

Giraffe Tutorials Guide



**Generic Interface Readily Accessible for Finite
Elements**

Developed at University of São Paulo, Brazil

November 2019.

Prof. Alfredo Gay Neto

Table of Contents

Introduction	3
Tutorial 1. Cantilever beam statics	4
Tutorial 2. Modal analysis of a circular plate	12
Tutorial 3. Buckling of a pipe triggered by a compression follower load	20
Tutorial 4. Buckling of a pipe triggered by internal pressure	29
Tutorial 5. Static and Dynamic analysis of a cantilever plate	38
Tutorial 6. Static and Dynamic analysis of a cantilever plate with a rigid edge.....	48
Tutorial 7. Using several solution steps in a cantilever beam model	58
Tutorial 8. Evaluation of system modal analysis during a static simulation	67
Tutorial 9. Evaluation of a system modal analysis during buckling of a cantilever beam	73
Tutorial 10. Transient dynamics of a beam	80
Tutorial 11. Two degrees of freedom lumped system	85
Tutorial 12. Simplified bogie dynamics.....	93
Tutorial 13. Baseball bat hitting the ball	106
Tutorial 14. Contact between cantilever beams	118
Tutorial 15. Lateral buckling of a pipeline laid on a flat rigid surface	129
Tutorial 16. Suspension of a catenary mooring line	143
Acknowledgements.....	158
References.....	158

Introduction

This document contains a series of tutorials to be solved using Giraffe 2.0. It is organized in increasing complexity level, until tutorial 10. The remaining tutorials are devoted to specific problems or modeling techniques. The user is invited to follow all tutorials, in their presenting sequence. At the end, an overview of Giraffe resources will have been covered.

The Giraffe users' manual should be used together with this tutorials guide, in order to complement the understanding on each keyword employed to construct the numerical models. The user is also invited to use his/her creativity to create variations of boundary conditions, loads and change the characteristics of each tutorial model. This will provide a deeper learning than simply reproducing and solving each model here presented. At the end of some tutorials there is a list of suggested changes to be explored in a section named "What to try now?".

Giraffe input is a text file that may be edited in any text editor, such as Notepad, jEdit or Notepad++. The extension of the input file must be ".inp". In each tutorial, after preparing and reviewing your input file, Giraffe has to be launched to solve the simulation. To install Giraffe in your computer and to solve a simulation, please follow the instructions of Giraffe users' manual.

Each tutorial has some comments about post-processing techniques, employing Paraview™ software. We encourage you also to try using new resources, exploring Paraview™ software.

São Paulo, 2019.

Alfredo Gay Neto

Tutorial 1. Cantilever beam statics

Input file name: tutorial01.inp

Developed by Alfredo Gay Neto

Objective

This tutorial presents a geometrically-nonlinear simulation of a cantilever beam. It aims at showing how to apply nodal loads and nodal boundary conditions using Giraffe to solve a simple static geometrically-nonlinear analysis.

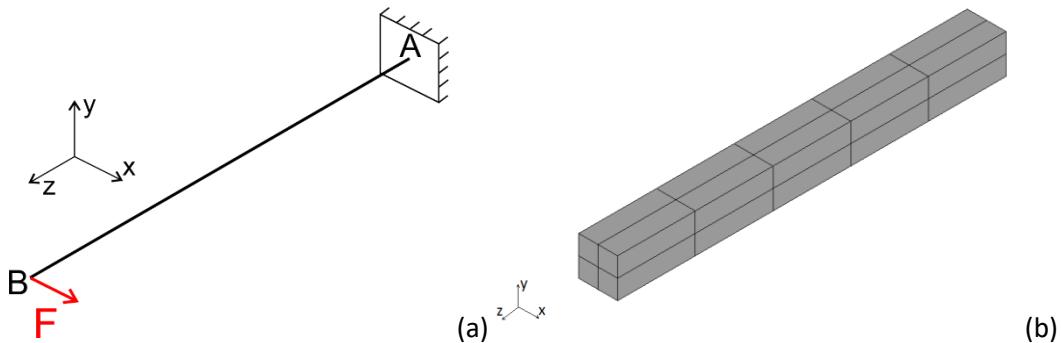


Figure 1-1 – (a) geometric description (b) 3D rendering for visualization purposes

Description

The beam AB (1.0 m length) is meshed using 11 nodes and 5 elements, using the element Beam_1. There is a clamp boundary condition at point A (origin of the global coordinate system). A force F is applied at point B. The cross section of the beam is square. The beam is initially straight and aligned in direction z.

Input file step-by-step

Creation of nodes

Eleven nodes are created to establish a finite element mesh.

```
//Creation of nodes
Nodes 11
//Number      X      Y      Z
Node   1      0      0      0.0
Node   2      0      0      0.1
Node   3      0      0      0.2
Node   4      0      0      0.3
Node   5      0      0      0.4
Node   6      0      0      0.5
Node   7      0      0      0.6
Node   8      0      0      0.7
Node   9      0      0      0.8
Node  10      0      0      0.9
Node  11      0      0      1.0
```

Creation of node sets

Node sets are created to establish loads and constraints. Node sets may be created by distinct syntaxes. In present case, the more convenient is to directly list the nodes included in each node set.

```
//Creation of node sets
NodeSets      2
//Point A
NodeSet       1      Nodes 1      List  1
//Point B
NodeSet       2      Nodes 1      List  11
```

Creation of elements

Five elements are defined. All the elements are assigned to the same material properties (identification number 1), same cross section (identification number 1) and same coordinate system (identification number 1). Each Beam_1 element is defined by three nodes, which **must** be equally spaced and contained in a straight line.

```
//Creation of elements
Elements      5
Beam_1 1     Mat    1      Sec   1      CS    1      Nodes 1      2      3
Beam_1 2     Mat    1      Sec   1      CS    1      Nodes 3      4      5
Beam_1 3     Mat    1      Sec   1      CS    1      Nodes 5      6      7
Beam_1 4     Mat    1      Sec   1      CS    1      Nodes 7      8      9
Beam_1 5     Mat    1      Sec   1      CS    1      Nodes 9      10     11
```

Creation of materials

One only material model is defined, which identification number is 1.

```
//Creation of materials
Materials     1
Hooke 1      E     1e7      Nu    0.3      Rho   2000
```

Creation of cross-sections

One only cross-section is defined, a rectangle. Both B and H are set to 0.1 m (base and height), to represent a square cross-section. The identification number of the cross-section is 1.

```
//Creation of sections
Sections      1
Rectangle    1      B     0.1      H     0.1
```

Creation of coordinate systems

One only coordinate system is input. This is used to define the element properties. Note that, since employing Beam_1 element, the direction E3 must lie in the beam alignment direction. The identification number of the coordinate system is 1.

Note: In case of creation of not-aligned beam elements (for example, for representing an approximation of a curved beam or a frame), each beam element must be associated with a defined coordinate system, which direction E3 is aligned with the element direction. This is done

in order to permit creation of arbitrary alignment between successive cross sections of a beam-like structure, composed possibly by many cross-sections or alignments of these.

```
//Creation of coordinate systems
CoordinateSystems 1
CS    1      E1    1      0      0      E3    0      0      1
```

Creation of solution steps

This tutorial has a single solution step (static). This is created establishing the end time equal to “1”. This is an arbitrary tracking scalar parameter for statics. Time-steps are established to enforce total load to be divided in increments of 10%. Note that IncFactor 1.0 permits no increasing of time-stepping. The option Sample 2 ensures saving post-processing files for each 2 converged configurations. The larger the Sample set, the smaller number of files for post-processing that Giraffe will save along simulation evolution.

```
//Creation of the solution steps
SolutionSteps1
Static 1
EndTime      1
TimeStep     0.1
MaxTimeStep  0.1
MinTimeStep  0.01
MaxIt       20
MinIt       3
ConvIncrease 4
IncFactor    1.0
Sample 2
```

Creation of loads

A single nodal load is created. This is done establishing a time-varying force in direction X. Giraffe automatically performs linear interpolation for undefined time values following input data. Note that the user may choose local coordinate systems to define vector quantities of loads (forces/momenta). In current case CS 1 was the choice.

```
//Creation of loads
Loads 1
NodalLoad   1      NodeSet   2      CS      1      NTimes 2
//Time FX    FY    FZ      MX      MY      MZ
0           0      0       0       0       0
1           1000  0       0       0       0
```

Creation of constraints

A single nodal constraint is created. This is done fixing DOFs: UX, UY, UZ, ROTX, ROTY and ROTZ. Note that BoolTable resource is employed. In this case the nodal constraint is established for the solution step 1 (the only step in this tutorial). So, just number “1” is entered, as a meaning of a Boolean variable “true” (to activate such constraint in the first solution step).

Note: In case of simulations with multiple solution steps the user may create a sequence of “1” and “0” to activate or deactivate constraints. Each “true” or “false” will refer to corresponding solution steps, in a solution sequence.

```
//Creation of constraints
Constraints 1
NodalConstraint 1      NodeSet      1
    UX   BoolTable  1
    UY   BoolTable  1
    UZ   BoolTable  1
    ROTX BoolTable  1
    ROTY BoolTable  1
    ROTZ BoolTable  1
```

Creation of solver options

Here we request Giraffe to apply a single processor thread to solve this case. If the user sets a larger number of processors, Giraffe will automatically use a shared memory parallel processing based on OpenMP™. The linear system solution method chosen is Direct. Giraffe employs PARDISO™ and MKL™ Intel library to solve linear systems of equations along solution evolution.

```
//Creation of solver options
SolverOptions
Processors 1      LinSys Direct
```

Creation of monitors

We request Giraffe to monitor 2 nodes during the simulation: numbers 1 and 11. This will create automatically a folder of monitors for post-processing data. Monitors are very useful for analyzing particular time-series of nodes, node-sets, contact regions or elements. When the user is interested in analyzing the whole mesh results at a given instant, the more convenient is not using monitors, but looking at solution post-processing files, instead. These are located in “post” folder and are organized according to the sequence of solution steps established for the simulation.

```
//Creation of monitors
Monitor      Sample 1
MonitorNodes 1      11
```

Creation of post-files

We choose WriteMesh, WriteRenderMesh and WriteForces options for post-processing using Paraview™. The MagFactor parameters can be used to generate plots with exaggerated displacement patterns, to help to visualize in case of small displacements. This is not the case of this example. Then, we will assume a unit value, then, requesting that the deformed shape of Paraview™ frames will reflect the real scale. Use the MagFactor resource with care. It may create unrealistic plots and not-straightforward interpretations. If in doubt, the best is to set MagFactor 1.

```
//Creation of post files
PostFiles
MagFactor      1
WriteMesh      1
WriteRenderMesh    1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces     0
WriteForces     1
WriteConstraints 0
WriteSpecialConstraints 0
WriteContactForces 0
WriteRenderRigidBodies 0
WriteRenderParticles 0
```

Note that the Giraffe input file commands not necessarily has to follow the here proposed sequence. The important is that all the commands here presented are included in this tutorial.

