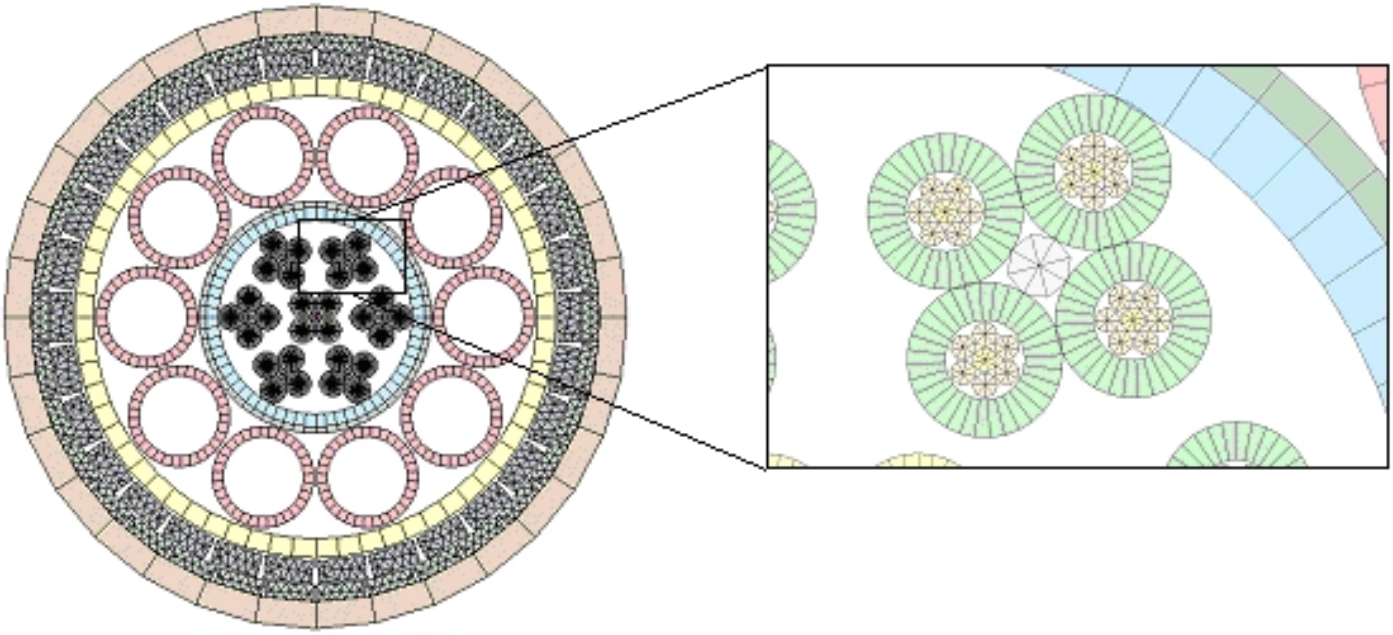
UFLEX2D Version 2.8.8
User Manual

April 11, 2018



Contents

List of Figures

Notation

CST: Constant Strain Triangular element
Q_x: Axial force
M_x: Torque
M_y: Bending moment about y-axis
M_z: Bending moment about z-axis
κ_x: Torsion
κ_y: Curvature about y-axis
κ_z: Curvature about z-axis
ε: Strain
σ: Stress

List of changes

0.1 Changes in version 2.8.8

General changes:

- Capacity data may now be obtained by running UPOST2D using the new **CAPACIT** input card.
- Improved stability of stick/slip behaviour after tuning. Prior versions were highly sensitive to changes in contacts during a curvature history.
- General bugfixes.

Cards that have been added:

CAPACIT - NEW - New output type of UPOST2D.

0.2 Changes in version 2.8.6

General changes:

- Bug fix related to the use of **EXTCON** with tuning.
- Bug fix related to isotropic hardening. The bug was introduced in 2.8.4.

0.3 Changes in version 2.8.4

General changes:

- Fatigue data may now be obtained by including **CURV-SLIP** in **VISRES**, followed by providing input to UPOST2D using the new **FATIGUE** input card.
- Improvements with regards to the calculation of slip curvature for components with non-linear material relations.
- All vonMises stress components are now allowable result types for **NOPLOT**. If they are not relevant for a given element type they evalute to zero.

Cards that have been added:

FATIGUE - NEW - New output type of UPOST2D.

0.4 Changes in version 2.8.2

General changes:

- Fixed a bug which affected the tuning of slip curvatures for bodies which use the **RESULTANT** material model.
- Optional relaxation of constraints for **LMERGE** applied to shells, to avoid excessive bending stresses.
- Removed the need for **TUGEOM** when using **FRICMODEL 1**.
- Negative lengths in straight **GEOM** segments are now disallowed.

Cards that have been changed:

LMERGE - CHANGED - Added optional relaxation option **RELOPT**.

0.5 Changes in version 2.8

General changes:

- Contact interfaces are generated automatically by including the **CNTINTFIND** card. This feature is currently under development.
- A new friction model, based on plane surfaces remain plane strains during stick. Also, bugfixes related to the calculation of slip curvatures and changes to how tuned slip curvatures are reported in the summary file. The new friction model allows improved modelling of friction stresses during multi-directional bending.
- Modifying result display such that radial stresses are taken into account for the Von Mises stress if **THICKWALL** is specified for shell geometries of linear material.
- Modifying the end cap load such that inner and outer radius is accounted for if **THICKWALL** is specified for shell geometries of linear material type.
- New command line parameter to adjust the amount of memory booked at runtime.
- Bugfix related to use of **VOID**, to the display of bitumen stresses and result types for shell elements. Also, legends in the mpf output from UPOST2D now support result types with longer names.
- Improving friction results for **RESULTANT** material by using the axial scaling factor for displayed stress results.
- Bugfix related to **BITUMEN** material, possibilities to have more than one geometry in contact with the material. Appropriate time steps are now determined automatically when using the **BITUMEN** material.

Cards that have been changed:

CONTROL - CHANGED - Optional parameters have been removed.

Cards that have been added:

CNTINTFIND - NEW - Generate contact interface definitions automatically. This feature is currently under development.

FRICMODEL - NEW - Choice of friction model.

0.6 Changes in version 2.6

Cards that have been changed:

CONTROL - CHANGED - If parameter **AVRGT** average tension is negative, alternative meaning of step number to performe tuning at.

Cards that have been added:

TUGEOM - NEW - Geometry to base tuning of friction spring at.

TIME - NEW - **TIME** card added.

BITUMEN - NEW - **BITUMEN** material added, must be used with **TIME** card.

0.7 Changes in version 2.4

General changes:

- Enable printing out information in UFLEX2D log-file, regarding critical contact interface and contact geometry. This is useful in case of divirgence, then corresponding contact settings may be modified.
- Termoelastoplastic material is implemented for beams.
- Modelling of asymmetric torsion is implemented.
- Enable giving input file as argument on command line for UFLEX2D and UPOST2D

Cards that have been changed:

MATERIAL - CHANGED - Termoelastoplastic material added.

CONTROL - CHANGED - Three new optional parameters for controlling convergence are added.

CONTINT - CHANGED - Optional buffer parameter is added.

Cards that have been added:

TLOAD - NEW - Temperature load.

TORSCENTRE - NEW - Shift torsion centre.

0.8 Changes in version 2.2

General changes:

- Anisotropic material is added.
- Two dimensional cross-section with geometry ovalization is possible to analyze, relevant only for high lay angles.
- Return of exit code. Value 0 for successful run, 1 for controlled error return and larger than 128 for crash.
- Improvement of error messages related to incorrect input.

Cards that have been changed:

MATERIAL - CHANGED - Anisotropic material added.

CONTROL - CHANGED - New option. By using negative value for **NLPRINT**, print of iteration and obtained convergence to screen is avoided.

CONTINT - CHANGED - Implemented option for automatic generation of friction material.

Cards that have been added:

CONTA2D - NEW - Two dimensional cross-section with geometry ovalization.

0.9 Changes in version 2.1

General changes:

- New result type **NOPOS-X**, **NOPOS-Y** and **NOPOS-Z** for use in UPOST2D.
- Allow for longer strings in input file. Maximum string length is now 128 characters.
- Name for geometries etc. can now be contained in each other.
- Files will be generated in standing directory, regardless of where input files is located.
- New optional digit parameter in the UPOST2D input file.
- Ability to generate 3D profile from the 2D profile.
- Some missing result types for elastoplastic ans hyperelastic beam included. Only visual effect.

Cards that have been added:

DISPLACE - NEW - Displace the complete cross section along the longitudinal axis.

ROTATE - NEW - Rotate the cross section.

CENTERFIX - NEW - Lock the position of the center of a geometry.

EXP3DCONTROL - NEW - Define parameters for 3d export.

EXP3DGEOM - NEW - Define rectangular geometry for 3d export of components.

EXP3DLAYER - NEW - Define a 3d export layer.

EXP3DCORE - NEW - Define 3d export core.

EXP3DOUTER - NEW - Define 3d export oouter sheet.

EXP3DARMOUR - NEW - Define reduced number of members in 3d export, intended for use with armour layers.

EXP3DELDEF - NEW - Define varying element lengths in the longitudinal direction.

0.10 Changes in version 2.0

- Visualizing contact stress on both master and slave geometry.
- Improving friction stress calculations.

CONTROL - CHANGED - New parameter **AVRGT** average tension. If **MATERIAL** of type **FRICSPRING** is applied, this parameter is used for automatic determination of stick/slip level. Removing optional parameters **ROTFAC**, **TRANSFAC**, **SOFTFAC** and **CONSHELLFAC**.

MATERIAL-FRICSPRING - CHANGED - By default the given material curve for **FRICSPRING** is now dummy, and proper levels for stick/slip are calculated by the program. However, the user can force the use of given curve by applying negative sign for the hardening parameter.

FRICTION - CHANGED - The parameter for material name is removed. The material name given in the **CONTINT** card will be used.

VISRES - NEW - Using this card, the user can limit the result types that are stored to .raf file.

THICKWALL - NEW - To obtain thick wall stress results for shell, this card can be applied.

1 Introduction

1.1 Purpose of document

The purpose of this document is to describe the input to version 2.8.8 of the UFLEX2D program system. For details regarding testing/verification of UFLEX2D and theoretical background, see (Sævik, 2004a) and (Sævik, 2004b).

1.2 Program basis

UFLEX2D is a FEM based computer tool for stress analysis of umbilicals exposed to internal pressure, external pressure, tension, torsion, bending and external contact loads. The present version of the program works in two dimensions utilising concepts from differential geometry with respect to the kinematics.

1.3 Program structure

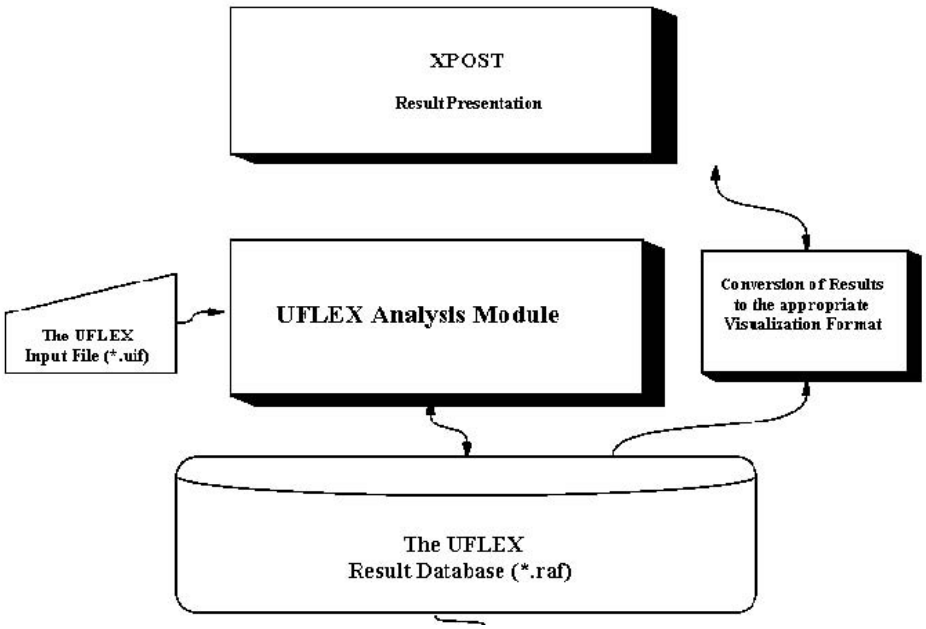


Figure 1.1: UFLEX2D system architecture.

1.4 UFLEX2D program files

Version 2.8.8 includes the following files:

prefix.2if UFLEX2D Input File.

prefix.2of UFLEX2D Output File includes all model information.

prefix.2lf UFLEX2D Log File where warnings and error messages are written.

prefix.rep UFLEX2D summary REPort file where cross-section key parameters are written.

prefix.geo The model geometry file where the visual model stored on GLVIEW format.

prefix_model.geo Mathematical model geometry file where the mathematical model is stored on GLVIEW format. In addition, model information is stored on GLVIEW format for each individual geometry. The file name is geometry name.geo.

prefix.raf The results data base where all numeric and visual results are stored.

In addition, mesh files (*.msh) and boundary files (*.bdf) are generated for the different geometries which are defined in the input file.

1.5 Executing the inputfile

UFLEX2D is executed from command line, and can be started using the following formats:

```
> uflex2d
```

will promt user for input file name. The input file name should be given without the .2if ending. To specify file name directly from command line:

```
> uflex2d <filename>
```

In addition the following options can be specified:

```
-m <fac>
```

Adjust the memory allocated by UFLEX2D with factor fac. The factor should be larger that 0.1. The upper limit will depend on availiable memory on your computer. The memory option can be given alone, before or after filename:

```
uflex2d -m <fac>
uflex2d -m <fac> <filename>
```

```
uflex2d <filename> -m <fac>
```

2 Modelling

This chapter contains a general overview of the modeling concepts of UFLEX2D. In the last section some comments are made about difficulties that have been experienced by users. This section might be useful after you have tried to run UFLEX2D.

As described in Chapter 1, UFLEX2D is file based. A model is built in a ascii file, and output is given to different files. The main output file is the binary raf-file, which is the result database. Results can be inspected directly by using the graphical interface XPOST. Alternatively, the UPOST2D post processor can be applied to subtract information form the raf-file.

The input file is split into sections by the use of CARDS. They are keywords that must be entered at the start of a line. By using the cards, the user supplies different information and parameters according to a specified format described in Chapter 3.

Some of the cards are required in all simulations, while others will be defined only when needed. In the following, the cards are grouped and the purposes of the cards in the group are described.

The described model is a cross section of the umbilical, and in the calculations a 1 mm long section of the umbilical is assumed.

2.1 Simulation parameters

The **CONTROL** card are required in every simulation. The purpose of this card is to supply general simulation settings such as convergence radius and how many times results will be stored to the raf database. Global parameters as constant of gravity and average tension for the system is also given here.

The remaining cards **HEAD**, **VISRES** and **THICKWALL** are optional. **HEAD** can be used to add a description to the simulation, of which the first line will appear in the XPOST window. By **VISRES**, the user can specify which types of results that will be stored to the result database. **THICKWALL** can be used for the objects where the user would like stresses to be calculated according to thick walled shell theory, default is thin walled shell theory.

