`BMOMY(1) BMOMY(N)`

- `BMOMY(1): real`: Bending moment around y-axis $[\mathrm{FL}]$ corresponding to curvature values given above in `Curvature'.
- `BMOMY(N)`

`CURV(1), BMOMY(1)` have to be zero. Positive slope required, i.e.: `BMOMY(I+1) > BMOMY(I)`.

Bending moment around y-axis as function of curvature

5.4.2.13. Bending stiffness. Case 4 IEJ=N1, IPRESS=N2 (Not implemented)

This specification consists of three different input lines.

5.4.2.13.1. Curvature

`CURV(1) ... CURV(N)`

- `CURV(1): real`: Curvature values for which bending moment is specified $[1/\mathrm{L}]$
- .
- .
- .
- `CURV(N)`: To be specified in increasing order

`CURV=1/CURVATURE RADIUS`

CURV(1) has to be zero

5.4.2.13.2. Pressure

`PRESS(1) ... PRESS(N)`

- `PRESS(1): real`: Pressure levels for which bending moment is specified $[\mathrm{F/L^2}]$
- `PRESS(N)`:

5.4.2.13.3. Bending moment, y-axis

`BMOMY(1,1) BMOMY(N1,N2)`

- `BMOMY(1,1): real`: Bending moment at curvature I and pressure J $[\mathrm{FL}]$.
- `BMOMY(N1,N2)`

`BMOMY(1,J), J=1,N2` have to be zero, see also the figure below. Positive slope with increasing curvature is required, i.e.: `BMOMY(I+1,J) > BMOMY(I,J)`.

Bending moment around y-axis as function of curvature and pressure

5.4.2.14. Torsion stiffness

No data required for `IGT=0`.

5.4.2.14.1. Constant torsion stiffness. Case 1 |IGT|=1

`GT- GT+`

- `GT-: real > 0`: Torsion stiffness $[\mathrm{FL^2/Radian}]$
- `GT+: real`: D.o. for positive twist. Dummy if `IGT=1`

5.4.2.14.2. Nonlinear torsion stiffness. Case 2 |IGT|=N

`TMOM(1) TROT(1) . . . TMOM(N) TROT(N)`

- `TMOM(1): real`: Torsion moment $[\mathrm{FL}]$
- `TROT(1): real`: Torsion angle/length $[\mathrm{Radian/L}]$
- .
- .
- .

- `TMOM(N)`:
- `TROT(N): real`:

If `IGT` is positive `TMOM(1)` and `TROT(1)` have to be zero. `TROT` must be given in increasing order.

5.4.2.15. Damping specification

This data group is optional. It enables the user to specify cross sectional damping properties of the following types: - mass proportional damping - stiffness proportional damping - axial damping properties

Specification of mass and stiffness proportional damping specification will overrule corresponding damping specification given on global level as input to Dynmod data group Time integration and damping parameters.

5.4.2.15.1. Data group identifier and selection of damping types

`DAMP CHTYPE1 CHTYPE2 CHTYPE3 CHTYPE4`

- `DAMP: character(4)`: Data group identifier (the text ``DAMP")
- `CHTYPE1: character(5)`:
  - `=MASPR: Mass proportional damping
  - `=STFPR: Stiffness proportional damping
  - `=AXDMP: Local axial damping model
  - `=AXFRC: Local axial friction model
- `CHTYPE2: character(5)`: Similar to CHTYPE1
- `CHTYPE3: character(5)`: Similar to CHTYPE1
- `CHTYPE4: character(5)`: Similar to CHTYPE1

Between one and four damping types may be selected. The order of the damping type selection is arbitrary.

In the following the damping parameters for the selected damping types is described. The input lines have to be given in one block and in the order described below. Skip input for damping types which are not selected.

5.4.2.15.2. Parameters for mass proportional damping, if MASPR is specified

`A1T A1TO A1B`

- `A1T: real`: Factor for mass proportional damping in axial dofs.
- `A1TO: real, default: A1T`: Factor for mass proportional damping in torsional dofs.
- `A1B: real, default: A1TO`: Factor for mass proportional damping in bending dofs.

The element stiffness proportional damping matrix is computed by:

$$\mathbf{c_m} = \mathrm{a_{1t}}\mathbf{m}_\mathrm{t} + \mathrm{a_{1to}}\mathbf{m}_\mathrm{to} + \mathrm{a_{1b}}\mathbf{m}_\mathrm{b}$$

where $\mathbf{m}$ is the local stiffness matrix and the subscripts `t'',` to'' and ``b'' refer to axial, torsional and bending contributions, respectively.

5.4.2.15.3. Parameters for stiffness proportional damping, if STFPR is specified

`A2T A2TO A2B  DAMP_OPT`

- `A2T: real`: Factor for stiffness proportional damping in axial dofs.
- `A2TO: real, default: A2T`: Factor for stiffness proportional damping in torsional dofs.
- `A2B: real, default: A2TO`: Factor for stiffness proportional damping in bending dofs.
- `DAMP_OPT: character(4), default: TOTA`: Option for stiffness contribution to Rayleigh damping
    - = TOTA: Stiffness proportional damping is applied using total stiffness, i.e. both material and geometric stiffness
    - = MATE: Stiffness proportional damping is applied using material stiffness only

The element stiffness proportional damping matrix is computed by:

$$\mathbf{c_k} = \mathrm{a_{2t}}\mathbf{k}_\mathrm{t} + \mathrm{a_{2to}}\mathbf{k}_\mathrm{to} + \mathrm{a_{2b}}\mathbf{k}_\mathrm{b}$$

where $\mathbf{k}$ is the local stiffness matrix and the subscripts `t'',` to'' and ``b'' refer to axial, torsional and bending contributions, respectively.

5.4.2.15.4. Parameters for local axial damping, if AXDMP is specified

The local axial damping model is written:

$$\mathrm{F} = \mathrm{C}(\varepsilon) \times |\dot{\varepsilon}|^\mathrm{P} \times \mathrm{sign}(\dot{\varepsilon})$$

where: - F: damping force - C: damping coefficient (strain dependent) - ε: relative elongation - $\dot{\varepsilon}$: strain velocity - P: exponent for strain velocity (P >= 1)

`IDMPAXI EXPDMP`

- `IDMPAXI: integer`: Damping coefficient code
  - = 1: Constant damping coefficient
  - = N: Table with N pairs of damping coefficient - elongation to be specified.
  - N >= 2
- `EXPDMP: real`: Exponent for strain velocity

IDMPAXI = 1 ~ DMPAXI ~

- `DMPAXI: real`: Damping coefficient (constant)

IDMPAXI >1 ~ DMPAXI(1) ELONG(1) . . . . DMPAXI(IDMPAXI) ELONG(IDMPAXI) ~

- `DMPAXI(1): real`: Damping coefficient corresponding to relative elongation `ELONG(1)`
- `ELONG(1): real`: Relative elongation ( )

`ELONG` must be given in increasing order for the pairs of `DMPAXI` and `ELONG` . All pairs are given on a single input line

5.4.2.15.5. Parameters for local axial friction, if AXFRC is specified

`FRCAXI(1) ELONG(1) FRCAXI(2) ELONG(2)`

- `FRCAXI(1): real`: Static friction force corresponding to elongation `ELONG(1)`
- `ELONG(1): real`: Relative elongation ( )
- `FRCAXI(2): real, default: FRCAXI(1)`: Dynamic friction force corresponding to elongation `ELONG(2)`
- `ELONG(2): real, default: 1.1 x ELONG(1)`: Relative elongation ( )

`ELONG(2) > ELONG(1)`

5.4.2.16. Hydrodynamic load type identification, One optional input line

`CHLOAD`

- `CHLOAD: character`: = `HYDR` - Text to identify hydrodynamic coefficients

Note: Required if non-Morison loads are to be specified

5.4.2.16.1. Load type identification if CHLOAD=HYDR, One input line

`CHTYPE`

- `CHTYPE: character`: Type of load coefficients
  - `= NONE`: No hydrodynamic load coefficients
  - `= MORI`: Slender element hydrodynamic coefficients
  - `= MACF`: MacCamy-Fuchs with quadratic drag load coefficients
  - `= POTN`: Potential flow with quadratic drag load coefficients
  - `= TVIV`: Time domain VIV load coefficients. Restricted option.

Note that the option `POTN` currently is under testing. Potential flow forces are only available for irregular time domain analysis.

Note that the option `TVIV` is currently under development.

5.4.2.16.2. Hydrodynamic force coefficients if CHTYPE=MORI

Interpretation of hydrodynamic coefficients are dependent on the input parameter `ICODE`. Input of dimensional hydrodynamic coefficient is specified giving `ICODE=1` while input of nondimensional of hydrodynamic coefficients for circular cross sections is specified giving `ICODE=2`.

Definitions of dimensional/nondimensional hydrodynamic force coefficients are given below.

`CQX CQY CAX CAY CLX CLY ICODE D SCFKN SCFKT`

- `CQX: real`: Quadratic drag coefficient in tangential direction
  - `ICODE=1: CQX=CDX`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
  - `ICODE=2: CQX=Cdt`: nondimensional drag force coefficient
- `CQY: real`: Quadratic drag coefficient in normal direction
  - `ICODE=1: CQY=CDY`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
  - `ICODE=2: CQY=Cdn`: nondimensional drag force coefficient
- `CAX: real`: Added mass per unit length in tangential direction
  - `ICODE=1: CAX=AMX`: added mass $[\mathrm{M}/\mathrm{L}]$

  - `ICODE=2: CAX=Cmt`: nondimensional added mass coefficient

- `CAY: real`: Added mass per unit length in normal direction
  - `ICODE=1: CAY=AMY`: added mass $[\mathrm{M}/\mathrm{L}]$
  - `ICODE=2: CAY=Cmn`: nondimensional added mass coefficient
- `CLX: real`: Linear drag force coefficient in tangential direction
  - `ICODE=1: CLX=CDLX`: dimensional linear drag coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T}) \times \mathrm{L})]$
  - `ICODE=2: CLX=CdtL`: nondimensional linear drag force coefficient
- `CLY: real`: Linear drag force coefficient in normal direction
  - `ICODE=1: CLY=CDLY`: dimensional linear drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T}) \times \mathrm{L})]$
  - `ICODE=2:CLY=CdnL`: nondimensional linear drag force coefficient
- `ICODE: integer, default: 1`: `ICODE` Code for input of hydrodynamic force coefficients
  - `ICODE=1`: Dimensional coefficients
  - `ICODE=2`: Nondimensional coefficients
- `D: real, default:`$\sqrt{\frac{4}{\pi}(\mathrm{AE})}$: Hydrodynamic diameter of the pipe $[\mathrm{L}]$.
  - Default value is calculated from external cross-sectional area given as input in data section Mass and volume
  - Note that the hydrodynamic diameter is a key parameter in `VIVANA`. `D` is dummy in the other modules for `ICODE=1` unless marine growthis apploed.
- `SCFKN: real, default: 1`: Scaling factor for the Froude-Krylov term in Morison’s equation in normal direction
- `SCFKT: real, default: 1`: Scaling factor for the Froude-Krylov term in Morison’s equation in tangential direction. Only the values 0.0 and 1.0 are permitted.

5.4.2.16.3. Definition of hydrodynamic force coefficients

The tangential force which is a friction force per unit length acting in local x-axis, Ft is computed by:

$$\mathrm{Ft} = \mathrm{CDX} \times \mathrm{VRELX} \times |\mathrm{VRELX}| + \mathrm{CDLX} \times \mathrm{VRELX}$$

The drag force per unit length acting normal to the local x-axis, $\mathrm{F_n}$, is computed by assuming that the instantaneous drag force direction is parallel to the instantaneous transverse relative velocity component:

$$\mathrm{F_n} = \mathrm{CDY}(\mathrm{VRELY}^2 + \mathrm{VRELZ}^2) + \mathrm{CDLY} \times \sqrt{\mathrm{VRELY}^2 + \mathrm{VRELZ}^2}$$

where: - $\mathrm{CDX}, \mathrm{CDY}$: are the dimensional quadratic drag force coefficients in local x- and y-directions (i.e. tangential and normal directions) - $\mathrm{CDLX}, \mathrm{CDLY}$: are the dimensional linear drag force coefficients in local x- and y-directions - $\mathrm{VRELX}, \mathrm{VRELY}, \mathrm{VRELZ}$: are relative water velocities in local x,y and z-directions

The nondimensional hydrodynamic force coefficients for a circular cross section are defined according to the following expressions: - $\mathrm{CDX} = \frac{1}{2}\rho \mathrm{S_W} \mathrm{C_{dt}}$ - $\mathrm{CDY} = \frac{1}{2}\rho \mathrm{D} \mathrm{C_{dn}}$ - $\mathrm{CDLX} = \rho\sqrt{\mathrm{g S_W}} \times \mathrm{S_W^2} \mathrm{C_{dt}^L}$ - $\mathrm{CDLY} = \rho\sqrt{\mathrm{gD}} \times \mathrm{D^2} \mathrm{C_{dt}^L}$ - $\mathrm{AMX} = \rho\frac{\pi \mathrm{D}^2}{4}\mathrm{C_{mt}}$ - $\mathrm{AMY} = \rho\frac{\pi \mathrm{D}^2}{4}\mathrm{C_{mn}}$

where:

- ρ: water density
- g: acceleration of gravity
- $\mathrm{S_W}$: cross sectional wetted surface $(= \pi \mathrm{D})$
- D: hydrodynamic diameter of the pipe
- $\mathrm{C_{dt}}$: nondimensional quadratic tangential drag coefficient
- $\mathrm{C_{dn}}$: nondimensional quadratic normal drag coefficient
- $\mathrm{C_{dt}^L}$: nondimensional linear tangential drag coefficient
- $\mathrm{C_{dn}^L}$: nondimensional linear normal drag coefficient
- $\mathrm{C_{mt}}$: nondimensional tangential added mass coefficient
- $\mathrm{C_{mn}}$: normal added mass coefficient
  - ($\mathrm{C_{mn}}$ is normally equal to 1.0 for a circular cross section)

5.4.2.16.4. Hydrodynamic force coefficients if CHTYPE=MACF

MacCamy-Fuchs frequency-dependent hydrodynamic loads on a stationary vertical circular cylinder will be applied for `CHTYPE=MACF`. MacCamy-Fuchs forces are pre-computed based on the element position after static calculation. MacCamy-Fuchs forces are only available for irregular time domain analysis.

Quadratic drag may also be applied on cross-sections with MacCamy-Fuchs loading.

======= Hydrodynamic force coefficients

`CQX CQY CAX ICODE D`

- `CQX: real`: Quadratic drag coefficient in tangential direction
  - `ICODE=1: CQX=CDX`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
  - `ICODE=2: CQX=Cdt`: nondimensional drag force coefficient
- `CQY: real`: Quadratic drag coefficient in normal direction
  - `ICODE=1: CQY=CDY`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
  - `ICODE=2: CQY=Cdn`: nondimensional drag force coefficient
- `CAX: real, default: 0.0`: Added mass per unit length in tangential direction
  - `ICODE=1: CAX=AMX`: added mass $[\mathrm{M}/\mathrm{L}]$
  - `ICODE=2: CAX=Cmt`: nondimensional added mass coefficient
- `ICODE: integer`: Code for input of hydrodynamic drag coefficients
  - `ICODE=1`: Dimensional coefficients
  - `ICODE=2`: Nondimensional coefficients
- `D: real, default:`$\sqrt{\frac{4}{\pi}(\mathrm{AE})}$: Hydrodynamic diameter of the pipe $[\mathrm{L}]$.
  - Default value is calculated from external cross-sectional area given as input in data section Mass and volume

======= Simplified radiation force

The horizontal radiation loads is based on an added mass coefficient and a damping coefficient. ~ AMY DAMP ~

- `AMY: real, default: 0.0`: Added mass per unit length in normal direction $[\mathrm{M}/\mathrm{L}]$
- `DAMP: real, default: 0.0`: Damping in normal direction

Note: The input CHTYPE=MACF is extended in Riflex 4.13 and is not compatible with earlier versions of Riflex.

5.4.2.16.5. Hydrodynamic force coefficients if CHTYPE=POTN

Frequency-dependent added mass, radiation damping, and excitation forces based on the first order potential flow solution will be applied for `CHTYPE=POTN`. The radiation and diffraction coefficients are to be given by a separate input file specified under the data group Potential flow library specification (@ref inpmod_b_poten_lib).

Quadratic drag may also be applied on cross-sections with potential flow loading.

`CQX CQY ICODE D SCFKT`

- `CQX: real`: Quadratic drag coefficient in tangential direction
    - `ICODE=1: CQX=CDX`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
    - `ICODE=2: CQX=Cdt`: nondimensional drag force coefficient
- `CQY: real`: Quadratic drag coefficient in normal direction
    - `ICODE=1: CQY=CDY`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
    - `ICODE=2: CQY=Cdn`: nondimensional drag force coefficient
- `ICODE: integer, default: 1`: `ICODE` Code for input of hydrodynamic force coefficients
    - `ICODE=1`: Dimensional coefficients
    - `ICODE=2`: Nondimensional coefficients
- `D: real, default:`$\sqrt{\frac{4}{\pi}(\mathrm{AE})}$: Hydrodynamic diameter of the pipe $[\mathrm{L}]$.
    - Default value is calculated from external cross-sectional area given as input in data section Mass and volume
- `SCFKT: real, default: 1`: Scaling factor for the Froude-Krylov term in Morison’s equation in tangential direction. Only the values 0.0 and 1.0 are permitted.

5.4.2.16.6. Hydrodynamic force coefficients if CHTYPE=TVIV

Restricted option. Under implementation.

`CQX CQY CAX CAY CLX CLY ICODE D SCFKN SCFKT`

See the description above for Morison coefficients.

`CV FNULL FMIN FMAX NMEM CVIL ALPHIL CHH`

`CV: real > 0`: Vortex shedding force coefficient for the (instantaneous) cross-flow load term (nondimensional)

- `FNULL: real > 0`: Natural vortex shedding frequency (nondimensional)
- `FMIN: real > 0`: Minimum vortex shedding frequency (nondimensional)
- `FMAX: real > FMIN`: Maximum vortex shedding frequency (nondimensional)
- `NMEM: integer > 0`: Number of time steps used in calculation of standard deviation
- `CVIL: real >= 0, default: 0.0`: Vortex shedding force coefficient for (instantaneous) in-line load term (nondimensional)
- `ALPHIL: real >= 0, default: 0.0`: Nondimensional parameter giving freedom to in-line excitation frequency
- `CHH: real >= 0, default: 0.0`: Higher harmonic load coefficient (nondimensional)

Specifying `CVIL`, `ALPHIL` and `CHH` as zero will give excitation only in the updated cross-flow direction.

5.4.2.17. Aerodynamic load type identification, One optional input line

`CHLOAD`

- `CHLOAD: character`: = `WIND` - Text to identify wind coefficients

5.4.2.17.1. Load type identification if CHLOAD=WIND, One input line

`CHTYPE`

- `CHTYPE: character`: Type of load coefficients
  - `= MORI`: Morison-like loading, Drag term

5.4.2.17.2. Drag coefficients if CHTYPE=MORI, One input line

`CDXAERO CDYAERO ICODE D`

- `CDXAERO: real`: Quadratic drag coefficient in tangential direction
  - `ICODE=1: CDXAERO=CDXa`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
  - `ICODE=2: CDXAERO=Cdta`: non-dimensional drag force coefficient
- `CDYAERO: real`: Quadratic drag coefficient in normal direction

    - `ICODE=1: CDYAERO=CDYa`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
    - `ICODE=2: CDYAERO=Cdna`: non-dimensional drag force coefficient

- `ICODE: integer, default: 1`: Code for input of aerodynamic force coefficients
    - `ICODE=1`: Dimensional coefficients
    - `ICODE=2`: Nondimensional coefficients

- `D: real, default:`$\sqrt{\frac{4}{\pi}(\mathrm{AE})}$: Aerodynamic diameter of the pipe $[\mathrm{L}]$.
    - Default value is calculated from external cross-sectional area given as input in data section Mass and volume
    - Dummy for `ICODE=1`

The tangential force which is a friction force per unit length acting in local x-axis, $\mathrm{F_t}$ is computed by:

$$\mathrm{F_t} = \mathrm{CDXa} \times \mathrm{VRELX} \times |\mathrm{VRELX}|$$

The drag force per unit length acting normal to the local x-axis, $\mathrm{F_n}$, is computed by assuming that the instantaneous drag force direction is parallel to the instantaneous transverse relative velocity component:

$$\mathrm{F_n} = \mathrm{CDYa}(\mathrm{VRELY}^2 + \mathrm{VRELZ}^2)$$

where: - $\mathrm{CDXa}, \mathrm{CDYa}$: are the dimensional quadratic drag force coefficients in local x- and y-directions (i.e. tangential and normal directions) - $\mathrm{VRELX}, \mathrm{VRELY}, \mathrm{VRELZ}$: are relative wind velocities in local x,y and z-directions

The nondimensional aerodynamic force coefficients for a circular cross section are defined according to the following expressions: - $\mathrm{CDXa} = \frac{1}{2}\rho_\mathrm{a}\mathrm{S_W}\mathrm{C_{dta}}$ - $\mathrm{CDYa} = \frac{1}{2}\rho_\mathrm{a}\mathrm{D}\mathrm{C_{dna}}$

where:

- ρ_a: air density
- $\mathrm{S_W}$: cross sectional perimeter $(= \pi\mathrm{D})$
- D: aerodynamic diameter of the pipe
- $\mathrm{C_{dta}}$: nondimensional quadratic tangential drag coefficient
- $\mathrm{C_{dna}}$: nondimensional quadratic normal drag coefficient

5.4.2.18. Capacity parameter

`TB YCURMX`

- `TB: real`: Tension capacity $[\mathrm{F}]$
- `YCURMX: real`: Maximum curvature $[1/\mathrm{L}]$

These parameters are dummy in the present version

5.4.3. CRS2 - Double symmetric cross section

5.4.3.1. Data group identifier

`NEW COMPonent CRS2`

5.4.3.2. Component type identifier

`CMPTYP-ID TEMP`

- `CMPTYP-ID: character(8)`: Component type identifier
- `TEMP: real`: Temperature at which the specification applies
  - Dummy in present version

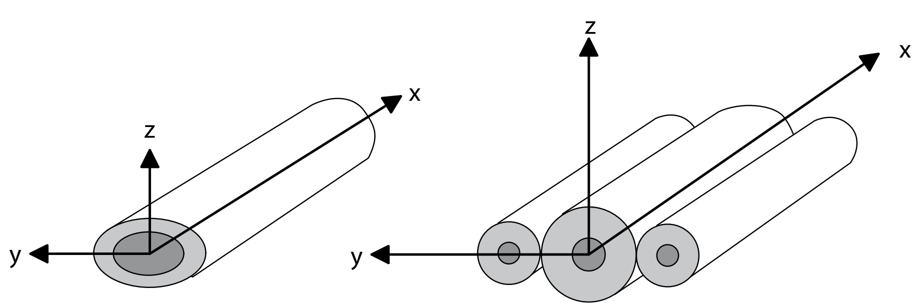

Figure 23. Cross section with 2 symmetry planes

5.4.3.3. Mass and volume

`AMS AE AI RGYR`

- `AMS: real`: Mass per unit length $[\mathrm{M}/\mathrm{L}]$
- `AE: real`: External cross-sectional area $[\mathrm{L}^2]$
- `AI: real`: Internal cross-sectional area $[\mathrm{L}^2]$

`RGYR: real`: Radius of gyration about local x-axis $[\mathrm{L}]$

`AI` is used to calculate additional mass of internal fluid if present. Otherwise `AI` is dummy.

Note that the mass center is located along the local X-axis, i.e. at the origin of the local Y- and Z-axes.

5.4.3.4. Stiffness properties classification

`IEA IEJ IGT IPRESS`

- `IEA: integer, default: 0`: Axial stiffness code
  - 1 - constant stiffness
  - N - table with N pairs of tension-elongation to be specified
  - N >= 2
- `IEJ: integer, default: 0`:
  - `0` - zero bending stiffness
  - 1 - constant stiffness
  - N - table with N pairs of bending moment - curvature to be specified.
  - N >= 2
- `IGT: integer, default: 0`: Torsion stiffness code
  - `0` - zero torsional stiffness
  - 1 - constant stiffness
  - -1- non-symmetric ``constant'' stiffness
  - N - symmetric, N (positive) pairs specified
  - -N- general torsion/relation (non-symmetric) N pairs specified
  - N >= 2
- `IPRESS: integer, default: 0`: Pressure dependency parameter related to bending moment
  - `0` - no pressure dependency
  - 1 - linear dependency (not implemented)

    - NP - NP sets of stiffness properties to be given, corresponding to a table of NP pressure values (not implemented)
        - 2 ⇐ NP ⇐ 10

Normally `IEJ` and `IGT` should both be zero or both greater than zero to assure stability in the `FEM` analysis.

`IPRESS=0` in this version of the program.

5.4.3.5. Bending-torsion geometric coupling specification

This data group is optional, and will only be applied for `IEJ=1` and `IGT=1`.

`BTGC`

- `BTGC: character(4)`: bending-torsion coupling identifier.

If the `BTGC` identifier is present, geometric coupling between torsion and bending is accounted for.

5.4.3.6. Axial stiffness. Case 1 IEA=1

`EA`

- `EA: real > 0`: Axial stiffness $[\mathrm{F}]$

5.4.3.7. Axial stiffness. Case 2 IEA=N

`EAF(1) ELONG(1) . . . EAF(N) ELONG(N)`

- `EAF(1): real`: Axial force corresponding to relative elongation `ELONG(1)` $[\mathrm{F}]$
- `ELONG(1): real`: Relative elongation ()
- .
- .
- .
- `EAF(N): real`:
- `ELONG(N): real`:

The pairs of `EAF` and `ELONG` must be given in increasing order. See also the figure Axial force corresponding to relative elongation.

5.4.3.8. Bending stiffness properties

The amount of input depends upon the parameters `IEJ` and `IPRESS` according to the table below: - Case: 0, `IEJ`: 0, `IPRESS`: 0, Data required: None. - Case: 1, `IEJ`: 1, `IPRESS`: 0, Data required: `EJY, EZJ, MFY, MF2`. - Case: 2, `IEJ`: 1, `IPRESS`: 1, Data required: Not implemented. - Case: 3, `IEJ`: N, `IPRESS`: 0, Data required: `CURV(I): I=1,N. BMOMY(I): I=1,N. BMOMZ(I)` - Case: 4, `IEJ`: N1, `IPRESS`: N2, Data required: Not implemented.

Thus, the following data are required for the respective cases:

5.4.3.9. Bending stiffness. Case 1, IEJ=1 IPRESS=0

```
EJY EJZ GAsZ GAsY
```

- `EJY: real > 0`: Bending stiffness around local y-axis $[\mathrm{FL}^2]$
- `EJZ: real > 0`: Bending stiffness around z-axis $[\mathrm{FL}^2]$
- `GAsZ: real`: Shear stiffness in Z-direction $[\mathrm{F}]$
- `GAsY: real`: Shear stiffness in Y-direction $[\mathrm{F}]$

The shear stiffness, `GAsZ` and `GAsY`, are optional input parameters.

Specified `GAsZ>0` and `GAsY>0` will include shear deformation. This requires that all stiffness properties are constant, i.e. `IEA = 1`, `IEJ = 1`, `IGT = 1`.

Note that the shear center is located along the local X-axis, i.e. at the origin of the local Y- and Z-axes.

5.4.3.10. Bending stiffness. Case 2, IEJ=1 IPRESS=1 (Not implemented)

```
EJY(1) EJZ(1) PRESS(1) EJY(2) EJZ(2) PRESS (2)
```

- `EJY(1): real`: Bending stiffness around local y-axis $[\mathrm{FL}^2]$
- `EJZ(1): real`: Bending stiffness around local z-axis $[\mathrm{FL}^2]$
- `PRESS(1): real`: Pressure at which the above values apply $[\mathrm{F/L}^2]$
- `EJY(2): real`: Bending moments corresponding to 2nd pressure level, see description above
- `EJZ(2): real`:
- `PRESS(2): real`:

`PRESS(1) < PRESS(2)`

Bending stiffness around y-axis as function of pressure.

Values at other pressure levels than `PRESS(1)` and `PRESS(2)` are obtained by linear interpolation/ extrapolation.

5.4.3.11. Bending stiffness description. Case 3 IEJ=N IPRESS=0

This specification consists of three different input lines.

5.4.3.11.1. Curvature

`CURV(1) ... CURV(N)`

- `CURV(1): real`: Curvature values for which bending moment is specified $[1/\mathrm{L}]$
- .
- .
- .
- `CURV(N): real`: To be specified in increasing order

`CURV=1/CURVATURE RADIUS`

5.4.3.11.2. Bending moment, y-axis

`BMOMY(1) . . . BMOMY(N)`

- `BMOMY(1): real`: Bending moment around local y-axis $[\mathrm{FL}]$
- .
- .
- .
- `BMOMY(N): real`

5.4.3.11.3. Bending moment, z-axis

`BMOMZ(1) . . . BMOMZ(N)`

- `BMOMZ(1): real`: Bending moment around local z-axis $[\mathrm{FL}]$

.

- .
- .
- `BMOMZ(N): real`

`CURV(1), BMOMY(1)` and `BMOMZ(1)` have to be zero. See also the figure Bending moment around y-axis as function of curvature.

5.4.3.12. Bending stiffness. Case 4 IEJ=N1, IPRESS=N2 (Not implemented)

This specification consists of four different input lines.

5.4.3.12.1. Curvature

```
CURV(1) ... CURV(N)
```

- `CURV(1): real`: Curvature values for which bending moments are specified $[1/\mathrm{L}]$
- .
- .
- .
- `CURV(N): real`: To be specified in increasing order

```
CURV=1/CURVATURE RADIUS
```

CURV(1) has to be zero. See also the figure Bending moment around y-axis as function of curvature and pressure.

5.4.3.12.2. Pressure

```
PRESS(1) ... PRESS(N)
```

- `PRESS(1): real`: Pressure levels for which bending moments are specified $[\mathrm{F}/\mathrm{L}^2]$
- .
- .
- .
- `PRESS(N): real`:

5.4.3.12.3. Bending moment, y-axis

`BMOMY(1,1) . . . BMOMY(N1,N2)`

- `BMOMY(I,J): real`: Bending moment about local y-axis at curvature I and pressure J $[\mathrm{FL}]$.
- .
- .
- .
- `BMOMY(N1,N2):real`:

`BMOMY(1,J), J=1, N2` have to be zero.

5.4.3.12.4. Bending moment, z-axis

`BMOMZ(1,1) . . . BMOMZ(N1,N2)`

- `BMOMZ(I,J): real`: Bending moment about local Z-axis at curvature I and pressure J $[\mathrm{FL}]$.
- .
- .
- .
- `BMOMZ(N1,N2):real`:

`BMOMZ(1,J), J=1, N2` have to be zero.

5.4.3.13. Torsion stiffness

5.4.3.13.1. Constant torsion stiffness. Case 1 |IGT|=1

`GT- GT+`

- `GT-: real > 0`: Torsion stiffness (negative twist) $[\mathrm{FL}^2/\mathrm{Radian}]$
- `GT+: real`: D.o. for positive twist. Dummy for `IGT=1`

5.4.3.13.2. Nonlinear torsion stiffness. Case 2 |IGT|= N

`TMOM(1) TROT(1) . . . TMOM(N) TROT(N)`

- `TMOM(1): real`: Torsion moment $[\mathrm{FL}]$
- `TROT(1): real`: Torsion angle/length $[\mathrm{Radian}/\mathrm{L}]$

If `IGT` is positive `TMOM(1)` and `TROT(1)` have to be zero. `TROT` must be given in increasing order.

5.4.3.14. Damping specification

Identical to input for cross-section type CRS1, see data group Damping specification (@ref inpmod_c_crs1_damping_specification).

5.4.3.15. Hydrodynamic load type identification, One input line

`CHLOAD`

- `CHLOAD: character: = HYDR` - Text to identify hydrodynamic coefficients

Note: Required if non-Morison loads are to be specified

5.4.3.15.1. Load type identification for CHLOAD=HYDR, One input line

`CHTYPE`

- `CHTYPE: character`: Type of load coefficients
  - `= NONE`: No hydrodynamic load coefficients
  - `= MORI`: Slender element hydrodynamic coefficients
  - `= MACF`: MacCamy-Fuchs with quadratic drag load coefficients
  - `= POTN`: Potential flow with quadratic drag load coefficients

Note that the option `POTN` currently is under testing. Potential flow forces are only available for irregular time domain analysis.

5.4.3.15.2. Hydrodynamic force coefficients if CHTYPE=MORI

`CDX CDY CDZ CDTMOM AMX AMY AMZ AMTOR CDLX CDLY CDLZ SCFKN SCFKT`

- `CDX: real`: Drag force coefficient for local x-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDY: real`: Drag force coefficient for local y-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDZ: real`: Drag force coefficient for local z-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDTMOM: real`: Drag force coefficient for local x-rotation. Not used in present version.

- `AMX: real`: Added mass per length in x-direction $[\mathrm{M/L}]$
- `AMY: real`: Added mass per length in y-direction $[\mathrm{M/L}]$
- `AMZ: real`: Added mass per length in z-direction $[\mathrm{M/L}]$
- `AMTOR: real`: Added mass for local x-rotation $[\mathrm{ML^2/L}]$ Not used in present version.
- `CDLX: real, default: 0`: Linear drag force coefficients in local x-direction $[\mathrm{F/((L/T) \times L)}]$
- `CDLY: real, default: 0`: Linear drag force coefficients in local y-direction $[\mathrm{F/((L/T) \times L)}]$
- `CDLZ: real, default: 0`: Linear drag force coefficients in local z-direction $[\mathrm{F/((L/T) \times L)}]$
- `SCFKN: real, default: 1`: Scaling factor for the Froude-Krylov term in Morison's equation in normal direction
- `SCFKT: real, default: 1`: Scaling factor for the Froude-Krylov term in Morison's equation in tangential direction. Only the values 0.0 and 1.0 are permitted.

The drag forces per unit length acting in the local coordinate system are computed as: - $\mathrm{F_x = CDX \times VRELX \times VRELX + CDLX \times VRELX}$ - $\mathrm{F_y = CDY \times \sqrt{VRELY^2 + VRELZ^2} \times VRELY + CDLY \times VRELY}$ - $\mathrm{F_z = CDZ \times \sqrt{VRELY^2 + VRELZ^2} \times VRELZ + CDLZ \times VRELZ}$

where: - $\mathrm{CDX, CDY, CDZ}$: are the input quadratic drag force coefficients in local x, y and z-directions - $\mathrm{CDLX, CDLY, CDLZ}$: are the input linear drag force coefficients in local x, y and z-directions - $\mathrm{VRELX, VRELY, VRELZ}$: are relative water velocities in local x, y and z-directions

The input quadratic drag force coefficients CDX, CDY and CDZ will normally be calculated as: - $\mathrm{CDX} = \frac{1}{2}\rho \mathrm{S_{2D} C_{dx}}$ - $\mathrm{CDY} = \frac{1}{2}\rho \mathrm{B_y C_{dy}}$ - $\mathrm{CDZ} = \frac{1}{2}\rho \mathrm{B_z C_{dz}}$

where: - ρ: water density - $\mathrm{S_{2D}}$: cross sectional wetted surface - $\mathrm{B_y, B_z}$: projected area per. unit length for flow in local y and z-direction, respectively - $\mathrm{C_{dx}, C_{dy}, C_{dz}}$: nondimensional drag coefficients in local x, y and z-directions, respectively

The input added mass per. unit length AMX, AMY and AMZ can be calculated as: - $\mathrm{AMX} = \rho \mathrm{AC_{mx}}$ - $\mathrm{AMY} = \rho \mathrm{AC_{my}}$ - $\mathrm{AMZ} = \rho \mathrm{AC_{mz}}$

where: - ρ: water density - A: cross sectional area - $\mathrm{C_{mx}, C_{my}, C_{mz}}$: nondimensional added mass coefficients in local x, y and z-directions, respectively

5.4.3.15.3. Hydrodynamic force coefficients if CHTYPE=MACF

MacCamy-Fuchs frequency-dependent hydrodynamic loads on a stationary vertical circular cylinder will be applied for `CHTYPE=MACF`. MacCamy-Fuchs forces are pre-computed based on the element position after static calculation. MacCamy-Fuchs forces are only available for irregular time domain analysis.

Quadratic drag may also be applied on elements with MacCamy-Fuchs loading. McCamy Fuchs assumes that the cross-section is circular, so a single transverse quadratic drag coefficient is given (CDZ will be set to CDY).

```
CQX CQY ICODE D
```

- `CQX: real`: Quadratic drag coefficient in tangential direction
  - `ICODE=1: CQX=CDX`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
  - `ICODE=2: CQX=Cdt`: nondimensional drag force coefficient
- `CQY: real`: Quadratic drag coefficient in normal direction
  - `ICODE=1: CQY=CDY`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
  - `ICODE=2: CQY=Cdn`: nondimensional drag force coefficient
- `ICODE: integer`: Code for input of hydrodynamic drag coefficients
  - `ICODE=1`: Dimensional coefficients
  - `ICODE=2`: Nondimensional coefficients
- `D: real, default:`$\sqrt{\frac{4}{\pi}(\mathrm{AE})}$: Hydrodynamic diameter of the pipe $[\mathrm{L}]$.
  - Default value is calculated from external cross-sectional area given as input in data section Mass and volume

5.4.3.15.4. Hydrodynamic force coefficients if CHTYPE=POTN

Frequency-dependent added mass, radiation damping, and excitation forces based on the first order potential flow solution will be applied for `CHTYPE=POTN`. The radiation and diffraction coefficients are to be given by a separate input file specified under the data group Potential flow library specification (@ref inpmod_b_poten_lib).

Quadratic drag may also be applied on cross-sections with potential flow loading.

```
CQX CQY CQZ ICODE D SCFKT
```

- `CQX: real`: Quadratic drag coefficient in local x-direction
  - `ICODE=1: CQX=CDX`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
  - `ICODE=2: CQX=Cdt`: nondimensional drag force coefficient
- `CQY: real`: Quadratic drag coefficient in local y-direction
  - `ICODE=1: CQY=CDY`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
  - `ICODE=2: CQY=Cdn`: nondimensional drag force coefficient
- `CQZ: real`: Quadratic drag coefficient in local z-direction
  - `ICODE=1: CQZ=CDZ`: dimensional drag force coefficient $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
  - `ICODE=2: CQZ=Cdn`: nondimensional drag force coefficient
- `ICODE: integer, default: 1`: `ICODE` Code for input of hydrodynamic force coefficients
  - `ICODE=1`: Dimensional coefficients
  - `ICODE=2`: Nondimensional coefficients
- `D: real, default:`$\sqrt{\frac{4}{\pi}(\mathrm{AE})}$: Hydrodynamic diameter $[\mathrm{L}]$.
  - Default value is calculated from external cross-sectional area given as input in data section Mass and volume
- `SCFKT: real, default: 1`: Scaling factor for the Froude-Krylov term in Morison's equation in tangential direction. Only the values 0.0 and 1.0 are permitted.

5.4.3.16. Aerodynamic load type identification, One optional input line

`CHLOAD`

- `CHLOAD: character:= WIND` - Text to identify wind coefficients
- `CHTYPE: character`: Type of wind load coefficients
  - `= MORI`: Morison-like loading, Drag term
  - `= AIRC`: Air foil cross section to be specified (Not implemented)
  - `= AIRF`: Air foil cross section, Refers to a air foil library file

5.4.3.17. Load type identification, One optional input line

CHTYPE

- `CHTYPE: character`: Type of wind load coefficients
  - `= MORI`: Morison-like loading, Drag term
  - `= AIRC`: Air foil cross section to be specified (Not implemented)
  - `= AIRF`: Air foil cross section, Refers to a air foil library file

5.4.3.17.1. CHTYPE=MORI: Morison-like aerodynamic drag, One input line

CDXAERO CDYAERO CDZAERO

- `CDXAERO: real`: Dimensional quadratic drag coefficient for local x-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDYAERO: real`: Dimensional quadratic drag coefficient for local y-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDZAERO: real`: Dimensional quadratic drag coefficient for local z-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$

The drag forces per unit length acting in the local coordinate system are computed as: - $\mathrm{F_x} = \mathrm{CDXAERO} \times \mathrm{VRELX} \times |\mathrm{VRELX}|$ - $\mathrm{F_y} = \mathrm{CDYAERO} \times \mathrm{VRELY} \times \sqrt{\mathrm{VRELY}^2 + \mathrm{VRELZ}^2}$ - $\mathrm{F_z} = \mathrm{CDZAERO} \times \mathrm{VRELZ} \times \sqrt{\mathrm{VRELY}^2 + \mathrm{VRELZ}^2}$

where: - $\mathrm{CDXAERO}, \mathrm{CDYAERO}, \mathrm{CDZAERO}$: are the input quadratic drag force coefficients in local x, y and z-directions - $\mathrm{VRELX}, \mathrm{VRELY}, \mathrm{VRELZ}$: are relative wind velocities in local x, y and z-directions

The input quadratic drag force coefficients CDX, CDY and CDZ will normally be calculated as: - $\mathrm{CDXAERO} = \frac{1}{2}\rho_{\mathrm{air}}\mathrm{S_{2D}}\mathrm{C_{dx}}$ - $\mathrm{CDYAERO} = \frac{1}{2}\rho_{\mathrm{air}}\mathrm{B_y}\mathrm{C_{dy}}$ - $\mathrm{CDZAERO} = \frac{1}{2}\rho_{\mathrm{air}}\mathrm{B_z}\mathrm{C_{dz}}$

where: - ρ_{air}: air density - $\mathrm{S_{2D}}$: cross sectional surface area - $\mathrm{B_y}, \mathrm{B_z}$: projected area per. unit length for flow in local y and z-direction, respectively - $\mathrm{C_{dx}}, \mathrm{C_{dy}}, \mathrm{C_{dz}}$: nondimensional drag coefficients in local x, y and z-directions, respectively

If the component is part of a wind turbine tower line, only the `CDY` component is used for tower shadow computation.

5.4.3.17.2. CHTYPE=AIRF: Coefficients on file. ID and chord length, One input line

CHCOEF CHORDL YFC ZFC ROTFAX

- `CHCOEF: character`: Air foil coefficient identifier. Must be found on the air foil library file
- `CHORDL: real`: Chord length of foil section. $[\mathrm{L}]$
    - It is used to scale the air foil load coefficients.
- `YFC: real, default: 0`: Y-coordinate of foil origin $[\mathrm{L}]$
- `ZFC: real, default: 0`: Z-coordinate of foil origin $[\mathrm{L}]$
- `ROTFAX: real, default: 0`: Inclination of foil system $[\mathrm{deg}]$

The blade coordinate system and origin coincides with the elastic (local) $(\mathrm{X_L}, \mathrm{Y_L}, \mathrm{Z_L})$ coordinate system. The aerodynamic coordinate system $(\mathrm{X_{AF}}, \mathrm{Y_{AF}})$ is located at (YFC,ZFC) referred to the local coordinate system, and is rotated about the blade x axis by the angle ROTFAX, as indicated in the figure below. The $\mathrm{X_L}$ axis is pointing into the paper plane, while the $\mathrm{Z_{AF}}$ is pointing out of plane. Note that the air foil coefficients has to be referred to the aerodynamic coordinate system as indicated by the corresponding angle of attack in the figure. For airfoil elements that are part of a wind turbine blade, the local $\mathrm{X_L}$-axis is pointing towards the blade tip.

Note that suppliers of wind turbine blades normally give the foil twist relative to the the areodynamic coordinate system, i.e. as twist around the $\mathrm{Z_{AF}}$ -axis.

Definition of foil center and inclination of foil system in the local cross section (strength

In coupled analysis, a `SIMO` wind type with `IWITYP >= 10` must be used if the case contains elements with wind force coefficients that are not on the blades of a wind turbine.

5.4.3.18. Capacity parameter

```
TB YCURMX ZCURMX
```

- `TB: real`: Tension capacity $[\mathrm{F}]$
- `YCYRMX: real`: Maximum curvature around local y-axis $[1/\mathrm{L}]$
- `ZCURMX: real`: Maximum curvature around local z-axis $[1/\mathrm{L}]$

These parameters are dummy in the present version

5.4.4. CRS7 - General cross section

5.4.4.1. Data group identifier

```
NEW COMPonent CRS7
```

5.4.4.2. Component type identifier

```
CMPTYP-ID TEMP ALFA
```

- `CMPTYP-ID: character(8)`: Component type identifier
- `TEMP : real`: Temperature at which the specification applies
    - Dummy in present version
- `ALPHA: real`: Thermal expansion coefficient $[\mathrm{Temp}^{-1}]$
    - Dummy in present version

General cross-section

5.4.4.3. Mass

```
YECC_MASS ZECC_MASS
```

- `YECC_MASS: real`: Mass center coordinate $\mathrm{Y_m}$ in beam element system $[\mathrm{L}]$
- `ZECC_MASS: real`: Mass center coordinate $\mathrm{Z_m}$ in beam element system $[\mathrm{L}]$

```
AMS RGYR
```

- `AMS : real`: Mass per unit length $[\mathrm{M/L}]$
- `RGYR: real`: Radius of gyration about mass center $(\mathrm{Y_m}, \mathrm{Z_m})$ $[\mathrm{L}]$

5.4.4.4. Buoyancy

```
YECC_BUOY ZECC_BUOY
```

- `YECC_BUOY: real`: Buoyancy center Y-coordinate in beam element system $[\mathrm{L}]$
    - Dummy in present version. Bouyancy center set equal to mass center.
- `ZECC_BUOY: real`: Buoyancy center Z-coordinate in beam element system $[\mathrm{L}]$
    - Dummy in present version. Bouyancy center set equal to mass center.

```
AE AI
```

- `AE: real`: External cross-sectional area $[\mathrm{L}^2]$
    - Basis for calculation of buoyancy

`AI: real`: Internal cross-sectional area $[\mathrm{L}^2]$

- Dummy in present version

5.4.4.5. Stiffness properties

Only constant stiffness properties are allowed.

5.4.4.6. Area center and principal axes

The area center is the cross-section point where the axial force acts through. The principal axes are formally determined from the requirement $\int_A V\,W\ \mathrm{d}A = 0$, where V and W denote the principal coordinates and A is the cross-section area. The orientation of the principal axes is defined in terms of a positive X-rotation θ relative to the beam element YZ-coordinate system as shown in the figure General cross-section ~ YECC_AREACENT ZECC_AREACENT THETA ~

- `YECC_AREACENT: real`: Area center coordinate $\mathrm{Y_a}$ in beam element system $[\mathrm{L}]$
- `ZECC_AREACENT: real`: Area center coordinate $\mathrm{Z_a}$ in beam element system $[\mathrm{L}]$
- `THETA: real`: Orientation θ of principal axes V and W [deg.]. See figure General cross-section.

5.4.4.7. Shear center

The shear center represents the attack point of the shear forces.

`YECC_SHEARCENT ZECC_SHEARCENT`

- `YECC_SHEARCENT: real`: Shear center coordinate $\mathrm{Y_s}$ in beam element system $[\mathrm{L}]$
- `ZECC_SHEARCENT: real`: Shear center coordinate $\mathrm{Z_s}$ in beam element system $[\mathrm{L}]$

5.4.4.8. Axial stiffness

`EA`

- `EA: real > 0`: Axial stiffness $[\mathrm{F}]$

5.4.4.9. Bending stiffness

The bending stiffness refers to the principal axes V and W, see figure General cross-section.

`EJV EJW`

- `EJV: real > 0`: Bending stiffness about principal V-axis $[\mathrm{FL}^2]$

- `EJW: real > 0`: Bending stiffness about principal W-axis $[\mathrm{FL}^2]$

5.4.4.10. Shear stiffness

The shear stiffness refers to the principal axes V and W, see figure General cross-section.

`GAsW GAsV`

- `GAsW: real`: Shear stiffness in principal W-direction $[\mathrm{F}]$
- `GAsV: real`: Shear stiffness in principal V-direction $[\mathrm{F}]$

The shear stiffness, `GAsW` and `GAsV`, are optional input parameters.

Specified `GAsW>0` and `GAsV>0` will include shear deformation.

5.4.4.11. Torsion stiffness

`GT`

- `GT: real > 0`: Torsion stiffness $[\mathrm{FL}^2/\mathrm{Radian}]$

For a circular cross-section the torsion stiffness is given by the polar moment of inertia. Note that this is not the case for non-circular cross-sections.

5.4.4.12. Bending-torsion geometric coupling

This data group is optional.

`BTGC`

- `BTGC: character(4)`: bending-torsion coupling identifier.

If the `BTGC` identifier is present, geometric coupling between torsion and bending is accounted for.

5.4.4.13. Damping specification

Identical to input for cross-section type CRS1, see data group Damping specification (@ref inpmod_c_crs1_damping_specification).

The stiffness matrix used as basis for the Rayleigh damping includes only the material stiffness matrix. The geometric stiffness matrix is not included as this would introduce damping of the rigid body motion for CRS7.

5.4.4.14. Hydrodynamic load type identification, One input line

CHLOAD

- `CHLOAD: character`: = `HYDR` - Text to identify hydrodynamic coefficients

Note: Required if non-Morison loads are to be specified

5.4.4.14.1. Load type identification for CHLOAD=HYDR, One input line

CHTYPE

- `CHTYPE: character`: Type of load coefficients
  - `= NONE`: No hydrodynamic load coefficients
  - `= MORI`: Slender element hydrodynamic coefficients

5.4.4.14.2. Hydrodynamic force coefficients if CHTYPE=MORI

CDX CDY CDZ CDTMOM AMX AMY AMZ AMTOR CDLX CDLY CDLZ SCFKN SCFKT

- `CDX: real`: Drag force coefficient for local x-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDY: real`: Drag force coefficient for local y-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDZ: real`: Drag force coefficient for local z-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDTMOM: real`: Drag force coefficient for local x-rotation. Not used in present version.
- `AMX: real`: Added mass per length in x-direction $[\mathrm{M}/\mathrm{L}]$
- `AMY: real`: Added mass per length in y-direction $[\mathrm{M}/\mathrm{L}]$
- `AMZ: real`: Added mass per length in z-direction $[\mathrm{M}/\mathrm{L}]$
- `AMTOR: real`: Added mass for local x-rotation $[\mathrm{ML}^2/\mathrm{L}]$
- `CDLX: real, default: 0`: Linear drag force coefficients in local x-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T}) \times \mathrm{L})]$
- `CDLY: real, default: 0`: Linear drag force coefficients in local y-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T}) \times \mathrm{L})]$
- `CDLZ: real, default: 0`: Linear drag force coefficients in local z-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T}) \times \mathrm{L})]$
- `SCFKN: real, default: 1`: Scaling factor for the Froude-Krylov term in Morison's equation in normal direction

`SCFKT: real, default: 1`: Scaling factor for the Froude-Krylov term in Morison's equation in tangential direction. Only the values 0.0 and 1.0 are permitted.

The drag forces per unit length acting in the local coordinate system are computed as: - $\mathrm{F_x = CDX \times VRELX \times VRELX + CDLX \times VRELX}$ - $\mathrm{F_y = CDY \times VRELY \times VRELY + CDLY \times VRELY}$ - $\mathrm{F_z = CDZ \times VRELZ \times VRELZ + CDLZ \times VRELZ}$

where: - $\mathrm{CDX, CDY, CDZ}$: are the input quadratic drag force coefficients in local x, y and z-directions - $\mathrm{CDLX, CDLY, CDLZ}$: are the input linear drag force coefficients in local x, y and z-directions - $\mathrm{VRELX, VRELY, VRELZ}$: are relative water velocities in local x, y and z-directions

The input quadratic drag force coefficients CDX, CDY and CDZ will normally be calculated as: - $\mathrm{CDX} = \frac{1}{2}\rho \mathrm{S_{2D} C_{dx}}$ - $\mathrm{CDY} = \frac{1}{2}\rho \mathrm{B_y C_{dy}}$ - $\mathrm{CDZ} = \frac{1}{2}\rho \mathrm{B_z C_{dz}}$

where: - ρ: water density - $\mathrm{S_{2D}}$: cross sectional wetted surface - $\mathrm{B_y, B_z}$: projected area per. unit length for flow in local y and z-direction, respectively - $\mathrm{C_{dx}, C_{dy}, C_{dz}}$: nondimensional drag coefficients in local x, y and z-directions, respectively

The input added mass per. unit length AMX, AMY and AMZ can be calculated as: - $\mathrm{AMX} = \rho \mathrm{AC_{mx}}$ - $\mathrm{AMY} = \rho \mathrm{AC_{my}}$ - $\mathrm{AMZ} = \rho \mathrm{AC_{mz}}$

where: - ρ: water density - A: cross sectional area - $\mathrm{C_{mx}, C_{my}, C_{mz}}$: nondimensional added mass coefficients in local x, y and z-directions, respectively

5.4.4.15. Aerodynamic load type identification, One optional input line

```
CHLOAD
```

- `CHLOAD: character`: = `WIND` - Text to identify wind coefficients

5.4.4.16. Load type identification, One optional input line

```
CHTYPE
```

- `CHTYPE: character`: Type of wind load coefficients
  - `= MORI`: Morison-like loading, Drag term
  - `= AIRC`: Air foil cross section to be specified (Not implemented)
  - `= AIRF`: Air foil cross section, Refers to a air foil library file

5.4.4.16.1. CHTYPE=MORI: Morison-like aerodynamic drag, One input line

```
CDXAERO CDYAERO CDZAERO
```

- `CDXAERO: real`: Dimensional quadratic drag coefficient for local x-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDYAERO: real`: Dimensional quadratic drag coefficient for local y-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDZAERO: real`: Dimensional quadratic drag coefficient for local z-direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$

The drag forces per unit length acting in the local coordinate system are computed as: - $\mathrm{F_x} = \mathrm{CDXAERO} \times \mathrm{VRELX} \times |\mathrm{VRELX}|$ - $\mathrm{F_y} = \mathrm{CDYAERO} \times \mathrm{VRELY} \times \sqrt{\mathrm{VRELY^2 + VRELZ^2}}$ - $\mathrm{F_z} = \mathrm{CDZAERO} \times \mathrm{VRELZ} \times \sqrt{\mathrm{VRELY^2 + VRELZ^2}}$

where: - $\mathrm{CDXAERO}, \mathrm{CDYAERO}, \mathrm{CDZAERO}$: are the input quadratic drag force coefficients in local x, y and z-directions - $\mathrm{VRELX}, \mathrm{VRELY}, \mathrm{VRELZ}$: are relative wind velocities in local x, y and z-directions

The input quadratic drag force coefficients CDX, CDY and CDZ will normally be calculated as: - $\mathrm{CDXAERO} = \frac{1}{2}\rho_{\mathrm{air}}\mathrm{S_{2D}}\mathrm{C_{dx}}$ - $\mathrm{CDYAERO} = \frac{1}{2}\rho_{\mathrm{air}}\mathrm{B_y}\mathrm{C_{dy}}$ - $\mathrm{CDZAERO} = \frac{1}{2}\rho_{\mathrm{air}}\mathrm{B_z}\mathrm{C_{dz}}$

where: - ρ_{air}: air density - $\mathrm{S_{2D}}$: cross sectional surface area - $\mathrm{B_y}, \mathrm{B_z}$: projected area per. unit length for flow in local y and z-direction, respectively - $\mathrm{C_{dx}}, \mathrm{C_{dy}}, \mathrm{C_{dz}}$: nondimensional drag coefficients in local x, y and z-directions, respectively

If the component is part of a wind turbine tower line, only the `CDY` component is used for tower shadow computation.

5.4.4.16.2. CHTYPE=AIRF: Coefficients on file. ID and chord length, One input line

```
CHCOEF CHORDL YFC ZFC ROTFAX
```

- `CHCOEF: character`: Air foil coefficient identifier. Must be found on the air foil library file
- `CHORDL: real`: Chord length of foil section. $[\mathrm{L}]$
  - It is used to scale the air foil load coefficients.
- `YFC: real, default: 0`: Y-coordinate of foil origin $[\mathrm{L}]$
- `ZFC: real, default: 0`: Z-coordinate of foil origin $[\mathrm{L}]$
-

`ROTFAX: real, default: 0`: Inclination of foil system $[\mathrm{deg}]$

The blade coordinate system and origin coincides with the elastic (local) $(\mathrm{X_L}, \mathrm{Y_L}, \mathrm{Z_L})$ coordinate system. The aerodynamic coordinate system $(\mathrm{X_{AF}}, \mathrm{Y_{AF}})$ is located at (YFC,ZFC) referred to the local coordinate system, and is rotated about the blade x axis by the angle ROTFAX, as indicated in the figure below. The $\mathrm{X_L}$ axis is pointing into the paper plane, while the $\mathrm{Z_{AF}}$ is pointing out of plane. Note that the air foil coefficients has to be referred to the aerodynamic coordinate system as indicated by the corresponding angle of attack in the figure. For airfoil elements that are part of a wind turbine blade, the local $\mathrm{X_L}$-axis is pointing towards the blade tip.

Note that suppliers of wind turbine blades normally give the foil twist relative to the the areodynamic coordinate system, i.e. as twist around the $\mathrm{Z_{AF}}$ -axis.

Definition of foil center and inclination of foil system in the local cross section (strength

In coupled analysis, a `SIMO` wind type with `IWITYP >= 10` must be used if the case contains elements with wind force coefficients that are not on the blades of a wind turbine.

5.4.4.17. Capacity parameter

```
TB YCURMX ZCURMX
```

- `TB: real`: Tension capacity $[\mathrm{F}]$
- `YCYRMX: real`: Maximum curvature around local y-axis $[1/\mathrm{L}]$
- `ZCURMX: real`: Maximum curvature around local z-axis $[1/\mathrm{L}]$

These parameters are dummy in the present version

5.4.5. BODY - Description of attached bodies

5.4.5.1. Data group identifier

```
NEW COMPonent BODY
```

5.4.5.2. Component type identifier

```
CMPTYP-ID
```

- `CMPTYP-ID: character(8)`: Component type identifier

A body is a component that may be attached at supernodes and segment interconnection points. The following essential properties should be observed: - The `BODY` is directly attached to a nodal point and has no motion degrees of freedom by itself. - The `BODY` component serves to add concentrated masses (inertia force), weight or buoyancy forces to the system.

5.4.5.3. Mass and volume

`AM AE`

- `AM: real`: Mass $[\mathrm{M}]$
- `AE: real`: Displacement volume $[\mathrm{L}^3]$

5.4.5.4. Hydrodynamic coefficients

`ICOO CDX CDY CDZ AMX AMY AMZ`

- `ICOO: character(5)`: Coordinate system code
  - `ICOO=GLOBAL`: Coefficients refer to global coordinate system
  - `ICOO=LOCAL`: Coefficients refer to local coordinate system of neighbour elements in the actual line
- `CDX: real`: Drag force coefficient in X-direction $[\mathrm{F}/(\mathrm{L}/\mathrm{T})^2)]$
- `CDY: real`: Drag force coefficient in Y-direction $[\mathrm{F}/(\mathrm{L}/\mathrm{T})^2)]$
- `CDZ: real`: Drag force coefficient in Z-direction $[\mathrm{F}/(\mathrm{L}/\mathrm{T})^2)]$
- `AMX: real`: Added mass in X-direction $[\mathrm{M}]$
- `AMY: real`: Added mass in Y-direction $[\mathrm{M}]$
- `AMZ: real`: Added mass in Z-direction $[\mathrm{M}]$

The drag forces acting in the global/local coordinate system are computed as: - $\mathrm{F_x = CDX \times VRELX \times VRELX}$ - $\mathrm{F_y = CDY \times VRELY \times VRELY}$ - $\mathrm{F_z = CDZ \times VRELZ \times VRELZ}$

where: - $\mathrm{CDX, CDY, CDZ}$: are the input drag force coefficients in global/local x, y and z-directions, respectively - $\mathrm{VRELX, VRELY, VRELZ}$: are relative water velocities in global/local x, y and z-directions respectively

The input quadratic drag force coefficients CDX, CDY and CDZ will normally be calculated as: - $\mathrm{CDX} = \frac{1}{2}\rho \mathrm{B_x C_{dx}}$ - $\mathrm{CDY} = \frac{1}{2}\rho \mathrm{B_y C_{dy}}$ - $\mathrm{CDZ} = \frac{1}{2}\rho \mathrm{B_z C_{dz}}$

where: - ρ: water density - $\mathrm{B_x, B_y, B_z}$: projected area for flow in global/local y and z-direction - $\mathrm{C_{dx}, C_{dy}, C_{dz}}$: nondimensional drag coefficients in global/local x, y and z-directions

5.4.6. CONB - Description of ball joint connectors

This component can be used to model balljoint, hinges and universal joints. The component has zero length, and adds 6 degrees of freedom to the system model. The forces due to mass and weight are assumed to act at the nodal point at which the component is specified. Note that this component can not be used in branch lines in standard systems, or in combination with bar elements. Should also be used with care at supernodes with user defined boundary conditions for rotations in `AR` system to avoid singularities in the `FEM` solution procedure.

5.4.6.1. Data group identifier

```
NEW COMPonent CONB
```

5.4.6.2. Component type identifier

```
CMPTYP-ID
```

- `CMPTYP-ID: character(8)`: Component type identifier

5.4.6.3. Mass and volume

```
AM AE
```

- `AM: real`: Mass $[\mathrm{M}]$
- `AE: real`: Displacement volume $[\mathrm{L}^3]$

5.4.6.4. Hydrodynamic coefficients

```
ICOO CDX CDY CDZ AMX AMY AMZ
```

- `ICOO: character`: Coordinate system code
    - `ICOO=GLOBAL`: Coefficients refer to global coordinate system
    - `ICOO=LOCAL`: Coefficients refer to local coordinate system of neighbour elements in the actual line
- `CDX: real`: Drag force coefficient in X-direction $[\mathrm{F}/(\mathrm{L}/\mathrm{T})^2)]$
- `CDY: real`: Drag force coefficient in Y-direction $[\mathrm{F}/(\mathrm{L}/\mathrm{T})^2)]$
- `CDZ: real`: Drag force coefficient in Z-direction $[\mathrm{F}/(\mathrm{L}/\mathrm{T})^2)]$
- `AMX: real`: Added mass in X-direction $[\mathrm{M}]$
- `AMY: real`: Added mass in Y-direction $[\mathrm{M}]$
- `AMZ: real`: Added mass in Z-direction $[\mathrm{M}]$

The drag forces acting in the global/local coordinate system are computed as: - $\mathrm{F_x = CDX \times VRELX \times VRELX}$ - $\mathrm{F_y = CDY \times VRELY \times VRELY}$ - $\mathrm{F_z = CDZ \times VRELZ \times VRELZ}$

where: - $\mathrm{CDX, CDY, CDZ}$: are the input drag force coefficients in global/local x, y and z-directions, respectively - $\mathrm{VRELX, VRELY, VRELZ}$: are relative water velocities in global/local x, y and z-directions respectively

The input quadratic drag force coefficients CDX, CDY and CDZ will normally be calculated as: - $\mathrm{CDX} = \frac{1}{2}\rho \mathrm{B_x C_{dx}}$ - $\mathrm{CDY} = \frac{1}{2}\rho \mathrm{B_y C_{dy}}$ - $\mathrm{CDZ} = \frac{1}{2}\rho \mathrm{B_z C_{dz}}$

where: - ρ: water density - $\mathrm{B_x, B_y, B_z}$: projected area per. unit lengt for flow in global/local y and z-directions, respectively - $\mathrm{C_{dx}, C_{dy}, C_{dz}}$: nondimensional drag coefficients in global/local x, y and z-directions, respectively

5.4.6.5. Degrees of freedom

```
IRX IRY IRZ
```

- `IRX: integer, default: 0`: Rotation freedom code, x-axis
- `IRY: integer, default: 0`: Rotation freedom code, y-axis
- `IRZ: integer, default: 0`: Rotation freedom code, z-axis
  - `1` - Fixed (no deformation)
  - `0` - Free (zero moment)

x-, y- and z-axes refer to local coordinate system of the neighbour element in the line where the ball joint is specified.

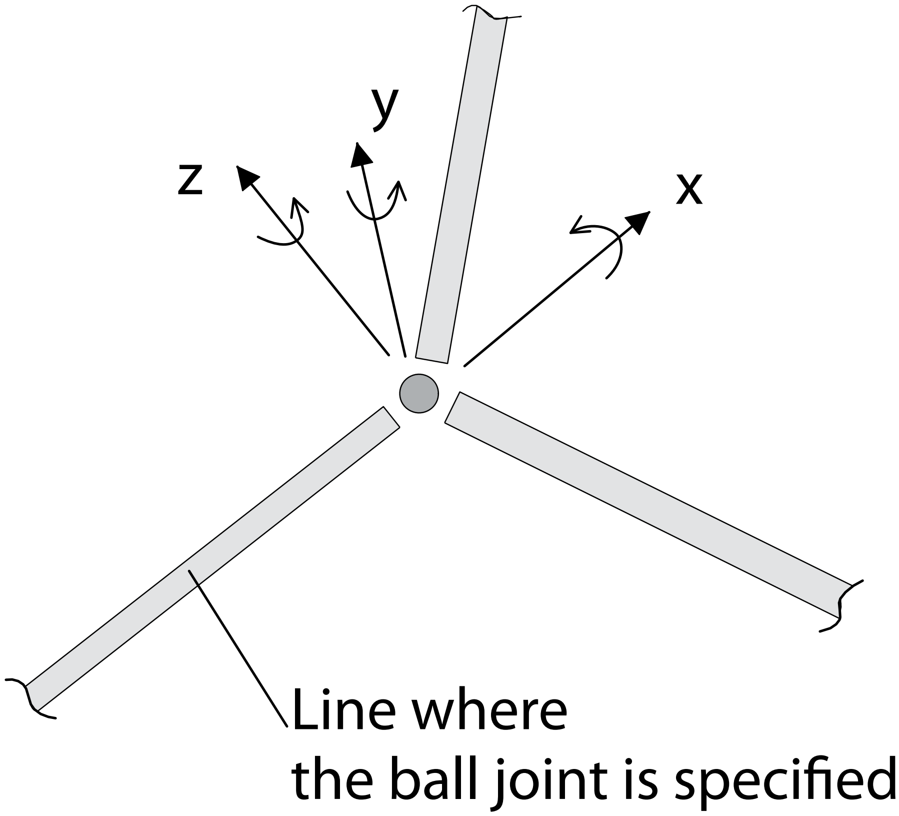

Figure 24. Rotation freedom for a ball joint component

5.4.7. FLEX - Description of flex-joint connectors

This component can be used to model ball joints, hinges and universal joints with specified rotational stiffness. It will introduce one extra element with zero length at the segment end to which it is attached, and add 6 degrees of freedom to the system model. The translation dofs of freedom are suppressed by use of linear constraint equations. Note that this component can not be used in branch lines in standard systems, or in combination with bar elements. It should also be used with care at supernodes with user defined boundary conditions for rotations in `AR` system to avoid singularities in the `FEM` solution procedure.

In present version, flex-joint connectors may only be used for nonlinear static and dynamic analysis.

5.4.7.1. Data group identifier

```
NEW COMPonent FLEX
```

5.4.7.2. Component type identifier

```
CMPTYP-ID
```

- `CMPTYP-ID: character(8)`: Component type identifier

5.4.7.3. Mass and volume

```
AM AE RGX RGY RGZ CRX CRY CRZ
```

- `AM: real, default: 0`: Mass $[\mathrm{M}]$
- `AE: real, default: 0`: Displacement volume $[\mathrm{L}^3]$
- `RGX: real, default: 0`: Radius of gyration around local x-axis $[\mathrm{L}]$
- `RGY: real, default: 0`: Radius of gyration around local y-axis $[\mathrm{L}]$
- `RGZ: real, default: 0`: Radius of gyration around local z-axis $[\mathrm{L}]$
- `CRX: real, default: 0`: Damping coeff. Rotational velocity around local x-axis $[\mathrm{FLT}/\mathrm{deg}]$
- `CRY: real, default: 0`: Damping coeff. Rotational velocity around local y-axis $[\mathrm{FLT}/\mathrm{deg}]$
- `CRZ: real, default: 0`: Damping coeff. Rotational velocity around local z-axis $[\mathrm{FLT}/\mathrm{deg}]$

5.4.7.4. Hydrodynamic coefficients

```
CDX CDY CDZ AMX AMY AMZ AMXROT AMYROT AMZROT
```

- `CDX: real, default: 0`: Drag coeff. in local x-direction $[\mathrm{F}/(\mathrm{L}/\mathrm{T})^2)]$
- `CDY: real, default: 0`: Drag coeff. in local y-direction $[\mathrm{F}/(\mathrm{L}/\mathrm{T})^2)]$
- `CDZ: real, default: 0`: Drag coeff. in local z-direction $[\mathrm{F}/(\mathrm{L}/\mathrm{T})^2)]$
- `AMX: real, default: 0`: Added mass in local x-direction $[\mathrm{M}]$
- `AMY: real, default: 0`: Added mass in local y-direction $[\mathrm{M}]$
- `AMZ: real, default: 0`: Added mass in local z-direction $[\mathrm{M}]$
- `AMXROT: real, default: 0`: Added mass rotation around local x-direction $[\mathrm{FL} \times \mathrm{T}^2]$
- `AMYROT: real, default: 0`: Added mass rotation around local y-direction $[\mathrm{FL} \times \mathrm{T}^2]$
- `AMZROT: real, default: 0`: Added mass rotation around local z-direction $[\mathrm{FL} \times \mathrm{T}^2]$

The tangential drag force, the force acting in local x-axis, is computed by:

$\mathrm{FX} = \mathrm{CDX} \times \mathrm{VRELX} \times |\mathrm{VRELX}|$

The drag force acting normal to the local x-direction, is assumed to act in the same direction as the relative velocity transverse component and are computed according to: -
$\mathrm{FY} = \mathrm{CDY} \times \sqrt{\mathrm{VRELY}^2 + \mathrm{VRELZ}^2} \times \mathrm{VRELY}$ -
$\mathrm{FZ} = \mathrm{CDY} \times \sqrt{\mathrm{VRELY}^2 + \mathrm{VRELZ}^2} \times \mathrm{VRELZ}$

5.4.7.5. Stiffness properties classification

`IDOF IBOUND RAYDMP`

- `IDOF: character(4)`: Degree of freedom
  - `IDOF = IRX`: Rotation around local x-axis
  - `IDOF = IRY`: Rotation around local y-axis
  - `IDOF = IRZ`: Rotation around local z-axis
  - `IDOF = IRYZ`: Rotation around bending axis
- `IBOUND: integer`: Constraint
  - `IBOUND = -1`: Fixed (Legal if 2 of 3 dofs are fixed)
  - `IBOUND = 0`: Free. Not available with `IDOF = IRYZ`
  - `IBOUND = 1`: Constant stiffness
  - `IBOUND > 1`: Table with `IBOUND` pairs of moment - rotational angle to be specified
- `RAYDMP: real`: Stiffness proportional damping coefficient

3 or 2 input lines to be specified: `IRX, IRY, IRZ` or `IRX, IRYZ`

x, y and z-axes refer to the local coordinate system of the element to which the flex joint is attached. This is similar to the ball joint connector as illustrated in the figure Rotation freedom for a ball joint component.