Results and post-processing

Result files are automatically saved during the solving process on the folder “/tutorial01/post/” and “/tutorial01/monitors/”

Paraview™ post-processing

Opening in Paraview™ the “.pvf” files generated during simulation leads to visualization of the model results. The user may perform animations and extract high-quality images for reports.

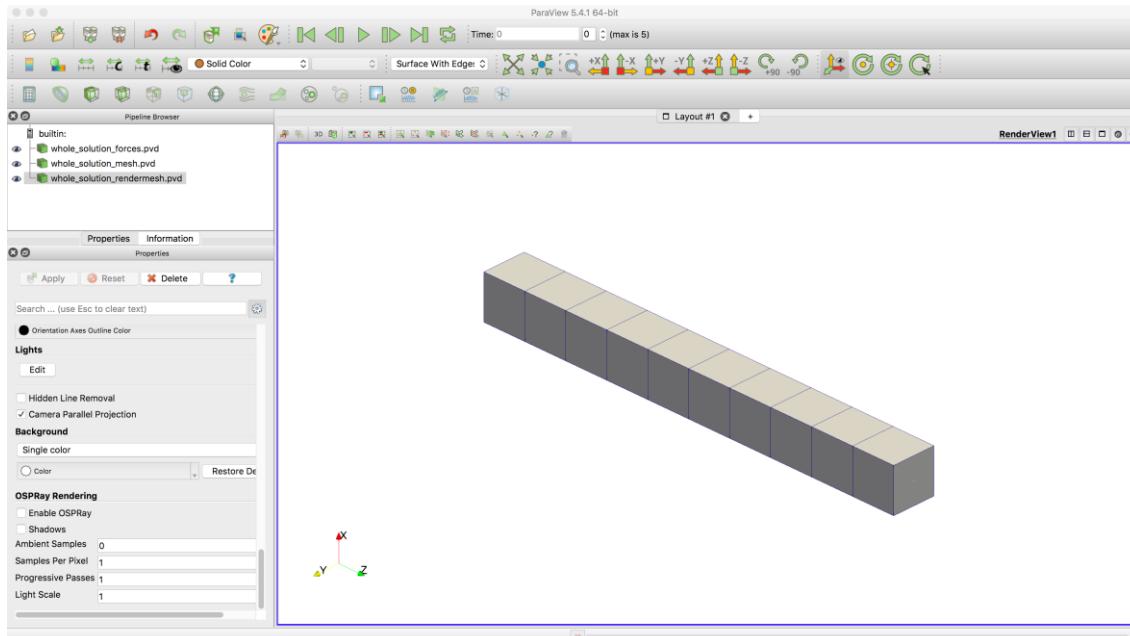


Figure 1-2 – Results rendered using Paraview™

As a particular feature, forces output may be used in Paraview™ together with Glyphs in order to establish special visualizations. For that, select “whole_solution_forces.pvd” in the Pipeline Browser. Then, click in the Glyph feature button to create a Glyph.

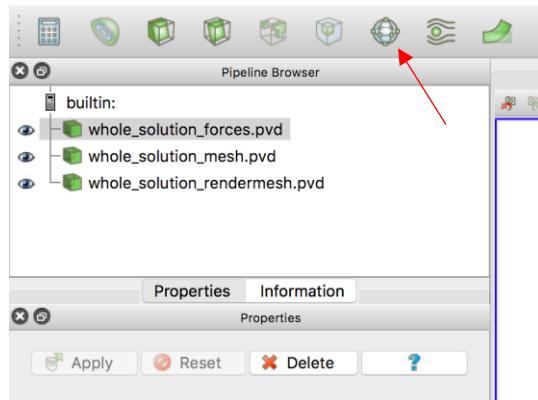


Figure 1-3 – Using Glyphs in Paraview™

After creating the Glyph, select it on Pipeline Browser to modify its default options as depicted next. Glyph type is set to “Arrow”. Scale Mode is set to “vector” and scale factor is set to 0.001. This value is arbitrary and may vary from model to model and controls the size of the created Glyph feature. It should be chosen based on the desired size of the Glyph feature on the visualization of the model.

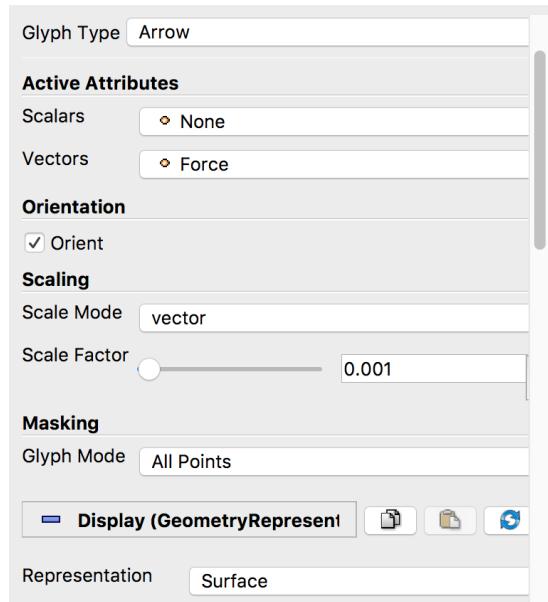


Figure 1-4 – Setting Glyph attributes in Paraview™

Visualization will look like the shown in Figure 1-5 (changing the color of the Glyph feature to red):

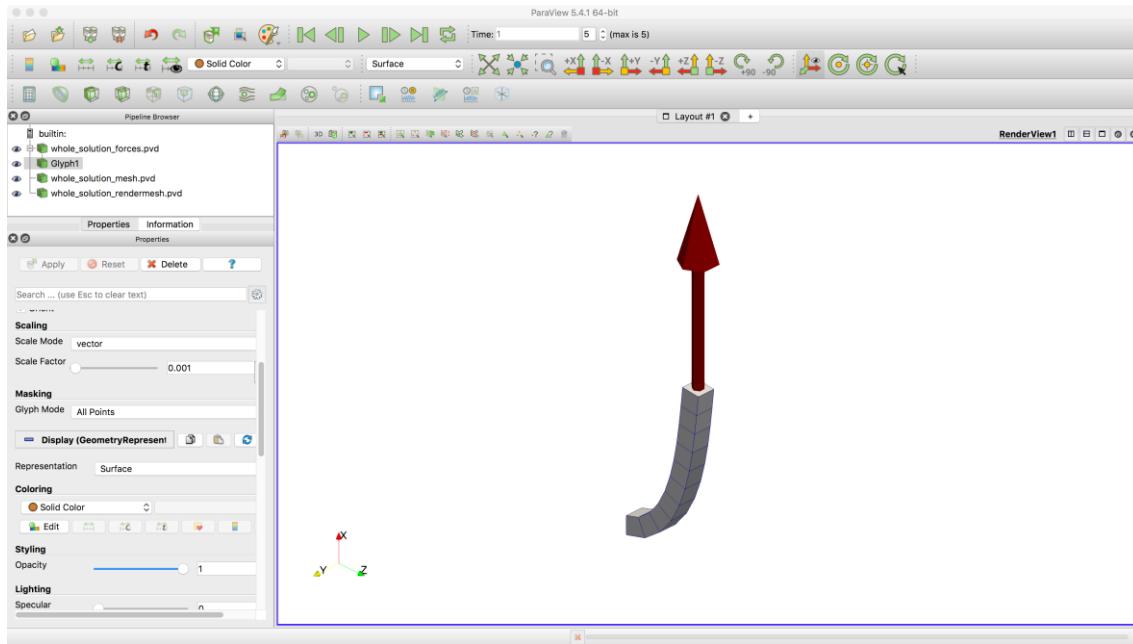


Figure 1-5 – Using Glyphs in Paraview™

The user may animate results to see the Glyph feature changing its size, according to the force magnitude along time.

Monitors post-processing

Monitors results are text files that saves time-series of specific results. In this example, they are associated with nodes. An example of post-processing is to plot the point B position along "time", from 0-1.

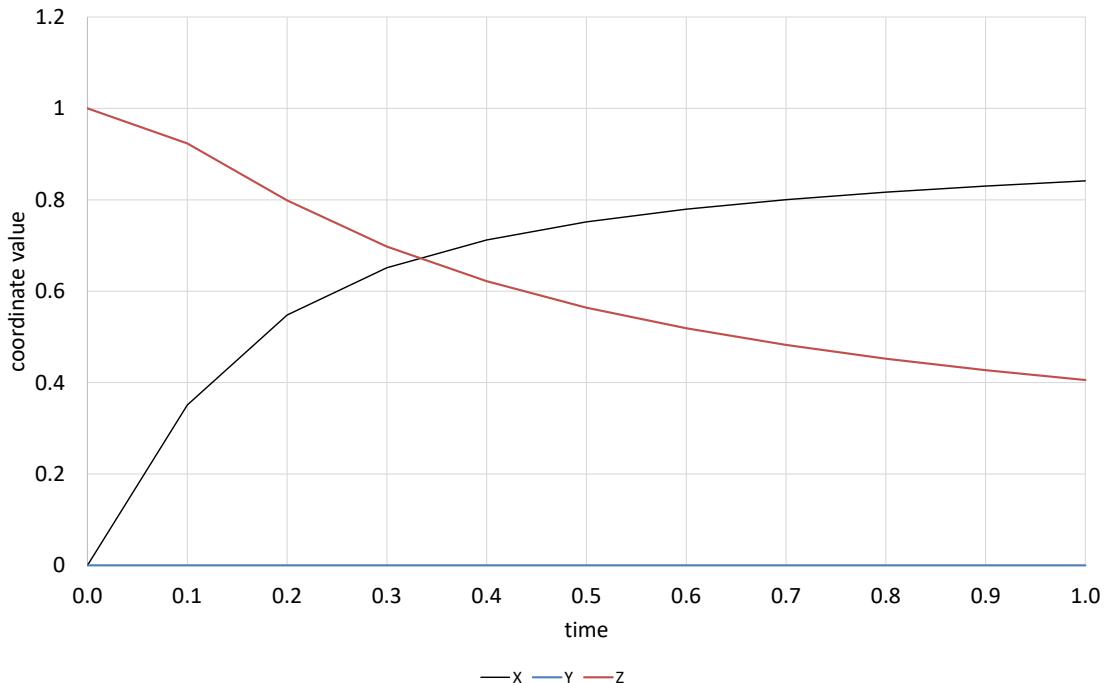


Figure 1-6 – Monitoring the point B coordinates time-evolution

What to try now?