2.2 Building the cross section

CROSS, **GEOM** and **MATERIAL** are all required in a simulation. The card **CROSS** is used to describe a members location in the cross section. It is also used to select the element type for a member and which material it consists of. The material will then be further described by a corresponding **MATERIAL** card. In the **CROSS** card is also a reference to a geometry description, which will be found in a corresponding **GEOM** card. This card describes the geometry of the member, by circular curves and straight segments. Meshing of the member is also defined in this card.

There is also a version of the **CROSS** card that only gives the location of a member, and that the member will be described in more detail later on. This is used when the member itself consist of several sub-members. Each sub-member is described by a detailed description of material, element type and geometry later on.

VOID enables the user to fill areas which can be described by the boundaries of other objects. This concept is used to enable accurate weight calculations reported to the .rep file. It can not be used to describe structural elements in the cross section.

2.3 Boundary conditions

LOBON and **GLBON** are used to supply global or local boundary conditions to the system.

2.4 Loads

GCLOAD and **LCLOAD** are used to apply global or local loads respectively. **INISTRAIN** defines initial strains, **PRLOAD** pressure on inner or outer surface of a member and **LPDISP** and **GPDISP** can be used to apply local or global prescribed displacements.

For all load specifications, reference to a history must be given. This history must be found in a corresponding **HIST** card. On this card, a load pattern is given. This pattern can vary over the steps, and the resulting load is defined by the load from the load card multiplied with its load pattern.

2.5 Contact between members

If two geometries are in contact and this effect should be modeled, a **CONTINT** card referring to the considered geometries must be given. This card also refers to a material

type, which must be defined by a corresponding **MATERIAL** card.

A database of friction coefficients, valid for contact between two materials, can be stored by use of the **FRICTION** card.

A special contact object, typically used to model external contact structures such as Caterpillar can be defined by the **EXTCON** card.

2.6 Merging

For stability reasons, and to provide a model that behaves according to the mean stress concept applied in all industry standards for such structures, it may sometimes be convenient to restrict the motion of members that belongs to the same layer of a cross section, the same shell, layers of shells etc. To achieve this, the cards **LMERGE**, **SMERGE** and **CMERGE** have been defined.

Layer merge makes sure that all the members of a layer move identically with respect to radial motion.

Shell merge causes all elements in a shell to have the same radial motion, i.e. the circular shape of the shell will be kept.

Contact merge ensures that members that are attached to each other will remain attached. This concept is made to handle coating in tubulars.

2.7 Order of a member

It is important to understand how the term order is used in UFLEX2D. A member which is straight is defined to be of first order. A second order member is a member which is wound about a first order member and has a specified lay angle (or pitch length). A third order member is a member which is wound about a second order member. Viewing the second and third order members as a composite member, this composite member will then be wound about the first order member.

2.8 Coordinate system and DOFs for different element types

When applying local boundary conditions or loads, the direction to which it applies is important. There are two orthogonal systems that are important to relate to, that is the global system of the model and the local system of the different elements.

The global system is visualized when the model is displayed in XPOST. The axis of this system is defined in such a manner that the x-axis is pointing directly out of the paper plane. The y-axis is pointing from the umbilical cross section center and along the 0° direction, or pointing horizontally and towards the right. The z-axis is directed from the umbilical cross section center and along the 90° direction, pointing vertically and upwards.

For the local element systems it is not as straight forward, as they differ depending on element type. A component that is created by the **BEAM** or **BEAMSH** element type has only one local system for each component. For first order components, this system is aligned with the global system. For the higher order components, the local system of the component is described by a z-axis going from its one order lower center object and to the center of the component itself. The x-axis is pointing along the components longitudinal axis, and depending on the direction chosen when defining the geometry the axis will point in or out of the paper plane.

A component of type **SHELL** has a separate local system for each of the elements it consists of. For each of the elements the local x-axis is directed along the line that joins the two nodes, that is the hoop-direction of the tube, along the segment/element. The y-axis is parallel with the tube longitudinal axis and directed in or out of the paper plane, depending on the direction chosen when the geometry is defined.

Components of type **CONSHELL** are similar to shell, with the local x-axis pointing from the first to the second node of the element and the y-axis perpendicular to the paper plane.

2.9 Recursive definition of cross section

The cross sections are defined recursively in UFLEX2D, and a description of the concept follows. In Fig. 2.1 is shown a layer from a cross section. This layer consists of ten second order branches as defined in the first section in the example below.

The simplest branch is the one that consists of a beam only. The name of this branch is FILLER, and element type, geometry and material name are defined directly in the **CROSS** definition. In addition there are two other types of branches that are more complex. They are named QUAD1 and FO, and the cross section definition of these are not completed in the first **CROSS** group. By entering **NEXT**, the branch is defined to have the given pitch length etc.

The detailed description of the branch, given in a local system, will be defined later on. For FO this is done in the second section of the example. It consists of four different geometries, three shells (two of them are neighbors, and can not be separated in the

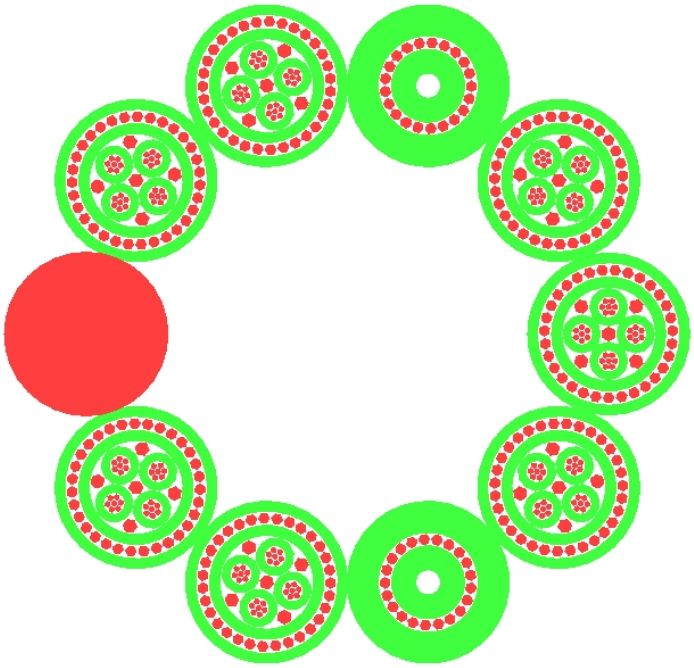


Figure 2.1: Layer of a UFLEX2D cross section.

figure) and beams that are wound about the local center of the branch. In this case, the shells in FO are of second order, and the beams of third order.

QUAD1 is described in the third section of the example, in the same manner as FO. But for this description, another sub-branch named COND1 is introduced. This branch is then described in the last section of the example. In this case, the copper conductors with geometry CU21 are of fourth order. The remaining beams (except for the center beam) and the small shells are of third order. The two large shells and the center beam are of order two.

EXAMPLE:

```
#     LINENO NAME    PROCESS  PROCNO PITCH   RAD    THETA   ELTYPE  GEONAME     MATNAME
#
# Cross section layer
CROSS 1      QUAD1   NEXT     1      -1725   34.85      0
CROSS 1      QUAD1   NEXT     1      -1725   34.85     36
CROSS 1      FO      NEXT     1      -1725   34.85     72
CROSS 1      QUAD1   NEXT     1      -1725   34.85    108
CROSS 1      QUAD1   NEXT     1      -1725   34.85    144
CROSS 1      FILLER  FILLER   1      -1725   34.85    180   BEAM    FILLER      POLYE
CROSS 1      QUAD1   NEXT     1      -1725   34.85    216
CROSS 1      QUAD1   NEXT     1      -1725   34.85    252
```

```
CROSS 1      FO      NEXT      1     -1725   34.85    288
CROSS 1      QUAD1   NEXT      1     -1725   34.85    324
#
# FO details
CROSS 1      FO      FO
CROSS 1      FOOUT   FOOUT     1       0     0        0    SHELL   FOOUT       MDPE
CROSS 1      ARM     ARM      21     260     5.75     0    BEAM    ARM4        GSW120
CROSS 1      FOINT   FOINT     1       0     0        0    SHELL   FOINT       MDPE
CROSS 1      FOROR   FOROR     1       0     0        0    SHELL   FOROR       STEEL
#
# QUAD1 details
CROSS 1      QUAD1   QUAD1
CROSS 1      Q1OUT   Q1OUT     1       0     0        0    SHELL   Q1OUT       LLDPE
CROSS 1      ARM     ARM      31     366     8.55     0    BEAM    ARM3        GSW120
CROSS 1      Q1INT   Q1INT     1       0     0        0    SHELL   Q1INT       LLDPE
CROSS 1      COND1   NEXT      1     140     3.6      0
CROSS 1      COND1   NEXT      1     140     3.6     90
CROSS 1      COND1   NEXT      1     140     3.6    180
CROSS 1      COND1   NEXT      1     140     3.6    270
CROSS 1      FILL0   FILL0     1       0     0        0    BEAM    FILLER0     MDPE
CROSS 1      FILL1   FILL1     1     140     5.20    45    BEAM    FILLER1     MDPE
CROSS 1      FILL1   FILL1     1     140     5.20   135    BEAM    FILLER1     MDPE
CROSS 1      FILL1   FILL1     1     140     5.20   225    BEAM    FILLER1     MDPE
CROSS 1      FILL1   FILL1     1     140     5.20   315    BEAM    FILLER1     MDPE
#
# Conductor
CROSS 1      COND1   COND1
CROSS 1      KAPPC   KAPPC   1       0   0        0    SHELL   CUKAPP1     LLDPE
CROSS 1      CU CU            1       0   0        0    BEAM    CU11        COPPER
CROSS 1      CU CU            1      90   0.88     0    BEAM    CU21        COPPER
CROSS 1      CU CU            1      90   0.88    60    BEAM    CU21        COPPER
CROSS 1      CU CU            1      90   0.88   120    BEAM    CU21        COPPER
CROSS 1      CU CU            1      90   0.88   180    BEAM    CU21        COPPER
CROSS 1      CU CU            1      90   0.88   240    BEAM    CU21        COPPER
CROSS 1      CU CU            1      90   0.88   300    BEAM    CU21        COPPER
```

2.10 General advice

General advices and notes:

- The input file of UFLEX2D is based on cards. When a card is read, all parameters until the next valid card will be read. An effect of this is that misspelling of a card in general will give error return from the card just above.
- Due to the internal UFLEX2D handeling of strings, some letters or combination of letters must not be applied as part of the names given in the input file. The following words are forbidden regardless of upper- or lower-case: -, shell, beam, beamsh, central, connect, conshell, suface and interface.
- Be careful when using **LOBON** and **GLBON** at the same time, in particular when both works in the same direction.

- Note that there must always be a first order member present in the cross section. If you don't have one in your real cross section, you can add a very soft core that does not influence the behavior of the model.

- You should in general use different geometries for different layers. This has several reasons: Contact is defined between geometries, friction work is averaged over a geometry and layer merge causes all the members of a geometry to behave the same way.

- There are many possible approaches to modelling friction. The choice depends on the level of control desired by the user. There is no need to specify the **FRICSPRING** unless the user wishes to manually control the exact slip level for a contact interface. Specifying **AUTO** for **FRICMATNAME** in **CONTINT**allows friction springs to be generated automatically. However, if detailed control is desired, the user should take care to define only one friction spring material for each contact interface.

- The recommended approach to modelling friction is to generate friction springs automatically, and to tune the slip level by providing a negative value for **AVRGT** in **CONTROL**, as described in Section 3.8. Also, **FRICMODEL 1** should be provided, to choose the strain based friction model. The older friction model, based on accumulated work in the friction springs, is known to be over-conservative with respect to bending stiffness.

3 UFLEX2D input guide

3.1 General

All input is described on ASCII file format. The maximum number of letters in one single text string is 128.

Force and length are given in N and mm.

Comment text strings are defined by introducing a # at the start of the line.

The following identifiers define the different data groups:

HEAD	Heading describing the cross-section, see Section 3.25
CONTROL	Control parameters, see Section 3.8
CONTA2D	Alternative contact search, see Section 3.5
CROSS	Cross section definition, see Section 3.9
EXTCON	External contact definition, see Section 3.18
GEOM	Geometry definition, see Section 3.22
CONTINT	Contact interfaces, see Section 3.6
CNTINTFIND	Card for automatic contact interface list generation, see Section 3.7
MATERIAL	Material property data, see Section 3.32
BITUMEN	Bitumen material, see Section 3.2
TIME	Time definition, see Section 3.37
FRICTION	Friction property data, see Section 3.20
FRICMODEL	Choice of friction model, see Section 3.19
TUGEOM	Choice of tuned geometry, see Section 3.40
CMERGE	Contact merged surface defintion, see Section 3.4
LMERGE	Layer merge defintion, see Section 3.29
SMERGE	First order shell merge defintion, see Section 3.35
VOID	Void definition, see Section 3.42
LOBON	Local boundary conditions, see Section 3.30
GLBON	Global boundary conditions, see Section 3.23
LCLOAD	Local concentrated loading , see Section 3.28
GCLOAD	Global concentrated loading, see Section 3.21
PRLOAD	Pressure loading, see Section 3.33
TLOAD	Temperature loading, see Section 3.38
INISTRAIN	Initial strain loading, see Section 3.27

LPDISP	Local prescribed displacement, see Section 3.31
GPDISP	Global prescribed displacement, see Section 3.24
HIST	Load history data, see Section 3.26
THICKWALL	Display thickwall stresses, see Section 3.36
VISRES	Selection of result types, see Section 3.41
CENTERFIX	Specify the center of a geometry, see Section 3.3
TORSCENTRE	Shift torsion centre, see Section 3.39
DISPLACE	Global displacement of cross section, see Section 3.10
ROTATE	Rotation of cross section, see Section 3.34
EXP3DARMOUR	Reduce memebers in armour layers when exporting to 3D file, see Section 3.11.
EXP3DGEOM	Define geometry other than cylinder when exporting to 3D file, see Section 3.15.
EXP3DCORE	Define core when exporting to 3D file, see Section 3.13.
EXP3DCONTROL	Control parmeters for exporting to 3D file, see Section 3.12.
EXP3DELDEF	Define element distribution when exporting to 3D file, see Section 3.14.
EXP3DLAYER	Define layer when exporting to 3D file, see Section 3.16.
EXP3DOUTER	Define outer sheat when exporting to 3D file, see Section 3.17.

3.2 BITUMEN

Bitumen is a type of viscoelastic coating that adds bending resistance to the genuine stiffness of the structural element. For example, bitumen can cover steel armour. Bitumen stiffness will then be added to the steel stiffness. Bitumen material properties are applicable for beam, beamshell and shell. One should avoid the case when steel shell is coated with plastic sheath and then covered by bitumen, since bitumen will be connected to the outer plastic sheath, and not a steel tube. Also, the currently implemented bitumen model assumes constant or slowly varying material stiffness of the cross-sectional member covered with bitumen. Therefore it is advised to only use bitumen cover for linear materials, or with certain approximation for hyperelastic, elastoplastic or resultant materials.