5.4.7.6. Stiffness data

Stiffness data are to be given in the sequence `IRX, IRY` and `IRZ` or `IRX` and `IRYZ`. Stiffness data are to be omitted for `IBOUND <= 0`

5.4.7.6.1. Linear stiffness

`IBOUND = 1`, One input line ~ STIFF ~

- `STIFF: real`: stiffness with respect to rotation $[\mathrm{FL/deg}]$

5.4.7.6.2. Nonlinear stiffness; IBOUND > 1

`IBOUND > 1`, IBOUND input lines ~ MOMENT ANGLE ~

- `MOMENT: real`: Moment corresponding to rotational angle $[\mathrm{FL}]$
- `ANGLE: real`: Rotational angle $[\mathrm{deg}]$

`MOMENT` and `ANGLE` must be given in increasing order. Linear extrapolation will be used outside the specified range of values.

For dofs `IRX, IRY` and `IRZ`, both negative and positive values should be given.

For dof `IRYZ`, `MOMENT` and `ANGLE` have to be greater or equal to zero. To avoid convergence problems, the first pair should be 0.0, 0.0.

5.4.8. FLUID - Specification of internal fluid flow

5.4.8.1. Data group identifier

`NEW COMPonent FLUId`

5.4.8.2. Component type number

`CMPTYP-ID`

- `CMPTYP-ID: character(8)`: Component type identifier

5.4.8.3. Fluid flow characteristics

`RHOI VVELI PRESSI DPRESS IDIR`

- `RHOI: real`: Density $[\mathrm{M/L^3}]$
- `VVELI: real`: Volume velocity $[\mathrm{L^3/T}]$
- `PRESSI: real`: Pressure at fluid inlet end $[\mathrm{F/L^2}]$
- `DPRESS: real`: Pressure drop $[\mathrm{F/L^3}]$
- `IDIR: integer, default: 1`: Flow direction code
    - 1: Inlet at supernode end 1 of the line

2: Inlet at supernode end 2 of the line

The pressure drop is assumed to be uniform over the line length. For further clarification of pressure definition, confer Theory Manual.

In this version only `RHOI` is used to calculate weight and mass for static and dynamic analysis. The other parameters are used for calculating wall force (flange force) only depending on output option (`OUTMOD`)

5.4.9. EXT1 - External wrapping

This component can be used to model additional weight or buoyancy modules attached to a riser line. The specified additional weight and buoyancy are used to adjust the average properties of the segment.

5.4.9.1. Data group identifier

```
NEW COMPonent EXT1
```

5.4.9.2. Component type identifier

```
CMPTYP-ID
```

- `CMPTYP-ID: character(8)`: Component type identifier

5.4.9.3. Mass and volume

```
AMS AE RGYR FRAC
```

- `AMS: real`: Mass per unit length $[\mathrm{M/L}]$
- `AE: real`: Buoyancy volume/length $[\mathrm{L}^2]$
- `RGYR: real`: Radius of gyration around local x-axis $[\mathrm{L}]$
- `FRAC: real`: Fraction of the segment that is covered $[1]$
  - `0 ⇐ FRAC ⇐ 1`

The resulting properties of the segment with external wrapping are:

Mass / length: - $\mathrm{AMS_{res}} = \mathrm{AMS_{cs}} + \mathrm{AMS_{ext}} \times \mathrm{FRAC}$

Resulting radius of gyration: -
$\mathrm{RGYR_{res}} = (\mathrm{AMS_{cs}} \times \mathrm{RGYR_{cs}^2} + \mathrm{AMS_{ext}} \times \mathrm{RGYR_{ext}^2} \times \mathrm{FRAC})/(\mathrm{AMS_{cs}} + \mathrm{AMS_{ext}} \times \mathrm{FRAC})$

Resulting external area for buoyancy: - $\mathrm{AE_{res}} = \mathrm{AE_{cs}} + \mathrm{AE_{ext}} \times \mathrm{FRAC}$

Where: - cs denotes the original cross section properties; i.e. without wrapping. - ext denotes the properties of the wrapping given in this data group. - res denotes the resulting average segment properties

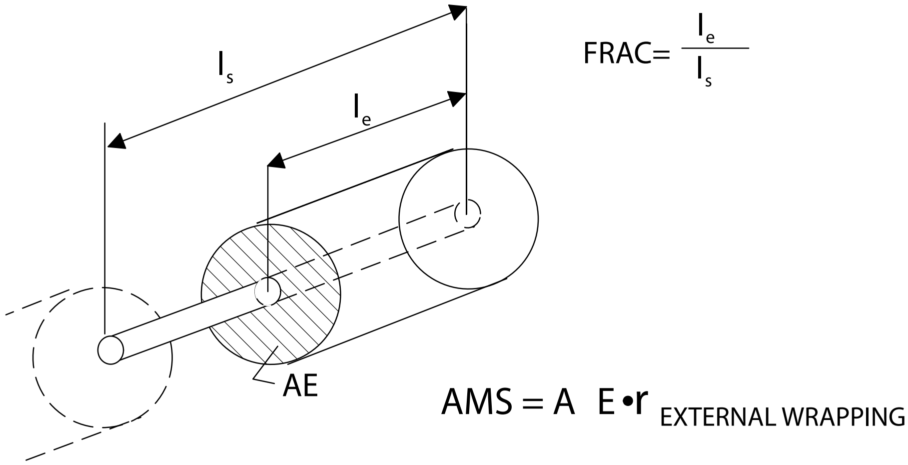

Figure 25. Description of external wrapping

5.4.9.4. Hydrodynamic coefficients

`CDX CDY AMX AMY CDLX CDLY`

- `CDX: real`: Drag force coefficient in tangential direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDY: real`: Drag force coefficient in normal direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `AMX: real`: Added mass per length in tangential direction $[\mathrm{M}/\mathrm{L}]$
- `AMY: real`: Added mass per length in normal direction $[\mathrm{M}/\mathrm{L}]$
- `CDLX: real, default: 0`: Linear drag force coefficients in tangential direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T}) \times \mathrm{L})]$
- `CDLY: real, default: 0`: Linear drag force coefficients in normal direction $[\mathrm{F}/((\mathrm{L}/\mathrm{T}) \times \mathrm{L})]$

The coefficients specified for the external wrapping are added directly to the coefficients specified for the pipe.

The drag forces per unit length acting in the local x-direction is computed as: -
$\mathrm{F_x} = (\mathrm{CDX_R} + \mathrm{FRAC} \times \mathrm{CDX})\mathrm{VRELX} \times \mathrm{VRELX} + (\mathrm{CDLX_R} + \mathrm{FRAC} \times \mathrm{CDLX})\mathrm{VRELX}$

In the case of an axis symmetric cross section the drag force per unit length, $\mathrm{F_n}$, acting normal to the local x-axis is computed by assuming that the instantaneous drag force direction is parallel to

the instantaneous transverse relative velocity component -
$\mathrm{F_n = (CDY_R + FRAC \times CDY)(VRELY^2 + VRELZ^2)}$
$\mathrm{+(CDLY_R + FRAC \times CDLY)\sqrt{VRELY^2 + VREZ^2}}$

In the case of a cross section with 2 symmetry planes the drag force per unit length in the local y and z directions are computed as: -
$\mathrm{F_y = (CDY_R + FRAC \times CDY)VRELY \times VRELY + (CDLY_R + FRAC \times CDLY)VRELY}$
-
$\mathrm{F_z = (CDZ_R + FRAC \times CDZ)VRELZ \times VRELZ + (CDLZ_R + FRAC \times CDLZ)VRELZ}$

Where: - $\mathrm{CDX_R, CDY_R, CDZ_R}$: are the input quadratic drag force coefficients of the riser in local x,y and z-directions - $\mathrm{CDLX_R, CDLY_R, CDLZ_R}$: are the input linear drag force coefficient of the riser in local x,y and z-directions - $\mathrm{CDX, CDY}$: are the input quadratic drag force coefficients of the external wrapping in local x and y-directions - $\mathrm{CDLX, CDLY}$: are the input linear drag force coefficients of the external wrapping in local x and y-directions - $\mathrm{VRELX, VRELY, VRELZ}$: are relative water velocities in local x, y and z-directions

For an axis symmetric pipe with external wrapping the input drag force coefficient in normal direction can be calculated as: - $\mathrm{CDY} = \frac{1}{2}\rho(\mathrm{DC_{dn} - D_R C_{dnR}})$

The added mass per unit length in normal direction can be calculated as: -
$\mathrm{AMY} = \rho\frac{\pi}{4}(\mathrm{D^2 C_{mn} - D_R^2 C_{mnR}})$

where: - ρ: water density - D: outer diameter of the external wrapping - $\mathrm{D_R}$: outer diameter of the pipe - $\mathrm{C_{dn}}$: normal drag coefficient - $\mathrm{C_{dnR}}$: normal drag coefficient of the pipe - $\mathrm{C_{mn}}$: normal added mass coefficient of the external wrapping - $\mathrm{C_{mnR}}$: normal added mass coefficient of the pipe

5.4.10. CRS3 - Partly submerged axisymmetric cross section

This cross section is used for floating structural members. It is different from cross section type CRS1 only for hydrodynamic load computation, in the sense that partly submerged cross section is taken care of. For hydrodynamic load computation the element is divided into subelements specified in data section Segment specification. NSEG input lines.. The cross section can only be used when consistent formulation is applied (see [STAMOD: External, static loads] (@ref stamod_b_definition_external1)) and for nonlinear dynamic analysis with regular waves.

Note that hydrodynamic force coefficients are specified for a fully submerged cross section. The actual values are taken proportional with the instantaneous cross section submergency for hydrodynamic load computation.

5.4.10.1. Exceptions compared to cross section type CRS1

Hydrodynamic load specification. Only Morison type loading is available. Use of the input card CHLOAD load type HYDR is not implemented.

Scaling of Froude-Krylov is not available. The SCFKN and SCFKT parameters are dummy and not used.

Aerodynamic load type specification is not implemented.

5.4.10.2. Data group identifier

```
NEW COMPonent CRS3
```

5.4.10.3. Component type identifier

Identical to Component type identifier for CRS1.

5.4.10.4. Mass and volume

Identical to Mass and volume for CRS1.

5.4.10.5. Stiffness properties classification

Identical to Stiffness properties classification for CRS1

5.4.10.6. Axial stiffness. Case 1, IEA=1

Identical to Axial stiffness. Case 1

5.4.10.7. Axial stiffness. Case 2, IEA=N

Identical to Axial stiffness. Case 2

5.4.10.8. Bending stiffness. Case 1a, IEJ=1 IPRESS=0 IMF=0

Identical to Bending stiffness. Case 1a

5.4.10.9. Bending stiffness. Case 1b, IEJ=1 IPRESS=0 IMF=1

Identical to Bending stiffness. Case 1b

5.4.10.10. Bending stiffness. Case 2, IEJ=1 IPRESS=1 (Not implemented)

Identical to Bending stiffness. Case 2

5.4.10.11. Bending stiffness description. Case 3 IEJ=N IPRESS=0

Identical to Bending stiffness description. Case 3 IEJ=N IPRESS=0 for CRS1

5.4.10.12. Bending stiffness. Case 4 IEJ=N1, IPRESS=N2 (Not implemented)

Identical to Bending stiffness. Case 4 IEJ=N1

5.4.10.13. Torsion stiffness

Identical to Torsion stiffness for CRS1

5.4.10.14. Damping specification

Identical to Damping specification for CRS1

5.4.10.15. Hydrodynamic force coefficients

Similar to Hydrodynamic force coefficients for CRS1, but only Morison type loading is available.

5.4.10.16. Capacity parameter

Identical to Capacity parameter for CRS1

5.4.11. CRS4 - Partly submerged double symmetric cross section

This cross section is used for floating structural members. It is different from cross section type CRS2 only for hydrodynamic load computation, in the sense that partly submerged cross section is taken care of. There is also a difference in the formulation of drag force. For hydrodynamic load computation the element is divided into subelements specified in data group Segment specification. NSEG input lines.. The cross section can only be used when consistent formulation is applied (see [STAMOD: External, static loads] (@ref stamod_b_definition_external1)) and for nonlinear dynamic analysis with regular waves.

Note that hydrodynamic force coefficients are specified for a fully submerged cross section. The actual values are taken proportional with the instantaneous cross section submergence. The actual submergence is based on an equivalent circular control area identical to the external area, AE.

The drag force per unit length acting normal to the local z-direction, is assumed to act in the same direction as the relative velocity transverse component and are computed according to: - $\mathrm{F_y = CDY \times (A_{sub}/AE) \times \sqrt{(VRELY^2 + VRELZ^2)} \times VRELY}$ - $\mathrm{F_z = CDZ \times (A_{sub}/AE) \times \sqrt{(VRELY^2 + VRELZ^2)} \times VRELZ}$

where $\mathrm{A_{sub}}$ is the instantaneous submerged area of the cross section.

5.4.11.1. Exceptions compared to section type CRS2

Hydrodynamic load specification. Only Morison type loading is available. Use of the input card CHLOAD load type HYDR is not implemented.

Scaling of Froude-Krylov is not available. The SCFKN and SCFKT parameters are dummy and not used.

Aerodynamic load type specification is not implemented.

5.4.11.2. Data group identifier

```
NEW COMPonent CRS4
```

5.4.11.3. Component type identifier

Identical to Component type identifier for CRS2.

5.4.11.4. Mass and volume

Identical to Mass and volume for CRS2.

5.4.11.5. Stiffness properties classification

Identical to Stiffness properties classification for CRS2

5.4.11.6. Axial stiffness. Case 1, IEA=1

Identical to Axial stiffness. Case 1

5.4.11.7. Axial stiffness. Case 2, IEA=N

Identical to Axial stiffness. Case 2

5.4.11.8. Bending stiffness. Case 1, IEJ=1 IPRESS=0

Identical to Bending stiffness. Case 1

5.4.11.9. Bending stiffness. Case 2, IEJ=1 IPRESS=1 (not implemented)

Identical to Bending stiffness. Case 2

5.4.11.10. Bending stiffness description. Case 3 IEJ=N IPRESS=0

Identical to [Bending stiffness description. Case 3 IEJ=N IPRESS=0 for CRS2]

5.4.11.11. Bending stiffness. Case 4 IEJ=N1, IPRESS=N2 (not implemented)

Identical to [Bending stiffness. Case 4 IEJ=N1]

5.4.11.12. Damping specification

Identical to [Damping specification for CRS1 and CRS2]

5.4.11.13. Hydrodynamic force coefficients

Similar to [Hydrodynamic force coefficients for CRS2], but only Morison type loading is available.

5.4.11.14. Capacity parameter

Identical to [Capacity parameter for CRS2]

5.4.12. CRS5 - Partly submerged general shaped cross section

This cross section is used for floating structural members. It can only be used for elements with local z-axis approximately parallel the global z-axis pointing in the same direction. The roll and pitch angles are restricted to 30° as a practical upper limit. The stiffness properties are based on two symmetry planes, while area and hydrodynamic force coefficients are calculated based on the cross section description represented by offset points symmetric with regard to the vertical (z) axis. For hydrodynamic load computation the element is divided into subelements specified in data section [Segment specification. NSEG input lines.]. The cross section can only be used when consistent formulation is applied (see [STAMOD: External, static loads] (@ref stamod_b_definition_external1)) and for nonlinear dynamic analysis with regular waves.

5.4.12.1. Data group identifier

```
NEW COMPonent CRS5
```

5.4.12.2. Component type identifier

Identical to [Component type identifier for CRS2].

5.4.12.3. Mass and volume

Identical to [Mass and volume for CRS2].

5.4.12.4. Stiffness properties classification

Identical to Stiffness properties classification for CRS2

5.4.12.5. Axial stiffness. Case 1, IEA=1

Identical to Axial stiffness. Case 1

5.4.12.6. Axial stiffness. Case 2, IEA=N

Identical to Axial stiffness. Case 2

5.4.12.7. Bending stiffness. Case 1, IEJ=1 IPRESS=0

Identical to Bending stiffness. Case 1

5.4.12.8. Bending stiffness. Case 2, IEJ=1 IPRESS=1 (not implemented)

Identical to Bending stiffness. Case 2

5.4.12.9. Bending stiffness description. Case 3 IEJ=N IPRESS=0

Identical to Bending stiffness description. Case 3 IEJ=N IPRESS=0 for CRS2

5.4.12.10. Bending stiffness. Case 4 IEJ=N1, IPRESS=N2 (not implemented)

Identical to Bending stiffness. Case 4 IEJ=N1

5.4.12.11. Damping specification

Identical to Damping specification for CRS2

5.4.12.12. Hydrodynamic force coefficients

```
CDX CDY CDZ CDTMOM AMX
```

- `CDX: real`: Drag force coefficient per length, tangential $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDY: real`: Drag force coefficient per length, local y-axis $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDZ: real`: Drag force coefficient per length, local z-axis $[\mathrm{F}/((\mathrm{L}/\mathrm{T})^2 \times \mathrm{L})]$
- `CDTMOM: real`: Drag coefficient around local x-axis
  - Dummy in present version.

`AMX: real`: Added mass per length, tangential $[\mathrm{M/L}]$

All hydrodynamic force coefficients applies to fully submerged cross section. That is, the actual coefficients are proportional to submerged volume. The drag force coefficients are to be scaled according to consistent units used, see Unit names and gravitational constant.

The tangential drag force which is a friction force acting along the local x-direction is calculated according to: - $\mathrm{F_t = CDX \times (A_{sub}/A_{tot}) \times V_{x,rel} \times |V_{x,rel}|}$

The viscous normal force per unit length is calculated using the drag force term in Morison's equation and assuming the drag force direction is parallel the instantaneous relative velocity transverse component: - $\mathrm{F_y = CDY \times (A_{sub}/A_{tot}) \times \sqrt{V_{y,rel}^2 \times V_{z,rel}^2} \times V_{y,rel}}$

- $\mathrm{F_z = CDZ \times (A_{sub}/A_{tot}) \times \sqrt{V_{y,rel}^2 \times V_{z,rel}^2} \times V_{z,rel}}$

where: - $\mathrm{A_{sub}}$: is instantaneous cross section submergence - $\mathrm{A_{tot}}$: is total external areal of the cross section - $\mathrm{V_{x,rel}}$: is relative water velocity in local x-direction - $\mathrm{V_{y,rel}}$: is relative water velocity in local y-direction - $\mathrm{V_{z,rel}}$: is relative water velocity in local z-direction

5.4.12.13. Description of cross section shape

`NOB NSUB NROLL NDFS`

- `NOB: integer`: Number of offset points to describe the cross section shape.
  - Only one half of the shape is described due to assumed symmetry about local z-axis.
  - 3 ⇐ NOB ⇐ 20
- `NSUB: integer, default: 20`: Number of points of submergence in table of submerged volume as function of submergence and roll angle.
- `NROLL: integer, default: 20`: Number of roll angles in table of submerged volume as function of submergence and roll angle.
- `NDFS: integer, default: 20`: Number of points of submergence in table of added mass and poten- tial damping as function of submergence.

The submerged cross section area is calculated for a number of submergence positions and relative roll angles in the range (`0` - π`/2`). The instantaneous submerged area is found by linear interpola- tion for points lying between those given in the table.

Tables of two-dimensional added mass and potential damping as function of submergence are calculated using the Frank.closefit technique. The instantaneous added mass and damping are found by linear interpolation for points lying between those given in the tables.

5.4.12.14. Offset points

`INB YB ZB`

- `INB: integer`: Offset point number
- `YB: real`: Local y-coordinate for offset point `INB`
- `ZB: real`: Local z-coordinate for offset point `INB`

Only one half of the cross section shape is modelled due to the assumed symmetry about local z-axis.

The offset points must be given in increasing order with decreasing value of the z-coordinate. `YB` and `ZB` are referred to the principal local axis. `YB >= 0` and first and last value of `YB` has to be zero, see the figure below.

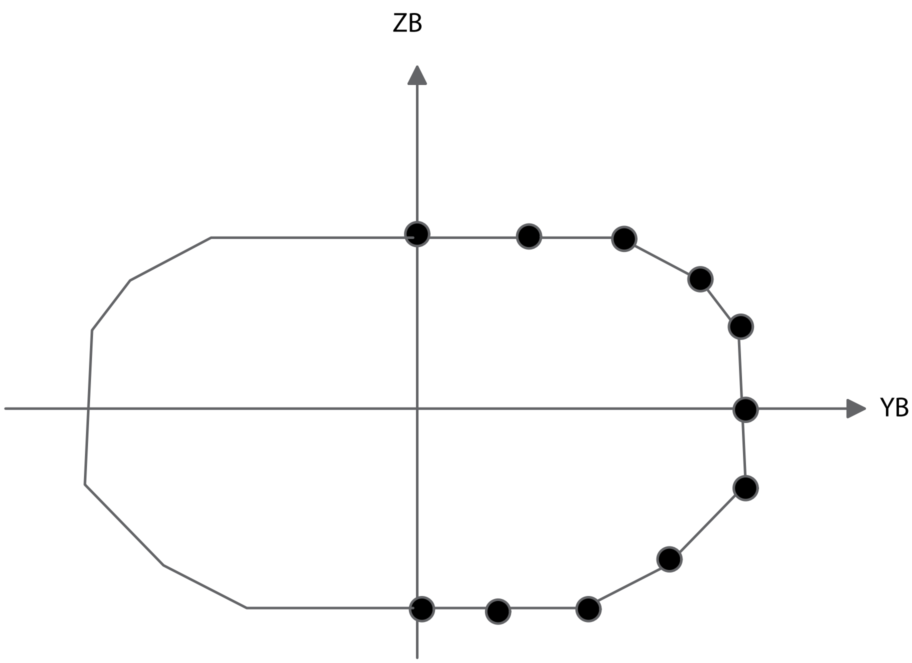

Figure 26. Example of modelling cross sectional shapes of frame elements

5.4.12.15. Capacity parameter

Identical to Capacity parameter for CRS2

5.4.13. CONTACT - Contact point of roller type

Available for elastic contact surface description only.

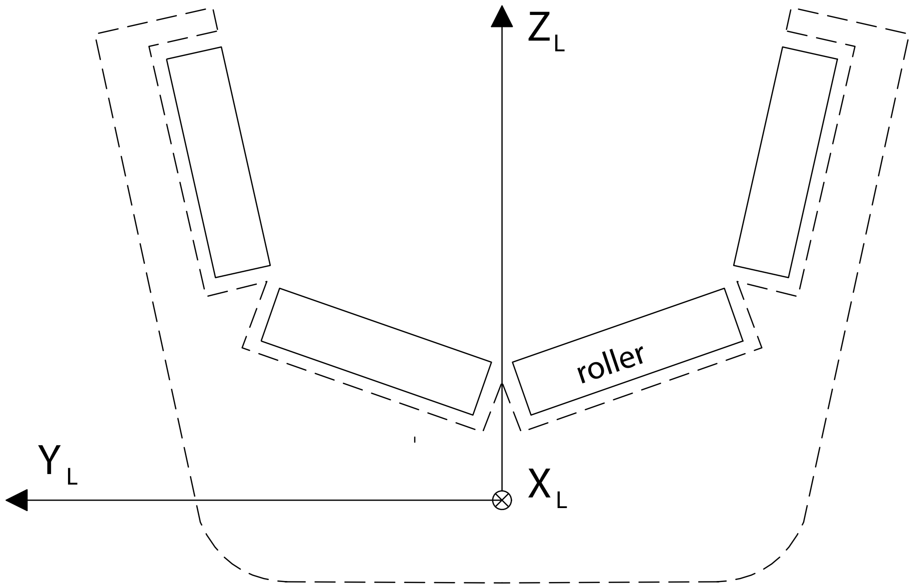

Figure 27. Example of a pipe support consisting of four rollers

The local coordinate system (X_L, Y_L, Z_L) of the elastic contact surface is indicated. The X_L-axis is pointing into the paper plane.

The contact point may contain several rollers.

The rollers are located in the $Y_L Z_L$-plane of the element to which the contact point is attached. Besides the location, each roller is described by its length, which may be infinite, by its stiffness and dash pot damping. The location and orientation of a roller is defined by a point and an inclination angle referred in the local coordinate system of the contact surface element. A roller of finite length is shown in the figure below. The roller origin (starting point) is defined by the point (Y_R, Z_R) and the inclination angle (`ROTX`) is defined by a clockwise rotation around the contact surface X_L-axis.

Roller with finite length located in the local coordinate system of an element contributing to the elastic contact surface

The X_L-axis is pointing into the paper plane.

5.4.13.1. Data group identifier

```
NEW COMPonent CONTact
```

5.4.13.2. Component type identifier

```
CMPTYP-ID
```

- `CMPTYP-ID: character(8)`: Component type identifier

5.4.13.3. Number of rollers

`NROLLS`

- `NROLLS: integer`: Number of rollers

The following 3 data groups (`Location and orientation', `Stiffness properties' and `Spring stiffness, Case 1 or 2') must be given in blocks for each of the `NROLLS` roller.

5.4.13.4. Location and orientation of roller axis

`ROTX YR ZR RLENG`

- `ROTX: real, default: 0`: Direction of roller axis. (Clockwise around the X_L-axis of the actual surface plane) $[\mathrm{deg}]$
- `YR: real, default: 0`: Y-coordinate of roller origin $[\mathrm{L}]$
- `ZR: real, default: 0`: Z-coordinate of roller origin $[\mathrm{L}]$
- `RLENG: real, default: 0`: Length of roller $[\mathrm{L}]$
  - `= 0` means infinite length

In case of infinite roller length, `YR` and `ZR` describe coordinates of an arbitrary point on the roller principal axis.

5.4.13.5. Stiffness properties classification and damping

`IKS DAMP`

- `IKS: integer`: Stiffness code1
  - 1 : Constant spring compression stiffness
  - N : Table with N pairs of pressure force - displacements to be specified
    - N > 2
- `DAMP: real, default: 0`: Dash pot damping coefficient $[\mathrm{FT/L}]$

5.4.13.6. Spring stiffness, Case 1 IKS = 1

`STIFFR RADROL`

- `STIFFR: real`: Spring compression stiffness $[\mathrm{F/L}]$
- `RADROL: real`: Radius of roller $[\mathrm{L}]$

The figure below describes the interpretation of contact force in case that `IKS=1`. The spring is active when the distance between the principal axis of the roller and the pipe is less than $\Delta_0 = \mathrm{RADROl} + \mathrm{RTUBE}$. The external radius of the tube, `RTUBE`, is calculated from the external area of the cross section of the element in contact with the roller.

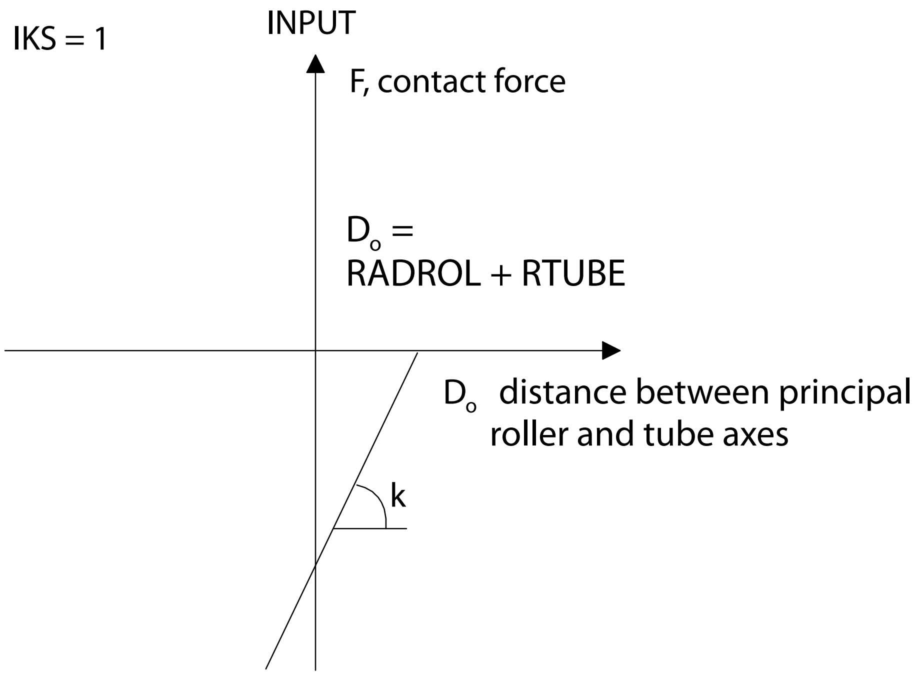

Figure 28. Spring stiffness IKS = 1

5.4.13.7. Spring stiffness, Case 2 IKS > 2

`FS(1) ZS(1) ... FS(N) ZS(N)`

- `FS(1): real < 0`: Pressure force corresponding to compression `ZS(1)` $[\mathrm{F}]$
- `ZS(1): real`: Spring compression $[\mathrm{L}]$
- .
- .
- .

`ZS(i)` must be given in increasing order.

The figure below describes the interpretation of contact force in case that `IKS>2`. The specified stiffness characteristics is moved to account for the external radius of the tube, `RTUBE`. The external radius of the tube, `RTUBE`, is calculated from the external area of the cross section of the element in contact with the roller.

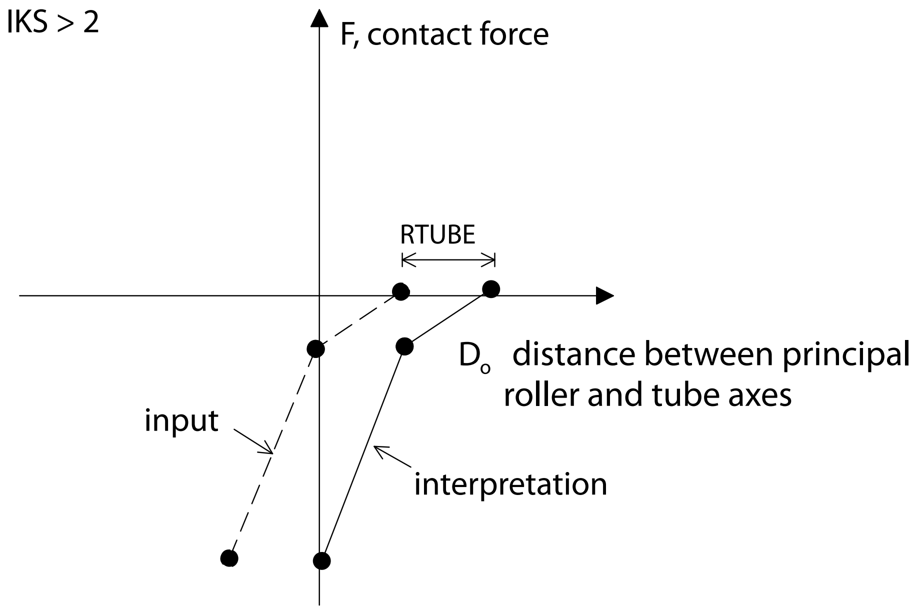

Figure 29. Spring stiffness IKS > 2

The three data groups `Location and orientation', `Stiffness properties' and `Spring stiffness, Case 1 or 2' are to be repeated NROLLS times.

5.4.14. Tensioner

Available for elastic contact surface description only.

The function of the tensioner is to grip and apply tension to the pipeline during the lay operation. In dynamic analysis the tensioner accounts for the pipeline pay out or pay in to prevent large oscillations in the pipeline tension. This is modelled as a dynamic boundary condition with respect to the applied axial force, eg. the applied load is `T0` plus/minus a dead band range. Outside the dead band range the load is constant. The applied load which acts in the longitudinal direction of the tube, is formulated as a discrete element load. During static analysis the tensioner applies a constant load, `T0`, to the pipeline.

5.4.14.1. Data group identifier, one input line

```
NEW COMPonent TENSioner
```

5.4.14.2. Component type identifier

```
CMPTYP-ID
```

- `CMPTYP-ID: character(8)`: Component type identifier

5.4.14.3. Characteristics of tensioner

```
T0 TMAX TMIN DELTA SIGNX
```

- `T0: real`: Applied load during static analysis $[\mathrm{F}]$
- `TMAX: real`: Maximum load transmitted from the tensioner $[\mathrm{F}]$
- `TMIN: real`: Minimum load transmitted from the tensioner $[\mathrm{F}]$
- `DELTA: real`: Pipeline displacement through the tensioner for a load variation of: `TMAX-TMIN` $[\mathrm{L}]$
- `SIGNX: real, default: 1`: Direction of applied load referring to local x-axis of the element going through the tensioner $[]$
    - `SIGNX = 1.0`: The load will act in local x-axis direction
    - `SIGNX = -1.0`: The load will act opposite local x-axis

The stiffness characteristics of the tensioner will be derived from `DELTA` as: `STIFF = (TMAX-TMIN)/DELTA`

5.4.15. Tubular contact component

This component is available for elastic contact surface description only.

5.4.15.1. Data group identifier, one input line

```
NEW COMponent TUBUlar contact
```

5.4.15.2. Component type number

```
CMPTYP-ID
```

- `CMPTYP-ID: character(8)`: Component type identifier

5.4.15.3. Specification of contact force characteristics

```
RCONT CHDIR IKS DAMP STIFFR FRICST FRICDY CHAXI CHROT VELLIM
```

- `CONT: real`: Contact radius $[\mathrm{L}]$
- `CHDIR: character`: Contact direction: `INWARDS` or `OUTWARDS`
- `IKS: integer`: Stiffness code for radial contact force
    - `= 1` Constant contact compression stiffness
    - `= N` Table with N pairs of contact force - displacement to be specified

`RELDAM: real`: = Desired relative damping level at estimated eigen period in the pipe, pipe and contact spring system (see below) $[1]$. Damping is only applied in the radial direction. Not used in static analysis.

- `DAMP: real`: Dash pot damping coefficient $[\mathrm{FT/L}]$. Damping is only applied in the radial direction. Not used in static analysis.
- `STIFFR: real`: Spring stiffness associated with static friction coefficient `FRICST`, $[\mathrm{F/L}]$. The spring stiffness is applied in the ring and axial directions until the spring force exceeds the static friction force. Not used in static analysis. Dummy if `CHAXI = OFF`.
- `FRICST: real`: Static friction coefficient $[1]$. Not used in static analysis. Dummy if `CHAXI = OFF`.
- `FRICDY: real`: Dynamic sliding friction coefficient $[1]$. `FRICDY ⇐ FRICST`. Not used in static analysis. Dummy if `CHAXI = OFF`.
- `CHAXI: character`: Control parameter for axial sliding friction
  - `= ON`
  - `= OFF`
- `CHROT: character`: Control parameter for friction caused by rotation
  - `= ON` Requires `CHAXI=ON`
  - `= OFF`
- `VELLIM: real`: Velocity limit for determining that sliding has stopped $[\mathrm{L/T}]$. If the relative sliding velocity between the pipes falls below `VELLIM`, the spring stiffness `STFFR` is applied. Should be small, but not zero. Not used in static analysis. Dummy if `CHAXI = OFF`.

Based on specified damping level the stiffness proportional damping coefficient is calculated by

$$\mathrm{a}_2 = \mathrm{RELDAM} \times 2 \times \sqrt{\frac{(\mathrm{AMS}\times\mathrm{L})_\mathrm{M}+(\mathrm{AMS}\times\mathrm{L})_\mathrm{S}}{\mathrm{STIFF}}}$$

where $(\mathrm{AMS} \times \mathrm{L})_\mathrm{M}$ and $(\mathrm{AMS} \times \mathrm{L})_\mathrm{S}$ are total structural element mass of the master pipe and the slave pipe respectively and STIFF is contact spring sitffness.

5.4.15.4. Contact spring stiffness; IKS = 1

`STIFF`

- `STIFF: real`: Spring compression stiffness $[\mathrm{F/L}]$

5.4.15.5. Contact spring stiffness; IKS > 1

`FS(1) ZS(1) ........ FS(N) ZS(N)`

- `FS(1): real < 0`: Pressure force corresponding to compression `ZS(1)` $[\mathrm{F}]$
- `ZS(1)`: Spring compression $[\mathrm{L}]$

`ZS(i)` must be given in increasing order

5.4.16. Seafloor contact

The seafloor contact properties are relevant for riser systems with tubular cross sections, which are partly resting on the bottom. This may be the case for `SB` and `AR` systems.

5.4.16.1. Data group identifier, one input line

`NEW COMPonent SEAFloor contact`

5.4.16.2. Component type identifier and type

`CMPTYP-ID CHSFCT`

- `CMPTYP-ID: character(8)`: Component identifier
- `CHSFCT: character(4)`: Seafloor contact component type
    - `= SPRI`: Original RIFLEX seafloor springs normal to the seafloor and separate axial and lateral spring-friction contact in the seafloor plane.
    - `= SOIL`: Consolidated riser-soil interaction model
    - `= CARI`: Carisima seafloor contact (restricted option)

5.4.16.3. Original RIFLEX seafloor spring contact

The following three lines of input must be given if `CHSFCT = SPRI`

5.4.16.3.1. Seafloor normal contact parameters

`STFBOT DAMBOT`

- `STFBOT: real > 0`: Seafloor stiffness normal to the seafloor $[\mathrm{F}/\mathrm{L}^2]$
- `DAMBOT: real >= 0, default: 0`: seafloor damping coefficient normal to the seafloor $[\mathrm{F} \times \mathrm{T}/\mathrm{L}^2]$

`STFBOT` is used for computing the spring stiffness normal to the seafloor, $\mathrm{k_V}$, for seafloor contact. $\mathrm{k_V}$ = `STFBOT` $\times \mathrm{L}$ where L is the element length.

5.4.16.3.2. Seafloor in-plane contact parameters, two input lines

```
STFAXI FRIAXI DAMAXI
```

- `STFAXI: real >= 0, default: 0`: In-plane seafloor stiffness for friction in axial direction $[\mathrm{F/L^2}]$
- `FRIAXI: real >= 0, default: 0`: In-plane seafloor friction coefficient in axial direction [1]
- `DAMAXI: real >= 0, default: 0`: In-plane seafloor damping coefficient in axial direction $[\mathrm{F} \times \mathrm{T/L^2}]$

```
STFLAT FRILAT DAMLAT ILTOR
```

- `STFLAT: real >= 0, default: 0`: In-plane seafloor stiffness for friction in lateral direction $[\mathrm{F/L^2}]$
- `FRILAT: real >= 0, default: 0`: In-plane seafloor friction coefficient in lateral direction [1]
- `DAMLAT: real >= 0, default: 0`: In-plane seafloor damping coefficient in lateral direction $[\mathrm{F} \times \mathrm{T/L^2}]$
- `ILTOR: integer, default: 0`: Option for applying lateral contact forces at the external contact radius, giving a torsional moment
  - `= 0`: Lateral loads are applied at the node
  - `= 1`: Lateral loads are applied at the external contact radius if it is specified for the associated beam cross-section.

Contact in the seafloor plane is modelled independently in the axial and lateral directions. Contact is initially modelled with linear springs. Sliding will occur when an axial or lateral spring force reaches the friction force value. Springs will be reinstated if the line starts sliding in the opposite direction, or if the friction force increases and is greater than the spring force. The spring stiffness is calculated as $k_h = \mathrm{Stalks} \times L_h$, where $\mathrm{L_h}$ is the length of the element's horizontal projection. The seafloor friction forces are calculated as $F = \mathrm{FRIxxx} \times F_{vert}$ and are directed against the axial or lateral displacements, where $\mathrm{F_{vert}}$ is the vertical contact force between the pipe and the bottom.

5.4.16.4. Consolidated riser-soil seafloor contact

The following four lines of input must be given if `CHSFCT = SOIL`

The external contact radius `R_EXTCNT` must be positive for the segments that have consolidated riser-soil seafloor contact.

5.4.16.4.1. Seafloor soil properties

```
W A1 A2 V G
```

- `W: real > 0`: Soil submerged weight $[\mathrm{F}/\mathrm{L}^3]$
- `A1: real > 0`: Soil shear strength at seabed $[\mathrm{F}/\mathrm{L}^2]$
- `A2: real`: Soil shear strength vertical gradient $[\mathrm{F}/\mathrm{L}^3]$
- `V: real > 0`: Soil Poisson ratio $[1]$
- `G: real`: Soil G-modulus/compressive strength $[\mathrm{F}/\mathrm{L}^2]$

5.4.16.4.2. Consolidated riser-soil seafloor contact options

```
F ALPHA BETA KBC KT
```

- `F: real, default: 0.88`: Relationship between dynamic stiffness and G-modulus $[1]$
- `ALPHA: real, default: 1.0`: Control parameter for suction release displacement $[1]$
- `BETA: real, default: 1.0`: Scaling factor for peak soil suction relative to peak soil compression $[1]$
- `KBC: real, default: 0.05`: Mobilization displacement for soil bearing capacity as fraction of pipe soil contact width $[1]$
- `KT: real, default: 0.08`: Mobilization displacement for max soil suction as fraction of pipe soil contact width $[1]$

5.4.16.4.3. In-plane contact parameters, two input lines

```
STFAXI FRIAXI DAMAXI
```

- `STFAXI: real >= 0, default: 0`: In-plane seafloor stiffness for friction in axial direction $[\mathrm{F}/\mathrm{L}^2]$
- `FRIAXI: real >= 0, default: 0`: In-plane seafloor friction coefficient in axial direction $[1]$
- `DAMAXI: real >= 0, default: 0`: In-plane seafloor damping coefficient in axial direction $[\mathrm{F} \times \mathrm{T}/\mathrm{L}^2]$

`STFLAT FRILAT DAMLAT`

- `STFLAT: real >= 0, default: 0`: In-plane seafloor stiffness for friction in lateral direction $[\mathrm{F/L^2}]$
- `FRILAT: real >= 0, default: 0`: In-plane seafloor friction coefficient in lateral direction [1]
- `DAMLAT: real >= 0, default: 0`: In-plane seafloor damping coefficient in lateral direction $[\mathrm{F \times T/L^2}]$

Contact in the seafloor plane is modelled independently in the axial and lateral directions. Contact is initially modelled with linear springs. Sliding will occur when an axial or lateral spring force reaches the friction force value. Springs will be reinstated if the line starts sliding in the opposite direction, or if the friction force increases and is greater than the spring force. The spring stiffness is calculated as $k_h = \mathrm{Stalks} \times L_h$, where $\mathrm{L_h}$ is the length of the element's horizontal projection. The seafloor friction forces are calculated as $F = \mathrm{FRIxxx} \times F_{vert}$ and are directed against the axial or lateral displacements, where $\mathrm{F_{vert}}$ is the vertical contact force between the pipe and the bottom.

5.4.16.5. Carisima seafloor contact, restricted option

The following input must be given if `CHSFCT = CARI` (restricted optio).

The soil can be modeled either as an elastic foundation with coulomb friction or as a sandy or clayey soil. The sand/clay model account for initial penetration by weight and digging due to small displacements of the pipe. The penetration is used to determine vertical and lateral stiffness and breakout forces.

5.4.16.5.1. Carisima factors and debug options

`UFA1 UFA2 IPRSO IPREL IPRPO ILIN ISEG INOD IPCE`

- `UFA1: real, default: 1`: Factor for converting default values from $[\mathrm{kN/m^2}]$ to actual unit for $[\mathrm{F/L^2}]$
- `UFA2: real, default: 1`: Factor for converting default values from $[\mathrm{kN/m^3}]$ to actual unit for $[\mathrm{F/L^3}]$
- Temporary input for debug print of detailed `CARISIMA` results (use with caution):
    - Either:
        - `IPRSO: integer, default: 0`: Debug print switch
        - `IPREL: integer, default: 0`: Element for original print (`IPRSO`)

`IPRPO: integer, default: 0`: Contact point for original print (`IPRSO`)

- Or specify up to 10 soil contact points:
    - `ILIN: integer, default: 0`: Line for print of detailed `CARISIMA` results
    - `ISEG: integer, default: 0`: Local segment for print of detailed `CARISIMA` results
    - `INOD: integer, default: 0`: Local element for print of detailed `CARISIMA` results
    - `IPCE: integer, default: 0`: Local contact point for print of detailed `CARISIMA` results

5.4.16.5.2. Elastic foundation with coulomb friction

```
ISTYP RKU RKV RKW RMUU RMUV
```

- `ISTYP: character(4):= COUL`
- `RKU: real, default: 145`: Longitudinal stiffness per unit pipe length $[\mathrm{F}/\mathrm{L}^2]$
- `RKV: real, default: 145`: Transverse stiffness per unit pipe length $[\mathrm{F}/\mathrm{L}^2]$
- `RKW: real, default: 200`: Vertical stiffness per unit pipe length $[\mathrm{F}/\mathrm{L}^2]$
- `RMUU: real, default: 0.7`: Axial friction coefficient
- `RMUV: real, default: 0.7`: Transverse friction coefficient

Indicated default values for parameters with dimension $[\mathrm{F}/\mathrm{L}^2]$ or $[\mathrm{F}/\mathrm{L}^3]$ are in $[\mathrm{kN}/\mathrm{m}^2]$ or $[\mathrm{kN}/\mathrm{m}^3]$ and will be scaled by `UFA1` or `UFA2` before being applied. See echo of input if in doubt.

5.4.16.5.3. Sandy soil

Use one of the lists.

```
ISTYP GRADING
```

- `ISTYP: character(4):= SAND`
- `Grading: character`: One of: `LOOSe, MEDIum, DENSe`

Indicated default values for parameters with dimension $[\mathrm{F}/\mathrm{L}^2]$ or $[\mathrm{F}/\mathrm{L}^3]$ are in $[\mathrm{kN}/\mathrm{m}^2]$ or $[\mathrm{kN}/\mathrm{m}^3]$ and will be scaled by `UFA1` or `UFA2` before being applied. See echo of input if in doubt.

`ISTYP PHI WSOI POI ES RMUU RMUV`

- `ISTYP: character(4): = SAND`
- `PHI: real, default: 30`: Angle of friction $[\mathrm{deg}]$
- `WSOI: real, default: 9.1`: Submerged unit weight of soil $[\mathrm{F}/\mathrm{L}^3]$
- `POI: real, default: 0.35`: Poisson's ratio
- `ES: real, default: 0.7`: Void ratio
- `RMUU: real, default: 0.6`: Axial friction coefficient
- `RMUV: real, default: 0.6`: Transverse friction coefficient (not used)

Indicated default values for parameters with dimension $[\mathrm{F}/\mathrm{L}^2]$ or $[\mathrm{F}/\mathrm{L}^3]$ are in $[\mathrm{kN}/\mathrm{m}^2]$ or $[\mathrm{kN}/\mathrm{m}^3]$ and will be scaled by `UFA1` or `UFA2` before being applied. See echo of input if in doubt.

5.4.16.5.4. Clay soil

Use one of the lists.

`ISTYP GRADING`

- `ISTYP: character(4): = CLAY`
- `Grading: character`: One of: `LOOSe, SOFT, STIFf, HARD`

Indicated default values for parameters with dimension $[\mathrm{F}/\mathrm{L}^2]$ or $[\mathrm{F}/\mathrm{L}^3]$ are in $[\mathrm{kN}/\mathrm{m}^2]$ or $[\mathrm{kN}/\mathrm{m}^3]$ and will be scaled by `UFA1` or `UFA2` before being applied. See echo of input if in doubt.

`ISTYP SU WSOI SUG POI ES PIX OCR RMUU RMUV RFVMX`

- `ISTYP: integer/character: = CLAY`
- `SU: real, default: 5`: Undrained shear strength $[\mathrm{F}/\mathrm{L}^3]$
- `WSOI: real, default: 4.4`: Submerged unit weight of soil $[\mathrm{F}/\mathrm{L}^3]$
- `SUG: real, default: 0`: Shear strength gradient $[\mathrm{F}/\mathrm{L}^3]$
- `POI: real, default: 0.45`: Poisson's ratio $[1]$
- `ES: real, default: 2`: Void ratio $[1]$
- `PIX: real, default: 60`: Plasticity index $[\%]$

`OCR: real, default: 1`: Over-consolidation ratio

- `RMUU: real, default: 0.2`: Axial friction coefficient
- `RMUV: real, default: 0.2`: Transverse friction coefficient (not used)
- `RFVMX: real, default: 2.5`: Max transverse capacity ratio $[1]$

Indicated default values for parameters with dimension $[\mathrm{F}/\mathrm{L}^2]$ or $[\mathrm{F}/\mathrm{L}^3]$ are in $[\mathrm{kN}/\mathrm{m}^2]$ or $[\mathrm{kN}/\mathrm{m}^3]$ and will be scaled by `UFA1` or `UFA2` before being applied. See echo of input if in doubt.

5.4.16.5.5. Clay soil (final model)

Use one of the lists.

`ISTYP GRADING`

- `ISTYP: character:= CARI`
- `Grading: character(4)`: One of: `LOOSE, SOFT, STIF, HARD`

The input lines must be given in sequence of increasing frequency values.

`ISTYP SU WSOI SU0 POI ES PIX OCR RMUU RMUV RFVMX`

- `ISTYP: character(4):= CARI`
- `SU: real, default: 5`: Undrained shear strength $[\mathrm{F}/\mathrm{L}^3]$
- `WSOI: real, default: 4.4`: Submerged unit weight of soil $[\mathrm{F}/\mathrm{L}^3]$
- `SU0: real, default: 0`: Shear strength gradient $[\mathrm{F}/\mathrm{L}^3]$
- `POI: real, default: 0.45`: Poisson's ratio $[1]$
- `ES: real, default: 2`: Void ratio $[1]$
- `PIX: real, default: 60`: Plasticity index $[\%]$
- `OCR: real, default: 1`: Over-consolidation ratio $[1]$
- `RMUU: real, default: 0.2`: Axial friction coefficient
- `RMUV: real, default: 0.2`: Transverse friction coefficient (not used)
- `RFVMX: real, default: 2.5`: Max transverse capacity ratio $[1]$

Indicated default values for parameters with dimension $[\mathrm{F/L^2}]$ or $[\mathrm{F/L^3}]$ are in $[\mathrm{kN/m^2}]$ or $[\mathrm{kN/m^3}]$ and will be scaled by `UFA1` or `UFA2` before being applied. See echo of input if in doubt.

5.4.16.5.6. Options for calculation of contact forces

`FLAG ISYMR IZVEC IRINI RZ0 RZ1 IMOMC NCPE ILUCO`

- `FLAG: character: = SOPT`
- `ISYMR: integer, default: 1`: Option for keeping symmetry in force model.
  - Remaining function symmetry option
  - `= 0`: Trench shoulders move independently
  - `= 1`: Trench shoulders move symmetrically
  - Used for soil types `SAND` and `CLAY` only.
- `IZVEC: integer, default: 0`: Option for considering velocities in calculations
  - `=0`: Ignore velocities
  - `=1`: Use velocities to improve accuracy at plastic to elastic drop
  - Used for soil types `COULOMB`, `SAND` and `CLAY` only.
- `IRINI: integer, default: 0`: Option for re-initialization of trench position
  - `= 0`: No re-initialization allowed
  - `= 1`: Re-initialization of trench at new position allowed; i.e. the initial trench will ``follow" the line to a new contact position.
- `RZ0: integer, default: 0.001`: Penetration ratio for secant stiffness
- `RZ1: integer, default: 0.01`: Initial penetration ratio
- `IMOMC: integer, default: 1`: Options for including rotation / moment terms
  - `= 0`: Related rotation/moment terms not included, i. e. only translational dofs active in seafloor contact.
  - `= 1`: Related rotation/moment terms included in load, stiffness and damping
- `NCPE: integer >= 2, default: 3`: Number of contact points to be used along each element

- `ILUCO: integer, default: 0`: Option to calculate contact at element ends
    - `= 0`: Contact forces calculated at NCPE points along each element
    - `= 1`: Contact forces calculated at element ends
    - Dummy for bar elements, for which forces are calculated at the element ends.

5.4.16.5.7. Stiffness data

`FLAG ISSMO RKU RKVI RKWI RKWN RKWX`

- `FLAG: character`: = `STIM`
- `ISSMO: integer, default: 1`:
    - `= 0`: Linear springs in all directions
    - `< 0`: Linear springs in axial and transverse directions
    - `= ±1`: Stiffness based on DNV/Guideline model
    - `= ±2`: Stiffness based on STATOIL/SINTEF model
    - Used for soil types `SAND` and `CLAY`
- `RKU: real, default: 145`: Fixed stiffness in axial direction (`ISSMO >= 0`) $[\mathrm{F/L^2}]$
    - Used for soil types `COUL`, `SAND` and `CLAY` and `ISSMO ⇐ 0`
- `RKVI: real, default: 145`: Fixed stiffness in transverse direction (`ISSMO >= 0`) $[\mathrm{F/L^2}]$
    - Used for soil types `COUL`, `SAND` and `CLAY` and `ISSMO ⇐ 0`
- `RKWI: real, default: 200`: Fixed stiffness in vertical direction (`ISSMO >= 0`) $[\mathrm{F/L^2}]$
    - Used for soil types `COUL`, `SAND` and `CLAY` and `ISSMO = 0`
- `RKWN: real, default: 10`: Minimum vertical stiffness $[\mathrm{F/L^2}]$
    - Used for soil types `COUL`, `SAND` and `CLAY` and `ISSMO` $\neq$ `0`
- `RKWX: real, default: 15000`: Maximum vertical stiffness $[\mathrm{F/L^2}]$
    - Used for soil types `COUL`, `SAND` and `CLAY` and `ISSMO` $\neq$ `0`
    - Also used for soil type `CARI`?

Indicated default values for parameters with dimension $[\mathrm{F}/\mathrm{L}^2]$ or $[\mathrm{F}/\mathrm{L}^3]$ are in $[\mathrm{kN}/\mathrm{m}^2]$ or $[\mathrm{kN}/\mathrm{m}^3]$ and will be scaled by `UFA1` or `UFA2` before being applied. See echo of input if in doubt.

Specified values of `RKU`, `RKVI` and `RKWI` overwrite values specified (or set to default) for soil type `COUL`, or default values set for `SAND`, `CLAY` and `CARI`.

5.4.16.5.8. Suction data

5.4.16.5.9. If `ISTYP = COUL, SAND` or `CLAY` the input is given as:

```
FLAG ISU RFO RC1 RC2 RC3 RC4
```

- `FLAG: character(4)`: = `SSUC`
- `ISU: integer, default: 0`: Option for including soil suction
  - `= 0`: No suction
  - `= ± 1`: Suction force is related to vertical penetration force (static capacity) at actual penetration.
    - Reference length is the distance from the seafloor to the origin for vertical contact, but not more than one diameter.
  - `= ± 2`: Suction force is related to then maximum contact force since last lift off (or laydown).
    - Reference length is the distance from the seafloor to the origin for vertical contact, but not more than one diameter.
  - `= ±`3 : Suction force is related to $\mathrm{S_u} \times \mathrm{D}$
    - `= -3`: Reference length is footprint.
    - `= +3`: Reference length is diameter.
- `RF0: real, default: 0.25`: Suction force factor $[1]$
- `RC1: real, default: 0.7`: Total elongation $[1]$
- `RC2: real, default: 0.12`: Elastic fraction $[1]$
- `RC3: real, default: 0.7`: Degradation fraction $[1]$
- `RC4: real, default: 0.75`: Chusion force factor $[1]$

This data group is dummy for `ISTYP = COUL` or `SAND`.

The vertical extent of suction over the origin for vertical contact is to `RC1` $\times$ reference length.

- The `RC2` fraction of `RC1` $\times$ reference length is the elastic fraction.
- The `RC3` fraction of `RC1` $\times$ reference length is the degradation fraction.

The maximum suction force is `RC0` $\times$ reference force. The maximum chusion force is `RC4` $\times$ `RC0` $\times$ reference force.

======= If `ISTYP = CARI` the input is given as:

`FLAG ISU CC TCO PFO PFN`

- `FLAG: character(4)`: = `SSUC`
- `ISU: integer, default: 1`: Suction option
  - `= 0`: off
  - `> 0`: on
- `CC: real, default: 0.63E7`: Consolidation coefficient $[\mathrm{T}/\mathrm{L}^2]$
- `TCO: real, default: 0.432E5`: Initial consolidation time $[\mathrm{T}]$
- `PF0: real, default: -1`: Initial contact force expressed in penetration $[1]$
  - `> 0`: penetration given in diameters
  - `< 0`: penetration to trench depth
- `PFN: real, default: 0`: Minimum contact force to be used expressed in penetration $[1]$
  - `> 0`: penetration in diameters
  - `< 0`: penetration to trench depth

5.4.16.5.10. Damping data

`FLAG RDA TREF DHFAC`

- `FLAG: character`: = `SDAM`
- `RDA: real, default: See below`: Effective soil damping ratio $[1]$
- `TREF: real, default: 10`: Reference period for damping ratio $[\mathrm{s}]$

`DHFAC: real, default: 1`: Ratio of longitudinal and transverse damping to vertical damping

Stiffness proportional damping is applied in the vertical direction with the coefficient $\mathrm{a2}$ found as $\mathrm{a2} = \mathrm{RDA} \times \mathrm{TREF}/\pi$. The $\mathrm{a2}$ coefficients in the longitudinal and transverse directions are scaled by `DHFAC`.

The default value of `RDA` depends on the soil type and grading: - `SAND`: - `LOOS`: 0.14 - `MEDI`: 0.10 - `DENS`: 0.08 - `Other`: 0.14 - `CLAY / CARI`: - `LOOS`: 0.17 - `SOFT`: 0.14 - `STIF`: 0.06 - `HARD`: 0.02 - `Other`: 0.17

For `CARI`, additional vertical damping may come from suction.

5.4.16.5.11. Trench data

```
FLAG NTRPT TRSHI TRLSC TRDSC
```

- `FLAG: character`: = `TREN`
- `NTRPT: integer`: Number of points in the trench input table (⇐ 200 in present version)
- `TRSHI: real, default: 0`: Trench shift parameter $[\mathrm{L}]$
- `TRLSC: real, default: 1`: Trench length scale factor $[1]$
- `TRDSC: real`: Trench depth scale factor $[1]$

Trench profile, one input line per distance from supernode 1, i.e. `NTRPT` input lines ~ X TD TW ~

- `X: real`: Arc distance from supernode 1
- `TD: real`: Depth - to be scaled by `TRLSC`
- `TW: real`: Excess width $[\mathrm{L}]$

Total trench width will be TW + pipe diameter.

5.4.16.5.12. Additional nonlinear spring. Only if IADSPR=1

```
ALFU BETU ALFV BETV
```

- `ALFU: real, default: 0.01`: Number of points in the trench input table (⇐ 200 in present version)
- `BETU: real, default: 0.1`: Additional offloading axial stiffness ratio
- `ALFV: real, default: 0.01`: Additional onloading transverse stiffness ratio

- `BETV: real, default: 0.1`: Additional offloading transverse stiffness ratio

5.4.16.5.13. Default values for sandy soils:

Grading	PHI	WSOI	POI	ES	RDA
	ϕs	W_{soil}	n	e_s	ξd
	$[Deg]$	$[kN/m^3]$	-	-	$[\%]$
LOOSe	30	9.1	0.35	0.7	14
MEDIum	35	9.6	0.35	0.5	10
DENSe	40	10.1	0.35	0.4	8

5.4.16.5.14. Default values for clay (OCR=1):

Grading	SU	WSOI	POI	ES	RDA	PIX
	S_u	W_{soil}	n	e_s	ξd	i_p
	$[kN/m^3]$	$[kN/m^3]$	-	-	$[\%]$	$[\%]$
LOOSe	5	4.4	0.45	2.0	17	60
SOFT	17	5.4	0.45	1.8	14	55
STIFf	70	7.4	0.45	1.3	6	35
HARD	280	9.4	0.45	0.8	2	20

5.4.17. Drag chain element

The drag chain element is a single node element that models a simplified contact between a drag chain and the seafloor.

5.4.17.1. Data group identifier, one input line

```
NEW COMPonent DRAGchain
```

5.4.17.2. Component type identifier, one input line

```
CMPTYP-ID
```

- `CMPTYP-ID: character(8)`: Component type identifier

5.4.17.3. Drag chain element properties, one input line

```
LDC WDC FRDC LCAB WCAB
```

- `LDC: real`: Drag chain length $[\mathrm{L}]$
- `WDC: real`: Drag chain weight $[\mathrm{F}/\mathrm{L}]$
- `FRDC: real`: Chain / seafloor friction coefficient $[1]$
- `LCAB: real, default: 0`: Cable length $[\mathrm{L}]$
- `WCAB: real, default: 0`: Cable weight $[\mathrm{F}/\mathrm{L}]$

5.4.18. Fibre rope cross section

5.4.18.1. Data group identifier

```
NEW COMPonent FIBRe_rope
```

5.4.18.2. Component type identifier

```
CMPTYP-ID TEMP ALPHA BETA
```

- `CMPTYP-ID: character(8)`: Component type identifier
- `TEMP: real, default: 0`: Temperature at which the specification applies $[\mathrm{Temp}]$
- `ALPHA: real, default: 0`: Thermal expansion coefficient $[\mathrm{Temp}^{-1}]$
- `BETA: real, default: 0`: Pressure expansion coefficient $[1/(\mathrm{F}/\mathrm{L}^2)]$
  -

`BETA` gives the expansion of an element with zero effective tension from the difference between the internal and the external pressure.

5.4.18.3. Mass and volume

```
AMS AE R_EXTCNT
```

- `AMS: real`: Mass/unit length $[\mathrm{M}/\mathrm{L}]$
- `AE: real`: External cross-sectional area $[\mathrm{L}^2]$
- `R_EXTCNT: real, default: 0`: External contact radius $[\mathrm{L}]$

The outer contact radius of the cross section, `R_EXTCNT`, is used for seafloor contact. The default value of `R_EXTCNT` is zero.

5.4.18.4. Stiffness properties classification

```
NOC NOWC NWC TMAX
```

- `NOC: integer, default: 0`: Original curve, number of point pairs
- `NOWC: integer, default: 0`: Original working curve, number of point pairs
- `NWC: integer, default: 0`: Working curve, number of point pairs
- `TMAX: real, default: 0`: Maximum mean tension $[\mathrm{F}]$

The non-linear material curve used in static analysis is given by shifting the working curve by redefining the initial stress-free length so that the working and original working curves intersect at tension `TMAX`. See figure Tension strain curves.

5.4.18.5. Axial stiffness curves

```
EAF(1) ELONG(1) . . . EAF(N) ELONG(N)
```

- `EAF(1): real`: Tension corresponding to strain `ELONG(1)` $[\mathrm{F}]$
- `ELONG(1): real`: Strain $[\%]$

The pairs of `EAF` and `ELONG` must be given in increasing order on a single input line. Three sets of pairs must be given, for the working curve, original working curve and original curve, respectively. Each curve must begin with the point pair (0.0, 0.0). For the three curves, `N` = `NOC`, `N` = `NOWC` and `N` = `NWC`, respectively.

5.4.18.6. Dynamic stiffness coefficients

`DSCA DSCB`

- `DSCA: real, default: 1.0`: Dynamic stiffness coefficient a
- `DSCB: real, default: 0.0`: Dynamic stiffness coefficient b

The linear material curve used in dynamic analysis is given by $\mathrm{DSCA} + \mathrm{DSCB} \cdot \mathrm{TMEAN}$, where `TMEAN` is the mean tension of the segment, and by redefining the initial stress-free length such that the tension is identical between static and dynamic analysis given the elongation of static analysis. See figure Tension strain curves.

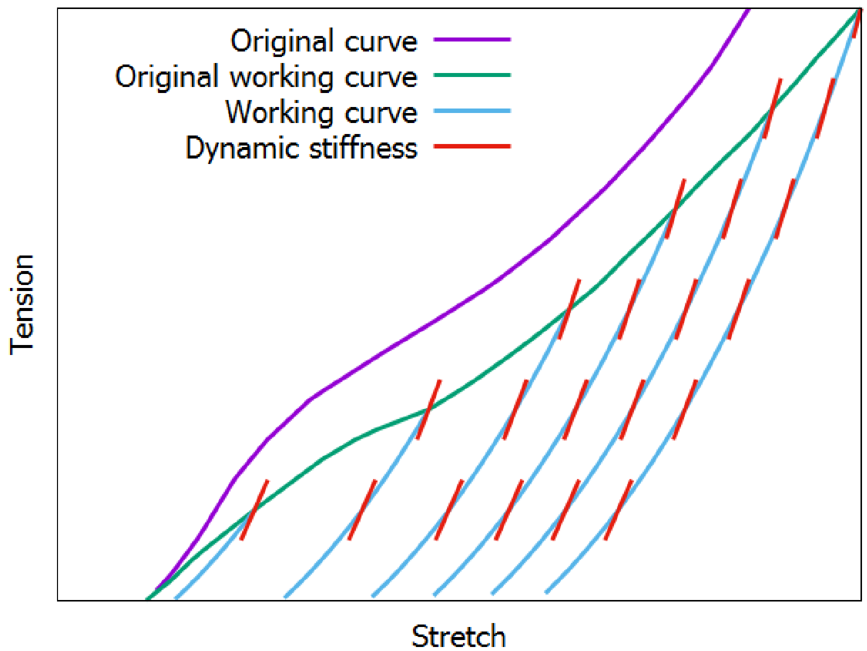

Figure 30. Tension strain curves

5.4.18.7. Damping specification

Identical to Damping specification for CRS1

5.4.18.8. Hydrodynamic force coefficients

Similar to Hydrodynamic force coefficients for CRS1, but only Morison type loading is available.

5.4.18.9. Capacity parameter

Identical to Capacity parameter for CRS1

5.4.19. Growth - Specification of marine growth profile

5.4.19.1. Data group identifier, one input line

```
NEW COMPonent GROWth
```

5.4.19.2. Component type identifier, one input line

```
CMPTYP-ID    NGRLEV
```

- `CMPTYP-ID: character(8)`: Component type identifier
- `NGRLEV : integer`: Number of growth levels

5.4.19.3. Growth profile, one input line per growth level, i.e. NGRLEV input lines

```
GRLEV GRTH GRDENS
```

- `GRLEV: real`: Z coordinate of level given in global coordinate system $[\mathrm{L}]$
- `GRTH: real`: Growth thickness $[\mathrm{L}]$
- `GRDENS: real`: Growth density at this level $[\mathrm{M}/\mathrm{L}^3]$

The input lines must be given for decreasing values of `GRLEV`; i.e. with increasing depth. Linear interpolation will be used to find values at intermediate levels. Outside the specified range, the growth thickness is set to zero, i.e. for `Z > GRLEV(1)` and `Z < GRLEV(NGRLEV)` the thickness is zero.

Marine growth will be applied to elements with `CRS0`, `CRS1`, `CRS2`, `CRS3`, `CRS4` and `CRS7` cross-sections.

The volume loads will be modified if the external area is non-zero. A circular cross-section is assumed and the thickness of the marine growth is added to the radius corresponding to the initial external area.

The Morison quadratic drag and added mass coefficients will be modified if the hydrodynamic diameter is non-zero. For `CRS2`, `CRS4` and `CRS7` cross-sections, the diameter of a circular cross-sections with the same external area is used as the hydrodynamic diameter. The quadratic drag coefficients will be scaled by the ratio of the updated to the initial hydrodynamic diameter. The added mass coefficients will be scaled by the square of this ratio. Linear drag coefficients will not be modified.

Marine growth will be applied if it is specified as a load group in `STAMOD`. The load incrementation procedure is specified as input to the `STAMOD` module.

Currently, only one growth profile may be given.

Note that specification of marine growth profile cannot be used in combination with drag amplification (@ref stamod_sawu).

5.5. Data Group D: Environmental Data

A complete environmental description consists of environmental constants, wave and current data. When an environment description has been completed, a new one may be given by repeating data groups `Identification of the environment' to `Current parameters' with the appropriate data for the new environment. Up to 10 complete environmental descriptions may be given as input to `INPMOD` in one run each identified by an unique identifier given in data group `Identification of the environment'. The minimum data required in one environmental condition is environmental constants (i.e. data groups `Water depth and wave indicator' and `Environment constants' are required).

Note that this data group is dummy for coupled analysis.

5.5.1. Identification of the environment

5.5.1.1. Data group identifier, one input line

```
ENVIronment IDENtification
```

5.5.1.2. Describing text. One input line

```
< TEXT >
```

- `character(60)`:

Description of the environment by alphanumerical text. Note: May be empty, but must be present.

5.5.1.3. Data-set identifier. One input line

```
IDENV
```

- `IDENV: character(6)`: Data set identifier for this environment description. Each environment must have a unique identifier.

5.5.2. Water depth and wave indicator

5.5.2.1. Data group identifier, one input line

```
WATErdepth AND WAVEtype
```

5.5.2.2. Water depth and control parameters. One input line

```
WDEPTH NOIRW NORW NCUSTA NWISTA
```

- `WDEPTH: real`: Water depth $[\mathrm{L}]$
- `NOIRW: integer`: Number of irregular wave cases, maximum 10
- `NORW: integer`: Number of regular wave cases, maximum 10
- `NCUSTA: integer`: Number of current states, maximum 10
- `NWISTA: integer, default: 0`: Number of wind states, maximum 10

`WDEPTH>0`. This water depth is defined as a scalar. This parameter is used in calculation of water particle motions.

An environment description can contain up to 10 irregular wave cases and 10 regular wave cases. A uniquely defined environment used in `STAMOD` or `DYNMOD` must refer to the actual environment by the identifier `IDENV` and wave case number.

If a numerically defined spectrum is used, `IWASP1=5` in Irregular wave control, the number of irregular wave cases is limited to `NOIRW=1`.

If a current line (Spatially varying current) is specified, then `NCUSTA` must be set to the total number of current profiles given.

5.5.3. Environment constants

5.5.3.1. Data group identifier, one input line

```
ENVIronment CONStants
```

5.5.3.2. Constants. One input line

```
AIRDEN WATDEN WAKIVI AIRKIVI
```

- `AIRDEN: real > 0`: Air density $[\mathrm{M}/\mathrm{L}^3]$
- `WATDEN: real > 0`: Water density $[\mathrm{M}/\mathrm{L}^3]$
- `WAKIVI: real, default:` 1.188×10^{-6}: Kinematic viscosity of water $[\mathrm{L}^2/\mathrm{T}]$
- `AIRKIVI: real, default:` 1.516×10^{-5}: Kinematic viscosity of air $[\mathrm{L}^2/\mathrm{T}]$

Typical values of `AIRDEN` and `WATDEN` are $[\mathrm{AIRDEN} = 1.3\mathrm{kg}/\mathrm{m}^3]$ and $[\mathrm{WAYDEN} = 1025\mathrm{kg}/\mathrm{m}^3]$ if the units `m' and `kg' are used.

5.5.4. Irregular waves

This data group is given only if `NOIRW > 0`, and is then repeated `NOIRW` times.