- change the magnitude and the direction of the force
- create a load composed by moments, instead of forces
- change the boundary conditions and apply loads in distinct nodes
- change the time-step size to smaller or larger values to see the influence on simulation evolution
- change the sampling rate for results saving both on monitor and on solution steps
- change the beam cross-section and material properties
- refine the mesh

Tutorial 2. Modal analysis of a circular plate

Input file name: tutorial02.inp

Developed by Paulo Roberto Refachinho & Gabriel Maki

Objective

This tutorial presents the modal analysis of a circular plate, clamped on its boundaries. The tutorial objective is to show how to perform a modal analysis and post-process results using Paraview™ software.

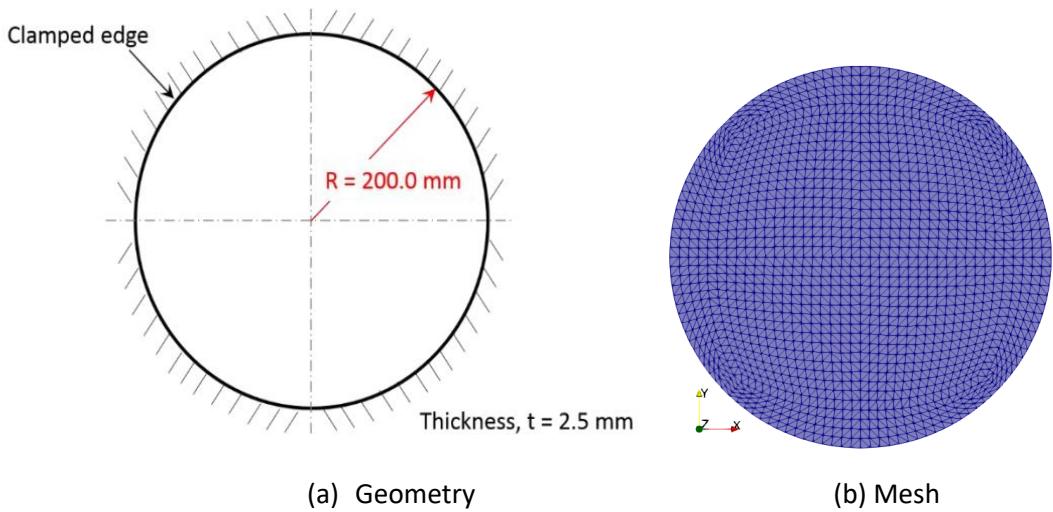


Figure 2-1 – Thin circular plate (a) Geometry (b) Mesh

Description

The plate is meshed using 6197 nodes and 3036 elements, using the element Shell_1. There is a clamp boundary condition on the edges of the plate. The section has a thickness of 2.5 mm. The plate is contained in the plane perpendicular to the direction z.

Input File step-by-step

Creation of nodes

A total of 6197 nodes are created to establish a finite element mesh. Note that here a precision of 14 digits is used to write the input file. This is recommended always when generating Giraffe input files. The scientific notation format is recommended, as shown in this example, unless when dealing with a very simple nodal sequence, such as described in tutorial 01.

```
//Creation of nodes
Nodes 6197
Node 1 0.00000000000000E+00 0.00000000000000E+00 0.00000000000000E+00
Node 2 5.00000000000000E+00 0.00000000000000E+00 0.00000000000000E+00
Node 3 1.00000000000000E+01 0.00000000000000E+00 0.00000000000000E+00
Node 4 1.49999999999999E+01 0.00000000000000E+00 0.00000000000000E+00
Node 5 1.99999999999999E+01 0.00000000000000E+00 0.00000000000000E+00
Node 6 2.49999999999999E+01 0.00000000000000E+00 0.00000000000000E+00
Node 7 2.99999999999999E+01 0.00000000000000E+00 0.00000000000000E+00
Node 8 3.49999999999999E+01 0.00000000000000E+00 0.00000000000000E+00
Node 9 4.00000000000000E+01 0.00000000000000E+00 0.00000000000000E+00
Node 10 4.50000000000000E+01 0.00000000000000E+00 0.00000000000000E+00
Node 11 5.00000000000000E+01 0.00000000000000E+00 0.00000000000000E+00
Node 12 5.49999999999999E+01 0.00000000000000E+00 0.00000000000000E+00
Node 13 6.00000000000000E+01 0.00000000000000E+00 0.00000000000000E+00
Node 14 6.49999999999999E+01 0.00000000000000E+00 0.00000000000000E+00
Node 15 6.99999999999999E+01 0.00000000000000E+00 0.00000000000000E+00
Node 16 7.49999999999999E+01 0.00000000000000E+00 0.00000000000000E+00
Node 17 7.99999999999999E+01 0.00000000000000E+00 0.00000000000000E+00
Node 18 8.49999999999999E+01 0.00000000000000E+00 0.00000000000000E+00
Node 19 9.00000000000000E+01 0.00000000000000E+00 0.00000000000000E+00
Node 20 9.50000000000000E+01 0.00000000000000E+00 0.00000000000000E+00
```

...

Node	6165	1.53058982842033E+02	-1.14026258730851E+02	0.00000000000000E+00
Node	6166	1.47913129412780E+02	-1.21843677440843E+02	0.00000000000000E+00
Node	6167	1.42432380390826E+02	-1.29390594783211E+02	0.00000000000000E+00
Node	6168	1.74941348706255E+02	-4.84057409095254E+00	0.00000000000000E+00
Node	6169	1.74668214191052E+02	-1.43671212253454E+01	0.00000000000000E+00
Node	6170	1.74020872610914E+02	-2.38210467458590E+01	0.00000000000000E+00
Node	6171	1.72999502067091E+02	-3.31867762377589E+01	0.00000000000000E+00
Node	6172	1.71605951573866E+02	-4.24483843164989E+01	0.00000000000000E+00
Node	6173	1.69852358429996E+02	-5.15942395774600E+01	0.00000000000000E+00
Node	6174	1.67736588624359E+02	-6.06070008207730E+01	0.00000000000000E+00
Node	6175	1.65264625712240E+02	-6.94721590337491E+01	0.00000000000000E+00
Node	6176	1.62451464824384E+02	-7.81804125423577E+01	0.00000000000000E+00
Node	6177	1.59292155070117E+02	-8.67116171422745E+01	0.00000000000000E+00
Node	6178	1.55801013230708E+02	-9.50559369314019E+01	0.00000000000000E+00
Node	6179	1.51987536563878E+02	-1.03201993087607E+02	0.00000000000000E+00
Node	6180	1.47855378730407E+02	-1.11133271194406E+02	0.00000000000000E+00
Node	6181	1.43412535206243E+02	-1.18836169266648E+02	0.00000000000000E+00
Node	6182	1.38680440329816E+02	-1.26309596164411E+02	0.00000000000000E+00
Node	6183	1.64989172058302E+02	-4.71629909177040E+00	0.00000000000000E+00
Node	6184	1.64847824124615E+02	-1.39824570121293E+01	0.00000000000000E+00
Node	6185	1.64382576894756E+02	-2.31704236670644E+01	0.00000000000000E+00
Node	6186	1.63591871432435E+02	-3.22667440764171E+01	0.00000000000000E+00
Node	6187	1.62484093347860E+02	-4.12620306230941E+01	0.00000000000000E+00
Node	6188	1.61064694050891E+02	-5.01458000449104E+01	0.00000000000000E+00
Node	6189	1.59328661600592E+02	-5.89021671438646E+01	0.00000000000000E+00
Node	6190	1.57290898762299E+02	-6.75253544690905E+01	0.00000000000000E+00
Node	6191	1.54955010068830E+02	-7.60041636670082E+01	0.00000000000000E+00
Node	6192	1.52320974309620E+02	-8.43241501432042E+01	0.00000000000000E+00
Node	6193	1.49397692831169E+02	-9.24769180839118E+01	0.00000000000000E+00
Node	6194	1.46199126112956E+02	-1.00458125059544E+02	0.00000000000000E+00
Node	6195	1.42717259396612E+02	-1.08246358021446E+02	0.00000000000000E+00
Node	6196	1.39324988441771E+02	-1.15795202781407E+02	0.00000000000000E+00
Node	6197	1.35336888727668E+02	-1.23202163313770E+02	0.00000000000000E+00

Creation of node sets

A node set, which has 248 nodes, is created to establish the constraints.

```
//Creation of node sets (in this case, a single node set)
NodeSets 1
```

NodeSet	1	Nodes	248	List	41	42	43	44	45	46	47
---------	---	-------	-----	------	----	----	----	----	----	----	----

...

4739	4740	4741	4742	4743	4744	4745	4746	4747	4748	4749
------	------	------	------	------	------	------	------	------	------	------

Creation of elements

3036 elements are defined. All the elements have the same material properties (identification number 1) and the same cross-section (identification number 1). Each Shell_1 element has the shape of a triangle and is defined by six nodes, where three nodes are located at the vertices and the others are positioned at the midpoints of the sides of the triangle. The nodes within an element must lie in a plane at the given reference configuration mesh. After deforming, arbitrary displacements may occurs and the shell element will be no longer contained in a plane.

//Creation of elements
Elements 3036
Shell_1 1 Mat 1 Sec 1 Nodes 143 157 186 156 185 1590
Shell_1 2 Mat 1 Sec 1 Nodes 150 186 188 192 187 1589
Shell_1 3 Mat 1 Sec 1 Nodes 152 188 190 193 189 1588
Shell_1 4 Mat 1 Sec 1 Nodes 154 190 71 194 1587 155
Shell_1 5 Mat 1 Sec 1 Nodes 157 159 196 158 195 1586
Shell_1 6 Mat 1 Sec 1 Nodes 186 196 198 202 197 1585
Shell_1 7 Mat 1 Sec 1 Nodes 188 198 190 203 1584 189
Shell_1 8 Mat 1 Sec 1 Nodes 190 200 69 204 1583 191
Shell_1 9 Mat 1 Sec 1 Nodes 159 161 206 160 205 1582
Shell_1 10 Mat 1 Sec 1 Nodes 196 206 198 212 1581 197
Shell_1 11 Mat 1 Sec 1 Nodes 198 208 200 213 1580 199
Shell_1 12 Mat 1 Sec 1 Nodes 200 210 67 214 1579 201
Shell_1 13 Mat 1 Sec 1 Nodes 161 163 216 162 215 1578
Shell_1 14 Mat 1 Sec 1 Nodes 206 216 208 222 1577 207

...