The following format applies:

BITUMEN GEONAME G' G'' Ω **CONST**

where

GEONAME: Geometry name where bitumen coating is applied.
G1: Storage modulus, unit: MPa. (G')
G2: Loss modulus, unit: MPa. (G'')
OMEGA: Frequency, unit: rad/sec. (ω)
CONST: Force constant, unit: 1/mm.

Note that bitumen is modelled using following two parameters $K = G'\cdot\text{Const}$, $C = G''\omega^{-1}\cdot\text{Const}$, see theory manual for details.

EXAMPLE:

```
#       Geoname G'  G'' Omega Const
BITUMEN  ARMO1 1E2 1E2    1     1
```

3.3 CENTERFIX

By the card **CENTERFIX**, the user can specify the center of a geometry. Otherwise, the center of the geometry will be in the center of gravity. **CENTERFIX** is meant only for first order geometries, since higher order components can be displaced by specifying proper center-to-center radius.

CENTERFIX GEONAME YCOR ZCOR

The parameters required are:

GEONAME: Geometry name.
YCOR: y-coordinate, relative to local system for the geometry.
ZCOR: z-coordinate, relative to local system for the geometry.

The card can be given several times.

In the example below, the geometry FILL is a beam of radius 80, and with a hole that is off center. If **CENTERFIX** is not issued, the center of the beam will be moved slightly towards the hole.

EXAMPLE:

```
#    Geoname   FORM   X0  Y0  Curvcode P1 P2  P3   P4  NINTER INTERFACE
GEOM FILL     CLOSED 0.0 0.0 CI       0  360 80.0 0   200    1
                     -3.5 0.0 CI       0  360 22.7 0    96    2

#         Geomane   xcor  ycor
```

```
CENTERFIX FILL     -80    0
```

3.4 CMERGE - Contact MERGEd surface definition

In general, each of the different defined geometries is assumed to interact in terms of contact elements. However, in many cases such as when applying coating to tubulars, the geometries are directly linked to each other. This means that in the structural model, no new degrees of freedom need to be defined. In order to define such behaviour, the **CMERGE** concept is introduced, i.e. for a number of elements containing geometry names of a defined type, only one set of dofs need to be introduced. The format of the **CMERGE** card is as follows:

CMERGE MGEONAME SGEONAME ...

where:

MGEONAME: Name of master geometry, normally the steel tube.
SGEONAME: Name of slave geometry, normally the coating.

Several slave geometries can be given.

With respect to the associated element types to each geometry the following combinations are allowed:

- Master geometry shell against slave geometry shell
- Master geometry beam against slave geometry beam
- Master geometry beam against slave geometry shell

The shell-shell combination is normally used for pipe coatings, where the dofs related to the coating layer are defined as a slave of the steel tube layer. This allows modelling of layer bonding and will eliminate numerical problems due to the large difference in stiffness between these layers.

The beam-shell combination can only be used if the shell and the beam is of order 1. Note that the shells of order 1 translation dofs are transformed to the shell local coordinate system and that the merging is only applied in the surface normal direction. This is to allow fixing the first order shells, which have no transverse stiffness, to bodies that have stiffness, improving the system numerical stability behaviour.

The shell-beamshell combination is used for study of deformations of plastic coating (beamshell) covering shell. This merging can be done for both first and higher order

geometries. All three translational degrees of freedom for the nodes, common for shell and beamshell, are coupled. **CMERGE** can not be applied on beamshell.

EXAMPLE:

```
#       mgeoname      sgeoname
CMERGE PIPE1         KAPP1
```

3.5 CONTA2D - Cross-sectional ovalization

When lay angles are significant it is recomended to apply this card, otherwise the card may be omitted. Note also that for non-circular geometries or for beam elements with holes the contact direction is more correct when this card is issued. Option for two dimensional cross-section with geometry ovalization **CONTA2D** to be used as follows:

CONTA2D ACTIV [DELX NX]

ACTIV: Dummy

DELX: Longitudinal increment along vertical axis of umbilical, default value is 0.1.

NX: Number **NX** of **DELX**-steps up and **NX** of **DELX**-steps down will be performed along particular helical vertical axis. Default value is 400.

Correct choice of **DELX** and **NX** parameters enables full projection of the 3-dimensional helical object into 2-dimensional ovalized image. The last two input parameters may be skipped from the input, which means that the default values are used. However if obtained geometry will be unexpected, customised choice of the last two input parameters should be done.

EXAMPLE:

```
#       activ  delx   nx
CONTA2D  1      0.1    400
```

3.6 CONTINT - CONTact INTerfaces

In order to optimize the contact search, the contact interfaces need to be defined. The friction coefficient depends on the neighboring material and it is therefore convenient to store this as a matrix, see the **FRICTION** data group, Section 3.20. However, if this matrix is unknown, it is possible to define the necessary properties as part of the present

data group. If the friction properties related to the material names of the two bodies are found in the **FRICTION** data group, that value will always be used by the program. Note however that the **SURFSTIFF** parameter should still be set to a positive and realistic value in the **CONTINT** card, even when it is further given in the **FRICTION** card.

The **CONTINT** data group is defined as follows:

CONTINT MNAME SNAME FRICMATNAME INTERFACE SURFSTIFF FRICOEFF FRICLEV GAP1 GAP2 COCODE [EPSI]

where:

MNAME: The name of the master surface geometry.

SNAME: The name of the neighboring candidate slave surface geometry.

FRICMATNAME: The friction material name defining the stick-slip transition curve (friction interface for unbonded cases) or shear stress interaction curve (for bonded cases). By specifying the word **AUTO**, a contact material of type **FRICSPRING** with automatic calculation of slip level is generated, based on the **TAVRG** parameter of the **CONTROL** card.

INTERFACE: The interface number from which the contact search is started referred to interface **MNAME** above and with respect to the geometry definition. The value is either 1 or 2 for shell, arbitrary for beam. For shell 1 means outside, 2 means inside. For beam 1 means outside, whereas other values refers to the respective internal surface.

SURFSTIFF: Surface stiffness modulus, unit MPa/mm.

FRICOEFF: Friction coefficient.

FRICLEV: Load level to activate friction. In some cases it is useful to enable flexibility with respect to when in the load history friction forces should be activated. The value refers to the load steps defined in the **HIST** data group. If a negative value is given, this indicates that this contact interface is to be taken into account when tuning friction springs at that specified step. See Section 3.26 and Section 3.8.

GAP1: For gaps less than the specified value, contact is introduced from the start of analysis.

GAP2: For gaps less than the specified value, contact is considered during loading.

COCODE: Contact element code. **COCODE** can have the following values:

- **0 :** Ordinary contact element is applied.
- **1 :** Gaps are eliminated, i.e. the contact element will act as a spring.

EPSI: Contact interface buffer, optional parameter, to be used combined with scaling parameters in **CONTROL** card. **EPSI** must correspond to the body sizes involved

in contact. If scale1 parameter from **CONTROL** card is less than 1, and **EPSI** is not explicitly given, the default value **EPSI**=0.2mm is chosen.

Note that the two parameters **GAP1** and **GAP2** should be defined with care. The principles of the contact surface is described below, and may be helpful for the user to define proper values for the parameters. See Fig. 3.1 for illustration.

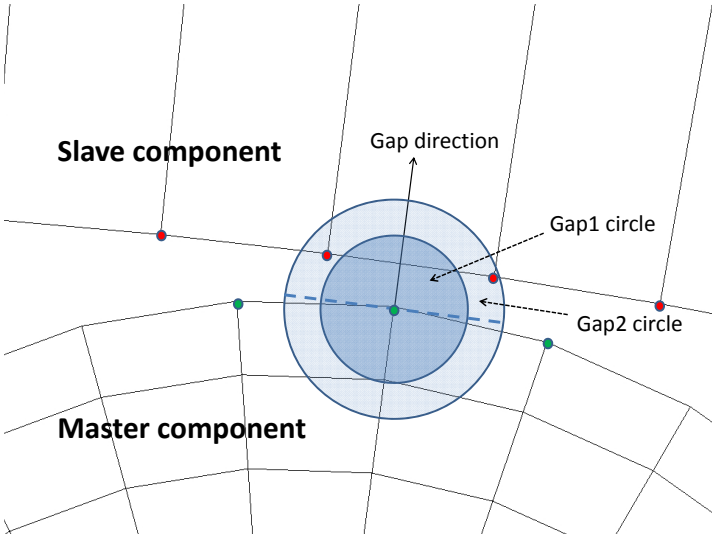


Figure 3.1: Gap definitions.

For each node on the master surface geometry, the closest node on the slave geometry is selected. This selection is made for the initial (no load) configuration. If the initial gap (gap0) between the two nodes measured along the gap direction is smaller than **GAP1** a contact element is defined. The gap between them is eliminated, that means that the contact element will carry load in the cases where gap - gap0 is negative. This will be the case for the slave nodes within the inner circle of Fig. 3.1, in this case none. Any slave nodes with negative gap0 value (below the dotted line) and within the outer circle will also be in contact from the start.

If the initial gap between two nodes is larger than **GAP1** but less than **GAP2**, that is within the large circle but outside the small circle, a contact element will still be defined. However, this element will be activated only if the gap between the two nodes becomes negative. That is the case when the slave node has crossed the dotted line. (True contact)

For the remaining possible contact elements, which have initial gaps larger than **GAP2**, no contact elements will be defined.

The selection of **GAP1** should be in the range of the distance between the two geometries in contact. However, this must be tuned for each case. **GAP2** should be in the range

of expected displacement of the geometries, and should anyway not be larger than the radius (or half the thickness in case of sheaths) of the slave geometry.

A smart strategy to find reasonable gap values is to apply very small tension and bending loads when building the model, and inspect the resulting contact stresses.

Also note that the meshing of the components should preferably be made such that the slave node falls on the gap direction array, see Fig. 3.1, as the direction of the contact element will be in the gap direction.

EXAMPLE:

```
# Name mname sname fricmatname inteface surfstiff fricoeff friclev
# gap1 gap2 cocode CONTINT PIPE1 SURF1 FRIC1 1 1000 0.2 1000 0.003
# 0.003 0
```

3.7 CNTINTFIND - Automatic CONTINT search

This feature is currently under development.

Card for automatic contact interface **CONTINT** list generation, see Section 3.6. In case of complex cross-section with numerous cross-sectional geometries, it may become tedious to create the list of all possible **CONTINT**s manually. Card **CNTINTFIND** is to be used as follows:

CNTINTFIND NONE

CNTINTFIND card does not have any parameters. In case **CNTINTFIND** appears in the input file, a text file CONTINT.txt will be generated in the same folder where the input file is located. The program execution will then be terminated. File CONTINT.txt contains a list of all possible contact **CONTINT**s, with some default **CONTINT** parameters: friction, surface stiffness etc. The user may adjust those parameters, then paste updated **CONTINT**s into the original input file and run it. The user must remove or comment out the **CNTINTFIND** card in the input file, in order for the analysis to be executed.

NONE: This card does not have any parameters.

N.B. If using **CNTINTFIND**, it is advised to arrange more simple **GEOM**s to appear in the geometries list before complex geometries (with multiple interfaces) in the input file.

EXAMPLE:

```
# activate
CNTINTFIND
```

3.8 CONTROL - CONTROL parameters

The control parameters are defined using one data card starting with **CONTROL** and defined as follows:

CONTROL **NLPRINT** **MAXIT** **CONVR** **SEADEN** **GACC** **AVRGT**

NLPRINT: Number of load levels with print. By entering a negative number, output of iterations to screen is avoided.

MAXIT: Maximum number of iterations.

CONVR: Convergence radius, recommended value: 10e-5

SEADEN: Sea water density. Unit: kg/mm^3

GACC: Acceleration of gravity. Unit: m/s^2

AVRGT: Average tension of system. If negative value, this is specifying the step for tuning friction springs. Unit: N if positive. See below for more details.

The parameter **AVRGT** is used to automatically determine the slip level of contact interfaces, unless **MATERIAL** of type **FRICSPRING** is applied with a negative hardening parameter. In that case, the parameter is dummy.

Instead of determining slip levels from a positive value of **AVRGT**, the actual contact forces and axial tensions at a given load step can be used. This is referred to as tuning of the friction spring slip levels.

Note that for tuning at a certain step based on the actual contact forces at this step, the load step to be used for tuning must be specified in the corresponding **CONTINT** card. See Section 3.6, Section 3.32.7, Section 3.20, Section 3.19 and Section 3.40 for input cards related to friction and tuning of slip levels.

EXAMPLE:

```
#       nlprint maxit convr   seaden     gacc  avrgt
CONTROL 30      40    1.0E-3  1025.0E-9  9.81  -10
```

3.9 CROSS - CROSS-section definition

The cross-section is defined using the Nexan recursive concept, where the cross-section is divided into elements and sub-elements until the last element is uniquely defined in terms of it's geometry. However, in order to uniquely define the geometry with respect to the needs for FEM analysis, some modifications have been made compared to the example attached in the Nexan specification (bib,).

The cross-section is defined by a number of bodies each having it's specific and unique name. The position of each body is defined using polar coordinates. Note that in order to enable consistent definition of the model, multiple bodies having the same position in terms of their polar coordinates must be defined as separate groups. This means that a certain combination of radius and angle can not be repeated.

The cross-section is defined by the following line format.

CROSS LINENO NAME PROCESS PROCNO PITCH RAD THETA ELTYPE GEONAME MATNAME

LINENO: Line number, dummy.

NAME: The name of the process/component. The name is a label, and is only used in connection with the next parameter **PROCESS** to identify if the component has sub structures or not.

PROCESS: An alphanumeric parameter used to indicate whether the present geometry consists of smaller bodies that need to be defined in more detail. Note that description of the sub-groups need to be in the last **CROSS** cards that are defined.

NAME : The body is uniquely defined by the parameters defined in the same line.

NEXT : The geometry has to be defined in more detail by a sub-groups of second order of higher, ie. subdefinition of helices. Search is thencarried out in the **CROSS** input data until **NAME** = **PROCESS** is found.

PROCNO: The number of processes. If the number is > 1 then distribute these bodies evenly from **THETA** (see below) to **THETA** $+ 360°$ according to the number of processes. This is convenient for armour layers consisting of a large number of bodies at each layer.

PITCH: The pitch length, $2\,\pi$ **RAD**$/\tan(\alpha)$, where α is the lay angle positive for right hand lay. If pitch $\neq 0$ then the body is helically wound.

RAD: The body radius to the centroid of the group.

THETA: The angular position in degrees. See Fig. 3.2

ELTYPE: The element type.

SHELL : for tubulars.

BEAM : for filled bodies.

BEAMSH : for filled bodies. Reference is given to the theory manual (Sævik, 2004b).

GEONAME: Geometry name. A unique geometry name must be used for each body.

MATNAME: Material name. Each body is also linked to an unique material name, where each material type is defined in the **MATERIAL** data group.

Note: If the **CROSS** object has a radius **RAD** different from zero, pitch length **PITCH** must be different from zero. Zero pitch length is only allowed for center objects with radius zero, as pitch length in this case is dummy. However, when the **DISPLACE** command is issued the center objects need to have a proper pitch length if rotation of these objects are required. A filler would then need to be defined with the same pitch length as the objects that it holds.