5.5.4.1. Data group identifier, one input line

```
NEW IRREgular SEAState
```

5.5.4.2. Irregular wave control data

5.5.4.2.1. Irregular wave control

```
NIRWC IWASP1 IWADR1 IWASP2 IWADR2
```

- `NIRWC: integer`: Irregular wave case number
- `IWASP1: integer`: Wave-spectrum type (wind sea)
    - `IWASP1=1`: Two-parameter Pierson-Moscowitz type spectrum
    - `IWASP1=2`: One-parameter Pierson-Moscowitz type spectrum
    - `IWASP1=3`: Jonswap spectrum
    - `IWASP1=4`: Derbyshire-Scott spectrum
    - `IWASP1=5`: Numerically defined spectrum
    - `IWASP1=6`: Ochi spectrum
        - To be used only for SI and modified SI units
    - `IWASP1=7`: Bretschneider I spectrum
        - To be used only for SI and modified SI units
    - `IWASP1=8`: Bretschneider II spectrum
        - To be used only for SI and modified SI units
    - `IWASP1=9`: Three parameter Jonswap spectrum
        - To be used only for SI and modified SI units
    - `IWASP1=10`: Double peaked spectrum (Torsethaugen)
        - To be used only for SI and modified SI units
- `IWADR1: integer`: Wave-direction code (wind sea)

- `IWADR1=0`: Unidirectional
  - `IWADR1>1`: Cosine-spreading function, `IWADR1` directions are used.
  - `IWADR1=1`: Cosine-spreading function, 11 directions are used. `IWADR1=1` is thus equivaent to specifying `IWADR1=11`.

- `IWASP2: integer`: Wave-spectrum type (swell)
  - `IWASP2=0`: No swell spectrum
  - For interpretation of other values, see `IWASP1` above
- `IWADR2: integer`: Wave-direction code (swell) (dummy if `IWASP2=0`)
  - `IWADR2=0`: Unidirectional
  - `IWADR2>1`: Cosine-spreading function, `IWADR2` directions are used.
  - `IWADR2=1`: Cosine-spreading function, 11 directions are used. `IWADR2=1` is thus equivaent to specifying `IWADR2=11`.

Bretschneider I is based on fetch and wind speed. Bretschneider II is based on wave height and wave period.

For `IWADR1 > 0`, the directions will be evenly spaced around the average wave propagation direction `WADIR1` at intervals of 180/(`IWADR1`+1) degrees. Specifying an even numbers of directions should be avoided as the average wave propagation direction will not be included in this case. The same applies to `IWADR2`.

5.5.4.3. Wave spectrum parameters (wind sea)

5.5.4.3.1. Data group identifier, one input line

```
WAVE SPECtrum WIND
```

5.5.4.3.2. Spectrum parameters

One of (i), (ii), …., (x) is given, depending on the value of the `IWASP1` parameter in the data group Irregular wave control data above.

======= (i) Two-parameter Pierson-Moscowitz (`IWASP1=1`), one input line.

```
SIWAHE AVWAPE
```

- `SIWAHE: real`: Significant wave-height, H_S [L]
-

`AVWAPE: real > 0`: Zero-crossing wave-period, $\mathrm{T_Z}$ $[\mathrm{T}]$

1. - $\mathrm{S}_\eta(\omega) = \mathrm{A}\omega^{-5}\mathrm{exp}[-^{\mathrm{B}}/\omega^4]; 0 < \omega < \infty$

   - $\mathrm{A} = 124.2\mathrm{H_S^2}/\mathrm{T_Z^4}$ - $\mathrm{B} = 496/\mathrm{T_Z^4}$

The relation between peak period, $\mathrm{T_p}$ and zero-crossing period is $\mathrm{T_Z} \approx \mathrm{T_p}/1.408$

======= (ii) One-parameter Pierson-Moscowitz (`IWASP1=2`), one input line

`SIWAHE`

- `SIWAHE: real > 0`: Significant wave-height, $\mathrm{H_S}$ $[\mathrm{L}]$

1. - $\mathrm{S}_\eta(\omega) = \mathrm{A}\omega^{-5}\mathrm{exp}[-^{\mathrm{B}}/\omega^4]; 0 < \omega < \infty$

   - $\mathrm{A} = 0.0081\mathrm{g}^2$ - $\mathrm{B} = 3.11/\mathrm{H_S^2}$

======= (iii) Jonswap spectrum (`IWASP1=3`), one input line

`PEAKFR ALPHA BETA GAMMA SIGMAA SIGMAB`

- `PEAKFR: real > 0`: Peak frequency (wp) $[\mathrm{radians}/\mathrm{T}]$
- `ALPHA: real, default: 0.008`: Phillip's constant
- `BETA: real, default: 1.25`: Form parameter
- `GAMMA: real, default: 3.3`: Peakedness parameter giving the ratio of the maximum spectral energy to that of the corresponding Pierson- Moscowitz spectrum
  - `0 < GAMMA ⇐ 20`
- `SIGMAA: real > 0, default: 0.07`: Spectrumwidth parameter
- `SIGMAB: real > 0, default: 0.09`: Spectrumwidth parameter

1. - $\mathrm{S}_\eta(\omega) = \alpha\mathrm{g}^2\omega^{-5}\mathrm{exp}(-\beta(\frac{\omega_\mathrm{p}}{\omega})^4) \times \gamma^{\mathrm{exp}(-\frac{(\omega-\omega_\mathrm{p})^2}{2\sigma^2\omega_\mathrm{p}^2})}$

   - $\alpha = 1.2905\mathrm{H_S^2}/\mathrm{T_Z^4}$ - $\beta = 1.25$ for North Sea conditions - $\gamma = \begin{cases} 1.0; \quad \mathrm{T_p} >= 5\sqrt{\mathrm{H_S}} \\ \mathrm{exp}(5.75 - 1.15\mathrm{T_p}/\sqrt{\mathrm{H_S}}) \\ 5.0; \quad \mathrm{T_p} < 3.6\sqrt{\mathrm{H_S}} \end{cases}$

   - $\sigma = \begin{cases} \sigma_\mathrm{a} = 0.07 \quad \text{for} \quad \omega <= \omega_p \\ \sigma_\mathrm{b} = 0.09 \quad \text{for} \quad \omega <= \omega_p \end{cases}$

- $\omega_\mathrm{p} = \frac{2\pi}{\mathrm{T_p}}$ - $\frac{\mathrm{T_p}}{\mathrm{T_Z}} = 1.407(1 - 0.287\ln\gamma)^{1/4}$

======= (iv) Derbyshire-Scott spectrum (`IWASP1=4`), one input line

`SPEC1 SPEC2 SPEC3 SIWAHE AVWAPE TRUNCL TRUNCU`

- `SPEC1: real, default: 0.214`: Spectrum parameter, a $[\mathrm{T/rad}]$
- `SPEC2: real > 2, default: 0.065`: Spectrum parameter, b $[\mathrm{rad/T}]$
- `SPEC3: real, default: 0.26`: Spectrum parameter, d $[\mathrm{rad/T}]$
- `SIWAHE: real`: Significant wave height, $\mathrm{H_S}$ $[\mathrm{L}]$
- `AVWAPE: real > 0`: Average wave period, T $[\mathrm{T}]$
- `TRUNCL: real, default: 0.0414`: Lower truncation parameter $[\mathrm{radians/T}]$
- `TRUNCU: real, default: 10.367`: Upper truncation parameter $[\mathrm{radians/T}]$
- $\mathrm{S}_\eta(\omega) = \alpha \mathrm{H_S^2} \mathrm{exp}\sqrt{\frac{(\omega - \omega_\mathrm{p})^2}{\mathrm{b}(\omega - \omega_\mathrm{p} + \mathrm{d})}}$ for $\mathrm{TRUNCL} < \omega < \mathrm{TRUNCU}$

======= (v) Numerically defined spectrum (`IWASP1=5`)

Both (v.1) and (v.2) must be given.

(v.1) Number of discrete frequencies, one input line. ~ NDFRQ1 ~

- `NDFRQ1: integer >= 4`: Number of discrete frequencies

(v.2) Spectrum values, NDFRQ input lines. Either: ~ FRQ DSPDEN ~

- `FRQ: real`: Frequency $[\mathrm{radians/T}]$
- `DSPDEN: real`: Associated discrete spectral density value $[\mathrm{L^2T}]$

The input lines must be given in sequence of increasing frequency values.

======= (vi) Ochi spectrum (`IWASP1=6`), one input line.

`SIWAHE`

- `SIWAHE: real`: Significant wave height $[\mathrm{L}]$

======= (vii) Bretschneider spectrum I (`IWASP1=7`), one input line

`FETCH WISPD`

- `FETCH: real`: Fetch $[\mathrm{L}]$
- `WISPD: real`: Wind speed $[\mathrm{L}/\mathrm{T}]$

======= (viii) Bretschneider spectrum II (`IWASP1=8`), one input line

`SIWAHE SIWAPE`

- `SIWAHE: real`: Significant wave height $[\mathrm{L}]$
- `SIWAPE: real > 0`: Significant wave period $[\mathrm{T}]$

======= (ix) Three parameter JONSWAP spectrum (`IWASP1=9`), one input line.

`SIWAHE PEAKPE GAMMA`

- `SIWAHE: real`: Significant wave height $[\mathrm{L}]$
- `PEAKPE: real > 0`: Peak period $[\mathrm{T}]$
- `GAMMA: real, default: see below`: Peakedness parameter giving the ratio of the maximum spectral energy to that of the corresponding Pierson-Moscowitz spectrum
  - `0< GAMMA <= 20`

Default value of `GAMMA` is calculated from `SIWAHE` and `PEAKPE`, see (iii) Jonswap spectrum (`IWASP1=3`):

$$\mathrm{GAMMA} = \exp[5.75 - 1.15 \times \frac{\mathrm{PEAKPE}}{\sqrt{\mathrm{SIWAHE}}}]$$

$$1 <= \mathrm{GAMMA} <= 5$$

Note that use of the three parameter `JONSWAP` spectrum requires that the SI units m and s be used.

======= (x) Double peaked JONSWAP spectrum (`IWASP1=10`) (described by Torsethaugen) , one input line.

`SIWAHE PEAKPE`

- `SIWAHE: real`: Significant wave height $[\mathrm{L}]$
- `PEAKPE: real > 0`: Peak period $[\mathrm{T}]$

Note that use of the double peaked `JONSWAP` spectrum requires that the SI units m and s be used.

5.5.4.4. Wave spectrum parameters (swell)

This data group is omitted for `IWASP2=0`, see Irregular wave control data (no swell present).

5.5.4.4.1. Data group identifier, one input line

`WAVE SPECtrum SWELl`

5.5.4.4.2. Spectrum parameters

One of (i), (ii), …, (x) is given, depending on the value of the IWASP2 parameter given in data group Irregular wave control data. The input is identical to input of wind sea spectrum and is therefore not repeated, see Wave spectrum parameters (wind sea).

5.5.4.5. Direction parameters of waves

5.5.4.5.1. Data group identifier, one input line

`DIRECTION PARAMETERS`

The two input lines below (`Wave direction parameters (wind sea)' and `Wave direction parameters (swell)') must be given in sequence if both are present.

5.5.4.5.2. Wave direction parameters (wind sea), one input line

`WADR1 EXPO1`

- `WADR1: real`: Average propagation direction of waves, measured in degrees from the global X-axis.
  - Confer the figure Location of support vessel coordinate system
- `EXPO1: real`: Exponent of cosine distribution
  - dummy if `IWADR1=0`, see Irregular wave control data.

If `IWADR1 > 0`, a cosine directional spreading function is used: $f(\alpha_i) = \frac{[\cos(\alpha_i - \mathrm{WADR1})]^{\mathrm{EXPO1}}}{\sum[f(\alpha_j)]}$ where α_i is one of the `IWADR1` short-crested wave directions. The sum in the denominator is taken over all `IWADR1` directions. The total wind sea energy is thus kept.

5.5.4.5.3. Wave direction parameters (swell), one input line

This data group is omitted for `IWASP2=0`, see Irregular wave control data (no swell present).

`WADR2 EXPO2`

- `WADR2: real`: Average propagation direction of waves, measured in degrees from the global X-axis.

Confer the figure Location of support vessel coordinate system

- `EXPO2: real`: Exponent of cosine distribution
  - dummy if `IWADR2=0`, see Irregular wave control data.

If `IWADR2 > 0`, a cosine directional spreading function is used: $\mathrm{f}(\alpha_\mathrm{i}) = \frac{[\cos(\alpha_\mathrm{i} - \mathrm{WADR2})]^{\mathrm{EXPO2}}}{\sum[\mathrm{f}(\alpha_\mathrm{j})]}$ where α_i is one of the `IWADR2` short-crested wave directions. The sum in the denominator is taken over all `IWADR2` directions. The total swekk energy is thus kept.

5.5.5. Regular waves

This data group is given only if `NORW > 0`.

5.5.5.1. Data group identifier, one input line

```
REGULAR WAVE DATA
```

5.5.5.2. Regular wave data, NORW input lines

```
INRWC AMPLIT PERIOD WAVDIR
```

- `INRWC: integer`: Regular wave case number
- `AMPLIT: real`: Wave amplitude $[\mathrm{L}]$
- `PERIOD: real > 0`: Wave period $[\mathrm{T}]$
- `WAVDIR: real`: Wave propagation direction from the global X-axis $[\mathrm{deg}]$
  - Confer the figure Location of support vessel coordinate system

5.5.6. Current parameters

This data group is given only if `NCUSTA > 0`, and is then repeated `NCUSTA` times.

5.5.6.1. Data group identifier, one input line

May be omitted if no current is present for actual environment.

```
NEW CURRENT STATE
```

5.5.6.2. Current dimension parameter, one input line

```
ICUSTA NCULEV L_EXT
```

- `ICUSTA: integer`: Current state number
- `NCULEV: integer`: Number of current levels
- `L_EXT: integer, default: 0`: Flag to indicate if current data is given in this input file, or if it shall be read from an external file.
  - For details on the format of the external file, confer CURMOD User's Documentation.
  - `0`: Data specified on this file
  - `1`: Data specified on external file

`1 ⇐ NCULEV ⇐ 30`. Current states must be given in increasing order, i. e. `1,2, …, NCUSTA`

5.5.6.3. Current profile, one input line per current level, i.e. NCULEV input lines

5.5.6.3.1. This data group is given only if `L_EXT = 0`

`CURLEV CURDIR CURVEL`

- `CURLEV: real`: Z coordinate of level given in global coordinate system $[\mathrm{L}]$
- `CURDIR: real`: Current velocity direction at this level. The angle is measured in degrees from the global X-axis counter-clockwise around the global Z-axis. (seen from above)
- `CURVEL: real`: Current velocity at this level $[\mathrm{L}/\mathrm{T}]$

The input lines must be given in sequence of decreasing Z coordinates. Linear interpolation is applied between the levels. Outside the specified range of levels a ``flat'' extrapolation is used, i.e. for `Z > CURLEV(1)`` the velocity is set to `CURVEL(1)` and for `Z < CURLEV(NCULEV)` the velocity is set to `CURVEL(NCULEV)`

This current profile may be scaled when applied in static or dynamic analysis.

Z coordinate is zero at mean water level and negative below sea surface.

5.5.6.3.2. This data group is given only if `L_EXT = 1`

`CURRFILE`

- `CURRFILE: character(120)`: Name of external file with specified current data

5.5.7. Spatially varying current

5.5.7.1. Data group identifier, one input line

`NEW CURRENT LINE`

5.5.7.2. Current line control parameters, one input line

`ICUSTA NPT`

- `ICUSTA: integer`: Current state number
- `NPT: integer`: Number of current profiles given

The number of current states `NCUSTA` (see Water depth and control parameters) must be increased by `NPT` for each current line specified.

Current states must be given in ascending order

5.5.7.3. Current dimension parameters, one input line

`IPT NCULEV XPT YPT`

- `IPT: integer`: Current profile number. Must be given from 1 to `NPT` consecutively
- `NCULEV: integer`: Number of current levels
  - `1 < NCULEV ⇐ 30`
- `XPT: real`: Global X- and Y- coordinates
- `YPT: real`: For which this current profile is specified

5.5.7.4. Current profile, one input line per current level, i.e. NCULEV input lines

`CURLEV CURDIR CURVEL`

- `CURLEV: real`: Z coordinate of level given in global coordinate system $[\mathrm{L}]$
- `CURDIR: real`: Current velocity direction at this level. The angle is measured in degrees from the global X-axis counter-clockwise around the global Z-axis. (seen from above)
- `CURVEL: real`: Current velocity at this level $[\mathrm{L}/\mathrm{T}]$

The input lines must be given in sequence of decreasing Z coordinates. Linear interpolation is applied between the levels. Outside the specified range of levels a `` `flat'' extrapolation is used, i.e. for `Z > CURLEV(1)`` the velocity is set to `CURVEL(1)` and for `Z < CURLEV(NCULEV)` the velocity is set to `CURVEL(NCULEV)`

This current profile may be scaled when applied in static or dynamic analysis.

Z coordinate is zero at mean water level and negative below sea surface.

5.5.8. Wind parameters

This data group is given only if `NWISTA > 0`, and is then repeated `NWISTA` times.

5.5.8.1. Data group identifier, one input line

May be omitted if no wind is present for actual environment.

```
NEW WIND SPECification
```

5.5.8.2. Wind case number, one input line

```
IWISTA
```

- `IWISTA: integer`: Wind case number

5.5.8.3. Wind type, one input line

```
IWITYP
```

- `IWITYP: integer`: Wind type
  - `IWITYP=10`: Stationary uniform wind with shear, values interpolated at grid points
  - `IWITYP=11`: Fluctuating uniform 2-component wind
  - `IWITYP=12`: Fluctuating 3-comp. wind read from files (IECWind format)
  - `IWITYP=13`: Fluctuating 3-comp. wind read from files (TurbSim Bladed style format)
  - `IWITYP=14`: Stationary uniform wind with shear

The wind types 10 - 14 are intended for wind turbine analyses. However, they may also be applied for other type of analysis.

For the IECWind fluctuating 3-component wind (`IWITYP=12`), only the fluctuating part of the wind is given in the wind input files. The mean wind speed `UMEAN` given above is added to the yield the total wind velocity in the longitudinal direction. The input files must conform to the 3-dimensional 3-component wind time series from the rectangular IEC format (See Thomsen, K., 2006. Mann turbulence for the IEC Code Comparison Collaborative (OC3). Risø National Laboratory). More specifically they must include time series of wind velocity in binary format, with a 3-dimensional array having indices in vertical direction running fastest, then indices in lateral direction and indices in longitudinal direction running slowest.

For wind files from NREL’s TurbSim (`IWITYP=13`), the mean wind speed and shear are included in the binary files. The input files must be generated by TurbSim with WrBLFF=True. Both the .wnd file and .sum file are needed.

5.5.8.4. Wind type specifications

5.5.8.4.1. Stationary uniform wind with shear, values interpolated at grid points (`IWITYP=10`)

Wind direction, one input line ~ WIDIR ~

- `WIDIR: real`: Wind propagation direction in global XY-plane $[\mathrm{deg}]$

Wind velocity, one input line ~ UMVEL VMVEL WMVEL ~

- `UMVEL: real`: Longitudinal wind velocity component $[\mathrm{L}/\mathrm{T}]$
- `VMVEL: real`: Lateral wind velocity component $[\mathrm{L}/\mathrm{T}]$
- `WMVEL: real`: Vertical (global Z-axis) wind velocity component $[\mathrm{L}/\mathrm{T}]$

The parameters `UMVEL` and `VMVEL` refer to the direction given by the `WIDIR` parameter

Number of levels in shear profile, one input line ~ NZPROF ~

- `NZPROF: integer`: Number of vertical levels for defining the shear profile

Wind velocity profile definition, `NZPROF` input lines ~ ZLEV UFACT VFACT WFACT ~

- `ZLEV: real`: Vertical coordinate of profile level $[\mathrm{L}]$
- `UFACT: real`: Wind speed scaling factor for longitudinal wind velocity
- `VFACT: real`: Wind speed scaling factor for the lateral wind velocity
- `WFACT: real`: Wind speed scaling factor for the vertical wind velocity

Wind field domain location, one input line ~ Z0 ~

- `Z0: real`: Z coordinate of the lower edge of the wind field domain $[\mathrm{L}]$

Domain size, one input line ~ NZ ~

- `NZ: integer`: Number of grid points in Z- (vertical) direction

Domain resolution, one input line ~ DLWFZ ~

- `DLWFZ: real`: Domain resolution in the vertical direction $[\mathrm{L}]$

5.5.8.4.2. Fluctuation uniform 2-component wind read from file (`IWITYP=11`)

Wind direction, one input line ~ WIDIR ~

- `WIDIR: real`: Wind propagation direction in global XY-plane $[\mathrm{deg}]$

Wind data file name, one input lines ~ CHWIFI ~

- `CHWIFI: character(256)`: Path and filename for import of wind velocity time series. See the SIMO User Manual (`Reading wind time series from file' in `Initialization of time domain simulation' in `Use of DYNMOD') for explanation on file format.

5.5.8.4.3. Fluctuating 3-component wind field read from file (IWITYP=12)

Mean wind direction, one input line ~ WIDIR ~

- `WIDIR: real`: Wind propagation direction in global XY-plane $[\mathrm{deg}]$

Mean wind velocity, one input line ~ UMVEL ~

- `UMVEL: real`: Mean wind velocity along `WIDIR` $[\mathrm{L/T}]$

Number of levels in shear profile, one input line ~ NZPROF ~

- `NZPROF: integer`: Number of vertical levels for defining the shear profile

Wind velocity profile definition, `NZPROF` input lines ~ ZLEV UMFACT UFACT VFACT ZFACT ~

- `ZLEV: real`: Vertical coordinate of profile level $[\mathrm{L}]$
- `UMFACT: real`: Scaling factor for the mean wind velocity
- `UFACT: real`: Scaling factor for fluctuating part of the longitudinal wind velocity
- `VFACT: real`: Scaling factor for fluctuating part of the lateral wind velocity
- `ZFACT: real`: Scaling factor for fluctuating part of the vertical wind velocity

Name of file containing the fluctuating longitudinal wind time series, one input line ~ CHWFU ~

- `CHWFU: character(256)`: Path and filename for the fluctuating U-component wind time series

Name of file containing the fluctuating lateral wind time series, one input line ~ CHWFV ~

- `CHWFV: character(256)`: Path and filename for the fluctuating V-component wind time series

Name of file containing the fluctuating vertical wind time series, one input line ~ CHWFW ~

- `CHWFW: character(256)`: Path and filename for the fluctuating Z-component wind time series

Wind field domain location, one input line ~ X0LL Y0LL Z0LL ~

- `X0LL: real`: X-coordinate of the lower left corner of the upstream border of the wind field domain $[\mathrm{L}]$
- `Y0LL: real`: Y-coordinate of the lower left corner of the wind field domain $[\mathrm{L}]$
- `Z0LL: real`: Z-coordinate of the lower left corner of the wind field domain$[\mathrm{L}]$

These three coordinates defines the lower left corner of the wind field domain, which is defined as a rectangular cuboid. The coordinates refers to a coordinate system centred at the global origin, with the x-axis (longitudinal direction) pointing in the down-stream mean wind speed direction and the z-axis coincident with the global z-axis.

Domain size, one input line ~ NX NY NZ ~

- `NX: integer`: Number of grid points in X- (longitudinal) direction $[\mathrm{L}]$
- `NY: integer`: Number of grid points in Y- (lateral) direction $[\mathrm{L}]$
- `NZ: integer`: Number of grid points in Z- (vertical) direction$[\mathrm{L}]$

Field size, one input line ~ LWFX LWFY LWFZ ~

- `LWFX: real`: Field size in X- (longitudinal) direction
- `LWFY: real`: Field size in Y- (lateral) direction
- `LWFZ: real`: Field size in Z- (vertical) direction

Buffer size, one input line ~ NSLICE ~

- `NSLICE: integer, default: 10`: Buffer size: Number of wind crossectional planes (Slices) in memory

5.5.8.4.4. Fluctuating 3-component wind field read from TurbSim file (IWITYP=13)

The wind field domain- and field size are extracted from the TurbSim .sum file.

The wind field domain location in the global coordinate system is not given explicitly by the user. The vertical position of the wind field center is the same as in TurbSim; i.e. taken as the hub height given on the turbsim .sum file.

Horizontally, the wind field is positioned around the global origin, but with a half grid width downwind of the origin. Since the TurbSim wind is non-periodic, this is necessary to ensure that the entire turbine lies in the same part of the wind field at the start of the simulation. The wind at the global origin will thus not start at the first slice

The wind field must be large enough to ensure that the whole structure is within the wind field during the entire simulation. As the TurbSim wind field is nonperiodic, the beginning and end of the wind field will not fit together.

Mean wind direction, one input line ~ WIDIR ~

- `WIDIR: real`: Wind propagation direction in global XY-plane $[\mathrm{deg}]$

Name of binary (.wnd) file containing the TurbSim fluctuating wind time series, one input line ~ CHWFTW ~

- `CHWFTW: character(256)`: Path and filename for the binary TurbSim (.wnd) file

Name of the summary (.sum) file from TurbSim, one input line ~ CHWFTS ~

- `CHWFTS: character(256)`: Path and filename for the summary TurbSim (.sum) file

Buffer size, one input line ~ NSLICE ~

- `NSLICE: integer, default: 800`: Buffer size: Number of wind crossectional planes (Slices) in memory

Note: Since TurbSim files are not periodic, time series are shifted by 1/2 Grid Width. The number of slices in memory must be greater than (Grid Width/MeanWindSpeed/WindFileTimeStep).

5.5.8.4.5. Stationary uniform wind with shear (`IWITYP=14`)

Wind direction, velocity and shear profile type, one input line ~ WIDIR UMVEL WMVEL CH_SHEAR ~

- `WIDIR: real`: Wind propagation direction in global XY-plane $[\mathrm{deg}]$
- `UMVEL: real`: Longitudinal wind velocity component $[\mathrm{L/T}]$
- `WMVEL: real`: Vertical (global Z-axis) wind velocity component $[\mathrm{L/T}]$
- `CH_SHEAR: character(4)`: Shear profile type
    - `NONE` - No shear profile

- `POWR` - Power shear profile
- `LOGA` - Logarithmic shear profile

`UMVEL` is the wind velocity in the direction `WIDIR`.

Poser shear profile input, one input line, only given if `CH_SHEAR = POWR` ~ ZREF ALPHA ~

- `ZREF: real`: Reference height $[\mathrm{L}]$
- `ALPHA: real`: Wind shear exponent $[-]$

Logarithmic shear profile input, one input line, only given if `CH_SHEAR = LOGA` ~ ZREF Z0 ~

- `ZREF: real`: Reference height $[\mathrm{L}]$
- `Z0: real`: Roughness length $[\mathrm{L}]$

5.6. Data Group E: Support Vessel Data

Observe that the motion transfer function definition is related to the wave field definition. See `Motion Transfer Functions' in the Theory Manual for the definition of wave field and motion transfer functions. Transfer functions based on other definitions must be converted by appropriate phase shift operations before they are used as input to this program.

This data group needs not be given for systems with no vessel attachment points. Note that either `Support vessel data on file' or `Identification' through `Transfer function input' is to be given.

5.6.1. Support vessel data on file

5.6.1.1. Data group identifier, one line

```
TRANsfer FUNCtion FILE
```

5.6.1.2. File name

```
CHFTRA
```

- `CHFTRA: character(80)`: File name with transfer functions data

File with transfer function data given in RIFLEX format terminated with an `END` statement.

This group replaces the rest of group E if given. Either `Support vessel data on file' or `Identification' through `Transfer function input' should be given. If `Support vessel data on file' is given, the content on the file should be `Identification' to `Transfer function input' with an `END` termination.

5.6.2. Identification

5.6.2.1. Data group identifier, one input line

```
SUPPort VESSel IDENtification
```

5.6.2.2. Heading, one input line

```
Heading
```

Text describing the transfer functions.

Always one input line, which may be blank. The line may contain up to 60 characters.

5.6.2.3. Identifier, one input line

```
IDWFTR
```

- `IDWFTR: character(6)`: Identifier for transfer functions. The value `NONE' is not allowed.

5.6.3. Transfer function reference position

This data group is used as control parameter and compared with vessel position specified in Data Group B: Single Riser Data.

5.6.3.1. Data group identifier, one input line

```
HFTRan REFErence POSItion
```

5.6.3.2. Reference position, one input line

```
ZG
```

- `ZG: real`: Vertical position of the support vessel coordinate system. $[L]$
    - (The global Z coordinate for which the vessel motion transfer function is calculated.)
    - Confer the figure Location of support vessel coordinate system (below).
    - This parameter is used as control parameter and compared with `ZG` in data groups.

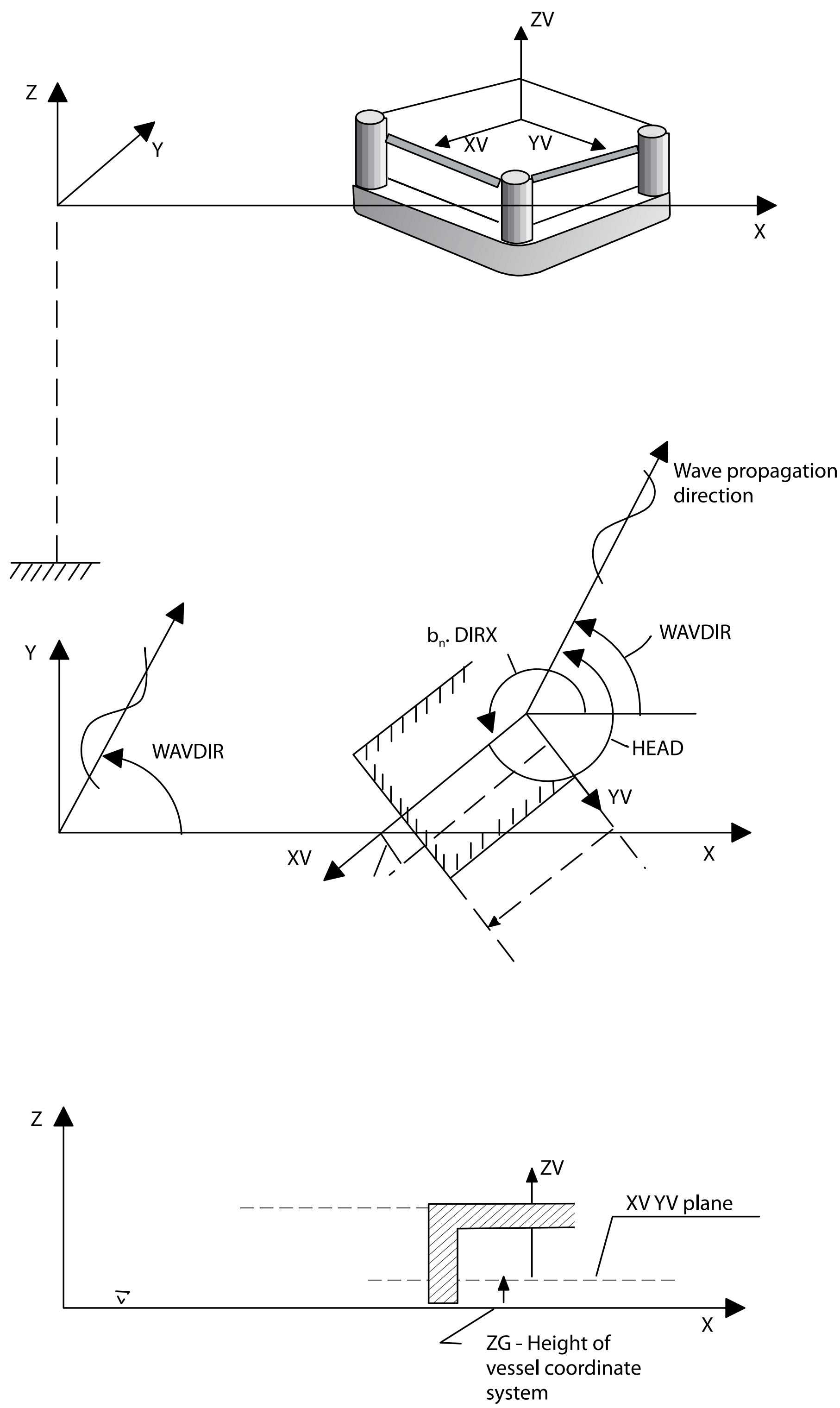

Figure 31. Location of support vessel coordinate system

5.6.4. Dimension parameter and input type code

5.6.4.1. Data group identifier, one input line

```
HFTRansfer CONTrol DATA
```

5.6.4.2. Dimension parameters, one input line

```
NDHFTR NWHFTR ISYMHF ITYPIN
```

- `NDHFTR: integer`: No. of directions for which transfer functions are given
  - `NDHFTR=1` or `NDHFTR>=4`
- `NWHFTR: integer`: No of frequencies for which transfer functions are given
  - `NWHFTR>=4`
- `ISYMHF: integer`: Symmetry code related to transfer functions
  - `ISYMHF=0`: No symmetry
  - `ISYMHF=1`: Symmetry about XV-ZV plane
  - `ISYMHF=2`: Symmetry about XV-ZV and YV-ZV planes
- `ITYPIN: integer`: Code for which format the HF-transfer function are given
  - `ITYPIN=1`: Non-dimensional complex form
  - `ITYPIN=2`: Non-dimensional amplitude ratio and phase, where phase is given in degrees
  - `ITYPIN=3`: Non-dimensional amplitude ratio and phase, where phase is given in radians

The complex form and the amplitude ratio are to be given as non-dimensional. This means: - L/L for freedoms surge, sway and heave - radian/radians or degrees/degrees for freedoms roll, pitch and yaw, giving motion angle/surface wave slope amplitude. The wave slope amplitude is defined by $\gamma_a = k\zeta_a$ where k is the wave number and ζ_a is the wave amplitude.

If `NDHFTR=1`, `ISYMHF` is dummy (set to zero by the program). In this case the specified transfer function is used, regardless of the wave direction. Note that the specified motions are applied in the local vessel coordinate system regardless of the wave direction; e.g. surge motions are applied in the local vessel x-direction. In practice, the wave direction will not have any effect on the vessel motions in this case. To rotate the motions with the wave direction, update the vessel heading when the wave direction is changed. This method must however be used with care, if the model is not symmetric and connected to the vessel in the local vessel origo, it may not give the desired effect.

5.6.5. Specifications of wave directions

Wave directions given in vessel coordinate system for input of motion transfer functions.

5.6.5.1. Data group identifier, one input line

```
WAVE DIREctions
```

5.6.5.2. Directions, NDHFTR input lines

```
IHEAD HEAD
```

- `IHEAD: integer`: Direction number
- `HEAD: real`: Direction. The angle `HEAD` is measured in degrees from the XV axis counter clockwise to the wave propagation vector

`IHEAD` and `HEAD` must be given in ascending order

If `NDHFTR=1`, `HEAD` is dummy.

If the directions do not cover a full circle, the transfer functions for the first direction will be repeated for the direction `HEAD(1) + 360`. For `ISYMHF`> 0, the last direction after mirroring is `360 - HEAD(1)`.

5.6.6. Specification of wave frequencies

5.6.6.1. Data group identifier, one input line

```
WAVE FREQuencies
```

5.6.6.2. Frequencies, NWHFTR input lines

```
IFREQ WHFTR
```

- `IFREQ: integer`: Frequency number
- `WHFTR: real`: Frequency $[\mathrm{rad/T}]$

Frequencies must be given in increasing order.

5.6.7. Transfer function input

5.6.7.1. Data group identifier, one input line

```
HFTRansfer FUNCtion "DOF"
```

`` `dof'' is either `SURGE, SWAY, HEAVE, ROLL, PITCH `` or `YAW`

5.6.7.2. Transfer function for HF ``dof'' motion, NDHFTR x NWHFTR input lines

`IDIR IFREQ A B`

- `IDIR: integer`: Direction number
- `IFREQ: integer`: Frequency number
- `A: real`: Interpretation according to value of `ITYPIN` (given in the data group Reference position, one input line)
  - `ITYPIN=1`: A - Real part
  - `ITYPIN=2`: A - Amplitude ratio
  - `ITYPIN=3`: A - Amplitude ratio
- `B: real`: Interpretation according to value of `ITYPIN` (given in the data group Reference position, one input line)
  - `ITYPIN=1`: B - Imaginary part
  - `ITYPIN=2`: B - Phase (degrees)
  - `ITYPIN=3`: B - Phase (radians)

Data group identifier, one input line is repeated for each degree of freedom included in motion description. If one (or more) degrees of freedom are omitted, they are set equal to zero.

For phase and sign convention, see `Motion Transfer Functions' in the Theory Manual.

If only one direction is specified (`NDHFTR=1`), the transfer function is used independent of incoming wave direction.

Amplitudes and phase angles at required frequencies are calculated by linear interpolation/extrapolation in the dynamic analysis. Transfer functions will therefore be extrapolated for spectral components outside the frequency range defined for the transfer function. Ensure that the amplitude values given for the two highest/lowest frequencies give physical realistic values when extrapolated. Add two zero amplitude components at both ends of the frequency range if no extrapolation is wanted.

Linear interpolation is also used for wave direction.

5.6.8. Termination of input data

Do not forget the `END` input line if this is the `` `last'' data group given in this `INPMOD `` run. See also Termination of input data.

END

5.7. Data Group F: Floater Force Model Data

This option enables user to perform coupled analysis which means: simultaneous analysis of vessel motions and mooring line and riser dynamics.

The vessel load model may account for wind, wave and current forces, which are applied as nodal loads. For further description of vessel load model, confer `SIMO`.

Note that his option requires access to the computer program `SIMO`.

5.7.1. Data group identifier, one input line

```
FLOAter FORCe MODEl
```

5.7.2. Number of SIMO bodies, one input line

```
NSBODY
```

- `NSBODY: integer`: The number of `SIMO` bodies
  - `NSBODY>0`

5.7.2.1. SIMO Body identification, location and optional artificial stiffness

These data are specified on input lines `SIMO Body identification' through `SIMO Body artificial stiffness' and must be given in one block for each `SIMO` body.

5.7.2.1.1. SIMO Body identification, SIMO body node identification and location option

```
CHBODY CHBODY_NOD_ID CHLOCA_OPT
```

- `CHBODY: character(8)`: `SIMO` Body identification
- `CHBODY_NOD_ID: character(8), default: 'SBDYi'`: Optional `SIMO` body node identifier, where `i` is the `RIFLEX` internal number of the `SIMO` body so that the first `SIMO` body will have a default `SIMO` body node identifer equal to `SBDY1'.
- `CHLOCA_OPT: character(4), default: 'ELEM'`: Optional `SIMO` body location option
  - `CHLOCA_OPT='ELEM'`: The `SIMO` body location is specified in terms of a line segment element end.
  - `CHLOCA_OPT='NODE'`: The `SIMO` body location is specified in terms of a line segment node.

`CHLOCA_OPT='POSI'`: The `SIMO` body location is specified in terms of a position.

`CHBODY_NOD_ID` is the `SIMO` body node identifier which is set automatically if not specified by the user. The automatic naming convention is based on concatenating the character string `'SBDY'` and the `RIFLEX` internal number of the `SIMO` body starting at 1 for the first `SIMO` body.

5.7.2.1.2. SIMO Body location, orientation and artificial stiffness option for CHLOCA_OPT=`ELEM'

The input line below must be specified for the option `CHLOCA_OPT='ELEM'`, and whenever `CHBODY_NOD_ID` and `CHLOCA_OPT` is omitted.

```
LINE-ID ISEG IEL IEND ROTX ROTY ROTZ IST
```

The `SIMO` body is connected to - `LINE-ID: character(8)`: Reference to line identifier - `ISEG: integer`: Local segment number within line - `IEL: integer`: Local element number within segment - `IEND: integer`: Element end (1 or 2)

The initial orientation of the `SIMO` body is: - `ROTX0: real, default: 0`: Rotation around X-axis [deg] - `ROTY0: real, default: 0`: Rotation around Y-axis [deg] - `ROTZ0: real, default: 0`: Rotation around global Z-axis [deg]

`ROTX0`, `ROTY0` and `ROTZ0` are the euler angles taken in the order `ROTZ0 → ROTY0 → ROTX0`.

- `IST: integer, default:0`: Artificial stiffness option
  - `IST=0`: No artificial stiffness
  - `IST=1`: Artificial stiffness is specified

5.7.2.1.3. SIMO Body location, orientation and artificial stiffness option for CHLOCA_OPT=`NODE'

The input line below is specified for the option `CHLOCA_OPT='NODE'` only.

```
LINE-ID ISEG ISEGNOD ROTX ROTY ROTZ IST
```

The `SIMO` body is connected to - `LINE-ID: character(8)`: Reference to line identifier - `ISEG: integer`: Local segment number within line - `ISEGNOD: integer`: Local node within segment

The initial orientation of the `SIMO` body is: - `ROTX0: real, default: 0`: Rotation around X-axis [deg] - `ROTY0: real, default: 0`: Rotation around Y-axis [deg] - `ROTZ0: real, default: 0`: Rotation around global Z-axis [deg]

`ROTX0`, `ROTY0` and `ROTZ0` are the euler angles taken in the order `ROTZ0 → ROTY0 → ROTX0`.

- `IST: integer, default:0`: Artificial stiffness option
  - `IST=0`: No artificial stiffness

- `IST=1`: Artificial stiffness is specified

5.7.2.1.4. SIMO Body position, orientation, boundary conditions and artificial stiffness option for CHLOCA_OPT=`POSI'

The input lines in this section are specified for the option `CHLOCA_OPT='POSI'` only. The first input line reads:

`CHBOUND IST`

- `CHBOUND: character(4), default: FREE`: Boundary condition for all nodal DOFs
    - `CHBOUND='FREE'`: All DOFs for the `SIMO` body node are free
    - `CHBOUND='FIXEd'`: All DOFs for the `SIMO` body node are initially fixed
- `IST: integer, default:0`: Artificial stiffness option
    - `IST=0`: No artificial stiffness
    - `IST=1`: Artificial stiffness is specified

The next input line defines the initial configuration of the `SIMO` body:

`XG0 YG0 ZG0 ROTX0 ROTY0 ROTZ0`

- `XG0: real, default: 0`: Initial global X-coordinate of `SIMO` body node $[\mathrm{L}]$
- `YG0: real, default: 0`: Initial global Y-coordinate of `SIMO` body node $[\mathrm{L}]$
- `ZG0: real, default: 0`: Initial global Z-coordinate of `SIMO` body node $[\mathrm{L}]$
- `ROTX0: real, default: 0`: Initial rotation around X-axis $[\mathrm{deg}]$
- `ROTY0: real, default: 0`: Initial rotation around Y-axis $[\mathrm{deg}]$
- `ROTZ0: real, default: 0`: Initial rotation around global Z-axis $[\mathrm{deg}]$

`ROTX0`, `ROTY0` and `ROTZ0` are the euler angles taken in the order `ROTZ0 → ROTY0 → ROTX0`.

If `CHBOUND='FIXEd'`, an additional input line defining the `SIMO` body configuration at final static equilibrium must be included. However, the values must at present be identical to the values specified for the initial configuration: ~ XG YG ZG ROTX ROTY ROTZ ~ - `XG: real, default: XG0`: Global X-coordinate of `SIMO` body node at final static equilibrium $[\mathrm{L}]$ - `YG: real, default: YG0`: Global Y-coordinate of `SIMO` body node at final static equilibrium $[\mathrm{L}]$ - `ZG: real, default: ZG0`: Global Z-coordinate of `SIMO` body node at final static equilibrium $[\mathrm{L}]$ - `ROTX: real, default: ROTX0`: Rotation around X-axis at final static equilibrium $[\mathrm{deg}]$ -

`ROTY: real, default: ROTY0`: Rotation around Y-axis at final static equilibrium $[\mathrm{deg}]$ - `ROTZ: real, default: ROTZ0`: Rotation around global Z-axis at final static equilibrium $[\mathrm{deg}]$

`ROTX`, `ROTY` and `ROTZ` are the euler angles taken in the order `ROTZ → ROTY → ROTX`.

5.7.2.1.5. SIMO Body artificial stiffness

This input line is given if `IST=1` only.

```
STX STY STZ SRX SRY SRZ
```

- `STX: real, default: 0`: Stiffness in global X-direction $[\mathrm{F}/\mathrm{L}]$
- `STY: real, default: 0`: Stiffness in global Y-direction $[\mathrm{F}/\mathrm{L}]$
- `STZ: real, default: 0`: Stiffness in global Z-direction $[\mathrm{F}/\mathrm{L}]$
- `SRX: real, default: 0`: Stiffness around global X-direction $[\mathrm{FL}/\mathrm{deg}]$
- `SRY: real, default: 0`: Stiffness around global Y-direction $[\mathrm{FL}/\mathrm{deg}]$
- `SRZ: real, default: 0`: Stiffness around global Z-direction $[\mathrm{FL}/\mathrm{deg}]$

The artificial stiffness is applied in static analysis only for improving the convergence properties. It does not affect the final static solution.

5.7.3. Termination of input data

Do not forget the `END` input line if this is the `` `last'' data group given in this `INPMOD `` run. See also Termination of input data.

```
END
```

5.8. Additional Features

This group gives a description of special features that normally are not used in analysis of flexible riser system.

5.8.1. Local element axis definition

Additional to Arbitrary system.

This data group may be used to specify a reference vector that is used to determine the initial orientation of the local y- and z-axes of beam elements.

If a local element axis is not specified for an element, the default procedure described in Line, line type and supernode connectivity is used.

This data group must be given for all elements with a CRS6 cross section as the reference vector and the element's x-axis define the net plane during static and dynamic analysis.

5.8.1.1. Data group identifier, one input line

`LOCAl ELEMent AXIS`

5.8.1.2. Number of input lines for special axis definition, one input line

`NAXDEF`

- `NAXDEF: integer`: Number of input lines for special axis definition

5.8.1.3. Specification of reference vector for definition of the local axes in the initial configuration, NAXDEF input lines

`LINE-ID ISEG IEL RNX RNY RNZ`

- `LINE-ID: character(8)`: Line identifier.
- `ISEG: character/integer`: Local segment number within line `LINE-ID`
  - = ``0' or `ALL' means all segments in specified line
- `IEL: character/integer`: Local element number within segment `ISEG`
  - `= `0' or `ALL' means all elements in specified segment `ISEG`
- `RNX: real`: X-component of the reference vector
- `RNY: real`: Y-component of the reference vector
- `RNZ: real`: Z-component of the reference vector

The reference vector is to be given in global system.

The element's local x-axis goes from end 1 to end 2 of the element.

The element's local z-axis is given by the cross product between the element's local x-axis and the reference vector.

The element's local y-axis is given by the cross product of the local z-axis and the local x-axis.

The reference vector must not be parallel with the element's initial x-axis. For CRS6 cable/bar cross sections, the reference vector must be chosen so that it is not parallel to the element's x-axis during the static and dynamic analyses.

For beam elements, the element axes are found at the stress-free configuration and will subsequently follow the element.

5.8.2. Fish net cross section

Additional to [Data Group C: Component Data]

A fish net is subdivided into sections where each section is represented by a cable/bar element with equivalent properties.

Note that the fish net load model requires that the net plane is defined. The net plane is the plane containing the updated local element X-axis and the fixed reference vector specified in the input group [LOCAL ELEMENT AXIS].

5.8.2.1. Data group identifier

```
NEW COMPonent CRS6
```

5.8.2.2. Component type identifier

```
CMPTYP-ID TEMP
```

- `CMPTYP-ID: character(8)`: Component type identifier
- `TEMP: real`: Temperature at which the specification applies. In the present version this item serves only as label

5.8.2.3. Mass and volume

```
AMS AE
```

- `AMS: real`: Mass/unit length $[M/L]$
- `AE: real`: External volume per length $[L^2]$

5.8.2.4. Stiffness properties classification

```
IEA
```

- `IEA: integer`: Axial stiffness code
  - `1`: constant stiffness
  - `N`: table with N paris of tension - elongation to be specified $2 \Leftarrow N \Leftarrow 10$

5.8.2.5. Axial stiffness. Case 1 IEA = 1

`EA`

- `EA: real > 0`: Axial stiffness $[\mathrm{F}]$

5.8.2.6. Axial stiffness. Case 2 IEA = N

`EAF(1) ELONG(1) . . . EAF(N) ELONG(N)`

- `EAF(1): real`: Axial force corresponding to relative elongation `ELONG(1)` $[\mathrm{F}]$
- .
- .
- .
- `ELONG(N): real`: Relative elongation $[1]$

`EAF` and `ELONG` must be given in increasing order.

5.8.2.7. Net properties

`SN WIDTH1 WIDTH2 REDVEL`

- `SN: real`: Solidity ratio $[1]$
- `WIDTH1: real`: Net width at segment end 1 $[\mathrm{L}]$
- `WIDTH2: real`: Net width at segment end 2 $[\mathrm{L}]$
- `REDVEL: real`: Reduction factor for current velocity

5.8.2.8. Damping specification

Identical to input for cross-section type CRS1, see data group Damping specification.

5.8.2.9. Hydrodynamic force coefficients

`AMX AMY`

- `AMX: real`: Added mass per length, tangential direction $[\mathrm{M/L}]$
- `AMY: real`: Added mass per length, normal direction $[\mathrm{M/L}]$

5.8.2.10. Capacity parameter

`TB YCURMX`

- `TB: real`: Tension capacity $[\mathrm{F}]$

- `YCURMX: real`: Maximum curvature $[\mathrm{I/L}]$

These parameters are dummy in the present version.

5.9. Additional Input Files

5.9.1. Specification of internal control system for blade pitch and electrical power

5.9.1.1. Description of internal control system

The implemented control system is based on the choice of a conventional variable-speed, variable blade-pitch-to-feather configuration wind turbine and consists of two basic control systems: a generator torque controller and a full span rotor-collective blade-pitch controller. The two control systems are designed to work independently. The objective of generator-torque controller is to maximize power capture below the rated operation point. The goal of the blade-pitch controller is to regulate generator speed above the rated operation point.

5.9.1.1.1. Control measurement filter

Both the generator torque and blade pitch controllers use the generator speed as the sole feedback input. A recursive, single-pole low-pass filter exponential smoothing to reduce the high frequency excitation of the control systems is provided. The discrete time recursion equation for this filter is

$$\omega_{\mathrm{f.k}} = \alpha\omega_{\mathrm{f.k-1}} + (1-\alpha)\omega_{\mathrm{K}}$$

where

$$\alpha = \exp((-\Delta\mathrm{t})/(\mathrm{TC}))$$

where $\Delta\mathrm{t}$ is the discrete time step, TC is the filter time constant, α is the low-pass filter coefficient ω_{f} is low pass filtered generator speed and k indicates the time step. The relation between the filter time constant and the cut off (corner) frequency $\mathrm{f_C}$ is given by:

$$\mathrm{TC} = \frac{1}{2\pi\mathrm{f_C}}$$

5.9.1.1.2. Generator torque controller

The generator torque is computed as a tabulated function of the filtered generator speed, incorporating five control regions: 1, 1 1/2, 2, 2 1/2 and 3 as illustrated in the figure `Illustration of the variable speed controller - Generator torque versus generator speed' below. Region 1 is a control region before cut-in wind speed, where the generator torque is zero and no power is extracted. Instead, the wind is used to accelerate the rotor for start-up. Region 2 is a control region for optimizing power capture. Here, the generator torque is proportional to the quare of the filtered generator speed to maintain a constant (optimal) tip-speed ratio. In region 3, the generator torque or the generator power is held constant. In case of constant power the generator torque is inversely proportional to the filtered generator speed.

5.9.1.1.3. Blade pitch controller

In region 3, the collective blade pitch angle commands are computed using a gain-scheduled proportional-integral (PI) control on the speed error between the filtered generator speed and the rated generator speed. The PI regulator is represented by the Laplace transform: $(\mathrm{K}(\mathrm{s}+\mathrm{a}))/\mathrm{s}$ where K and a are the proportional gain and the integrator gain. The corresponding and simple regulator algorithm is given by

$$\mathrm{R}(\mathrm{t}+\Delta\mathrm{t})=\mathrm{R}(\mathrm{t})+\Delta\omega\Delta\mathrm{t}$$

$$\theta=\mathrm{K_P}\Delta\omega\mathrm{a}\mathrm{K_P}\mathrm{R}(\mathrm{t}\Delta\mathrm{t})=\mathrm{K_P}\Delta\omega\mathrm{K_I}\mathrm{R}(\mathrm{t}\Delta\mathrm{t})$$

where $\Delta\mathrm{t}$ is the regulator time step, $\Delta\omega$ is the rotor speed error, i.e. the difference between filtered rotor speed and rated rotor speed. R is accumulated time integrated speed error which is set to zero for filtered generator speed less than rated generator speed. θ is the instructed/required collective blade pitch angle.

5.9.1.1.4. Gain scheduling

Gain scheduling is introduced because the optimal proportional and integrator gains are dependent of the blade pitch angle. At each step the gain will be corrected based on the pitch angle applied in the previous step. The user may specify a gain scheduling law or choose to apply the default law presented in the table `The defaults gain scheduling law'. For intervening generator speeds, linear interpolation is used.

Illustration of the variable speed controller - Generator torque versus generator speed

======= The defaults gain scheduling law

Collective Blade Pitch Angle	Correction Factor
$[\mathrm{deg}]$	$[1]$
0.0	1.00
5.0	0.56
10.0	0.39

Collective Blade Pitch Angle	Correction Factor
15.0	0.30
20.0	0.24
90.0	0.05

5.9.1.2. Input description

5.9.1.2.1. Engine Data, Generator, One input line

`GBRATIO GNS_RATE TRQ_RATE RGN3MP`

- `GBRATIO: real >= 1`: Gear box ratio. Number of rotations of the high speed shaft for one rotation of the low speed shaft, i.e. generator versus rotor
- `GNS_RATE: real > 0`: Rated generator speed $[\mathrm{rad/T}]$
- `TRQ_RATE: real > 0`: Rated generator torque $[\mathrm{FL}]$
- `RGN3MP: real`: Minimum pitch angle for which electrical torque versus generator speed will stay in region 3 $[\mathrm{deg}]$

5.9.1.2.2. Engine Data, Generator One input line

`RGN15SP RGN20SP RGN25SP RGN30SP TRQRGN2`

- `RGN15SP: real > 0`: Transitional generator speed between region 1 and 1 1⁄2. Start speed for extracting power. $[\mathrm{rad/T}]$
- `RGN20SP: real > RGN15SP`: Transitional generator speed between region 1 1⁄2 and 2. $[\mathrm{rad/T}]$
- `RGN25SP: real > RGN20SP`: Transitional generator speed between region 2 and 2 1⁄2 $[\mathrm{rad/T}]$
- `RGN30SP: real > RGN25SP`: Transitional generator speed between region 2 1⁄2 and 3 $[\mathrm{rad/T}]$
- `TRQRGN2: real > 0`: Generator torque constant in region 2$[\mathrm{FL}/(\mathrm{rad/T})^2]$

5.9.1.2.3. Engine Data, Generator One input line

- `METRGN3: character(6)`: Method for power extraction in region 3
    - `POWER`: Constant Power
    - `TORQUE`: Constant Torque

5.9.1.2.4. Engine Data, Generator actuator One input line

`TRQ_MAXRAT TRQ_MAX`

- `TRQ_MAXRAT: real > 0`: Maximum torque rate $[\mathrm{FL}/\mathrm{T}]$
- `TRQ_MAX: real > 0`: Maximum electrical torque $[\mathrm{FL}]$

5.9.1.2.5. Blade pitch Controller/actuator One input line

`PC_MINPIT PC_MAXPIT PC_MAXRAT`

- `PC_MINPIT: real`: Minimum pitch setting in pitch controller $[\mathrm{deg}]$
- `PC_MAXPIT: real`: Maximum pitch setting in pitch controller $[\mathrm{deg}]$
- `PC_MAXRAT: real`: Maximum pitch rate $[\mathrm{deg}/\mathrm{T}]$

5.9.1.2.6. Controller Data (PI regulator : K(s+a)/s One input line

`KP KI G_SHEDULE TC`

- `KP: real`: Proportional gain at zero pitch angle
- `KI: real`: Integral gain
- `G_SHEDULE: character`: Gain scheduling; Default or Tabulated
    - `= D`: Default
    - `= T`: Tabulated
- `TC: real`: Time constant for first order low pass filter, $\mathrm{TC} = 1/\omega$ $[\mathrm{s}/\mathrm{rad}]$

Input refer to low-speed shaft

5.9.1.2.7. Gain scheduling (G_SHEDULE=T) One input line

`GSNumber`

-

NOP_GST: integer > 0 : Number of points in gain scheduling table. The maximum is currently 30.

5.9.1.2.8. Gain Scheduling; Gain correction factors. NOP_SGST input lines

```
BPITCH GCF
```

- `BPITCH: real`: Blade pitch angle [deg]
- `GCF: real`: Gain correction factor

5.9.1.2.9. Controller sample interval

```
DTSAMP
```

- `DTSAMP: real > 0`: Controller sample interval [T]

5.9.1.2.10. Example input for control system

```
‘
'gbratio        gnsrate        trq_rate       rgn3mp
  97            122.911        43.09355        1.0
'rgn15sp        rgn20sp        rgn25sp        rgn30sp        trqrgn2
  70.16         91.208         119.0137       121.6805       0.002332288
'metrgn3
 TORQUE
'trq_maxrat     trq_max
   15.0         43.09355
'pc-minpit      pc-maxpit      pc-maxrat
  0.            90.            8.
'kp             ki             g_shedule      TC
0.006275604     0.000896514    D              0.6366
‘
‘dtsamp
0.0125
```

5.9.2. Interface for external wind turbine control system

5.9.2.1. Control system files needed for RIFLEX simulation

The input to RIFLEX for the external wind turbine control system requires the (path and) name of the executable .jar file, the class to be used within that .jar file, and the (path and) name of a file which may contain input data for the external control. See Wind turbine specification. For example, the input may be:

```
'jarName
MyController.jar
'className
no.marintek.wind.control.WindTurbineController
```

```
'config
ControlInput.txt
```

5.9.2.2. Description of control system interface

The external control class must contain at least the three main methods (init, step, and finish) that are called by RIFLEX.

5.9.2.2.1. init

`init` is called once at the beginning of RIFLEX DYNMOD. The controller receives input from RIFLEX about the time step, number of blades, and the name of the control input file.

```
public void init(double dt, int nblades, String filename)
```

5.9.2.2.2. step

`step` is called at the beginning of each time step in RIFLEX DYNMOD. The controller receives measurements (rotor velocity, pitch angles, hub position/velocity/acceleration, instantaneous wind velocity at the hub, etc.) from RIFLEX and returns feedback (torque to be applied on the rotor, pitch angle to be applied on each blade, and - for output purposes - gear shaft rotor speed and generated electrical power).

```
public void step(Measurements measurements, Feedback feedback)
```

`measurements` contains:

- omega: Rotor velocity $[\mathrm{rad}/\mathrm{T}]$. May be extracted with the Java method getOmega().
- pitch angles: Blade pitch angles for all blades $[\mathrm{rad}]$. May be extracted with the Java method getPitchAngle(i), i = 0, …, nblades-1.
- position: (x,y,z,rx,ry,rz) dynamic displacement $[\mathrm{L}]$ and rotation $[\mathrm{rad}]$ from final static position / orientation for the node at the second end of the shaft flex-joint. Values are in the shaft measurement system described below. May be extracted with the Java method getPosition().
- velocity: (vx,vy,vz,vrx,vry,vrz) velocity $[\mathrm{L}/\mathrm{T}]$ and angular velocity $[\mathrm{rad}/\mathrm{T}]$ for the node at the second end of the shaft flex-joint. Values are in the shaft measurement system described below. May be extracted with the Java method getVelocity().
- acceleration: (ax,ay,az,arx,ary,arz) acceleration $[\mathrm{L}/\mathrm{T}^2]$ and angular acceleration $[\mathrm{rad}/\mathrm{T}^2]$ for the node at the second end of the shaft flex-joint. Values are in the shaft measurement system described below. May be extracted with the Java method getAcceleration().
- hub wind: (wix,wiy,wiz) hub wind velocity $[\mathrm{L}/\mathrm{T}]$, global coordinate system. May be extracted with the Java method getHubWindVelocity().

blade root torsional moment around local element x-axis for all blades $[\mathrm{FL}]$. May be extracted with the Java method getBladeRootBMX(i), i = 0, …, nblades-1.

- blade root bending moment around local element y-axis for all blades $[\mathrm{FL}]$. May be extracted with the Java method getBladeRootBMY(i), i = 0, …, nblades-1.
- blade root bending moment around local element z-axis for all blades $[\mathrm{FL}]$. May be extracted with the Java method getBladeRootBMZ(i), i = 0, …, nblades-1.
- aerodynamic torsional moment load around local element x-axis for all blades $[\mathrm{FL}]$. May be extracted with the Java method getBladeRootAeroTor(i), i = 0, …, nblades-1.
- nodal measurements for the `NNOD_MEAS` specified nodes. 18 values for each node: 3 displacements $[\mathrm{L}]$ and 3 rotations $[\mathrm{rad}]$ relative to the stress-free configuration, 6 velocities $[\mathrm{L/T}]$ or $[\mathrm{rad/T}]$ and 6 accelerations $[\mathrm{L/T^2}]$ or $[\mathrm{rad/T^2}]$. The rotations (R1,R2,R3) are taken in the order R3, R2, R1 ; i.e. first rotation around the global z-axis, then around the local (rotated) y-axis and then finally around the local (rotated) x-axis. Global or shaft measurement system as selected by input. May be extracted with the Java method getAddNodeMeas(i) i, i = 0, …, nnod_meas-1. nnod_meas may be extracted with the Java method getNnodMeas().
- element measurements for the `NEL_MEAS` specified elements. 10 values for each element: effective tension $[\mathrm{F}]$, torsional moment $[\mathrm{FL}]$, My-moment end 1 and 2 $[\mathrm{FL}]$, Mz-moment end 1 and 2 $[\mathrm{FL}]$, Qy-shear force end 1 and 2 $[\mathrm{F}]$, Qz-shear force end 1 and 2 $[\mathrm{F}]$. Local element system. May be extracted with the Java method getAddNelMeas(i), i = 0, …, nel_meas-1. nel_meas may be extracted with the Java method getNelMeas().

The shaft measurement system is used for the hub position, velocity and acceleration measurements and may be chosen for the additional nodal measurements. This coordinate system is determined from the stress-free orientation of the shaft element where the electrical torque is applied. - the local z-axis is vertical, - the local x-is the horizontal projection of the element's x axis - the local y-axis is oriented so that the x-, y- and z-axes are a right-handed coordinate system (i.e. the y-axis is the vector cross product of the local z- and x-aces)

If the stress-free orientation of the shaft is in the global X-Z plane, the shaft measurement coordinate system will coincide with the global coordinate system.

`feedback` contains:

- Torque request: Actuator torque to apply on rotor axis $[\mathrm{ML^2/T^2}]$
- Pitch angle request for all blades: Controller pitch angle $[\mathrm{rad}]$
- Gear shaft omega: Gear shaft rotor speed $[\mathrm{rad/T}]$ Used for presentation only.
- Generated power: Generated electrical power $[\mathrm{kW}]$ Used for presentation only.

5.9.2.2.3. finish

`finish` is called at the end of the simulation and primarily serves to close files.

```
public void finish()
```

5.9.3. Airfoil library file

The airfoil library file contains the coefficient data required for aerodynamic calculations. All of the airfoils which are referred to must be included in a single file; i.e. whether or not they are part of a wind turbine. The airfoils may be listed in any order. Airfoils which are not referred to will be ignored.

Comment lines may be included on any line, as in other input files. Blank lines between airfoils will at present be ignored, but it is recommended to use comment lines instead. Blank lines at other locations in the file will be interpreted as input lines and the default values will be used.

For each airfoil, the following data groups must be given.

5.9.3.1. Airfoil identifier

```
CHAIRF
```

- `CHAIRF: character(64)`: Name of the airfoil

5.9.3.2. Airfoil table extension parameters

```
IATEXT ATAIL1 ATAIL2 ANOSE1 ANOSE2 RNOSEC
```

- `IATEXT: integer`: Flag for extending the table to deep stall regime
  - `= 1`: Extend the table
  - `= 0`: Do not extend the table
- `ATAIL1: real`: tail angle between a line perpendicular to the flow and the line from the tip of the wedge, low (negative) angles of attack $[\mathrm{deg}]$
- `ATAIL2: real`: as ATAIL1, but for high (positive) angles of attack $[\mathrm{deg}]$
- `ANOSE1: real`: nose angle between a line perpendicular to the flow and the line from the tip of the wedge, low (negative) angles of attack $[\mathrm{deg}]$
- `ANOSE2: real`: as ANOSE1, but for high (positive) angles of attack $[\mathrm{deg}]$
- `RNOSEC: real`: ratio of the nose radius to the chord of the airfoil $[1]$

5.9.3.3. Airfoil table size parameters

```
NRE NGEO DSIN
```

- `NRE: integer`: Number of Reynolds numbers with coefficient data
- `NGEO: integer, default: 0`: number of points describing the airfoil geometry
- `DSIN: integer, default: 0`: flag for user-defined dynamic stall initialization parameters
  - `= 1`: User-defined dynamic stall initialization parameters will be given below
  - `= 0`: No user-defined dynamic stall initialization parameters will be given

5.9.3.4. Airfoil data

One block per Reynolds number, i.e. NRE data blocks

If only one block is given, the data will be used for all Reynolds numbers. For Reynolds numbers outside the range of Reynolds numbers given, the data for the closest Reynolds number will be used; i.e. flat extrapolation.

5.9.3.4.1. Reynolds number

```
RE NAOA
```

- `RE: real`: Reynolds number for this block $[1]$
- `NAOA: integer`: number of angle-of-attack points for this data block

5.9.3.4.2. Dynamic stall initialization parameters

To be given only if `DSIN`=1

```
AOA0 DCLDA1 DCLDA2 AOAFS1 AOAFS2
```

- `AOA0: real`: the angle of attack where there is zero lift, upcrossing $[\mathrm{deg}]$
- `DCLDA1: real`: maximum slope of the lift curve in the linear region above AOA0 $[1/\mathrm{deg}]$
- `DCLDA2: real`: maximum slope of the lift curve in the linear region below AOA0 $[1/\mathrm{deg}]$
- `AOAFS1: real`: angle of attack corresponding to full separation above AOA0 $[\mathrm{deg}]$
- `AOAFS2: real`: angle of attack corresponding to full separation below AOA0 $[\mathrm{deg}]$

The dynamic stall initialization parameters are shown in the figure `Illustration of dynamic stall initialization parameters' below.

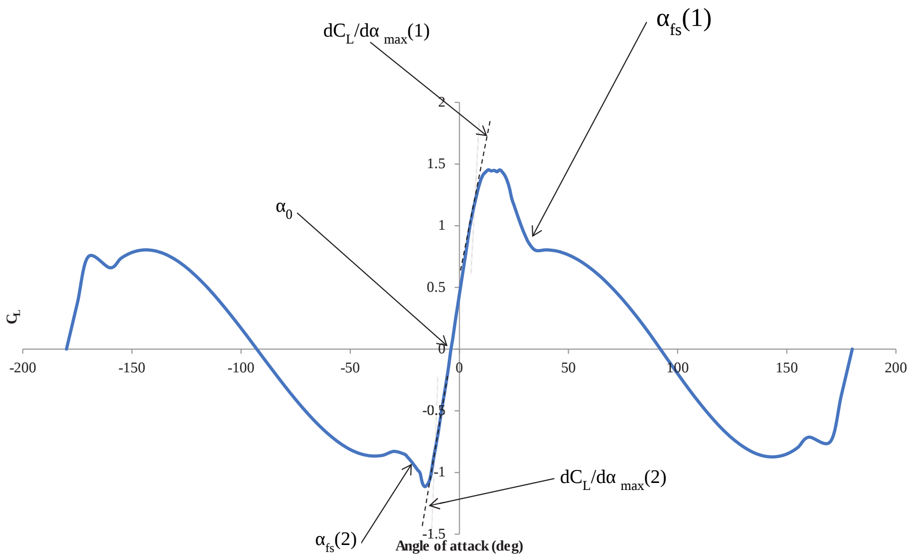

Figure 32. Illustration of dynamic stall initialization parameters

5.9.3.4.3. Aerodynamic coefficients

NAOA input lines

`AOA CL CD CM`

- `AOA: real`: angle of attack [deg]
- `CL: real`: non-dimensional lift coefficient [1]
- `CD: real`: non-dimensional drag coefficient [1]
- `CM: real`: non-dimensional moment coefficient [1]

5.9.3.5. Normalized airfoil geometry

NGEO input lines, to be given only if `NGEO > 0`.
Airfoil coordinates are normalized by the chord length, with the origin at the aerodynamic reference point. Points around the full airfoil should be given, i.e. several lines may have the same `XGEO` and different values for `YGEO`.

`XGEO YGEO`

- `XGEO: real`: x coordinate of geometry point, normalized by chord length.
- `YGEO: real`: y coordinate of geometry point, normalized by chord length.

6. Input to STAMOD

6.1. General Information

The input to the `STAMOD MODULE` is divided into 3 main sections, referred to ass

- Data Group A: Control Information (@ref stamod_a)
- Data Group B: Static Analysis with Fixed Parameters (@ref stamod_b)
- Data Group C: Static Analysis with Parameter Variation (@ref stamod_c)

Theoretical description of the static analysis procedures is given in the chapters `Static Catenary Analysis' and `Static Finite Element Analysis' in the `RIFLEX` Theory Manual. Guidance to static analysis is given in the `Guidance to Static Analysis' section of the `Static Finite Element Analysis' part of the `RIFLEX` Theory Manual.

6.2. Note: Static Analysis with Fixed Parameters and Parameter Variation

In static analysis with fixed parameters, loads types can be activated by the user specified input. For specification of the load types, see data group identfier [LOAD GROUP DATA] (@ref stamod_b_incremental_load_data).

In static analysis with parameter variation, loads types are activated using the data group identfier PARAmeter VARIation DEFInition (@ref stamod_c_parameter_number). The loads that can be activated by user specification are

- Static offset variation
- Current variation
- Specified force variation
- Bottom friction forces
- Global springs
- Material memory function
- Boundary change

In addition, a set of load types are automatically activated using the data group identfier PARAmeter VARIation DeFInition (@ref stamod_c_parameter_number). These loads types are - Volume forces - Tensioner forces - Roller contact forces - Initail stressed elements - Floater forces

Note that if current, wind and specfied force are specified in `STAMOD` but not activated using the data group identfier [LOAD GROUP DATA] (@ref stamod_b_incremental_load_data), current load, wind load and specified force will be applied in `DYNMOD`. Volume forces will be activated i `DYNMOD` if not activated by using the data group identfier `LOAD GROUP DATA`.

6.3. Data Group A: Control Information

This data group is mandatory for all types of analysis with `STAMOD`.