//Creation of elements
Elements 3036
Shell_1 3022 Mat 1 Sec 1 Nodes 4819 35 33 4820 34 6183
Shell_1 3023 Mat 1 Sec 1 Nodes 4817 4819 4783 4818 4834 6184
Shell_1 3024 Mat 1 Sec 1 Nodes 4815 4817 4781 4816 4833 6185
Shell_1 3025 Mat 1 Sec 1 Nodes 4813 4815 4779 4814 4832 6186
Shell_1 3026 Mat 1 Sec 1 Nodes 4811 4813 4777 4812 4831 6187
Shell_1 3027 Mat 1 Sec 1 Nodes 4809 4811 4775 4810 4830 6188
Shell_1 3028 Mat 1 Sec 1 Nodes 4807 4809 4773 4808 4829 6189
Shell_1 3029 Mat 1 Sec 1 Nodes 4805 4807 4771 4806 4828 6190
Shell_1 3030 Mat 1 Sec 1 Nodes 4803 4805 4769 4804 4827 6191
Shell_1 3031 Mat 1 Sec 1 Nodes 4801 4803 4767 4802 4826 6192
Shell_1 3032 Mat 1 Sec 1 Nodes 4799 4801 4765 4800 4825 6193
Shell_1 3033 Mat 1 Sec 1 Nodes 4797 4799 4763 4798 4824 6194
Shell_1 3034 Mat 1 Sec 1 Nodes 4795 4797 4761 4796 4823 6195
Shell_1 3035 Mat 1 Sec 1 Nodes 4795 4759 4757 4822 4758 6196
Shell_1 3036 Mat 1 Sec 1 Nodes 4793 4757 4750 4821 4756 6197

Creation of materials

One only material is created, which identification number is 1.

//Creation of materials

Materials	1				
Hooke	1	E	200000	Nu	0.30

Creation of shell sections

One shell section is defined, which has a thickness of 2.5 cm and it is homogeneous. Its identification number is 1.

```
//Creation of shell sections
ShellSections 1
Homogeneous 1      Thickness    2.5
```

Creation of constraints

A nodal constraint is created to fix all degrees of freedom of the nodes contained in the node set defined previously. In this case, the nodal constraint is established for the solution step 1 (in current tutorial there is a single solution step), which will be shown in the next topic.

```
//Creation of constraints
Constraints 1
NodalConstraint   1      NodeSet      1
UX   BoolTable   1
UY   BoolTable   1
UZ   BoolTable   1
ROTX BoolTable   1
ROTY BoolTable   1
ROTZ BoolTable   1
```

Creation of solution steps

This tutorial has a single solution step (modal). There is the possibility of choosing the number of modes to be evaluated, the tolerance for eigenvalues and the number of frames written for post-processing the mode shapes, using Paraview™. Alternatively, the user may export matrices involved in modal analysis (mass and stiffness matrix). However, this will generate large-size text files. We chose here not to export such files.

```
//Creation of solution steps
SolutionSteps1
Modal 1
ExportMatrices 0
NumberModes 12
Tolerance 1e-6
ComputeEigenvectors 1
NumberFrames 12
```

Creation of solver options

Here it is possible to choose the number of processors used (parallel processing). Besides, we must define the solver type for systems of linear equations. In this case, two threads are employed and the solver choice is “Direct”.

```
//Creation of solver options
SolverOptions
Processors 2 LinSys Direct
```

In this tutorial, there is no need to create monitors and post-files.

Results and post-processing

In the case of a single solution step analysis, Giraffe saves result files on “/post/solution_1”. The file “solution_1_mesh.pvd” contain all the twelve requested modes.

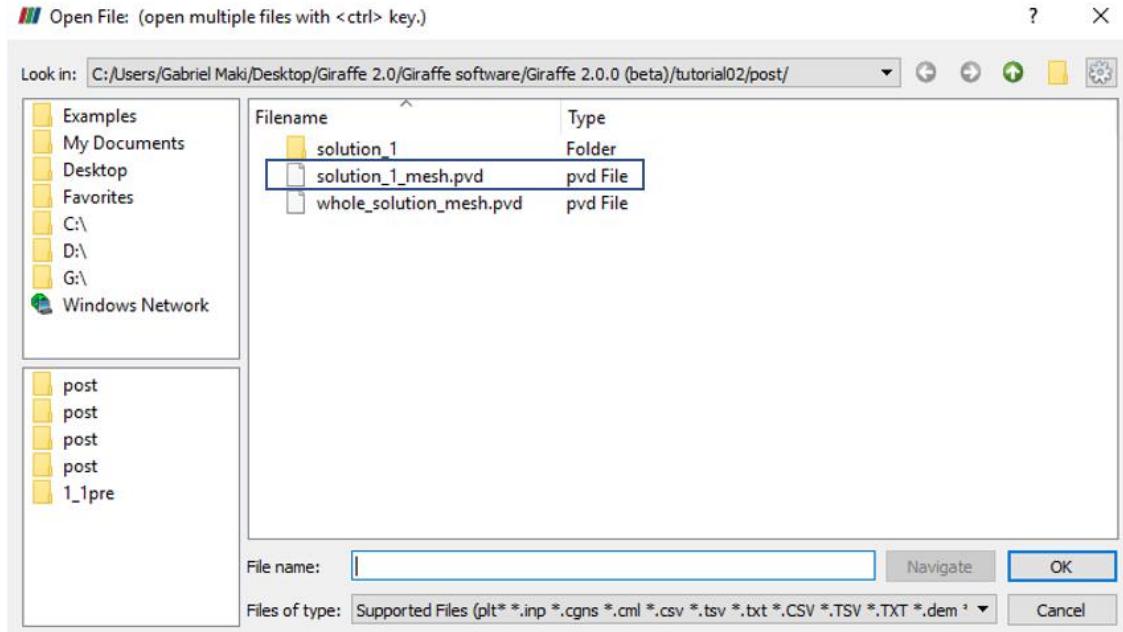


Figure 2-2 – Opening the file in Paraview™.

To view a specific vibration mode, first we must open in Paraview™ the file mentioned, as shown in Figure 2-2, and click the preview icon and the Apply button on the Pipeline browser (Figure 2-3).

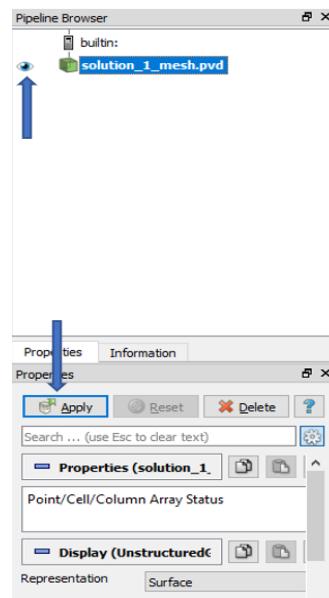


Figure 2-3 – Pipeline Browser.

The following figure shows how Paraview™ displays the results.

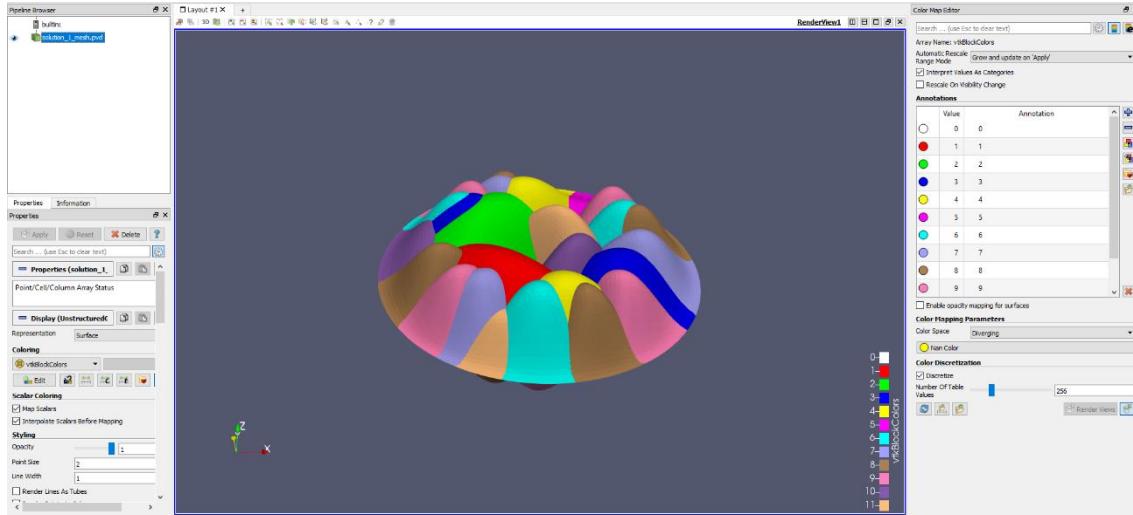


Figure 2-4 – Results of modal analysis.

The second step is extracting the mode of interest, which can be done using the “Extract Block” filter.

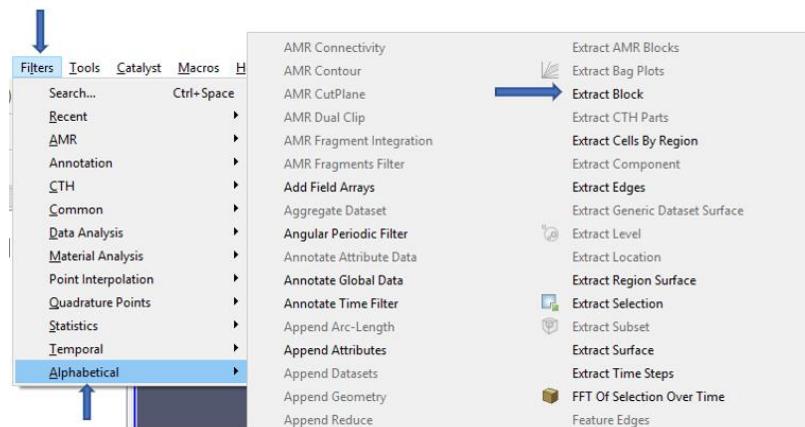


Figure 2-5 – Finding the “Extract Block”.