Note: For the center objects, **THETA** is the rotation about origo.

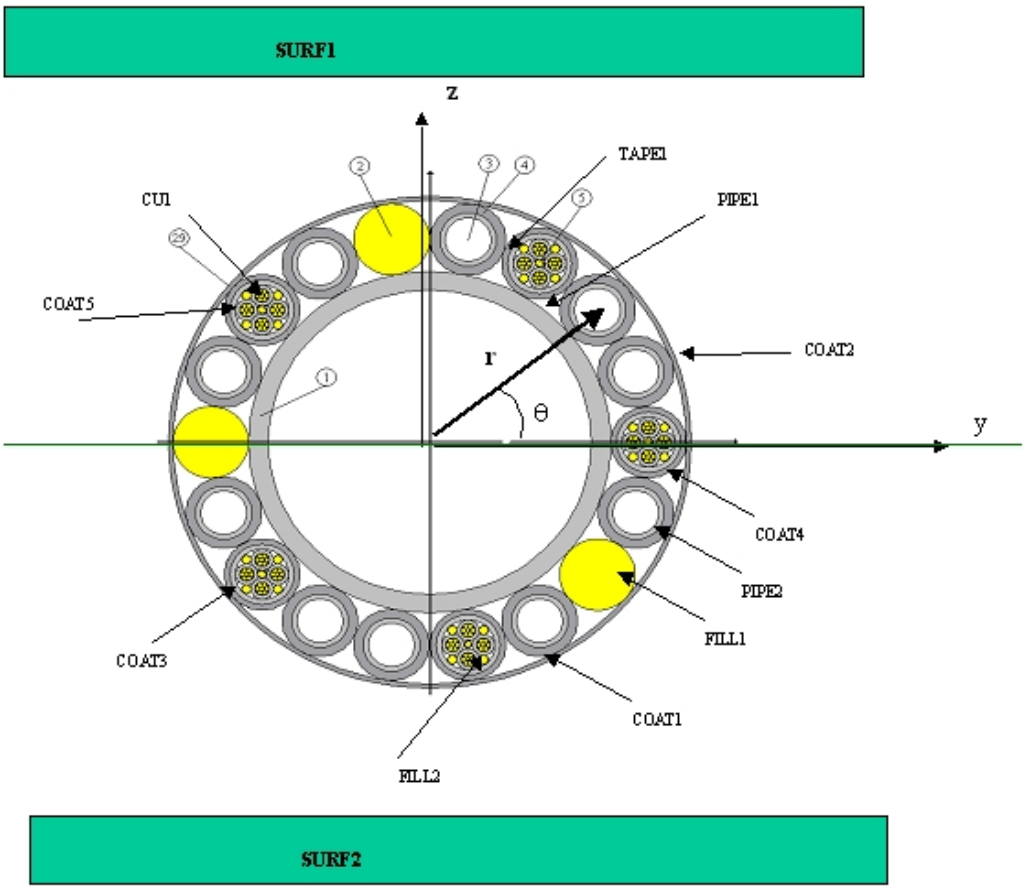


Figure 3.2: Cross-section geometry description.

EXAMPLE:

```
#     Line Name  Process Procno Pitch    Rad  Theta   Eltype  Geoname  Matname
CROSS 1    FYLL  FYLL    1          0      0     0    BEAM    FYLL1    PLASTIC1
CROSS 1    ROR1  NEXT    1      -2400  33.50     0
CROSS 1    ROR1  NEXT    1      -2400  33.50    60
CROSS 1    ROR1  NEXT    1      -2400  33.50   120
CROSS 1    ROR1  NEXT    1      -2400  33.50   180
CROSS 1    ROR1  NEXT    1      -2400  33.50   240
CROSS 1    ROR1  NEXT    1      -2400  33.50   300
#
CROSS 1    ROR1  ROR1
CROSS 1    ROR   ROR     1          0      0     0    SHELL   PIPE1    DUPLEXSIGY450
CROSS 1    KAPPE KAPPE   1          0      0     0    SHELL   KAPP1    PLASTIC2
```

3.10 DISPLACE

By the card **DISPLACE**, the user can displace the complete cross section in the longitudinal direction. The pitch lengths will be applied to find new angles for the objects. Note that the center objects (with radius = 0) will not be rotated if they have zero pitch length. To obtain a proper rotation for eg. a center filler, the pitch length for the filler should be set equal to the pitch length of its related objects.

DISPLACE LDISP

The only parameter required is:

LDISP: Length of displacement.

The card can be given several times. When issued several times the displacement will be the sum of the given displacements.

3.11 EXP3DARMOUR

The **EXP3DARMOUR** command allows the user to reduce the number of members that will be exported to the 3d profile for a specified geometry. This is convenient for armour layers. The stiffness for the 3D armouring layer will be increased to maintain the same total stiffness as for the 2D profile, but the area of each armour member will be kept the same as for the 2D case.

The following format applies:

EXP3DARMOUR GEONAME NUM ANG

GEONAME: Geometry name

NUM: Number of armour members in 3d profile

ANG: Angle of first member in 3d profile

In the example below, the number of *pipe2* geometry members have been reduced to two. The first of them will be positioned at angle 0. In UFLEX3D this will be along the negative z-axis.

EXAMPLE:

```
EXP3DARMOUR  PIPE2  2  0
```

3.12 EXP3DCONTROL

The **EXP3DCONTROL** card defines parameters of the 3D export, and must be present for 3D export to take place. To avoid 3D export, it is sufficient to remove the **EXP3DCONTROL** card.

EXP3DCONTROL FILENAME3D LENGTH SECTION NEL

FILENAME3D: Name of 3d file. Extension 3if will be added

LENGTH: Axial length of the 3d section. Unit: mm, will be translated to m in 3d profile

SECTION: Location of the 2d section along the 3d profile. Value ban be between **0** and **LENGTH**, where **0** means that the 2d cross section is located at the start of the 3d profile. **LENGTH** means that the 2d cross section is located at the end of the 3d profile. **SECTION** will be located at x=0.

NEL: Number of elements in axial direction for the 3d profile. Note that if a **EXP3DELDEF** card is issued, this parameter will be overruled.

EXAMPLE:

```
EXP3DCONTROL deimos_3d  3000  0  9
```

3.13 EXP3DCORE

When exporting a 3d profile, the core must be defined by the **EXP3DCORE** command. If more than one geometry is specified, the structural properties of all the geometry will be summed and used for the core.

Note that it is required for the 3D simulation to have a first order pipe, ie. a core or a outer sheet.

EXP3DCORE GEONAME ... ...

GEONAME: Geometry names

EXAMPLE:

```
EXP3DCORE CPIPE CTAPE
```

3.14 EXP3DELDEF

By the **EXP3DELDEF** card, the user can define varying element lengths along the x-axis. In this case, the number of elements specified in the **EXP3DCONTROL** card, see Section 3.12, will be ignored. Constant element length is assumed for the region ahead of first x-location with an element length definition, and behind the last x-location.

If different element lengths are specified for two points, UFLEX2D will gradually change the element lengths in the area between them.

If the definition is such that the region can not be covered of a integer number of elements, the number of elements required will be rounded upwards, which means that the region must be expanded. After all node positions have been defined, the x-coordinates are adjusted by a factor to ensure that the last node will be positioned in the end of the umbilical. This means that the point **X1** ect. is not necessarily located at a node, and that the element lengths may be slightly smaller than specified.

EXP3DELDEF TYPE X1 LENGTH1
... ...

TYPE: Method to use for changing element length. Options: **LINEAR** or **PERCENT**.
X: Location along x-axis.
LENGTH: Element length at the given location.

In the example below, constant element length of 100 will be applied in the first region of length 950. This requires 9.5 element, and is then rounded upward to a number of 10 elements. The next region is of length 450 and contains elements of decreasing length. If the **LINEAR** option has been selected, the region will be exactly covered by 6 elements, starting at length 100 and decreasing by 10 for each element until the

element length is 50. Given an umbilical length of 2000, the remaining length of 600 will be covered by 12 elements of length 50.

The total length defined is now 2050, and all element lengths must be reduced by the factor 2000/2050 to ensure that the last node is at the end of the umbilical.

EXAMPLE:

```
EXP3DELDEF  linear        950      100
                         1400       50
```

3.15 EXP3DGEOM

If a member is required to have a geometry other than tube in the 3D profile, the **EXP3DGEOM** card must be issued. This allows the user to specify that a rectangular shape should be applied, typically for armour members.

The format is as follows:

EXP3DGEOM CODE GEONAME WIDTH THICK

CODE: Which type of geometry to export, only **REC** is allowed, and means rectangular geometry. Default is tube.

GEONAME: Name of the geometry.

WIDTH: Width of one armour tendon.

THICK: Thickness of one armour tendon.

EXAMPLE:

```
EXP3DGEOM  REC PIPE2  40  20
```

3.16 EXP3DLAYER

To export the 3d model, the user needs to define the different layers to be exported. For each layer, the user must issue a **EXP3DLAYER** card. The order of the cards matters, in the sense that the layer cards should be issued with innermost layer first and the outermost as the last layer card. The 3d model will only contain 1. order straight tubes and 1. order helices. Higher order members from the 2d model will be combined with the 1. order helix they are wound about.

It is sufficient to refer to *one* geometry in a complex structure, and the complete structure will be combined and defined in the 3d model. The name of the 2d member will contain the referred geometry.

EXP3DLAYER GEONAME ... ...

GEONAME: Name of geometry

In the example below, the *pipe2* geometry will be the first layer. *pipe3* and *pipe4* will be the second layer, and the outer layer will have the geometries *pipe5*, *fill1* and *quad1*.

EXAMPLE:

```
EXP3DLAYER PIPE2
EXP3DLAYER PIPE3 PIPE4
EXP3DLAYER PIPE5 FILL1 QUAD1
```

3.17 EXP3DOUTER

To export an outer sheating, the relevant geometry must be specified by the **EXP3DOUTER** card. If more than on geometry is referenced, the properties off all geometries will be combined.

EXP3DOUTER GEONAME ... ...

GEONAME: Name of geometry.

EXAMPLE:

```
EXP3DOUTER OSHEATH
```

3.18 EXTCON - EXTternal CONtact definition

The external contact bodies need to be uniquely defined in terms of geometry, element type and material properties in a similar way as the cross-section data group.

The format is as follows:

EXTCON NAME Y0 Z0 ELTYPE GEONAME MATNAME

where the parameters have the following meaning:

NAME: The first order name.

Y0: Y-position of the contact surface centroid relative to the umbilical centroid.

Z0: Z-position of the contact surface centroid relative to the umbilical centroid.

ELTYPE: Element type. Only **CONSHELL** allowed.

GEONAME: Geometry name. The geometry need to be defined as for the cross-section in the **GEOM** data group.

MATNAME: Material name. The material properties need to be defined as for the cross-section in the **MATERIAL** data group.

EXAMPLE:

```
#       Name     y0 z0      Eltype     Geoname        Matname
EXTCON CONTACT 0   11.60   CONSHELL   SURF1          CONT1
```

3.19 FRICMODEL - choice of friction model

This input card allows the user to choose between two friction models. The first model (**FRICMODEL** 0) is based on work accumulated in friction springs. This is the oldest of the models. Prior to UFLEX2D version 2.8 it was the only model. The second model (**FRICMODEL** 1) is based on known strains during stick, which are the plane surfaces remain plane strains.

The card should be specified only once in the input file. If the card is omitted, the first model is used. This is to prevent the user from being unknowingly surprised by new results. We recommend using **FRICMODEL** 1, because **FRICMODEL** 0 is known to be over-conservative with respect to bending stiffness.

FRICMODEL **FRICID**

The parameter is defined as follows:

FRICID: The friction model to be used. **FRICID** can have the following values:

- **0 :** Friction based on accumulated work in friction springs.
- **1 :** Friction based on known strains during stick.

EXAMPLE:

```
#         FricId
FRICMODEL 1
```

3.20 FRICTION - FRICTION properties

In order to optimise the contact search, the contact interfaces need to be defined, see Section 3.6. The friction coefficient depend on the neighbouring material and it is therefore convenient to store this by the **FRICTION** card, as defined below:

FRICTION MATNAMEA MATNAMEB STIFF COEFF

The parameters are defined as follows:

MATNAMEA: The material name of one surface.

MATNAMEB: The material name of the candidate surface.

STIFF: Surface stiffnes modulo, unit: MPa/mm.

COEFF: Friction coefficient.

The card can be repeated as many times as needed.

For a contact interface wich has its material-pair listed in a **FRICTION** card, the values from the **FRICTION** card will overrule those given in the **CONTINT** card. Note however that for the stiffness, it is still required that a positive and realistic value is supplied in the **CONTINT** card.

When friction between two different materials is defined, the friction parameter will be valid for contact in both directions for the two materials. It is not relevant which material belongs to the slave or the master geometry.

EXAMPLE:

```
#        Matnamea       Matnameb       stiff  coeff
FRICTION DUPLEXSIGY450 DUPLEXSIGY450 116    0.2
```

3.21 GCLOAD - Global Concentrated LOADs

The data group is defined by:

GCLOAD DIR VAL HIST

where:

DIR: Dir that load are applied in applies for. Values can be:

- **1 :** load x-direction

2 : load y-direction

3 : load z-direction

4 : torsion moment about x-axis

5 : bending moment about y-axis

6 : bending moment about z-axis

VAL: Value. The load or moment, unit: N or Nmm.

HIST: The load history number (may be an arbitrary number as long as the same number is found in the **HIST** data group) See Section 3.26.

EXAMPLE:

```
#       Dir  Val     Hist
GCLOAD 1     7000    100
```

3.22 GEOM - GEOMetries

The geometry is defined by a local right handed Cartesian xyz-coordinate system which is related to the global pipe coordinate XYZ system (see Fig. 3.2) as shown in Fig. 3.3. The XYZ system is positioned at the umbilical centroid. The local x-axis points along the helix.

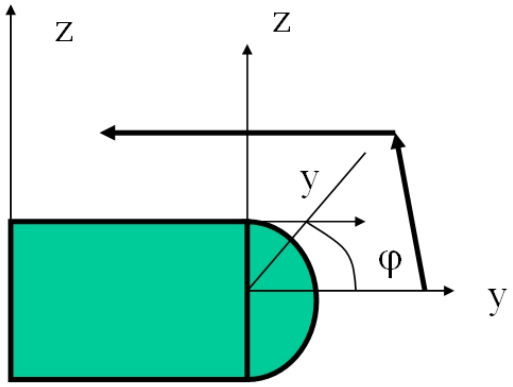


Figure 3.3: Axis-systems.

The geometry is defined by boundary curves where the boundary is divided into a number of segments which again may be divided into an arbitrary number of intervals **NINTER**. The geometry of the boundary curves is always to be specified in counter-clockwise order about the local x-axis.

GEOM NAME FORM Y0 Z0 CURVCODE P1 P2 P3 P4 NINTER INTERFACE ... ...

NAME: The name of the geometry.

FORM: In order to uniquely define the meshing of the shell elements, it is necessary to define whether the cross-section is **OPEN** or **CLOSED**.