6.3.1. Principal run parameters

6.3.1.1. Data group identifier, one input line

```
STAMod CONTrol INFOrmation CHVERS
```

- `CHVERS: character(8)`: `RIFLEX` input file version, e.g. 3.6

6.3.1.2. Heading, three input lines

Identification of the run alphanumerical text. This text will be output when running `STAMOD`.

```
 HEAD1 HEAD2 HEAD3
```

- `HEAD1: character`: Line 1 of heading text
- `HEAD2: character`: Line 2 of heading text
- `HEAD3: character`: Line 3 of heading text

Always 3 input lines which may all be blank

6.3.1.3. Options and print switches, one input line

```
IRUNCO IDRIS IANAL IPRDAT IPRCAT IPRFEM IPFORM IPRNOR IFILFM IFILCO
```

- `IRUNCO: integer, default: 0`: Run code for data check or executable run
  - `IRUNCO = 0`: Data check run
  - `IRUNCO = 1`: Analysis run
  - `IRUNCO = 2`: Restart analysis. The data group `Specification of restart run' must be given subsequent to this input line
- `IDRIS: character(6)`: Data set identifier corresponding to data for one riser system established by `INPMOD`

`IANAL: integer`: Type of analysis to be performed

  - `IANAL = 1`: Static analysis. Data group B must be provided for this analysis
  - `IANAL = 2`: Static analysis with parameter variation. Data groups B and C must be provided

- `IPRDAT: integer, default: 2`: Print switch for the amount of output from the data generation
  - `IPRDAT=1`: Only identifiers and a few key data are printed
  - `IPRDAT>2`: Tabulated print of system and environmental data (recommended)
  - `IPRDAT=-5`: Print of system presented by segment and print of environment data
- `IPRCAT: integer, default: 1`: Print switch for the amount of output from the catenary analysis
  - `IPRCAT=1`: Gives print of final catenary solution
  - `IPRCAT=5`: Gives in addition print of stress free configuration
  - `IPRCAT=10`: Gives in addition print of the catenary iteration (for debug purposes)
- `IPRFEM: integer, default: 1`: Print switch for the amount of output from `FEM` analysis
  - `IPRFEM=1`: Results are printed for the equilibrium configuration at the end of the final static load step. If the analysis includes parameter variation, results are also printed after the final parameter variation load step.
  - `IPRFEM=5`: Results are printed for equilibrium configurations at the end of each load group
  - `IPRFEM=10`: Results are printed for the equilibrium configurations at every load step
- `IPFORM: integer, default: 1`: Format for print of `FEM` results
  - `IPFORM=-1`: Debug format
  - `IPFORM=1`: Results are presented line by line. Moments and curvatures are given at both element ends
  - `IPFORM=2`: Results are presented line by line. Moments and curvatures are averaged at internal line nodes. Results at element ends are used at line ends (similar to `OUTMO` processing)
  - `IPFORM=3`: Results are presented segment by segment. Moments and curvatures are averaged at internal segment nodes. Results at element ends are used at segment

ends.

- `IPRNOR: integer, default: 1`: Print switch for convergence norms in static `FEM` analysis
  - `IPRNOR=0`: No output of convergence norms
  - `IPRNOR=1`: Print of convergence norms
- `IFILFM: integer, default: 2`: Option for Matrix Plot file
  - `IFILFM=0`: No print
  - `IFILFM` $\neq$ `0`: Matrix Plot file named `<prefix>_stamod.mpf` (`IPRFEM` controls how often static key results are written to the Matrix Plot file)
- `IFILCO: integer, default: 0`: Print switch for storing system configuration to ascii files. Initial configuration used for FE-analysis and configurations after each load-group during the loading sequence are stored. The stored configurations may be used as start configuration for subsequent `STAFEM`-analysis.
  - `IFILCO=0`: No additional files
  - `IFILCO=1`:
    - Configurations stored to ASCii files:
      - `<prefix>_config-lg<i>` where the number `<i>` indicates the load group number.
    - For configurations after parameter variation the files are named:
      - `<prefix>_config-lg<i>-<n>` where `<n>` is the step number

Note that projection angles will only be printed to `<prefix>_stamod.res` for lines that consist of bar elements.

6.3.1.4. Specification of restart run

These input lines are only given for a restart run of `STAMOD` (`IRUNCO=2` in the previous data group, `Options and print switches, one input line'), and should be given subsequent to `STAMOD CONTROL INFORMATION`.

Restart of `STAMOD` makes it possible to continue computation from the last successful load group.

A restart run of `STAMOD` requires two data files from the previous run:

- `IFNDMP` which contains the entire work array

`IFNSTA` which contains the results to be processed by `OUTMOD`

During a run of `STAMOD`, the entire work array (all data in core) will be written to file `IFNDMP` at the end of each completed load group. The file is overwritten each time, so the content is always related to the last (successful) load group. Therefore a restart will normally start with the next load group. In case of restart from parameter variations, the analysis will continue with the next parameter variation step. This makes the parameter variation more flexible as the user can choose to vary one parameter at a time.

It is also possible to carry out several runs with parameter variations from the same static equilibrium configuration. The procedure to be used in this case is illustrated through the following example:

1. Run `STAMOD` in a normal way, (`IANAL=1`). A file prefix of SA_ is inserted for the example.

2. Make a copy of the files SA_IFNDMP.SAM and SA_IFNSTA.FFI to e.g. BACKUP.SAM and BACKUP.FFI, respectively.

3. Run restart with parameter variations, (`IANAL=2`).

4. Now, a new restart can be made from the original static equilibrium configuration by copying BACKUP.SAM and BACKUP.FFI to SA_IFNDMP.SAM and SA_IFNSTA.FFI prior to execution.

The procedure with backup of files can easily be automated in a run command procedure.

Note that current must be present in the original run if restart with parameter variation of current data is specified. Otherwise, the original current will be loaded in the first parameters variation step.

```
RESTart PARAmeters STAMod
```

```
IDSTA
```

- `IDSTA: character(6)`: Identifier for original static analysis results

6.3.2. Data set identifier for present analysis

6.3.2.1. Data group identifier, one input line

```
RUN IDENtification
```

6.3.2.2. Data set identifier for results, one input line

```
IDRES
```

- `IDRES: character(6)`: Data set identifier for this run

6.3.3. Identifier of environment data

6.3.3.1. Data group identifier, one input line

```
ENVIronment REFErence IDENtifier
```

6.3.3.2. Identifier of environment data, one input line

```
IDENV
```

- `IDENV: character(6)`: Identifier of environment data given as input to the `INPMOD` module

This data group is dummy for coupled analysis.

6.3.4. Export of element responses, one input line

This specification is optional. Specifying export of element responses enables visualization of the incremental loading configurations by use of the computer program `SIMVIS` subsequent to static analysis. In present version it is possible to specify element responses in form of effective tension, resulting curvature and longitudinal stress (if available).

```
STORe VISUalisation RESPonses
```

6.3.4.1. Detailed specification of exported element responses

This data group is optional.

By default effective tension for all lines will be exported. This input line makes it possible to limit or specify response types for selected lines in the system.

Number of input lines: as many as necessary.

```
OPTION CHRESP CHILIN
```

- `OPTION: character:`
  - `= STORE`
  - `= NOSTORE`
- `CHRESP: character`: Response type to be exported
  - `= EFF-AX-FORCE`: Effective tension
  - `= RES-CURV`: Resultant curvature

- `= LONG-STRESS`: Longitudinal stress
- `= ALL`: All of the above described responses

- `CHILIN: character`:
  - `= LINE-ID`: Reference to line identifier
  - `= ALL`: All lines

6.4. Data Group B: Static Analysis with Fixed Parameters

This section is mandatory, and the analysis must always be carried out before any type of dynamic analysis is started.

6.4.1. Definition of subsequent input

The below example specifies one nodal load, applies the 1st current profile and tells that wind is not used. Subsequent sections give further details.

6.4.1.1. Data group identifier

```
STATic CONDition INPUt
```

6.4.1.2. External, static loads

```
NLCOMP ICURIN CURFAC IWINDIN
```

- `NLCOMP: integer, default: 0`: Number of additional load components. Loads to be specified in next data group, `Additional, static load components', which is omitted if `NLCOMP=0`.
- `ICURIN: integer, default: 0`: Current indicator
  - `ICURIN = 0`: No current
  - `ICURIN = N`: Current profile no. N on referenced environmental description (`IDENV`) is used. The profile may be scaled by `CURFAC`
- `CURFAC: integer, real: 1`: Scaling factor to amplify or reduce the referred current profile $[1]$
- `IWINDIN: integer, default: 0`: Wind indicator
  - `IWINDIN = 0`: No wind
  -

- `IWINDIN = N`: Wind specification no. N on referenced environmental description (`IDENV`) is used.

If current loads are applied in `STAMOD`, the current active at the end of the static analysis will be used in `DYNMOD`. If current loads are not applied in `STAMOD`, the specified current profile `ICURIN` will be used in `DYNMOD` with the scaling factor `CURFAC` specified here.

`CURFAC` must be 1.0 for a restart analysis.

6.4.1.3. Additional, static load components

This data specification is omitted if `NLCOMP=0`. Otherwise `NLCOMP` input lines are specified.

`LINE-ID ILSEG ILNODE ILDOF RLMAG CHICOO`

- `LINE-ID: character(8)`: Reference to line identifier.
- `ILSEG: integer`: Segment number within the actual line.
- `ILNODE: integer`: Local node (`CHICOO=GLOBAL`) or element number (`CHICOO=LOCAL`).
- `ILDOF: integer`: Degree of freedom within the specified node/element.
  - `ILDOF = 7….12` at end 2 of an element.
- `RLMAG: real, default: 0`: Magnitude of load component (F or FL).
- `CHICOO: character(6), default: GLOBAL`: Reference system for application of nodal load components.
  - `CHICOO = GLOBAL`: Force component refers to global system. The force is applied at the specified node.
  - `CHICOO = LOCAL`: Force component refers to local system. The force is applied to the specified element.
  - If skew boundaries or vessel boundary are specified at the node `CHICOO=GLOBAL` means that the load component acts in the skew (vessel) system.

6.4.1.4. Load formulation and matrix format

`LCONS ISOLVR`

- `LCONS: integer, default: 0`: Switch for lumped or consistent load and mass formulation.
  - `LCONS = 0`: Lumped load and mass formulation.

`LCONS = 1`: Consistent load and mass formulation.

  - Applies also to DYNMOD

- `ISOLVR: integer, default: 1`: Matrix storage format. Applies also to `DYNMOD`.
  - `ISOLVR = 1`: Skyline, recommended for simple, single line riser models.
  - `ISOLVR = 2`: Sparse, recommended for coupled analysis and complex models with branch points.
  - The free vibration option in `DYNMOD` requires `ISOLVR = 1`

When applying the lumped load and mass formulation (`LCONS=0`), the element nodes are assigned part of the distributed external element load and mass directly. When using a consistent formulation (`LCONS=1`), the external element load and mass is distributed to the nodes using the same interpolation (shape) functions as applied when determining the element stiffness matrix. See the theory manual for more details.

The numerical solution speed for static and subsequent dynamic analysis depends on the matrix storage format. For a typical single riser analysis where the resulting stiffness matrix is rather narrow banded, the skyline matrix storage method is the most efficient, `ISOLVR=1`.

In case of many branch points, or when doing coupled analysis with many mooring lines and/or risers connected to the same floating vessel, the stiffness matrix may be significantly more broad banded. In such a case, choosing sparse matrix storage may increase the numerical solution speed, `ISOLVR=2`.

Note: Since the difference in solution speed could be practically negligible for small models, sparse matrix storage (`ISOLVR=2`) may be a good initial choice. Later, one could check if skyline matrix storage gives better numerical performance.

6.4.2. Computational procedure selection

6.4.2.1. Data group identifier, one input line

```
COMPutational PROCedure
```

6.4.2.2. Method for static equilibrium computation

```
AMETH
```

- `AMETH: character(6)`: Code for computation method. The following options are available:
  - `CAT`: Catenary analysis, bending stiffness neglected

`CATFEM`: Finite element method based on catenary start configuration

- `STAFEM`: Finite element method based on start configuration read from file.
- `FEM`: Finite element method based on stress free start configuration

Selection of options according to system type and whether a subsequent dynamic analysis is to be carried out or not:

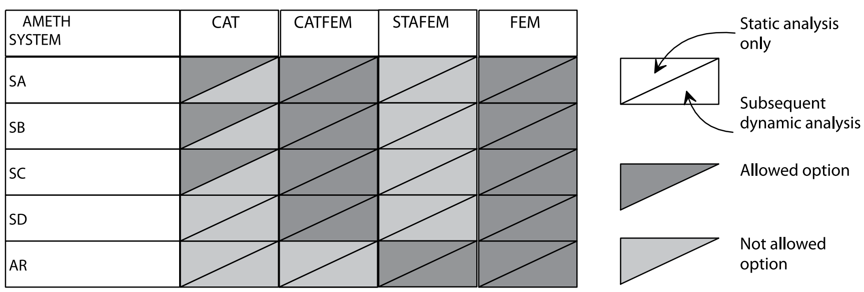

Figure 33. Available options for static and dynamic analysis

6.4.3. Catenary analysis procedure, CAT

6.4.3.1. Data group identifier

```
CATEnary ANALysis PARAmeters
```

6.4.3.2. Parameters for catenary analysis

```
XL50 FL10 XU1TOL XU3TOL
```

- `XL50: real, default: see below`: Initial estimate of angle from vertical at the point where the catenary calculation starts $[\mathrm{deg}]$
- `FL10: real, default: see below`: Initial estimate of axial force at the point where the catenary calculation starts $[\mathrm{F}]$
- `XU1TOL: real, default: see below`: Tolerance of X1 coordinate at upper end $[\mathrm{L}]$
- `XU3TOL: real, default: see below`: Tolerance of X3 coordinate at upper end $[\mathrm{L}]$

Default values will be computed for the standard riser system based on geometry and specified weights and forces, if the parameter is given a ``slash"(/). For standard system SA, SB and SD the calculation of the catenary solution starts at the upper end, for SC it starts at the lower end.

For `XL50` and `FL10` default values are recommended. The values are printed out on the result file. For the SC system these parameters are dummy.

The default values of `XU1TOL` and `XU3TOL` are calculated as 10^{-4} (length of the riser).

6.4.4. Catenary and subsequent finite element analysis, CATFEM

6.4.4.1. Data group identifier

```
CATFem ANALysis PARAmeters
```

6.4.4.2. Parameters for catenary equilibrium calculation

```
XL50 FL10 XU1TOL XU3TOL
```

- `XL50: real, default: see below`: Initial estimate of angle from vertical at the point where the catenary calculation starts $[\mathrm{deg}]$
- `FL10: real, default: see below`: Initial estimate of axial force at the point where the catenary calculation starts $[\mathrm{F}]$
- `XU1TOL: real, default: see below`: Tolerance of X1 coordinate at upper end $[\mathrm{L}]$
- `XU3TOL: real, default: see below`: Tolerance of X3 coordinate at upper end $[\mathrm{L}]$

Default values will be computed for the standard riser system based on geometry and specified weights and forces, if the parameter is given a ``slash"(/).

For standard system SA, SB and SD the calculation of the catenary solution starts at the upper end, for SC it starts at the lower end.

For `XL50` and `FL10` default values are recommended. The values are printed out on the result file. For the SC system these parameters are dummy.

The default values of `XU1TOL` and `XU3TOL` are calculated as 10^{-4} (length of the main riser line).

Next data group is Incremental loading procedure.

6.4.5. Finite element analysis from start configuration, STAFEM

Preliminary test version.

In present version it is assumed that the start solution represent a catenary solution.

6.4.5.1. Data group identifier

```
STAFem ANALysis PARAmeters
```

6.4.5.2. Name of file containing the start solution

```
CHFSTA KFORM
```

- `CHFSTA: character(60)`: File name with specification of start configuration . The content of this file is described in Define Start Configuration.

- `KFORM: integer, default: 1`: Code for file format

  - `KFORM = 1`: Co-ordinates and base vector system given for all FEM-nodes in increasing node number order.

Next data group is Incremental loading procedure.

6.4.6. Finite element analysis, FEM

6.4.6.1. Data group identifier

```
FEM ANALysis PARAmeters
```

6.4.7. Incremental loading procedure

This data group describes the incremental loading procedure from catenary solution (`CATFEM`), from a specified start solution (`STAFEM`) or from stress free configuration (`FEM`) to the final static equilibrium configuration.

A brief summary of the incremental loading procedure applied, is given in the following. For a more detailed description including analysis guidance, see `Static Finite Element Analysis' in the RIFLEX Theory Manual.

Based on load groups, the user is free to specify an arbitrary load sequence. Incrementation and iteration parameters are specified separately for each load group. One or several load types can be applied within each load group. Simultaneous application of several load types and user-defined order of the load application is therefore possible.

The incremental loading is normally carried out in the following sequence:

- Load group 1: Volume forces (weight and buoyancy)

- Load group 2: Specified displacements (i.e. displacements to final position of nodal points with specified boundary conditions)

- Load group 3: Specified forces (nodal point loads)

- Load group 4: Position dependent forces (current forces)

All user-defined load types have to be specified within a load group in order to be applied during the incremental loading of the system. Examples are roller and tensioner contact forces (elastic contact surface), initially stressed segments or floater forces.

The user may specify the load group for application of bottom friction and global linear springs. It is also possible to neglect friction and springs in normal static analysis and activate friction during static parameter variation or at the start of dynamic analysis.

6.4.7.1. CATFEM analysis

Volume forces have to be applied within one incremental step in the first specified load group. This is because volume forces and prescribed translations from stress free to final positions of terminal points are included in the catenary start solution. Deviations between the catenary and the final `FEM` solution are, however, present due to different mathematical formulations and neglection of bending stiffness in the catenary analysis. The first load group applying volume forces in one incremental step, is therefore a simple equilibrium iteration starting from the catenary solution with weight and buoyancy forces acting. The equilibrium iteration may fail if there are significant differences between the catenary solution and the final solution due to bending stiffness. It is therefore possible to apply the bending stiffness in several incremental steps to reach the final solution.

The iterative approach on boundary conditions used in the catenary analysis will give deviations between specified translating boundary conditions (i.e. x and z co-ordinates) and boundary condition computed by the catenary analysis. Further, specified boundary conditions for rotations at the supports will not be satisfied by the catenary analysis due to neglection of bending stiffness. A load group for prescribed displacements should therefore be included to account for inaccuracies in boundary conditions from the catenary analysis.

6.4.7.2. STAFEM analysis

The load types applied for the start configuration have to be indicated in the first specified load group. These load types will act with full force while the residual forces will be off-loaded the specified number of load steps. The residual forces are the unbalanced forces based on the indicated load types and the internal forces that appear when the start configuration is described in the finite element formulation.

Note that it is assumed that final boundary conditions are included in the start configuration. As a consequence the procedure will be indifferent to specification of prescribed displacements.

6.4.7.3. Load group specification

The input lines `Data group identifier…', `Load group incrementation…' and `Load types to be activated…' have to be given in one block for each load group.

6.4.7.3.1. Data group identifier, one input line

```
LOAD GROUP DATA
```

6.4.7.3.2. Load group incrementation and iteration parameters, one input line

`NSTEP MAXIT RACU CHNORM EACU`

- `NSTEP: integer`: Number of load steps
- `MAXIT: integer, default: 10`: Maximum number of iterations during application of load
- `RACU: real, default:` 10^{-6}: Required accuracy measured by displacement norm $[1]$
- `CHNORM: character(4), default: DISP`: Convergence norm switch - `= DISP`: Use the default Euclidean displacement norm only - `= BOTH`: Use both the default Euclidean displacement norm and the energy norm
- `EACU: real, default:` 10^{-6}: Required accuracy measured by energy norm - Dummy if `CHNORM=DISP`

6.4.7.3.3. Load types to be activated, one line for each load type to be activated within the load group

`LOTYPE ISPEC`

- `LOTYPE: character(4)`: Load type to be applied
  - `= VOLU`: Volume forces
  - `= DISP`: Specified displacements
  - `= SFOR`: Specified forces
  - `= CURR`: Current forces
  - `= TENS`: Activate tensioner contact forces
  - `= ROLL`: Activate roller and tubular contact forces
  - `= PIPE`: Activate pipe-in-pipe contact forces
  - `= ISTR`: Initially pre-stressed segments
  - `= FLOA`: Floater forces
  - `= FRIC`: Activate bottom friction forces
  - `= SPRI`: Activate global springs
  - `= BEND`: Bending stiffness
  - `= TEMP`: Temperature variation
  - `= PRES`: Pressure variation

    - `= MEMO`: Activate material memory formulation (Isotropic/kinematic hardening)
    - `= BOUN`: Activate boundary change
    - `= WINC`: Run winch(es)
    - `= GROW`: Apply cross-section changes from growth profile
    - `= WIND`: Wind forces
- `ISPEC: integer`: Parameter used for further description of applied load type:
    - `LOTYPE = PIPE`:
        - `ISPEC = 0` (default): Possible pipe-in-pipe contact enabled (ENTERED)
        - `= 1`: Specify start condition for pipe-in-pipe contact
    - `LOTYPE = TEMP`:
        - `ISPEC = NLSPEC`: Number of subsequent input lines for specification of temperature variation.
    - `LOTYPE = PRES`;
        - `ISPEC = Nxxx`: Number of subsequent input lines for specification of pressure variation
    - `LOTYPE = BOUN`:
        - `ISPEC = NBOUND`: Number of nodes with change in boundary conditions.
    - `LOTYPE = WINC`:
        - `ISPEC = Nxxx`: Number of subsequent input lines for specification of winch run
    - `LOTYPE = GROW`:
        - `ISPEC = 0` (default): No scaling of growth profile.
        - `ISPEC = 1`: Number of subsequent input lines for specification of growth scaling.
    - `LOTYPE = WIND`:
        - `ISPEC = 0` (default): No wind on turbine blades
        -

ISPEC = 1: Number of subsequent input lines for specification of wind on turbine blades.

- ISPEC is dummy for other load types

Note that some static loads will be incremented over NSTEP load steps while others will be activated at the beginning of the load group. For example volume forces and current forces are incremented over the specified load steps while element memory and contact forces are activated at the beginning of the load group in which they are specified.

Volume forces have to be applied in the first specified load group in case of CATFEM or STAFEM analysis. For CATFEM analysis the number of load steps in this load group has to be one (NSTEP = 1).

6.4.7.3.4. Pipe-in-pipe contact; One input line given if LOTYPE=PIPE and ISPEC = 1

CHPCNT

- CHPCNT: character:
  - = ENTERED
  - = NOT ENTERED

Note that ENTERED should be used for analysis of slender structures such as risers, cables and umbilicals. NOT ENTERED is intended to be used for marine operations. The master and slave pipe in static free condition should be modelled as close to the final configuration as possible, see illustraion in the figures `Example: pipe-in-pipe modelled as `ENTERED’ and `Example: pipe-in-pipe modelled as `NOT ENTERED’ below.

Example: pipe-in-pipe modelled as ENTERED

Example: pipe-in-pipe modelled as NOT ENTERED

6.4.7.3.5. Temperature variation, NLSPEC input lines only given if LOTYPE = TEMP

LINE-ID ISEG IEC TEMP

- LINE-ID: character(8): Reference to line identifier
- ISEG: integer/character: Segment number
  - = 0 / ``ALL'': All segments in specified line
- IEL: integer/character: Element number
  - = 0 / ``ALL'': All elements in specified line

`Temp: real`: Temperature at end of temperature variation

The temperature is varied linearly during the load group from the starting temperature given in the cross-sectional data in `INPMOD` and ending with the temperature specified here.

A linear variation of temperature over a sequence of elements may be specified by giving a negative element number at the second end of the linear variation

6.4.7.3.6. Pressure variation, NLSPEC input lines if LOTYPE = PRES

`MRL-ID PRESSI DPRESS VVELI`

- `MRL-ID: character(8)`: Reference to Main Riser Line identifier
- `PRESSI: real, default: 0`: Final pressure at inlet end $[\mathrm{F}/\mathrm{L}^2]$
- `DPRESS: real, default: 0`: Final pressure drop $[\mathrm{F}/\mathrm{L}^3]$
- `VVELI: real, default: 0`: Final fluid velocity $[\mathrm{L}^3/\mathrm{T}]$
  - Dummy in present version

6.4.7.3.7. Boundary changes, 2 x NBOUND input lines if LOTYPE = BOUN

Identification of node for boundary change ~ IREF-ID ILSEG ILNODE IOP ~

- `IREF-ID: character(8)`: Reference to line or supernode identifier.
- `ILSEG: integer`:
  - If `IREF-ID` refers to a line, `ILSEG` is the segment number within this line
  - If `IREF-ID` refers to a supernode, `ILSEG` must be zero
- `ILNODE: integer`:
  - If `IREF-ID` refers to a line, `ILNODE` is the node number within segment `ILSEG`
  - If `IREF-ID` refers to a supernode, `ILNODE` must be zero
- `IOP: integer`: Parameter for boundary change option
  - `= 0`: Boundary conditions: fixed, pre-scribed or free
  - `= -1`: Boundary conditions: rigid node connection (The node will become a slave node.)

Ordinary (line end) supernodes and SIMO body nodes with `CHLOCA_OPT='POSI'` may have boundary change.

Status for nodal degrees of freedom. To be given if `IOP = 0` ~ IPOS IX IY IZ IRX IRY IRZ ~

- `IPOS: integer`: Boundary condition type
  - `IPOS = 0`: The node is fixed in global system
  - `IPOS = N`: The node is attached to support vessel no N
- `IX: integer`: Boundary condition code for translation in X-direction
  - `IX = 0`: Free
  - `IX = 1`: Fixed of prescribed
- `IY: integer`: Boundary condition code for translation in Y-direction
  - Same interpretation as for `IX`.
- `IZ: integer`: Boundary condition code for translation in Z-direction
  - Same interpretation as for `IX`.
- `IRX: integer`: Boundary condition code for rotation around X-direction
  - Same interpretation as for `IX`.
- `IRY: integer`: Boundary condition code for rotation around Y-direction
  - Same interpretation as for `IX`.
- `IRZ: integer`: Boundary condition code for rotation around Z-direction
  - Same interpretation as for `IX`.

Identification of master node. To be given if `IOP = -1` ~ LINE-ID ILSEG ILNODE ~

- `LINE-ID: character(8)`: Reference to line identifier
- `ILSEG: integer`: Segment number within the actual line
- `ILNODE: integer`: Local node number within segment

6.4.7.3.8. Winch run, NLSPEC input lines only given if LOTYPE = WINC

```
IWINCH WILNG
```

- `IWINCH: integer`: Line number
- `WILNG: real`: Total run length $[L]$

`WILNG > 0`: winching out, i.e. the winch run will increase the active length.

6.4.7.3.9. Growth profile, ISPEC=1 input lines only given if LOTYPE = GROW

```
GFAC
```

- `GFAC: real, default: 1.0`: Scaling of growth profile

6.4.7.3.10. Wind force, ISPEC=1 input lines only given if LOTYPE = WIND

```
WindOnTurbineBlades
```

- `WindOnTurbineBlades: character(8), default: OFF`: Code for wid loads on turbine blades
  - `WindOnTurbineBlades = OFF`: No wind loads on turbine blades in static analysis.
  - `WindOnTurbineBlades = ON`: Wind loads on turbine blades in static analysis.

Note that wind loads will be applied on turbine blades and the rest of the structure in dynamic analysis.

6.4.8. Define stressfree configuration

This data group is optionally available for AR systems. It enables the user to define an arbitrary stressfree configuration without having to establish a complex line/supernode system model. The option is useful for effective modelling of pre-bent sections.

This data group will redefine the stressfree configuration, but will not affect the coordinates for static equilibrium position given as input to `INPMOD`, data group Specification of boundary conditions, stressfree configuration and static equilibrium configuration.

6.4.8.1. Data group identifier

```
DEFIne STREssfree CONFiguration
```

6.4.8.2. File name

```
CHFCON
```

- `CHFCON: character(80)`: File name with definition of stress free configuration

6.4.8.3. File format

`KFORM`

- `KFORM: integer, default: 1`: Code for file format
  - `KFORM = 1`: Stress free co-ordinates given for all `FEM`-nodes in increasing node number order

6.4.9. Bottom geometry file

This data group is given if `IBOT 3D = 1` in the `INPMOD` input file, and allows the user to define an uneven seabed, using depth data in a regular grid with equidistant spacing.

6.4.9.1. Data group identifier

`BOTTom GEOMetry FILE`

6.4.9.2. File name

`CHFBOT`

- `CHFBOT: character(80)`: File name with seabed geometry data

The content of this file is described in Define uneven seabed geometry.

6.4.9.3. Coordinates of the seabed file reference system

`XOS YOS ZOS ANGOS`

- `XOS: real, default: 0`: x-coordinate of the origin of the seabed file reference system, in the global reference system $[\mathrm{L}]$
- `YOS: real, default: 0`: y-coordinate of the origin of the seabed file reference system, in the global reference system $[\mathrm{L}]$
- `ZOS: real, default: 0`: z-coordinate of the origin of the seabed file reference system, in the global reference system $[\mathrm{L}]$
- `ANGOS: real, default: 0`: Angle between the x-axis of the seabed file reference system and the x-axis of the global reference system $[\mathrm{deg}]$

6.5. Data Group C: Static Analysis with Parameter Variation

6.5.1. Parameter variation definition

6.5.1.1. Data group identifier, one input line

`PARAmeter VARIation DEFInition`

6.5.1.2. Number of variations and variation codes

`NSTVAR IOFPOS ICUVAR IFOVAR MAXIPV RACUPV CHNORM EACUPV`

- `NSTVAR: integer`: Number of steps in parameter variations
- `IOFPOS: integer, default: 0`: Code for static offset variation
  - `IOFPOS = 0`: The parameter is not varied
  - `IOFPOS = 1`: The parameter is varied in `NSTVAR` steps according to subsequent specification
- `ICUVAR: integer, default: 0`: Code for current variation
  - `ICUVAR = 0`: The parameter is not varied
  - `ICUVAR = 1`: The parameter is varied in `NSTVAR` steps according to subsequent specification
- `IFOVAR: integer, default: 0`: Code for specified force variation
  - `IFOVAR = 0`: The parameter is not varied
  - `IFOVAR = 1`: The parameter is varied in `NSTVAR` steps according to subsequent specification
- `MAXIPV: integer, default: 1`: Maximum number of iterations for each variation
- `RACUPV: real, default:`10^{-5}: Required accuracy measured by displacement norm
- `CHNORM: character(4), default: DISP`: Convergence norm switch `- = DISP`: Use the default Euclidean displacement norm only `- = BOTH`: Use both the default Euclidean displacement norm and the energy norm
- `EACUPV: real, default:` 10^{-5}: Required accuracy measured by energy norm - Dummy if `CHNORM=DISP`

The total number of load steps in parameter variation is `NSTVAR`.

All parameter for which variations are specified, are varied simultaneously

Information about parameter values are to be specified in the subsequent data groups. The initial configuration as specified according to data section B is automatically taken as the first case.

`ICURIN` must be greater than zero and `CURFAC` must be 1.0 (See External, static loads).

See also Note: Static Analysis with Fixed Parameters and Parameter Variation.

6.5.1.3. Load types to be activated. One line for each load type. Optional input, maximum 4 specifications.

`LOTYPE ISPEC`

- `LOTYPE: character(4)`: Load type to be applied
  - `= FRIC`: Activate bottom friction forces
  - `= SPRI`: Activate global springs
  - `= MEMO`: Activate material memory formulation (Isotropic/kinematic hardening)
  - `= BOUN`: Boundary change
- `ISPEC: integer`: Parameter used for further description of applied load type:
  - `LOTYPE = BOUN`:
    - `ISPEC = NBOUND`; Number of nodes with change in boundary conditions.
  - `ISPEC` is dummy for other load types

If specified load type is activated before, the input given here is disregarded.

Activation of sea floor friction is given by `FRIC`.

6.5.1.3.1. Specification of boundary change, 2 x NBOUND lines

Identification of node for boundary change ~ IREF-ID ILSEG ILNODE IOP ~

- `IREF-ID: character(8)`: Reference to line or supernode identifier.
- `ILSEG: integer`:
  - If `IREF-ID` refers to a line, `ILSEG` is the segment number within this line
  - If `IREF-ID` refers to a supernode, `ILSEG` must be zero
- `ILNODE: integer`:
  - If `IREF-ID` refers to a line, `ILNODE` is the node number within segment `ILSEG`
  - If `IREF-ID` refers to a supernode, `ILNODE` must be zero

`IOP: integer`: Parameter for boundary change option

- `= 0`: Boundary conditions: fixed, pre-scribed or free
- `= -1`: Boundary conditions: rigid node connection (The node will become a slave node.)

Ordinary (line end) supernodes and SIMO body nodes with `CHLOCA_OPT='POSI'` may have boundary change.

Status for nodal degrees of freedom. To be given if `IOP = 0` ~ IPOS IX IY IZ IRX IRY IRZ ~

- `IPOS: integer`: Boundary condition type
  - `IPOS = 0`: The node is fixed in global system
  - `IPOS = N`: The node is attached to support vessel no N
- `IX: integer`: Boundary condition code for translation in X-direction
  - `IX = 0`: Free
  - `IX = 1`: Fixed of prescribed
- `IY: integer`: Boundary condition code for translation in Y-direction
  - Same interpretation as for `IX`.
- `IZ: integer`: Boundary condition code for translation in Z-direction
  - Same interpretation as for `IX`.
- `IRX: integer`: Boundary condition code for rotation around X-direction
  - Same interpretation as for `IX`.
- `IRY: integer`: Boundary condition code for rotation around Y-direction
  - Same interpretation as for `IX`.
- `IRZ: integer`: Boundary condition code for rotation around Z-direction
  - Same interpretation as for `IX`.

Identification of master node. To be given if `IOP = -1` ~ LINE-ID ILSEG ILNODE ~

- `LINE-ID: character(8)`: Reference to line ID
- `ILSEG: integer`: Segment number within the actual line

- `ILNODE: integer`: Local node number within segment

6.5.1.4. Variation of static positions

This data group is relevant only if `IOFPOS=1`.

6.5.1.4.1. Data group identifier

```
STATic OFFSet INCRements
```

6.5.1.4.2. Static position increments

```
CHIREF DXOFF DYOFF DZOFF IROT DROT
```

- `CHIREF: character(8)`: Reference to moving point
  - `CHIREF = -IVES`: Support vessel number `IVES`
  - `CHIREF = SNOD_ID`: Supernode identifier. `SNOD_ID` must refer to a supernode with specified position
- `DXOFF: real, default: 0`: Displacement increment, X-direction $[\mathrm{L}]$
- `DYOFF: real, default: 0`: Displacement increment, Y-direction $[\mathrm{L}]$
- `DZOFF: real, default: 0`: Displacement increment, Z-direction $[\mathrm{L}]$
- `IROT: integer, default: 0`: Rotation code
  - `IROT = 0`: No rotation
  - `IROT = 1`: The given rotation is taken about X-axis
  - `IROT = 2`: The given rotation is taken about Y-axis
  - `IROT = 3`: The given rotation is taken about Z-axis
- `DROT: real, default: 0`: Rotation increment $[\mathrm{deg}]$
  - Dummy if `IROT=0`

If a support vessel is specified (`IREF=-IVES`) the displacement increments refer to the support vessel coordinate system. Otherwise the increments refer to the global coordinate system.

Only rotation around the Z-axis may be given if `IREF = -IVES`

If the supernode has boundary conditions specified in a skew co-ordinate system or in the vessel system, displacements and the rotation increment `DROT` take place in the skew / vessel system. It should also be noted that the orientation of such a skew co-ordinate system is kept constant during the static position incrementation (relevant for AR systems only).

6.5.2. Variation of current velocity and direction

This data group is relevant only if `ICUVAR=1`.

6.5.2.1. Data group identifier, one input line

```
CURRent VARIation INCRements
```

6.5.2.2. Current velocity and direction increments

```
DCUVEL DCUDIR
```

- `DCUVEL: real, default: 0`: Current velocity increment $[\mathrm{L/T}]$
- `DCUDIR: real, default: 0`: Current direction increment $[\mathrm{deg}]$

In the case of multilayer current specification, these increments are interpreted as follows.

- DCUVEL: This applies directly to the uppermost layer. The lower layers are incremented in the same proportion, so that the shape of the current profile is maintained.
- DCUDIR: This applies to all current layers, so that the whole current profile is rotated the same amount.

6.5.3. Variation of specified forces

This data group is relevant only if `IFOVAR=1` and `NLCOMP >= 1` (See Definition of subsequent input).

6.5.3.1. Data group identifier, one input line

```
SPECified FORCe INCRements
```

6.5.3.2. Force increments, NLCOMP input lines to be given

```
DRLMAG
```

- `DRLMAG: real, default: 0`: Force increment on specified forces `RLMAG` (F or FL).
  - See Additional, static load components.

6.5.4. Control parameters for printing of results

6.5.4.1. Data group identifier, one input line

```
STAMod PRINt CONTrol
```

6.5.4.2. Control parameters for print of results from static parameter variation analysis

```
ISTEP ISFOR ISPOS
```

- `ISTEP: integer, default: 1`: Step interval for print of specified parameters.
  - Print is given for the following steps: 1, 1+`ISTEP`, 1+2 × `ISTEP`, 1+3 × `ISTEP`, …, `NSTVAR`
  - `ISTEP = 1`: Gives print of results for all variation steps
- `ISFOR: integer, default: 1`: Parameter for print of forces
  - `ISFOR = 0`: No print of external forces
  - `ISFOR = 1`: Print of external force components in global x, y, z direction at all supernodes with the following status codes: `TSNFX, TSNFX2, TSNPOSI1`
  - Not yet implemented
- `ISPOS: integer, default: 1`: Parameter for print of position
  - `ISPOS = 0`: No print of positions
  - `ISPOS = 1`: Print of x, y and z coordinates for all free supernodes (i.e. status code `TSNBRA` and `TSNFRE`)

6.6. Static Analysis with Updated Drag Forces

An important consequence of VIV response is increased in-line current forces. One of the key results from the `VIVANA` program is therefore drag amplification factors along the structure. The objective of this input is to enable static and dynamic analysis using the updated drag forces.

To use this option the `RIFLEX` program modules must be run in the following order: - `INPMOD` - `STAMOD` - `VIVANA` - `STAMOD` with the drag amplification data group specified

Note that drag amplication cannot be used in combination with marine growth profile specification (@ref inpmod_c_growth).

6.6.1. Data group identifier, one input line

```
DRAG AMPLIFICATION INPUT
```

6.6.2. Specification of file for input of drag amplification, one input line

```
CHFDRG CHIOP
```

- `CHFDRG: character(60)`: File with drag amplification coefficient; e.g case22_vivana.mpf
- `CHIOP: character(60)`: Format of file with drag amplification coefficients

The `CHFDRG` file may be generated by running `VIVANA`.

The only file type currently available is the MatrixPlot file format. Thus; `CHIOP = MPF`.

6.7. Description of Additional Input Files

6.7.1. Define Stressfree Configuration

The file ``CHFCON" specified in Data Group B: Static Analysis with Fixed Parameters, contains definition of stressfree configuration. The file is a free format sequential ASCII-file.

The file description depends on the parameter ``KFORM" specified in Finite element analysis from start configuration, STAFEM.

File description: `KFORM = 1`.

6.7.1.1. Number of nodes, one input line

```
NFSNOD
```

- `NFSNOD: integer`: Number of `FEM`-nodes for which coordinates for stressfree configuration are specified. In this version `NFSNOD` must be equal to the total number of `FEM`-nodes

6.7.1.2. Coordinates for stressfree configuration, NFSNOD input lines

```
INOD X0 Y0 Z0
```

- `INOD: integer`: Node number 1 >= `INOD` >= `NFSNOD`
- `X0: real`: x-coordinate describing stressfree configuration
- `Y0: real`: y-coordinate describing stressfree configuration
- `Z0: real`: z-coordinate describing stressfree configuration

The coordinate must be consistent with the stressfree element length

6.7.2. Define Start Configuration

The file `CHFSTA' specified in Finite element analysis from start configuration, STAFEM, contains definition of start configuration. The file is a free format sequential ASCII-file.

The file description depends on the parameter `KFORM' specified in Finite element analysis from start configuration, STAFEM.

File description: `KFORM = 1`.

6.7.2.1. Number of nodes, one input line

```
NFSNOD
```

- `NFSNOD: integer`: Number of `FEM`-nodes for which coordinates for stressfree configuration are specified. In this version `NFSNOD` must be equal to the total number of `FEM`-nodes

6.7.2.2. Coordinates for start configuration, NFSNOD input lines

```
INOD X0 Y0 Z0 T11 T12 T13 T21 T22 T23 T31 T32 T33
```

- `INOD: integer`: Node number 1 >= `INOD` >= `NFSNODI`
- `X0: real`: x-coordinate describing stressfree configuration
- `Y0: real`: y-coordinate describing stressfree configuration
- `Z0: real`: z-coordinate describing stressfree configuration

Nodal base vector system: - x-direction of node system referred to the global system - `T11: real, default: 1`: component in global x-direction - `T12: real, default: 0`: component in global y-direction - `T13: real, default: 0`: component in global z-direction - y-direction of node system referred to the global system - `T21: real, default: 0`: - `T22: real, default: 1`: - `T23: real, default: 0`: - z-direction of node system referred to the global system - `T31: real, default: 0`: - `T32: real, default: 0`: - `T33: real, default: 1`:

6.7.3. Define uneven seabed geometry

The seabed geometry data is given on a regularly spaced grid. The grid can be rotated and translated relatively to the reference system of the seabed geometry file. (In addition the reference system of the seabed geometry file can be rotated relatively to the global reference system ref. Coordinates of the seabed file reference system).

A MATLAB script is available, to generate such a file from data given as a set of x, y, z coordinates

At each step of the static and dynamic analysis, it is checked that every node of the model has x and y coordinates that are within the grid. Excursions from the grid will cause the program to terminate.

6.7.3.1. Description text of geometry

```
CHBOTT
```

- `CHBOTT: character`: Descriptive text of geometry

6.7.3.2. Grid dimension and extension, one input file

```
NGX NGY XSmin XSmax YSmin YSmax DGX DGY
```

- `NGX: integer`: Number of points in the grid in the x direction
- `NGY: integer`: Number of points in the grid in the y direction
- `Xsmin: real`: Coordinate of the first point in the grid in the x direction $[\mathrm{L}]$
- `Xsmax: real`: Coordinate of the last point in the grid in the x direction $[\mathrm{L}]$. Used to check consistency of the grid input and whether a node is outside the grid.
- `Ysmin: real`: Coordinate of the first point in the grid in the y direction $[\mathrm{L}]$
- `Ysmax: real`: Coordinate of the last point in the grid in the y direction $[\mathrm{L}]$. Used to check consistency of the grid input and whether a node is outside the grid.
- `DGX: real`: Distance between grid points in the x direction $[\mathrm{L}]$
- `DGY: real`: Distance between grid points in the y direction $[\mathrm{L}]$

The x and y coordinates of the grid corners and the distances between grid ponts are converted to integer values with unit $[\mathrm{L}/100]$.

6.7.3.3. Grid orientation, one input line

```
XOL YOL ANGOL
```

- `XOL: real`: x-coordinate of the origin of the grid, in the seabed file reference system $[\mathrm{L}]$
- `YOL: real`: y-coordinate of the origin of the grid, in the seabed file reference system $[\mathrm{L}]$
- `ANGOL: real`: Angle between the x axis of the grid and the x axis of the seabed file reference system $[\mathrm{deg}]$

6.7.3.4. Depths at gridpoint, NGY input lines

`IZBOT1 ........ IZBOTngx`

- `IZBOT1: integer`: $100 \times$ Depth at first x value $[\mathrm{L}/100]$
- .
- .
- .
- `IZBOTngx: integer`: $100 \times$ Depth at last x value $[\mathrm{L}/100]$

The input line may be given over several lines of text by using the continuation character `&'.

7. Input to DYNMOD

7.1. General Information

The input description to the `DYNMOD` module is divided into 5 sections, each section describing one data-section referred to as A-E.

- Data Group A: Control Information
- Data Group B: Free Vibration Analysis
- Data Group C: Regular Wave, Time Domain Analysis
- Data Group D: Irregular Wave, Time Domain Analysis
- Data Group E: Time Domain Procedure and File Storage Parameters

Three different types of analysis are possible. Complete input for these types is shown in the list below.

- Type number 1, Free vibration, requires the data sections A and B.
- Type number 2, Regular wave, requires the data sections A, C and E.
- Type number 3, Irregular wave, requires the data sections A, D and E.

7.2. Data Group A: Control Information

This data-group is mandatory for all types of analysis with `DYNMOD`. The prescribed sequence must be followed.

7.2.1. Principal run parameters

7.2.1.1. Data group identifier, one input line

`DYNMod CONTrol INFOrmation CHVERS`

- `CHVERS: character(8)`: RIFLEX input file version, e.g. 3.2

7.2.1.2. Heading, three input lines

- Heading, line no 1
- Heading, line no 2
- Heading, line no 3

Identification of the run by alphanumerical text

Always three input lines which may all be blank. Each line may contain up to 60 characters

7.2.1.3. Options and identifiers, one input line

`IRUNCO IANAL IDRIS IDENV IDSTAT IDIRR IDRES`

- `IRUNCO: character(4), default: DATA`: Code for data check or executable run
    - `= FREM`: Data generation for `FREMOD`
    - `= DATA`: Data check
    - `= ANALysis`: Analysis
- `IANAL: character(4)`: Type of analysis to be performed
    - `= EIGEn`: Eigenvalue analysis.
        - Data section B must be given
    - `= REGUlar`: Regular wave, time domain analysis.
        - Data sections C and E must be given.
    - `= IRREgular`: Irregular wave, time domain analysis.
        - Data sections D and E must be given.
- `IDRIS: character(6)`: Data set identifier corresponding to data for one riser system established by `INPMOD` and followed by a static analysis. See INPMOD: Data Group B: Single Riser Data and STAMOD: Options and print switches.

`IDENV: character(6)`: Environment identifier, corresponding to data for one environment on file established by `INPMOD`. See data-group INPMOD: Identification of the environment of input description for `INPMOD`.

  - Reference to actual wave case is given in a later data-group
  - Dummy for `IRUNCO=DATAcheck`

- `IDSTAT: character(6)`: Static condition identifier, corresponding to data on file established by `STAMOD`. See STAMOD: Principal run parameters of input description for `STAMOD`.
- `IDIRR: character(6)`: Data set identifier for irregular wave and motion data, either established by a previous run or used as reference to results stored on file by this run.
- `IDRES: character(6)`: Data set identifier for this run, used as reference to results stored on files

7.2.2. Static load condition

For special purposes it may be convenient to change applied static loads at the start of dynamic analysis. This option should be used with care! One useful application is for analysing free vibration after scaling a static nodal force to zero (`SCALSF=0.0`).

Note that some static loads are applied and some modelling features are activated in dynamic analysis even if they were not applied nor activated in static analysis. See also Note: Static Analysis with Fixed Parameters and Parameter Variation.

7.2.2.1. Data group identifier, one input line

```
STATic LOAD CONDition
```

7.2.2.2. Scaling parameters, one line

```
SCALVF SCALSF SCALCF
```

- `SCALVF: real, default: 1`: Scaling of volume forces
- `SCALSF: real, default: 1`: Scaling of specified (nodal) forces
- `SCALCF: real, default: 1`: Scaling of current velocities

All forces are scaled simultaneously as $\mathrm{F_s} = \mathrm{SCALiF} \times \mathrm{F_s^0}$ Only `SCALSF` is active in the present version of the program.

7.2.3. Random number generator

In version 4.18 and later, the algorithm for generating pseudo-random numbers may be selected by the user. The ``mersenne twister'' is the recommended method and should be used unless backwards compatibility with previous versions is required. Note that the default value may change in a future release. The choice of random number generator will apply to: - generation of irregular wave time series - initial phase angles for time domain VIV loads specified for cross-sections in INPMOD - generation of phase angles for application of harmonic loads from a VIVANA frequency domain analysis.

It has been identified that the legacy method can give non-gaussian and non-stationary wave elevation in SIMO for short crested waves with more than about 30-50 discrete directions, depending on wave spectrum and simulation duration. By choosing the mersenne twister, these issues are avoided.

For coupled analysis, wave time series will be generated using the random number generator specified in SIMA.

7.2.3.1. Data group identifier, one input line

```
RANDom NUMBer GENErator
```

7.2.3.2. Random number generator input, one line

```
CHRAN  ISEED
```

- `CHRAN: character (7), default: LEGACY`: Choice of random number generator
    - `= 'LEGACY'`: Legacy random number generator used. Results will be consistent with previous RIFLEX versions.
    - `= 'TWISTER'`: Mersenne Twister’ random number generator used. Results will NOT be consistent with previous RIFLEX versions.
- `ISEED: integer, default: 7`: Starting parameter of random number generator for use when input of starting value is not available; e.g. time domain VIV loads. Currently not used.

7.3. Data Group B: Free Vibration Analysis

This data-group is given if and only if `IANAL=EIGEn`, see Options and identifiers, one input line.

7.3.1. Free vibration options

7.3.1.1. Data group identifier, one input line

```
FREE VIBRation OPTIons
```

7.3.1.2. Number of eigenvalues and -vectors, one input line

```
NEIG NVEC
```

- `NEIG: integer`: Number of eigenvalues to be calculated and stored on file
- `NVEC: integer`: Number of eigenvectors to be calculated and stored on file

7.3.1.3. Computation parameters, one input line

The parameters below correspond to Lanczos' method for solution of eigenvalue problems. For a detailed discussion, see B. Nour-Omid, B.N. Parlett, R.L. Taylor: Lanczos versus Subspace Iteration for Solution of Eigenvalue Problems, International Journal for Numerical Methods in Engineering, Vol. 19, pp. 859-871, 1983. or B.N. Parlett: The Symmetric Eigenvalue Problem, Prentice-Hall, 1980.

```
EPS1 EPS2 EPS3 KSR MAXIT KEX SHIFT MAXNIV
```

- `EPS1: real, default: 0`: Maximum acceptable relative error in computed eigenvalues
- `EPS2: real, default: 0`: Limit value for singularity test during factorization
- `EPS3: real, default: 0`: Orthogonality limit:
  - If $\mathbf{v}_\mathrm{i}{}^\mathrm{T}\mathbf{v}_\mathrm{i} = \delta_\mathrm{ii}$ and $\mathbf{v}_\mathrm{i}{}^\mathrm{T}\mathbf{v}_\mathrm{j} = \delta_\mathrm{ij}$ and $|\delta_\mathrm{ij}| < \mathrm{EPS3} \times \delta_\mathrm{ii}$, $\mathbf{v}_\mathrm{i}$ and $\mathbf{v}_\mathrm{j}$ are orthogonal
- `KSR: integer, default: 1`: Start vector code:
  - `KSR =`$\pm$`1`: a pseudo-random start vector is generated by the eigenvalue solver
  - `KSR =`$\pm$`2`: the diagonal of the mass matrix is used as start vector
  - `KSR =`$\pm$`3`: a start vector of unit elements is used
  - For positive `KSR` the start vector is premultiplied with $\mathbf{H}$ before use; if a negative value is specified the start vector is used directly.
- `MAXIT: integer, default: 5`: Maximum no of iterations in reorthogonalization.
  - If a negative value is specified, reorthogonalization is not iterative; e.g. `MAXIT = -2` will cause a two-pass Gram-Smith orthogonalization to be employed to all new $\mathbf{v}_\mathrm{i}$ (i>1), irrespective of `EPS3`.
  - For high values of `NEIG` (>50) a doublepass orthogonalization is recommended (`MAXIT = -2`)
- `KEX: integer, default: 0`: Parameter controlling the frequency with which the small tridiagonal eigenvalue problem is solved.

- Must be in the range of `0 ⇐ KEX ⇐ 5`.
- If a zero value is specified, a default value of 2 is used

- `SHIFT: real, default: 0`: The shift value σ
- `MAXNIV: integer, default: 0`: Number of Lanczos steps to be used.
  - A default value suitable for the eigenvalue routines is automatically computed if a `0` is specified.
  - `MAXNIV` should only be given a value $\neq$ `0` for small problems

If zero or negative values are specified for `EPS1-EPS3` default values are inserted

7.3.2. Print options for results

7.3.2.1. Data group identifier, one input line

```
EIGEnvalue PRINt OPTIons
```

7.3.2.2. Print selection parameters, one input line

```
NPEIG NPVEC IPRESW
```

- `NPEIG: integer`: Number of eigenvalues to be printed (⇐ `NEIG`)
- `NPVEC: integer`: Number of eigenvectors to be printed (⇐ `NVEC`)
- `IPRESW: integer, default: 0`: Debug print switch for eigenvalue routines

7.3.3. Termination of input data

To terminate an input data stream, simply give the following, which is interpreted as a data group identifier.

```
END
```

Note that the `END` image cannot be omitted

7.4. Data Group C: Regular Wave, Time Domain Analysis

This data group is given for `IANAL = REGUlar`, see Options and identifiers, one input line. Data-group A and E must also be given for complete definition of a regular time domain analysis.

7.4.1. Parameters for definition of analysis and further input

7.4.1.1. Data group identifier, one input line

```
REGUlar WAVE ANALysis
```

7.4.1.2. Analysis parameters, one input line

```
NPER NSTPPR IRWCN IMOTD
```

- `NPER: integer`: Number of periods for regular wave analysis, referring to wave or motion periods (of first vessel)
- `NSTPPR: integer, default: 80`: Number of integration time steps per period, recommended value: 50-120
- `IRWCN: integer`: Wave parameter
  - `IRWCN = 0`: No wave acting, motions must be present
  - `IRWCN = N`: Wave acting. Regular wave case N on actual environment used in present analysis
  - If no waves are acting, the period for harmonic motions is specified in Motion amplitudes of support vessel, one input line
- `IMOTD: integer`: Platform motion parameter
  - `IMOTD = 0`: No motions, waves must be present
  - `IMOTD = 1`: Platform motion generated on the basis of wave data (wave period and amplitude) and motion transfer functions. Reference to transfer functions given in Options and identifiers, one input line.
  - `IMOTD = 2`: Platform motions specified in Regular vessel motion

The platform motions are independent of the wave loading parameters given in Load modelling, regular waves.

Extreme values of response parameters from last integration period will normally be stored on file (cfr. File storage of displacement response). In addition, displacement histories from selected nodes and force and curvature histories from selected elements can be stored if wanted. Specification of such data storage is given in data groups File storage of displacement response, File storage for internal forces and File storage for curvature response.

7.4.2. Load modelling, regular waves

This data group is given if `IRWCN >= 1` (data group Analysis parameters, one input line above).

7.4.2.1. Data group identifier, one input line

```
REGUlar WAVE LOADing
```

7.4.2.2. Method for wave load calculation, one input line

```
IWTYP ISURF IUPPOS
```

- `IWTYP: integer, default: 1`: Wave theory parameter
  - `IWTYP = 1`: Airy linear wave theory
  - `IWTYP = 2`: Stoke 5th order wave theory
- `ISURF: integer, default: 1`: Sea surface definition, see the figure `Definition of sea surface' below.
  - Dummy if `IWTYP = 2`
  - `ISURF = 1`: Integration of wave forces to mean water level
  - `ISURF = 2`: Integration of wave forces to wave surface, deformation of potential by stretching and compression
  - `ISURF = 3`: Integration of wave forces to wave surface, move of potential to actual surface
  - `ISURF = 4`: Integration of wave forces to wave surface by keeping the potential constant from mean water level to wave surface
- `IUPPOS: integer, default: 2`: Riser position parameter
  - `IUPPOS = 0`: as 1, but the riser is kept fixed in static position, for computation of surface penetrating element. I.e. a node that is wet or dry at the end of the static analysis will continue to be considered wet or dry with regards to kinematics in the dynamic simulation. Recommended only for comparison with linear methods.
  - `IUPPOS = 1`: Wave induced velocities and accelerations calculated at static riser position
  - `IUPPOS = 2`: Wave induced velocities and accelerations calculated at updated (dynamic) positions

Note: The option `IUPPOS = 0` cannot be combined with linear analysis, `ITDMET = 1`, or nonlinear analysis, `ITDMET = 2` and `SIMO` bodies.

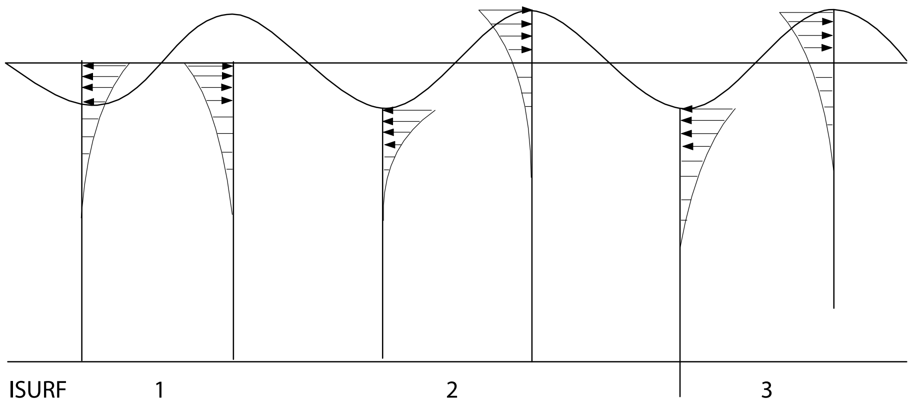

Figure 34. Definition of sea surface

7.4.3. Regular vessel motion

This data group is given only if `IMOTD=2` (see input group Analysis parameters, one input line).

7.4.3.1. Data group identifier, one input line

```
REGUlar VESSel MOTIon
```

7.4.3.2. Definition of vessel motion, two lines for each vessel

`Motion amplitudes of support vessel' and `Motion phase angles' must be given for all `NVES` vessels in systems (totally 2x`NVES` lines).

7.4.3.2.1. Motion amplitudes of support vessel, one input line

Forced displacements are specified for the support vessel. Forced displacements for the terminal points are found by transformations.

```
XAMP YAMP ZAMP XRAMP YRAMP ZRAMP PER
```

- `XAMP: real`: Motion amplitude, global x-direction [L]
- `YAMP: real`: Motion amplitude, global y-direction [L]
- `ZAMP: real`: Motion amplitude, global z-direction [L]
- `XRAMP: real`: Motion amplitude, global x-rotation [degrees]
- `YRAMP: real`: Motion amplitude, global y-rotation [degrees]
- `ZRAMP: real`: Motion amplitude, global z-rotation [degrees]
- `PER: real`: Period of motion [T]

`PER` is dummy input if a regular wave is specified, i.e. `IRWCN > 0` (data group Analysis parameters).
In the case of multiple vessels, `PER` is only read for the first vessel and the specified period used for all vessels.

7.4.3.2.2. Motion phase angles, one input line

`XPHA YPHA ZPHA XRPHA YRPHA ZRPHA`

- `XPHA: real`: Phase angle, x-motion $[\mathrm{degrees}]$
- `YPHA: real`: Phase angle, y-motion $[\mathrm{degrees}]$
- `ZPHA: real`: Phase angle, z-motion $[\mathrm{degrees}]$
- `XRPHA: real`: Phase angle, x-rotation $[\mathrm{degrees}]$
- `YRPHA: real`: Phase angle, y-rotation $[\mathrm{degrees}]$
- `ZRPHA: real`: Phase angle, z-rotation $[\mathrm{degrees}]$

All phase angles are defined as follows:

Positive angle: Forward phase shift; motion before sea surface at global origin.

Surface: $\eta = \eta_{\mathrm{a}} \sin(\omega \mathrm{t} + \phi_{\mathrm{p}}), \quad \phi_{\mathrm{p}} = -\mathrm{kx} \cos(\beta) - \mathrm{ky} \sin(\beta)$

Motion: $\mathrm{x_i} = \mathrm{x_{ai}} \sin(\omega \mathrm{t} + \phi_{\mathrm{i}})$

Where: - $\mathrm{x_i}$ is equation of motion - η_{a} is wave amplitude - $\mathrm{x_{ai}}$ is motion amplitude `XAMP`, `YAMP`, etc. - ϕ_{i} is phase angle, `XPHA`, `YPHA`, etc. - k is wave number - ω is angular frequency - x, y is global coordinates

If the forward phase shift $\phi_{\mathrm{i}}^{\mathrm{xy}}$ between wave and motion at the same point (x,y) is known, the phase into `RIFLEX` must be modified as follows:

$\phi_{\mathrm{i}} = \phi_{\mathrm{i}}^{\mathrm{xy}} + \phi_{\mathrm{p}}$

in order to obtain phase relation between motion at (x,y) and a wave with start at global origin as defined above.

7.5. Data Group D: Irregular Wave, Time Domain Analysis

This data group is given for `IANAL=IRREgular`, see Options and identifiers, one input line. Data group A and E must also be given for complete definition of an irregular time domain analysis.

7.5.1. Irregular time series parameters

The input in this data group is used to specify the method used for computation of the underlaying irregular waves, i.e. the seed used for random number generation and the frequency resolution.

The data group may be skipped if default values are wanted. The data group is dummy if any floater force models are present in the model. (The analysis is done in combination with `SIMO`, so-called coupled analysis, and the irregular time series parameters defined by input to `SIMO`).

7.5.1.1. Data group identifier, one input line

```
IRREgular TIMEseries PARAmeters
```

7.5.1.2. Parameters, one input line

```
IRAND TIMGEN DTGEN CHFREQ CHAMP
```

- `IRAND: integer, default: 1`: Starting parameter of random number generator
- `TIMGEN: real, defaul: 16384`: Length of prescribed wave and motion time series $[\mathrm{T}]$
- `DTGEN: real, defaul: 0.5`: Time increment of pre-sampled time series $[\mathrm{T}]$
- `CHFREQ: character(4), default: FFT`: Option for selecting wave frequency components
    - `= 'FFT'`: Wave frequency components are selected among the FFT frequencies given by `TIMEGEN` and `DTGEN`. The default criteria are used to find the first and last frequencies.
- `CHAMP: character(5), default: DET`: Option for selecting wave component amplitudes
    - `= 'DET'`: Deterministic wave amplitudes are used.
    - `= 'STOCH'`: Stochastic wave amplitudes are used.
    - `= 0`: Interpreted as `DET`. Included for compatibility with earlier versions.

Note that this data group is dummy for coupled analysis.

Also note that: - `TIMGEN` should be equal or longer than the simulation length, `TIME`, given in Irregular response analysis and subsequent input. - `TIMGEN` will, if necessary, be increased to give a power of 2 time steps (`DTGEN`). - The actual time increment used for time domain analysis is defined by the parameter `DT`, see Irregular response analysis and subsequent input. - To represent the wave surface- and motion time series properly, time increments, `DTGEN`, in the range 0.5-1 s are normally acceptable.

7.5.2. Irregular response analysis and subsequent input

7.5.2.1. Data group identifier, one input line

```
IRREgular RESPonse ANALysis
```

7.5.2.2. Analysis parameters, one input line

```
IRCNO TIME DT CHWAV CHMOT CHLFM TBEG ISCALE
```

- `IRCNO: integer/character`: Irregular wave case number in actual environment applied in this run. Dummy for coupled analysis.
  - `IRCNO = FILE` or `IRCNO = -1`: Wave time series read from file. Data groups Irregular waves and Wave time series file must be given
- `TIME: real, default: 11000`: Length of simulation $[\mathrm{T}]$
- `DT: real, default: 0.1`: Time step $[\mathrm{T}]$
  - See below
- `CHWAV: character(4), default: NEW`: Irregular wave indicator
  - `= 'NONE'`: No wave forces in present analysis. If specified the riser will have forced excitation at upper end and oscillate in undisturbed water or in constant current
  - `= NEW`: Wave forces present. New data generated. Data group Irregular waves must be given.
- `CHMOT: character(4), default: STAT`: Irregular motion indicator
  - `= 'NONE'`: No irregular motions in present analysis
  - `= STAT`: Forced irregular motions present. Computation of prescribed motions will be based on vessel position in final static position.
  - `= NEW`: Interpreted as `CHMOT=STAT`
  - `= FILE`: Forced irregular motions present. Wave frequency motion time series read from file. Data group Wave frequency motion time series file must be given.
- `CHLFM: character(4), default: 'NONE'`: Low frequency motion indicator
  - `= 'NONE'`: No low frequency irregular motions present
  - `= FILE`: Forced low frequency irregular motions present. Low frequency motion time series read from file. Data group Low frequency motion time series file must be given.

`TBEG: real, default: 0`: Time in wave and motion time series that dynamic simulation will start from $[\mathrm{T}]$

- `ISCALE: integer, default: 0`: Switch for scaling of terminal point motions
    - `ISCALE = 0`: No scaling
    - `ISCALE = 1`: Scaling: Input line Support vessel motion scaling factors has to be given

`DT` will be adjusted to get an integer ratio between `DTGEN` and `DT`. `DT` given as negative integer defines the ratio between time step used in pre-simulation of waves and/or WF-motions and the time step to be used in the time simulation. (`DTGEN/DT >= 1`)

`TBEG` allows for arbitrary start point in the pre-generated time series. If the end of the time series is reached during dynamic integration, a warning is written and motions and water kinematics will be taken from the start. This can also be useful for elimination of transients from the time series statistics.

An irregular analysis without waves or vessel motions may be run by specifying `CHWAV = 'NONE'`, `CHMOT = 'NONE'` and `CHLFM = 'NONE'`. `IRCNO` must still reference a legal irregular wave case, but the wave will not be used as no wave kinematics will be generated and not vessel motions be applied.

7.5.2.3. Support vessel motion scaling factors. Only given for ISCALE=1. One line for each vessel in system (NVES lines)

```
SCALX SCALY SCALZ SCALXR SCALYR SCALZR
```

- `SCALX: real, default: 1`: Scaling for global X-motion $[1]$
- `SCALY: real, default: 1`: Scaling for global Y-motion $[1]$
- `SCALZ: real, default: 1`: Scaling for global Z-motion $[1]$
- `SCALXR: real, default: 1`: Scaling for global X-rotation $[1]$
- `SCALYR: real, default: 1`: Scaling for global Y-rotation $[1]$
- `SCALZR: real, default: 1`: Scaling for global Z-rotation $[1]$

The motions are scaled directly as $\mathrm{DISP_i} = \mathrm{SCAL_i} \times \mathrm{Motion_i}$ where $\mathrm{Motion_i}$ is the precomputed motion quantity $_\mathrm{i}$.

7.5.3. Irregular waves

This data group is omitted for `CHWAV='NONE'`, see data group Analysis parameters, one input line.

The data group also controls the method for computation of wave kinematics and motions of the support vessels. In this context FFT or FFT and cosine series combined means that the vessel motion is pre-generated by means of FFT, while the wave kinematics are either pre-generated (FFT) or computed during the actual simulation by use of cosine series. `Cosine series only'' means that both vessel motion and wave kinematics are computed based on cosine series. It is possible to overrule the cosine series application for wave kinematics for parts of the the system by specifying FFT in the detailed specifications, see Additional detailed specification of wave kinematics points (optional).` (FFT'' or ``FFT and cosine series combined only''.)

7.5.3.1. Data group identifier, one input line

```
IRREgular WAVE PROCedure
```

7.5.3.2. Procedure for wave force calculation, one input line

```
IUPPOS ISURF KINOFF CHSTEP NODSTP ZLOWER ZUPPER IOPDIF IOPWKI
```

- `IUPPOS: integer, default: 1`: Position for calculation of irregular wave kinematics
    - `= 1`: Kinematics at static positions
    - `= 2`: Kinematics at instantaneous positions calculated by summation of cosine components.
    - `= -2`: Kinematics at static positions calculated by summation of cosine components. This option is mainly useful for testing.
    - `= 0`: As 1 but riser fixed in static position, (`wet'' elements also` wet'' dynamic)
- `ISURF: integer, default: 1`: Code for kinematics in wave zone
    - `= 1`: Integration of wave forces to mean water level
    - `= 2`: Integration of wave forces to wave surface by stretching and compression of the wave potential
    - `= 3`: Integration of wave forces to wave surface by moving the potential to actual surface
    - `= 4`: Integration of wave forces to wave surface by keeping the potential constant from mean water level to wave surface
    - `= 5`: 2nd order wave (integration of wave forces to wave surface)
        - The formulation for 2nd order wave kinematics is based on the Stoke 2nd order wave theory. Only available for kinematics calculated at static position; `IUPPOS = 1 or IUPPOS = 0`.

- `KINOFF: integer, default: 0`: Code for default kinematics points procedure
  - `= 0`: Default procedure on. The initial selection of positions for computation of kinematics is determined by the parameters `NODSTP`, `ZLOWER` and `ZUPPER` for all lines in the system. Subsequent specification (see Additional detailed specification of wave kinematics points (optional)) will replace the initial selection.
  - `= 1`: Default procedure off. Kinematics will only be computed at positions given by subsequent specification (see Additional detailed specification of wave kinematics points (optional))
- `CHSTEP: character(4)`: Code for interpretation of the next parameter
  - `= NODE`: Next parameter interpreted as `NODSTP`
- `NODSTP: integer`: Node step for calculating wave kinematics. (Dummy for `KINOFF = 1`)
  - Kinematics calculated for every `NODSTP` node between `ZLOWER` and `ZUPPER` (see Definition of NODSTP, ZLOWER and ZUPPER).
  - For intermediate nodes kinematics are derived by linear interpolation.
  - Wave kinematics will always be calculated at submerged supernodes.
  - Note that a negative value of `NODSTP` may be given. The distance between `ZUPPER` and `ZLOWER` is then divided into 4 (equal) intervals and `NODSTP` is increased from `ABS(NODSTP)` in the upper interval via `2xABS(NODSTP)` in the next interval and `4xABS(NODSTP)` to `8xABS(NODSTP)` in the two lower intervals, see Definition of NODSTP, ZLOWER and ZUPPER.
- `ZLOWER: real, default: -WDEPTH`: Z-coordinate indicating lowest node position for which wave kinematics are calculated $[\mathrm{L}]$
  - See Definition of NODSTP, ZLOWER and ZUPPER.
  - Dummy for `KINOFF = 1`
  - For `WDEPTH`, see INPMOD: Water depth and wave indicator
- `ZUPPER: real, default: 4 x STD_WA`: Upper limit for wave kinematics $[\mathrm{L}]$
  - Dummy for `KINOFF = 1`
  - `STD_WA` is the standard deviation of the total wave elevation
- `IOPDIF: integer, default: 0`: Option for specification of wave kinematic transfer function.

- `IOPDIF = 0`: No transfer function to be specified
- `IOPDIF = 1`: Read transfer functions from the file specified in Wave kinematics transfer function file name (below).

- `IOPWKI: integer, default: 0`: Option for specification of wave kinematic time series.
  - `IOPWKI = 0`: No wave kinematics time series to be specified
  - `IOPWKI = 2`: Read wave kinematics time series from the binary file specified in Wave kinematics time series file name (below).

`NODSTP`, `ZLOWER` and `ZUPPER` will normally be sufficient for specifying the selection of wave kinematics points.

Note that for large or complicated systems Additional detailed specification of wave kinematics points (optional) may be used to override the selection given by `NODSTP`, `ZLOWER` and `ZUPPER`; e.g. skip generation of wave kinematics for selected lines, generate kinematics at more points along an important line.

Definition of NODSTP, ZLOWER and ZUPPER

Note that the definition of ISURF is also used to determine where to apply wind forces to airfoil cross sections near the water line. That is, no wind forces are applied to wet sections of the element. The wind speed is nevertheless taken to be zero at or below the mean water level.