After clicking in the filter, a menu will appear to choose the desired block. Each block contains results for a mode. In this case, there will be twelve blocks (associated with the requested twelve first vibration modes). To animate results for a vibration mode, we must choose the desired block and click “Apply”.

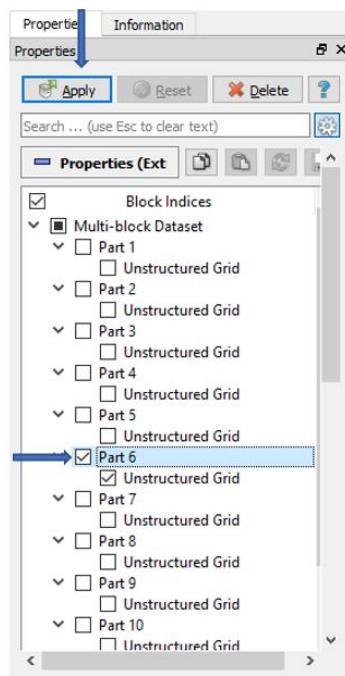


Figure 2-6 – Choosing the sixth mode of vibration.

Figure 2-7 shows an example of mode of vibration plot. It shows the expected shape when the structure experiences vibration patterns associated with a corresponding natural frequency.

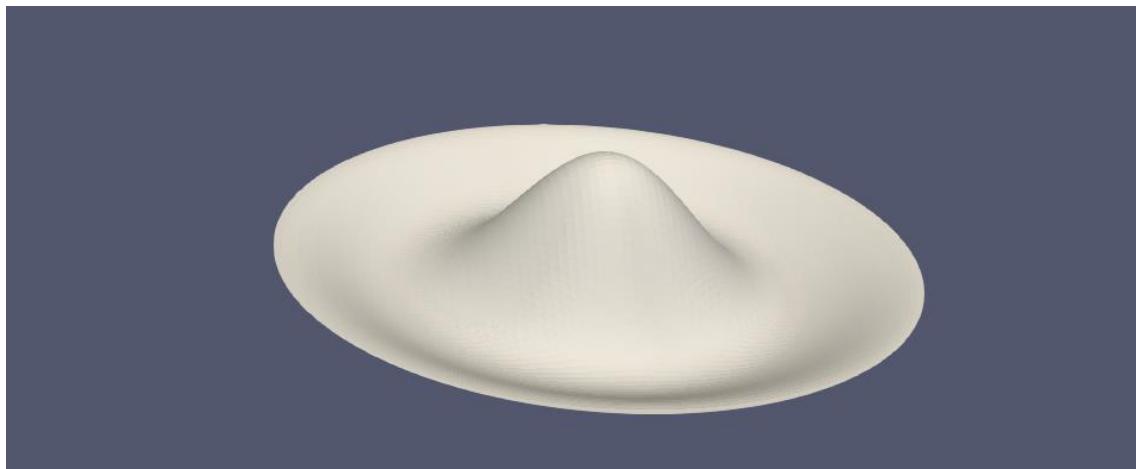


Figure 2-7 – Animation of the chosen mode of vibration.

Finally, there is a text file “solution_1_eigenvalues” that contains the eigenvalues obtained from the modal analysis. It is located inside the folder “post/solution_1”. To obtain the natural frequencies, we must extract the square root of the eigenvalues and divide this by 2π .

Eigenvalues - solution number 1		
Mode	Re	Im
1	1009519.548036	0.000000
2	4316095.037999	0.000000
3	4316095.038019	0.000000
4	11630174.148244	0.000000
5	11561681.465562	0.000000
6	14897498.883645	0.000000
7	24801521.212492	0.000000
8	24801521.212497	0.000000
9	46155343.969949	0.000000
10	46177633.073695	0.000000
11	34857355.962447	0.000000
12	34857355.962431	0.000000

Tutorial 3. Buckling of a pipe triggered by a compression follower load

Input file name: tutorial03.inp

Developed by Alfredo Gay Neto & Gabriel Maki

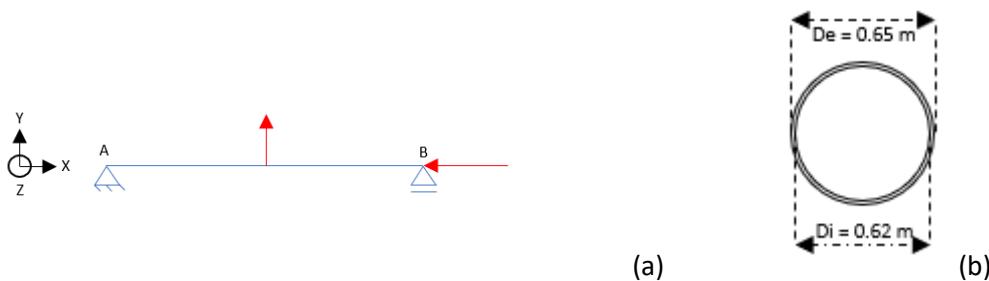


Figure 3-1 – (a) Model of a pipe experiencing buckling (b) Cross section of the pipe

Objective

This tutorial presents a geometrically-nonlinear simulation of a pipe. The aim is to evaluate the buckling of the structure, when subjected to a compression load. For that, it is necessary to introduce an imperfection on the pipe and the compression (trigger) load, as shown in Figure 3-1(a).

Description

The pipe AB (100 m length) is meshed considering 101 equally-spaced nodes and 50 elements, (Pipe_1). Two nodal loads are applied sequentially. First, at the middle-spam of the pipe, a lateral load is applied. After, at point B, a compressive nodal load is applied. The force in the middle-spam is necessary because it is not possible to perform a nonlinear buckling analysis without an initial imperfection on the structure. This imperfection induces certain buckling mode. It could be considered also a geometric imperfection directly in the model mesh. In our case, however, the lateral load introduces some curvature to the pipe that leads to subsequent buckling, when the structure experiences the compression load. The cross section of the pipe is shown in Figure 3-1(b) and the pipe is initially straight and aligned in direction x.

Input File Step by Step

Creation of nodes

A total of 101 nodes is created to establish a finite element mesh.

```
//Creation of nodes
Nodes 101
Node 1 0 0 0
Node 2 1 0 0
Node 3 2 0 0
Node 4 3 0 0
Node 5 4 0 0
Node 6 5 0 0
```

Node	7	6	0	0
Node	8	7	0	0
Node	9	8	0	0
Node	10	9	0	0
Node	11	10	0	0
Node	12	11	0	0
Node	13	12	0	0
Node	14	13	0	0
Node	15	14	0	0
...				
Node	90	89	0	0
Node	91	90	0	0
Node	92	91	0	0
Node	93	92	0	0
Node	94	93	0	0
Node	95	94	0	0
Node	96	95	0	0
Node	97	96	0	0
Node	98	97	0	0
Node	99	98	0	0
Node	100	99	0	0
Node	101	100	0	0

Creation of node sets

Three node sets are defined to establish the constraints and loads. Node sets 1 and 2 are associate with nodes located at the ends of the pipe. Node set 3 contains the node located at the middle of the pipe, where the imperfection load takes place.

//Creation of node sets
NodeSets 3
//First node
NodeSet 1 Nodes 1 List 1
//Last node
NodeSet 2 Nodes 1 List 101
//Middle node
NodeSet 3 Nodes 1 List 51

Creation of elements

A total of 50 elements is created. All the elements have the same cross section (identification number 1) and the same coordinate system (identification number 1). Each Pipe_1 element is defined by three nodes, which must be equally spaced and contained in a straight line.

//Creation of elements
Elements 50
Pipe_1 1 PipeSec 1 CS 1 Nodes 1 2 3
Pipe_1 2 PipeSec 1 CS 1 Nodes 3 4 5
Pipe_1 3 PipeSec 1 CS 1 Nodes 5 6 7
Pipe_1 4 PipeSec 1 CS 1 Nodes 7 8 9
Pipe_1 5 PipeSec 1 CS 1 Nodes 9 10 11
Pipe_1 6 PipeSec 1 CS 1 Nodes 11 12 13
Pipe_1 7 PipeSec 1 CS 1 Nodes 13 14 15
Pipe_1 8 PipeSec 1 CS 1 Nodes 15 16 17
Pipe_1 9 PipeSec 1 CS 1 Nodes 17 18 19
Pipe_1 10 PipeSec 1 CS 1 Nodes 19 20 21
Pipe_1 11 PipeSec 1 CS 1 Nodes 21 22 23
Pipe_1 12 PipeSec 1 CS 1 Nodes 23 24 25
Pipe_1 13 PipeSec 1 CS 1 Nodes 25 26 27
Pipe_1 14 PipeSec 1 CS 1 Nodes 27 28 29
Pipe_1 15 PipeSec 1 CS 1 Nodes 29 30 31

...							
Pipe_1 40	PipeSec	1	CS	1	Nodes	79	80
Pipe_1 41	PipeSec	1	CS	1	Nodes	81	82
Pipe_1 42	PipeSec	1	CS	1	Nodes	83	84
Pipe_1 43	PipeSec	1	CS	1	Nodes	85	86
Pipe_1 44	PipeSec	1	CS	1	Nodes	87	88
Pipe_1 45	PipeSec	1	CS	1	Nodes	89	90
Pipe_1 46	PipeSec	1	CS	1	Nodes	91	92
Pipe_1 47	PipeSec	1	CS	1	Nodes	93	94
Pipe_1 48	PipeSec	1	CS	1	Nodes	95	96
Pipe_1 49	PipeSec	1	CS	1	Nodes	97	98
Pipe_1 50	PipeSec	1	CS	1	Nodes	99	100
							101

Creation of the cross section

One only pipe cross section is created, which is shown in the Figure 3-1(b). It is important to mention that, unlike the tutorials previously presented, there is no need to define the material properties. Instead, here we use equivalent stiffness properties of the pipe to rule the structural behavior. These are to be directly input by the user (axial, bending, torsion and shear stiffness). This element may also be employed for simulations involving drag forces, when using it to represent offshore pipelines, risers or similar structures.