Y0: In order to describe the geometry of hollow section BEAM elements it is necessary to use two boundary curves. **Y0** defines the y-start point of the inner boundary curves relative to where the outer boundary curve starts.

Z0: In order to describe the geometry of hollow section BEAM elements it is necessary to use two boundary curves. **Z0** defines the z-start point of the inner boundary curves relative to where the outer boundary curve starts.

CURVCODE: Curve code, given as a character string which may have the following values:

S : Segment is straight

CI : Segment is circular with the circle centre at the cross section inside.

CO : Segment is circular with the circle centre at the cross section outside.

P1: If **CURVCODE** = **S**: length of segment, unit: mm. Only positive lengths are allowed. If **CURVCODE** = **CI** or **CO**: angle at start of circle, unit: deg.

P2: If **CURVCODE** = **S**: angular orientation of segment local YZ system, where the Y-axis is directed along the element and the Z-axis is pointing outwards, unit: deg. If **CURVCODE** = **CI** or **CO**: angle at end of circle, unit: deg.

P3: Circle radius. Note that when specifying shells, the radius is always the outer radius, unit: mm. Dummy for **CURVCODE** =**S**.

P4: Shell thickness, unit: mm. Dummy for BEAM and BEAMSH elements.

NINTER: Number of intervals within the segment. The element meshing in the structural model depends on this parameter. *Note:* For helical shells **NINTER** should be 2, 4, 8, 16, 32, 64 ..... For beamshell (BEAMSH elements) geometries, it is preferred that the boundary is defined with nodes exactly on the top, at the bottom and at the leftmost and rightmost location of the geometry. This will minimize unintended effects from constraints that are applied to obtain stability.

INTERFACE: The local interface number. For BEAM and BEAMSH elements the number of interfaces may be arbitrary. For SHELL, only **INTERFACE** = **1** is allowed. Interface numbers higher than 1 are used to describe holes in a geometry.

Maximum 100 different geometries can be defined.

EXAMPLE:

```
#    Name   FORM   X0 Y0 Curvcode P1   P2   P3    P4     NINTER INTERFACE
```

```
GEOM PIPE1  CLOSED 0  0  CI        0  360  9.10  1.1    40     1
GEOM SURF1  OPEN   0  0  S        35    0  0     5      20     1
GEOM SURF2  OPEN   0  0  S        35  180  0     5      20     1
```

3.23 GLBON - Global BOuNdary conditions

The boundary conditions may be specified in two ways:

- As local boundary conditions which applies to each individual body.
- As global boundary conditions which applies to the overall umbilical.

The global boundary conditions applies to the overall pipe structure in terms of:

- Axial, transverse and vertical displacements.
- Torsion.
- Bending.

The **GLBON** data group includes only the direction parameter:

GLBON DOF

where:

DOF: Dof that the boundary condition applies for. Values can be:

- **1 :** x-direction
- **2 :** y-direction
- **3 :** z-direction
- **4 :** torsion, rotation about x-axis
- **5 :** bending about y-axis
- **6 :** bending about z-axis

EXAMPLE:

```
#       Direction
GLBON   2
GLBON   3
GLBON   4
GLBON   5
GLBON   6
```

3.24 GPDISP - Global Prescribed DISPlacements

Global prescribed displacement can be applied by the **GDISP** data group:

GPDISP DIR VAL HIST

where:

DIR: Direction that displacement is applied in. Values can be:

- **1 :** x-direction
- **2 :** y-direction
- **3 :** z-direction
- **4 :** torsion x-direction
- **5 :** curvature y-direction
- **6 :** curvature z-direction

VAL: Displacement or torsion or curvature, unit: mm, radians/mm or 1/mm.

HIST: The load history number (may be an arbitrary number as long as the same number is found in the **HIST** data group) See Section 3.26.

EXAMPLE:

```
#      Dir  Value Hist
GPDISP 1    10    100
```

3.25 HEAD - HEADing

The **HEAD** command is used to supply a descriptive text to the analysis. Up to six lines of text may be used to describe the analysis. Each line is started with **HEAD** followed by an arbitrary text string. The format is as follows:

HEAD TEXT

where:

TEXT: Descriptive analysis text.

EXAMPLE:

```
#
HEAD ANTLANTIS Umbilical
HEAD Bending test
#
```

3.26 HIST - load HISTories

The **HIST** data group is defined by:

```
HIST HISTNO LOADLEVEL VALUE
             ..        ..
```

Pairs of **LOADLEVEL** and **VALUE** can be repeated as many times as required.

The parameters are defined as:

HISTNO: Load history ID. To be referenced in the different load definitions etc.

LOADLEVEL: Load level number.

VALUE: The load scaling factor of the load at a certain loadlevel. A reference value, L_0, for the load is given in the load definitions. The applied load at a certain loadlevel will then be: $L_0 \times$ **VALUE**.

EXAMPLE:

```
#     Histno  Loadlevel   Value
HIST  100             1   0.001
                      2   0.010
                     10   1.000
```

3.27 INISTRAIN - INItial STRAIN

Initial strain can be applied by the **INISTRAIN** data group:

```
INISTRAIN GEONAME DIR VAL HIST
```

where:

GEONAME: Name of geometry.

DIR: Direction that strain is applied in. Values can be:

- **1 :** axial strain
- **4 :** torsion
- **5 :** curvature about local y-axis
- **6 :** curvature about local z-axis

-5 : initial strain automatically determined from helix curvature. (To evaluate manufacturing effects)

-6 : initial strain automatically determined from helix curvature. (To evaluate manufacturing effects)

VAL: Strain, torsion or curvature, unit: -, rad/mm or mm^{-1}.

HIST: The load history number (may be an arbitrary number as long as the same number is found in the **HIST** data group) See Section 3.26.

EXAMPLE:

```
#         Geoname  Dir  Val    Hist
INISTRAIN PIPE2    1    0.001  100
```

3.28 LCLOAD - Local Concentrated LOADs

The data group is defined by:

LCLOAD GEONAME NODE DIR CODE VAL HIST

where:

GEONAME: The geometry name.

NODE: Nodal position point. The following six values are allowed for SHELL, BEAMSH and CONSHELL (for BEAM only **MID** is applicable):

LEFT : The load is introduced at the node of minimum local Y-coordinate

MID : The load is introduced at the node of maximum local Y-coordinate

RIGHT : The load is introduced at the node of mean local Y-coordinate

UPPER : The load is introduced at the node of maximum local Z-coordinate

LOWER : The load is introduced at the node of minimum local Z-coordinate

MIDHEIGHT : The load is introduced at the node of maximum local Z-coordinate

DIR: Direction that load are applied in. Values can be:

1 : x-direction

2 : y-direction

3 : z-direction

CODE: Code. Allowable values are: 1/2 = the element loads will be in the local element direction.

VAL: Value. The load, unit: N

HIST: The load history number (may be an arbitrary number as long as the same number is found in the **HIST** data group) See Section 3.26.

See also Section 2.8 for a description of the local systems of different element types.

EXAMPLE:

```
#       Geoname  Node  Dir  Code  Val    Hist
LCLOAD  SURF2    MID   3    1     -1000  100
```

3.29 LMERGE - Layer MERGE definition

For armour layers in contact, the 2D assumption implies that each body move individually according to the contact conditions relevant for the considered node. This may imply local stress effects which is not representative for the average stress condition applied in industry standards. This is due to the fact that the bodies are helically wound having alternating contact with other bodies in the longitudinal direction. Since stress checking is based on the average stress concept, it is important to enable simulation of these effects in the numerical model. This is performed by introducing the layer merge command as follows:

LMERGE GEONAME [RELOPT]

where:

GEONAME: Name of geometry to be merged. The optional argument **[RELOPT]** can have the following values:

- **0 :** No relaxation of the constraints on SHELL geometries (default).
- **1 :** Relaxation of the constraints on SHELL geometries, to avoid excessive bending stresses.

The effect of merging a beam layer is that all the beams in the layer will have the same radial translation. UFLEX2D achieves this as visualized in Fig. 3.4. By defining the first body to be a master body and the rest of the bodies to be slaves in the radial dof, they will all behave identical when it comes to radial motion. The user does not have to consider which body to be the master body.

The effect of merging a shell layer is visualized in Fig. 3.5. The first body will act as a master body, in the sense that radial and tangential dofs at the nodes of the master body will be master dofs. The radial and tangential dofs at the corresponding nodes of the slave bodies will behave identical to the master. As for beam, the master-slave

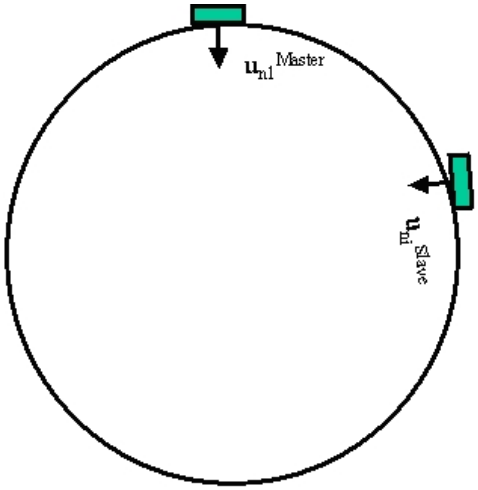


Figure 3.4: Merging a beam layer.

principle is applied internally in UFLEX2D, and the user does not need to consider which object is master or slave.

Merging a beamshell layer has the same effect as the merge of a shell layer. The first body will act as a master body when it comes to tangential and radial motion. That means that all the members within the layer will move in the same way. It also means that the local deformations will be the same for all bodies within the layer.

An option exists to relax the contraints applied to merged shell layers. If this is done, only the radial dof at the center of the master body is applied as a constraint on the corresponding nodes of slave bodies. The purpose of this option is to avoid excessive stresses on shells during bending. The option has no effect on other geometries than shell.

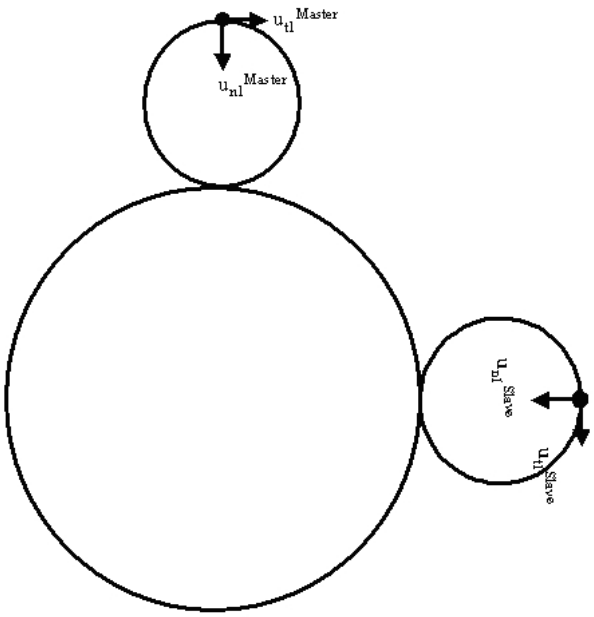


Figure 3.5: Merging a shell layer.

EXAMPLE:

```
#       Geoname  RelOpt
LMERGE PIPE1    1
```

3.30 LOBON - Local BOuNdary conditions

The boundary conditions may be specified in two ways:

- As local boundary conditions which applies to each individual body.
- As global boundary conditions which applies to the overall umbilical.

The local boundary conditions are linked to the geometry names and are specified as follows by the **LOBON** card:

LOBON GEONAME NODE DOF CODE

where:

GEONAME: The geometry name.

NODE: Nodal position point. The following six values are allowed for SHELL, BEAMSH and CONSHELL (for BEAM only **MID** is applicable):

LEFT : Boundary condition is introduced at the node of minimum local Y-coordinate

MID : Boundary condition is introduced at the node of mean local Y-coordinate

RIGHT : Boundary condition is introduced at the node of maximum local Y-coordinate

UPPER : Boundary condition is introduced at the node of maximum local Z-coordinate

LOWER : Boundary condition is introduced at the node of minimum local Z-coordinate

MIDHEIGHT : Boundary condition is introduced at the node of maximum local Z-coordinate

DOF: Dof that boundary condition applies for. Values can be:

1 : x-direction

2 : y-direction

3 : z-direction

4 : rotation about x-axis

5 : rotation about y-axis

6 : rotation about z-axis

CODE: Allowable values for shell, beamshell and beam nodes are: 1-2 = fixed in local direction. Allowable values for conshell nodes are: 1/2 = fixed in global/local direction, 4 = fixed in all dofs, which means that all dofs related to the geometry are eliminated from the final equation system.

See also Section 2.8 for a description of the local systems of different element types.

One typical use of this will be to prevent a first order outer coating from rotating. This can be done by the two or four first lines in the example below, keeping the hoop direction fixed at two or four points along the coating.

EXAMPLE:

```
#      Geoname  Node   Dof   Code
LOBON  PIPE1    UPPER  1      1
LOBON  PIPE1    LOWER  1      1
LOBON  PIPE1    LEFT   1      1
LOBON  PIPE1    RIGHT  1      1
#
LOBON  SURF1    LEFT   1      1
LOBON  SURF1    LEFT   2      1
LOBON  SURF1    LEFT   4      1
LOBON  SURF1    LEFT   5      1
LOBON  SURF1    LEFT   6      1
LOBON  SURF1    RIGHT  1      1
```

```
LOBON  SURF1    RIGHT  2      1
LOBON  SURF1    RIGHT  4      1
LOBON  SURF1    RIGHT  5      1
LOBON  SURF1    RIGHT  6      1
LOBON  SURF2    LEFT   1      1
LOBON  SURF2    LEFT   2      1
LOBON  SURF2    LEFT   4      1
LOBON  SURF2    LEFT   5      1
LOBON  SURF2    LEFT   6      1
LOBON  SURF2    RIGHT  1      1
LOBON  SURF2    RIGHT  2      1
LOBON  SURF2    RIGHT  4      1
LOBON  SURF2    RIGHT  5      1
LOBON  SURF2    RIGHT  6      1
```

3.31 LPDISP - Local Prescribed DISPlacements

Local prescribed displacement can be applied by the **LDISP** data group:

LPDISP GEONAME NODE DIR CODE VAL HIST

where:

GEONAME: Geometry name.

NODE: Nodal position point. The following six values are allowed for SHELL and CONSHELL (for BEAM only **MID** is applicable):

- **LEFT :** Prescribed displacement is introduced at the node of minimum local Y-coordinate
- **MID :** Prescribed displacement is introduced at the node of maximum local Y-coordinate
- **RIGHT :** Prescribed displacement is introduced at the node of mean local Y-coordinate
- **UPPER :** Prescribed displacement is introduced at the node of maximum local Z-coordinate
- **LOWER :** Prescribed displacement is introduced at the node of minimum local Z-coordinate
- **MIDHEIGHT :** Prescribed displacement is introduced at the node of maximum local Z-coordinate

DIR: Direction that displacement is applied in. Values can be:

- **1 :** x-direction

2 : y-direction

3 : z-direction

4 : rotation about x-axis

5 : rotation about y-axis

6 : rotation about z-axis

CODE: Allowable values for beam and shell nodes: 1/2 = local direction. Allowable values for conshell nodes are: 1/2 = global/local direction.