Note that the option `IUPPOS = 0` cannot be combined with linear analysis, `ITDMET = 1`, or nonlinear analysis, `ITDMET = 2` and `SIMO` bodies.

7.5.3.2.1. Wave kinematics transfer function file name

This data group is omitted for `IOPDIF = 0` ~ CHFDIF ~

- `CHFDIF: character(80)`: File name with wave kinematic transfer function.
  - The file format is described in Diffracted Wave Transfer Functions at Points.

7.5.3.2.2. Wave kinematics time series file name

This data group is omitted for `IOPWKI = 0`

Wave kinematics read from file will replace the corresponding wave kinematics calculated by `DYNMOD`. These kinematics will then be used in the calculation of Morison type hydrodynamic loads on `RIFLEX` elements. Loads on `SIMO` bodies will NOT be affected.

`RIFLEX` vessel motions based on vessel motion transfer functions and MacCamy Fuchs and Potential flow loads on `RIFLEX` elements are per-generated from the wave Fourier components and are therefore NOT affected by the wave kinematics read from file. Elements with MacCamy Fuchs or Potential flow loads may not have kinematics read from file.

If kinematics read from file are used in a simulation with `SIMO` bodies, vessel motions based on vessel motion transfer functions or pre-generated hydrodynamic loads, the user must ensure that the kinematics are consistent with the Fourier components.

```
CHFDIF
```

- `CHFWKI: character(80)`: File name with wave kinematic time series.
  - The file format is specified by `IOPWKI` and is the same as the kinematics file with the same format exported from DYNMOD using [Storage of irregular wave kinematics](@ref dynmod_d_storage)

```
ICOLMX ICOLTM
```

- `ICOLMX: integer, default: 0`: Maximum number of columns on file. For binary format, `IOPWKI = 2`, this includes the two columns of FORTRAN specific data. Please see the key file `key_<prefix>_wavkin.txt` generated when storing kinematics.
- `ICOLTM: integer, default: 2`: Column number on file for time

7.5.3.2.3. Additional detailed specification of wave kinematics points (optional)

As many input lines as needed. Note three alternative formats.

======= For wave kinematics calculated by the program from the undisturbed waves.

```
LINE-ID CHSTEP = NODE NODSTP
```

- `LINE-ID: character(8)`: Line identifier
- `CHSTEP: character(4)`: `= Node`
- `NODSTP: integer`: Node step for calculating wave kinematics
  - `= 0`: No kinematics for this line
  - `> 0`: Kinematics for each `NODSTP` node

======= For wave kinematics given by wave kinematics transfer functions (diffracted waves)

```
LINE-ID CHSTEP= DIFF ILSEG ILNODE IVES PTNOUS
```

- `LINE-ID: character(8)`: Line identifier
- `CHSTEP: character(4)`: `= DIFF`
- `ILSEG: integer`: Local segment number within line LINE-ID

- `ILNODE: integer`: Local node number within ILSEG
- `IVES: integer`:
  - `= 0`: Use undisturbed wave kinematics at this node
  - `> 0`: Support vessel number. Used as reference to transfer function for diffracted wave kinematics.
- `PTNOUS: integer`: Point reference(s) to transfer function for diffracted wave kinematics

Up to 30 values of PTNOUS may be given on each line. The diffracted kinematics at the specified node will be generated by interpolation based on the nearest point references.

======= For wave kinematics given by wave kinematics time series

`LINE-ID CHSTEP= WKFI ILSEG ILNODE ICOLST`

- `LINE-ID: character(8)`: Line identifier
- `CHSTEP: character(4)`: `= WKFI`
- `ILSEG: integer`: Local segment number within line LINE-ID
- `ILNODE: integer`: Local node number within ILSEG
- `ICOLST: integer`: Column number for the first wave kinematics time series for this node

7.5.4. Wave and motion time series files

7.5.4.1. Wave time series file

This data group is given only if `IRCNO = FILE`.

7.5.4.1.1. Data group identifier, one input line

`WAVE TIME SERIEs`

7.5.4.1.2. Wave time series file information

`CHFTSF IFORM ICOTIM ICOWAV`

- `CHFTSF: character(60)`: File name
- `IFORM: integer, default: 1`: File format code
  - `= ASCI`: Column organised ASCII file

    - `= STAR`: Startimes file

- `ICOTIM: integer, default: 1`: Column number for time
    - Dummy for `IFORM = STAR`
- `ICOWAV: integer/real, default: 2`: Column or time series number for wave elevation

The wave direction is given by the parameter `WADR1` given in `INPMOD` for the irregular wave case `IRCNO` referred to in Analysis parameters, one input line.

`ICOTIM` and `ICOWAV` will refer to columns on an ASCII file; e.g. `ICOTIM=1` and `ICOWAV=2` if the time and wave elevation are in the first and second columns; or to a time series number on a Startimes file; e.g. `ICOWAV=10.01` for time series 10, version 1.

An arbitrary time step may be used on an ASCII file, while the Startimes file has a fixed step. Linear interpolation is used to get the motions at the time step (DTWF)

7.5.4.1.3. Direction, location of measurement and cut-off for filtering

```
WAVDIR XGWAV YGWAV TMIN TMAX
```

- `WAVDIR: real, default: 0`: Wave direction [deg]
- `XGWAV: real, default: 0`: Global x-coordinate for position where time series is measured
- `YGWAV: real, default: 0`: Global y-coordinate for position where time series is measured
- `TMIN: real, default: 0`: Period corresponding to cut-off frequency for filtering
- `TMAX: real, default: 0`: Period corresponding to cut-off frequency for filtering

If `TMIN` and `TMAX` are both zero: No filtering

If `TMIN` and `TMAX` are both different from zero: band-pass filtering

Filtering is not implemented in present version

7.5.4.2. Wave frequency motion time series file

This data group is given only if `CHMOT=FILE`. Note that data must be given for all vessels in the system.

7.5.4.2.1. Data group identifier, one input line

`WFMOtion TIME SERIes`

7.5.4.2.2. Wave frequency motions file information, NVES input lines

`IVES CHFTSF IFORM IKIND IROT ICOTIM ICOXG ICOYG ICOZG ICOXGR ICOYGR ICOZGR`

- `IVES: integer`: Vessel Number
- `CHFTSF: character(60)`: File name
- `IFORM: character(4), default: ASCI`: File format code
    - `= ASCI`: Column organised ASCII file
    - `= STAR`: Startimes file
    - `= NONE`: No wave frequency motions for this vessel. The remainder of this input line is dummy
- `IKIND: character(4), default: POSI`: Kind of motion time series input
    - `= POSI`: Global positions, i.e. global coordinates. The rotations are applied in the Euler sequence: Rz-Ry-Rx. Consistent with vessel motion time series from `SIMO`.
    - `= DYND`: Global dynamic displacements; i.e. global coordinates minus the final static position. The rotations are applied in the Euler sequence: Rx-Ry-Rz
- `IROT: character(4), default: DEGR`: Unit of rotations
    - `= DEGR`: Rotations given in degrees
    - `= RADI`: Rotations given in radians
- `ICOTIM: integer, default: 1`: Column number for time
    - Dummy for `IFORM = STAR`
- `ICOXG: integer/real, default: 0`: Column or time series number for specification of global x-motion. Absolute position if `IKIND = POSI`, relative to static position if `IKIND = DYND`.
- `ICOYG: integer/real, default: 0`: Column or time series number for specification of global y-motion. Absolute position if `IKIND = POSI`, relative to static position if `IKIND = DYND`.
- `ICOZG: integer/real, default: 0`: Column or time series number for specification of global z-motion. Absolute position if `IKIND = POSI`, relative to static position if `IKIND = DYND`.

- `ICOXGR: integer/real, default: 0`: Column or time series number for specification of global x-rotation. Absolute position if `IKIND = POSI`, relative to static position if `IKIND = DYND`.

- `ICOYGR: integer/real, default: 0`: Column or time series number for specification of global y-rotation. Absolute position if `IKIND = POSI`, relative to static position if `IKIND = DYND`.

- `ICOZGR: integer/real, default: 0`: Column or time series number for specification of global z-rotation. Absolute position if `IKIND = POSI`, relative to static position if `IKIND = DYND`.

Dofs may be omitted by giving `ICOxxx=0`

`ICOxxx` will refer to a column for an ASCII file; e.g. `ICOX=2` if the dynamic x motion time series is in the second column; or to a time series number for a Startimes file, e.g. `ICOX=1.02` for time series 1, version 2.

An arbitrary time step may be used on an ASCII file, while the Startimes file has a fixed time step. Linear interpolation is used to get the motions at the time step (`DTWF`).

Translational dofs are given in length units. Rotational dofs are given in degrees or radians, depending on the option `IROT`.

If only one rotation is nonzero or if all rotations are small, the order in which the rotations are applied will not be significant.

Please note that the line length of ASCII input files is currently limited to 260 characters, see Formats in How to Run the Program. Note that a RIFLEX input line may be split into several lines on the input file.

7.5.4.3. Low frequency motion time series file

This data group is given only if `CHLFM=FILE`. Note that data must be given for all vessels in the system.

7.5.4.3.1. Data group identifier, one input line

```
LFMOtion TIME SERIes
```

7.5.4.3.2. Low frequency motions file information, NVES input lines

```
IVES CHFTSF IFORM IKIND IROT ICOTIM ICOXG ICOYG ICOZGR
```

- `IVES: integer`: Vessel Number

- `CHFTSF: character(60)`: File name

- `IFORM: character(4), default: ASCI`: File format code
    - `= ASCI`: Column organised ASCII file
    - `= STAR`: Startimes file
    - `= NONE`: No wave frequency motions for this vessel. The remainder of this input line is dummy
- `IKIND: character(4), default: POSI`: Kind of motion time series input
    - `= POSI`: Global positions, i.e. global coordinates. The rotations are applied in the Euler sequence: Rz-Ry-Rx. Consistent with vessel motion time series from `SIMO`.
    - `= DYND`: Global dynamic displacements; i.e. global coordinates minus the final static position. The rotations are applied in the Euler sequence: Rx-Ry-Rz
- `IROT: character(4), default: DEGR`: Unit of rotations
    - `= DEGR`: Rotations given in degrees
    - `= RADI`: Rotations given in radians
- `ICOTIM: integer, default: 1`: Column number for time
    - Dummy for `IFORM = STAR`
- `ICOXG: integer/real, default: 0`: Column or time series number for specification of global x-motion. Absolute position if `IKIND = POSI`, relative to static position if `IKIND = DYND`.
- `ICOYG: integer/real, default: 0`: Column or time series number for specification of global y-motion. Absolute position if `IKIND = POSI`, relative to static position if `IKIND = DYND`.
- `ICOZGR: integer/real, default: 0`: Column or time series number for specification of global z-rotation. Absolute position if `IKIND = POSI`, relative to static position if `IKIND = DYND`.

Dofs may be omitted by given `ICOxxx=0`

`ICOxxx` will refer to a column for an ASCII file; e.g. `ICOSUR=2` if the dynamic x motion time series is in the second column; or to a time series number for a Startimes file, e.g. `ICOX=1.02` for time series 1, version 2.

An arbitrary time step may be used on an ASCII file, while the Startimes file has a fixed time step. Linear interpolation is used to get the motions at the time step (`DTWF`).

Translational dofs are given in length units. Rotational dofs are given in degrees or radians, depending on the option `IROT`.

If only one rotation is nonzero or if all rotations are small, the order in which the rotations are applied will not be significant.

7.5.5. Print options for FFT analysis

7.5.5.1. Data group identifier, one input line

```
IRREgular FOURier PRINt
```

7.5.5.2. Fourier print options

```
IPMOTI IPWAFO IPHFTS IPLFTS IPTOMO IPVEAC
```

- `IPMOTI: integer, default: 0`: Print option for the main routine
  - `⇐ 0`: No print
  - `> 1`: Key information printed
  - `> 2`: Some more data printed
  - `> 5`: Low level debug print during numerical integration activated
- `IPWAFO: integer, default: 0`: Print option for the wave fourier component generation
  - Not active in present version
- `IPHFTS: integer, default: 0`: Print option for HF-time series generation
  - `⇐ 0`: No print
  - `> 0`: Print of wave frequency vessel motion time series
- `IPLFTS: integer, default: 0`: Print option for LF-time series generation
  - `⇐ 0`: No print
  - `> 0`: Print of low frequency vessel motion time series
- `IPTOMO: integer, default: 0`: Print option for TOTAL motion time series generation
  - `⇐ 0`: No print
  - `> 0`: Print of total vessel motion time series

`IPVEAC: integer, default: 0`: Print option for generation of water particle velocities and acceleration

  - `⇐ 0`: No print
  - `> 1`: Key information printed
  - `> 2`: Some data printed
  - `> 5`: Extensive debug print of arrays with water particle velocities and accelerations

This data-group is normally supposed to be omitted. Increasing value of print options gives increasing amount of print.

7.5.6. Storage of irregular wave kinematics (optional)

7.5.6.1. Data group identifier, one input line

```
IRREgular KINEmatics STORage
```

7.5.6.2. Wave kinematics storage options one input line

```
NLKINE IKINFM
```

- `NLKINE: integer, default: 0`: Number of specifications for storage of wave kinematics
  - `= 0`: Wave elevation, velocities and accelerations are stored for all kinematics nodes. Currently, no other value is allowed.
- `IKINFM: integer, default: 2`: File format for kinematics storage
  - `= 1`: ASCII format
  - `= 2`: Binary format

Pre-generated wave kinematics are written to `<prefix>_wavkin.asc` or `<prefix>_wavkin.bin`. Kinematics calculated during the simulation; `IUPPOS = 2 or -2`; are written to `<prefix>_updkin.asc` or `<prefix>_updkin.bin`.

The contents are described in `key_<prefix>_wavkin.txt` or `key_<prefix>_updkin.txt`.

7.6. Data Group E: Time Domain Procedure and File Storage Parameters

This data group must always be given for `IANAL = REGU` and `IRRE` (time domain analysis) specified in input line Options and identifiers, one input line.

7.6.1. Method of analysis and subsequent input

7.6.1.1. Data group identifier, one input line

```
TIME DOMAin PROCedure
```

7.6.1.2. Method and subsequent input, one input line

```
ITDMET INEWIL
```

- `ITDMET: integer, default: 1`: Method indicator
  - `ITDMET = 0`: Prestochastic analysis only. The rest of the data in input groups E are irrelevant
  - `ITDMET = 1`: Linear analysis
  - `ITDMET = 2`: Nonlinear analysis. More information to define method is given in Nonlinear step by step integration
- `INEWIL: integer, default: 1`: Procedure indicator
  - `INEWIL = 1`: Newmark's procedure
  - `INEWIL = 2`: Wilson's procedure, illegal for non-linear analysis

7.6.1.3. Time integration and damping parameters, one input line

This data group can be omitted if default values are wanted.

```
BETIN GAMMA TETHA A1 A2 A1T A1TO A1B A2T A2TO A2B DAMP_OPT
```

- `BETIN: real, default: 4/6`: Inverse value of beta-parameter of the Newmark beta-family of integration operators
  - `BETIN = 4.0` gives beta=1/4,i.e. constant average acceleration method
- `GAMMA: real, default: 0.5`: Value of the parameter gamma of the Newmark operators (usually equal to 0.5)
- `TETHA: real, default: See below`: Value of the parameter tetha in Wilson's integration method
- `A1: real, default: 0`: Global mass proportional damping factor a_1, see definition below
- `A2: real, default: 0.001/0`: Global stiffness proportional damping factor a_2
- `A1T: real, default: 0`: Additional local mass proportional damping factor a_{1t} for tension

- `A1TO: real, default: 0`: Additional local mass proportional damping factor a_{1to} for torsion
- `A1B: real, default: 0`: Additional local mass proportional damping a_{1b} for bending
- `A2T: real, default: 0`: Additional local stiffness proportional damping factor a_{2t} for tension
- `A2TO: real, default: 0`: Additional local stiffness proportional damping factor a_{2to} for torsion
- `A2B: real, default: 0`: Additional local stiffness proportional damping factor a_{2b} for bending
- `DAMP_OPT: character(4), default: TOTA`: Option for stiffness contribution to Rayleigh damping
  - `= TOTA`: Stiffness proportional damping is applied using total stiffness, i.e. both material and geometric stiffness
  - `= MATE`: Stiffness proportional damping is applied using material stiffness only

Default values:

- For `INEWIL=1` (Newmark) the following alternative default values are: `BETIN=4.0`, `THETA=1.0`, `A2=0.001`
- For `INEWIL=2` (Wilson) default values are: `BETIN=6.0`, `TETHA=1.4` (linear)

7.6.1.3.1. Global proportional damping formulation:

$$\mathbf{C} = a_1\mathbf{M} + a_2\mathbf{K}$$

This means that the global damping matrix $\mathbf{C}$ is established as a linear combination of the global mass ($\mathbf{M}$) and the total or material stiffness ($\mathbf{K}$) matrices.

The mass and stiffness-proportional damping specified here will not be applied to elements for which mass- and/or stiffness-proportional damping is specified in INPMOD.

7.6.1.3.2. Numerical values of a_1 and a_2:

Let the structural damping to critical damping ratio, $c/(2m\omega)$, at two natural frequencies ω_1 and ω_2 be λ_1 and λ_2, respectively.
Then `A1` and `A2` can be computed as: - $a_1 = \frac{2\omega_1\omega_2}{\omega_2^2-\omega_1^2}(\lambda_1\omega_2 - \lambda_2\omega_1)$ - $a_2 = \frac{2(\omega_2\lambda_2-\omega_1\lambda_1)}{\omega_2^2-\omega_1^2}$

7.6.1.3.3. Additional local proportional damping formulation:

In this approach, the damping coefficients are introduced in the local degrees of freedom in order to allow for different damping levels in bending, torsion and tension. The element damping matrix can the be written as

$$\mathbf{c} = a_1\mathbf{M} + a_{1t}\mathbf{m}_t + a_{1to}\mathbf{m}_{to} + a_{1b}\mathbf{m}_b + a_2\mathbf{K} + a_{2t}\mathbf{k}_t + a_{2to}\mathbf{k}_{to} + a_{2b}\mathbf{k}_b$$

where subscripts $_t$, $_{to}$ and $_b$ refer to tension, torsion and bending contributions, respectively, and the matrices $\mathbf{c}$, $\mathbf{m}$ and $\mathbf{k}$ are local element matrices; e.g. $\mathbf{k}_b$ includes all bending deformation terms in the local element stiffness matrix.

For cross sections applied for blades of a operating wind turbine the matrix $\mathbf{k}$ should only include the material stiffness matrix. The geometric stiffness matrix should not be included as this would introduce damping of the rigid body motion.

One should be careful with global mass proportional damping as this may introduce internal damping from rigid body motion.

If $a_1 = 0$, a_2 simply becomes $2\lambda/\omega$.

Note that proportional damping (global and local) adds to a possible structural damping arising from hysteresis in bending moment/curvature relation.

7.6.1.4. Non-linear force model, one input line. Always submit for linear and non-linear analysis

`INDINT INDHYD MAXHIT EPSHYD TRAMP INDREL ICONRE ISTEPR LDAMP`

- `INDINT: integer, default: 1`: Indicator for modelling forces from internal slug flow
  - Nonlinear analysis only.
  - `INDINT = 1`: Forces from internal slug flow not considered
  - `INDINT = 2`: Forces from internal slug flow considered.
  - Data group Slug force calculations or Import of internal flow data from file must be given.
- `INDHYD: integer, default: 1`: Indicator for hydrodynamic force model. Linear analysis only.
  - (see `Dynamic Time Domain Analysis' in the Theory Manual).
  - `INDHYD = 1`: No force iteration, use of displacements and velocities at previous time step
  - `INDHYD = 2`: No force iteration, use of displacements, velocities and accelerations at previous time step (not recommended)

    `INDHYD = 3`: Force iteration performed

- `MAXHIT: integer, default: 5`: Maximum number of load iterations. Linear analysis only.
    - A negative value gives print of convergence for each step, then `MAXHIT = ABS(MAXHIT)`
- `EPSHYD: real, default: 0.01`: Convergence control parameter for force iteration. Linear analysis only.
    - Dummy for `INDHYD = 1, 2` $[1]$
- `TRAMP: real, default: 10`: Duration of start-up procedure $[T]$
- `INDREL: integer, default: 0`: Indicator for rupture/release
    - `INDREL = 0`: No riser rupture/release
    - `INDREL = 1`: Riser rupture/release will be simulated
- `ICONRE: integer, default: 0`: Ball joint connector no. to be released
    - `ICONRE = 0`: All ball joint connectors in the system are released simultaneously
    - `ICONRE = i`: Ball joint connector no. i is released. See reference number (`` `ref no'') in the table Components on the ` ``STAMOD result file for connector numbering. The connectors are normally numbered from the first end as 1, 2 etc. following the `FEM` model.
- `ISTEPR: integer, default: 0`: Time step no. for release (nonlinear analysis only)
    - In linear analysis the ball joint connector will be released at the first step
- `LDAMP: integer, default: 0`: Option for calculation of proportional damping matrix in nonlinear analysis.
    - Irrelevant for linear analyses
    - `LDAMP = 0`: Use constant proportional damping matrix calculated at static position
    - `LDAMP = 1`: Use updated proportional damping matrix according to instantaneous mass and stiffness matrices

For non-linear analysis (`ITDMET = 2`, see Method and subsequent input, one input line) `INDHYD` can have the values 1 or 2. Input of 3 will be interpreted as 2. Load iteration for non-linear analysis will always be performed in connection with equilibrium iteration, but not during equilibrium correction.

If load convergence is not obtained after `MAXHIT` iteration, computation will proceed after output of warning.

As a release/rupture analysis is very sensitive, a short time step and rather firm convergence limit is required. If the response of part of the system is not of interest after the release, the Boundary change option may be used to fix the nodes in this part of the system.

Definition of clutch (start up procedure)

7.6.2. Nonlinear step by step integration

This data group is only given for `ITDMET=2` (input group Method and subsequent input, one input line).

7.6.2.1. Data group identifier, one input line

```
NONLinear INTEgration PROCedure
```

7.6.2.2. Specification of incrementation procedure, one input line

```
ITFREQ ISOLIT MAXIT DACCU ICOCOD IVARST ITSTAT CHNORM EACCU
```

- `ITFREQ: integer, default: 1`: Frequency of equilibrium iteration
  - `ITFREQ ⇐ 0`: Iteration will not be performed
  - `ITFREQ >= 1`: Iteration will be performed every `ITFREQ` time step. For steps without iteration equilibrium correction will be performed.
  - The remaining variables in this input line are dummy if `ITFREQ ⇐ 0`
- `ISOLIT: integer, default: 1`: Type of iteration if iteration is to be performed
  - `ISOLIT = 1`: True Newton-Raphson, updating of geometric stiffness from axial force
  - `ISOLIT = 2`: Modified Newton-Raphson iteration
  - Modified Newton-Raphson iteration is not included in the current version of the program
- `MAXIT: integer, default: 10`: Maximum number of iterations for steps with iteration
- `DACCU: real, default:` 10^{-6}: Desired accuracy for equilibrium iteration measured by a modified Euclidean displacement norm (norm of squared translations)
  - Recommended values: $10^{-6} - 10^{-5}$ cfr. `STAMOD` analysis $[1]$

- `ICOCOD: integer, default: 1`: Code for continuation after iteration
  - `ICOCOD = 0`: Computations interrupted if accuracy requirements are not fulfilled
  - `ICOCOD = 1`: Computations continue even if accuracy requirements are not fulfilled. Warning is printed
- `IVARST: integer, default: 0`: Code for automatic subdivision of time step
  - `IVARST = 0`: No subdivision
  - `IVARST > 0`: Automatic subdivision of time step if required accuracy is not obtained with original time step or if incremental rotations are to large.
  - Maximum number of subdivisions: 2^{IVARST}
- `ISTAT: integer, default: 1`: Code for time integration information
  - `ITSTAT = 0`: No information
  - `ITSTAT > 1`: Number of iterations, subdivisions and obtained accuracy are presented
- `CHNORM: character(4), default: DISP`: Convergence norm switch - `= DISP`: Use the default Euclidean displacement norm only - `= BOTH`: Use both the default Euclidean displacement norm and the energy norm
- `EACCU: real, default:` 10^{-6}: Required accuracy measured by energy norm - Dummy if `CHNORM=DISP`

7.6.3. Modification to water kinematics

Modification to water kinematics due to moonpool kinematics may be specified. The water kinematics will be based on the velocities and acceleration of the actual support vessel or floater force model specified.

7.6.3.1. Data group identifier, one input line

```
WATEr KINEmatic CONDition
```

7.6.3.2. Rigid moonpool column, one input line

```
RIGId MOONpool COLumn
```

7.6.3.2.1. Specification of number of moonpools, one input line

```
NLSPEC
```

`NLSPEC: integer`: Number of Rigid Moonpool Columns

7.6.3.2.2. Specification of support vessel moonpool, one input line.

`CHSUPP IVES ZLLOW ZLUP`

- `CHSUPP: character`: Type of support vessel
  - `= VESSEL`: `RIFLEX` support vessel (Prescribed motions)
  - `= FLOATER`: Floater force model
- `IVES: integer`: Support vessel number
- `ZLLOW: real`: Lower Z limit (local vessel system) $[\mathrm{L}]$
- `ZLUP: real`: Upper Z limit (local vessel system) $[\mathrm{L}]$

One input line

7.6.3.2.3. Specification of lines within present moonpool, one input line

`LINE-ID1 LINE-ID2 ....... LINE-IDi .........LINE-IDn`

- `LINE-ID: character(8)`: Line identifiers within moon pool

The data groups `Specification of support vessel moonpool' and `Specification of lines within present moonpool' are to be repeated `NLSPEC` times.

- Rigid moonpool column may not be combined with CHMOT=`NONE': No irregular motions, for irregular wave analysis .
- Rigid moonpool column may not be combined with IMOTD = 0: No motions, for regular wave analysis.
- If current is loaded in static analysis, the current forces will be removed at start of dynamic analysis for lines within moonpool and may create a transient.

7.6.4. Slug force calculations

This data group is only given for `INDINT=2` (Non-linear force model, one input line. Always submit for linear and non-linear analysis), and slug forces can only be specified for single risers.

7.6.4.1. Data group identifier, one input line

Restrictions - The main riser line has to be modelled by beam elements - Consistent formulation (Lumped mass option is prohibited)

Assumptions - The total slug mass is constant, $\mathbf{M_S}$. Initial length is $\mathbf{L_{S0}}$ - The specified velocity refers to the gravity centre of the slug, initially at the half length. - The slug specification is superimposed on the riser mass, including any internal fluid flow. - The internal cross-section area is not used in the slug modelling - The slug length is divided into sections. Initially the sections are of equal length $\mathbf{dl_{S,0}}$. The density, (mass per unit length) is constant within each section. Initially the mass per unit length is $\mathbf{m_0 = M_S/L_{S0}}$

7.6.4.1.1. Input description for slug force specification

```
SLUG FORCe SPECification
```

7.6.4.2. Specification of slug data, one input line

```
TSLUG ICOSLG SLGLEN SLGMAS SLGVEL IDENS IVEL NCYCLE CYCTIM
```

- `TSLUG: real, default: 0`: Time when slug enters first end of main riser line $[\mathrm{T}]$
- `ICOSLG: integer, default: 1`: Interruption parameter
  - `=0`: Analysis termination controlled by slug
  - `=1`: Analysis termination controlled by specified length of simulation (TIME)
- `SLGLEN: real`: Initial slug length $[\mathrm{L}]$
- `SLGMAS: real`: Slug mass $[\mathrm{M}]$
- `SLGVEL: real`: Initial slug velocity $[\mathrm{L/T}]$
- `IDENS: integer, default: 0`: Control parameter density
  - `= 0`: Constant density
  - `= 1`: Variable density with vertical position
- `IVEL: integer, default: 0`: Control parameter velocity
  - `= 0`: Constant velocity
  - `= 1`: Variable velocity
  - The specified velocity refers to the gravity centre of the slug
- `NCYCLE: integer, default: 1`: Number of slug cycles
- `CYCTIM: real`: Slug cycle time (dummy if `NCYCLE = 1`) $[\mathrm{T}]$

7.6.4.2.1. if `IDENS = 1`:

```
Z2 SLGMA2 ZREF
```

- `Z2: real`: Second vertical position where the slug unit mass is specified $[\mathrm{L}]$
- `SLGMA2: real`: Slug unit mass at `Z2` $[\mathrm{M/L}]$
- `ZREF: real < 0`: Reference depth $[\mathrm{L}]$
  - `ZREF` < $\mathrm{Z_{MIN}}$, where $\mathrm{Z_{MIN}}$ is lowest vertical position along the main riser line

The unit mass at a specific z-position is calculated according to the following equation:

$$\mathrm{m(Z_i) = A(Z_i - Z_{REF})^{\alpha}}$$

where - $\alpha = \frac{\mathrm{ln(m_1/m_2)}}{\mathrm{ln(\frac{Z_1 - Z_{REF}}{Z_2 - Z_{REF}})}}$ - $\mathrm{A} = \frac{\mathrm{m_1}}{\mathrm{(Z_1 - Z_{REF})^{\alpha}}}$ - $\mathrm{m_1}$: `SLGMAS/SLGLEN` - $\mathrm{m_2}$: `SLGMA2` - $\mathrm{Z_1}$: Vertical coordinate at inlet, end 1 of main riser line

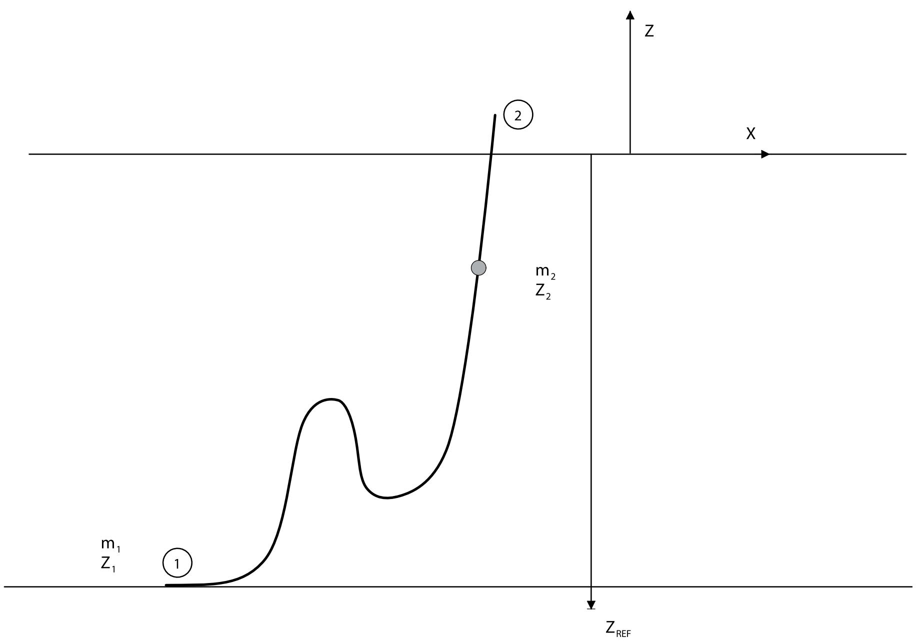

Figure 35. Internal slug flow

7.6.4.2.2. if `IVEL = 1`:

```
DELVEL VEXP
```

- `DELVEL: real`: Velocity specification
- `VEXP: real`: Exponent for velocity

The unit mass at a specific z-position is calculated according to the following equation:

$$V(Z_i) = V_1 - \Delta V|Z_i - Z_1|^\alpha \quad \text{for} \quad (Z_i - Z_1) >= 0$$

$$V(Z_i) = V_1 + \Delta V|Z_i - Z_1|^\alpha \quad \text{for} \quad (Z_i - Z_1) < 0$$

Where: - V_1: Initial slug velocity (Velocity at inlet) - ΔV: **DELVEL** - Z_i: Vertical coordinate at inlet, end 1 of main riser line - α: **VEXP**

7.6.5. Import of internal flow data from file

This data group is only given for `INDINT=2` (see Non-linear force model, one input line. Always submit for linear and non-linear analysis)

7.6.5.1. Data group identifier, 1 input line

```
IMPOrt FLOW DATA
```

7.6.5.2. Specification of input flow file, one input line

```
IMRL CHOPAD CHFFLW
```

- `IMRL: integer, default: 0`: Main riser line number
  - `= 0`: All lines
- `CHOPAD: character(4), default: REPL`: Fluid contents option
  - `= REPL`: Specified flow replaces that given in the Main Riser Line definition
  - `= ADDI`: Specified flow is in addition to that given in the Main Riser Line definition
- `CHFFLW: character(70)`: Name of flow data file

The flow input file is described in See Internal flow description

7.6.6. Dynamic current variation

Available for nonlinear dynamic analysis, but only when the current profile is specified explicitly on the `INPMOD` input file. This means that this data group cannot be given for coupled analysis or when the current is specified on a `CURMOD` input file. However, dynamic current conditions can alternatively be specified using `CURMOD`.

Varying current velocity and direction are specified at the current levels defined in the preceding static analysis. The varying current is to be described in a separate file. For description of the file format, confer chapter Description of Additional Input Files: Dynamic Current Variation.

7.6.6.1. Data group identifier, one input line

```
DYNAmic CURRent VARIation
```

7.6.6.2. File name

```
CHFCUR
```

- `CHFCUR: character(80)`: File name with current velocity and direction

ASCII file containing current velocity and direction at specified time instants. The velocity and directions have to be given at all levels defined in the preceding static analysis.

7.6.7. Dynamic nodal forces

This data group enables the user to specify additional dynamic nodal force components. The force components may either be described by simple functions or read from a separate input file. For file description, see chapter Description of Additional Input Files: Dynamic Nodal Forces.

7.6.7.1. Data group identifier, one input line

```
DYNAmic NODAl FORCes
```

7.6.7.2. Number of specified components specified by functions or by time series on file

```
NDCOMP CINPUT CHFLOA
```

- `NDCOMP: integer`: Number of load components to be specified
- `CINPUT: character(6), default: 'NOFILE'`: Type of force specification
    - `CINPUT = NOFILE`: Forces described by simple expression
    - `CINPUT = FILE`: Forces described by time series on file
- `CHFLOA: character(80)`: File name for time series of force components.
    - Dummy if `CINPUT = NOFILE`

7.6.7.3. Force component description

```
LINE-ID ILSEG ILNOD ILDOF CHICOO IFORTY TIMEON TIMEOF P1 P2 P3
```

- `LINE-ID: character(8)`: Line identifier
- `ILSEG: integer`: Segment number within actual line

- `ILNOD: integer`: Local node/element number within segment
- `ILDOF: integer`: Degree of freedom within the specified node/element
  - `ILDOF = 7…12` at end 2 of an element
- `CHICOO: character(6)`: Coordinate system code
  - `CHICOO = GLOBAL`: Force component refers to global system, unless the node has skew or vessel boundaries. If the node has skew or vessel boundaries, `CHICOO=GLOBAL` means that the load component acts in the skew (vessel) system. The force is applied at the specified node.
  - `CHICOO = LOCAL`: Force component refers to local system. The force is applied to the specified element.
- `IFORTY: integer`: Force component type
  - `IFORTY = 1`: Constant force
  - `IFORTY = 2`: Harmonic force
  - `IFORTY = 3`: Ramp
- `TIMEON: real`: Time for switching component on
- `TIMOFF: real`: Time for switching component off
- `P1: real`: Force component parameter
  - `IFORTY = 1`: Magnitude, $[\mathrm{F}, \mathrm{FL}]$
  - `IFORTY = 2`: Amplitude, $[\mathrm{F}, \mathrm{FL}]$
  - `IFORTY = 3`: Force derivative, $[\mathrm{F}/\mathrm{T}, \mathrm{FL}/\mathrm{T}]$
- `P2: real`: Force component parameter
  - `IFORTY = 1`: Dummy
  - `IFORTY = 2`: Period $[\mathrm{T}]$
  - `IFORTY = 3`: Dummy
- `P3: real`: Force component parameter
  - `IFORTY = 1`: Dummy
  - `IFORTY = 2`: Phase $[\mathrm{deg}]$

- `IFORTY = 3`: Dummy

`IFORTY, TIMEON, TIMEOFF, P1, P2` and `P3` are dummy for `CINPUT = FILE`, time series on file. For file description, see chapter Description of Additional Input Files: Dynamic Nodal Forces.

For simulation time, t, `TIMEON` ⇐ t ⇐ `TIMOFF` the force component (F) will be applied as: - `IFORTY = 1`: $\mathrm{F} = \mathrm{P1}$ - `IFORTY = 2`: $\mathrm{F} = \mathrm{P1} \times \sin(\frac{2\pi}{\mathrm{P2}} \times (\mathrm{t} - \mathrm{TIMEON}) + \mathrm{P3}\frac{\pi}{180})$ - `IFORTY = 3`: $\mathrm{F} = \mathrm{P1} \times (\mathrm{t} - \mathrm{TIMEON})$

7.6.8. Dynamic tension variation

7.6.8.1. Data group identifier, one input line

```
DYNAmic TENSion VARIation
```

7.6.8.2. Specification of dynamic tension variation

```
SNOD-ID TCX TCV TCA IOPDTV
```

- `SNOD-ID: character(8)`: Supernode identifier for dynamic tension variation.
  - Must be identical to the last node-id in stroke storage specification if stroke storage is specified.
- `TCX: real, default: 0`: Coefficient for tension variation due to relative displacement between vessel and riser $[\mathrm{F}/\mathrm{L}]$
- `TCV: real, default: 0`: Coefficient for tension variation due to relative velocity between vessel and riser $[\mathrm{FT}/\mathrm{L}]$
- `TCA: real, default: 0`: Coefficient for tension variation due to relative acceleration between vessel and riser $[\mathrm{FT}^2/\mathrm{L}]$
- `IOPDTV: integer, default: 0`: Option for updating tension during iterations (relevant for nonlinear time domain analysis only):
  - `= 0`: Not updated
  - `= 1`: Updated

The resulting dynamic tension is given by:

$$\Delta\mathrm{T} = \mathrm{TCX} \times \mathrm{x} + \mathrm{TCV} \times \dot{\mathrm{x}} + \mathrm{TCA} \times \ddot{\mathrm{x}}$$

where x is the relative vertical displacement between the vessel and the riser. The vertical riser displacements are directly available in a nonlinear time domain analysis. In a linear analysis, the vertical displacements are estimated from the displacements along lines `ILIN1` …. `ILINN` (as in linear stroke calculations). File storage for stroke response must be given if specification of dynamic tension variation is included. In both linear and nonlinear analyses platform motions will be modified for platform setdown if `SETLEN > 0` in File storage for stroke response.

7.6.9. Time domain loading

7.6.9.1. Data group identifier, one input line

```
TIME DOMain LOADing
```

7.6.9.2. Load type to be activated, one input line

```
LOTYPE NLSPEC CINPUT CHFLOA IFORM
```

- `LOTYPE: character:`
  - `= SEGV`: Segment length variation (Nonlinear analysis only)
  - `= TEMP`: Temperature variation (Nonlinear analysis only)
  - `= PRES`: Pressure variation (Nonlinear analysis only)
  - `= BOUN`: Boundary change (Nonlinear analysis only)
  - `= VIVA`: Harmonic loads from VIVANA (Nonlinear analysis only)
  - `= WINC`: Winch run (Nonlinear analysis only)
  - `= WIND`: Wind event. Only available for `IWITYP=14`, Stationary uniform wind with shear.
  - `= SHUT`: Wind turbine shutdown fault options (Nonlinear analysis only)
  - `= BLAD`: Wind turbine blade pitch fault options (Nonlinear analysis only)
- `NLSPEC: integer, default: See below`: Number of load specification to follow
- `CINPUT: character, default: 'NOFILE':`
  - `= NOFILE`: All load specification given below
  - `= FILE`: Load specification read from file `CHFLOA`
- `CHFLOA: character, default: See below`: Load specification file.
  - Dummy for `CINPUT = NOFILE`
- `IFORM: integer, default: 1`: File format

For `LOTYPE = VIVA`: - `NLSPEC = 1, CINPUT=FILE` and `IFORM=1` - The default value of `CHFLOA` is `<prefix>_ifnviv.ffi`

For `LOTYPE = WIND`: - `NLSPEC = 1, CINPUT=NOFILE`

For `LOTYPE = SHUT`: - `NLSPEC = 1, CINPUT=NOFILE`

For `LOTYPE = BLAD`: - `NLSPEC = 1, CINPUT=NOFILE`

7.6.9.3. Segment length variation, NLSPEC input lines for LOTYPE = SEGV

`LINE-ID ISEG TBEG TENO SLRATE`

- `LINE-ID: character(8)`: Line identifier
- `ISEG: integer`: Local segment within line `LINE-ID`
- `TBEG: real`: Start time for segment length variation $[\mathrm{T}]$
- `TEND: real`: End time for segment length variation $[\mathrm{T}]$
  - `TEND > TBEG`
- `SLRATE: real`: Segment length variation per time unit $[\mathrm{L}/\mathrm{T}]$

7.6.9.4. Temperature variation, NLSPEC input lines if LOTYPE = TEMP

`LINE-ID ISEG IEL TBEG TEND TEMP`

- `LINE-ID: character(8)`: Line identifier
- `ISEG: integer/character`: Local segment number within line `LINE-ID`
  - `= 0 / ALL`: All segments in specified line
- `IEL: integer/character`: Local element number within segment `ISEG`
  - `= 0 / ALL`: All elements in specified segment
- `TBEG: real`: Start time for temperature variation $[\mathrm{T}]$
- `TEND: real`: End time for temperature variation $[\mathrm{T}]$
  - `TEND > TBEG`
- `TEMP: real`: Temperature at end of temperature variation

The temperature is varied linearly during the load group from the starting temperature ending with the temperature specified here.

A linear variation of temperature over a sequence of elements may be specified by giving a negative element number at the second end of the linear variation.

7.6.9.5. Pressure variation, NLSPEC input lines if LOTYPE = PRES

`MRL-ID TBEG TEND PRESSI DPRESS VVELI`

- `MRL-ID: character(8)`: Reference to Main Riser Line identifier
- `TBEG: real`: Start time for pressure variation $[\mathrm{T}]$
- `TEND: real`: End time for pressure variation $[\mathrm{T}]$
    - `TEND > TBEG`
- `PRESSI: real, default: 0`: Final pressure at inlet end $[\mathrm{F}/\mathrm{L}^2]$
- `DPRESS: real, default: 0`: Final pressure drop $[\mathrm{F}/\mathrm{L}^3]$
- `VVELI: real, default: 0`: Final fluid velocity $[\mathrm{L}^3/\mathrm{T}]$
    - Dummy in present version

7.6.9.6. Boundary change, 3 x NLSPEC input lines for LOTYPE = BOUN

7.6.9.6.1. Time for boundary change

`TIMCHG`

- `TIMCHG: real`: Time for boundary change $[\mathrm{T}]$

7.6.9.6.2. Identification of node for boundary change

`IREF-ID ILSEG ILNODE IOP`

- `IREF-ID: character(8)`: Reference to line or supernode identifier.
- `ILSEG: integer`:
    - If `IREF-ID` refers to a line, `ILSEG` is the segment number within this line
    - If `IREF-ID` refers to a supernode, `ILSEG` must be zero
- `ILNODE: integer`:

- If `IREF-ID` refers to a line, `ILNODE` is the node number within segment `ILSEG`
- If `IREF-ID` refers to a supernode, `ILNODE` must be zero

- `IOP: integer`: Parameter for boundary change option
  - `= 0`: Boundary conditions: fixed, pre-scribed or free
  - `= -1`: Boundary conditions: rigid node connection (The node will become a slave node.)

Ordinary (line end) supernodes and SIMO body nodes with `CHLOCA_OPT='POSI'` may have boundary change.

7.6.9.6.3. Status for nodal degrees of freedom if `IOP = 0`

```
IPOS IX IY IZ IRX IRY IRZ
```

- `IPOS: integer`: Boundary condition type
  - `IPOS = 0`: The node is fixed in global system
  - `IPOS = N`: The node is attached to support vessel no N
- `IX: integer`: Boundary condition code for translation in X-direction
  - `IX = 0`: Free
  - `IX = 1`: Fixed of prescribed
- `IY: integer`: Boundary condition code for translation in Y-direction
  - Same interpretation as for `IX`.
- `IZ: integer`: Boundary condition code for translation in Z-direction
  - Same interpretation as for `IX`.
- `IRX: integer`: Boundary condition code for rotation around X-direction
  - Same interpretation as for `IX`.
- `IRY: integer`: Boundary condition code for rotation around Y-direction
  - Same interpretation as for `IX`.
- `IRZ: integer`: Boundary condition code for rotation around Z-direction
  - Same interpretation as for `IX`.

7.6.9.6.4. Identification of master node if `IOP = 1`

`LINE-ID ILSEG ILNODE`

- `LINE-ID: character(8)`: Line identifier
- `ILSEG: integer`: Segment number within the actual line
- `ILNODE: integer`: Local node number within segment

7.6.9.7. Specification of harmonic loads from VIVANA, one input line for LOTYPE = VIVA

`CHFRQ ALIM ISEED TPLOT`

- `CHFRQ: character, default: DOMI`:
  - `= ALL`: All responses frequencies from `VIVANA` included
  - `= AMIN`: Response frequencies with normalized cross-flow response larger than `AMIN` included
  - `= DOMI`: Only the dominating response frequency included
- `ALIM: real, default: 0`: Cross-flow displacement to diameter ratio $[1]$
  - Dummy for `CHFRQ` $\neq$ `ALIM`
- `ISEED: integer, default: 280495`: Seed
- `TPLOT: real, default: 2`: VIV response plot interval. Key VIV results from the last `TPLOT` interval of the simulation are stored on the `_dynmod.mpf` file
  - `TPLOT > 0`: Given as number of whole response periods
  - `TPLOT < 0`: Given as time $[\mathrm{T}]$

7.6.9.8. Winch run, NLSPEC input lines for LOTYPE = WINC

`IWINCH TBEG TEND WIVEL`

- `IWINCH: integer`: Winch number
- `TBEG: real`: Start time for winch run $[\mathrm{T}]$
- `TEND: real`: End time for winch run $[\mathrm{T}]$
- `WIVEL: real`: Winch velocity $[\mathrm{L}/\mathrm{T}]$

- `WIVEL > 0`: Winching out, i.e. the winch run will increase the active line length.

7.6.9.9. Wind event specification, two or three input lines for LOTYPE = WIND

In the following IEC 2005 refers to the standard ``IEC 61400-1 Wind turbines – Part 1: Design requirements – 2005".

An IEC 2005 extreme wind event may only be applied to a stationary uniform wind with shear, `IWITYP=14`.

7.6.9.9.1. Start time and wind turbine reference

```
TIME WIND-TURBINE-ID
```

- `TIME: real`: Start time for wind event $[\mathrm{T}]$
- `WIND-TURBINE-ID: character(8)`: Wind turbine identifier given in `INPMOD`. `NONE` may be given to skip the wind turbine reference for the events ECD, EOG and EDC with `CLASS = NONE`; i.e. detailed specification of event.

7.6.9.9.2. Extreme wind event

```
CHEVEN CLASS CHDIR
```

- `CHEVEN: character(12)`: Extreme wind event. The following values are currently available:
  - `= IEC2005_ECD`: IEC 2005 extreme coherent gust with direction change
  - `= IEC2005_EWSV`: IEC 2005 extreme vertical wind shear
  - `= IEC2005_EWSH`: IEC 2005 extreme horizontal wind shear
  - `= IEC2005_EOG`: IEC 2005 extreme operating gust
  - `= IEC2005_EDC`: IEC 2005 extreme direction change
- `CLASS: character(4)`: Wind turbine class, ref IEC 2005. Legal values are `IA`, `IIA`, `IIIA`, `IB`, `IIB`, `IIIB`, `IC`, `IIC`, `IIIC`, `S` or `NONE`, detailed specification of event parameters.
- `CHDIR: character(4)`: Direction of event. Dummy for `CHEVEN = IEC2005_EOG`.
  - `= POS`: For ECD and EDC, the wind shifts clockwise (viewed from above). For EWSV, the wind increases at the top of the rotor disk and decrease at the bottom. For EWSH, the wind increases on the left side of the rotor disk and decrease on the right side when viewed along the shaft from the hub.

    - `= NEG`: For ECD and EDC, the wind shifts counter-clockwise (viewed from above). For EWSV, the wind decreases at the top of the rotor disk and increases at the bottom. For EWSH, the wind decreases on the left side of the rotor disk and increases on the right side when viewed along the shaft from the hub.

    - `= NONE`: Only allowed for `CHEVEN = IEC2005_EOG`.

7.6.9.9.3. Additional input for wind turbine class S

If `CLASS = S`, the following additional input line is given:

`VREF IREF`

- `VREF: real`: Reference wind speed average over 10 min $[\mathrm{L}/\mathrm{T}]$

- `IREF: real`: Expected value of the turbulence intensity at 15 m/s $[1]$

7.6.9.9.4. Detailed specification of IEC2005 ECD event

If `CLASS = NONE` and `CHEVEN = IEC2005_ECD`, the following additional input line is given:

`VEL_EVENT DIR_EVENT TIME_EVENT`

- `VEL_EVENT: real, default: 0.0`: Velocity change $[\mathrm{L}/\mathrm{T}]$

- `DIR_EVENT: real, default: 0.0`: Direction change $[\mathrm{deg}]$

- `TIME_EVENT: real > 0`: Duration of event $[\mathrm{T}]$

7.6.9.9.5. Detailed specification of IEC2005 EWSV or EWSH event

If `CLASS = NONE` and `CHEVEN = IEC2005_EWSV or IEC2005_EWSH`, the following additional input line is given:

`VEL_EVENT TIME_EVENT`

- `VEL_EVENT: real, default: 0.0`: Maximum velocity change at edge of rotor $[\mathrm{L}/\mathrm{T}]$

- `TIME_EVENT: real > 0`: Duration of event $[\mathrm{T}]$

7.6.9.9.6. Detailed specification of IEC2005 EOG event

If `CLASS = NONE` and `CHEVEN = IEC2005_EOG`, the following additional input line is given:

`VEL_EVENT TIME_EVENT`

-

`VEL_EVENT: real, default: 0.0`: Range of velocity from minimum to maximum during the event $[\mathrm{L}/\mathrm{T}]$

- `TIME_EVENT: real > 0`: Duration of event $[\mathrm{T}]$

7.6.9.9.7. Detailed specification of IEC2005 EDC event

If `CLASS = NONE` and `CHEVEN = IEC2005_EDC`, the following additional input line is given:

```
DIR_EVENT TIME_EVENT
```

- `DIR_EVENT: real, default: 0.0`: Direction change $[\mathrm{deg}]$
- `TIME_EVENT: real > 0`: Duration of event $[\mathrm{T}]$

7.6.9.10. Wind turbine shutdown fault options

The specifications given for turbine shutdown will overrule commanded blade pitch and torque, given by the wind turbine control system. Wind turbine blade pitch faults will override the wind turbine shutdown options.

7.6.9.10.1. Start time and wind turbine reference

```
TSTART WIND-TURBINE-ID
```

- `TSTART: real`: Start time for shutdown $[\mathrm{T}]$
- `WIND-TURBINE-ID: character(8)`: Reference to wind turbine identifier

7.6.9.10.2. Number of pairs in rate of change in pitch and maximum pitch

```
NPAIR
```

- `NPAIR: integer`: Number of pairs in tabulated rate of pitch change and maximum pitch at the rate of pitch change

7.6.9.10.3. Rate of change in pitch and maximum pitch at the rate of change in pitch, NPAIR input lines

```
RATE  MAX_PITCH
```

- `RATE: real`: Rate of change in pitch angle (absolute value) $[\mathrm{deg}/\mathrm{T}]$
  - `RATE > 0`
- `MAX_PITCH: real`: Maximum pitch angle for the rate of change in pitch $[\mathrm{deg}]$

```
MAXPITCH > 0
```

`MAX_PITCH` values must be given in increasing order.

Example:

Type of shutdown	Pitch change rate	Maximum pitch
normal	1.0 deg/T to	90.0 deg

Example:

Type of shutdown	Pitch change rate	Maximum pitch
open-loop	8.0 deg/T to	40.0 deg
-	4.0 deg/T to	90.0 deg

Example:

Type of shutdown	Pitch change rate	Maximum pitch
emergency	8.0 deg/T to	90.0 deg

7.6.9.10.4. Generator torque fault options

`CHFAULT`

- `CHFAULT: character(6)`:
    - `= NONE`: No generator torque fault, the calculated generator torque will be applied in full
    - `= LOSS`: Total loss of generator torque
    - `= BACKUP`: Backup power, generator torque will follow scaled torque control

======= Scale factor for generator torque, One input line for `CHFAULT = BACKUP`

`SF`

- `SF: real`: Scale factor for generator torque

- SF >= 0

7.6.9.10.5. Mechanical brake option

`CHBRAKE`

- `CHBRAKE: character(6)`:
  - `= NONE`: No mechanical brake
  - `= BRAKE`: Mechanical brake (Linear damping)

======= Torque damping coefficient and brake uploading duration, One input line for `CHBRAKE = BRAKE`

`TORQUE_DAMP UPL_DURATION`

- `TORQUE_DAMP: real`: Linear torque damping coefficient $[\mathrm{FLT/deg}]$
  - `TORQUE_DAMP >= 0`
- `UPL_DURATION: real`: Brake uploading duration to full braking torque $[\mathrm{T}]$
  - `UPL_DURATION >= 0`

7.6.9.11. Wind turbine blade pitch fault options

Wind turbine blade pitch faults will override commanded blade pitch given by the wind turbine control system or by the wind turbine shutdown options.

7.6.9.11.1. Wind turbine reference

`WIND-TURBINE-ID`

- `WIND-TURBINE-ID: character(8)`: Reference to wind turbine identifier

7.6.9.11.2. Number of blades for fault specification

`NBL_FAULT`

- `NBL_FAULT: integer`: Number of blades for fault specification
  - `NBL_FAULT >= 0`

The subsequent input specification must be given per blade with pitch fault

7.6.9.11.3. Start time and line (foil blade) reference for fault specification

`TSTART LINE-ID`

- `TSTART: real`: Start time for blade pitch fault $[\mathrm{T}]$
- `LINE-ID: character(8)`: Reference to line identifier

7.6.9.11.4. Type of blade pitch fault

`CHFAULT`

- `CHFAULT: character(4)`:
  - `= SEIZ`: Seized - Fixed pitch from time of occurrence
  - `= RUNA`: Runaway - Pitch change rate from time of occurrence to final pitch
  - `= BIAS`: Actuator bias - Fixed pitch fault from time of occurrence

======= Rate of change in pitch and final pitch, One input line for `CHFAULT = RUNA`

`RATE  FINAL_PITCH`

- `RATE: real`: Rate of change in pitch (absolute value) $[\mathrm{deg}/\mathrm{T}]$
  - `RATE >= 0`
- `FINAL_PITCH: real`: Final pitch $[\mathrm{deg}]$

======= Pitch deviation from required pitch, One input line for `CHFAULT = BIAS`

`DEL_PITCH UPL_DURATION`

- `DEL_PITCH: real`: Fixed pitch deviation from required pitch $[\mathrm{deg}]$
- `UPL_DURATION: real`: Bias uploading duration to full pitch deviation $[\mathrm{T}]$
  - `UPL_DURATION >= 0`

7.6.10. File storage of displacement response

Before specifying file storage of response, note that meaningful output from `OUTMOD` can be dependent on which and how much information that is stored on file from `DYNMOD`. Examples of such output options in `OUTMOD` are time series of element angles and distance between elements, see Element angle time series from time domain analysis and Distance time series calculated from the time domain analyses.

There are limitations in storage capacity due to: - Disk/user size - Maximum number of arrays that may be stored on the `ifndyn.ffi` file. A message is printed if this limit is exceeded. The maximum number of arrays on the file may be changed using the environmental variable RIFLEX_MAXDYN_IFNDYN. The minimum value is 50000 and the maximum is 2000000. The default is 200000.

7.6.10.1. Data group identifier, one input line

```
DISPlacement RESPonse STORage
```

7.6.10.2. Specification of displacements to be stored

7.6.10.2.1. Amount of storage, one input line

```
IDISP NODISP IDISFM
```

- `IDISP: integer`: Code for storage of nodal displacements. Storage for every `IDISP` time step (`IDISP=2` gives storage for every second step).
- `NODISP: integer > 0`: Number of input lines given specifying node numbers where displacements are stored.
- `IDISFM: integer, default: 0: integer`: Format code for storage and/or output of nodal displacements.
    - `IDISFM = 0`: Storage only on `ifndyn` file.
    - `IDISFM = 1`: Storage on `ifndyn` file and additional file in ASCII format.
    - `IDISFM = 2`: Storage on `ifndyn` file and additional file in BINARY format.
    - `IDISFM = -1`: Storage only on additional file in ASCII format. Results are not available in `OUTMOD`.
    - `IDISFM = -2`: Storage only on additional file in BINARY format. Results are not available in `OUTMOD`.

Note that data must be stored on the `ifndyn` file in order to be available for `OUTMOD`.

If `IDISFM`$\neq$ `0` is specified, an additional result file and a key file will be created. The file names will be based on the name of the `DYNMOD` result file; `<prefix>_dynmod.res`. An additional ASCII file will be `<prefix>_noddis.asc` and an additional binary file will be `<prefix>_noddis.bin`. The key file `key_<prefix>_noddis.txt` will describe how data is stored on the additional output file. The key file may be viewed in a text editor.

7.6.10.2.2. Specification of nodes for displacement storage, NODISP input lines

`LINE-ID ISEG INOD`

- `LINE-ID: character(8)`: Line identifier
- `ISEG: integer`: Segment number of line
- `INOD: integer/character`: Local node number on actual segment

Consecutively numbered nodes may be specified implicitly by assigning a negative value to the last of two adjacent `INOD`. In this case `LINE-ID` and `ISEG` must be the same for the two nodes.

All nodes within one segment may be specified by simply giving `ALL` as input to `INOD`.

7.6.11. File storage for internal forces

7.6.11.1. Data group identifier, one input line

`FORCe RESPonse STORage`

7.6.11.2. Specification of forces to be stored

7.6.11.2.1. Amount of storage, one input line

`IFOR NOFORC IFORFM IELTFM`

- `IFOR: integer`: Code for file storage of internal forces. Forces are stored for every `IFOR` time step. (`IFOR=3` gives storage for every third step)
- `NOFORC: integer > 0`: Number of input lines given to specify elements for which forces are stored.
- `IFORFM: integer, default: 0`: Format code for storage and / or output of element forces
  - `IFORFM = 0`: Storage only on `ifndyn` file
  - `IFORFM = 1`: Storage on `ifndyn` file and additional file in ASCII format
  - `IFORFM = 2`: Storage on `ifndyn` file and additional file in BINARY format
  - `IFORFM = -1`: Storage only on additional file in ASCII format. Results are not available in `OUTMOD`.
  - `IFORFM = -2`: Storage only on additional file in BINARY format. Results are not available in `OUTMOD`.
- `IELTFM: integer, default: 0`: Format code for or output of element transformation matrices

  - IELTFM = 0: No output
  - IELTFM = ±1: Output on additional file in ASCII format
  - IELTFM = ±2: Output on additional file in BINARY format

Note that data must be stored on the ifndyn file in order to be available for OUTMOD.

If IFORFM≠ 0 is specified, an additional result file and a key file will be created. The file names will be based on the name of the DYNMOD result file; <prefix>_dynmod.res. An additional ASCII file will be <prefix>_elmfor.asc and an additional binary file will be <prefix>_elmfor.bin. The key file key_<prefix>_elmfor.txt will describe how data is stored on the additional output file. The key file may be viewed in a text editor.

For nonlinear analysis with pipe-in-pipe elements, the contact forces will be written to <prefix>_cntfor.asc or .bin if IFORFM≠ 0. The contents are described on the corresponding key file key_<prefix>_cntfor.txt.

Roller contact forces will be stored on separate result files if IDCOM = LAYFLX, see INPMOD: Selection of riser type and identifier .

If IELTFM≠ 0 is specified, an additional result file and a key file will be created. The file names will be based on the name of the DYNMOD result file; <prefix>_dynmod.res. An additional ASCII file will be <prefix>_elmtra.asc and an additional binary file will be <prefix>_elmtra.bin. The key file key_<prefix>_elmtra.txt will describe how data is stored on the additional output file. The key file may be viewed in a text editor.

7.6.11.2.2. Specification of elements for force storage, NOFORC input lines

LINE-ID ISEG IEL

- LINE-ID: character(8): Line identifier
- ISEG: integer: Local segment number within line LINE-ID
- IEL: integer/character: Local element number within segment ISEG
  - = ALL: All elements in specified segment

Consecutively numbered elements may be specified implicitly by assigning a negative value to the last of two adjacent elements, IEL. In this case LINE-ID and ISEG must be the same for the two elements.

All elements within one segment may be specified by simply giving ALL as input to IEL.

7.6.12. File storage for curvature response

Curvature estimates based on nodal displacements may be generated by `OUTMOD` (see Curvature time series calculated from dynamic nodal displacements) even though curvatures are not stored from `DYNMOD`.

7.6.12.1. Data group identifier, one input line

```
CURVature RESPonse STORage
```

7.6.12.2. Specification of curvature to be stored

7.6.12.2.1. Amount of storage, one input line

```
ICURV NOCURV ICURFM
```

- `ICURV: integer`: Code for storage of curvature response. Curvature is stored for every `ICURV` time step
- `NOCURV: integer > 0`: Number of input lines given to specify elements for which curvatures are stored.
- `ICURFM: integer, default: 0: integer`: Format code for storage and/or output of element curvature.
  - `ICURFM = 0`: Storage only on `ifndyn` file.
  - `ICURFM = 1`: Storage on `ifndyn` file and additional file in ASCII format.
  - `ICURFM = 2`: Storage on `ifndyn` file and additional file in BINARY format.
  - `ICURFM = -1`: Storage only on additional file in ASCII format. Results are not available in `OUTMOD`.
  - `ICURFM = -2`: Storage only on additional file in BINARY format. Results are not available in `OUTMOD`.

Note that data must be stored on the `ifndyn` file in order to be available for `OUTMOD`.

If `ICURFM`$\neq$ `0` is specified, an additional result file and a key file will be created. The file names will be based on the name of the `DYNMOD` result file; `<prefix>_dynmod.res`. An additional ASCII file will be `<prefix>_elmcur.asc` and an additional binary file will be `<prefix>_elmcur.bin`. The key file `key_<prefix>_elmcur.txt` will describe how data is stored on the additional output file. The key file may be viewed in a text editor.

7.6.12.2.2. Specification of elements for curvature storage, NOCURV input lines

```
LINE-ID ISEG IEL
```

`LINE-ID: character(8)`: Line identifier

- `ISEG: integer`: Local segment number within line `LINE-ID`
- `IEL: integer/character`: Local element number within segment `ISEG`
  - `= ALL`: All elements in specified segment

Consecutively numbered elements may be specified implicitly by assigning a negative value to the last of two adjacent elements, `IEL`. In this case `LINE-ID` and `ISEG` must be the same for the two elements.

All elements within one segment may be specified by simply giving `ALL` as input to `IEL`.

7.6.13. Envelope curve specification

This data group enables the user to compute envelopes from both regular and irregular analysis. For irregular analysis mean and standard deviation of response will be printed on the _dynmod.res file.

7.6.13.1. Data group identifier, one line

```
ENVElope CURVe SPECification
```

7.6.13.2. Specification

7.6.13.2.1. Options for calculation and printing

```
IENVD IENVF IENVC TENVS TENVE NPREND NPRENF NPRENC IFILMP
```

- `IENVD: integer, default: 1`: Calculation option for displacement envelopes
  - `= 0`: not calculated
  - `= 1`: calculated
- `IENVF: integer, default: 1`: Calculation option for force envelopes
  - `= 0`: not calculated
  - `= 1`: calculated
- `IENVC: integer, default: 1`: Calculation option for curvature envelopes
  - `= 0`: not calculated
  - `= 1`: calculated

- TENVS: real: Simulation start time for calculating envelopes $[\mathrm{T}]$
- TENVE: real, default: 10^6: Simulation end time for calculating envelopes $[\mathrm{T}]$
- NPREND: integer, default: 0: Print option for displacement envelopes
  - = 0: Not printed
  - = 1: print
- NPRENF: integer, default: 0: Print option for force envelopes
  - = 0: not printed
  - = 1: print
- NPRENC: integer, default: 0: Print option for curvature envelopes
  - = 0: not printed
  - = 1: print
- IFILMP: integer, default: 2: MatrixPlot file option; specifies amount of results written to the file <prefix>_dynmod.mpf. 0 ⇐ IFILMP ⇐ 4.
  - = 0: No print
  - = 1: Minimum values, maximum values and standard deviations
  - = 2: Minimum values, maximum values and standard deviations (identical to specifying IFILMP = 1)
  - = 3: Minimum values, maximum values, standard deviations, mean values and mean-crossing periods
  - = 4: Minimum values, maximum values, standard deviations, mean values, mean-crossing periods, skewness and kurtosis

Note that the mean-crossing period, skewness and kurtosis will be inaccurate for time series with constant or near constant values.

7.6.14. File storage for stroke response

The stroke is stored for presentation and / or post-processing in OUTMOD

7.6.14.1. Data group identifier, one input line

```
STROke RESPonse STORage
```

7.6.14.2. Specification of stroke calculation and storage

```
ISTRO SNOD-ID IOPSTR SETLEN XRSTRO YRSTRO NLINST LINE-ID1 .. LINE-IDnlinst
```

- `ISTRO: integer, default: 1`: Code for storage of stroke response. Storage for every `ISTRO` time step (`ISTRO=2` gives storage for every second step)
- `SNOD-ID: character(8)`: Supernode identifier for stroke calculation
- `IOPSTR: integer, default: 0`: Option for reference coordinates
  - `= 0`: Initial stressfree configuration used as reference
  - `= 1`: Final static configuration used as reference (under implementation)
- `SETLEN: real, default: 0`: Tendon length for set-down correction
- `XRSTRO: real, default: 0`: Global X coordinate of node `INODST`'s reference point for set-down calculations.
  - Dummy of `SETLEN = 0`.
- `YRSTRO: real, default: 0`: Global Y coordinate of node `INODST`'s reference point for set-down calculations.
  - Dummy of `SETLEN = 0`.
- `NLINST: integer, default: 0`: Number of lines used in calculating stroke
  - Dummy for nonlinear analysis
- Lines (line identifiers) used in stroke calculation
  - Dummy for nonlinear analysis
  - `LINE-ID1: character(8)`:
  - .
  - .
  - .
  - `LINE-IDnlinst: character(8)`:

Stroke may only be calculated for supernodes. No set-down correction if `SETLEN = 0.0`

7.6.15. File storage for sum forces

The element sum forces are the sum of the stiffness, damping and inertia forces. The sum force in the local axial direction will be stored for each specified element.

7.6.15.1. Data group identifier, one input line

```
SUMFORCe RESPonse STORage
```

7.6.15.2. Specification of forces to be stored

7.6.15.2.1. Amount of storage, one input line

```
ISFOR NOSFOR ISFOFM
```

- `ISFOR: integer`: Code for file storage of sum forces. Forces are stored for every `ISFOR` time step (`ISFOR=3` gives storage for every third step)
- `NOSFOR: integer > 0`: Number of input lines given to specify elements for which sum forces are stored.
- `ISFORM: integer, default: -1: integer`: Format code for storage and/or output of sum element forces.
    - `ISFORM = -1`: Storage on additional file in ASCII format only.
    - `ISFORM = -2`: Storage on additional file in BINARY format only.

This data group is available for nonlinear time domain analysis only.

If `ISFORM`$\neq$ `0` is specified, an additional result file and a key file will be created. The file names will be based on the name of the `DYNMOD` result file; `<prefix>_dynmod.res`. An additional ASCII file will be `<prefix>_elmsfo.asc` and an additional binary file will be `<prefix>_elmsfo.bin`. The key file `key_<prefix>_elmsfo.txt` will describe how data is stored on the additional output file. The key file may be viewed in a text editor.

7.6.15.2.2. Specification of elements for force storage, NOSFOR input lines

```
LINE-ID ISEG IEL
```

- `LINE-ID: character(8)`: Line identifier
- `ISEG: integer`: Local segment number within line `LINE-ID`
- `IEL: integer/character`: Local element number within segment `ISEG`
    - `= ALL`: All elements in specified segment

Consecutively numbered elements may be specified implicitly by assigning a negative value to the last of two adjacent elements, `IEL`. In this case `LINE-ID` and `ISEG` must be the same for the two elements.

All elements within one segment may be specified by simply giving `ALL` as input to `IEL`.

7.6.16. File storage for wind turbine responses

This option enables export of wind turnine key responses to file in binary or ASCII format

7.6.16.1. Data group identifier, one input line

```
TURBine RESPonse STORage
```

7.6.16.1.1. Time interval for storage, one input line

```
DT_WTR
```

- `DT_WTR: real`: Desired time interval for storage [T]
  - `DT_WTR = 0`: Storage at each simulation time step

Note that `DT_WTR` will be adjusted to get an integer ratio between the simulation time step `DT` and the specified storage interval `DT_WTR`.

7.6.16.1.2. Amount of storage, one input line

```
NOTURB ITURBFM
```

- `NOTURB: integer`: Number of wind turbines for storage
- `ITURBFM: integer`: File format code for storage
  - `ITURBFM = 1`: Storage on file in ASCII format.
  - `ITURBFM = 2`: Storage on file in binary format.

Note that `NOTURB` has to be set to 1 in present program version

The wind turbine responses are written to `<prefix>_witurb.asc` or `<prefix>_witurb.bin`.

The contents are described in `key_<prefix>_witurb.txt`

7.6.16.1.3. Wind turbine identification for storage, `NOTURB` input lines

```
TURB-ID
```

`TURB-ID: character(8)`: Wind turbine identifier

7.6.17. File storage for wind turbine blade responses

This option enables export of wind turbine blade responses to file in binary or ASCII format.

7.6.17.1. Data group identifier, one input line

```
WTBLade RESPonse STORage
```

7.6.17.1.1. Specification of the amount of responses, one input line

```
AMOUNT
```

- `AMOUNT: character(3)`: Amount of blade responses storage
  - `AMOUNT = MIN`: Minimum amount of responses: - Drag and lift force intensities in foil system - Relative wind velocity in foil system - Angle of attack in foil system
  - `AMOUNT = MED`: Medium amount of responses. In addition to minimum amount: - Drag-, lift and moment coefficients in foil system - Induced wind speed in foil system - Remote incoming wind speed including tower effect in foil system - Separation point position in foil system - Axial and tangential induction factors in rotor system - Axial and tangential load intensities in rotor system - Annulus average axial- and tangential induction velocity
  - `AMOUNT = MAX`: Maximum amount of responses. In addition to medium amount: - Transformation matrix between foil and rotor systems

7.6.17.1.2. Time interval for storage, number of input lines and file format code, one input line

```
DT_TBR  NOSPEC  IBLADFM
```

- `DT_TBR: real`: Desired time interval for storage [T]
  - `DT_TBR = 0`: Storage at each simulation time step
- `NOSPEC: integer > 0`: Number of input lines given to specify elements for which blade responses are stored.
- `IBLADFM: integer`: File format code for storage
  - `IBLADFM = 1`: Storage on file in ASCII format.
  - `IBLADFM = 2`: Storage on file in binary format.