//Creation of pipe sections
PipeSections 1
PS 1 EA 5984734005.0 EI 301817616.8 GJ 301817616.8 GA
2992367003 Rho 239.3893602 CDt 0 CDn 0 CAT 0
CAn 0 De 0.65 Di 0.62

Creation of coordinate systems

Two coordinate systems are created. The first one is used to define the element properties. Note that, since employing Pipe_1 element, direction E3 must lie in the pipe alignment direction (exactly as previously discussed for Beam_1 element). The second coordinate system is used to define the direction of application of the nodal loads. It is optional but may provide a convenient way to create loads. Alternatively, the user could also employ the first coordinate system to create the loads.

//Creation of coordinate systems
CoordinateSystems 2
CS 1 E1 0 1 0 E3 1 0 0
CS 2 E1 1 0 0 E3 0 0 1

Creation of solution steps

This tutorial has two solution steps (both are statics). The first step is created establishing the end time equal to "1". The time step is "1", which means that the load (imperfection force) is applied in a single step. The second step is created establishing the end time equal to "2". The time step used is "0.005", that is, the compression force is divided into increments of 0.5% (variable according to an automatic time-stepping scheme based on difficulties for achieving convergence along simulation evolution).

```
//Creation of solution steps
SolutionSteps      2
//Step 1 - Static - introduction of the perturbation load. Time: 0-1
Static 1          EndTime      1      TimeStep      1.0      MaxTimeStep 1.0
               MinTimeStep 0.01  MaxIt 15    MinIt 3      ConvIncrease 2
               IncFactor   1.0    Sample 1
//Step 2 - Static - introduction of the follower load. Time: 1-2
Static 2          EndTime      2      TimeStep      0.005  MaxTimeStep 1
               MinTimeStep 0.002  MaxIt 15    MinIt 3      ConvIncrease 2
               IncFactor   1.0    Sample 1
```

Creation of loads

One nodal load and one nodal follower load are defined. Both are established by a force time series. The first is defined in Y direction and the second in X direction (considering CS 2). It is important to remember that the first load is the imperfection force, used to make feasible the evaluation of buckling. The second load is the compression force, which causes the buckling phenomenon studied. Besides, the follower load has a particular characteristic: it updates direction according to the rotations experienced by the node. So, it is possible to state that the compression force is always aligned with the pipe tip, in current configuration. Here the user may also replace the triggering nodal follower load by a nodal load to see the differences in results.

```
//Creation of loads
Loads 2
//Imperfection load in the middle of the pipe
NodalLoad 1      NodeSet 3      CS 2      NTimes 2
//Time FX FY FZ MX MY MZ
0      0      0      0      0      0      0
1      0      1000  0      0      0      0
//Follower load in the last node of the pipe to compress it
NodalFollowerLoad 2      NodeSet 2      CS 2      NTimes 2
//Time FX FY FZ MX MY MZ
1      0      0      0      0      0      0
2      -362288.4648 0      0      0      0      0
```

Creation of constraints

Two nodal constraints are created. The first, at the point A of the Figure 3-1, is done by fixing DOFs: UX, UY, UZ, ROTX. The second constraint, at the point B, is done by fixing DOFs: UY and UZ. Note that Figure 3-1 represents a bi-dimensional modeling of the problem, but the modeling in GIRAFFE is 3D, so the degree of freedom corresponding to a rotation in X direction must be fixed to restrict such rigid body movement, but it is not represented in Figure 3-1. Note that the degrees of freedom that were not fixed do not need to appear on the file.

```
//Creation of constraints
Constraints 2
NodalConstraint 1      NodeSet 1
               UX BoolTable 1
               UY BoolTable 1
               UZ BoolTable 1
               ROTX BoolTable 1
NodalConstraint 2      NodeSet 2
               UY BoolTable 1
               UZ BoolTable 1
```

Creation of solver options

Here it is possible to choose the number of processors (threads) to be used (parallel processing). Besides, we must define the solver type for systems of linear equations. In this case, four processors are employed and the direct type of linear system solver is our choice.

```
//Creation of solver options
SolverOptions
Processors    4      LinSys Direct
```

Creation of monitors

We request Giraffe to monitor 3 nodes during the simulation: numbers 1 (point A), 51 (middle-spam) and 101 (point B).

```
Monitor      Sample 1
MonitorNodes 1      101      51
```

Creation of post files

The options used here are the same as the tutorial 01, where the WriteMesh, WriteRenderMesh and WriteForces are chosen for post-processing using Paraview™ and the MagFactor parameters assume a unit value so the deformed shape of Paraview™ frames will reflect the real scale.

```
PostFiles
MagFactor    1
WriteMesh     1
WriteRenderMesh    1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces    0
WriteForces    1
WriteConstraints    0
WriteSpecialConstraints    0
WriteContactForces    0
WriteRenderRigidBodies    0
WriteRenderParticles    0
```

Creations of convergence criteria

Here, we must define some convergence and divergence tolerances. For detailed explanation on the meaning of each of these quantities, refer to Giraffe users' manual.

Note: altering default convergence criteria is recommended only for advanced users, since one may create incorrect simulation results by doing so.

ConvergenceCriteria	
ForceTolerance	1.000000e-004
MomentTolerance	1.000000e-004
ForceMinimumReference	1.000000e-001
MomentMinimumReference	1.000000e-001
ConstraintMinimumReference	1.000000e-007
DisplacementTolerance	1.000000e-004
RotationTolerance	1.000000e-004
LagrangeTolerance	1.000000e-004
DisplacementMinimumReference	1.000000e-006
RotationMinimumReference	1.000000e-006
LagrangeMinimumReference	1.000000e-006
DivergenceReference	1.000000e+015

Results and Post-Processing

To post-process results using Paraview™, first, we must open the files “whole_solution_forces.pvd” and “whole_solution_mesh.pvd”, which are located at the folder “post” in “tutorial03”.

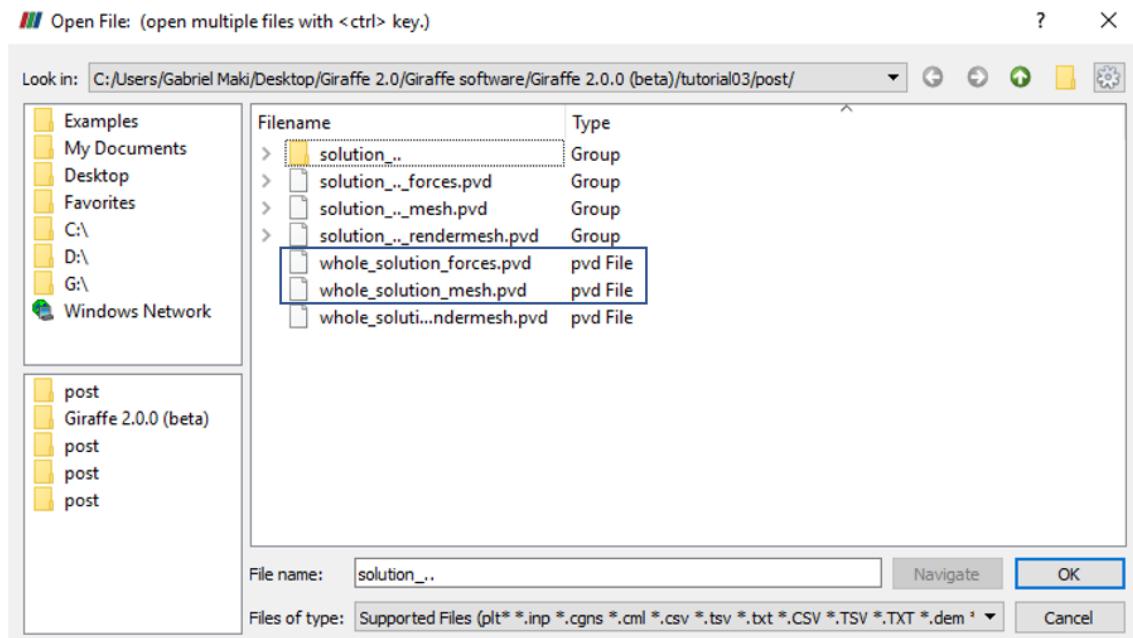


Figure 3-2 – Opening the files in Paraview™.

Then, we must click the preview icon and the Apply button on the Pipeline browser. Note that the user has also the possibility of opening individual solution steps results, according to necessity.

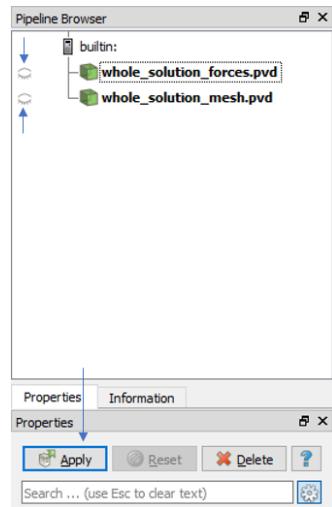


Figure 3-3 – Pipeline Browser.

Figure 3-4 shows how Paraview™ displays the results.

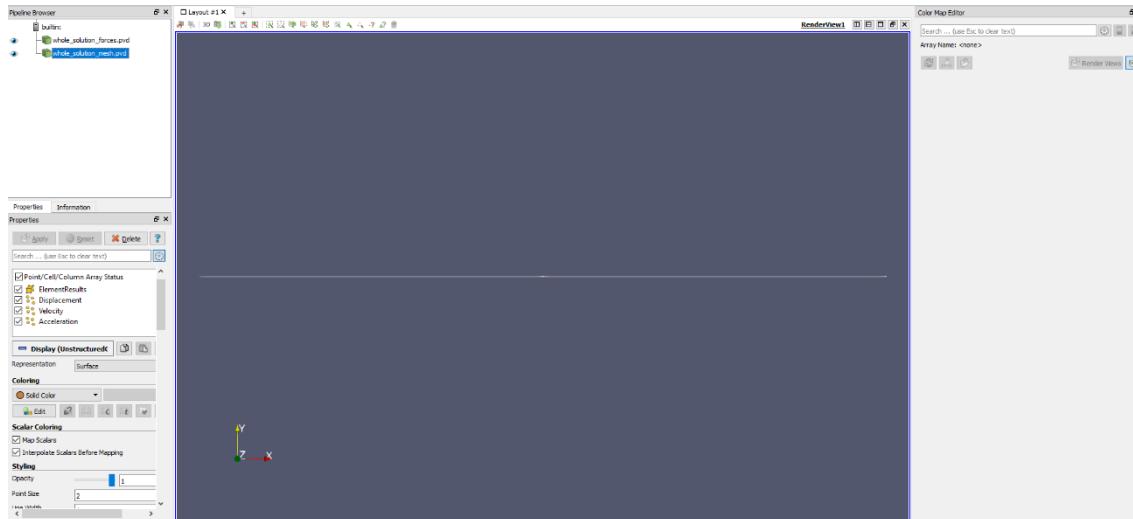


Figure 3-4 – Results using Paraview™

Next, we can create a Glyph for observing the load being applied. For that, go to the Pipeline Browser and click in “whole_solution_forces.pvd”. Then, click the Glyph, as shown in Figure 3-5.