VAL: Displacement or rotation, unit: mm or radians.

HIST: The load history number (may be an arbitrary number as long as the same number is found in the **HIST** data group) See Section 3.26.

See also Section 2.8 for a description of the local systems of different element types.

EXAMPLE:

```
#       Geoname  Node  Dir  Code  Val  Hist
LPDISP  SURF2    MID   3    1     -10  100
```

3.32 MATERIAL - MATERIAL properties

The following material types are available in version 2.8.8 of UFLEX2D:

LINEAR : Linear material properties.

ELASTOPLASTIC : Elastoplastic strain-stress material.

TERMOELASTOPL : Termoelastoplastic strain-stress material.

HYPERELASTIC : Hyperelastic (nonlinear elastic) strain-stress material.

RESULTANT : User defined material property based on describing the material curve on resultant level.

EPCURVE : Elastoplastic shear stress - relative displacement, force-strain or moment-curvature behaviour.

HYCURVE : Hyperelastic (nonlinear elastic) shear stress - relative displacement, force-strain or moment-curvature material behaviour.

FRICSPRING : Elastoplastic displacement - unit force material behaviour.

SHEARSPRING : Hyperelastic displacement - shear stress (per unit length) material behaviour.

CONTENT : Content, only weight and insulation.

FIXCONTENT : Fixed content, only weight.

ANISOTR : Linear material with anisotropic nature. To be used only for SHELL with different mechanical properties in radial and longitudinal directions.

For beamshell, **LINEAR** is the only material type allowed.

The format is as follows:

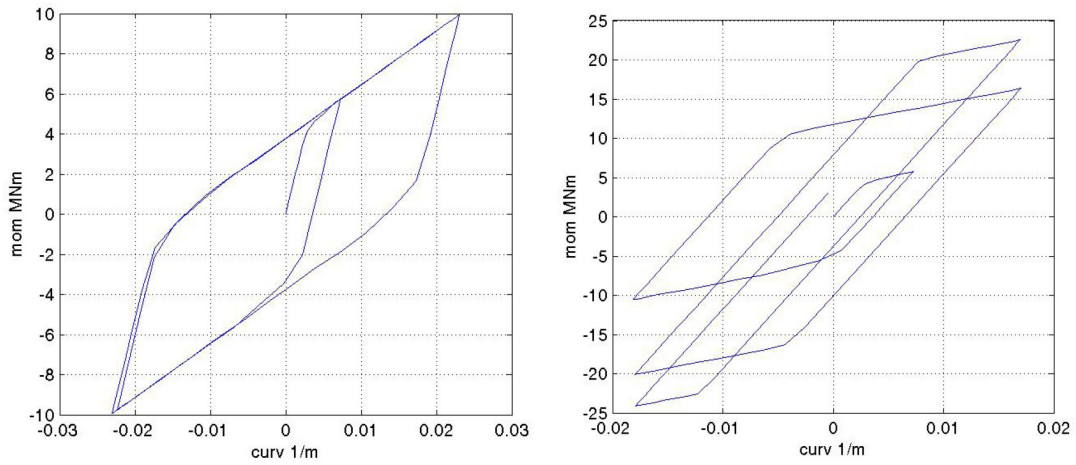
```
MATERIAL MNAME MTYPE ...
```

where

MNAME: Material name.
MTYPE: Material type as defined above.

3.32.1 ELASTOPLASTIC

Elastoplastic strain-stress material behaviour with kinematic/isotropic hardening (applicable for beam and shell only). The difference between kinematic and isotropic hardening is illustrated in Fig. 3.6. Reference is made to (Mroz, 1973) for description of material models.



(a) Kinematic hardening (b) Isotropic hardening

Figure 3.6: Illustration of material hardening effect.

The following format applies:

```
MATERIAL MNAME ELASTOPLASTIC
HPAR POISS DENS TELONG TCOND HCAP STRAIN STRESS
                                    ...      ...
```

where:

HPAR: Hardening parameter Values: from 0 to 1, where:

0 : Isotropic hardening

1 : Kinematic hardening

POISS: Poissons ratio, unit: - .

DENS: Density, unit: kg/mm^3.

TELONG: Temperature elongation coefficient, unit: -. Dummy unless **TLOAD** is given.

TCOND: Thermal conductivity, unit: W/mK. Dummy.

HCAP: Heat capacity, unit: J/kgK. Dummy.

STRAIN: Strain, unit: -.

STRESS: Stress, unit: MPa.

The **STRAIN** and **STRESS** pair can be repeated as many times as reqired.

EXAMPLE:

```
#        Mname   Mtype          Hpar Poiss Dens    Telong   Tcond Hcap Strain Stress
MATERIAL DUPLEX  ELASTOPLASTIC 0.0  0.3   7850E-9 1.00E-05 50    500  0          0
                                                                       0.00166 350.0
                                                                       0.005   450.0
                                                                       0.1     550.0
```

3.32.2 TERMOELASTOPLASTIC

Termoelastoplastic strain-stress material behaviour with kinematic/isotropic hardening (applicable for beam only). Analagous to the elastoplastic material, with linear interpolation within temperature range. Material-temperature curves are user supplied as follows:

```
MATERIAL MNAME TERMOELASTOPLASTIC
HPAR POISS DENS TELONG TCOND HCAP TEMP1 –
                                  STRAIN STRESS
                                  ...    ...
                                  TEMP2 –
                                  STRAIN STRESS
                                  ...    ...
```

where:

HPAR: Hardening parameter Values: from 0 to 1, where:

0 : Isotropic hardening

1 : Kinematic hardening

POISS: Poissons ratio, unit: - .

DENS: Density, unit: kg/mm^3.

TELONG: Temperature elongation coefficient, unit: -. Dummy unless **TLOAD** is given.

TCOND: Thermal conductivity, unit: W/mK. Dummy.

HCAP: Heat capacity, unit: J/kgK. Dummy.

STRAIN: Strain, unit: -.

STRESS: Stress, unit: MPa.

The **STRAIN** and **STRESS** pair can be repeated as many times as reqired.

EXAMPLE:

```
#        Mname    Mtype             Hpar Poiss  Dens     Telong  Tcond Hcap Strain Stress
MATERIAL PLASTIC TERMOELASTOPLASTIC 0.0  0.3   7850E-9 1.00E-05 50    500  0.00   -
                                                                            0.00   0.00
                                                                            0.01   100.00
                                                                            0.02   150.00
                                                                            0.03   200.00
                                                                            10.00  -
                                                                            0.00   0.00
                                                                            0.0095 95.0
                                                                            0.02   147.5
                                                                            0.03   197.5
```

3.32.3 HYPERELASTIC

Hyperelastic strain-stress material behaviour, ie. non-linear elastic. (applicable for beam only).

The following format applies:

MATERIAL MNAME HYPERELASTIC
POISS DENS TELONG TCOND HCAP STRAIN STRESS
... ...

where:

POISS: Poissons ratio, unit: - .

DENS: Density, unit: kg/mm^3.

TELONG: Temperature elongation coefficient, unit: -. Dummy unless **TLOAD** is given.

TCOND: Thermal conductivity, unit: W/mK. Dummy.

HCAP: Heat capacity, unit: J/kgK. Dummy.

STRAIN: Strain, unit: -.

STRESS: Stress, unit: MPa.

The **STRAIN** and **STRESS** pair can be repeated as many times as required.

EXAMPLE:

```
#        Mname    Mtype         Poiss Dens    Telong    Tcond Hcap Strain  Stress
MATERIAL PLASTIC1 HYPERELASTIC  0.3   950E-9  1.00E-05  50    500  -10    -550
                                                                    10     550
```

3.32.4 RESULTANT

User defined material property based on describing the material curve on resultant level (for general description of superelement non-linear behavior, only applicable for beam). Stress/strain convertion factors with respect to tension, torsion and bending parameters are included to increase the flexibility with respect to calculating stresses and strains for arbitrary geometries. The same factors are used for both the stress and the strain cases. This means that different factors must be used depending on whether it is stress or strains that are to be considered. The resultant option includes reference to:

EPCURVE : Elastoplastic force-strain or moment-curvature material behaviour with kinematic/isotropic hardening (when linked to the **RESULTANT** option).

HYCURVE : Hyperelastic (nonlinear elastic) force-strain or moment-curvature material behaviour (when linked to the **RESULTANT** option).

The following format applies:

MATERIAL MNAME RESULTANT POISS WEIGHT TELONG TCOND HCAP EMOD BETA FQX FMX FMY FMZ AXMAT TORMAT YBENMAT ZBENMAT

where:

POISS: Poisson ratio, unit -.

WEIGHT: Weight, unit: kg/mm.

TELONG: Temperature elongation coefficient, unit: -. Dummy unless **TLOAD** is given.

TCOND: Thermal conductivity, unit: W/mK. Dummy.

HCAP: Heat capacity, unit: J/kgK. Dummy.

EMOD: E-modulus, unit: MPa. Dummy.

BETA: Torque/tension coupling parameter, unit: mm.

FQX: Axial stress or strain factor f_{qx}. Axial stress = Axial force × **FQX**, unit: $1/\text{mm}^2$. (Alternatively the factor is used for axial strain: Axial strain$_i$ = Axial strain$_{total}$ × **FQX**, unit: -).

FMX: Torsion stress or strain factor f_{mx}. Shear stress = Torque × **FMX**, unit: $1/\text{mm}^3$. (Alternatively: Shear strain = Torsion × **FMX**, unit: mm).

FMY: Y-moment stress or strain factor f_{my}. Bending stress = Moment × **FMY**, unit: $1/\text{mm}^3$. (Alternatively: Bending strain = Curvature × **FMY**, unit: mm.)

FMZ: Z-moment stress or strain factor f_{mz}. Bending stress = Moment × **FMZ**, unit: $1/\text{mm}^3$. (Alternatively: Bending strain = Curvature × **FMZ**, unit: mm.)

Note:

$$\sigma_{xx} = Q_x f_{qx} + M_y f_{my} + M_z f_{mz}$$
$$\sigma_{yz} = M_x f_{mx}$$

$$\varepsilon_{xx_i} = \varepsilon_{xx_{total}} f_{qx} + \kappa_y f_{my} + \kappa_z f_{mz}$$
$$\varepsilon_{yz} = \kappa_x f_{mx}$$

Note that the factors are used for result generation only. During calculations the specified material curves are applied.

AXMAT: Axial stress resultant material curve, refering to an **EPCURVE** of **HYCURVE** name.

TORMAT: Torsion material curve name, refering to an **EPCURVE** of **HYCURVE** name.

YBENMAT: Bending about local y-axis material curve name, refering to an **EPCURVE** of **HYCURVE** name.

ZBENMAT: Bending about local z-axis material curve name, refering to an **EPCURVE** of **HYCURVE** name.

EXAMPLE:

```
#        Mname    Mtype     Poiss Weight Telong Tcond Hcap Emod    beta
MATERIAL PLASTIC1 RESULTANT 0.30  950e-9 1e-4   0.2   1000 5.5e+01 0
#
# fqx      fmx     fmy     fmz     Axmat   Tormat  Ybenmat Zbenmat
  1.039e-3 1.18e-4 2.37e-4 2.37e-4 SPRING1 SPRING2 SPRING3 SPRING3
#
#
#        Mname    Mtype    Hpar   Disp        Val
MATERIAL SPRING1  EPCURVE  1.0    0             0
                                  0.001        52.9162
                                  0.01        529.162
```

```
#
MATERIAL SPRING2  HYCURVE       -10.0   -31164000
                                 10.0    31164000
#
MATERIAL SPRING3  HYCURVE       -10.0   -40513000
                                 10.0    40513000
```

3.32.5 EPCURVE

Elastoplastic shear stress - relative displacement, force-strain or moment-curvature behaviour with kinematic/isotropic hardening, can either be used to describe resultant curves or user defined shear stress curves for contact interfaces. Note that the surface stiffness for contact interfaces must be given in MPa per mm, which means: Force per mm helix in length direction per mm displacement and per mm circumferential width of the master contact surface.

The following format applies:

```
MATERIAL MNAME EPCURVE HPAR DISP VAL
                                ...   ...
```

where:

HPAR: Hardening parameter Values: from 0 to 1, where

- **0 :** Isotropic hardening
- **1 :** Kinematic hardening

DISP: Displacement. Strain, torsion or curvature. Unit: mm, -, rad/mm or 1/mm.

VAL: Shear stress or stress resultant value. Unit: MPa/mm, N or Nmm.

The pair **DISP** and **VAL** can be repeated as many times as required.

EXAMPLE:

```
#        Mname    Mtype    Hpar   Disp        Val
MATERIAL SPRING1  EPCURVE  1.0    0             0
                                  0.001        52.9162
                                  0.01        529.162
#
```

3.32.6 HYCURVE

Hyperelastic (nonlinear elastic) shear stress - relative displacement, force-strain or moment-curvature material behaviour, can either be used to describe resultant curves or user define shear stress curves for contact interfaces. Note that the surface stiffness for contact interfaces must be given in MPa per mm, which means: Force per mm helix in length direction per mm displacement and per mm circumferential width of the master contact surface.

Note that the curve is *only* valid in its range, ie. it will not be extrapolated.

The following format applies:

```
MATERIAL MNAME HYCURVE DISP VAL
                              ...   ...
```

where:

DISP: Displacement. Strain, torsion or curvature. Unit: mm, -, rad/mm or 1/mm.
VAL: Shear stress or stress resultant value. Unit: MPa/mm, N or Nmm.

The pair **DISP** and **VAL** can be repeated as many times as required.

EXAMPLE:

```
#        Mname    Mtype     Disp    Val
MATERIAL SPRING2 HYCURVE  -10.0   -31164000
                           10.0    31164000
```

3.32.7 FRICSPRING

Elastoplastic displacement - unit force material behaviour with kinematic/isotropic hardening, see Fig. 3.6. Used to describe friction stick-slip behaviour. The unit force obtained from the **FRICSPRING** curve is multiplied with (contact force)×(friction coefficient).

The following format applies:

```
MATERIAL MNAME FRICSPRING HPAR DISP VAL
                                    ...   ...
```

where:

HPAR: Hardening parameter. Can have value 1 or -1. Both gives kinematic hardening, and -1 forces the given material curve to be applied. With value 1, UFLEX2D estimates the slip level based on the average tension in the system, given in the **CONTROL** card, see Section 3.8. The second **DISP** parameter will in this case be dummy.

DISP: Relative displacement, unit: mm.

VAL: Unit shear stress value, unit: -. The unit value will be multiplied with friction coefficient and contact pressure.

The pair **DISP** and **VAL** can be repeated as many times as required.

Note: The **DISP-VAL** curve must have at least three points, and the second of the displacement values will be overwritten by the program to obtain correct level for stick-/slip behavior. However, the user can force the given curve to be applied by entering negative sign for the **HPAR**. It is important to ensure that the last section of the curve is nearly flat to obtain Coulomb friction.

Note: For different contact surfaces, the shape of the curve will in general be different. This implies that a different material name should be used for each contact surface if detailed modelling of friction is required. See **CONTINT**, Section 3.6. Using the same material name for several contact surfaces when the curve is automatically determined by UFLEX2D will result in conservative shear interaction characteristics.