Note that DT_TBR will be adjusted to get an integer ratio between the simulation time step DT and the specified storage interval DT_TBR.

The wind turbine responses are written to <prefix>_blresp.asc or <prefix>_blresp.bin. <

The contents are described in key_<prefix>_blresp.txt

7.6.17.1.3. Specification of elements for blade response storage, NOSPEC input lines

```
LINE-ID  ISEG  IEL
```

- LINE-ID: character(8): Line identifier
- ISEG: integer/character: Local segment number within line LINE-ID
- IEL: integer/character: Local element number within segment ISEG

All elements within the line may be specified by simply giving ALL as input to ISEG. Thus IEL will be dummy input.

All elements within one segment may be specified by giving ALL as input to IEL.

7.6.18. Export of element responses

This option enables export of element responses for subsequent communication with general advanced animation tools. The instruction is applicable for non-linear dynamic analysis only.

7.6.18.1. Data group identifier, one input line

```
STORe VISUalisation RESPonses
```

7.6.18.2. Amount of response storage and file format, one input line

```
TCONDS TCONDE DELT CHFORM
```

- TCONDS: real, default: 0: Start time for export
- TCONDE: real, default: 10^5: End time for export
- DELT: real, default: See below: Time increment for export
- CHFORM: character:
  - = VIS: Export to file format used by the computer program SIMVIS for response visualization subsequent to dynamic analysis
  - = RAF: Export to file format of type RAF

Default values of `DELT`: - `DTWF`: Time increment used for pre-sampling of irregular waves and prescribed motions - `DT`: Time increment used in time integration for regular analysis

7.6.18.3. Detailed specification of exported element responses

This data group is optionally given for `CHFORM = VIS`

In present version it is possible to specify element responses in form of effective tension, resulting curvature and longitudinal stress (if available). By default all available element responses for all lines will be exported. This input line makes it possible limit or specify response types for selected lines in the system.

Number of input lines: as many as necessary. ~ OPTION CHRESP CHILIN ~

- `OPTION: character`:
  - `= STORE`
  - `= NOSTORE`
- `CHRESP: character`: Response type to be exported
  - `= EFF-AX-FORCE`: Effective tension
  - `= RES-CURV`: Resultant curvature
  - `= LONG-STRESS`: Longitudinal stress
  - `= ALL`: All of the above described responses
- `CHILIN: character`:
  - `= LINE-ID`: Line identifier
  - `= ALL`: All lines

7.6.19. Termination of input data

To terminate an input data stream, simply give the following, which is interpreted as a data-group identifier.

`END`

7.7. Description of Additional Input Files

7.7.1. Dynamic Current Variation

The file ``CHFCUR" specified in Dynamic current variation, contains the description of dynamic current variation. The file is a free format sequential ASCII-file.

The current velocity and direction have to be specified at all levels defined in the preceding static analysis. The static current profile is interpreted as the current profile at time equal to zero. The dynamic current profile is described at an arbitrary number of time instants, given by increasing values. Linear interpolation is used for intermediate values. If the last defined time instant is exceeded during simulation, the current profile is assumed constant and equal to the last specification for the continued simulation.

File description

7.7.1.1. Number of specified time instants, one input line

```
NDYCUR
```

- `NDYCUR: integer > 1`: The number of time instants for which current profile is given.

The input data in `Number of levels and time instant, one input line' and `Current velocity and direction, one input line per current level, i.e. NLCUR input lines' (below) must be given in one block for each defined time instant.

7.7.1.1.1. Number of levels and time instant, one input line

```
NLCUR TIMDCU
```

- `NLCUR: integer`: Number of levels in current profile. The number of levels has to be equal the number used in the preceding static analysis
- `TIMDCU: real > 0`: Time instant for the specified current profile $[\mathrm{T}]$

7.7.1.1.2. Current velocity and direction, one input line per current level, i.e. NLCUR input lines

```
CURDIR CURVEL CURVEZ
```

- `CURDIR: real`: Direction of current velocity.
    - The angle is measured in degrees from global x-axis counter clockwise to the current vector, confer Current parameters
- `CURVEL: real, default: 0`: Current velocity $[\mathrm{L}/\mathrm{T}]$
- `CURVEZ: real, default: 0`: Vertical current velocity $[\mathrm{L}/\mathrm{T}]$

7.7.2. Dynamic Nodal Forces

The file ``CHFLOA'' specified in Dynamic nodal forces, contains the description of dynamic nodal load components; i.e. user-deined external dynamic loads given as time series. The file is a free format sequential ASCII-file. Two alternative formats are available; the original format with multiple input lines for each time instant loads are specified for and the column format with one line for each time instant loads are specified for. The first input line in the file is used to determine which format the file is read in. If only one number is found on the first input line, the file is read using the original format. If more than one number is found, the file is read using the column format.

The dynamic nodal load components are described by values at specified time instants, which must be increasing. Intermediate values are found by linear interpolation. Between the start of the simulation and the first time instant with specified loads, the loads are linearly increased from zero to the first values given. If the simulation continues after the last defined time instant, the nodal load components are kept constant at the last values given.

The number of nodal load components, location and direction are defined in Dynamic nodal forces. This data group also defines the order in which the load components are to be specified on the file.

7.7.2.1. File description - original format

7.7.2.2. Number of specified time instants, one input line

```
NTDFO
```

- `NTDFO: integer >= 1`: Number of time instants for which nodal load components are specified.

The input data in `Number of load components and time instant, one input line' and `Load components, MDCOMP input lines' (below) must be given in one block for each defined time instant.

7.7.2.3. Number of load components and time instant, one input line

```
MDCOMP TIMDFO
```

- `MDCOMP: integer`: Number of load components.
    - Used for control: `MDCOMP = NDCOMP`
    - `NDCOMP` is specified in Dynamic nodal forces
- `TIMDFO: real > 0`: Time instant for the specified load components $[\mathrm{T}]$

7.7.2.4. Load components, MDCOMP input lines

```
RLMAG
```

- `RLMAG: real`: Magnitude of load component $[\mathrm{F}]$, $[\mathrm{FL}]$

7.7.2.5. File description - column format

7.7.2.6. Time Step and Load Components, one line for each time step

```
TIMDFO  RLMAGi ..... RLMAGn
```

- `TIMDFO: real > 0`: Time instant for the specified load components $[\mathrm{T}]$
- `RLMAGi: real`: Magnitude of load component $[\mathrm{F}]$, $[\mathrm{FL}]$

`RLMAGi` must be repeated n=`MDCOMP` times.

7.7.3. Diffracted Wave Transfer Functions at Points

The file ``CHFDIF'' specified in Irregular Waves contains the wave kinematics transfer functions.

7.7.3.1. Data group identifier, one input line

```
FIRSt ORDEr DIFFracted wave transfer functions
```

7.7.3.2. Text describing the linear incoming wave to diffracted wave transfer functions, two input lines

```
TXDI1
```

- `TXDI1: character(60)`: Character string

7.7.3.3. Point reference, one input line

```
PTNOUS IVES
```

- `PTNOUS: integer`: Point number defined by user
- `IVES: integer`: Support vessel number

7.7.3.4. Point coordinates, one input line

```
XBDY YBDY ZBDY
```

- `XBDY: real`: x-coordinate of where transfer function is calculated, given in support vessel coordinate system $[\mathrm{L}]$
- `YBDY: real`: y-coordinate of where transfer function is calculated, given in support vessel coordinate system $[\mathrm{L}]$

- `ZBDY: real`: z-coordinate of where transfer function is calculated, given in support vessel coordinate system $[\mathrm{L}]$

7.7.3.5. Dimensioning parameters, one input line

```
NDIR NFRE ITYPIN
```

- `NDIR: integer`: Total number of wave directions (for this point)
- `NFRE: integer`: Total number of frequencies (for this point)
- `ITYPIN: integer`: Code for which format the transfer functions are given in
  - `= 1`: Complex form
  - `= 2`: Amplitude ratio $[1]$ and phase $[\mathrm{deg}]$
  - `= 3`: Amplitude ratio $[1]$ and phase $[\mathrm{rad}]$

7.7.3.6. Data identification, one input line

```
WAVE DIREctions DIFFracted wave transfer functions
```

7.7.3.7. Directions, NDIR input lines

```
IDIR DIR
```

- `IDIR: integer`: Direction number (between 1 and `NDIR`)
- `DIR: real`: Propagation direction of incoming wave, $[\mathrm{deg}]$

7.7.3.8. Data identification, one input line

```
WAVE FREQuencies DIFFracted wave transfer functions
```

7.7.3.9. Frequencies, NFRE input lines

```
IFRE FRE
```

- `IFRE: integer`: Frequency number (between 1 and `NFRE`)
- `FRE: real`: Angular frequency of incoming wave, $[\mathrm{rad}/\mathrm{T}]$

7.7.3.10. Data identification, one input line

```
WAVE ELEVation DIFFracted wave transfer function
```

or

`XVELocity DIFFracted WAVE transfer function`

`YVELocity DIFFracted WAVE transfer function`

`ZVELocity DIFFracted WAVE transfer function`

7.7.3.11. Diffracted wave transfer function, NDIR x NFRE input lines

`IDIR IFRE A B`

- `IDIR: integer`: Direction number
- `IFRE: integer`: Frequency number
- `A: real`: Interpretation according to value of `ITYPIN`
  - `ITYPIN = 1`: Real part
  - `ITYPIN = 2`: Amplitude ratio $[1]$, $[\mathrm{rad/m}]$
  - `ITYPIN = 3`: Amplitude ratio $[1]$, $[\mathrm{rad/m}]$
- `B: real`: Interpretation according to value of `ITYPIN`
  - `ITYPIN = 1`: Imaginary part
  - `ITYPIN = 2`: Phase angle $[\mathrm{deg}]$
  - `ITYPIN = 3`: Phase angle $[\mathrm{rad}]$

Transfer functions for accelerations will be calculated based on velocity transfer functions

7.7.4. Internal flow description

The file ``CHFFLW" specified in Import of internal flow data from file (Deprecated functionality) contains a description of the time-varying internal flow.

7.7.4.1. Heading, one input line

- `<TEXT>: character(78)`:

The heading will be echoed on the `<prefix>_dynmod.res` result file.

7.7.4.2. Specification of internal flow conditions.

Data groups `Specification of time…' and `Specification of flow conditions…' (below) are repeated as many times as necessary. At least two time steps must be given.

7.7.4.2.1. Specification of time, one input line

```
CHIDEN TIME
```

- `CHIDEN: character(4)`:
    - `= TIME`
- `TIME: real`: Time for the following specified flow conditions

7.7.4.2.2. Specification of flow conditions, as many input lines as needed. (Zero input lines may be given.)

```
IFE1 IFE2 DEN VEL
```

- `IFE1: integer`: First flow element with these conditions
- `IFE2: integer`: Last flow element with these conditions
- `DEN: real`: Density of contents $[\mathrm{M/L^3}]$
- `VEL: real`: Velocity of contents $[\mathrm{L/T}]$

The elements in the MRL(s) are numbered consecutively along the MRL. A table of the flow element numbering may be found on the `<prefix>_stamod.res` file.

7.7.4.3. End, one input line

```
END
```

8. Input to OUTMOD

8.1. General Information

The post-processing module `OUTMOD` has two main purposes:

- Generate result printout from the `INPMOD`, `STAMOD` and `DYNMOD` modules.
- Prepare a plot file (`IFNPLO`) for later use by the plot module (`PLOMOD`). Note that this functionality is deprecated.

Locations for result presentation are identified by the line identifier and segment and element numbers specified as input to `INPMOD` for all single riser systems (i.e. SA, SB, SC, SD and AR systems).

The user chooses the amount of printout by giving the appropriate options as input data to `OUTMOD`. As for the other modules, input data to `OUTMOD` are organised in groups. Some groups consist of the data group identifier only, whilst other groups have additional input lines. The first part of the input data is data groups selecting data to be printed. Then a command must be given to start the printing. Then a new set of data groups selecting data may be given etc, see the figure below.

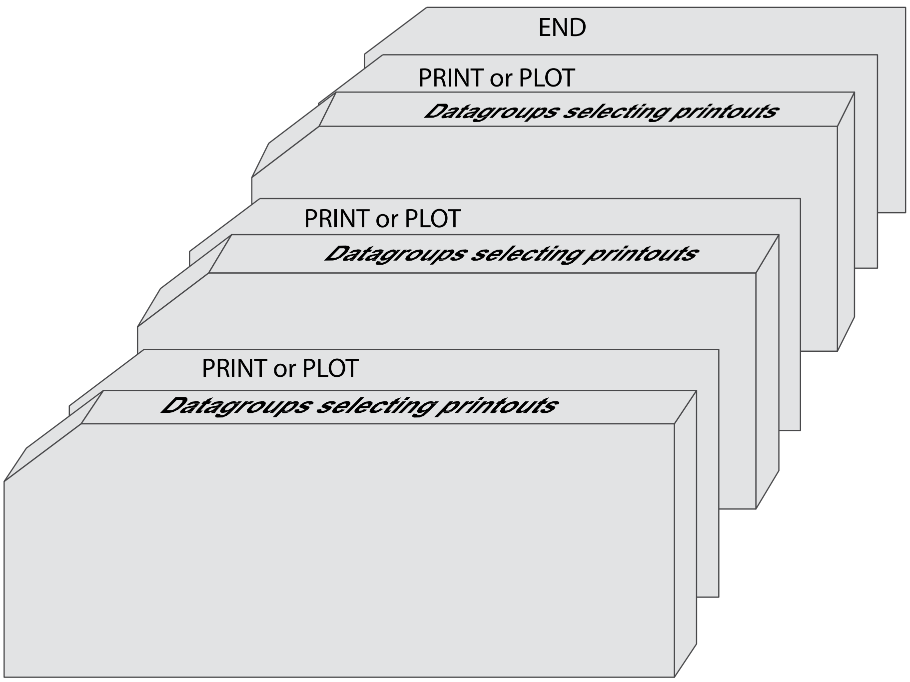

Figure 36. Organization of OUTMOD input

8.2. Data Group A: OUTMOD Identification and Control Data

8.2.1. OUTMOD identification text

If you want an identification text to be printed on the front page of the `OUTMOD` printout, the following data group may be given as the first data group:

8.2.1.1. Data group identifier, one input line

```
OUTMod IDENTIFICATION TEXT CHVERS
```

- `CHVERS: character(8)`: `RIFLEX` output file version, e.g. 3.2

8.2.1.2. Identification text, three input lines

```
Identification text, line no 1
Identification text, line no 2
Identification text, line no 3
```

If the data group is given, all three lines must be given, but they may be blank.

8.2.2. The PRINT command

When `OUTMOD` is used for print generation, the following data group identifier must be given subsequent to the specifications:

```
PRINT
```

After printing, all specifications will be deleted. Subsequent to the print command, a new sequence of specifications and a new print command may be given. This is useful if you want to repeat one or more specifications with different parameters.

8.2.3. Plot generation

This functionality is deprecated!

If you want to have plots from `STAMOD` or `DYNMOD`, you have to run `OUTMOD` first to produce the plot file, `IFNPLO`, which is the only file the plot module `PLOMOD` reads when plotting from the above mentioned modules.

It is not possible to generate plots of all data groups. This is marked by the word `Plot` or `NoPlot` in the right part of the data group identifier frame:

```
WFMOtion TIME SERies        Plot
```

can be plotted, while the following can not:

```
INFIrr CONTRol INFOrmation      NoPlot
```

When you want to use `OUTMOD` to build up file `IFNPLO`, the data group identifiers and print options are exactly the same as when you use `OUTMOD` for normal printout, except for the following:

- `IFNPLO` must be initialized by one of the data group identifiers described in `Initialization of the plot file IFNPLO' (below).
- Instead of the `PRINT` command, the `PLOT` command must be given (see The PLOT command). The `OUTMOD` print file will contain the same information as if the `PRINT` command was given, i.e. the `PLOT` command may be considered as an extension to the `PRINT` command.

Normally, one specification gives one plot which may consist of up to three graphs. In some cases, e.g. when one specification gives results for six degrees of freedoms, two plots are produced per specification.

8.2.3.1. Initialization of the plot file IFNPLO

When `OUTMOD` is to be used for plot generation, one of the following initialization commands must be given after specification of `OUTMOD` identification text (if given), but before any specification described in Sections Data Group B: Output from STAMOD, or Data Group C: Output from DYNMOD:

```
NEW PLOT FILE
```

or

```
APPEnd PLOT FILE
```

If the command `NEW PLOT FILE` is given, `OUTMOD` writes the plot arrays from the beginning of the file, i.e. the previous contents of the file, if any, are overwritten. It is, however, possible to append new plots after already existing plots on file `IFNPLO`. This is achieved by giving the command `APPEND PLOT FILE`. A check is carried out to ensure that the file already contains plots.

A maximum of 100 plots may be stored in one `IFNPLO` file.

If more than one initialization command is given throughout an `OUTMOD` run, they are simply ignored.

8.2.3.2. The PLOT command

When `OUTMOD` is used for plot generation, the following command is given instead of the `PRINT` command (see The PRINT command):

```
PLOT
```

If a plot command is given for a data group that cannot be plotted, a warning message is issued on `OUTMOD` print file and execution continues with the next data group specified.

There may be more than one `PLOT` within one run of `OUTMOD`, following the same rules as for the `PRINT` command. `PRINT` and `PLOT` commands may be mixed within one run.

If the `PLOT` command is given, and neither `NEW PLOT FILE` nor `APPEND PLOT FILE` has been given, the program will terminate with an error message.

8.2.4. Communication with the STARTIMES programs

The following command has been included for communication with the `STARTIMES` programs for statistical analysis of time series

```
STARtimes FILE      NoPlot
```

This command specifies that time series of a selected response quantity shall be written to a file in `STARTIMES` format (i.e. to a file readable by the `STARTIMES` programs). The `STARTIMES FILE` command can be used in connection with the following data group identifiers:

- `WAVE ELEVATION`
- `WFMOTION TIME SERIES`
- `LFMOTION TIME SERIES`
- `TOMOTION TIME SERIES`
- `DYNDISP TIME SERIES`
- `TOTDISP TIME SERIES`
- `DYNFORC TIME SERIES`
- `DYNCURV TIME SERIES`
- `SUPPF TIME SERIES`
- `ELMANGLE TIME SERIES`
- `TOTFORC TIME SERIES`
- `DISTANCE TIME SERIES`
- `CALCURV TIME SERIES`
- `STRESS TIME SERIES`
- `STROKE TIME SERIES`

A response quantity is written to the `STARTIMES` file by giving `STARTIMES FILE` immediately before the `PRINT` or `PLOT` command.

The name of the `STARTIMES` file is `<prefix>_outmod.ts`, and it is stored in the current working directory. A description of this file is found in Description of STARTIMES file.

8.2.5. The END command

To terminate the input data, the following data group identifier is given as the last input line on the `OUTMOD` input file.

```
END
```

The END data group is mandatory

8.3. Data Group B: Output from STAMOD

Description of result presentation from static analyses is given in the following.

8.3.1. Results from static fixed parameter analysis

Displacement and force data from static fixed parameter analysis are established by the `STAMOD` module and stored on file `IFNSTA`.

Specifying this output after a parameter variation run will produce the results of the last parameter variation step.

8.3.1.1. Static dimension information

If you want dimension parameters, such as no of load steps, no of nodes etc, to be printed, give

```
STATic DIMENsion PARAmeters         NoPlot
```

8.3.1.2. System information

If you want information about the connection between the local line, segment and element number given as input to `INPMOD` and the global `FEM` element/nodal numbers generated by `STAMOD`, give

```
STATic SYSTem INFOrmation       NoPlot
```

A more detailed description is given on the `STAMOD` print file.

8.3.1.3. Coordinates of final static configuration

8.3.1.3.1. Data group identifier, one input line

```
STATic COORdinates      Plot
```

8.3.1.3.2. Print options, one input line

```
ICONF LINE-ID IPROJ
```

- `CONF: integer`: Configuration switch
    - `ICONF=1`: Initial configuration (catenary configuration)
    - `ICONF=2`: Final configuration (Results from FEM or CATFEM analysis)
- `LINE-ID: character(8)`: Line identifier for which coordinates are wanted. You may specify `ALL` to include all lines in the system.
    - Note that specifying a specific line gives a 2D-plot, while specifying `ALL` gives a 3D-plot

`LINE-ID=0`: Plot of 2D geometry of all lines

- `IPROJ: integer`: Projection code
  - dummy if `LINE-ID=ALL`
  - `IPROJ=1`: Output of x-y coordinates
  - `IPROJ=2`: Output of x-z coordinates
  - `IPROJ=3`: Output of y-z coordinates

8.3.1.4. Axial forces from catenary analysis

Note that no moments are included in the catenary analysis.

8.3.1.4.1. Data group identifier, one input line

```
INITial AXIAl FORCe         Plot
```

8.3.1.4.2. Line specification, one input line

```
LINE-ID
```

- `LINE-ID: character(8)`: Line identifier for which forces are wanted. You may specify `ALL` to include all lines in the system

8.3.1.5. Forces from static fixed parameter analysis

Forces are printed as force, bending and torsional moments.

8.3.1.5.1. Data group identifier, one input line

```
FINAl STATic FORCes         Plot
```

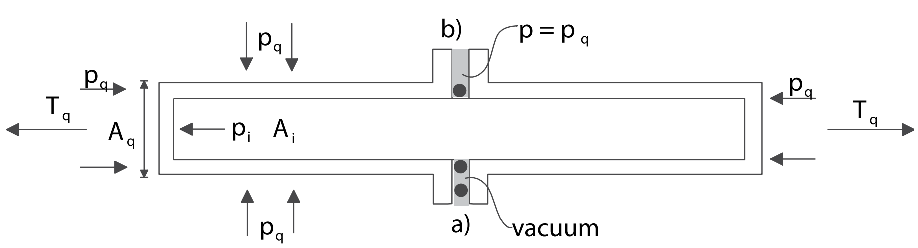

Figure 37. Pipe wall force calculation

Pipe wall force, axial:

$T_W = T_e + p_iA_i - p_eA_e[+m_iv_i^2]$

(In cases with high pressure(s) it may be important to include the radial stress when material strain is to be evaluated)

This is identical with the flange force in case of a double seal (at $r = ri$ and $r = re$)

a. $T_F = T_W$

In the case of an inner seal only:

b. $T_F = T_e + p_iA_i - p_eA_i[+m_iv_i^2]$

Any other sealing radius:

c. $T_F = T_e + (p_i - p_e)A_s[+m_iv_i^2]$

Where: - $A_s = \pi r_s^2$ - r_s = sealing radius

rs = sealing radius

$m_iv_i^2$ is an additional term for cases with internal fluid flow.

8.3.1.5.2. Print options, one input line

```
LINE-ID IDOF1 IDOF2 IDOF3 IDOF4
```

- `LINE-ID: character(8)`: Line identifier for which forces are wanted. You may specify `ALL` to include all lines in the system.
- `IDOF1: integer`: Degree of freedom for first figure
  - `IDOF1=0`: Not included
  - `IDOF1=1`: Axial force
  - `IDOF1=2`: Torsional moment
  - `IDOF1=3`: Bending moment about local y-axis
  - `IDOF1=4`: Bending moment about local z-axis
  - `IDOF1=5`: Pipe wall force, incl hydrostatic pressures
  - `IDOF1=6`: Shear force in local y-direction
  - `IDOF1=7`: Shear force in local z-direction

`IDOF2: integer`: Degree of freedom for second figure

  - Interpretation as for `IDOF1`

- `IDOF3: integer`: Degree of freedom for third figure

  - Interpretation as for `IDOF1`

- `IDOF4: integer`: Degree of freedom for fourth figure

  - Interpretation as for `IDOF1`

No of figures in one plot may vary from 1-3 depending on the number of response quantities specified (e.g. IDOFi).

Note that the print part of this option always will produce results for all stored degrees of freedom, i.e. axial force, torsional moment and bending moments about local y- and z-axes. The parameters are used to specify the dof's to be plotted.

8.3.1.6. Stress from static analysis

8.3.1.6.1. Data group identifier, one input line

```
FINAl STATic STREsses       Plot
```

8.3.1.6.2. Output options, one input line

```
LINE-ID IDOF
```

- `LINE-ID: character(8)`: Line identifier for which stresses are wanted. You may specify `ALL` to include all lines in the system.

- The following parameter is used to specify the dof to be considered

  - `IDOF: integer`: Stress component

    - `IDOF=1`: Axial stress
    - `IDOF=2`: Torsional stress
    - `IDOF=3`: Bending stress
    - `IDOF=4`: Axial + bending stress
    - `IDOF=5`: Shear stress
    - `IDOF=6`: Shear + torsional stress

IDOF=7: Equivalent stress

- IDOF=8: Hoop stress
- IDOF=9: Radial stress

Effect of internal/external pressure and fluid velocity are included

8.3.1.6.3. Specification of point for stress calculation, one input line

```
IMAX THETA INEX
```

- `IMAX: integer, default: 1`: Stress location option
  - `IMAX=1`: Maximum stresses in cross section estimated
  - `IMAX=0`: Stresses calculated at location specified by `THETA` and `INEX`
- `THETA: real, default: 0`: Angle (in degrees) from local y-axis for stress calculation.
  - Dummy for `IMAX=1`
- `INEX: integer, default: 2`: Location code
  - Dummy for `IMAX=1`
  - `INEX=1`: Inner wall
  - `INEX=2`: Outer wall

For `IMAX=1`, the maximum stresses of type `IDOF` in the cross section are estimated. The equivalent stress (von Mises) is supposed to be maximum where the bending stress is maximum or minimum.

8.3.2. Output from static parameter variation analysis

Displacement and force data from static parameter variation analysis are established by the `STAMOD` module and stored on file `IFNSTA`. Result presentation from static parameter variation analysis is described in the following.

8.3.2.1. System geometry from parameter variation analysis

8.3.2.1.1. Data group identifier, one input line

```
PARAmeter VARIation COORdinates         Plot
```

8.3.2.1.2. Line specification, one input line

`LINE-ID IOTYP IPV1 NVP`

- `LINE-ID: character(8)`: Line number for which geometry are wanted. You may specify `ALL` to include all lines in the system.
  - `ALL` gives a 3D plot of all lines.
  - `LINE-ID = 0` gives a 2D plot of all lines
- `IOTYP: integer`: Degree of freedom specification
  - Dummy if `ILINE = ALL`
  - `IOTYP=1`: x-y coordinates
  - `IOTYP=2`: x-z coordinates
  - `IOTYP=3`: y-z coordinates
- `IPV1: integer`: First parameter variation step to be included
- `NVP: integer`: No of parameter variation steps to be included

The first plot to appear will be for step no `NSTEP+IPV1` where `NSTEP` is total number of load steps used in the static analysis with fixed parameters.

Negative value of `IPV1` is possible, which allows for plotting of static configuration at all load steps in static analysis with fixed parameters.

It is also possible to plot static configurations from 1st load step to last successful solution when static analysis fails, which can be very useful for detection of possible instability problems.

8.3.2.2. Displacement of selected nodes from parameters variation analysis

8.3.2.2.1. Data group identifier, one input line.

```
PARAmeter VARIation DISPlacements       Plot
```

8.3.2.2.2. Output code, one input line

`IPV1 NPV IDOF1 IDOF2 IDOF3 NNODC`

- `IPV1: integer`: First parameter variation step to be included
- `NPV: integer`: No of parameter load steps to be included. (A large number includes the remaining steps)
- `IDOF1: integer`:

    - `IDOF1=1`: Translation in x-direction
    - `IDOF1=2`: Translation in y-direction
    - `IDOF1=3`: Translation in z-direction
- `IDOF2: integer`:
    - Interpretation as for `IDOF1`
- `IDOF3: integer`:
    - Interpretation as for `IDOF1`
- `NNODC: integer`: No. of input lines used for node specification

No of figures on each plot may vary from 1 to 3, depending on `IDOFi`

The first plot to appear will be for step no `NSTEP+IPV1` where `NSTEP` is the total number of load steps in the static analysis with fixed parameters.

Negative value of `IPV1` is allowed (see System geometry from parameter variation analysis).

8.3.2.2.3. Node specification, NNODC input lines

```
LINE-ID ISEG INODE
```

- `LINE-ID: character(8)`: Line identifier.
    - You may specify `ALL` to include all lines
- `ISEG: integer/character`: Segment number.
    - You may specify `ALL` to include all segments.
    - `ENDS` includes the end segments on the line
- `INODE: integer/character`: Node number.
    - `ALL` includes all nodes
    - `ENDS` includes end nodes on the above specified segment

8.3.2.3. Forces on selected elements from parameter variation analysis

8.3.2.3.1. Data group identifier, one input line

```
PARAmeter VARIation FORCes      Plot
```

8.3.2.3.2. Output options, one input line

```
IPV1 NPV IDOF1 IDOF2 IDOF3 NNELC
```

- `IPV1: integer`: First parameter variation step to be included
- `NPV: integer`: No of parameter load steps to be included. (A large number includes the remaining steps)
- `IDOF1: integer`: Degree of freedom specification for Figure 1
  - `IDOF1=0`: No output
  - `IDOF1=1`: Axial force
  - `IDOF1=2`: Torsional moment
  - `IDOF1=3`: Bending moment about local y-axis
  - `IDOF1=4`: Bending moment about local z-axis
- `IDOF2: integer`:
  - Interpretation as for `IDOF1`
- `IDOF3: integer`:
  - Interpretation as for `IDOF1`
- `NNELC: integer`: No. of input lines used for element specification

The first plot to appear will be for step no `NSTEP+IPV1` where `NSTEP` is the total number of load steps used in the static analysis with fixed parameters.

Negative value of `IPV1` is allowed (see System geometry from parameter variation analysis).

No of figures in one plot may vary from 1 to 3, depending on the number of response quantities specified (e.g. `IDOFi`).

8.3.2.3.3. Element specification, NNELC input lines

```
LINE-ID ISEG IELM
```

- `LINE-ID: character(8)`: Line identifier.
  - You may specify `ALL` to include all lines
- `ISEG: integer/character`: Segment number.

- You may specify `ALL` to include all segments.
- `ENDS` includes the end segments on the line

- `IELM: integer/character`: Element number.
    - `ALL` includes all elements
    - `ENDS` includes end elements on the above specified segment

8.4. Data Group C: Output from DYNMOD

8.4.1. Results from irregular wave analysis

Results from the irregular wave analysis consists of: - sampled Fourier components of waves stored on file `IFNIRR` at global origin, `x=y=z=0` - motion of the support vessel, stored on file `IFNIRR` - motion transfer functions for the support vessel

8.4.1.1. Control information

8.4.1.1.1. Data group identifier, one input line

```
IFNIrr CONTrol INFOrmation      NoPlot
```

In addition to dimension parameters, control information also consists of directions and frequencies for which Fourier components are stored.

8.4.1.2. Sampled Fourier components

8.4.1.2.1. Data group identifier, one input line

```
FOURier COMPonents WAVEs        Plot
```

8.4.1.2.2. Output parameters, one input line

```
ICOMP IDIR ISEC IW1 NW IJP
```

- `ICOMP: integer`: Component code
    - `ICOMP=1`: Wind sea
    - `ICOMP=2`: Swell
- `IDIR: integer`: Direction no wanted
- `ISEC: integer`: Sequence no wanted (dummy)

- `IW1: integer`: Number of the first frequency for which Fourier components are wanted
- `NW: integer`: No of frequencies for which Fourier components are wanted
- `IJP: integer, default: 1`: Jump parameter

Fourier components are printed for frequencies no `IW1`, `IW1+IJP`, `IW1`+2x`IJP`, …, `IW1`+(`NW`-1)x`IJP`

The components are printed/plotted as amplitude and phase angle (degrees)

8.4.1.3. Wave elevation

8.4.1.3.1. Data group identifier, one input line

```
WAVE ELEVation      Plot
```

8.4.1.3.2. Output parameters, one input line

```
ICOMP IDIR ISEC IT1 NTS XP1 XP2
```

- `ICOMP: integer`: Component code
  - `ICOMP=1`: Wind sea
  - `ICOMP=2`: Swell
- `IDIR: integer`: Direction no wanted
- `ISEC: integer`: Sequence no wanted (dummy)
- `IT1: integer`: First time step included
- `NTS: integer`: Number of time steps included
- `XP1: real, default: 0`: Global x-coordinate for wave elevation
- `XP2: real, default: 0`: Global y-coordinate for wave elevation

A Fourier transformation of the wave spectrum is performed. Maximum number of time steps will be (NWIMAX-1)*2. Use the option `IFNIRR CONTROL INFORMATION` (see Control information).

In case of longcrested sea one direction is applied. In case of shortcrested sea, 11 directions are used and mean wave direction is no. 6. The other directions are spread around the mean direction in the interval $[-75^\circ, 75^\circ]$ in intervals of 15°.

8.4.1.4. Wave frequency motion time series

8.4.1.4.1. Data group identifier, one input line

```
WFMOtion TIME SERIes        Plot
```

8.4.1.4.2. Output options, one input line

```
IOP IMOT IDERIV ISEQ1 NSEQ IT1 NTS ITJMP IVES
```

- `IOP: integer`: Code for type of output
    - `IOP=1`: Time series
    - `IOP=2`: Time series statistics
    - `IOP=3`: Spectral analysis
- `IMOT: integer`: Direction
    - `IMOT=1`: Displacement in global x-direction
    - `IMOT=2`: Displacement in global y-direction
    - `IMOT=3`: Displacement in global z-direction
    - `IMOT=4`: Rotation about x-axis
    - `IMOT=5`: Rotation about y-axis
    - `IMOT=6`: Rotation about z-axis
- `IDERIV: integer`: Code for derivative of response
    - `IDERIV=0`: Analyse original series
    - `IDERIV=1`: Analyse 1st derivative
    - `IDERIV=2`: Analyse 2nd derivative
- `ISEQ1: integer`: First sequence to be included (dummy)
- `NSEQ: integer`: No of sequence to be included (dummy)
- `IT1: integer`: First time step of each sequence to be included
- `NTS: integer`: No of time steps of each sequence to be included

`ITJMP: integer, default: 1`: Jump parameter

  - Time step nos. `IT1, IT1+ITJMP, IT1+2xITJMP,…, IT1+(NTS-1)xITJMP` are included

- `IVES: integer, default: 1`: Vessel number reference in case of multivessel systems.The vessels are numbered from 1 to `NVES`

Note that `IMOT` refers to the global coordinate system, not the vessel coordinate system.

8.4.1.4.3. Transformation of wave frequency motion time series, one input line

`ITRANS XP YP ZP`

- `ITRANS: integer, default: 0`: Transformation code
  - `ITRANS=0`: No transformation, motions of vessel reference point
  - `ITRANS=1`: Transformation gives motion `IMOT` (see previous input line) of point defined by `XP`, `YP` and `ZP`
- `XP: real, default: 0`: X-coordinate in global system, relative to the vessel reference point
- `YP: real, default: 0`: Y-coordinate in global system, relative to the vessel reference point
- `ZP: real, default: 0`: Z-coordinate in global system, relative to the vessel reference point

If `ITRANS=0`, `XP`, `YP` and `ZP` are dummy parameters

8.4.1.4.4. Options for the output distribution functions of the high frequency motion time series statistics, one input line

This input line is given only if `IOP=2`. ~ NCL XCMIN XCMAX ~

- `NCL: integer`: No of classes in the output distribution functions (i.e. no of points on the abscissa axis)
  - `0<NCL<41`
- `XCMIN: real`: Range of argument values for output distribution functions is `XCMIN*sx(1) - XCMAX*sx(1)` in which `sx(1)` is the standard deviation of `x` estimated from the first sequence.
- `XCMAX: real`:

8.4.1.4.5. Spectrum smoothing parameter for the spectral analysis of the high frequency motion, one input line

This input line is given only if `IOP=3`. ~ MSM ~

- `MSM: integer, default: 0`: Smoothing parameter
  - `MSM=0`: No smoothing
  - `MSM>0`: Smoothing by averaging over 2*`MSM`+1 values.

8.4.1.5. Low frequency motion time series

8.4.1.5.1. Data group identifier, one input line

```
LFMOtion TIME SERIEs        Plot
```

8.4.1.5.2. Output options, one input line

```
IOP IMOT IDERIV ISEQ1 NSEQ IT1 NTS ITJMP IVES
```

- `IOP: integer`: Code for type of output
  - `IOP=1`: Time series
  - `IOP=2`: Time series statistics
  - `IOP=3`: Spectral analysis
- `IMOT: integer`: Direction code
  - Legal values:
    - `IMOT=1`: Surge
    - `IMOT=2`: Sway
    - `IMOT=6`: Yaw
- `IDERIV: integer`: Code for derivative of response
  - `IDERIV=0`: Analyse original series
  - `IDERIV=1`: Analyse 1st derivative
  - `IDERIV=2`: Analyse 2nd derivative
- `ISEQ1: integer`: First sequence to be included (dummy)

- `NSEQ: integer`: No of sequence to be included (dummy)
- `IT1: integer`: First time step of each sequence to be included
- `NTS: integer`: No of time steps of each sequence to be included
- `ITJMP: integer, default: 1`: Jump parameter
  - Time step nos. `IT1, IT1+ITJMP, IT1+2xITJMP,…, IT1+(NTS-1)xITJMP` are included
- `IVES: integer, default: 1`: Vessel number reference in case of multivessel systems.The vessels are numbered from 1 to `NVES`

8.4.1.5.3. Transformation of the low-frequency motion time series, one input line

Identical to Transformation of high frequency motion time series, one input line for Wave frequency motion time series.

8.4.1.5.4. Options for the output distribution functions of the low frequency motion time series statistics, one input line

This input line is given only if `IOP=2`.

Identical to Options for the output distribution functions of the high frequency motion time series statistics, one input line for Wave frequency motion time series.

8.4.1.5.5. Spectrum smoothing parameter for the spectral analysis of the low frequency motion time series, one input line

This input line is given only if `IOP=3`.

Identical to Spectrum smoothing parameter for the spectral analysis of the high frequency motion, one input line for Wave frequency motion time series.

8.4.1.6. Total motion time series

8.4.1.6.1. Data group identifier, one input line

```
TOMOtion TIME SERIes        Plot
```

8.4.1.6.2. Output options, one input line

```
IOP IMOT IDERIV ISEQ1 NSEQ IT1 NTS ITJMP IVES
```

- `IOP: integer`: Code for type of output
  - `IOP=1`: Time series

    - `IOP=2`: Time series statistics
    - `IOP=3`: Spectral analysis
- `IMOT: integer`: Direction code
    - `IMOT=1`: Displacement in global x-direction
    - `IMOT=2`: Displacement in global y-direction
    - `IMOT=6`: Rotation about z-axis
- `IDERIV: integer`: Code for derivative of response
    - `IDERIV=0`: Analyse original series
    - `IDERIV=1`: Analyse 1st derivative
    - `IDERIV=2`: Analyse 2nd derivative
- `SEQ1: integer`: First sequence to be included (dummy)
- `NSEQ: integer`: No of sequence to be included (dummy)
- `IT1: integer`: First time step of each sequence to be included
- `NTS: integer`: No of time steps of each sequence to be included
- `ITJMP: integer, default: 1`: Jump parameter
    - Time step nos. `IT1, IT1+ITJMP, IT1+2xITJMP,…, IT1+(NTS-1)xITJMP` are included
- `IVES: integer, default: 1`: Vessel number reference in case of multivessel systems.The vessels are numbered from 1 to `NVES`

8.4.1.6.3. Transformation of total motion time series, one input line

Identical to Transformation of high frequency motion time series, one input line for Wave frequency motion time series.

8.4.1.6.4. Options for the output distribution functions of the time series statistics of total motion, one input line

This input line is given only if `IOP=2`.

Identical to Options for the output distribution functions of the high frequency motion time series statistics, one input line for Wave frequency motion time series.

8.4.1.6.5. Spectrum smoothing parameter for the spectral analysis of the total motion, one input line

This input line is given only if `IOP=3`.

Identical to Spectrum smoothing parameter for the spectral analysis of the high frequency motion, one input line for Wave frequency motion time series.

8.4.1.7. Vessel motion transfer functions

8.4.1.7.1. group identifier, one input line

```
WFTRansfer FUNCtion DOF      Plot
```

`DOF` means degree of freedom, and may be `XG`, `YG`, `ZG`, `XGROT`, `YGROT` or `ZGROT`.

8.4.1.7.2. Output options, one input line

```
IOP IDIR1 NDIR ITRAN IVES
```

- `IOP: integer`: Code for type of output
  - `IOP=1`: Complex form (real, imaginary)
  - `IOP=2`: Real form (amplitude ratio, phase (degrees))
  - `IOP=3`: Real form (amplitude ratio, phase (radians))
- `IDIR1: integer`: First direction to be included
- `INDIR: integer`: No of directions to be included
- `ITRAN: integer`: Code for transformation
  - `ITRAN=0`: No transformation
  - `ITRAN=1`: Transformation of origin motion to point (`XV1`, `XV2`, `XV3`), see next input line.
  - Dummy if degree of freedom is `XGROT`, `YGROT` or `ZGROT`
- `IVES: integer, defaul: 1`: Vessel number

8.4.1.7.3. The coordinates of the point on the vessel for which the vessel motion transfer functions are wanted, one input line

If `ITRAN=0`, or the degree of freedom is `XGROT`, `YGROT` or `ZGROT`, this input line is skipped. ~ XV1 XV2 XV3 ~

- `XV1: real`: X-coordinate of the point
- `XV2: real`: Y-coordinate of the point
- `XV3: real`: Z-coordinate of the point

The coordinates are referred to the global coordinate system, relative to the vessel reference point.

The transfer functions for different degrees of freedom may be given without the `PRINT` or `PLOT` statement between.

8.4.2. Results from time domain dynamic analysis

8.4.2.1. Storage information

8.4.2.1.1. Data group identifier, one input line

```
TIME DOMAin PARAmeters
```

8.4.2.1.2. Print options, one input line

```
IDNOD IFNOD ICNOD
```

- `IDNOD: integer, default: 1`: Switch for printing of nodes for which displacements are stored
  - `IDNOD=0`: No print
  - `IDNOD=1`: The nodes, for which displacements are stored, are printed
- `IFNOD: integer, default: 1`: Switch for printing of elements for which force data are stored
  - `IFNOD=0`: No print
  - `IFNOD=1`: The nodes, for which force data are stored, are printed
- `ICNOD: integer, default: 1`: Switch for printing of elements for which curvature data are stored
  - `ICNOD=0`: No print
  - `ICNOD=1`: The elements, for which curvature data are stored, are printed

8.4.2.2. Snapshot plot from time domain analysis

This option will create pictures of the dynamic configuration at several time steps.

8.4.2.2.1. Data group identifier, one input line

```
DYNAmic SNAPshot PLOT       Plot (only)
```

8.4.2.2.2. Plot options

```
IPROJ IT1 NTS NLIC IJUMP
```

- `IPROJ: integer`: Project in code
  - `IPROJ=1`: x-z coordinates
  - `IPROJ=2`: y-z coordinates
  - `IPROJ=3`: x-y coordinates
- `IT1: integer`: First stored time step to be included
- `NTS: integer/character`: No of stored time steps to be included.
  - You may specify `REST` to include the remaining time steps
- `NLIC: integer`: No. of input lines to describe line specification
- `IJUMP: integer, default: 1`: Plot every `IJUMP` stored time step

8.4.2.2.3. Line specification, NLIC input lines

```
LINE-ID
```

- `LINE-ID: character(8)`: Line identifier to be plotted. You may specify `ALL` to include all lines in the system

The lines are plotted only if at least the end node coordinates are stored. Line configurations for all stored time steps are plotted.

8.4.2.3. System snapshot plot from time domain analysis

This option is an extension to the option `DYNAMIC SNAPSHOT PLOT`. You are able to plot the wave particle motion, the vessel motion and the riser motion in one plot.

8.4.2.3.1. Data group identifier, one input line

```
SYSTem SNAPshot PLOT        Plot (only)
```

8.4.2.3.2. Plot options, one input line

`IPROJ IT1 NTS IJUMP NLIC NPVESP NPWAPO IVES XCGVES YCGVES ZCGVES`

- `IPROJ: integer`: Projection code
  - `IPROJ=1`: X-Z coordinates
  - `IPROJ=2`: Y-Z coordinates
  - `IPROJ=3`: X-Y coordinates
- `IT1: integer`: First stored time step to be included
- `NTS: integer`: No of stored time steps to be included. You may specify REST to include the remaining time steps
- `IJUMP: integer`: Include every `IJUMP` stored time steps
- `NLIC: integer`: No. of input lines to describe line specification
  - `NLIC=0`: No riser snapshot plot
- `NPVESP: integer`: No of coordinates to describe the vessel
  - `NPVESP=0`: No vessel snapshot plot
- `NPWAPO: integer`: No of coordinates to describe the wave particle motion
  - `NPWAPO=0`: No wave particle snapshot plot
- `IVES: integer, default: 1`: Vessel number
- `XCGVES: real`: Static X coordinate of the vessel
- `YCGVES: real`: Static Y coordinate of the vessel
- `ZCGVES: real`: Static Z coordinate of the vessel

8.4.2.3.3. Line specification, NLIC input lines

`LINE-ID`

- `LINE-ID: integer/character(8)`: Line identifier to be plotted. You may specify `ALL` to include all lines in the system

The lines are plotted only if at least the end node coordinates are stored.

8.4.2.3.4. Vessel description, NPVESP input lines. The specified points are connected by one line to illustrate a part of the vessel contour

```
IPV XVT YVT ZVT
```

- `IPV: integer`: Coordinate no.
- `XVT: real`: Vessels X-coordinate in global system referred from vessel origin $[\mathrm{L}]$
- `YVT: real`: Vessels Y-coordinate $[\mathrm{L}]$
- `ZVT: real`: Vessels Z-coordinate $[\mathrm{L}]$

8.4.2.3.5. Wave particle description, NPWAPO input lines

```
IPW XPW YPW ZPW
```

- `IPW: integer`: Coordinate no.
  - If `IPW<0`, then the intermediate coordinates between the previous coordinate specification and this one are automatically calculated. The intermediate coordinates are equally spaced on a straight line
- `XPW: real`: X-coordinate of the wave particle $[\mathrm{L}]$
- `YPW: real`: Y-coordinate of the wave particle $[\mathrm{L}]$
- `ZPW: real`: Z-coordinate of the wave particle $[\mathrm{L}]$

The wave particle coordinates are given in the global coordinate system in calm water, i.e. (0.,0.,0.) is wave at global origin. Specifying `ZPW` $\equiv 0.$ for all points will create a plot of the wave surface elevation.

8.4.2.4. Dynamic displacement time series from time domain analysis

Results include only the dynamic time dependant displacements (static values are not included).

8.4.2.4.1. Data group identifier, one input line

```
DYNDisp TIME SERIEs         Plot
```

8.4.2.4.2. Output options, one input line

```
IOP IDOF IT1 NTS NNODC      Plot
```

- `IOP: integer`: Code for type of output

- `IOP=1`: Time series
    - `IOP=2`: Time series statistics
    - `IOP=3`: Spectral analysis
- `IDOF: integer`: Code for degree of freedom
    - Rotational degrees of freedom are only to be presented from linearized dynamic analysis.
    - `IDOF=1`: Translation in x-direction
    - `IDOF=2`: Translation in y-direction
    - `IDOF=3`: Translation in z-direction
    - `IDOF=4`: Rotation about x-axis
    - `IDOF=5`: Rotation about y-axis
    - `IDOF=6`: Rotation about z-axis
- `IT1: integer`: First stored time step to be included
- `NTS: integer`: Number of stored time steps to be included (from `IT1`).
    - A large number includes the remaining time steps
- `NNODC: integer`: No. of input lines used for node specification

For `IOP=3` an FFT analysis is carried out. If `NTS` is not an integer power of 2, a reduced time series will be analysed. In order to get an effective analysis, `IT1` and `NTS` should be selected so that - $\mathrm{IT1} = \mathrm{NT} - 2^{\mathrm{M}} + 1$ - $\mathrm{NTS} = 2^{\mathrm{M}}$

Where NT is the total number of stored time steps and M is the largest integer so that $\mathrm{NTS} <= \mathrm{NT}$. Normally it is preferable to omit the first part of the time series due to transients

8.4.2.4.3. Node specification, NNODC input lines

```
LINE-ID ISEG INODE
```

- `LINE-ID: character(8)`: Line identifier.
    - You may specify `ALL` to include all lines
- `ISEG: integer/character`: Segment number.
    - You may specify `ALL` to include all segments.

- `ENDS` includes the end segments on the line
- `INODE: integer/character`: Node number.
  - `ALL` includes all nodes
  - `ENDS` includes end nodes on the above specified segment

Displacements are not necessarily stored for all nodes, see data group File storage of displacement response for storage information. If the user specifies nodes for which displacements are not stored, these nodes are ignored.

The data group Storage information may be used to obtain an overview of the stored data.

8.4.2.4.4. Options for the output distribution functions of the displacement time series statistics, one input line

This input line is given only if `IOP=2`. ~ NCL XCMIN XCMAX ~

- `NCL: integer`: No of classes in the output distribution functions (i.e. no of points on the abscissa axis)
  - `0<NCL<41`
- `XCMIN: real`: Range of argument values for output distribution functions is `XCMIN*sx(1) - XCMAX*sx(1)` in which `sx(1)` is the standard deviation of `x` estimated from the first sequence
- `XCMAX: real`: See above

8.4.2.4.5. Spectrum smoothing parameter for the spectral analysis of the displacement time series, one input line

This input line is given only if `IOP=3`. ~ MSM ~

- `MSM: integer, default: 0`: Smoothing parameter
  - `MSM=0`: No smoothing
  - `MSM>0`: Smoothing by averaging over 2*`MSM`+1 values

8.4.2.5. Dynamic resulting force time series from time domain analysis

The results include only the dynamic time dependent force. Static values are not included.

8.4.2.5.1. Data group identifier, one input line

```
DYNForce TIME SERIes        Plot
```

8.4.2.5.2. Output options, one input line

```
IOP IDOF IT1 NTS NNELC
```

- `IOP: integer`: Code for type of output
  - `IOP=1`: Time series
  - `IOP=2`: Time series statistics
  - `IOP=3`: Spectral analysis
- `IDOF: integer`: Code for degree of freedom
  - `IDOF=1`: Axial force
  - `IDOF=2`: Torsional moment
  - `IDOF=3`: Bending moment about local y-axis, end 1
  - `IDOF=4`: Bending moment about local y-axis, end 2
  - `IDOF=5`: Bending moment about local z-axis, end 1
  - `IDOF=6`: Bending moment about local z-axis, end 2
  - `IDOF=7`: Shear force in local y-direction, end 1
  - `IDOF=8`: Shear force in local y-direction, end 2
  - `IDOF=9`: Shear force in local z-direction, end 1
  - `IDOF=10`: Shear force in local z-direction, end 2
- `IT1: integer`: First stored time step to be included
- `NTS: integer`: Number of stored time steps to be included (from `IT1`).
  - A large number includes the remaining time steps.
- `NNELC: integer`: No. of input lines used for element specification

For `IOP=3` an FFT analysis is carried out. If `NTS` is not an integer power of 2, a reduced time series will be analysed. In order to get an effective analysis, `IT1` and `NTS` should be selected so that - $\mathrm{IT1} = \mathrm{NT} - 2^{\mathrm{M}} + 1$ - $\mathrm{NTS} = 2^{\mathrm{M}}$

Where NT is the total number of stored time steps and M is the largest integer so that $\mathrm{NTS} <= \mathrm{NT}$. Normally it is preferable to omit the first part of the time series due to transients

8.4.2.5.3. Element specification, NNELC input lines

`LINE-ID ISEG IELM`

- `LINE-ID: character(8)`: Line identifier.
    - You may specify `ALL` to include all lines
- `ISEG: integer/character`: Segment number.
    - You may specify `ALL` to include all segments.
    - `ENDS` includes the end segments on the line
- `IELM: integer/character`: Element number.
    - `ALL` includes all Elements
    - `ENDS` includes end elements on the above specified segment

Forces are not necessarily stored for all elements, see data group File storage for internal forces for storage information. If the user specifies elements for which forces are not stored these elements are ignored.

The data group Storage information may be used to obtain an overview of the stored data.

8.4.2.5.4. Options for the output distribution functions of the force time series statistics, one input line

This input line is given only if `IOP=2`. ~ NCL XCMIN XCMAX ~

- `NCL: integer`: No of classes in the output distribution functions (i.e. no of points on the abscissa axis)
    - `0<NCL<41`
- `XCMIN: real`: Range of argument values for output distribution functions is `XCMIN*sx(1) - XCMAX*sx(1)` in which `sx(1)` is the standard deviation of `x` estimated from the first sequence
- `XCMAX: real`:

8.4.2.5.5. Spectrum smoothing parameter for the spectral analysis of the force time series, one input line

This input line is given only if `IOP=3`. ~ MSM ~

- `MSM: integer, default: 0`: Smoothing parameter

`MSM=0`: No smoothing

- `MSM>0`: Smoothing by averaging over 2*`MSM`+1 values

8.4.2.6. Curvature time series from time domain analysis

Results include only the dynamic time dependant curvature (static values are not included)

See also data group [Curvature time series calculated from dynamic nodal displacements].

8.4.2.6.1. Data group identifier, one input line

```
DYNCURV TIME SERIES         Plot
```

8.4.2.6.2. Output options, one input line

```
IOP IDOF IT1 NTS NNELC
```

- `IOP: integer`: Code for type of output
  - `IOP=1`: Time series
  - `IOP=2`: Time series statistics
  - `IOP=3`: Spectral analysis
- `IDOF: integer`: Code for degree of freedom
  - `IDOF=1`: Curvature about local y-axis, end 1
  - `IDOF=2`: Curvature about local y-axis, end 2
  - `IDOF=3`: Curvature about local z-axis, end 1
  - `IDOF=4`: Curvature about local z-axis, end 2
- `IT1: integer`: First stored time step to be included
- `NTS: integer`: Number of stored time steps to be included (from `IT1`).
  - A large number includes the remaining time steps.
- `NNELC: integer`: No. of input lines used for element specification

For `IOP=3` an FFT analysis is carried out. If `NTS` is not an integer power of 2, a reduced time series will be analysed. In order to get an effective analysis, `IT1` and `NTS` should be selected so that - $\mathrm{IT1} = \mathrm{NT} - 2^{\mathrm{M}} + 1$ - $\mathrm{NTS} = 2^{\mathrm{M}}$

Where NT is the total number of stored time steps and M is the largest integer so that NTS <= NT. Normally it is preferable to omit the first part of the time series due to transients

8.4.2.6.3. Element specification, NNELC input lines

Identical to Element specification, NNELC input lines for Dynamic resulting force time series from time domain analysis.

8.4.2.6.4. Options for the output distribution functions of the curvature time series statistics, one input line

This input line is given only if `IOP=2`.

Identical to Options for the output distribution functions of the force time series statistics, one input line for Dynamic resulting force time series from time domain analysis.

8.4.2.6.5. Spectrum smoothing parameter for the spectral analysis of the curvature time series, one input line

This input line is given only if `IOP=3`.

Identical to Spectrum smoothing parameter for the spectral analysis of the force time series, one input line for Dynamic resulting force time series from time domain analysis.

8.4.2.7. Curvature time series calculated from dynamic nodal displacements

See also Curvature time series from time domain analysis for curvature component time series.

This option gives absolute value of curvature in 3D space at a specified node. Calculation of curvature is based on the interpolating polynomial through the positions of 3 adjacent nodes in the same line. Curvature can therefore only be calculated if displacement time series are stored for the specified node and two neighbouring nodes (see data group File storage of displacement response for storage information). The data group Storage information may be used to obtain an overview of the stored data.

Calculation of curvature at line ends is omitted.

8.4.2.7.1. Data group identifier, one input line

```
CALCurv TIME SERIes         Plot
```

Total curvature calculated from the selected node and the two neighbouring nodes.

8.4.2.7.2. Output options, one input line

```
IOP IT1 NTS NNODC
```

- `IOP: integer`: Code for type of output
  - `IOP=1`: Time series
  - `IOP=2`: Time series statistics
  - `IOP=3`: Spectral analysis
- `IT1: integer`: First stored time step to be included
- `NTS: integer`: Number of stored time steps to be included (from `IT1`).
  - A large number includes the remaining time steps.
- `NNODC: integer`: No. of input lines used for element specification

For `IOP=3` an FFT analysis is carried out. If `NTS` is not an integer power of 2, a reduced time series will be analysed. In order to get an effective analysis, `IT1` and `NTS` should be selected so that - $\mathrm{IT1} = \mathrm{NT} - 2^{\mathrm{M}} + 1$ - $\mathrm{NTS} = 2^{\mathrm{M}}$

Where NT is the total number of stored time steps and M is the largest integer so that $\mathrm{NTS} <= \mathrm{NT}$. Normally it is preferable to omit the first part of the time series due to transients.

8.4.2.7.3. Node specification, NNODC input lines

Identical to Node specification, NNODC input lines for Dynamic displacement time series from time domain analysis.

8.4.2.7.4. Options for the output distribution functions of the curvature time series statistics, one input line

This input line is given only if `IOP=2`.

Identical to Options for the output distribution functions of the displacement time series statistics, one input line for Dynamic displacement time series from time domain analysis.

8.4.2.7.5. Spectrum smoothing parameter for the spectral analysis of the curvature time series, one input line

This input line is given only if `IOP=3`.

Identical to Spectrum smoothing parameter for the spectral analysis of the displacement time series for Dynamic displacement time series from time domain analysis.

8.4.2.8. Displacement envelope curves

Envelope curves of displacements from time domain analysis are presented as: - Minimum, static and maximum x, y and z displacements for regular analysis - Mean, static and mean + standard deviation for irregular analysis.

Static values are identified as dashed lines while the others are solid.

8.4.2.8.1. Data group identifier, one input line

```
DISPlacement ENVElope CURVes        Plot
```

8.4.2.8.2. Print options, one input line

```
LINE-ID IPDOF1 IPDOF2 IPDOF3
```

- `LINE-ID: character(8)`: Line identifier for which displacements are wanted.
    - You may specify `ALL` to include all lines in the system.
    - The print part of this option will always produce results for all stored degrees of freedom, i.e. x-, y- and z-displacements. The following parameters are used to specify the dof’s to be plotted
- `IPDOF1: integer`: Degree of freedom for first figure
    - `IPDOF1=0`: Not included
    - `IPDOF1=1`: x-displacement
    - `IPDOF1=2`: y-displacement
    - `IPDOF1=3`: z-displacement
- `IPDOF2: integer`: Degree of freedom for second figure.
    - Interpretation as for `IPDOF1`
- `IPDOF3: integer`: Degree of freedom for third figure.
    - Interpretation as for `IPDOF1`

Each figure is presented on separate plot.

8.4.2.9. Force envelope curves

Envelope curves of forces from time domain analysis are presented as:

- Minimum, static and maximum axial force torsional moment or bending moments for regular analysis
- Mean, static and mean + standard deviation for irregular analysis

Static values are identified as dashed lines while the others are solid.

8.4.2.9.1. Data group identifier, one input line

```
FORCe ENVElope CURVes       Plot
```

8.4.2.9.2. Print options, one input line

```
LINE-ID IDOF1 IDOF2 IDOF3
```

- `LINE-ID: character(8)`: Line identifier for which forces are wanted.
  - You may specify `ALL` to include all lines in the system.
  - The print part of this option will always produce results for all stored degrees of freedom, i.e. axial force, torsional moment and bending moments about local y- and z-axes. The following parameters are used to specify the dof's to be plotted
- `IDOF1: integer`: Degree of freedom for first figure.
  - `IDOF1=0`: Not included
  - `IDOF1=1`: Axial force
  - `IDOF1=2`: Torsional moment
  - `IDOF1=3`: Bending moment about local y-axis
  - `IDOF1=4`: Bending moment about local z-axis
  - `IDOF1=5`: Pipe wall force, incl. hydrostatic pressures
    - Pipe wall force is only avaivable for PLOT
  - `IDOF1=6`: Shear force in local y-direction
  - `IDOF1=7`: Shear force in local z-direction
- `IPDOF2: integer`: Degree of freedom for second figure.
  - Interpretation as for `IPDOF1`
- `IPDOF3: integer`: Degree of freedom for third figure.
  - Interpretation as for `IPDOF1`

Each figure is presented on separate plot.

8.4.2.10. Curvature envelope curves

Envelope curves of curvatures from time domain analysis are presented as: - Minimum, static and maximum values of curvatures for a regular analysis - Mean, static and mean + standard deviation for irregular analysis

Static results are dashed, while the others are solid.

8.4.2.10.1. Data group identifier, one input line

```
CURVature ENVElope CURVes       Plot
```

8.4.2.10.2. Print options, one input line

```
LINE-ID IDOF1 IDOF2 IDOF3
```

- `LINE-ID: character(8)`: Line identifier for which curvatures are wanted.
    - You may specify `ALL` to include all lines in the system.
    - The print part of this option will always produce results for all stored degrees of freedom, i.e. local y- and z-curvatures and resulting curvature. The following parameters are used to specify the dof’s to be plotted
- `IPDOF1: integer`: Degree of freedom for first figure
    - `IDOF1=0`: Not included
    - `IDOF1=1`: Curvature about local y-axis
    - `IDOF1=2`: Curvature about local z-axis
    - `IDOF1=3`: Resulting curvature
    - Resulting curvature is taken as the vector sum of the curvatures about local y- and z-axis and will therefore always be positive
- `IPDOF2: integer`: Degree of freedom for second figure.
    - Interpretation as for `IPDOF1`
- `IPDOF3: integer`: Degree of freedom for third figure.
    - Interpretation as for `IPDOF1`

Each figure is presented on separate plot.

8.4.2.11. Support forces

Forces in both ends of specified lines are analyzed and presented in the global coordinate system. Forces due to static and dynamic loads are included. Forces due to hydrostatic pressures are not included, i.e. the axial component is the effective tension.

8.4.2.11.1. Data group identifier, one input line

```
SUPPf TIME SERIes       Plot
```

8.4.2.11.2. Output options, one input line

```
IOP IDOF IT1 NTS NLINC
```

- `IOP: integer`: Code for type of output
  - `IOP=1`: Time series
  - `IOP=2`: Time series statistics
  - `IOP=3`: Spectral analysis
- `IDOF: integer`: Code for degree of freedom
  - `IDOF=1`: Global x-direction
  - `IDOF=2`: Global y-direction
  - `IDOF=3`: Global z-direction
- `IT1: integer`: First stored time step to be included
- `NTS: integer`: Number of stored time steps to be included
- `NLINC: integer`: Number of input lines used for line specifications

For `IOP=3` an FFT analysis is carried out. If `NTS` is not an integer power of 2, a reduced time series will be analysed. In order to get an effective analysis, `IT1` and `NTS` should be selected so that - $\mathrm{IT1} = \mathrm{NT} - 2^{\mathrm{M}} + 1$ - $\mathrm{NTS} = 2^{\mathrm{M}}$

Where NT is the total number of stored time steps and M is the largest integer so that $\mathrm{NTS} <= \mathrm{NT}$. Normally it is preferable to omit the first part of the time series due to transients.

8.4.2.11.3. Element specification, NLINC input lines

```
LINE-ID
```

`LINE-ID: character(8), default: 0`: Line number. You may specify `ALL` to include all lines

8.4.2.11.4. Options for the output distribution functions of the force time series statistics, one input line

This input line is given only if `IOP=2`.

`NCL XCMIN XCMAX`

- `NCL: integer`: No of classes in the output distribution functions (i.e. no of points on the abscissa axis)
    - `0<NCL<41`
- `XCMIN: real`: Range of argument values for output distribution functions is `XCMIN*sx(1) - XCMAX*sx(1)` in which `sx(1)` is the standard deviation of `x` estimated from the first sequence
- `XCMAX: real, default: 0`:

8.4.2.11.5. Spectrum smoothing parameter for the spectral analysis of the force time series, one input line

This input line is given only if `IOP=3`. ~ MSM ~

- `MSM: integer, default: 0`: Smoothing parameter
    - `MSM=0`: No smoothing
    - `MSM>0`: Smoothing by averaging over 2*`MSM`+1 values

8.4.2.12. Element angle time series from time domain analysis

8.4.2.12.1. Data group identifier, one input line

```
ELMAngle TIME SERIes        Plot
```

8.4.2.12.2. Output options, one input line

`IOP IT1 NTS NNELC`

- `IOP: integer`: Code for type of output
    - `IOP=1`: Time series
    - `IOP=2`: Time series statistics

`IOP=3`: Spectral analysis

- `IT1: integer`: First stored time step to be included
- `NTS: integer`: Number of stored time steps to be included (from `IT1`).
    - A large number includes the remaining time steps
- `NNELC: integer`: No. of pairs of input lines used for element specification

Two of the subsequent input lines (Code for element specification and either Global or vessel axis and element specification or Element pair specification) must be given `NNELC` times.

8.4.2.12.3. Code for element specification

`IRELCO`

- `IRELCO: integer`: Code for type of output
    - `IRELCO=0`: Angle between fixed global axis and one specified element
    - `IRELCO=1`: Angle between support vessel coordinate axis and one specified element
    - `IRELCO=2`: Angle between two elements

8.4.2.12.4. Global or vessel axis and element specification

This input line is given only for `IRELCO=0` or `1`. ~ IAXIS IVES LINE-ID ISEG IELM HEAD ~

- `IAXIS: integer`: Code for axis
    - `IAXIS=1`: x-axis
    - `IAXIS=2`: y-axis
    - `IAXIS=3`: z-axis
- `IVES: integer, default: 1`: Vessel number if `IRELCO=1` else dummy
- `LINE-ID: character(8)`: Line identifier
- `ISEG: integer`: Segment number
- `IELM: integer`: Element number
- `HEAD: integer`: Vessel heading in final static position if `IRECLCO=1`, else dummy [deg]

The angle output is between 0 and 180 degrees. If the element direction (from end 1 to end 2) is along the specified axis, the relative angle is 0. Otherwise, if the element direction is along the negative axis direction, the angle is 180 degrees. The element direction is calculated as the direction along the secant from local end no 1 to local end no 2 (i.e. local element x-axis).

8.4.2.12.5. Element pair specification

This input line is given only for `IRELCO=2`. ~ LINE-ID1 ISEG1 IELM1 LINE-ID2 ISEG2 IELM2

~

- `LINE-ID1: character(8)`: Specification of first element
- `ISEG1: integer`:
- `IELM1: integer`:
- `LINE-ID2: character(8)`: Specification of second element
- `ISEG2: integer`:
- `IELM2: integer`:

The angle output is between 0 and 180 degrees. If the element direction (from end 1 to end 2) is along the specified axis, the relative angle is 0. Otherwise, if the element direction is along the negative axis direction, the angle is 180 degrees. The element direction is calculated as the direction along the secant from local end no 1 to local end no 2 (i.e. local element x-axis).

8.4.2.13. Total displacement time series from time domain analysis

Results include the total dynamic displacements (static values are included)

8.4.2.13.1. Data group identifier, one input line

```
TOTDisp TIMe SERIes         Plot
```

8.4.2.13.2. Output options, one input line

```
IOP IDOF IT1 NTS NNODC
```

- `IOP: integer`: Code for type of output
  - `IOP=1`: Time series
  - `IOP=2`: Time series statistics
  - `IOP=3`: Spectral analysis

- `IDOF: integer`: Code for degree of freedom
  - `IDOF=1`: Translation in x-direction
  - `IDOF=2`: Translation in y-direction
  - `IDOF=3`: Translation in z-direction
- `IT1: integer`: First stored time step to be included
- `NTS: integer`: Number of stored time steps to be included (from IT1).
  - A large number includes the remaining time steps.
- `NNODC: integer`: No of input lines used for node specification

For `IOP=3` an FFT analysis is carried out. If `NTS` is not an integer power of 2, a reduced time series will be analysed. In order to get an effective analysis, `IT1` and `NTS` should be selected so that - $\mathrm{IT1} = \mathrm{NT} - 2^{\mathrm{M}} + 1$ - $\mathrm{NTS} = 2^{\mathrm{M}}$

Where NT is the total number of stored time steps and M is the largest integer so that $\mathrm{NTS} <= \mathrm{NT}$. Normally it is preferable to omit the first part of the time series due to transients.

8.4.2.13.3. Node specification, NNODC input lines

Identical to Node specification, NNODC input lines for Dynamic displacement time series from time domain analysis.

8.4.2.13.4. Options for the output distribution functions of the displacement time series statistics, one input line

This input line is given only if `IOP=2`.

Identical to Options for the output distribution functions of the displacement time series statistics, one input line for Dynamic displacement time series from time domain analysis.

8.4.2.13.5. Spectrum smoothing parameter for the spectral analysis of the displacement time series, one input line

This input line is given only if `IOP=3`.

Identical to Spectrum smoothing parameter for the spectral analysis of the displacement time series for Dynamic displacement time series from time domain analysis.

8.4.2.14. Total resulting force time series from time domain analysis

The result force includes both the dynamic time dependent force and the static force.

8.4.2.14.1. Data group identifier, one input line

```
TOTForce TIME SERIes        Plot
```

8.4.2.14.2. Output options, one input line

```
IOP IDOF IT1 NTS NNELC
```

- `IOP: integer`: Code for type of output
  - `IOP=1`: Time series
  - `IOP=2`: Time series statistics
  - `IOP=3`: Spectral analysis
- `IDOF: integer`: Code for degree of freedom
  - `IDOF=1`: Axial force
  - `IDOF=2`: Torsional moment
  - `IDOF=3`: Bending moment about local y-axis, end 1
  - `IDOF=4`: Bending moment about local y-axis, end 2
  - `IDOF=5`: Bending moment about local z-axis, end 1
  - `IDOF=6`: Bending moment about local z-axis, end 2
  - `IDOF=7`: Shear force in local y-direction, end 1
    - Nonlinear dynamic analysis only in present version
  - `IDOF=8`: Shear force in local y-direction, end 2
    - Nonlinear dynamic analysis only in present version
  - `IDOF=9`: Shear force in local z-direction, end 1
    - Nonlinear dynamic analysis only in present version
  - `IDOF=10`: Shear force in local z-direction, end 2
    - Nonlinear dynamic analysis only in present version
  - `IDOF=11`: Axial wall force

- `IT1: integer`: First stored time step to be included
- `NTS: integer`: Number of stored time steps to be included (from `IT1`).
  - A large number includes the remaining time steps
- `NNELC: integer`: No of input lines used for element specification

For `IOP=3` an FFT analysis is carried out. If `NTS` is not an integer power of 2, a reduced time series will be analysed. In order to get an effective analysis, `IT1` and `NTS` should be selected so that - $\mathrm{IT1} = \mathrm{NT} - 2^{\mathrm{M}} + 1$ - $\mathrm{NTS} = 2^{\mathrm{M}}$

Where NT is the total number of stored time steps and M is the largest integer so that $\mathrm{NTS} <= \mathrm{NT}$. Normally it is preferable to omit the first part of the time series due to transients.