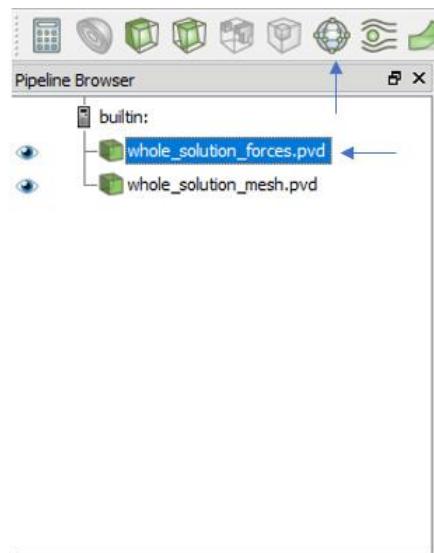


Figure 3-5 – Creation of a Glyph.

After creating the Glyph, we must adjust the scale of vectors. For that, go to the Pipeline Browser and, first, click the preview icon of Glyph. Then, in Scaling, set the scale mode to vector and change the scale factor to a small number, e.g.: 0.0001. Finally, click Apply.

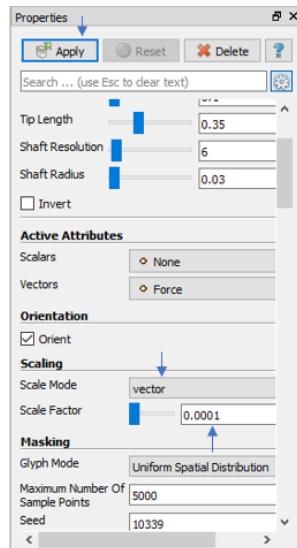


Figure 3-6 – Adjusting the Glyph.

Finally, we can make an animation and view the structure of the pipe experiencing buckling. If the user wants to know the values of displacement of selected nodes or forces in a certain instant, it is possible to obtain such information in the “monitors” folder. In this case, we requested monitors for three nodes (points where the forces were applied) but the user can request another node. Giraffe generates a text file for each node requested containing the values of displacement and force/moment in each global direction. It is possible to transfer the content of the file to a spreadsheet, getting easier to plot the values.

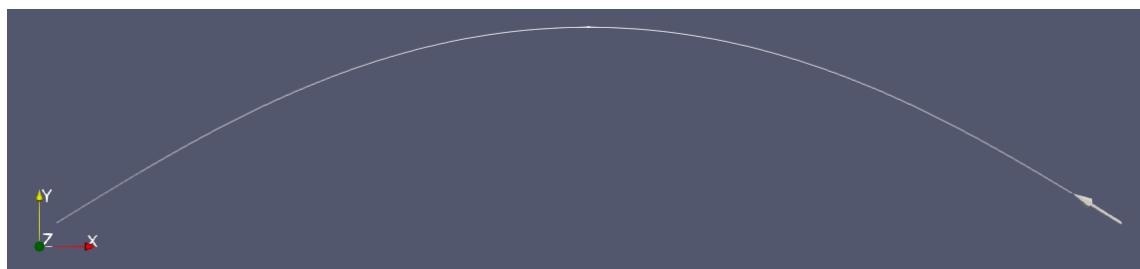


Figure 3-7 – Deformed shape of pipe at the end of simulation.

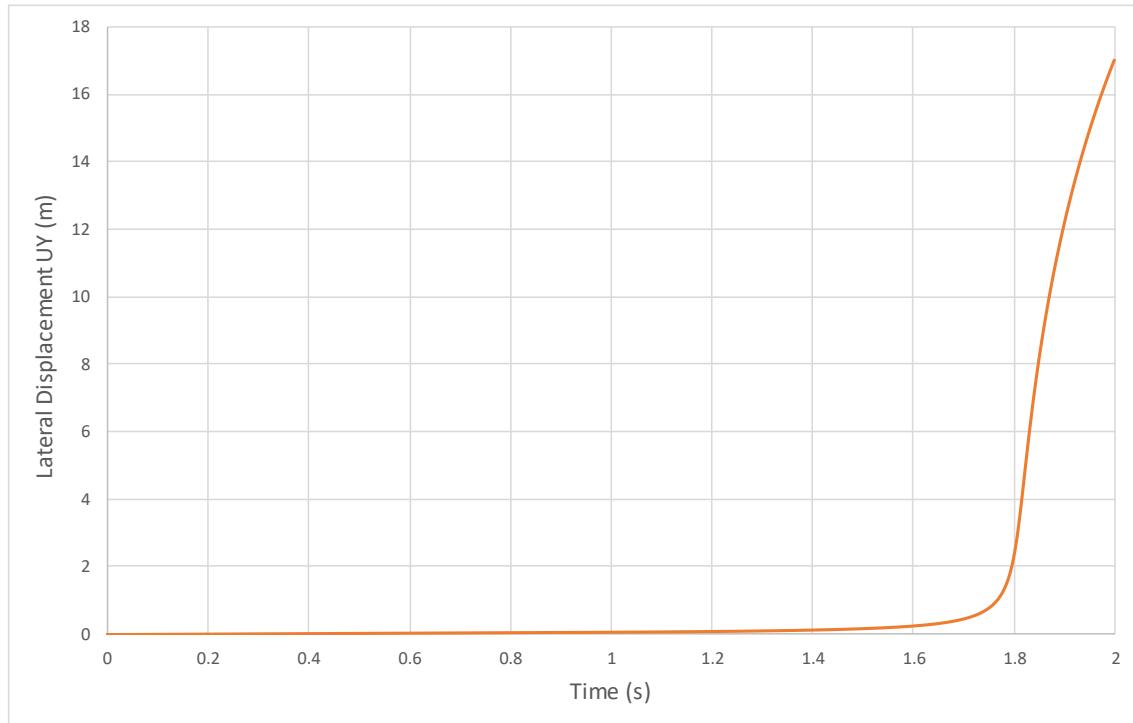


Figure 3-8 – Lateral displacement (UY) of the pipe middle span vs. time

Tutorial 4. Buckling of a pipe triggered by internal pressure

Input file name: tutorial04.inp

Developed by Alfredo Gay Neto & Gabriel Maki

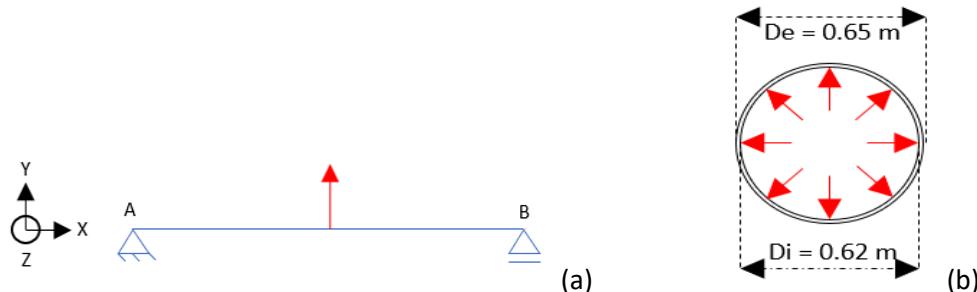


Figure 4-1 – (a) Beam model of a double-based beam (b) Cross section of a pipe subjected to internal pressure field.

Objective

This tutorial presents a geometrically-nonlinear simulation of a pipe. The aim is to evaluate the buckling of the structure, when subjected to an internal pressure load.

Description

The pipe AB (100 m length) is meshed considering 101 equally-spaced nodes and 50 elements (Pipe_1). The triggering load that induces buckling is the internal pressure on the pipe (see e.g. [1]). For that, it is necessary to introduce, like on the previous tutorial, an imperfection on the structure. This is again done by introducing a transversal force in the middle-span. Therefore, present tutorial has the definition of two loads. First, on the middle-spam of the pipe, a lateral nodal load is applied. After, an internal pressure field is applied along the entire pipe length.

Input file step-by-step

Creation of nodes

A total of 101 nodes is created to establish a finite element mesh.

```
//Creation of nodes
Nodes 101
Node 1 0 0 0
Node 2 1 0 0
Node 3 2 0 0
Node 4 3 0 0
Node 5 4 0 0
Node 6 5 0 0
Node 7 6 0 0
Node 8 7 0 0
Node 9 8 0 0
Node 10 9 0 0
Node 11 10 0 0
Node 12 11 0 0
Node 13 12 0 0
```

Node	14	13	0	0
Node	15	14	0	0
...				
Node	90	89	0	0
Node	91	90	0	0
Node	92	91	0	0
Node	93	92	0	0
Node	94	93	0	0
Node	95	94	0	0
Node	96	95	0	0
Node	97	96	0	0
Node	98	97	0	0
Node	99	98	0	0
Node	100	99	0	0
Node	101	100	0	0

Creation of node sets

Three node sets are defined to establish the constraints and loads. Node sets 1 and 2 are associate with nodes located at the ends of the pipe. Node set 3 contains the node located at the middle of the pipe, where the imperfection load takes place.

//Creation of node sets
NodeSets 3
//First node
NodeSet 1 Nodes 1 List 1
//Last node
NodeSet 2 Nodes 1 List 101
//Middle node
NodeSet 3 Nodes 1 List 51

Creation of elements

A total of 50 elements is created. All the elements have the same cross section (identification number 1) and the same coordinate system (identification number 1). Each Pipe_1 element is defined by three nodes, which must be equally spaced and contained in a straight line.