EXAMPLE:

```
#        Mname  Mtype       Hpar  Disp    Val
MATERIAL FRIC1  FRICSPRING  1.0    0      0
                                    0.05   1.0
                                    0.2    1.0
                                   10.0    1.0
```

3.32.8 SHEARSPRING

Hyperelastic displacement - shear stress (per unit length) material behaviour. Used to describe elastic contact behaviour. The unit force is multiplied with (contact force)×(friction coefficient).

The following format applies:

```
MATERIAL MNAME SHEARSPRING DISP VAL
                                ...   ...
```

where:

DISP: Relative displacement, unit: mm.

VAL: Unit shear stress value, unit: -. The unit value will be multiplied with friction coefficient and contact pressure.

The pair **DISP** and **VAL** can be repeated as many times as required.

EXAMPLE:

```
#        Mname  Mtype         Disp  Val
MATERIAL SHEAR  SHEARSPRING   0      0
                              0.05   1.0
                              0.2    1.0
                             10.0    1.0
```

3.32.9 LINEAR

Linear material properties (applicable for conshell, beam, beamshell and shell).

The following format applies:

MATERIAL MNAME LINEAR POISS DENS TELONG TCOND HCAP EMOD BETA CODE EA EIY EIZ GIT WEIGHT

where

POISS: Poissons ratio, unit: -.

DENS: Density, unit: kg/mm^3.

TELONG: Temperature elongation coefficient, unit: -. Dummy unless **TLOAD** is given.

TCOND: Thermal conductivity, unit W/mK. Dummy.

HCAP: Heat capacity, unit: J/kgK. Dummy.

EMOD: Young's modulus, unit: MPa.

BETA: Torque/tension coupling factor, unit: mm.

CODE: Material code. Values:

MANUAL : The following data are user defined

AUTO : The following data is determined by the program

EA: Axial stiffness, unit: N.

EIY: Bending stiffness about tangential axis (local y-axis), unit: Nmm^2.

EIZ: Bending stiffness about surface normal axis (local z-axis), unit: Nmm^2.

GIT: Torsion stiffness, unit: Nmm^2.

WEIGHT: Weight per mm, unit: kg/mm.

EXAMPLE:

```
#        Mname Mtype  Poiss Dens Telong   Tcond Hcap Emod beta Code
MATERIAL CONT1 LINEAR 0.3   7800 1.00E-05 50    500  1E5  0    MANUAL
#
#        EA       Eiy     Eiz   Git      Weight
         1E6      2E6     2E6   2E6      10
```

3.32.10 CONTENT

Content, only weight and insulation. Used to describe content of tubes which may be filled or not filled with content.

The following format applies:

MATERIAL MNAME CONTENT DENS TELONG TCOND HCAP CODE WEIGHT

where:

DENS: Density, unit: kg/mm^3.

TELONG: Temperature elongation coefficient, unit: -. Dummy unless **TLOAD** is given.

TCOND: Thermal conductivity, unit W/mK. Dummy.

HCAP: Heat capacity, unit: J/kgK. Dummy.

CODE: Material code. Values:

MANUAL : The following data are user defined

AUTO : The following data is determined by the program

WEIGHT: Weight per mm, unit: kg/mm.

EXAMPLE:

```
#        Mname Mtype   Dens     Telong    Tcond Hcap Code
MATERIAL VOID1 CONTENT 1025E-9 1.00E-04  0.2   1000 AUTO
```

3.32.11 FIXCONTENT

Fixed content, only weight. Used to describe content of voids which always will be filled with content.

The following format applies:

MATERIAL MNAME FIXCONTENT DENS TELONG TCOND HCAP CODE WEIGHT

where:

DENS: Density, unit: kg/mm^3.

TELONG: Temperature elongation coefficient, unit: -. Dummy unless **TLOAD** is given.

TCOND: Thermal conductivity, unit W/mK. Dummy.

HCAP: Heat capacity, unit: J/kgK. Dummy.

CODE: Material code. Values:

MANUAL : The following data are user defined

AUTO : The following data is determined by the program

WEIGHT: Weight per mm, unit: kg/mm.

EXAMPLE:

```
#        Mname  Mtype       Dens     Telong    Tcond Hcap Code
MATERIAL VOID2  FIXCONTENT  1055E-9 1.00E-04  0.2   1000 AUTO
```

3.32.12 ANISOTR

Linear anisotropic material is applicable only for SHELL, in case hoop and axial directions have different mechanical properties. This is the case for composite materials. In this section hoop direction is low-indexed as 1, and axial as 2. The following format applies:

MATERIAL MNAME ANISOTR POISS DENS TELONG TCOND HCAP EMOD1 BETA CODE GM EMOD2 EA EIY EIZ GIT WEIGHT

where

POISS: Poisson ratio ν_{12} in hoop-axial direction, unit: -.

DENS: Density, unit: kg/mm^3.

TELONG: Temperature elongation coefficient, unit: -. Dummy unless **TLOAD** is given.

TCOND: Thermal conductivity, unit W/mK. Dummy.

HCAP: Heat capacity, unit: J/kgK. Dummy.

EMOD1: Young's modulus E_1 in hoop direction, unit: MPa.

BETA: Torque/tension coupling factor, unit: mm.

CODE: Material code. Values:

MANUAL : The last five parameters are user defined

AUTO : The last five parameters are determined by the program

GM: Shear modulus GM_{12}, unit: MPa.

EMOD2: Young's modulus E_2 in axial direction, unit: MPa.

EA: Axial stiffness, unit: N.

EIY: Bending stiffness about tangential axis (local y-axis), unit: Nmm^2.

EIZ: Bending stiffness about surface normal axis (local z-axis), unit: Nmm^2.

GIT: Torsion stiffness, unit: Nmm^2.

WEIGHT: Weight per mm, unit: kg/mm.

Note that Poisson ratio ν_{21} in axial-radial direction does not need to be specified, as it is calculated from stiffness matrix symmetry property $E_1\nu_{21} = E_2\nu_{12}$.

EXAMPLE:

```
#        Mname Mtype  Poiss Dens Telong   Tcond Hcap Emod1 beta Code GM  Emod2
MATERIAL CONT1 ANISOTR 0.3   7800 1.00E-05 50    500  1E5   0    AUTO 8E4 2E5
```

3.33 PRLOAD - PRessure LOAD

Hydrostatic pressure can be applied on any interface except for beamshells. The **PRLOAD** data group is defined by:

PRLOAD GEONAME INTERFACE PRES HIST

where:

GEONAME: Geometry name.

INTERFACE: Interface number. Can have two values:

1 : External pressure (applied on the external surface)

2 : Internal pressure (applied on the internal surface)

PRES: Pressure, unit: MPa.

HIST: The load history number (may be an arbitrary number as long as the same number is found in the **HIST** data group) See Section 3.26.

Note: Only one single history is permitted for presure loading on one geometry. This means that identical history should be selected if applying bpth internal and external pressure load. If different histories are specified, UFLEX2D will apply the history belonging to the last **PRLOAD** card.

EXAMPLE:

```
#      Geoname Interface Pres Hist
PRLOAD PIPE1   2         20   200
```

3.34 ROTATE

By the card **ROTATE**, the user can rotate the complete cross section. This implies that the center objects, and the second order processes will be rotated.

ROTATE ANGLE

The only parameter required is:

ANGLE: Angle of rotation.

The card can be given several times. When issued several times the rotation will be the sum of the given rotations.

3.35 SMERGE - first order Shell MERGE definition

For shells of first order, it is in some cases convenient to suppress local deformation modes that may cause numerical instabilities. This is obtained by the shell merge command.

In UFLEX2D this is done by letting the first node of the geometry act as a master node for the others, i.e all nodes will have the same radial and longitudinal displacement. The user does not need to concider which node is master/slaves.

SMERGE GEONAME

where:

GEONAME: Name of geometry to be merged.

The effect of merging a first order shell layer is visualized in Fig. 3.7. The radial translation dof of the first shell node will act as a master dof for the corresponding dof of the other nodes, i.e. all nodes will have the same radial motion.

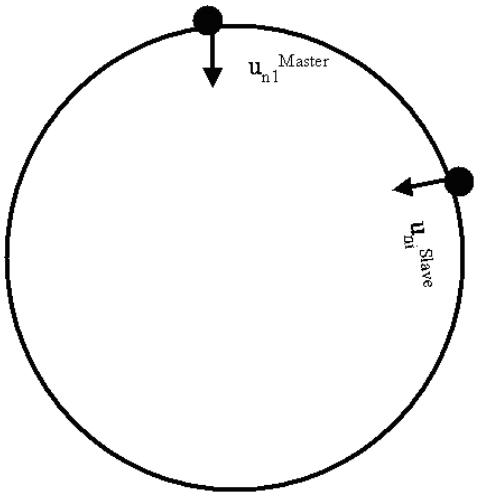


Figure 3.7: Merging a first order shell layer.

EXAMPLE:

```
#       Geoname
SMERGE SHEATH
```

3.36 THICKWALL - THICKWALL stresses

By entering the **THICKWALL** card, the user can define that certain geometries should be displayed with stresses according to thickwall shell theory. The end-cap effect will then conside inner and outer diameters, as apposed to mean diameter as used in thin wall theory. Hoop stresses will be displayed according to thickwall shell theory, and radial stress is available as an extra type of of result. The radial stresses are taken into account for the Von Mises stress, according to the following equation:

$$\sigma_{vm} = \sqrt{\sigma_{xx}^2 + \sigma_{yy}^2 + \sigma_{zz}^2 - \sigma_{xx}\sigma_{yy} - \sigma_{xx}\sigma_{zz} - \sigma_{yy}\sigma_{zz} + 3\sigma_{xy}^2}$$

where σ_{xx} is longitudinal tubes stresses, σ_{yy} is hoop stress and σ_{zz} is the Lamé radial stresses due to pressure.

THICKWALL GEONAME1 [GEONAME2 ...]

where:

GEONAME: Name of geometry.

An arbitrary number of geometry names can be listed after a **THICKWALL** key, and several **THICKWALL** cards can be used.

Note: Applying thickwall for geometries with element type other than shell has no effect. Moreover, the material type for the geometry must be **LINEAR**, otherwise the command will be ignored. The thickwall option affects only the end-cap effect, and the display of radial/hoop stress results.

EXAMPLE:

```
#         Geoname list
THICKWALL CPIPE58 1PIPE58 2PIPE58
```

3.37 TIME - Time card

The **TIME** card is to be specified only once in the input file, and only in case bitumen material is used. It specifies time for all bitumen coatings.

TIME is defined as follows:

TIME TOTALTIME HISTNO

The parameters are defined as:

TOTALTIME: Total time, sec.
HISTNO: Time history ID.

EXAMPLE:

```
#     TotalTime Histno
TIME    6.28     100
```

3.38 TLOAD - Temperature LOAD

The data group is defined by:

TLOAD GEOM VAL HIST

where:

GEOM: Geometry where temperature load will be applied.

VAL: Value of the temperature, unit: degrees Celsius.

HIST: The load history number (may be an arbitrary number as long as the same number is found in the **HIST** data group) See Section 3.26. Material elongation parameter **TELONG** found in **MATERIAL** cards will be activated if **TLOAD** is used.

EXAMPLE:

```
#       Geom  Val     Hist
TLOAD   PIPE1  90     100
```

3.39 TORSCENTRE

By the card **TORSCENTRE** user can specify shift of the torsion centre. The default torsion centre (0,0) refers to the centre of the first order geometry, if such present. Otherwise it is the centre of the any second order geometry group. **TORSCENTRE** enables modelling asymmetric torsion.

TORSCENTRE YCOR ZCOR

The parameters required are:

YCOR: y-coordinate shift (horizontal).

ZCOR: z-coordinate shift (vertical).

Here is an example

EXAMPLE:

```
#            ycor  zcor
TORSCENTRE   0     5
```

3.40 TUGEOM

This card is not be used together with FRICMODEL 1.

By the card **TUGEOM**, the user can specify which geometry will be the basis for tuning the friction springs. If not given, a the most conservative choice will be made. The card

will only affect friction spring tuning specified at a specified step, see Section 3.8 and Section 3.6.

TUGEOM GEOM

The only parameter required is:

GEOM: Geometry name.

3.41 VISRES - VISual RESults

By including the **VISRES** card, the user can select result types to be stored to .raf file. These result will be available in XPOST, and for post processing through UPOST2D. If no **VISRES** card is given, result type sigma-xx will be stored as default.

The format is as follows:

VISRES RESULT

where

RESULT: Result types that will be stored, can be a list. Options are:

Result type:		Valid for:
SIGMA-XX	σ_{xx} stress. Unit: MPa.	Shell, beam, beamshell
SIGMA-YY	σ_{yy} stress. Unit: MPa.	Shell, beamshell
SIGMA-ZZ	σ_{zz} stress. Unit: MPa.	Beamshell
SIGMA-XY	σ_{xy} stress. Unit: MPa.	Shell, beamshell
SIGMA-XZ	σ_{xz} stress. Unit: MPa.	Beamshell
SIGMA-YZ	σ_{yz} stress. Unit: MPa.	Beam, beamshell
SIGMA-VM	Von Mises stress.	Shell, beam, beamshell
SIGMA-CONTACT	Contact stresses.	Shell, beam, beamshell
SIGMA-RADIAL	Radial stress.	Shell with **THICKWALL**, see Section 3.36.
CURV-SLIP	Slip curvature.	Shell, beam, beamshell
STRA-XX	ε_{xx} strain. Unit: -.	Shell, beam, beamshell
STRA-YY	ε_{yy} strain. Unit: -.	Shell, beamshell
STRA-ZZ	ε_{zz} strain. Unit: -.	Beamshell
STRA-XY	ε_{xy} strain. Unit: -.	Shell, beamshell
STRA-XZ	ε_{xz} strain. Unit: -.	Beamshell
STRA-YZ	ε_{yz} strain. Unit: -.	Beam, beamshell

EXAMPLE:

```
#         result list
VISRES    sigma-xx sigma-yy
```

Note: If result type **SIGMA-VM** is requested, all the σ_{ij} stresses will also be stored, as they anyway are required for calculation of **SIGMA-VM**.

Note: Contact stresses are calculated in the contact element system. To integrate contact stresses into forces, the area of the contact element must be considered. The area of the contact element is based on its master geometry, using the masters segment length at the contact $\times$ the model thickess of 1 mm. If the master geometry is of type shell, the segment length is measured at the structural radius of the shell.

Some of the result types are only relevant for particular element types. Also, as the local element systems vary depending on type of element, the interpretation of the different strains/stresses vary depending on element type.

BEAMSH elements:
For components of type beamshell, the stresses and strain follows normal conventions, applying a local system for the component as described in Section 2.8.

BEAM elements:
The relevant stress results are **SIGMA-XX**, **SIGMA-YZ**, **SIGMA-VM** and **SIGMA-CONTACT** which are the longitudinal stresses, shear stresses for hoop/radial direction, the Von Mises stresses and the contact stresses. The Von Mises stresses for beam take into account **SIGMA-XX** and **SIGMA-YZ**.