8.4.2.14.3. Element specification, NNELC images

Identical to Element specification, NNELC input lines for Dynamic resulting force time series from time domain analysis.

8.4.2.14.4. Options for output distribution functions. Given only if IOP=2

This input line is given only if `IOP=2`.

Identical to Options for the output distribution functions of the force time series statistics, one input line for Dynamic resulting force time series from time domain analysis.

8.4.2.14.5. Spectrum smoothing parameter. Given only if IOP=3

This input line is given only if `IOP=3`.

Identical to Spectrum smoothing parameter for the spectral analysis of the force time series, one input line for Dynamic resulting force time series from time domain analysis.

8.4.2.15. Distance time series calculated from the time domain analyses

This option is mainly to be used in order to perform a check of collision risk between two risers, between a riser and the vessel or between a riser and a fixed structure. The minimum distance is calculated for only a part of the riser. All elements within the specified segments are searched to find this minimum distance at each time step.

Note that the distances are absolute, they are always positive values. The program cannot identify a line crossing situation.

8.4.2.15.1. Data group identifier, one input line

`DISTance TIME SERIes        Plot`

8.4.2.15.2. Output options, one input line

`IOP IT1 NTS IDITYP IMETH IVES XCGVES YCGVES ZCGVES`

- `IOP: integer`: Code for type of output
    - `IOP=1`: Time series
    - `IOP=2`: Time series statistics
    - `IOP=3`: Spectral analysis
- `IT1: integer`: First stored time step to be included
- `NTS: integer`: No of stored time steps to be included (from `IT1`).
    - A large number includes the remaining time steps
- `IDITYP: integer`: Type of distance to be calculated
    - `IDITYP=1`: Distance between specified segments on lines
    - `IDITYP=2`: Distance between specified segments on a line and a globally fixed line
    - `IDITYP=3`: Distance between specified segments on a line and a line fixed on the vessel
- `IMETH: integer, default: 1`: Method option
    - `IMETH=1`: Distance between elements are calculated
    - `IMETH=2`: Distance between nodes are calculated
- `IVES: integer, default: 1`: Vessel number in case of multivessel analysis and `IDITYP = 3`
- `XCGVES: real, default: 0`: Static X coordinate of the vessel in case of `IDITYP = 3`
- `YCGVES: real, default: 0`: Static Y coordinate of the vessel in case of `IDITYP = 3`
- `ZCGVES: real, default: 0`: Static Z coordinate of the vessel in case of `IDITYP = 3`

With the distance, we here mean the minimum distance. All elements within the specified segment(s) are scanned for each time step in order to find the one with the minimum distance.

Method 1 is more accurate, but more time consuming than method 2.

8.4.2.15.3. Specification of segments on lines which the minimum distance should be calculated from, one input line

```
LINE-ID NSEG ISEG1 ISEG2 . . ISEG(NSEG)
```

- `LINE-ID: character(8)`: Line identifier
- `NSEG: integer/character`: No of segments for which the minimum distances are to be calculated from
  - You may specify `ALL` in order to include all segments
- `ISEG: integer`: The included segment numbers

Searching through all elements may cause rather large computation time.

8.4.2.15.4. Specified segments to which the minimum distance are calculated, to be given only if IDITYP=1. One input line

```
LINE-ID NSEG ISEG1 ISEG2 ... ISEG(NSEG)
```

- `LINE-ID: character(8)`: Line identifier
- `NSEG: integer/character`: No of segments for which the minimum distances are to be calculated to
  - You may specify `ALL` in order to include all segments
- `ISEGj: integer`: The included segment numbers

Searching through all elements may cause rather large computation time.

8.4.2.15.5. Specification of a line in the global coordinate system to which the minimum distance are to be calculated, to be given only if IDITYP=2. One input line

```
XG1 YG1 ZG1 XG2 YG2 ZG2
```

- `XG1: real`: Global x-coordinate, end 1
- `YG1: real`: Global y-coordinate, end 1
- `ZG1: real`: Global z-coordinate, end 1
- `XG2: real`: Global x-coordinate, end 2
- `YG2: real`: Global y-coordinate, end 2
- `ZG2: real`: Global z-coordinate, end 2

8.4.2.15.6. Specification of a line in the global coordinate system relative to the vessel reference point to which the minimum distance are to be calculated, to be given only if IDITYP=3

```
XV1 YV1 ZV1 XV2 YV2 ZV2
```

- `XV1: real`: Vessel x-coordinate, end 1
- `YV1: real`: Vessel y-coordinate, end 1
- `ZV1: real`: Vessel z-coordinate, end 1
- `XV2: real`: Vessel x-coordinate, end 2
- `YV2: real`: Vessel y-coordinate, end 2
- `ZV2: real`: Vessel z-coordinate, end 2

8.4.2.15.7. Options for the output distribution functions of the distance time series statistics, one input line

This input line is given only if `IOP=2`.

```
NCL XCMIN XCMAX
```

- `NCL: integer`: No of classes in the output distribution functions (i.e. no of points on the abscissa axis)
  - `0<NCL<41`
- `XCMIN: real`: Range of argument values for output distribution functions is `XCMIN*sx(1) - XCMAX*sx(1)` in which `sx(1)` is the standard deviation of `x` estimated from the first sequence
- `XCMAX: real`:

8.4.2.15.8. Spectrum smoothing parameter for the spectral analysis of the distance time series, one input line

This input line is given only if `IOP=3`. ~ MSM ~

- `MSM: integer, default: 0`: Smoothing parameter
  - `MSM=0`: No smoothing
  - `MSM>0`: Smoothing by averaging over 2*`MSM`+1 values

8.4.2.16. Generate snapshot file from time domain analysis (special option)

This is a special option specified and commissioned by Norsk Hydro, for generation of input files for an animation program used by Norsk Hydro.

Nodes coordinates, element forces and curvatures from dynamic analysis are written to the following files: - `SNAPSNxx.DAT` - Node coordinates - `SNAPFOxx.DAT` - Element forces - `SNAPCUxx.DAT` - Element curvatures

Element forces and/or curvatures will only be written for lines for which the storage coincide with the storage of node displacements.

8.4.2.16.1. Data group identifier, one input line

```
GENERATE SNAPSHOT FILE      Plot
```

8.4.2.16.2. Print options, one input line

```
IT1 NTS IJUMP NLIC NPVESD IVES LFORCE LCURV IASCII XCGVES YCGVES ZCGVES
```

- `IT1: integer`: First stored time step to be included
- `NTS: integer`: Number of stored time steps to be included.
  - You may specify `REST` to include the remaining time step
- `IJUMP: integer`: Include every `"IJUMP"` stored time step
- `NLIC: integer`: No. of input lines to describe the line specification
  - `NLIC=0`: No riser snapshot
- `NPVESD: integer`: No of coordinates to describe the vessel
  - `NPVESD=0`: No vessel snapshot
- `IVES: integer, default: 1`: Vessel number in case of multi-vessel analysis
- `LFORCE: integer, default: 0`: Control parameter
  - `LFORCE=0`: Element forces are not written to file
  - `LFORCE=1`: Element forces are written to file
- `LCURV: integer, default: 0`: Control parameter
  - `LCURV=0`: Element curvatures are not written to file
  - `LCURV=1`: Element curvatures are written to file

- `IASCII: integer, default: 0`: Control parameter
  - `IASCII=0`: Unformatted snapshot files
  - `IASCII=1`: Formatted snapshot files
- `XCGVES: real`: Static X coordinate of vessel CG
- `YCGVES: real`: Static Y coordinate of vessel CG
- `ZCGVES: real`: Static Z coordinate of vessel CG

8.4.2.16.3. Line specification, NLIC input lines

`LINE-ID`

- `LINE-ID: character(8)`: Line identifier to be written to file. You may specify `ALL` to include all the lines in the system

The lines are written only if at least the displacements of the end nodes are stored, see data group File storage of displacement response for storage information.

8.4.2.16.4. Vessel description, NPVESD input lines. The specified points are connected by one line to illustrate a part of the vessel contour

`IPV XVT YVT ZVT`

- `IPV: integer`: Coordinate number
- `XVT: real`: Vessel's X-coordinate in global system, relative to the vessel reference point $[\mathrm{L}]$
- `YVT: real`: Vessel's Y-coordinate $[\mathrm{L}]$
- `ZVT: real`: Vessel's Z-coordinate $[\mathrm{L}]$

The vessel points are in global system, but they are relative to the vessel reference point (the attachment point).

8.4.2.17. Stress time series calculated from the time domain analysis

This option allows for calculation of stresses in circular metallic homogeneous risers.

The stress time series are calculated based on the stored force time series from `DYNMOD` and the component properties specified in `INPMOD`. Stresses may only be calculated for `CRS1` and `CRS0` components.

Stress time series are calculated for specified points on the tube circumference.

8.4.2.17.1. Data group identifier

```
STREss TIME SERIes
```

8.4.2.17.2. Output options, one input line

```
IOP IDOF IT1 NTS ISUBST NNELC
```

- `IOP: integer`: Code for type of output
  - `IOP = 1`: Time series
  - `IOP = 2`: Time series statistics
  - `IOP = 3`: Spectral analysis
  - `IOP = 1` in present version
- `IDOF: integer`: Stress components type 1
  - `IDOF = 1/11`: Axial stress at end 1/2
  - `IDOF = 2/12`: Torsional stress at end 1/2
  - `IDOF = 3/13`: Bending stress at end 1/2
  - `IDOF = 4/14`: Axial + bending stress at end 1/2
  - `IDOF = 5/15`: Shear stress at end 1/2
  - `IDOF = 6/16`: Shear stress + torsional stress at end 1/2
  - `IDOF = 7/17`: Equivalent stress at end 1/2
  - `IDOF = 8/18`: Hoop stress at end 1/2
  - `IDOF = 9/19`: Radial stress at end 1/2
  - `IDOF = 21/22`: External pressure at end 1/2
  - `IDOF = 23/24`: Internal pressure at end 1/2
- `IT1: integer`: First stored time step to be included
- `NTS: integer`: Number of stored time steps to be included (from `IT1`).

A large number includes the remaining time steps

- `ISUBST: integer, default: 0`: Code for subtracting the static stress contributions
  - `ISUBST = 0`: Total stresses calculated
  - `ISUBST = 1`: Static stress is subtracted
- `NNELC: integer`: Number of lines used for element specification

8.4.2.17.3. Point for stress calculation, one input line

```
THETA INEX IOPPRE
```

- `THETA: real, default: 0.0`: Angle from local y-axis for stress calculation $[\mathrm{Deg}]$
- `INEX: integer, default: 2`: Stress location switch
  - `INEX = 1`: Inner wall
  - `INEX = 2`: Outer wall
- `IOPPRE: integer, default: 1`: Code for updating inner and outer pressure values.
  - `|IOPPRE| = 1`: Static inner and outer pressure used.
    - Outer pressure is calculated as hydrostatic pressure from MWL.
  - `|IOPPRE| = 2`: Updated inner and outer pressure used.
    - Outer pressure is calculated as hydrostatic pressure from MWL.
  - `IOPPRE < 0`: Wall forces calculated using outer area given by the pipe diameter or the alternative cross section diameter.
    - Corresponds to evenly distributed shear forces between buoyancy material and pipe.
    - Warning: This option is under development!

Nonlinear time domain analysis only.

In the present version, the external pressure is calculated as a hydrostatic pressure from the MWL. The external pressure is updated for all structural elements.

The internal pressure is updated for all elements that are part of a Main Riser Line.

8.4.2.17.4. Element specification, NNELC input lines

`LINE-ID ISEG IELM`

- `LINE-ID: character(8)`: Line identifier.
  - You may specify `ALL` to include all lines
- `ISEG: integer/character`: Segment number.
  - You may specify `ALL` to include all segments.
  - `ENDS` includes the end segments on the line
- `IELM: integer/character`: Element number.
  - `ALL` includes all elements, and
  - `ENDS` includes end elements on the above specified segment

Stresses may only be calculated for elements for which forces are stored, see data group File storage for internal forces for storage information. If the user specifies elements for which forces are not stored, these elements are ignored.

The data group Storage information may be used to obtain an overview of the stored data.

8.4.2.17.5. Options for the output distribution functions of the stress time series statistics, one input line

This input line is given only if `IOP=2`. ~ NCL XCMIN XCMAX ~

- `NCL: integer`: No of classes in the output distribution functions (i.e. no of points on the abscissa axis)
  - `0<NCL<41`
- `XCMIN: real`: Range of argument values for output distribution functions is `XCMIN*sx(1) - XCMAX*sx(1)` in which `sx(1)` is the standard deviation of `x` estimated from the first sequence.
- `XCMAX: real`:

8.4.2.17.6. Spectrum smoothing parameter for the spectral analysis of the stress time series, one input line

This input line is given only if `IOP=3`. ~ MSM ~

- `MSM: integer, default: 0`: Smoothing parameter
  - `MSM=0`: No smoothing
  - `MSM>0`: Smoothing by averaging over 2*`MSM`+1 values.

8.4.2.18. Stress envelope curves

This option allows for calculation of stress envelopes from the element forces stored in `DYNMOD`, see data group File storage for internal forces for storage information.

The data group Storage information may be used to obtain an overview of the stored data.

8.4.2.18.1. Data group identifier

```
STREss ENVElope CURVes      Noplot
```

8.4.2.18.2. Print options, one input line

```
LINE-ID IDOF1 IDOF2 IDOF3
```

- `LINE-ID: character(8)`: Line identifier
  - `ILINE = ALL`: Stresses in all lines calculated
- `IDOF1:integer`: Stress component type 1
  - `IDOF1 = 1`: Axial stress
  - `IDOF1 = 2`: Torsional stress
  - `IDOF1 = 3`: Bending stress
  - `IDOF1 = 4`: Axial + bending stress
  - `IDOF1 = 5`: Shear stress
  - `IDOF1 = 6`: Shear + torsional stress
  - `IDOF1 = 7`: Equivalent stress
  - `IDOF1 = 8`: Hoop stress
  - `IDOF1 = 9`: Radial stress
- `IDOF2:integer`: Stress component type 2
  - See `IDOF1`
  - Dummy at present
- `IDOF3:integer`: Stress component type 3
  - See `IDOF1`

- Dummy at present

8.4.2.18.3. Stress calculations options, one input line

`TSTA TEND IOP DUR`

- `TSTA: real, default: 0`: Start time in stress time series [T]
- `TEND: real, default: 0`: End time in stress time series [T]
  - `TEND = 0.0`: Until last time step used
- `IOP: integer, default: 0`: Code for envelope type
  - `IOP = 1`: Min and max values presented
  - `IOP = 2`: Maximum range
  - `IOP = 3`: Standard deviations
  - `IOP = 4`: Estimated extreme values (not yet implemented)
- `DUR: real, default: 10800`: Duration used in extreme value estimation [T]
  - Dummy parameter in present version

8.4.2.18.4. Stress calculation location, one input line

`NPCS IOPPR THETA INEX IOPPRE`

- `NPCS: integer, default: See below`: Number of points around the cross-section
  - `= 0`: max stresses estimated
- `IOPPR: integer, default: 0`: Print option
  - `IOPPR = 0`: Print maximum stresses only
  - `IOPPR > 0`: Print stresses at all `NPRCS` points
- `THETA: real, default: 0`: Angle for stress calculation [Deg]
  - Dummy for `NPCS>1`
- `INEX: integer, default: 2`: Stress loction switch
  - `INEX = 1`: Inner wall

    `INEX = 2`: Outer wall

- `IOPPRE: integer, default: 1`: Code for updating inner and outer pressure values.
    - `|IOPPRE| = 1`: Static inner and outer pressure used.
    - `|IOPPRE| = 2`: Updated inner and outer pressure used.
    - Outer pressure calculated as hydrostatic pressure from MWL.
    - `IOPPRE < 0`: Wall forces calculated using outer area given by the pipe diameter or the alternative cross section diameter.
        - Corresponds to evenly distributed shear forces between buoyancy material and pipe.
        - Warning: This option is under development!

Nonlinear time domain analysis only.

The default value for `NPCS` is dependent on the value specified above for `IOP`: Default is `0` for `IOP = 1`, otherwise it is `4`.

In the present version, the external pressure is calculated as a hydrostatic pressure from the MWL. The external pressure is updated for all structural elements.

The internal pressure is updated for all elements that are part of a Main Riser Line.

8.4.2.18.5. Stress calculation parameters, one input line

```
IOPSTR ASTI WSTI DIASTI THSTI EMOD
```

- `IOPSTR: integer, default: 0`: Option for stress calculation
    - `IOPSTR=0`: Stresses calculated from bending moment (recommended)
    - `IOPSTR=1`: Stresses calculated from curvatures
- `ASTI: real, default: 0`: Alternative cross sectional area $[L^2]$
- `WSTI: real, default: 0`: Alternative cross section modulus $[L^3]$
- `DIASTI: real, default: 0`: Alternative cross section diameter $[L]$
- `THSTI: real, default: 0`: Alternative cross section wall thickness $[L]$
- `EMOD: real, default: 0`: Modulus of elasticity $[F/L^2]$

Bending stresses are calculated from curvature, diameter and `EMOD` if `IOPSTR=1` and `EMOD>0`

- $\mathrm{WST} = \frac{2}{\mathrm{EMOD} \times \mathrm{DIAST}}$

The default values of 0 for `ASTI, WSTI, DIASTI, THSTI` and `EMOD` are interpreted as no change from the cross-sectional properties given in `INPMOD`.

8.4.2.19. Riser stroke time series from time domain analysis

The riser stroke is calculated for the supernode specified in `DYNMOD` from the motions of the vessel and the vertical displacement of specified supernode.

This option is not of interest if the terminal point of the riser is vertically fixed to the vessel.

8.4.2.19.1. Data group identifier, one input line

```
STROKe TIME SERIes      Plot
```

8.4.2.19.2. Option to calculate the riser stroke time series, one input line

```
IOP IMOT IDERIV IT1 NTS
```

- `IOP: integer`: Code for type of output
  - `IOP = 1`: Time series
  - `IOP = 2`: Time series statistics
  - `IOP = 3`: Spectral analysis
- `IMOT: integer`:
  - `IMOT = 1`: Stroke
  - `IMOT = 2`: Platform heave motion only
  - `IMOT = 3`: Risers upper end heave motion only
- `IDERIV: integer`:
  - `IDERIV = 0`: Original
  - `IDERIV = 1`: First derivative
  - `IDERIV = 2`: Second derivative

`IT1: integer`: First stored time steps to be included

- `NTS: integer`: Number of stored time steps to be included

8.4.2.20. Code check curves

This option allows for code check of the response.

8.4.2.20.1. Data group identifier

```
CODE CHECk CURVes
```

8.4.2.20.2. Main output options, one input line

```
LINE-ID IOPCOD IOP IDIST DUR PROB
```

- `LINE-ID: character(8)`: Line identifier
  - `LINE-ID = ALL`: All lines checked
- `IOPCOD: integer, default: 1`: Option for type of code check
  - `IOPCOD = 1`: titanium code check
- `IOP: integer, default: 2`: Option for using maximum or estimated extreme values
  - `IOP = 1`: Maximum values from stress time series used
  - `IOP = 2`: Estimated extreme values used
- `IDIST: integer, default: 2`: Distribution type used in extreme value estimation
  - `IDIST = 1`: Rayleigh distribution used
  - `IDIST = 2`: Three parameter Weibull used
  - Dummy for `IOP = 1`
- `DUR: real, default: 10800`: Duration used in extreme value estimation $[\mathrm{T}]$
  - Dummy for `IOP = 1`
- `PROB: real, default: 0`: Probability level used in extreme value estimation
  - `PROB = 0.0`: Expected maximum value used
  - Dummy for `IOP = 1`

8.4.2.20.3. Time range and cross-section points, one input line

`TSTA TEND NPCS IOPPR`

- `TSTA: real, default: 0`: Start time in stress time series $[\mathrm{T}]$
- `TEND: real, default: 0`: End time in stress time series $[\mathrm{T}]$
  - `TEND = 0.0`: Until last time step used
- `NPCS: integer >= 0, default: see below`: Number of points around the cross-section
- `IOPPR: integer, default: 0`: Print option

The default value for `NPCS` is dependent on the value specified above for `IOP`:

Default is `0` for `IOP = 1`, otherwise it is `4`.

8.4.2.20.4. Static load step and load factors, one input line

`ISTEPF GAMF GAMC GAME GAMR`

- `ISTEPF: integer, default: 0`: Static step number for functional loads
  - `ISTEPF = 0`: Final static load step is used
- `GAMF: real, default: 1`: Load factor for functional loads
- `GAMC: real, default: 1`: Load effect factor for condition
- `GAME: real, default: 1`: Load factor for environmental loads
- `GAMR: real, default: 1`: Resistance factor

8.4.2.20.5. Stress calculation parameters, one input line

`SMYS EMOD NU F0 SMYSB TADD`

- `SMYS: real > 0`: Specified minimum yield stress $[\mathrm{F}/\mathrm{L}^2]$
- `EMOD: real > 0`: Modulus of elasticity $[\mathrm{F}/\mathrm{L}^2]$
- `NU: real, default: 0.3`: Poisson’s ratio
- `F0: real, default: 0.005`: Initial ovality
  - $= (\mathrm{D}_{\mathrm{max}} - \mathrm{D}_{\mathrm{min}})/\mathrm{D}$

- `SMYSB: real, default: SMYS`: Specified minimum stress used in axial capacity $[\mathrm{F}/\mathrm{L}^2]$

- `TADD: real, default: 0`: Additional torsion moment $[\mathrm{FL}]$

Typical values of `SMYS` and `EMOD` for steel are in the order of $[\mathrm{SMYS} = 220.0\mathrm{E}3\mathrm{kN}/\mathrm{m}^2] and$ if the units `m' and `kN' were chosen in `INPMOD`.

8.4.2.20.6. Cross-section parameters, one input line

```
ASTI WSTI DIASTI THSTI
```

- `ASTI: real, default: 0`: Alternative cross sectional area $[\mathrm{L}^2]$

- `WSTI: real, default: 0`: Alternative cross section modulus $[\mathrm{L}^3]$

- `DIASTI: real, default: 0`: Alternative cross section diameter $[\mathrm{L}]$

- `THSSTI: real, default: 0`: Alternative cross section wall thickness $[\mathrm{L}]$

The default values of 0 are interpreted as no change from the cross-sectional properties given in `INPMOD`

8.4.2.21. Time domain fatigue damage

This option allows for calculation of fatigue damage calculation from axial and bending stresses in circular metallic homogeneous risers using a specified SN curve and rainflow cycle counting.

The calculated fatigue damage is per year of the specified environmental conditions.

The fatigue damage is calculated based on the stored force time series from `DYNMOD` (see data group File storage for internal forces for storage information) and the component properties specified in `INPMOD`. Stresses may only be calculated for `CRS1` and `CRS0` components.

The fatigue damage is calculated for a specified number of points on the tube circumference.

8.4.2.21.1. Data group identifier, one input line

```
TIMEdomain FATIgue DAMAge       NoPlot
```

8.4.2.21.2. Control data, one input line

```
NSECT NPCS IOPPR TBEG TEND IOPSTR FAT-ID
```

- `NSECT: integer`: Number of riser cross sections to be considered
  - `NSECT = 0`: All cross section where forces are available is included in the analysis

- `NPCS: integer`: Number of points in the cross section where fatigue is calculated
- `IOPPR: integer`: Print option for fatigue results
  - `IOPPR = 0`: Print results only for most critical point in cross section
  - `IOPPR > 0`: Print results for all `NPCS` points
- `TBEG: real`: Beginning of stored stress time series for fatigue calculation Number $[\mathrm{T}]$
- `TEND: real`: End of stored stress time series for fatigue calculation $[\mathrm{T}]$
  - Default is the last stored time step
- `IOPSTR: integer, default: 0`: Option for stress calculation
  - `IOPSTR=0`: Bending stresses calculated from bending moment (recommended)
  - `IOPSTR=1`: Bending stresses calculated from curvatures. EMOD and DIAST must be given
- `FAT-ID: character(16)`: Identifier for fatigue calculation. Used in result presentation only

The remaining of the time series is used if `TEND` is less or equal to `TBEG` (Default is full time series).

8.4.2.21.3. Cross-sectional data, one input line

```
DSCFA DSCFY DSCFZ ASI WSTI DIAST EMOD CFRS LFRS TEFF
```

- `DSCFA: real, default: 1`: Default stress concentration factor for axial force contribution
- `DSCFY: real, default: DSCFA`: Default stress concentration factor for bending about the local Y axis
- `DSCFZ: real, default: DSCFA`: Default stress concentration factor for bending about the local Z axis
- `ASI: real, default: See below`: Optional cross-sectional area $[\mathrm{L}^2]$
- `WSTI: real, default: See below`: Optional section modulus $[\mathrm{L}^3]$. Dummy if stresses are are calculated from curvature (`IOPSTR = 1`)
- `DIAST: real, default: See below`: Cross section diameter. Used to calculate tresses from curvature if `IOPSTR = 1`. Otherwise not used.

`EMOD: real, default: See below`: Modules of elasticity. Used to calculate tresses from curvature if `IOPSTR = 1`. Otherwise not used.

- `CFRS: real, default: 0`: Constant correction coefficient, friction stress $[\mathrm{FL}^{-2}]$
- `LFRS: real, default: 0`: Linear correction coefficient $[\mathrm{L}^{-2}]$
- `TEFF: real, default: See below`: Effective thickness used together with the reference thickness `TREF` given below in thickness correction $[\mathrm{L}]$

The cross-sectional area, modulus and thickness defined for each cross section in `INPMOD` are used as defaults for `ASI`, `WSTI` and `TEFF`.

Stress range correction due to friction is given as:

$$\Delta\sigma_{\mathrm{f}} = \mathrm{CFRS} + \mathrm{LFRS} \times \mathrm{T_{avg}}$$

$\mathrm{T_{avg}}$ is the static value of the tension. The friction stress correction is added after Rainflow counting of the stress time series due to axial force and bending

$$\sigma_{\mathrm{tot}}(\mathrm{t}) = \sigma_{\mathrm{axial}}(\mathrm{t}) \times \mathrm{SCFA} + \sigma_{\mathrm{Y-bending}}(\mathrm{t}) \times \mathrm{SCFY} + \sigma_{\mathrm{Z-bending}}(\mathrm{t}) \times \mathrm{SCFY}$$

The units of the friction correction coefficients must be consistent with the Selection of unit system physical constant in `INPMOD`.

8.4.2.21.4. SN curve data, two input lines

Fatigue capacity curve description ~ NOSL LIMIND FATLIM RFACT TREF KEXP ~

- `NOSL: integer <= 5, default: 1`: Number of straight lines defining the SN curve
- `LIMIND: integer, default: 0`: Fatigue limit indicator
    - `LIMIND < 0`: Fatigue limit in terms of stress cycles is specified
    - `LIMIND = 0`: No fatigue limit for present curve
    - `LIMIND > 0`: Fatigue limit in terms of stress range is specified
- `FATLIM: real, default: 0`: Fatigue limit, interpretation dependent on `LIMIND`. See Example 1: 2 segments and fatigue limit.
    - `LIMIND < 0`: Logarithm of number of stress cycles for which the SN curve becomes horizontal
    - `LIMIND = 0`: `FATLIM` is dummy

`LIMIND > 0`: Stress range level for which the SN curve becomes horizontal $[\mathrm{S}]$

  - See `RFACT` below

- `RFACT: real, default: 1`: Factor between the stress unit $[\mathrm{S}]$ used to define the SN curve and the force and length units $[\mathrm{F}]$ and $[\mathrm{L}]$ chosen in `INPMOD`

  - $\mathrm{S} \times \mathrm{RFACT} = \frac{\mathrm{F}}{\mathrm{L}^2}$

- `TREF: real, default: 0`: Reference thickness for thickness correction $[\mathrm{L}]$. If `TREF = 0` the thickness correction will be omitted.

- `KEXP: real, default: 0`: Exponent for thickness correction

If kN and m were chosen as force and length units while the SN curve is given in MPa, `RFACT` should be set to `0.001`.

If the SI units N and m were chosen for force and length and the SN curve is in MPa, `RFACT` should be set to `1.0E-6`.

Fatigue capacity curve constants ~ RM1 RC1 RMi RNCi … ~

- `RM1: real`: Slope of the SN curve. First curve segment for `NOSL>1`, total curve for `NOSL=1`.

- `RC1: real`: Constant defining the SN curve. First segment or total curve

- `RMi: real`: Slope of curve segment i, i=2, …, `NOSL`

- `RNCi: real`: Transition point between curve segment (i-1), and i, i=2,…, `NOSL`| (log cycles)

See Frequency domain fatigue damage for details of the fatigue curve specification.

8.4.2.21.5. Cross section specification, NSECT input lines

`LINE-ID ISEG IEL IEND SCFA SCFY SCFZ`

- `LINE-ID: character(8)`: Line identifier

- `ISEG: integer >= 0`: Segment number on line

  - `= 0`: All segments in specified line

- `IEL: integer >= 0`: Local element number on specified segment

  - `= 0`: All elements in specified segment

- `IEND: integer`:

    - `IEND = 0`: Cross sections at both ends checked
    - `IEND = 1`: Cross section at end with smallest node number checked
    - `IEND = 2`: Cross section at end with largest node number checked
- `SCFA: real, default: DSCFA`: Stress concentration factor for axial force contribution
- `SCFY: real, default: SCFA`: Stress concentration factor for bending about local Y axis
- `SCFZ: real, default: SCFA`: Stress concentration factor for bending about local Z axis

Time domain forces for the specified elements must be stored in `DYNMOD`, see data group File storage for internal forces for storage information.

The data group Storage information may be used to obtain an overview of the stored data.

If several specifications match an element, the first specification will be used.

8.4.2.22. Time domain longterm data

This option allows for calculation of transfer function modulus and, in the future, also distribution parameters for the stresses from axial and bending force in circular metallic homogeneous risers. The results are intended to be processes in a longterm analysis like in `LONFLX` and `LOSSTA`.

The results are calculated based on the stored force time series from `DYNMOD` (see data group File storage for internal forces for storage information) and the component properties specified in `INPMOD`. Stresses may only be calculated for `CRS1` and `CRS0` components.

The transfer functions are calculated for a specified number of points on the tube circumference.

8.4.2.22.1. Data group identifier, one input line

```
TIMEdomain LONGterm DATA        NoPlot
```

8.4.2.22.2. Control data, one input line

```
NSECT NPCS TBEG TEND
```

- `NSECT: integer, default: 0`: Number of riser cross sections to be considered
    - `NSECT = 0`: All cross section where forces are available is included in the analysis
- `NPCS: integer, default: 16`: Number of points in the cross section where fatigue is calculated

`TBEG: real, default: 0`: Beginning of stored stress time series for fatigue calculation Number $[T]$

- `TEND: real, default: 0`: End of stored stress time series for fatigue calculation $[T]$
  - Default is the last stored time step.

The remaining of the time series is used if `TEND` is less or equal to `TBEG` (Default is full time series).

8.4.2.22.3. Calculation control data, one input line

```
MXFRQ FLOW FHIG IDIST
```

- `MXFRQ: integer`: Maximum number frequencies in the output of transfer functions
- `FLOW: real`: Lower frequency limit in the printing
- `FHIG: real`: Upper frequency limit in the printing
- `IDIST: integer`: Distribution type (Future use)

The actual number of frequencies in the output will usually be somewhat less than `MXFRQ` because the printing is going in integer steps over the calculated Fourier components. The intermediate points is used for smoothing of the transfer function

8.4.2.22.4. Cross sectional data, one input line

```
DSCFA DSCFY DSCFZ ASI WSTI
```

- `DSCFA: real, default: 1`: Default stress concentration factor for axial force contribution
- `DSCFY: real, default: DSCFA`: Default stress concentration factor for bending about Y axis
- `DSCFZ: real, default: DSCFA`: Default stress concentration factor for bending about Z axis
- `ASI: real, default: 0`: Optional cross sectional area
- `WSTI: real, default: 0`: Optional section modulus

The cross sectional area and modulus defined in `INPMOD` is used by default.

8.4.2.22.5. Cross section specification NSECT input lines

```
ILIN ISEG IEL IEND
```

- `ILIN: integer`: Line number
- `ISEG: integer`: Segment number on line
- `IEL: integer`: Local element number on specified segment
- `IEND: integer`:
  - `IEND = 1`: Cross section at end with smallest node number checked
  - `IEND = 2`: Cross section at end with largest node number checked

Time domain forces for the specified elements must be stored in `DYNMOD`, see data group File storage for internal forces for storage information.

The data group Storage information may be used to obtain an overview of the stored data.

8.5. Data Group D: Output from FREMOD

8.5.1. Frequency domain layer damage

This data group may be used to calculate wear and fatigue of tendons in a nonbonded flexible pipe cross section.

8.5.1.1. Data group identifier, one input line

```
FREQuency domain LAYEr DAMAge       NoPlot
```

8.5.1.2. Control data, one input line

```
NLAYER NSECT
```

- `NLAYER: integer`: Number of layers to be considered
- `NSECT: integer`: Number of riser cross sections to be considered

8.5.1.3. Layer data, 2 × NLAYER input lines

Axial stress and friction per unit (pressure/axial force/curvature) ~ IDLAY ALFA1 ALFA2 ALFA3 ALFA4 ALFA5 ~

- `IDLAY: integer`: Unique identification number for the layer data
- `ALFA1: real`: Axial stress in helix per unit pressure (difference)

ALFA2: real: Axial stress in helix per unit axial force

- ALFA3: real: Axial stress in helix per unit pipe curvature $[1/bendingradius]$
- ALFA4: real: Friction stress per unit pressure (difference)
- ALFA5: real: Friction stress per unit axial force

8.5.1.4. Wear and geometrical data; 1 data string:

```
BETA1 BETA2 THICK WSAFE SIGUL SIGLI
```

- BETA1: real: Wear factor per unit curvature and pressure
- BETA2: real: Wear factor per unit curvature and axial force
- THICK: real: Thickness of layer
- WSAFE: real ⇐ 1: Safety factor for wear
- SIGUL: real: Ultimate stress
- SIGLI: real: Limit stress

8.5.1.5. Cross section specification, NSECT input lines

```
LINE-ID ISEG IEL IEND IDLAY1 ... IDLAYn
```

- LINE-ID: character(8): Line identifier (dummy for IEL < 0)
- ISEG: integer: Segment number (dummy for IEL < 0)
- IEL: integer: Element number
  - IEL > 0: local element number
  - IEL < 0: global element number
- IEND: integer:
  - IEND = 1: Cross section at end with smallest node number checked
  - IEND = 2: Cross section at end with largest node number checked
- IDLAY1: integer: First layer to be checked
- IDLAYn: integer: Last layer to be checked

Frequency domain results for the specified element/ends must be stored on the `FREMOD` result file `ifnfre`.

8.5.2. Frequency domain fatigue damage

8.5.2.1. Data group identifier, one input line

```
FREQuencydomain FATIgue DAMAge      NoPlot
```

8.5.2.2. Control data, one input line

```
NOFC NSECT IRES
```

- `NOFC: integer`: Number of SN curves
- `NSECT: integer`: Number of riser cross sections to be considered
- `IRES: integer`: Response print option
  - `IRES>0`: print of total fatigue damage only
  - `IRES<0`: print of fatigue contributions

8.5.2.3. SN data, 2 × NOFC input lines

Fatigue capacity curve description ~ ISNC NOSL LIMIND FATLIM RFACT ~

- `ISNC: integer`: SN curve number - must be given in ascending order
- `NOSL: integer, default: 1`: Number of straight lines defining the SN curve
- `LIMIND: integer, default: 0`: Fatigue limit indicator
  - `LIMIND < 0`: Fatigue limit in terms of stress cycles is specified
  - `LIMIND = 0`: No fatigue limit for present curve
  - `LIMIND > 0`: Fatigue limit in terms of stress range is specified
- `FATLIM: real, default: 0`: Fatigue limit, interpretation dependent on LIMIND. See Example 1: 2 segments and fatigue limit.
  - `LIMIND < 0`: Logarithm of number of stress cycles for which the SN curve becomes horizontal
  - `LIMIND = 0`: `FATLIM` is dummy

`LIMIND > 0`: Stress range level for which the SN curve becomes horizontal

- `RFACT: real, default: 1`: Factor between the stress unit $[\mathrm{S}]$ used to define the SN curve and the force and length units $[\mathrm{F}]$ and $[\mathrm{L}]$ chosen in `INPMOD`
  - $\mathrm{S} \times \mathrm{RFACT} = \frac{\mathrm{F}}{\mathrm{L}^2}$

If kN and m were chosen as force and length units while the SN curve is given in MPa, `RFACT` should be set to `0.001`.

If the SI units N and m were chosen for force and length and the SN curve is in MPa, `RFACT` should be set to `1.0E-6`.

Fatigue capacity curve constants ~ RM1 RC1 RMi RNCi … ~

- `RM1: real`: Slope of the SN curve. First curve segment for `NOSL>1`, total curve for `NOSL=1`
- `RC1: real`: Constant defining the SN curve. First segment or total curve
- `RMi: real`: Slope of curve segment `i, i=2, …, NOSL`
- `RNCi: real`: Transition point between curve segment `(i-1)`, and `i`, `i=2, …, NOSL` (log cycles)

Explanation of the input parameters in input lines `SN data, 2 × NOFC input lines' (above). All parameters are found in figure below.

Example 1: 2 segments and fatigue limit. Note that FATLIM can be alternatively specified](figures/um_io_fig325.svg

Example 2: 3 segments and not fatigue limit. Illustration of input data for fatigue capacity curve definition](figures/um_io_fig326.svg

The SN curves defined by input parameters are always assumed to relate $\Delta\mathrm{S}$ (stress range) to number of cycles before failure.

A straight-lined SN curve in log-log scale is in general defined as

$\mathrm{N} = \mathrm{C} \times \Delta\mathrm{S}^{\mathrm{m}}$

or

$\mathrm{logN} = \mathrm{logC} + \mathrm{m} \times \mathrm{log}\Delta\mathrm{S}$

Where: - N: Number of cycles to failure - $\Delta\mathrm{S}$: Stress range

The two input parameters used to define the SN curves are directly found in the equation above, namely - $\mathrm{RC} = \mathrm{logC}$ (always positive) - $\mathrm{RM} = \mathrm{m}$ (always negative)

If the user has an SN curve without having these parameters explicitly defined, they can be calculated as follows:

](figures/um_io_fig329.svg

Using the two points A and B on the figure to define the straight line, we have

$$\mathrm{logN} = \frac{\mathrm{logN_2} - \mathrm{logN_1}}{\mathrm{log\Delta S_2} - \mathrm{log\Delta S_1}} \times \mathrm{log\Delta S} - \frac{\mathrm{logN_2} - \mathrm{logN_1}}{\mathrm{log\Delta S_2} - \mathrm{log\Delta S_1}} \times \mathrm{log\Delta S_1} + \mathrm{logN_1}$$

Hence: - $\mathrm{RM} = \frac{\mathrm{logN_2} - \mathrm{logN_1}}{\mathrm{log\Delta S_2} - \mathrm{log\Delta S_1}}$ (always negative) - $\mathrm{RC} = -\mathrm{RM} \times \mathrm{log\Delta S_1} + \mathrm{logN_1}$ (always positive)

The relation between these parameters specified for different unit systems is easily found from the equations above.

8.5.2.4. Cross section specification, NSECT input lines

`LINE-ID ISEG IEL IEND SCF IFAT1 ... IFATn`

- `LINE-ID: character(8)`: Line identifier
- `ISEG: integer`: Segment number in line
- `IEL: integer`: Local element number in specified segment
- `IEND: integer`:
  - `IEND = 1`: Cross section at end with smallest node number checked
  - `IEND = 2`: Cross section at end with largest node number checked
- `SCF: real, default: 1`: Stress concentration factor
- `IFAT1: integer`: First SN curve to be checked
- `.`
- `.`
- `.`
- `IFATn: integer`: Last SN curve to be checked

Frequency domain results for the specified element/ends must be stored in `FREMOD`

8.5.3. Frequency domain force results

8.5.3.1. Data group identifier, one input line

```
FREQuency FORCe RESUlts         NoPlot
```

8.5.3.2. Specification of number of sections, one input line

```
NSECT
```

- `NSECT: integer`: Number of sections to be specified

8.5.3.3. Section specification, NSECT input lines

```
LINE-ID ISEG IELM IEND
```

- `LINE-ID: character(8)`: Line identifier
- `ISEG: integer`: Segment number
- `IELM: integer`: Element number
- `IEND: integer`: Element end (1 or 2)

Both local (`LINE-ID SEG ELML`) numbering and global element numbering (`-ELMG`) can be given

This option is valid for linear bending stiffness only, i.e. `IEJ=1` in `INPMOD`.

Example:

parameter/numbering	line-id	iseg	ielm	iend
local:	1	2	4	1
Eqv.global:	0	0	-25	1

8.6. Description of STARTIMES File

8.6.1. Description of STARTIMES file generated by OUTMOD

8.6.1.1. General comments

A time series generated by `OUTMOD` is identified by a time series number and a version number. Response type is identified by the time series number, see description below. The selected node number or element number is identified by the version number.

For one response type versions are numbered `1,2,3,4……N` according to:

- version `1` for 1st selected element/node,
- version `2` for 2nd selected element/node,
- …
- version `N` for last selected element/node

Time series number and version number are printed to `OUTMOD` result file for each selected response. In addition, `FEM` element/node number is included in identification text for each time series stored on `STARTIMES` file.

8.6.1.2. Example

Output of time series of dynamic axial force for element `1, 33, 45` is specified in `OUTMOD`. Identifiers to generated time series are:

- Element 1: 40.01 (time series number 40, version number 1)
- Element 33: 40.02 (time series number 40, version number 2)
- Element 45: 40.03 (time series number 40, version number 3)

8.6.2. Description of time series numbers

Time series number	Contents
WF motion time series	
1	HF surge
2	HF sway
3	HF heave
4	HF roll
5	HF pitch

Time series number	Contents
6	HF yaw
LF motion time series	
7	LF surge
8	LF sway
9	LF yaw
WF and LF motion time series	
10	HF + LF surge
11	HF + LF sway
12	HF + LF yaw
Wave elevation time series	
13	Wave elevation
Wave kinematics	(Not implemented in `OUTMOD`)
14	Water particle velocity: x-direction
15	Water particle velocity: y-direction
16	Water particle velocity: z-direction

Time series number	Contents
17	Water particle acceleration: x-direction
18	Water particle acceleration: y-direction
19	Water particle acceleration: z-direction
Dyndisp time series	
20	Dynamic displacement: global x-direction
21	Dynamic displacement: global y-direction
22	Dynamic displacement: global z-direction
23	Dynamic rotation about: x-direction
24	Dynamic rotation about: y-direction
25	Dynamic rotation about: z-direction
Calcurv time series	
26	Total curvature calculated from nodal coordinates
Element angle time series	
27	Element angle [deg]
	IRELCO = 1: Angle between global z-axis and one element

Time series number	Contents
IRELCO = 2: Angle between support vessel axis and one element	
	IRELCO = 3: Angle between two elements
Distance time series	
28	Distance
	IDITYP = 1: Distance between specified segments
	IDITYP = 2: Distance between specified segments on a line and a globally fixed line
	IDITYP = 3: Distance between specified segments on a line and a line fixed on the vessel
Stroke time series	
29	Stroke
	IMOT = 1: Stroke
	IMOT = 2: Platform heave motion only
	IMOT = 3: Riser heave motion only
Support force time series	
30	Support force component: global x-direction

Time series number	Contents
32	Support force component: global z-direction
Dynforce time series	
40	Dynamic axial force
41	Dynamic Torsional moment
42	Dynamic bending moment about local y-axis: End 1
43	Dynamic bending moment about local y-axis: End 2
44	Dynamic bending moment about local z-axis: End 1
45	Dynamic bending moment about local z-axis: End 2
46	Dynamic shear force in local y-direction: End 1
47	Dynamic shear force in local y-direction: End 2
48	Dynamic shear force in local z-direction: End 1
49	Dynamic shear force in local z-direction: End 2
Dyncurv time series	
50	Dynamic curvature about local y-axis: End 1

Time series number	Contents
53	Dynamic curvature about local z-axis: End 2
Totforce time series	
54	Axial force
55	Torsional moment
56	Bending moment about local y-axis: End 1
57	Bending moment about local y-axis: End 2
58	Bending moment about local z-axis: End 1
59	Bending moment about local z-axis: End 2
60	Shear force in local y-direction: End 1
61	Shear force in local y-direction: End 2
62	Shear force in local z-direction: End 1
63	Shear force in local z-direction: End 2
64	Axial wall force
Totdisp time series	
65	Total displacements in global x-direction

Time series number	Contents
67	Total displacements in global z-direction

Stress time series

75	Axial + Bending stress: End 1
76	Axial + Bending stress: End 2
77	Torsional stress
78	Equivalent stress: End 1
79	Equivalent stress: End 2

9. Appendix A: Frequently Asked Questions

9.1. Modelling

9.1.1. Recommendations regarding pipe-in-pipe modelling

9.1.1.1. General recommendations

The contact between the master pipe and the slave pipe will be applied between a node on the master pipe and an element in the slave pipe. This results in nodal loads along the master pipe and discrete element loads along the slave pipe. This is similar to the roller contact which is described in the Theory Manual.

The contact stiffness, specified as stiffness per unit length along the master pipe, should be chosen large enough, but not stiffer than necessary. When selecting the stiffness it may be useful to consider penetration for a characteristic force, e.g. weight.

A relatively high stiffness proportional damping may be needed to damp out the high natural frequencies caused by stiff contact elements.

The contact force does not contribute to bending moment on the element level. Thus the number of elements in the master and slave pipe should be chosen large enough to reflect the occurring

Note that the contact points in the master pipe are placed at the first end of each element plus an additional contact point at the second end of the last element. If the inner diameter of the master pipe varies, this may not give the desired contact radius at the node where the radius changes. This may be avoided by specifying a separate pipe-in-pipe contact for each segment of the master pipe.

The length of the master pipe should be approximately of the same length as the slave pipe. As an example if the bellomuth is the master, the slave pipe (for instance a riser) of the same length as the master. You may use divide the riser into segments to obtain this. See figures `Example: pipe-in-pipe modelled as `ENTERED’ and `Example: pipe-in-pipe modelled as `ENTERED’ in LOAD GROUP DATA.

9.1.1.2. Input parameters

- `Master position` : The position of the master pipe in the pipe-in-pipe contact pair. This option decides if the outer or inner pipe should be the master pipe. The contact between the master pipe and the slave pipe will be applied between a node on the master pipe and an element in the slave pipe. This results in nodal loads along the master pipe and discrete element loads along the slave pipe. The number of artificial contact elements is equal to number of nodes in the master pipe.
- `Stiffness type` : Either constant (linear), or non-linear contact compression stiffness between the master and slave pipe. In the latter case, a table of force/displacement pairs must be entered. The contact stiffness, specified as stiffness per unit length along the master pipe, should be chosen large enough, but not stiffer than necessary. Very high stiffness values may lead to long run-times, instability and high-frequency numerical noise. When selecting the stiffness it may be useful to consider penetration for a characteristic force and also the convergence of the results that are of interest.
- `Relative damping level` : Relative damping at estimated eigenperiod in the master, slave and contact spring system.
- `Damping` : damping coefficient per unit length of master pipe
- `Spring stiffness` : associated with static friction coefficient : The stiffness used when loading friction forces.
- `Axial friction switch` : This swith controls if the axial friction between master and slave is included.
- `Rotational friction switch` : This swith controls if the rotational friction between master and slave is included.
- `Velocity limit` : for change from static to dynamic friction

9.1.1.3. Recommended default values

The following are recommended default values for setting up a typical pipe-in pipe analysis. It is recommended to always do a parameter variation after setting up a model to ensure convergence of results and that damping does not affect modes of interest. Only a small amount of axial friction is included in the suggested input to improve convergence (compared to turning axial friction off).

9.1.1.3.1. Pipe in pipe parameters

Parameter	Recommended value	Unit
Master position	Outer	(-)
Stiffness type	Linear	(-)
Linear spring stiffness	1e+5	(N/m^2)
Relative damping	0.7-1.0	(-)
Damping	0.0	(Ns/m^2)
Friction stiffness	2.0e+6	(N/m)
Static friction coefficient	1e-3	(-)
Dynamic (sliding) friction coefficient	1e-3	(-)
Axial friction switch	ON	(-)
Rotational friction switch	OFF	(-)
Velocity limit	0.1	(m/s)

9.1.1.4. Recommended static loading sequence

To improve convergence in the static analysis, it is not recommended to include pipe-in-pipe contact forces as the first load group. Loading Volume forces, initially pre-stressed segments etc. before pip-in-pipe forces will in most cases improve static convergence.

9.1.1.5. Tips for efficient analysis with pipe-in-pipe

- Set Pipe-in-pipe contact to ``entered'' in static calculation parameters. The internal pipe does not have to be placed in the exact centre, or even inside of the external pipe, it will be pulled towards the centre when the contact forces are applied.
- Load Volume forces before pipe-in-pipe forces. This will in many cases make the application of contact forces much more stable due to the additional drag forces and weight.
- Setting the external pipe as master will often improve convergence and run-times.
- It is recommended to use shorter elements in the master pipe. The opposite may lead to slower convergence and numerical noise.
- Note that the element mesh of the master pipe controls the number of contact elements and should therefore be chosen carefully with regards to convergence of results and speed. Use a refined mesh where necessary, for example in areas where large curvatures and contact is expected, inside bellmouths/guide-tubes, flexjoints etc. Run times increase exponentially with number of elements in pipe-in-pipe pairs, so it is important to keep number of elements to a minimum, but sufficient to ensure convergence of key results.
- Use linear contact stiffness and start off with a low stiffness for stability and increase gradually if needed.
- For a typical application, drilling-riser, an element length of 2-3 m for both the drillpipe and marine riser will be sufficient. In areas where the drillpipe is bent and large contact forces are expected (for example around flex-joints/ball-joints), the element length should be reduced.
- External contact radius of inner pipe must be larger than zero
- If the master position is ``outer'', all segments in the slave(inner) pipe segments must have external contact radius which is smaller than the internal contact radius of the outer(master) pipe and larger than zero. See illustration in the figure `Pipe-in-pipe master is modelled as outer'.
- If the master position is ``inner'', all segments in the master (inner) pipe segments must have external contact radius which is smaller than the internal contact radius of the outer(slave) pipe and larger than zero. See illustration in the figure `Pipe-in-pipe master is modelled as inner'.

Pipe-in-pipe master is modelled as outer

Pipe-in-pipe master is modelled as inner

9.1.2. Transfer functions - roll, pitch and yaw

In `RIFLEX` vessel roll, pitch and yaw are given as rotation per wave slope; i.e. dimensionless. If values are available as rotation per meter or foot of wave height, a conversion must be performed.

9.1.2.1. Conversion example:

9.1.2.2. Linux script for conversion of rao values:

```
##! /usr/bin/awk -f

##

##  Expects input as:    freq(rad/s)  trf(rad/m)     phase

##  and gives output as: freq(rad/s)  trf(rad/rad)   phase

##

##  Waterdepth may be given on the command line;

##      convert_trf 'd=100.0' <infile>

##

BEGIN { d = 310.0; g = 9.81}

  NF  { w = $1; i = 0;
          k0 = 0.0; k1 = w*w/g;

##       print i, k1;
        while ( (k1-k0) > 0.00005 || (k0-k1) > 0.00005 )
           { i++;
             k0   = k1;
             x    = k0*d;
             tanh = 1.0;
             if (x < 50.) tanh = (exp(x)-exp(-x))/(exp(x)+exp(-x))
             k1   = w*w/(g*tanh);

##             print i, k1
           }
        print $1,$2/k1,$3
      }
```

9.1.3. Example MATLAB script to generate a 3D seafloor grid

The MATLAB scripts `iFundi.m` and `seafloor.m` may be used to generate a regular spaced grid of 3D seafloor data. seafloor.m includes the set of (x,y,z) coordinates that define the 3D seafloor.

9.1.3.1. iFundi.m

```
%
%                                   iFundi
%                                   ______
%
% MATLAB script that generates a RIFLEX seafloor input file, on the basis of
% column based data.
%
% This script is a basis, which is expected to be modified by the user when need-
```

```
ed.
%
% Philippe Maincon, MARINTEK, 28/6/2000
%
clear                                                      % Clear MATLAB's memory, to
```

```
                                                % avoid surprises
% Read the seafloor data

seafloor;                                       % call the script in file
                                                % seafloor.m, which defines
```

3

% vectors x,y and z, of sam-

```
e
                                                     % size.

% Define the grid to be generated

Xsmin = 540;                                         % define the grid to be
DGX   = .5;                                          % generated
Xsmax = 650;                                         % x values on the grid to b-
```

```
e
Ysmin = 80;                                         % between Xsmin and Xsmax, with
DGY   = .5;                                         % step DGX
Ysmax = 140;

% Create the interpolated grid

[b,a]=meshgrid(Ysmin:DGY:Ysmax,Xsmin:DGX:Xsmax);    % for the purpose of
                                                    % interpolation, generate the
                                                    % matrices a and b of x and y
                                                    % data respectively, for each
                                                    % point of the grid.
                                                    % DO NOT switch a and b, x and
                                                    % y. !!!
c=griddata(x,y,z,a,b,'linear');                     % interpolate z data to grid c

% Plot the result - for feedback to the user


figure(1); clf; hold on;axis equal;
mesh(a,b,c);                                        % plot the interpolated grid as
                                                    % a "mesh"
plot3(x,y,z,'k.','markersize',3);                   % plot the original data as
                                                    % black dots
grid;                                               % put a grid as background to
                                                    % the plot

% Write the RIFLEX seafloor input file

depth=round(c*100);
[NGX,NGY]=size(depth)
fnut=char(39);zero=0

file=fopen('bottom.rif','w');                       % You can change the name of
                                                    % the file here
fprintf(file,'%s\n','Sample seabed profile');       % the syntax here is that of
                                                    % C language
fprintf(file,'%s %s\n',fnut,'NGX NGY Xsmin Xsmax Ysmin Ysmax DGX DGY');
% Mind that a RIFLEX input file is never to have more than 80 columns...
fprintf(file,'%4d %4d %8.2f %8.2f %8.2f %8.2f %8.2f %8.2f\n',...
              NGX,NGY,Xsmin,Xsmax,Ysmin,Ysmax,DGX,  DGY);
fprintf(file,'%s %s\n',fnut,'XOS YOS ANGOL');
fprintf(file,'%8.2f %8.2f %8.2f\n',zero,zero,zero);
fprintf(file,'%s %s\n',fnut,'Depth of seabed [ul*100]');
for col = 1:NGY
   for lin =1:NGX-1
      fprintf(file,'%8d %s\n',depth(lin,col),'&');
   end
   fprintf(file,'%8d \n',depth(NGX,col));
end
fclose(file);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% end %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

9.1.3.2. seafloor.m

```
tmp=[  46318.500     62176.000    -134.0976
   46319.500     62176.000    -134.0722
   46320.500     62176.000    -134.0749
   46321.500     62176.000    -134.0644
```

[…] ~ 46427.125 62283.000 -126.0274 46428.125 62283.000 -125.9467];

x = tmp(:,1)-tmp(1,1); y = tmp(:,2)-tmp(1,2); z = tmp(:,3); clear tmp; r1 = x<400; r2 = x>400; ~

9.2. Analysis

9.2.1. Work array size in STAMOD, DYNMOD and OUTMOD

The size of the work arrays may be specified using the environment variables `RIFLEX_STAMOD_MEM`, `RIFLEX_DYNMOD_MEM` and `RIFLEX_OUTMOD_MEM`. The variables give the size in million Bytes, i.e. 4 times the number of million integer words. The minimum size is 4 and the maximum size is 800. From `RIFLEX` 3.6.17 / 3.7.25 the maximum size is increased to 1600. The value used is echoed on the `.res` file.

The default size of the `STAMOD` work array is 4 million integer words. This corresponds to specifying 16.

The default size of the `DYNMOD` work array is 8 million integer words. This corresponds to specifying 32.

The work array in `OUTMOD` may be specified in `RIFLEX` versions 3.4.7 and higher. The default size of the `OUTMOD` work array is 8 million integer words. This corresponds to specifying 32.

The procedure for setting environmental variables is described in Setting Environment variables on Windows and Setting Environment variables on Linux.

The `RIFLEX` for Windows utility should be restarted after setting the environment variables.

9.2.2. Maximum number of arrays on the ifnsta, ifnirr and ifndyn files

The maximum number of arrays on the `ifnsta`, `ifnirr` and `ifndyn` files may be specified using the environment variables `RIFLEX_MAXSTA_IFNSTA`, `RIFLEX_MAXIRR_IFNIRR` and `RIFLEX_MAXDYN_IFNDYN`. The values used are echoed on the `_stamod.res` or `_dynmod.res` file.

The minimum size of `RIFLEX_MAXSTA_IFNSTA` is 2000 and the maximum is 2000000. The default is 20000.

The minimum size of `RIFLEX_MAXIRR_IFNIRR` is 100 and the maximum is 100000. The default is 2000.

The minimum size of `RIFLEX_MAXDYN_IFNDYN` is 50000 and the maximum is 2000000. The default is 200000.

The procedure for setting environmental variables is described in Setting Environment variables on Windows and Setting Environment variables on Linux.

The RIFLEX for Windows utility should be restarted after setting the environment variables.

9.2.3. Transforming time series on file ifndyn

After completing the time domain simulation, DYNMOD reads stored displacements, forces and curvature back from the ifndyn file and extracts times series which are then stored on the ifndyn file for subsequent use by OUTMOD. This will cause a lot of file IO if storage is specified for many response quantities at many time steps and may be very time consuming.

Depending on the available space in the DYNMOD work array, DYNMOD may not be able to transform the time series for all stored nodes / elements at the same time. Check the dynmod.res file for information about the time series transformation, e.g. for a very small test case:

```
Transforming displacements to time series in
1 groups of 50 nodes/elements each
took 0.06 s (wall clock time)
```

If the transformation is split in many groups, increasing the size of the DYNMOD work array may help (ref FAQ Work array size in STAMOD, DYNMOD and OUTMOD).

The amount of data transformed may be reduced by not storing response for nodes / elements that are not needed for post-processing in OUTMOD; e.g. for presentation of times series, response statistics, fatigue damage, distance time series. Note that one can obtain minima, maxima, mean, standard deviation and estimated period on the dynmod.mpf file even if the response quantities are not stored.

If the transformation remains unacceptably time consuming, a solution may be to run the simulation twice. The initial simulation may either be a shorter simulation with full storage or a long simulation without storage. The response quantities of interest may then be selected from the time series or from the key response on the mpf file and the simulation rerun storing only the selected response.

9.2.4. Setting environment variables on Windows

Environmental variables may be used to specify work array size or the maximum number of arrays on files, see Work array size in STAMOD, DYNMOD and OUTMOD and Maximum number of arrays on the ifnsta, ifnirr and ifndyn files. If not set, default values will be used by RIFLEX. For example to increase the DYNMOD work array to twice the default size, set RIFLEX_DYNMOD_MEM to 64.

On Windows 7: 1. Click on Start at the lower left corner of the screen 2. Right mouse click on Computer in the right hand column 3. Choose Properties at the bottom of the menu that pops up 4. Choose Advanced System Settings at the bottom left of the System window 5. Choose

Environment Variables at bottom right 6. Set the desired variables 7. Choose OK to save the settings 8. Restart any command line window in order to update your environment

9.2.5. Running Linux scripts on Windows

The CYGWIN package (http://www.cygwin.com/) will enable you to run Linux-style shell scripts on your Windows machine.

9.3. Extracting Results

9.3.1. Reading RIFLEX results in MATLAB

The contents of the binary additional result files from DYNMOD may be read:

```
fid=fopen('noddis.bin');
F=fread(fid,[156 500],'float32');
F=F';
```

The binary additional result files contain two columns more than the corresponding ASCII files. For RIFLEX version before RIFLEX 3.6.7 (or development versions before 3.7.8), please add 1 to the specified column number and note that an extra column appears after the described columns with data. For RIFLEX 3.6.7 (3.7.8) and higher, the column numbers on the key file are correct.

ASCII file; e.g .res and .mpf files; may be read using fscanf: ~ fid = fopen(armour_sa_ANGLES_outmod.res'); fseek(fid,5176,`bof'); angle1=fscanf(fid,%g %g %g %g %g %g',[6,600]); fseek(fid,766,cof'); angle2=fscanf(fid,%g %g %g %g %g %g',[6,600]); status = fclose(fid); ~

Matrices on a .mpf file may alternatively be read using get_matrix.m.

9.3.1.1. get_matrix.m

```
function matrix=get_matrix(mpffile,title)

%   Matlab function to get the matrices that match the string title
%   from the mpf file mpffile.
%

% *** Open mpf file
fid = fopen(mpffile,'r');


% *** Read entire file into character string

filetext = fscanf(fid,'%c');
fclose(fid);

nlen = length(filetext);
```

```
% *** Find start of all matrices on file + dummy pointer at end
ixmat = findstr('MATRIX',filetext);
nmat = length(ixmat);
ixmat(nmat+1) = nlen;


% *** Find start of all matrices of "title"
text = [ 'MATRIX    ' , title ];
ix0 = findstr(text,filetext);
nmat = length(ix0);


% *** Get matrices from file

nlin = 0;
for i=1:nmat

%  fprintf(1,'%s \n', filetext(ix0(i):ix0(i)+60));

   % Find which matrix
   imat=find(ixmat == ix0(i));

   % Start of values - add 10 to skip 'VALUES    '
   ix1 = ixmat(imat) + findstr('VALUES',filetext(ixmat(imat)+1:ixmat(imat+1))) + 10;

   % Values end at start of next matrix
   ix2 = ixmat(imat+1) - 1;

   values = str2num(filetext(ix1:ix2));

   if (i  == 1)
     all_values = shiftdim(values,1);
   else
   all_values = [ all_values , shiftdim(values,1) ];
   end

end


matrix = shiftdim(all_values,1);
```

9.3.2. Utility programs for STARTIMES files

The utility program `prtsc` may be used to list the contents of a Startimes file.

The utility program `tsprn` may be used to extract time series from a Startimes file.

SINTEF Ocean customers may download the utilities from the MSE e-room.

9.3.3. Stress time series

Stress time series from `OUTMOD` are available in two ways:

- Printed on the `_outmod.res` file if `PRINT` is specified and `STARTIMES` is not specified

- Stored in Startimes format on on the `_outmod.ts` file if `STARTIMES` is specified

If stored in Startimes format, time series may be viewed / accessed in several ways:

- Using the utility MatrixPlot
- Converted to ASCII format using the utility TSPRN (see Utility programs for STARTIMES files below)
- Port-processed using the Startimes package for time series

9.4. Licensing

9.4.1. Getting the FLEXlm HOSTID of a PC

The `RIFLEX` FLEXlm license file is generated for a specific computer, which is identified by its HOSTID. The HOSTID is the MAC address of the primary network card of the PC where the software will be run.

The easiest way to get the MAC address is to run the command `ipconfig/all >out.txt’ from the DOS command prompt. The out.txt file may be opened with a text editor, e.g. Notepad. The MAC address is given by the variable` Physical Address’ (12 alphanumeric characters). Mobile PCs will normally have several network cards and thus several MAC addresses - chose the MAC address under `Local Area Connection', not the one under `Wireless'.

On Windows 7 the DOS command prompt may be started by choosing Start, Search programs and files and then typing `cmd`.

On Windows XP the DOS command prompt may be started by choosing Start, Run and then typing `cmd`. Alternatively, chose Start, All Programs, Accessories and finally Command Prompt.

Send the `out.txt` file or the MAC address to your license provider so that they can generate a license file.

9.4.2. Specifying the FLEXlm license server or file on Windows

The first time `RIFLEX` is run on a PC a dialog will appear and the user may specify a license server or a file location.

Choose and specify either a license server, e.g. `2002@riflex.marin.ntnu.no’ (for non-commercial student use at NTNU) or chose license file and browse to select the license file on your computer, e.g.` C:\SINTEF\ntnu_larsen.lic’

The specified license server or file will then be stored with other registry information. To change the specified FLEXlm server or file, choose `START`, `Run`, `regedit`, `HKEY_LOCAL_MACHINE`, SOFTWARE and then FLEXlm license manager. Click with the right mouse button on MARINTEK_LICENSE_FILE and choose modify or delete.

Several license files or a combination of license files and license servers may be specified by giving a list with semicolons (;) between the entries, e.g. ``C:\SINTEF\ntnu_larsen.lic;@gimli'

9.4.3. FLEXlm error: Terminal Server remote client not allowed

Standalone uncounted licenses are intended for single user use on the PC with the specified HOSTID. They can therefore not be used via remote desktop. The solution is to acquire a server license.

9.4.4. Troubleshooting FLEXlm license server problems

The first step is to check with the IT staff that the license server is running and that the RIFLEX / VIVANA license has been installed.

The second step is to check the specification of the license server on the computer attempting to run RIFLEX / VIVANA. See Specifying the FLEXlm license server or file on Windows (or Specifying the FLEXlm license server or file on Linux). The entry should be set to @ and the name of the license server machine, e.g. @moses or @moses.marintek.sintef.no. It is normally not necessary to specify the port, so don't give a number before the @. Remove any old references to other license servers and/or license files.

The third step is to run the diagnostic utility lmdiag. On Windows start a Command Prompt by choosing START, All Programs, Accessories and finally Command Prompt. Run ~ lmutil lmdiag -c @SERVER -n RIFLEX_INPMOD ~ replacing SERVER with the name of your license server. This will allow you to check your connection with the license server and whether the RIFLEX / VIVANA license has been successfully installed. You may alternatively run ~ lmutil lmdiag -c @SERVER -n ~ to obtain a list of licenses that are available from the server.

When the specified license is available, a This license can be checked out'' message will appear. Note that a number of No licenses for …" messages may also appear in the list, e.g.

```
lmutil - Copyright (c) 1989-2003 by Macrovision Corporation. All rights reserved
```

```
.
FLEXlm diagnostics on Tue 11/11/2008 11:26

-----------------------------------------------------
License file: 27000@moses
-----------------------------------------------------
No licenses for RIFLEX_INPMOD in this license file
-----------------------------------------------------
License file: 27001@moses
-----------------------------------------------------
"RIFLEX_INPMOD" v1.0, vendor: MARINTEK
License server: moses
floating license expires: 15-jan-2009

This license can be checked out
-----------------------------------------------------
-----------------------------------------------------
License file: 27002@moses
-----------------------------------------------------
No licenses for RIFLEX_INPMOD in this license file
```

Trying to obtain a license that is not available results in only ``No licenses for…" messages for all ports.

Specifying an unreachable server name will result in only the first two lines of text; the line with the lmutul copyright information and the current date and time.

9.5. RIFLEX for Windows Utility

9.5.1. Specifying program modules in RIFLEX for Windows

The location of the `RIFLEX` program modules is stored on the `RIFLEX4Win.INI` file which is located on the same directory as `RIFLEX4Win.exe`. If the location is not specified on the `RIFLEX4Win.INI` file, e.g., when `RIFLEX` for Windows is started for the first time, the location is set to `RIFLEX_HOME\bin` using the environmental variable `RIFLEX_HOME`. The `INI` file `RIFLEX4Win.INI` is updated when `RIFLEX` for Windows is closed.

The location may thus be reset by replacing the `RIFLEX4Win.INI` file, resetting the `RIFLEX_HOME` environmental variable and then restarting `RIFLEX` for Windows. The `RIFLEX4Win.INI` file may be re-downloaded or edited by the user.

Overwriting the `INI` file may also be necessary if the directory where `RIFLEX` was last run is no longer available.

Alternatively, the path to the executables may be specified in `RIFLEX` for Windows. Select `Options` and then `Programs` and set the correct folder name for `HOME` and the individual program modules `INPMOD`, `STAMOD`, … .

The same procedure is applicable for `VIVANA` for Windows. This allows the user to have separate `RIFLEX` and `VIVANA` installations, using the correct `INPMOD` and `STAMOD` modules in each.

9.5.2. Run-time error 9: subscript out of range

This error message indicates that `RIFLEX` for Windows cannot find the expected key data for plotting. It is usually caused by empty or incomplete `.mpf` files and may occur after running an analysis or when starting `RIFLEX` for Windows or selecting a new directory.

The immediate remedy is usually to delete the offending files with the `mpf` suffix on the directory.

Check the `RIFLEX` / `VIVANA` `.res` files to see if the problem is caused by an error in the analysis; e.g. the analysis has been interrupted or failed. It may also occur after a successful analysis if no plots were stored on the `.mpf` file; e.g. pure Catenary `STAMOD` analyses, static analyses with the parameter `IFILFM = 0`, dynamic simulations that are shorter than the specified start time for response envelope calculation or where envelopes are not calculated or stored. Please check the `STAMOD` data group `STAMOD CONTROL INFORMATION` and the `DYNMOD` data groups `ENVELOPE CURVE SPECIFICATION` and `REGWAVE PRINT OPTIONS`.

9.6. Linux

9.6.1. Specifying the FLEXlm license server or file on Linux

Before running `RIFLEX` for the first time the environment variable `MARINTEK_LICENSE_FILE` must be set to the license server or file location, e.g. ~ MARINTEK_LICENSE_FILE=`@flexlm''` ~ ~ `MARINTEK_LICENSE_FILE`=/home/ep/riflex_test.lic" ~ or (for non-commercial student use at NTNU) ~ MARINTEK_LICENSE_FILE=``2002@riflex.marin.ntnu.no" ~

This value will be stored on the file `.flexlmrc` in the user’s home directory the first time one runs `RIFLEX`. This file may later be edited if necessary.

Several license files or a combination of license files and license servers may be specified by giving a list with semicolons (;) between the entries, e.g., ~ MARINTEK_LICENSE_FILE=``/home/ep/riflex_test.lic;@flexlm" ~

9.6.2. Setting Environment variables on Linux

Environmental variables be used to specify work array size or the maximum number of arrays on files, see Work array size in STAMOD, DYNMOD and OUTMOD and Maximum number of arrays on the ifnsta, ifnirr and ifndyn files. If not set, default values will be used by `RIFLEX`. For example to increase the `DYNMOD` work array to twice the default size.

To set in a Bourne / Korn / Bash shell ~ export RIFLEX_DYNMOD_MEM=``500" ~

10. RIFLEX Release Notes

RIFLEX 4.20.0 (2020-XX-XX)

Initial release `RIFLEX` 4.20.

RIFLEX 4.20.0 New / improved functionality

Nacelle yaw control (Under development)

Nacelle yaw control is implemented. The controller rotates the entire rotor-nacelle assemble (RNA), aligning it with the mean wind direction.

Coriolis centripal loads.

Coriolis centripetal load has been added for general cross sections CRS7, i.e. the effect of rotational speeds about 2 local element axes that gives rise to moment about the 3rd axis. This functionality was mainly implemented for analysis of operational wind turbines. It is normally insignificant for analysis with moderate dynamic rotational speeds; e.g. normal riser analysis.

Structural damping improvements

The cross sectional damping specified in `INPMOD` can now differ for axial, torsional and bending dofs in `VIVANA`.

The curvature-related damping specified in `VIVANA` is applied to local bending only.

Input to the internal wind turbine controller

The input to the internal wind turbine controller is now read with the FREAD package which is used for other RIFLEX input.

Visualization of regular wave

Visualization of regular wave is enabled.

The visualization is an approximation of the input wave used in the analysis. A single component wave spectrum is used to generate the wave and slight deviation in period may occur.

RIFLEX 4.20.0 Corrected errors

Mass-proportional damping

If the added mass of an attached body is given in the global system, it has not contributed to the Rayleigh mass-proportional damping.

Also, the MacCamy & Fuchs added mass from radiation force and the added mass at infinite frequency from potential wave flow theory have not contributed to the Rayleigh mass-proportional damping.

This is corrected and these contributions are now included.

Note that mass-proportional damping should be used with caution as it will give damping for long response periods.

RIFLEX 4.20.0 Known issues

No known issues in the 4.20.0 version.

RIFLEX 4.20.0 Input changes

Most input files used in version 4.18 can be used unchanged. The exceptions are described below.

External wind turbine pitch controller

The input to external wind turbine pitch controller is changed in version 4.19.1.

Removed some unit-dependent default values

Removed the default values for - air density (AIRDEN) and - water density (WATDEN) in the ENVIRONMENT CONSTANTS data group in INPMOD - minimum yield stress (SMYS) and - modulus of elasticity (EMOD) in the CODE CHECK CURVES data group in OUTMOD. The user must now specify these values.

The previous default values for AIRDEN and WATDEN were only valid if the units `m' and `kg' are used.

The previous default values for SMYS and EMOD were only valid if the units `m' and `kN' are used.

RIFLEX 4.20.0 Removed functionality

No previously available functionality have been removed in this version.

RIFLEX 4.20.0 Deprecated functionality

No functionality is deprecated in this version.

RIFLEX licensing

RIFLEX is license-managed using the FLEXlm / FLEXlnet software license management system. If you want RIFLEX to be used from any networked computer on your site, you must run a license manager on a server in your network. Alternatively, RIFLEX may be run on a single computer using a standalone license file.

Please note that version 4.2 and higher requires a licence file with a feature version that is equal or larger than the link date.

In order to issue a server license or a standalone license file, SINTEF Ocean or DNV GL needs the following info on your server:

- License type (server or standalone)
- Operating system and version (Windows 7, Windows XP, HP-UX and Linux currently supported)
- MAC address / FLEXlm hosted of the computer.

Your IT-staff is probably already familiar with this procedure as FLEXlm is used by a large number of other applications (e.g. Matlab).

RIFLEX version numbers

The version number consists of three numbers separated by periods, e.g. 4.16.0. The two first are the version. The third is updated for each subsequent (bug fix) release.

Even numbered versions, e.g. 4.14, 4.16, are reserved for official versions.

Odd numbered versions, e.g. 4.15, 4.17, are reserved for development versions. The next official release will therefore be 4.22.

RIFLEX 4.18.2 (2020-XX-YY)

Bugfix release of RIFLEX 4.18.

RIFLEX 4.18.2 New / improved functionality

Maximum number of arrays in ifndyn

Increase maximum arrays in ifndyn to 100million.

RIFLEX 4.18.2 Corrected errors

MacCamy-Fuchs loads for regular waves given as a tabulated spectrum

When MacCamy-Fuchs wave loads were calculated for a regular wave given as a tabulated wave spectrum, numerical inaccuracies could lead to the wave frequency being missed in the calculations. This resulted in the MacCamy-Fuchs loads being zero.

Error since RIFLEX 4.18.1, but also present in 4.16.7. The error has been corrected. Some changes in results may occur. The changes will be insignificant if the length of the pre-generated wave time series is reasonably long.

Effect of eccentric mass center on inertia forces not always included

Morison hydrodynamic loads had to be specified to include the effect of eccentric mass center in calculating inertia forces.

It is in general allowed to specify no hydrodynamic forces

The error applied to lumped formulation only.

Potential flow library file name

The potential flow library file name can now start with slash.

RIFLEX 4.18.1 (2020-06-01)

Bugfix release of `RIFLEX` 4.18.

RIFLEX 4.18.1 New / improved functionality

Improved MacCamy-Fuchs wave load generation

The performance in pre-generation of MacCamy-Fuchs wave loads has been significantly improved.

RIFLEX 4.18.0 (2020-03-23)

Initial release `RIFLEX` 4.18.

RIFLEX 4.18.0 Known issues

RIFLEX 4.18.0 New / improved functionality

Implementation of fibre rope characteristics

The SYROPE model for fibre ropes has been implemented in the new cross section component FIBR.

Reference: Falkenberg et. al.: The SYROPE Method For Stiffness Testing Of Polyester Ropes. In: Proc. of the ASME 2018 37th International Conference on Ocean, Offshore and Arctic Engineering (OMAE). 2018

Requirements to identical wind turbine blades

The requirement to identical blades in a wind turbine model have been relaxed. The wind turbine blades have still to be identical with regarding the element distribution, foil profile description and aerodynamic coefficients along the blades. However, the mass and stiffness distribution may be different.

Improved Rayleigh damping

An option to apply stiffness proportional damping based on material stiffness only has been implemented. This option should be used for wind turbine blades because the geometric stiffness will introduce damping of the rigid body motion. This will also ensure symmetrical behaviour of the blades when constant Rayleigh damping is specified.

Improve name on output signal `Electrical generator torque`

A signal named `Electrical generator torque` is a part of the Wind Turbine output. This is however not on the high speed shaft, but on the low speed shaft.

The name is changed from `Electrical generator torque` to `Mechanical generator torque on LSS`

Interpolation and extrapolation of airfoil coefficients

Airfoil coefficients may now be given for a single Reynolds number. They will then be used for all Reynolds numbers.

Extrapolation outside the range of Reynolds values given will now be allowed. The value for the closest Reynolds number will be used; i.e. flat extrapolation.

More efficient import of airfoil library

The import of the airfoil library for foil load coefficients has been improved. This will reduce the analysis time for both static and dynamic analysis. The reduction will be most noticeable in static analysis.

Choice of method for generating random numbers

The algorithm for generating pseudo-random numbers may be selected by the user. The ``mersenne twister'' is the recommended method and should be used unless backwards compatibility with previous versions is required. Note that the default value may change in a future release. The choice of random number generator will apply to: - generation of irregular wave time series - initial phase angles for time domain VIV loads specified for cross-sections in INPMOD - generation of phase angles for application of harmonic loads from a VIVANA frequency domain analysis.

It has been identified that the legacy method can give non-gaussian and non-stationary wave elevation in SIMO for short crested waves with more than about 30-50 discrete directions, depending on wave spectrum and simulation duration. By choosing the mersenne twister, these issues are avoided.

For coupled analysis, wave time series will be generated using the random number generator specified in SIMA.

RIFLEX 4.18.0 Input changes

Rayleigh damping

Stiffness-proportional damping was previously based on only the material stiffness for elements that were - axisymmetric cross section (CRS1) specified with with hysteresis generated by an internal friction moment at reversed curvature - general cross section (CRS7)

To reproduce dynamic results for models containing either of these, the stiffness contribution to damping has to be specified as based on the material stiffness only.

Input to the internal wind turbine controller

The input to the internal wind turbine controller is now checked more thoroughly. Some input that was previously accepted gave errors in the subsequent analysis. Illegal input will now give an error message and program termination.

RIFLEX 4.18.0 Corrected errors

MacCamy-Fuchs radiation contribution from dry elements

Corrected an error that caused MacCamy-Fuchs radiation forces (inertia and dissipation forces due to specified added mass and damping coefficients) to contribute even for elements above the mean water level (dry elements).

Abrupt stop during generation of second order waves

Corrected an error that could cause abrupt stop during generation of long second order wave time series.

Possible error in hydrodynamic loads for simulations without current

In rare cases without current, uninitialized values could occur for current velocity at the nodes. These could potentially result in incorrect hydrodynamic loads in dynamic simulations and thus to incorrect dynamic results.

The error was avoided if a current with zero velocity was included in the simulation.

Longstanding error

Coupled analysis with 6-dof SIMO body and only bar elements

Corrected an error in coupled analysis when a 6-dof SIMO body is attached to an element system that only contains bar elements. The error could cause overwriting of nodal coordinates and give incorrect results.

No error if at least one element in the system has bending stiffness, for example if a stiff beam element was used to connect the SIMO body to the rest of the element system.

Error since RIFLEX 4.16.0.

Tabulated wave spectrum with more than 4 values

Corrected the error that meant that only the first four values of a tabulated wave spectrum were used.

Error since RIFLEX 4.10 (2017-03-21).

Boundary change in the first step of static analysis

If a boundary change was used to change a master node from fixed to free on the first step of static analysis, the slave nodes would end up with the same coordinates as the master node.

Longstanding error.

MacCamy & Fuchs wave excitation load with simplified radiation load model

The damping contribution to the radiation load was not included. This has been fixed. Both the added mass and damping terms now contribute to the radiation loads.

Error since RIFLEX 4.14.0.

Avoid NaN from Torsethaugen wave spectrum

The peakedness factor gamma in Torsethaugen spectrum is restricted to be greater than, or equal to, 1.0 since lower values caused an illegal numerical operation and result in NaN in results. The problem has been encountered for seastates with extremely small significant wave height.

Long-standing error

Riflex specified inner pressure always applied at end 1

The inner pressure is now applied at end 2 if specified. The pressure can be changed for more than one MRL.

Time domain VIV loads and inconsistent units (restricted functionality)

The time domain VIV loads were incorrect if inconsistent mass and force units were used, i.e. GCONS different from 1.0. The error would normally be obvious, e.g. with the units kN and kg, GCONS is 0.001 and the applied loads were too large by a factor of 1000.

The error was avoided when running in SIMA as SIMA always uses a consistent set of units.

Error since RIFLEX 4.14.0.

In-line term of the time domain VIV loads (restricted functionality)

The in-line term of the time domain VIV loads has been calculated using the relative velocity at the first end of the element at both ends. This has been corrected.

Note that the time domain VIV load model is currently restricted functionality and that a license is required.

Error since RIFLEX 4.14.0.

RIFLEX 4.16.4 (2020-03-03)

Bugfix release with minor corrections and improvements.

RIFLEX 4.16.4 Corrected errors

Coupled analysis with 6-dof SIMO body and only bar elements

Corrected an error in coupled analysis when a 6-dof SIMO body is attached to an element system that only contains bar elements. The error could cause overwriting of nodal coordinates and give incorrect results.

No error if at least one element in the system has bending stiffness, for example if a stiff beam element was used to connect the SIMO body to the rest of the element system.

Error since RIFLEX 4.16.0.

RIFLEX 4.16.3 (2020-02-06)

Bugfix release with minor improvements.

RIFLEX 4.16.3 New / improved functionality

Interpolation and extrapolation of airfoil coefficients

Airfoil coefficients may now be given for a single Reynolds number. They will then be used for all Reynolds numbers.

Extrapolation outside the range of Reynolds values given will now be allowed. The value for the closest Reynolds number will be used; i.e. flat extrapolation.

More efficient import of airfoil library

The import of the airfoil library for foil load coefficients has been improved. This will reduce the analysis time for both static and dynamic analysis. The reduction will be most noticeable in static analysis.

RIFLEX 4.16.3 Corrected errors

Tabulated wave spectrum with more than 4 values

Corrected the error that meant that only the first four values of a tabulated wave spectrum were used.

Error since RIFLEX 4.10 (2017-03-21).

RIFLEX 4.16.2 (2019-11-12)

Bugfix release with RIFLEX corrections.

RIFLEX 4.16.2 Corrected errors

MacCamy & Fuchs wave excitation load with simplified radiation load model

The damping contribution to the radiation load was not included. This has been fixed. Both the added mass and damping terms now contribute to the radiation loads.

Error since RIFLEX 4.14.0.

RIFLEX 4.16.1 (2019-08-12)

Bugfix release with `RIFLEX` and `SIMO` corrections.

RIFLEX 4.16.1 New / improved functionality

Linearized time domain simulation with Simo elements

Simo body system elements, i.e. Floater Forces Model Data, may now be used in linearized time domain simulations.

This is a new implementation and should be used with caution.

RIFLEX 4.16.1 Corrected errors

Hydrodynamic interaction between SIMO bodies

Corrected an error that caused restriction in the number of SIMO bodies that could interact hydrodynamically. The simulation failed if the number of bodies was larger than 3.

Dynamic Current Variation

Corrected an error that caused simulations with dynamic current variation to fail unless the selected environment contained at least one irregular wave.

Long-standing error.

RIFLEX 4.16.0 (2019-05-06)

Initial release of `RIFLEX` 4.16.

RIFLEX 4.16.0 Known issues

Wave kinematics at updated position in coupled analysis

Currently, the FFT wave method must be selected in `SIMO` to allow wave kinematics at updated position to be used in `RIFLEX` for a coupled analysis.

Main Riser Line inner pressure

The Main Riser Line (MRL) input allows the pressure to be specified at either end of the MRL. The specified inner pressure is, however, always applied at end 1 of the MRL. This will give errors in true wall tension and some stress components calculated in OUTMOD.

Long-standing error.

Specified loads at nodes with skew boundary conditions

Specified nodal loads at nodes with user-defined skew boundary conditions are not handled correctly. The loads in the global directions are applied in the skew system. A warning is written to the .res file.

Nodes connected to a flex-joint will have degrees of freedom in a skew system. Specified loads at these nodes are not handled correctly. An error message will be written to the .res file and the analysis stopped.

RIFLEX 4.16.0 New / improved functionality

Visualization of airfoil profiles

Visualization is now enabled also for foil profiles not part of a wind turbine. The foil profiles have all to be described in a airfoil-library file.

Time domain fatigue calculations in OUTMOD

A thickness correction based on a reference thickness and an exponent may now be included in the calculation of the fatigue capacity.

The calculated fatigue damage is now printed on the _outmod.mpf file.

The user may now specify an identifier for the fatigue calculations and may also specify all segments / elements / ends for fatigue calculations.

The criteria for skipping fatigue calculations and print at the first end of elements has been modified. Fatigue is now always calculated for all locations specified in OUTMOD. If fatigue is calculated for all elements with stored forces; i.e. NSECT = 0; fatigue calculation at the first end of an element is skipped if calculation was performed at the second end of the neighbouring element in the same segment.

Previously, fatigue was only calculated at the first end of the first element with fatigue calculation in each segment. As forces may not be stored for all elements within a segment, this may skip more nodes than intended.

OUTMOD input array size

The size of the integer and real input arrays in OUTMOD has previously been fixed to 1000 and 500 places. This is now increased to a minimum of 10000 and 6000 places and will be linearly increased if the OUTMOD work array size is larger than the default size.

This change decreases the available space for other arrays and may cause a previously successful OUTMOD run to fail. Increasing RIFLEX_OUTMOD_MEM by one (million Bytes); e.g. from the default 32 to 33; will solve this.

External wind turbine controller measurements

The coordinate system for the measurements is now based on the stress-free orientation of the shaft element where the electrical torque is applied. It was previously based on this element’s orientation at the end of static analysis. No change in results unless the shaft rotates around the global z-axis during the static analysis.

A minor change has been made to the additional nodal measurements for an external wind turbine controller that were added in RIFLEX 4.14.0. The nodal rotations are changed from the updated orientation of the neighbouring element to the rotations of the specified node from the stress-free to the current orientation.

Wave loads on elements

The case that wave loads on elements are specified but no points for computation of wave kinematics are detected, will no longer lead to controlled program termination.

Extended specification of pre-curved line type

Transverse offsets may now also be specified for the second end of the last segment within a line type. See Transverse offset specification in INPMOD.

Boundary change for supenodes

Boundary change may be specified directly for a supernode. This is in addition to the existing boundary change for local segment nodes.

Simo body system element

The Simo body system element is introduced to make it possible to acount for hydrodynamic interaction between bodies in a Riflex ``coupled'' analysis. The element nodes are of Simo body type, a new type of supernode. One Simo body is automatically attached to one simo body node having the same number of degrees of freedom as the Simo body. The Simo body system element may have one or more simo body nodes dependig how many bodies that interact.

The simo body supernode may be attaced to any already existing FE-node, or be used as master in a rigid super node connection.

New file format for external dynamic nodal forces

The nodal forces can be given in a column format.

Rigid supernode connections in linear time domain analysis

Rigid supernode connections may now be used in in linear time domain analysis

Visualization for linear time domain analysis

Visualization is enabled for linear time domain analysis.

Bash shell script for running coupled analysis

A Bash shell script for running a complete coupled analysis has been added to the share/bin folder.

RIFLEX 4.16.0 Corrected errors

Eccentric mass center CRS7 (General Cross Section)

Error in system set up. Local line segment number used instead of global segment number for eccentric mass center; Cross sections of type CRS7.

This caused incorrect results if eccentric mass was specified for CRS7 cross-sections on lines other than the first line specified. The error has been corrected.

Error since 4.10.0

No wave forces in combination with export for visualization

The combination of no wave forces acting on elements and export for visualization leads to uncontrolled program termination. The error has been corrected.

Nodal component at end 2 in linear time domain

Include loads for nodal components at end 2 of a segment in linear time domain simulation. These were previously not included if the segment had more than one element.

Long-standing error

Potential flow and MacCamy-Fuchs loads

Neither Potential flow loads nor MacCamy-Fuchs loads are currently implemented for regular wave analysis. The is now checked and DYNMOD will give an error message and stop.

An alternative is to replace the regular wave with a tabulated (``numerical") wave spectrum with a single frequency component.

Kinematic viscosity in air is changed

The default value for viscosity in air is set to 1.516E-5, air temperature 20 degree centigrades.

Error in boundary change during parameter variation in static analysis

Error in static analysis if boundary change was specified during parameter variation. The error normally caused program termination, but could give error in results if the energy norm was used for solution accuracy.

Error since 4.12.0

Error in boundary change during parameter variation in static analysis

Error if a node is specified to become a slave node during static parameter variation.

Long-standing error

IEC extreme wind events with detailed specification

Corrected an error in the initialization of the IEC 2005 wind events - extreme vertical wind shear, EWSV - extreme horizontal wind shear, EWSH - extreme operating gust, EOG - extreme direction change, EDC The wind events were incorrectly initialized when defined by a detailed specification; i.e. velocity or direction change and event duration. The error resulted in the event duration being zero and no change in wind.

Error since RIFLEX 4.12.0.

IEC extreme wind events for wind turbine class S

Corrected an error in the initialization of the IEC 2005 wind events for wind turbine class S. The expected turbulence intensity, Iref, was incorrectly set to the reference wind speed, Vref.

Error since RIFLEX 4.12.0.

Error in time domain fatigue for systems with only bar elements

Corrected an error in fatigue calculations for systems with only bar elements.

Long-standing error.

Correct echo of Froude-Krylov scaling factor for time domain VIV

The echo on the _inpmod.res file of the Froude-Krylov scaling factor in the normal direction was incorrectly identified as in the tangential direction.

Note that the time domain VIV load model is currently restricted functionality and that a license is required.

Error since RIFLEX 4.14.0.

Time domain VIV loads and consistent formulation (restricted functionality)

The time domain VIV loads may now be used with consistent mass / load formulation.

Note that the time domain VIV load model is currently restricted functionality and that a license is required.

Error since RIFLEX 4.14.0.

RIFLEX 4.16.0 Input changes

Most input files used in version 4.14 can be used unchanged. The exception is described below.

Coupled analysis with IUPPOS = 0

The option IUPPOS = 0, hydrodynamic loads calculated with lines kept fixed in static position and no update of surface penetration, is no longer available for analyses with SIMO bodies.

RIFLEX 4.16.0 Removed functionality

No previously available functionality have been removed in this version.

RIFLEX 4.16.0 Deprecated functionality

Default values for air and water density (INPMOD)

The default values for air and water density in the ENVIRONMENT CONSTANTS data group in INPMOD will be changed. The current default values of 1.3 and 1025 for air and water respectively are only applicable if the units `m' and `kg' are used.

In RIFLEX 4.18.0, the current default values will be used if `kg' is specified for the mass unit. Alternative default values will be used if `Mg' is specified. For other mass unit names, teh air and water density must be specified by thee user.

Default values in CODE CHECK CURVES (OUTMOD)

Two default values will be removed in the OUTMOD data group CODE CHECK CURVES in RIFLEX 4.18.0. - minimum yield stress, SMYS, and - modulus of elasticity, EMOD.

The current default values of 220.0E3 and 2.1E8 are only applicable if the units `m' and `kN' are used.

CARISIMA riser-seafloor contact (restricted functionality)

The CARISIMA riser-seafloor contact is deprecated and will be removed in the future.

Old OUTMOD fatigue data group

The old, undocumented OUTMOD data group TIME FATIGUE LIFE will be removed following the 4.16 release.

The OUTMOD data group TIME FATIGUE DAMAGE should be used instead.

RIFLEX 4.14.0 (2018-11-01)

RIFLEX 4.14.0 New / improved functionality

Scaling of tangential Froude-Krylov loads

A scaling factor for the tangential Froude-Krylov load term has been added for Morison or potential flow loads. The scaling factor is available for the CRS0, CRS1, CRS2 and CRS7 cross-sections.

The option to specify no hydrodynamic loads has also been added for these cross-sections.

MacCamy & Fuchs - Simplified radiation force

The input is extended in order to make the loads on bottom-fixed cylindrical monopiles to be applicable for floating single column systems by including a simple load model representing the radiation forces.

The (horizontal) radiation loads will be based on an added mass coefficient and a damping coefficient, AMY and DAMP.

In the vertical direction the wave excitation forces is calculated in a similar manner as the corresponding load term in the Morison equation. A user specified added mass coefficient will be input and used for the vertical direction, CAX.

Example of input (with default values):

```
'-------------------------------------------------------
'chtype
MACF
'-------------------------------------------------------
'cqx              cqy              cax  icode hydrod
 0.550000000e+00 9.000000000e-01  0.    2     /
'
'Simplified radiation force:
'amy damp
 0.0 0.0
'-------------------------------------------------------
```

Note that the extension of the MacCacmy & Fuchs input in Riflex 4.14.0 is not compatible with earlier versions.

External wind turbine controller

Additional nodal or element responses may now be made available as additional measurements to an external wind turbine controller. This functionality is currently under implementation.

Irregular simulation with vessel transfer function NONE

Irregular simulations with vessel transfer function NONE are now permitted. The vessel motions must be read from file or not be applied, i.e. the irregular motion indicator must be FILE or NONE.

The Wind turbine BEM code

The BEM code has been made less time-consuming. Automatic reshaping of matrices when calling subroutines have been avoided.

Hydrodynamic loads based on potential theory

Improved memory handling to avoid program termination for analyses with large amount of input data.

3D seafloor geometry file

Improved accuracy for 3D grids with very short distances between grid points. Improved data checking and warnings when reading the seafloor geometry file.

RIFLEX 4.14.0 Corrected errors

Error in water plane stiffness for lumped loads

The water plane stiffness was zero if current was present and the element crossing the water line did not have attached nodal components.

Long-standing error.

Minor changes in convergence for cases with lumped formulation and current are expected.

Low frequency motions from file

Corrected a long-standing error for low frequency vessel motions from file. The specified low frequency motions were only applied if wave frequency motions were also specified.

External wind turbine controller

Corrected the calculation of the ``measurement'' accumulated aerodynamic torsional load which is made available to external wind turbine controllers. Previously, the accumulated global x-moment was exported to the control system.

No error in applied aerodynamic loads.

Error since 4.12.0.

RIFLEX 4.14.0 Known issues

Wave kinematics at updated position in coupled analysis

Currently, the FFT wave method must be selected in `SIMO` to allow wave kinematics at updated position to be used in `RIFLEX` for a coupled analysis.

Main Riser Line inner pressure

The Main Riser Line (MRL) input allows the pressure to be specified at either end of the MRL. The specified inner pressure is, however, always applied at end 1 of the MRL. This will give errors in true wall tension and some stress components calculated in OUTMOD.

Long-standing error.

RIFLEX 4.14.0 Input changes

MacCamy Fuchs loading

The MacCacmy & Fuchs input has been changed in RIFLEX 4.14.0. Old input must be updated.

Example of input (with default values):

```
'-------------------------------------------------------
'chtype
MACF
'-------------------------------------------------------
'cqx             cqy             cax  icode hydrod
 0.550000000e+00 9.000000000e-01  0.    2     /
'
'Simplified radiation force:
'amy damp
 0.0 0.0
'-------------------------------------------------------
```

RIFLEX 4.14.0 Removed functionality

No previously available functionality have been removed in this version.

RIFLEX 4.14.0 Deprecated functionality

No functionality is deprecated in this version.

RIFLEX 4.12.5

RIFLEX 4.12.5 Corrected errors

Avoid array overwriting in dynmod

Avoid overwriting the beginning of the array containing element forces and moments in nonlinear time domain simulation. The values for the first seven elements were temporarily zero during the time step. No consequence for normal analyses as the correct values were used in the calculation of geometric stiffness and in reporting.

RIFLEX 4.12.3

RIFLEX 4.12.3 Improved functionality

Increase the model size that can be read by INPMOD

The maximum number of DMS arrays is increased from 10503 to 20003 to allow larger models to be read by INPMOD. The INPMOD work array size is increased to 2 million integer words.

RIFLEX 4.12.3 Corrected errors

Error for SIMO generic external control systems in Coupled models

An error introduced in version 4.12.0 made it impossible to run generic external control systems in a `RIFLEX` Coupled model. This has now been resolved.

Application of marine growth

Restrict application of marine growth to CRS0, CRS1, CRS2. CRS3, CRS4 and CRS7 cross-sections. Volume loads and Morison hydrodynamic loads will be modified. The modifications are calculated with the assumption that the cross-section is circular.

The modification of Morison coefficients is now based on the increase in hydrodynamic diameter so that it is consistent with the initial non-dimensional coefficients.

Numerical errors and not-a-number (``NaN") results are avoided by skipping modification of the volume loads if the external area is zero.

Modification of the Morison coefficients is skipped if the initial hydrodynamic diameter is zero.

RIFLEX 4.12.2

RIFLEX 4.12.2 Corrected errors

Error when calculating eigenvalues for model with SIMO bodies

An error has been fixed where `RIFLEX` would crash when eigenvalues was calculated for a model with `SIMO` bodies (coupled model).

Error since 4.12.0.

Application of marine growth

The drag coefficients are corrected for marine growth.

Error exit for MacCamy-Fuchs loads and a wave condition containing swell

The MacCamy-Fuchs loads are only calculated for the wind-sea part of a wave condition. An error exit is therefore added if a wave condition containing swell is specified for a case with MacCamy-Fuchs loads.

RIFLEX 4.12.1

RIFLEX 4.12.1 Corrected errors

Application of marine growth

The application of marine growth in static analysis for systems with non-consistent units has been corrected. No error for analyses with consistent units; i.e. gcons = 1.0.

The error caused the volume loads of the whole system to be scaled by a factor of 1/gcons when growth was applied in static analysis. As the typical value of gcons for non-consistent units is 0.001, the error gave clearly non-physical results.

The error occurred even if a growth scaling factor of zero was specified.

Wave kinematics at updated position in coupled analysis

Wave kinematics may now be calculated at updated dynamic positions in a coupled simulation. This resulted in an error exit from `RIFLEX DYNMOD` in 4.12.0.

Hydrodynamic loads based on potential theory

Error corrections and enhanced funtionality to allow for swell and short-crested waves. The functionality is currently under testing.

RIFLEX 4.12.0

RIFLEX 4.12.0 New / improved functionality

Wind turbine response storage

Wind turbine response storage is available based on user input specification.

MacCamy-Fuchs wave loads

The drag force input specification has been expanded to allow for dimensional drag force coefficients.

Energy convergence criterion

A convergence criterion based on energy has been implemented in Stamod and Dynmod. The rotational degrees of freedom are included, as opposed to the existing displacement convergence criterion. The new convergence criterion is approximate because the unbalanced loads refer to the configuration at the previous equilibrium iteration. Hence, false convergence may occur for problems with oscillating convergence behavior. The energy convergence criterion can therefore not be applied without also using the displacement convergence criterion.

Wind turbine blade bending moment measurements passed to external controller

The blade root (at end 1 of the first element of the foil line) bending moments about the local x, y, z axes are now available to the external wind turbine controller.

Hydrodynamic drag on cross section with 2 symmetry planes

The Morison drag term has been modified to be in accordance with the drag term used for axis symmetric cross sections. Please confer the user documentation.

Implementation of stochastic wave amplitudes

The wave components of a realization of an irregular seastate may now have stochastic amplitudes. The new option CHAMP is used to specify deterministic (`DET') or stochastic (`STOCH') amplitudes.

Note that using stochastic wave amplitudes will cause the significant wave height to vary between realizations.

Wave kinematics at updated position

Wave kinematics may now be calculated at updated dynamic positions during a time domain simulation by specifying IUPPOS = 2.

Wave kinematics can be calculated during the simulation for ISURF ⇐4; i.e. potential fixed at mean water level (ISURF ⇐ 1 and ISURF = 4), potential stretched / compressed to instantaneous wave surface (ISURF = 2) and potential moved to instantaneous wave surface (ISURF = 3). Wave kinematics may not be calculated during the simulation for second order waves (ISURF = 5).

Except for very short simulations, this will be significantly slower than pre-generating the wave kinematics at the static position. This is because complete wave kinematics time series may be generated using FFT, while wave kinematics at individual time steps are calculated as a sum of all the non-zero wave components.

The option IUPPOS = -2; calculate wave kinematics at the static positions during the simulation; is added for verification and comparison.

Storage of wave kinematics on additional file

Wave kinematics may now alternatively be stored on an additional file in ASCII format. Previously, only the binary format was available.

Wave kinematics calculated during a simulation; IUPPOS = 2 or IUPPOS = -2; are stored on files _updkin.asc / _updkin.bin.

Undo changes to PATH at the end of riflex.bat

When riflex.bat is called with CALL, variables such as PATH will also be changed in the calling environment. Multiple calls to riflex.bat will thus increase the length of the PATH variable, which may lead to the execution stopping if it becomes too long.

Calling riflex.bat with `CMD /C'' will ensure that variables in the calling environment will not be changed. This is therefore recommended, e.g.: ~ CMD /C C:/SINTEF/riflex_simo-4.10.4.BIWR-46-win64/Riflex/bin/riflex.bat i stdin CMD /C C:/SINTEF/riflex_simo-4.10.4.BIWR-46-win64/Riflex/bin/riflex.bat s stdin CMD /C C:/SINTEF/riflex_simo-4.10.4.BIWR-46-win64/Riflex/bin/riflex.bat d stdin ~ Either backward slashes ("`) or forward slashes ("/”) may be used in the path to the riflex.bat file.

riflex.bat has been modified so that the PATH variable is now set back to its initial value after running RIFLEX.

Also included in `RIFLEX` 4.10.5.

Wind force as separate load group in Stamod

The wind load was earlier activated together with specified forces. Now the wind load on turbine blades and and the rest of the structure may be activated as a separate load group in Stamod.

Example of input:

```
LOAD GROUP DATA
WIND   1
'WindOnTurbineBlades
ON
```

Improved bending hysteresis model

The numerical performance of the hysteresis bending model for CRS1 with IEJ=1 and IMF=1 has been improved. A new procedure for updating the hysteresis bending moments based on Backward-Euler integration was implemented, which allows for a fully consistent tangent stiffness matrix that is expected to improve the convergence properties of the solution procedure.

Displacement storage for duplicated nodes also

A duplicated node occurs when the last node of a line segment is the first node of the next segment. This shared node has one unique global node id but has two local node ids known to the user: `lineid-segm1-lastnode` and `lineid-segm2-1`.

Previously, only the data for the first of these were written to the ascii or binary storage file. The other was skipped. This occured even if both were requested part of the `DISPLACEMENT RESPONSE STORAGE` card. Now, both will be written - if requested.

New stationary uniform wind with shear

Wind type 14, stationary uniform wind with shear, has been added. A power or logarithmic shear profile may be specified.

This wind type differs from wind type 10, stationary uniform wind with shear, values interpolated at grid points, in that the shear profile is used directly.

Extreme wind events in dynamic analysis

An extreme wind event from `` `IEC 61400-1 Wind turbines – Part 1: Design requirements – 2005'' may be applied in a time domain simualtion. The wind event can be applied to a stationary uniform wind with shear, IWITYP = 14, in both `RIFLEX `` analyses and in coupled `RIFLEX` - `SIMO` analyses. The following extreme wind events are available: - IEC 2005 extreme coherent gust with direction change, ECD - IEC 2005 extreme vertical wind shear, EWSV - IEC 2005 extreme horizontal wind shear, EWSH - IEC 2005 extreme operating gust, EOG - IEC 2005 extreme direction change, EDC

The EWSV and EWSH events can only be used for systems which include a `RIFLEX` wind turbine. The ECD, EOG and EDC events may be used for systems without a wind turbine, but some event parameters must then be specified; e.g. velocity change, direction change and duration of event for an ECD event.

Wind turbine shutdown with generator fault conditions

Wind turbine shutdown with generator fault conditions may be applied in a time domain analysis. The shutdown event can be applied to a RIFLEX modelled wind turbine in both `RIFLEX` analyses and in coupled `RIFLEX` - `SIMO` analyses. The generator fault options include total loss of generator torque and bakup power formulated as following scaled torque control. In addition a mechanical brake in form of linear torque damping may be modelled.

Wind turbine blade pitch fault conditions

Wind turbine blade pitch fault conditions may be applied in a time domain analysis. The fault conditions can be applied to a RIFLEX modelled wind turbine in both `RIFLEX` analyses and in coupled `RIFLEX` - `SIMO` analyses. The following blade pitch fault types may modelled: - Seized, i.e. fixed blade pitch from a specific time - Runaway, i.e. blade pitch change at a specific rate until reach of final pitch angle - Actuator bias, i.e. constant pitch deviation from required pitch

The blade pitch fault conditions are modelled individually for blades with pitch fault.

Pipe-In-pipe contact forces in global system

Pipe-in-pipe contact forces and moments in the global system are now stored on the file _cntfor.asc / _cntfor.bin along with the forces and moment in the local system. The contact forces are stored if element forces are stored on an additional ascii or binary file; i.e. IFORFM /= 0; and pipe-in-pipe elements are present in the system.

Visualization of wind speeds and forces for wind turbine blades

Wind speeds and aerodynamic forces acting on wind turbine blades are made available for visualization in SIMA and SimVis.

Hydrodynamic loads based on potential theory

A test version of hydrodynamic loads based on potential theory (WAMIT) has been implemented. The functionality is currently under testing.

Marine growth

The marine growth is defined by specifying the thickness and density for a range of depths.

Example of input:

```
'----------------------------
NEW COMPonent GROWth
'----------------------------
'ID NumbOfLevels
NOR300    5
```

```
'---------------------------
' MGLEV  MG   MDENS
'---------------------------
  2.0  0.06  1.325
 40.0  0.06  1.325
-40.1  0.03  1.325
-300.0  0.03  1.325
-700.0  0.03  1.325
'---------------------------
```

Marine growth is defined as a new static load group. During the load steps in this load group, the mass and diameter properties of the elements in this zone will be modified based on their current location. These properties will be fixed in the rest of the static analysis and in the dynamic analysis. The user may thus select the static configuration that is most representative for the conditions for which the marine growth is accumulated.

An overall scaling factor for the accumulated thickness will allow the user more easily to switch between no marine growth, partly accumulated marine growth and fully developed marine growth.

Example of input:

```
'---------------------------
LOAD GROUP DATA
'---------------------------
'nstep maxit raco
4 50 1.1e-5
'lotype NLSPEC
GROWTH    1
'MGFAC
1.0
'--------------
```

RIFLEX 4.12.0 Corrected errors

Wind turbine with number of blades different from 3

Riflex failed or gave incorrect results if the number of foil blades was different from 3.

Corrected in `RIFLEX` 4.10.4.

Hydrodynamic drag on CRS2 and CRS7 cross-sections, lumped formulation

Corrected a long-standing error in hydrodynamic drag loads on CRS2 and CRS7 cross-sections when the lumped load formulation is chosen; i.e. LCONS = 0 in STAMOD.

Corrected in RIFLEX 4.10.5.

Scaling of the Froude-Krilov term for CRS2 and CRS7 cross-sections, consistent formulation

The input scaling factor for the Froude-Krilov term was not used for for CRS2 and CRS7 cross-sections if the consistent load formulation was chosen. Error in applied loads if - consistent load formulation was chosen; LCONS = 1 in STAMOD; and - CRS2 or CRS7 cross-sections with SCFK /= 1.0 were used.

Corrected the line length used in result presentation

The accumulated line length used in result presentation is now calculated from the actual stress-free segment lengths given in the line type definition in INPMOD; i.e. parameter SLGTH0 if it is given, otherwise SLGTH.

The line length is mainly used for results on the MatrixPlot files.

Previously, the line length was calculated from the updated stress-free element lengths in STAMOD and DYNMOD and from the final static nodal coordinates in VIVANA and OUTMOD.

The change in line length will be small for cases without - length changes due to temperature or pressure changes - element length modification for elements on / leaving a winch - large static elongation / compression

Elctrical torque in wind turbine

A minor error in the application of the torque from the wind turbine controller has been corrected. The torque was applied in the flex-joint local element system and not in the skew system at the flex-joint nodes. The difference between the systems is small and the correction gives insignificant changes to the results.

Corrected in `RIFLEX` 4.10.2.

Airfoil forces

A long-standing error (since version RIFLEX 4.4) has caused incorrect results or uncontrolled error termination when computing airfoil forces on cross-sections that were not part of a wind turbine. The error has been corrected.

Corrected in `RIFLEX` 4.10.2.

Reporting of wind at wind turbine hub

For coupled analyses with `SIMO` wind type `IWITYP >= 10`: - `=10`: Stationary uniform wind with shear - `=11`: Fluctuating uniform 2-component wind - `=12`: Fluctuating 3-component wind read from files (IECWind format) - `=13`: Fluctuating 3-component wind read from files (TurbSim format)

The wind at the updated coordinates of the `RIFLEX` hub supernode is now reported on the wind turbine result file. Previously, the wind at the `SIMO` body `WIND_REF` was reported. The reported wind is also sent into the external control system, but is otherwise not used in the analyses. The incoming wind on the blades is found using the updated nodal coordinates along the blades and has not been changed.

No changes for coupled analyses with `SIMO` wind type `IWITYP < 10`; the wind at the `SIMO` body `WIND_REF` is reported. Note that this wind is calculated at the wind force coefficient height `ZCOEFF`. This wind is also used as the incoming wind along the wind turbine blades .

Possible change in response for wind turbines with external control system and `SIMO` wind type `IWITYP >= 10`.

Corrected in `RIFLEX` 4.10.2.

Ball-joint release for systems with flex-joints

Corrected an error that could occur when releasing a ball-joint in a nonlinear time domain simulation for a system that also contained one or more flex-joints. The error occurred if - a single, specified ball-joint was released and the system contained a flex-joint with the same reference number as the ball-joint - all ball-joints were released and the system contained any flex-joints

The error caused `DYNMOD` to terminate with an error message when the ball-joint connector was released. Error since flex-joints were introduced in `RIFLEX 3.6.0`.

Corrected in `RIFLEX` 4.10.2.

Export of water depth for DeepC animation

The export of hydrodynamic water depth to the .vtf file for animation in Xtract has been corrected.

Error in RIFLEX 4.10.0 and 4.10.1, corrected in RIFLEX 4.10.2.

Strain-dependent cross-sectional axial damping

The input of strain-dependent cross-sectional axial damping has been corrected. The table is given as `IDMPAXI` pairs of strain and damping coefficient values. All values are given on a single input line, as described in the User Manual. The table has previously been incorrectly read as two values on each input line.

This input may still be given as two values on each input line in `RIFLEX 4.10.x`. This may be chenged in later versions.

Corrected in `RIFLEX` 4.10.2.

Empty pipe-in-pipe contact force file

The contact force file _cntfor.asc / _cntfor.bin was empty if seafloor contact elements were present. The contents were instead written to the Fortran file 80.

Error since `RIFLEX` 4.10.0. Also corrected in `RIFLEX` 4.10.4.

Allow several vessels without motion transfer functions

Several support vessels may now be specified without motion transfer function, i..e with `IDWFTR = NONE`. Previously, only one vessel in a system could be specified without motion transfer functions.

Avoid error for some cases of visualization with several vessels

Corrected errors that could cause a dynamic simulation to fail during initialization of visualization. The error occurred if the simulation included several vessels and either - no vessel motions were included; CHMOT = NONE; or - vessel motions read from file; CHMOT = FILE; and several vessels did not have transfer functions.

RIFLEX 4.12.0 Input changes

Wind turbine response storage

Storage of wind turbine responses requires user input specification.

MacCamy-Fuchs wave loads

Since the input specification has been expanded to allow for dimensional drag force coefficients, an input code for hydrodynamic drag coefficients has to be specified. As a consequence the input is not back-compatible for MacCamy-Fuchs wave loads.

Static wind loads

Static wind loads on RIFLEX elements will now only be applied in static analysis if WIND is specified as a static load. Previously, the static wind loads were activated together with specified forces. Add the load type WIND to the static load group with SFOR, specified forces, to get the same static loading as before.

See the section Wind force as separate load group in Stamod below for more deals.

Require compatible wind type in coupled analysis with wind loads

The `SIMO` wind types that are not available in standalone `RIFLEX`; `IWITYP < 10`, are now only allowed in coupled analysis if wind loads are only applied to `SIMO` bodies and wind turbine blades. If the case includes non-blade elements with wind force coefficients `IWITYP >= 10` is required.

Previously, the wind time series at the first `SIMO` body with wind coefficients was used for all non-blade elements with wind coefficients. The analysis stopped with an error if none of the `SIMO` bodies had wind coefficients.

RIFLEX 4.12.0 Known issues

Specified loads at nodes with skew boundary conditions

Specified nodal loads at nodes with user-defined skew boundary conditions are not handled correctly. The loads in the global directions are applied in the skew system. A warning is written to the .res file.

Nodes connected to a flex-joint will have degrees of freedom in a skew system. Specified loads at these nodes are not handled correctly. An error message will be written to the .res file and the analysis stopped.

RIFLEX 4.12.0 Removed functionality

Generation of LF-motions from spectra

Generation of low frequency motions from spectra has been removed in `RIFLEX 4.11.3`. The data groups `LFMOTION SPECTRUM SURGE`, `LFMOTION SPECTRUM SWAY` and `LFMOTION SPECTRUM YAW` have been removed and the options `CHLFM = GEN` and `CHLFM = NEW` are no longer allowed in `IRREGULAR RESPONSE ANALYSIS`.

Time series of low frequency motions may be read from file.

RIFLEX 4.12.0 Deprecated functionality

No functionality is deprecated in the 4.12.0 version.

RIFLEX 4.10.0

RIFLEX 4.10.0 Input changes

Most input files from version 4.8 can be used unchanged. The exceptions are noted below. Input for new functionality is described in the User Manual.

Bottom tangent option

The bottom tangent option for SB and AR systems, `IBTANG`, has been changed. `IBTANG = 1` will now specify the 3D bottom formulation.

Note that the outer contact radius, `R_EXTCNT`, of the cross-section is used in the 3D bottom formulation while it was not used in the flat bottom formulation.

The 3D seafloor formulation gives contact on all nodes that are below `Z < ZBOT + R_EXTCNT`; i.e. contact at the outer contact radius. The original flat bottom formulation gives contact for nodes with `Z < ZBOT`; i.e. contact at the centreline. Insignificant changes are expected for cases in which `R_EXTCNT = 0` for the segments in contact with the seafloor.

Results will change for cases where segments with `R_EXTCNT > 0` had contact with the flat seafloor formulation. This may be handled in four alternative ways: - If the segments do not have other contact, set `R_EXTCNT = 0` - If `R_EXTCNT` is the same for all segments with seafloor contact, lower the seafloor by `R_EXTCNT`. - Raise the final static coordinates of the nodes with specified position on the seafloor by `R_EXTCNT`. The total line length may have to be modified to obtain the same configuration and tension. - Continue using the original flat bottom formulation (see below), but note that this will not be available permanently.

The original flat bottom formulation may be chosen by specifying `IBTANG = -9`. This option has been made available to allow investigation of differences between the two formulations and will be removed in a later version.

Seafloor friction contribution to torsional load

Previously, the seafloor friction contribution to torsional load was activated in `STAMOD` by specifying the load type `FRIT`. This is now specified though the seafloor contact parameter `ILTPR` given in `INPMOD`. Specifying `FRIT` in `STAMOD` will now result in an error message.

Modified input for Carisima seafloor contact

The Carisima `INPMOD` data group `NEW COMPonent SOIL` has been renamed `NEW COMPONENT SEAFloorcontact` and will now be used for other seafloor contact types as well. The first line of input has therefore been split into two lines and a the parameter `CHSFCT` added to specify the seafloor contact component type.

Example of old Carisima input:

```
NEW  COMP SOIL
' cmpty UFA1,UFA2,IPRSO,IPREL,IPRPO
  10     1 1
```

Example of new input:

```
NEW  COMP SEAF
' cmpty chsfct
  10     CARI
```

```
' UFA1,UFA2,IPRSO,IPREL,IPRPO
  1 1
```

RIFLEX 4.10.0 Corrected errors

File storage of element forces for cases with pipe-in-pipe contact

An error has been corrected in the storage of element forces for cases with pipe-in-pipe contact. The error could lead to specified element curvatures not being stored as expected.

Storage of element forces is specified in the `DYNMOD` data group `FORCE RESPONSE STORAGE`. If `IFORFM`, Format code for storage and / or output of element forces, is not 0, the element forces for the specified elements are output to the additional file; _elmfor.asc or _elmfor.bin. If the model contains pipe-in-pipe contact elements, the contact forces for all pipe-in-pipe elements are written to the file _cntfor.asc or _cntfor.bin.

No error if forces were not stored on additional Ascii or binary files; i.e. if the data group `FORCE RESPONSE STORAGE` was not given or if `IFILFM` = 0. No error if the number of elements for which force storage was specified was at least 40% of the number of pipe-in-pipe elements.

Error since `RIFLEX` 3.6.

Element-airfoil correspondence outside of wind turbines

An error in the selection of airfoil characteristics for elements which are not part of a wind turbine blade has been corrected. In previous versions, the airfoil characteristics for complex lines could be assigned out of order in some cases.

Calculation of zero wind

The calculation of wind velocity could fail if all resulting wind components at a point and time were exactly zero. Scaling of the components resulted in a division by zero and an illegal number (NaN) was returned. This has been corrected.

RIFLEX 4.10.0 New / improved functionality

Visualization of seafloor contact

Contact forces from all non-Carisima seafloor contact may be exported from `STAMOD` and `DYNMOD` for visualization in `SimVis`.

General cross-section

A new cross-section type CRS7 has been implemented which accounts for eccentric shear center, mass center and area center.

The CRS7 cross-section also employs a new element geometric stiffness matrix that accounts for the change of internal loads due to element rigid body rotation. This is expected to improve the Newton convergence properties and increase the maximum step size for simulations with large rigid body rotations and low tension.

Coupled bending and torsion model

A coupled bending and torsion model has been implemented for CRS0, CRS1, CRS2 and CRS7. The model employs a second order approximation of both the cross-section rotation and the longitudinal Green strain, and may therefore allow for increased element lengths in certain problems.

MacCamy-Fuchs loads on `RIFLEX` elements

MacCamy-Fuchs loads (with additional quadratic drag) may be applied to vertical cross sections, which are assumed to be circular. The input format for CRS0, CRS1, and CRS2 cross sections is modified, existing files may be used without modification.

3D Seafloor contact

The 3D seafloor contact formulation has been improved in order to increase the numerical robustness. The modification will give minor changes to the results for some cases.

Line tension measurements in DP system

A new option has been added that allows the `SIMO` DP system to receive line tension measurements from `RIFLEX` lines in a coupled simulation. Previously this has only been possible for `SIMO` catenary lines. See the the `SIMO` userguide for more details.

Option to remove induction calculation for a wind turbine

An advanced aerodynamic option is included in order to remove the induction calculation on a wind turbine. This option is useful for a parked or idling wind turbine, where the aerodynamic loading is better described by quasi-static airfoil loads.

Control of Prandtl factor calculation options

An aerodynamic option is included in order to user control of the Prandtl correction options. This is done by introducing on/off switches for correction at the blade tip, at the blade root, and a switch for how these correction factors are modified in yawed inflow.

Seafloor contact specification

The data group Seafloor contact specification may be given to specify which segments in an AR-system have contact with the seafloor. Different segments can have different spring-friction contact or have contact modelled using the new riser-soil contact formulation.

A single-node contact element is generated at each end of each beam or bar element in the specified segments.

2nd order waves

The 2nd order wave calculations have been made more efficient. Slight, but not significant, changes in the results are expected.

Short-crested sea

The restriction limiting the number of short-crested directions to a maximum of 17 has been removed. This applies to both standalone `RIFLEX` analyses and to coupled `RIFLEX - SIMO` analyses.

Maximum number of line types and components

The maximum number of line types is increased from 200 to 500 and the maximum number of components from 200 to 500.

RIFLEX 4.10.0 Removed functionality

No previously available functionality have been removed in version 4.10.0.

RIFLEX 4.10.0 Deprecated functionality

Original flat bottom formulation

The original flat bottom formulation will be removed in a later version. It may be specified with `IBTANG = -9` in `RIFLEX 4.10.x`.

LF-motion response spectrum

Generation of low frequency motions from spectra will be removed in a later version of `RIFLEX`. This functionality is currently available by setting `CHLFM = GEN` in `IRREGULAR RESPONSE ANALYSIS` and then giving the date groups `LFMOTION SPECTRUM`.

Kill and choke lines

The simplified modelling of kill and choke lines attached to a tensioned riser by including them in the riser elements will be removed in a later version of `RIFLEX`. This functionality is currently available through the variable `NAKC` in `ARBITRARY SYSTEM AR` and then giving the input described in Description of kill and choke lines.

RIFLEX 4.8

RIFLEX 4.8 Input changes

Input files used in version 4.6 can be used unchanged in version 4.8 with the following exceptions: - Local element axes must now be defined for all fish net elements (CRS6)

Input for new functionality is described in the User Manual.

RIFLEX 4.8 Corrected errors

Aerodynamic pitching moment

The sign of the applied pitching moment on airfoils which are not a part of a wind turbine was incorrect. This has been corrected.

Corrected in `RIFLEX` 4.8.4.

Linux binary files

The record length of the binary files was set to four times the correct value. The .ffi, .sam, .raf, .bin and .ts files were therefore four time their necessary size. The .bin and .ts files were not comparable with their documentation and pre-existing tools for reading them.

Error since 4.8.0.

Corrected in `RIFLEX` 4.8.3.

Flex-joint with free torsional rotation

An error in the dynamic implementation of flex-joint has been corrected. The error concerned flex-joints where the torsional rotation was free while the bending degrees of freedom were locked. This type of flex-joint is typically used in wind turbine modelling.

The error led to incorrect results if the local x-axis of the flex-joint did not coincide with the global x-axis. The error was negligible if the deviation from global x-axis was small. For larger deviations, the solution tended to diverge and the simulation stopped.

Consequences for wind turbine simulations where the flex-joint x-axis was close to aligned with the global x-axis: Except torsional moment, this error affects all reaction forces/moments in the first element in the shaft (between the hub and the flex-joint). Responses in the rest of the system are only slightly affected.

Corrected in RIFLEX 4.8.2.

Drag amplification input to STAMOD

The reading of the drag amplification factors from the specified MatrixPlot file has assumed that there were five lines between the line ~ MATRIX Drag amplification factor ~ and the first line with drag amplification values. This will not always be the case; the current version of VIVANA, for example, has four lines here.

The decoding now follows the description of the MatrixPlot file format in the MatrixPlot User Guide.

Corrected in RIFLEX 4.8.2.

3D seafloor friction

An error in the 3D seafloor friction forces has been corrected. In the transformation of relative displacements and velocities from the global to local seafloor system the transformation matrix for the first node of an element was used for both nodes of the element. No error if the seafloor slope was the same at both ends of the element. The consequences of the error are expected to be insignificant for most cases.

Corrected in RIFLEX 4.8.1.

Correct wave kinematics node selection

A possible error in selection of kinematics nodes for lines that cross the upper or lower limit for wave kinematics has been corrected. The error could cause incorrect kinematics at nodes with interpolated kinematics near the limit. Previously, the node with interpolated kinematics nearest a supernode without kinematics was selected as a kinematics node. This is now only done if there are no kinematics nodes between this node and the supernode. This correction may lead to less accurate kinematics. Please note that the selected kinematics nodes are printed on the `*_dynmod.res` file, so any change in behavior can be detected in this way.

Corrected in RIFLEX 4.8.0.

Error in vessel transfer functions and short-crested waves

A long-standing error could give extrapolation in direction of the vessel motion transfer functions for short-crested wave components. The error occurred if the average propagation direction, `WADR1` or `WADR2`, was less than

```
180/(IWADR+1) * (IWADR-1)/2
```

from the first or last direction that the transfer functions were given for. `IWADR` is the number of directions used in the spreading function, `IWADR1` or `IWADR2` in data group `NEW IRREgular`

SEAState.

For IWADR = 5; the transfer function will be extrapolated if the average wave direction is within 60 degrees of the first or last direction the transfer function is given for.

For IWADR = 11; the transfer function will be extrapolated if the average wave direction is within 75 degrees of the first or last direction the transfer function is given for.

Corrected in RIFLEX 4.8.0.

Avoid extrapolation of vessel transfer function direction

Previously, the vessel motion transfer functions could be extrapolated in direction if the first and last transfer function directions did not cover a full circle; i.e. HEAD(NDHFTR) - HEAD(1) < 360 for ISYMHF = 0 or HEAD(1) not 0 for for ISYMHF > 0.

To avoid extrapolation, the first direction will now be repeated for the direction `HEAD(1) + 360` if the specified directions do not cover a full circle.

Corrected in RIFLEX 4.8.0.

Irregular simulation without waves

Irregular simulation without waves may now be run. This previously led to an error termination at the beginning of the analysis.

Corrected in RIFLEX 4.8.0.

Irregular simulation with only low frequency motions

Irregular simulation with low frequency motions and no wave frequency motions may now be run. This previously led to an error termination during generation of motion time series in DYNMOD.

Corrected in RIFLEX 4.8.0.

Visualization of regular waves for coupled analysis

SIMA and SimVis have always visualized regular waves with zero degrees direction, regardless of the wave direction specified. This error has been corrected. The error has no consequences for other results.

Corrected in RIFLEX 4.8.0.

Non-linear Buoyancy Correction (NBC) available for coupled analysis

The functionality Nonlinear Buoyancy Correction (NBC) has been made available for coupled analysis. This was not the case in the RIFLEX version 4.6.

Corrected in RIFLEX 4.8.0.

Fish net load model, CRS6

The fish net load model requires that the net plane is defined. The plane is defined by the local element X-axis and a reference vector specified in the input group: LOCAL ELEMENT AXIS DEFINITION.

If the local reference vector was not given, an uncontrolled program termination occurred in STAMOD. The program will now check whether a reference vector is given for all fish net elements. In this case, the program terminates in a controlled way giving an appropriate error message.

Corrected in RIFLEX 4.8.0.

Pipe-in-pipe sheltered closed option

Several errors regarding the sheltered closed option for pipe-in-pipe contact have been corrected. This could cause errors in the calculation of the inner fluid load effects. Error in version 4.6 only.

Corrected in RIFLEX 4.8.0.

Time domain VIV

The input value of the seed for the phase angles of the different frequency components from VIVANA has previously not been used. This has been corrected in DYNMOD.

Corrected in RIFLEX 4.8.0.

RIFLEX 4.8 New / improved functionality

Second order wave kinematics for short-crested seas

Second order wave kinematics may now be used together with short-crested seas (irregular waves with directional spreading).

Upwind tower shadow modifications

The tower shadow influence for upwind wind turbines is modified such that the drag coefficient and Bak correction factor may be taken into account. If these inputs are zero (default), the same tower shadow as before is used.

Downwind wind turbine modelling

Downwind wind turbine modelling is enabled, including correct default blade orientation and control system actions. A cosine-squared type tower shadow for downwind wind turbines is implemented.

Morison-type aerodynamic drag forces

Morison-type aerodynamic drag forces may be applied to the dry part of CRS0, CRS1, CRS2, CRS3, and CRS4 elements.

IEC turbulent wind for Linux

Wind files generated by WASP Engineering's IEC Turbulence Simulator can now also be read by RIFLEX built for Linux.

New Linux release

The new Linux release of `SIMO`, `RIFLEX` and `VIVANA` is 64-bit and solves several issues. Unfortunately, this means that 32-bit Linux operating systems are no longer supported.

- Supports more than the 32-bit imposed limit of 2 GB of RAM
- No need to install 32-bit support libraries separately
- No special considerations are needed for writing output files larger than 2 GB
- The necessary runtime libraries are included in the installation package and no special consideration is needed for installation; the package is now fully relocatable
- Added man page for the `vrr` runner script

The package has been tested on the following Linux distributions:

- CentOS 7
- Ubuntu 14.04 LTS
- Linux Mint 17

RIFLEX 4.8 Removed functionality

No previously available functionality have been removed in version 4.8.

RIFLEX 4.8 Deprecated functionality

No functionality is deprecated in the 4.8 version.

RIFLEX 4.6

RIFLEX 4.6 Input changes

Most 4.4 input files may be used unchanged.

`INPMOD` now checks that the specified pipe-in-pipe identifiers are unique. The maximum length of a pipe-in-pipe ID has been increased from 8 to 16.

Input for new functionality is described in the User Manual.

Alphanumeric identifiers may now be used for main riser lines (MRL). Old inputs with numbered main riser lines may still be used.

The alphanumeric identifiers for pipe-in-pipe specifications must mow be unique. Note that they have a maximum length of eight characters.

Beam cross-sections with linear stiffness must now have positive, nonzero axial, bending and torsional stiffness. Specifying zero stiffness will result in an error in `INPMOD`.

RIFLEX 4.6 Corrected errors

Pipe-in-pipe corrections

The friction stiffness along the pipe surface has been corrected. This is expected to improve convergence in dynamic analysis.

The unit of STIFFR, spring stiffness associated with the static friction coefficient, was incorrect in both the User Manual and the echo on the `_inpmod.res` file. The correct unit is `F / L^2`.

Corrected in `RIFLEX` 4.6.2.

Main riser line as master in pipe-in-pipe pairs

An error that prevented the master pipe of a pipe-in-pipe pair to be a main riser line has been corrected. The error resulted in an controlled error termination before start of time integration when running dynamic analysis.

Corrected in `RIFLEX` 4.6.2.

Multiple main riser lines (MRL)

Corrected an error that prevented more than one main riser line being defined. Error since 4.0.

Waves from multiple directions in coupled simulation

Corrected an error in wave loading on RIFLEX elements from numerically defined (tabulated) spectra with multiple directions in coupled `SIMO-RIFLEX` simulations.

Tubular contact

Corrected errors in calculation of tubular contact modelled using `ELASTIC CONTACT SURFACE` and `NEW COMPONENT TUBULAR CONTACT`. Changes in results are expected.

No chnage for pipe-in-pipe contact.

RIFLEX 4.6 New / improved functionality

Pipe-in-pipe improvements

In both static and dynamic analysis, the stiffness terms along the pipe surface have been modified to improve convergence. The applied forces were not changed, so no significant change is expected in results with good convergence. Included in `RIFLEX` 4.6.2.

Fluid loading on the inner pipe in a pipe in pipe pair

An option has been added to the pipe in pipe specification to allow the user to specify whether the inner pipe is exposed to the external environmental loads or is shielded by the outer pipe.

The default is that the inner pipe is exposed to the external environmental loads; i.e. buoyancy based on the water density given for the selected environment and wave loading based on the waves and current. Old input files will therefore give unchanged behaviour.

Pipe in pipe contact with reference to main riser line

The pipes in a pipe-in-pipe pair may now be defined by main riser lines (MRL), lines or a combination of MRL and line.

Improved interface for external wind turbine controller

Wind velocity at hub height in global coordinates is added to available measurements for the external wind turbine controller. Existing external wind turbine controllers can be used unchanged.

Improved dynamic stall initialization options

The internal initialization of dynamic stall parameters may fail for non-typical airfoils. The search methods have been made more robust, and error messages have been improved. The user is also given the option to specify the dynamic stall initialization parameters.

Identifiers in modeling

Alphanumeric identifiers may now be used for main riser lines (MRL). Old inputs with numbered main riser lines may still be used.

RIFLEX 4.6 Removed functionality

No `RIFLEX` 4.4 functionality has been removed in `RIFLEX` 4.6.

RIFLEX 4.6 Deprecated functionality

The following functionality was also deprecated in `RIFLEX`4.2 and 4.4.

Kill and choke lines (INPMOD)

The simplified modelling of kill and choke lines attached to a tensioned riser by including them in a riser elements will be removed in the next version of RIFLEX. In RIFLEX 4.0 this functionality is available through the variable NAKC in ARBITRARY SYSTEM AR and then giving the data in B6.9.

LF-motion response spectrum (DYNMOD)

Generation of low frequency motions from spectra will be removed in the next version of RIFLEX. In RIFLEX 4.0 this functionality is available by setting CHLFM = GEN in IRREGULAR RESPONSE ANALYSIS and then giving the date groups LFMOTION SPECTRUM.

Import of internal flow data from file (DYNMOD)

The restricted functionality to import internal flow data from file will be removed in the next version of RIFLEX. In RIFLEX 4.4 this functionality is available for some users by setting INDINT=2 in data group E1.4 and then giving the data group IMPORT FLOW DATA.

RIFLEX 4.4

RIFLEX 4.4 Input changes

4.2 input files may be used unchanged, except for the case of regular analysis with multiple vessels and no wave selected, see Specified regular motions for multiple vessels (@ref release_4_4_0_input_regmot).

Input for new functionality is described in the User Manual.

Specified regular motions for multiple vessels

The motion period must now be the same for all vessels.

RIFLEX 4.4 Corrected errors

Specified regular motions for multiple vessels

If no regular wave is specified, regular motions must be given in a regular analysis. If the system contained multiple vessels, the motions were incorrect for vessels 2 - `NVES`. This is now corrected. Note that the motion period must be the same for all vessels.

Corrected in `RIFLEX` 4.4.0.

Fatigue damage for stresses outside the S-N curve range (OUTMOD)

The `OUTMOD` data group TIMEDOMAIN FATIGUE DAMAGE has previously reported zero fatigue damage if the stresses exceeded the stress level corresponding to failure after a single cycle. This is now handled as an error. The most probable cause of this occurring is that the units of the S-N curve and the calculated stresses are not compatible.

Also corrected in RIFLEX 4.4.2.

Corrected radius of gyration for CRS0 cross section

The expression for radius of gyration that is used for the moment of inertia around the local X-axis has been corrected. This concerns cross section type CRS0 and also cross sections created based on a stress joint specification. Further, the coating contribution to moment of inertia was not previously included. The radius of gyration has been calculated to be 71% of the correct value without coating. For cross sections with coating, the difference is larger. For normal riser analysis, the error had minor or insignificant influence on the results. However, this is system-dependent. The error has been corrected and the updated expression also accounts for the coating contribution to the moment of inertia.

Corrected in RIFLEX 4.4.1.

Corrected an error in the storage of animation for DeepC

Corrected an error in the storage of dynamic analysis animation for DeepC. Error in 4.4.0 only.

Corrected in RIFLEX 4.4.1.

Corrected error in specification of detailed wave kinematics

Corrected an error in the specification of explicitly selected nodes for undisturbed wave kinematics. This may be done by specifying `DIFF` combined with `IVES = 0` in the `DYNMOD` data group IRREgular WAVE PROCedure. Error in 4.4.0 only, where an error exit was caused.

Corrected in RIFLEX 4.4.1.

Corrected error with multiple support forces in OUTMOD

An error caused `OUTMOD` to fail if the `OUTMOD` data group SUPPf TIME SERIes was given more than once. The error has been corrected.

Corrected in RIFLEX 4.4.2.

RIFLEX 4.4 New / improved functionality

Improved wind turbine results

Blade pitch results are now presented for simulations which include an external control system.

Improved in RIFLEX 4.4.1.

Friction stiffness for internal friction moment

The factor adjusting initial stiffness for the friction moment functionality has been added as user input. See Stiffness properties classification in INPMOD.

Pre-curved line types

The line type definition may now include specification of a curved stress-free configuration. See Transverse displacement specification in INPMOD.

TurbSim 3D wind files

TurbSim 3D wind files may now be read by RIFLEX. See Fluctuating 3-component wind field read from TurbSim file.

Airfoil forces

Static airfoil forces may now be applied to CRS2 cross sections that are not blades in a wind turbine.

Read wave kinematics from file

A new option to read in and use wave kinematics from file has been added, see Additional detailed specification of wave kinematics points (optional).

Improved stability for coupled analysis with the old SIMO DP-system

The SIMO DP system is now called only at every SIMO time increment (Sampling time interval). Previously, the old SIMO DP system was called at each time step (Time integration step) while the new DP system was correctly called at the sampling time interval.

The correction will possibly increase the computational robustness when the old DP-system (marked deprecated) is applied.

Improved wind turbine results

Improved selection and presentation of results for wind turbines. The results are now presented for all blades in the non-rotating shaft system.

Minor improvements

Blank space is added around some numbers printed on the .res files. This has been done to increase readability and to facilitate comparison of numerical results.

Minor improvements in error handling and layout.

RIFLEX 4.4 Removed functionality

No RIFLEX 4.2 functionality has been removed in RIFLEX 4.4.

RIFLEX 4.4 Miscellaneous

Run time environment

Windows versions 4.4 and higher are 64 bit executables and therefore require different Fortran and Java DLLs than earlier version. The necessary DLLs are included in the download package. The preformance is improved.

RIFLEX 4.2

RIFLEX 4.2 Input changes

4.0 input files may be used unchanged.

Input for new functionality is described in the User Manual.

Changed default values for irregular analysis (DYNMOD)

The default sampling step of the pre-generated time series, `DTGEN`, is decreased to 0.5 s. The default length of the pre-generated time series, `TIMGEN`, is increased to 4.6 hours. The default simulation length for irregular analysis, `TIME`, is increased to 3.06 hours.

Changed default value for subdivision of time step (DYNMOD)

The default value of the `DYNMOD` parameter `IVARST`, Code for automatic subdivision of time step, is changed from 2 to 0. If a simulation fails to run, check if it previously used subdivision and if the default value of `IVARST` was used.

RIFLEX 4.2 Corrected errors

The following errors have been corrected in RIFLEX 4.2 versions.

Pipe-in-Pipe Contact Search Bug

Fixed a long-standing bug in pipe-in-pipe contact search that caused the search to fail in the vicinity of flex joints and ball joints. Users with pipe in pipe models that contains flex joints and/or ball joints should consider rerunning the simulations.

Corrected in `RIFLEX` 4.2.1.

Coupled analysis

Error in hydrodynamic drag forces on slender elements

If depth dependent current was applied and wind also was specified in the environmental condition, the computed drag forces on slender elements were in error. This was due to an incorrect z-position being used to interpolate the shear profile. The error has been present since version 4.2.0. The consequence is incorrect drag forces on the slender elements. **Analysis done with this combination by version 4.2.0 should be re-run!** The error has been corrected.

For other SIMO bugfixes, please refer to the SIMO Release Notes.

Corrected in `RIFLEX` 4.2.1.

Instability in prescribed displacements when vessel motions are imported from file Coupled analysis

Simulations occasionally failed when vessel motions were imported from external file. The algorithm for prescribed (vessel) rotation induced translation for nodes attached to the vessel has been modified to have improved numerical robustness. No or insignificant changes to results.

Corrected in `RIFLEX` 4.2.0.

Exported vessel velocity and accelerations in coupled analyses (DYNMOD)

An error in the exported vessel velocity and accelerations for coupled analysis with both SIMO bodies and RIFLEX support vessel was corrected. No error found in analysis results. No error for standalone RIFLEX or for coupled analyses with only SIMO bodies,

Corrected in RIFLEX 4.2.0.

Winch modelling

A long-standing error in the winch formulation has been detected and corrected. Analysis based on previous versions of RIFLEX should be re-run.

Corrected in RIFLEX 4.2.0.

Linear drag coefficients not applied in linearized time domain analysis

Linear drag coefficients have not been applied in linearized time domain analysis. Linear drag coefficients are seldom used, so the consequences are assumed to be minimal. The error has been corrected.

Corrected in RIFLEX 4.2.0.

Regular wave analysis with prescribed vessel motions

Error for combined regular wave loading with specified harmonic displacements (IMOTD = 2). The motions after the first wave period were incorrect if a different period was specified for the motions than the wave period (and the wave period was mot a multiple of the motion period). . Error since RIFLEX 3.6.7.

No error if motions are calculated from vessel transfer functions (IMOTD = 1).

Corrected in RIFLEX 4.2.0.

RIFLEX 4.2 New functionality

Geotechnical model

New modelling features have been added to model transverse contact between a vertical pipe and the soil. The new INPMOD data group GEO SPRING SPECIFICATIONS is used to give the input to the model.

Scaling of Froude-Kriloff term in Morison’s equation

Extend the generalized Morison's equation by optionally scaling the term associated with the Froude-Kriloff load. This to enhance the area of application of Morison's equation, i.e. allow for larger structural diameter versus wave length. Note that only loads normal to the principal axis of an element are scaled. Additional, optional input is added for cross section typed CRS0, CRS1, CRS2, CRS3 and CRS4.

2nd order wave kinematics

2nd order wave kinematics for long-crested waves have been added to RIFLEX. 2nd order wave kinematics are obtained by setting the DYNMOD input parameter ICOSIM to 5.

Store wave kinematics on additional file

Generated wave kinematics may now be stored on additional binary file from DYNMOD. Data group IRREGULAR KINEMATICS STORAGE to store wave kinematics on an additional file <prefix>_wavkin.bin. The contents are described in the text file key_<prefix>_wavkin.txt.

Wind turbine control system and results

The specified control interval for internal wind turbine control system has been activated. In addition the value of the integrator gain are kept within saturation limit. The reason for this implementation is to improve the behaviour of the internal control system.

Wind turbine results may now be stored on additional text file or binary file <prefix>_witurb.asc or <prefix>_witurb.bin. The contents are described in the text file key_<prefix>_witurb.txt. There is no special input for storing the wind turbine results. The results storage is based on storage specification for element forces, i.e. time interval for sampling and file format.

Detailed kinematic specification

Allow NODSTP = 0 i.e. no kinematics for this line in the detailed kinematic specification.

Available from RIFLEX 4.2.1.

RIFLEX 4.2 Removed functionality

No RIFLEX 4.0 functionality has been removed in RIFLEX 4.2.

RIFLEX 4.2 Miscellaneous

Java and HLALIB.jar

The included Java folder and the HLALIB.jar file have been updated in the Windows installation .zip file.