SHELL elements:
The relevant stress results are **SIGMA-XX**, **SIGMA-YY**, **SIGMA-XY**, **SIGMA-VM** and **SIGMA-CONTACT** which are the tube longitudinal stresses, hoop stresses, shear stresses for hoop/longitudinal direction, Von Mises stresses and contact stresses. If **THICK-WALL** is applied the radial stresses due to inner/outer pressure,**SIGMA-RADIAL** is calculated, and **SIGMA-YY** is adjusted for thickwall. Von Mises stresses take into account **SIGMA-XX**, **SIGMA-YY**, **SIGMA-RADIAL** (if thickwall) and **SIGMA-XY**.

3.42 VOID - Void definition

The geometry of voids is uniquely defined by the already defined geometries and can be considered as one external interface geometry minus a sum of internal geometries. The external geometry must be a shell, and the internal geometries can be of type shell or beam.

The **VOID** data group is defined as follows:

```
VOID EXTGEONAME EXTINTERF MATNAME INTGEONAME INTINTERF
                                        ... ...
```

where:

EXTGEONAME: Name of external surface geometry.

EXTINTERF: The interface number for the external surface geometry.

- **1 :** Outside
- **2 :** Inside

MATNAME: The name of the material property data used to fill the voids (either **CONTENT** or **FIXCONTENT**, see **MATERIAL** data group), Section 3.32.

INTGEONAME: Name of internal surface geometry.

INTINTERF: The interface number of internal surface geeometry.

- **1 :** Outside
- **2 :** Inside

Pairs of **INTGEONAME** and **INTINERF** can be repeated as many times as required.

4 UPOST2D Report Generator

4.1 General

UPOST2D includes the following files:

prefix.2pi the UPOST2D Input file.

prefix.2pl the UPOST2D Log file where warnings and error messages are written.

In addition references are given in the input file to the relevant .raf file and the associated output .mpf files. All output is given on MatrixPlot (.mpf) format, see (Giertsen, 1998).

4.2 Input data

All input is described on ASCII file format. The maximum number of letters in one single text string is 128.

Comment text strings are defined by introducing # at the start of the line.

The following identifiers define the different data groups:

GLPLOT for GLobal history PLOTs, see Section 4.3.

NOPLOT for NOdal history PLOTs, see Section 4.4.

ELPLOT for ELement history PLOTs, see Section 4.5.

FATIGUE for FATIGUE data, see Section 4.6.

CAPACIT for CAPACIT data, see Section 4.7.

A result which exists but is not valid is set to zero in the mpf-output file. The corresponding legend will be marked with a d for dummy.

4.3 GLPLOT - GLobal history PLOTs

The format is as follows:

GLPLOT RAF MPF XLEG XRES YLEG YRES XFAC YFAC [DIGIT]

where:

RAF: The UFLEX2D *.raf file name prefix. If the file name is defined by a combination of lower case and upper case letters, use ": e.g. "input-filename"

MPF: The output *.mpf file name prefix. If the file name is defined by a combination of lower case and upper case letters, use ": e.g. "mpf-filename"

XLEG: The legend name for the x-axis. If the legend name contains more than one word or if the name is defined by a combination of lower case and upper case letters, use ": e.g. "Displacement (m)"

XRES: x-axis result type. The following result types are available:

LOADSTEP : Load step.

AXSTRAIN : Global axial strain, unit: -.

TORSION : Global torsion, unit: 1/mm.

CURV-Y : Global curvature about y-axis, unit: 1/mm.

CURV-Z : Global curvature about z-axis, unit: 1/mm.

AXFORCE : Global axial force, unit: N.

TORSMOM : Global torsion moment, unit: Nmm.

BENDM-Y : Global bending moment about y-axis, unit: Nmm.

BENDM-Z : Global bending moment about z-axis, unit: Nmm.

HISTN : History number with reference to UFLEX2D input file, e.g HIST100, see Section 3.26.

YLEG: The legend name for the y-axis. If the legend name contains more than one word or if the name is defined by a combination of lower case and upper case letters, use ": e.g "Displacement (m)".

YRES: y-axis result type. The following result types are available (same as for **XRES**):

LOADSTEP : Load step.

AXSTRAIN : Global axial strain, unit: -.

TORSION : Global torsion, unit: 1/mm.

CURV-Y : Global curvature about y-axis, unit: 1/mm.

CURV-Z : Global curvature about z-axis, unit: 1/mm.

AXFORCE : Global axial force, unit: N.

TORSMOM : Global torsion moment, unit: Nmm.

BENDM-Y : Global bending moment about y-axis, unit: Nmm.

BENDM-Z : Global bending moment about z-axis, unit: Nmm.

HISTN : History number with reference to UFLEX2D input file, e.g HIST100, see Section 3.26.

XFAC: Scaling factor to be used for x-axis (to convert to convenient unit)

YFAC: Scaling factor to be used for y-axis (to convert to convenient unit)

DIGIT: Optional parameter used to define number of significant digits written to the mpf-file. must be given on the format "digit=N", where N is a number between 1 and 9. Default value is 3 digits.

4.4 NOPLOT - NOdal history PLOTs

The format is as follows:

NOPLOT RAF MPF XLEG XRES YLEG YRES GEOM POS XFAC YFAC [DIGIT]

where:

RAF: The UFLEX2D *.raf file name prefix. If the file name is defined by a combination of lower case and upper case letters, use ": e.g. "input-filename"

MPF: The output *.mpf file name prefix. If the file name is defined by a combination of lower case and upper case letters, use ": e.g. "mpf-filename"

XLEG: The legend name for the x-axis. If the legend name contains more than one word or if the name is defined by a combination of lower case and upper case letters, use ": e.g. "Displacement (m)"

XRES: x-axis result type. The following result types are available:

LOADSTEP : Load step.

AXSTRAIN : Global axial strain, unit: -.

TORSION : Global torsion, unit: 1/mm.

CURV-Y : Global curvature about y-axis, unit: 1/mm.

CURV-Z : Global curvature about z-axis, unit: 1/mm.

AXFORCE : Global axial force, unit: N.

TORSMOM : Global torsion moment, unit: Nmm.

BENDM-Y : Global bending moment about y-axis, unit: Nmm.

BENDM-Z : Global bending moment about z-axis, unit: Nmm.

HISTN : History number with reference to UFLEX2D input file, e.g HIST100, see Section 3.26.

YLEG: The legend name for the y-axis. If the legend name contains more than one word or if the name is defined by a combination of lower case and upper case letters, use ": e.g "Displacement (m)".

YRES: y-axis result type. The following result types are available:

NODISP-X : x-displacement, unit: mm.

NODISP-Y : y-displacement, unit: mm.

NODISP-Z : z-displacement, unit: mm.

NOPOS-X : x-coordinate, unit: mm.

NOPOS-Y : y-coordinate, unit: mm.

NOPOS-Z : z-coordinate, unit: mm.

In addition, results stored to .raf file by the **VISRES** command, see Section 3.41, are available.

GEOM: The geometry for which results are to be generated, refer to .raf file or .2if input file.

POS: Position filter (among nodes). The following positions are available:

LEFT : Position having minimum local y-coordinate value.

MID : Position having mean local y-coordinate value.

RIGHT : Position having maximum local y-coordinate value.

UPPER : Position having maximum local z-coordinate value.

LOWER : Position having minimum local z-coordinate value.

MIDHEIGHT : Position having mean local z-coordinate value.

OUTER : All positions on outer surface.

INNER : All positions on inner surface.

ALL : All positions.

XFAC: Scaling factor to be used for x-axis (to convert to convenient unit)

YFAC: Scaling factor to be used for y-axis (to convert to convenient unit)

DIGIT: Optional parameter used to define number of significant digits written to the mpf-file. must be given on the format "digit=N", where N is a number between 1 and 9. Default value is 3 digits.

4.5 ELPLOT - ELement history PLOTs

The format is as follows:

ELPLOT RAF MPF XLEG XRES YLEG YRES GEOM POS XFAC YFAC [DIGIT]

where:

RAF: The UFLEX2D *.raf file name prefix. If the file name is defined by a combination of lower case and upper case letters, use ": e.g. "input-filename"

MPF: The output *.mpf file name prefix. If the file name is defined by a combination of lower case and upper case letters, use ": e.g. "mpf-filename"

XLEG: The legend name for the x-axis. If the legend name contains more than one word or if the name is defined by a combination of lower case and upper case letters, use ": e.g. "Displacement (m)"

XRES: x-axis result type. The following result types are available:

LOADSTEP : Load step.

AXSTRAIN : Global axial strain, unit: -.

TORSION : Global torsion, unit: 1/mm.

CURV-Y : Global curvature about y-axis, unit: 1/mm.

CURV-Z : Global curvature about z-axis, unit: 1/mm.

AXFORCE : Global axial force, unit: N.

TORSMOM : Global torsion moment, unit: Nmm.

BENDM-Y : Global bending moment about y-axis, unit: Nmm.

BENDM-Z : Global bending moment about z-axis, unit: Nmm.

HISTN : History number with reference to UFLEX2D input file, e.g HIST100, see Section 3.26.

YLEG: The legend name for the y-axis. If the legend name contains more than one word or if the name is defined by a combination of lower case and upper case letters, use ": e.g "Displacement (m)".

YRES: y-axis result type. The following result types are available:

ELFORCE-X : Fx force, unit: N.

ELFORCE-Y : Fy force, unit: N.

ELFORCE-Z : Fz force, unit: N.

ELFSHMO-X : Local shell Mx moment, unit: Nmm. (Dummy for beam)

ELFSHMO-Y : Local shell My moment, unit: Nmm. (Dummy for beam)

ELFSHMO-Z : Local shell Mz moment, unit: Nmm. (Dummy for beam)

ELFMO-X : Mx moment about centroid x-axis, unit: Nmm.

ELFMO-Y : My moment about centroid y-axis, unit: Nmm.

ELFMO-Z : Mz moment about centroid z-axis, unit: Nmm.

ALLELFORC : All stress resultant components, 6 for beam, 9 for shell. Unit: N and Nmm.

ELDISP-X : x displacement, unit: mm.

ELDISP-Y : y displacement, unit: mm.

ELDISP-Z : z displacement, unit: mm.

ELDROT-X : Local shell x-rotation, unit: rad. (Dummy for beam)

ELDROT-Y : Local shell y-rotation, unit: rad. (Dummy for beam)

ELDROT-Z : Local shell z-rotation, unit: rad. (Dummy for beam)

ELDTOR-X : Torsion about centroid, unit: 1/mm.

ELDCUR-Y : y-curvature about centroid, unit: 1/mm.

ELDCUR-Z : z-curvature about centroid, unit: 1/mm.

ALLEDISP : All element displacement quantities, unit: mm, rad or 1/mm.

CONDIS-X : Contact element displacement in local x-direction, unit: mm. (For this type **POS** parameter is dummy.)

CONDIS-Y : Contact element displacement in local y-direction, unit: mm. (For this type **POS** parameter is dummy.)

CONDIS-Z : Contact element displacement in local z-direction, unit: mm. (For this type **POS** parameter is dummy.)

CONFOR-X : Contact element force in local x-direction, unit: N. (For this type **POS** parameter is dummy.)

CONFOR-Y : Contact element force in local y-direction, unit: N. (For this type **POS** parameter is dummy.)

CONFOR-Z : Contact element force in local z-direction, unit: N. (For this type **POS** parameter is dummy.)

GEOM: The geometry for which results are to be generated, refer .raf file.

POS: Position filter (among nodes). The following positions are available:

LEFT : Position having minimum local y-coordinate value.

MID : Position having mean local y-coordinate value.

RIGHT : Position having maximum local y-coordinate value.

UPPER : Position having maximum local z-coordinate value.

LOWER : Position having minimum local z-coordinate value.

MIDHEIGHT : Position having mean local z-coordinate value.

XFAC: Scaling factor to be used for x-axis (to convert to convenient unit)

YFAC: Scaling factor to be used for y-axis (to convert to convenient unit)

DIGIT: Optional parameter used to define number of significant digits written to the mpf-file. must be given on the format "digit=N", where N is a number between 1 and 9. Default value is 3 digits.

4.6 FATIGUE - FATIGUE data

In order to extract key data for fatigue calculations, certain conditions must be met:

- The **CURV-SLIP** result must be stored to the .raf file using the **VISRES** command, see Section 3.41.
- The load history must consist of two segments, axial loading followed by curvature. Both should increase monotonously. The axial load should be held constant during bending.
- The resolution of the curvature history should be sufficiently large to obtain a minimum of two data points with finite curvature prior to slip; also the curvature

should extend sufficiently to allow a minimum of two data points subsequent to slip.

The format is as follows:

```
FATIGUE RAF GEOM POS [DIGIT]
```

where:

RAF: The UFLEX2D *.raf file name prefix. If the file name is defined by a combination of lower case and upper case letters, use ": e.g. "input-filename"

GEOM: The geometry for which results are to be generated, refer to .raf file or .2if input file.

POS: Position filter (among nodes). The following positions are available:

LEFT : Position having minimum local y-coordinate value.

MID : Position having mean local y-coordinate value.

RIGHT : Position having maximum local y-coordinate value.

UPPER : Position having maximum local z-coordinate value.

LOWER : Position having minimum local z-coordinate value.

MIDHEIGHT : Position having mean local z-coordinate value.

OUTER : All positions on outer surface.

INNER : All positions on inner surface.

ALL : All positions.

DIGIT: Optional parameter used to define number of significant digits written to the output file. must be given on the format "digit=N", where N is a number between 1 and 9. Default value is 3 digits.

4.7 CAPACIT - CAPACITY data

In order to extract key data for capacity calculations, certain conditions must be met:

- The load history must consist of two segments, a segment of no curvature followed by a segment of only curvature. The latter should increase monotonously.
- The curvature should extend sufficiently to exceed the material capacity of the selected geometry during, but not prior to curvature. If this condition is not met, the reported capacity curvature will be either zero or infinity.

The format is as follows:

```
CAPACIT RAF STRESS-COMPONENT GEOM YIELD-STRESS [DIGIT]
```

where:

RAF: The UFLEX2D *.raf file name prefix. If the file name is defined by a combination of lower case and upper case letters, use ": e.g. "input-filename"

STRESS-COMPONENT: The stress-component relevant for the provided yield stress.

GEOM: The geometry for which results are to be generated, refer to .raf file or .2if input file.

YIELD-STRESS: The specified minimum yield stress (SMYS).

DIGIT: Optional parameter used to define number of significant digits written to the output file. must be given on the format "digit=N", where N is a number between 1 and 9. Default value is 3 digits.

Bibliography

Contract between marintek and nexans: Oc-=238823-kpi13, attachement 1, forslag til filformat.

Giertsen, E. (October 1998). Matrixplot users guide, marintek memo.

Mroz, Z. (1973). *Mathematical Models of Inelastic Material Behavior*. Waterloo : University of Waterloo.

Sævik, S. (2004a). UFLEX - *Test Manual*. MARINTEK, Trondheim, Norway, MT70-f02-027 edition.

Sævik, S. (December 2004b). Uflex - theory manual, marintek report no MT70 f02-018.

Index

R

S

T

U

V