//Creation of elements
Elements 50
Pipe_1 1 PipeSec 1 CS 1 Nodes 1 2 3
Pipe_1 2 PipeSec 1 CS 1 Nodes 3 4 5
Pipe_1 3 PipeSec 1 CS 1 Nodes 5 6 7
Pipe_1 4 PipeSec 1 CS 1 Nodes 7 8 9
Pipe_1 5 PipeSec 1 CS 1 Nodes 9 10 11
Pipe_1 6 PipeSec 1 CS 1 Nodes 11 12 13
Pipe_1 7 PipeSec 1 CS 1 Nodes 13 14 15
Pipe_1 8 PipeSec 1 CS 1 Nodes 15 16 17
Pipe_1 9 PipeSec 1 CS 1 Nodes 17 18 19
Pipe_1 10 PipeSec 1 CS 1 Nodes 19 20 21
Pipe_1 11 PipeSec 1 CS 1 Nodes 21 22 23
Pipe_1 12 PipeSec 1 CS 1 Nodes 23 24 25
Pipe_1 13 PipeSec 1 CS 1 Nodes 25 26 27
Pipe_1 14 PipeSec 1 CS 1 Nodes 27 28 29
Pipe_1 15 PipeSec 1 CS 1 Nodes 29 30 31
...
Pipe_1 40 PipeSec 1 CS 1 Nodes 79 80 81
Pipe_1 41 PipeSec 1 CS 1 Nodes 81 82 83
Pipe_1 42 PipeSec 1 CS 1 Nodes 83 84 85
Pipe_1 43 PipeSec 1 CS 1 Nodes 85 86 87
Pipe_1 44 PipeSec 1 CS 1 Nodes 87 88 89
Pipe_1 45 PipeSec 1 CS 1 Nodes 89 90 91
Pipe_1 46 PipeSec 1 CS 1 Nodes 91 92 93
Pipe_1 47 PipeSec 1 CS 1 Nodes 93 94 95

Pipe_1 48	PipeSec	1	CS	1	Nodes	95	96	97
Pipe_1 49	PipeSec	1	CS	1	Nodes	97	98	99
Pipe_1 50	PipeSec	1	CS	1	Nodes	99	100	101

Creation of element set

One element set is defined, which contains all the elements created previously. It will be used for the creation of the internal pressure load field.

```
//Creation of element sets
ElementSets 1
ElementSet 1 Elements      50      Sequence      Initial      1
               Increment 1
```

Creation of cross-section

One only cross section is created, which is shown in the Figure 4-1(b).

```
//Creation of pipe sections
PipeSections 1
PS    1     EA    5984734005.0 EI    301817616.8 GJ    301817616.8 GA
      2992367003   Rho    239.3893602 CDt    0     CDn    0     CAT    0
      CAn    0     De    0.65   Di    0.62
```

Creation of coordinate systems

Two coordinate systems are created. The first one is used to define the element properties. Note that, since employing Pipe_1 element, direction E3 must lie in the pipe alignment direction (exactly as previously discussed for Beam_1 element). The second coordinate system is used to define the direction of application of the nodal loads.

```
//Creation of coordinate systems
CoordinateSystems 2
CS    1     E1    0     1     0     E3    1     0     0
CS    2     E1    1     0     0     E3    0     0     1
```

Creation of solution steps

This tutorial has two solution steps (both are statics). The first step is created establishing the end time equal to "1". The time step is "1", which means that the load (imperfection force) is applied in a single step. The second step is created establishing the end time equal to "2". The time step used is "0.005", that is, the internal pressure is divided into increments of 0.5%.

```
//Creation of solution steps
SolutionSteps2
//Step 1 - Static - introduction of the perturbation load. Time: 0-1
Static 1    EndTime      1      TimeStep      1.0      MaxTimeStep 1.0
          MinTimeStep 0.01  MaxIt 15      MinIt 3      ConvIncrease 2
          IncFactor 1.0    Sample 1
//Step 2 - Static - introduction of the follower load. Time: 1-2
Static 2    EndTime      2      TimeStep      0.005  MaxTimeStep 1
          MinTimeStep 0.002 MaxIt 15      MinIt 3      ConvIncrease 2
          IncFactor 1.0    Sample 1
```

Creation of loads

A NodalLoad and an internal pressure element load (PipeLoad) are defined. Both are done establishing a time-series. It is important to remember that the first load is the imperfection force, used to make feasible the evaluation of buckling. The second load is the internal pressure, which causes the buckling phenomenon studied. Both are interpreted by Giraffe as ramp-loads, linear interpolations between provided values in time-series tables.

```
//Creation of loads
Loads 2
//Perturbation load in the middle of the pipe
NodalLoad    1      NodeSet      3      CS      2      NTimes 2
//Time FX FY FZ MX MY MZ
0      0      0      0      0      0
1      0      1000     0      0      0
//Internal pressure load along the whole pipe
PipeLoad    2      ElementSet   1      NTimes 2
//Time P0I    P0E    RhoI    RhoE
1      0      0      0
2      1200000    0      0      0
```

Creation of constraints

Two nodal constraints are created. The first is done by fixing DOFs: UX, UY, UZ, ROTX on the node set 1. The second constraint is done by fixing DOFs: UY and UZ on the node set 2.

```
//Creation of constraints
Constraints 2
NodalConstraint 1      NodeSet      1
    UX      BoolTable      1
    UY      BoolTable      1
    UZ      BoolTable      1
    ROTX   BoolTable      1
NodalConstraint 2      NodeSet      2
    UY      BoolTable      1
    UZ      BoolTable      1
```

Creation of solver options

Here it is possible to choose the number of processors (threads) to be used (parallel processing). Besides, we must define the solver type for systems of linear equations. In this case, four processors are employed and the direct type of linear system solver is our choice.

```
//Creation of solver options
SolverOptions
Processors 4      LinSys Direct
```

Creation of monitors

We request Giraffe to monitor 3 nodes during the simulation: numbers 1 (point A), 51 (middle-spam) and 101 (point B).

Monitor	Sample 1
MonitorNodes	1 101 51

Creation of post files

The options used here are the same as the tutorial 01, where the WriteMesh, WriteRenderMesh and WriteForces are chosen for post-processing using Paraview™ and the MagFactor parameters assume a unit value so the deformed shape of Paraview™ frames will reflect the real scale.

```
PostFiles
MagFactor      1
WriteMesh       1
WriteRenderMesh 1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces    0
WriteForces     1
WriteConstraints 0
WriteSpecialConstraints 0
WriteContactForces 0
WriteRenderRigidBodies 0
WriteRenderParticles 0
```

Creation of convergence criteria

Here, we may define convergence criteria. This is an optional task, which may be done only by advanced users.

```
//Convergence criteria
ConvergenceCriteria
ForceTolerance      1.000000e-004
MomentTolerance    1.000000e-004
ForceMinimumReference 1.000000e-001
MomentMinimumReference 1.000000e-001
ConstraintMinimumReference 1.000000e-007
DisplacementTolerance 1.000000e-004
RotationTolerance 1.000000e-004
LagrangeTolerance 1.000000e-004
DisplacementMinimumReference 1.000000e-006
RotationMinimumReference 1.000000e-006
LagrangeMinimumReference 1.000000e-006
DivergenceReference 1.000000e+015
```

Results and post-processing

To post-process results using Paraview™, first, we must open the files “whole_solution_forces.pvd” and “whole_solution_mesh.pvd”, which are located at the folder “post” in “tutorial04”.

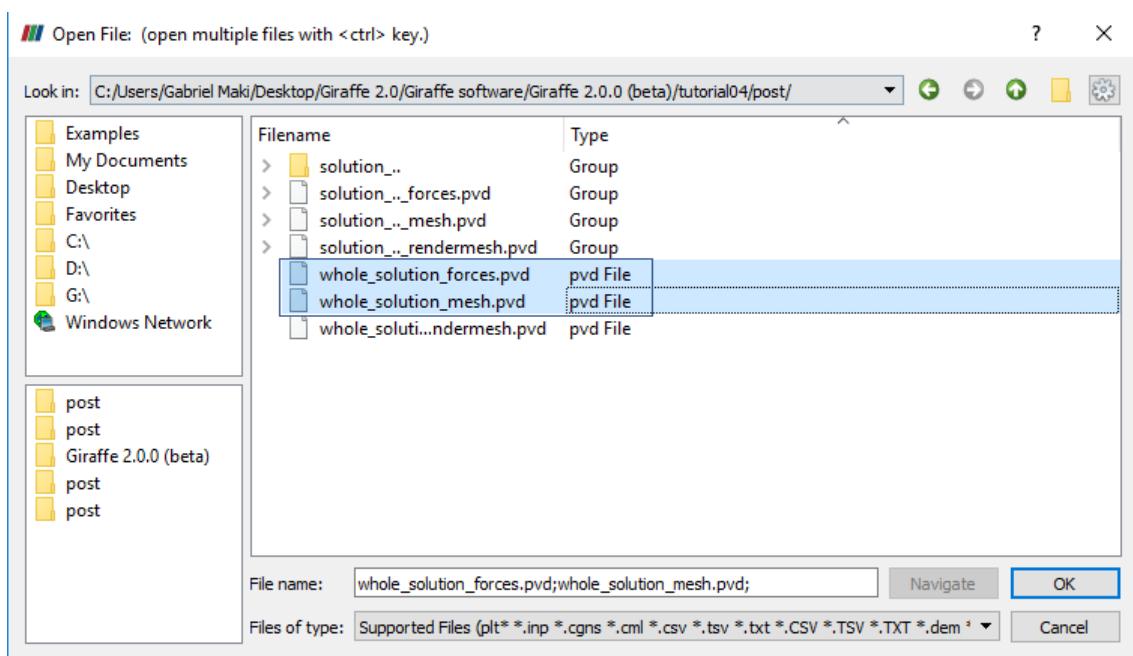


Figure 4-2 – Opening the files in Paraview™

Then, we must click the preview icon and the Apply button on the Pipeline browser.

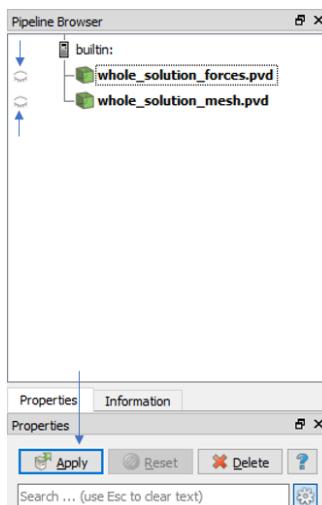


Figure 4-3 – Pipeline Browser.

Figure 4-4 shows how Paraview™ displays the results.



Figure 4-4 – Results using Paraview™

Next, we can create a Glyph for observing the load being applied. For that, go to the Pipeline Browser and click in “whole_solution_forces.pvd”. Then, click the Glyph, as shown in Figure 4-5.

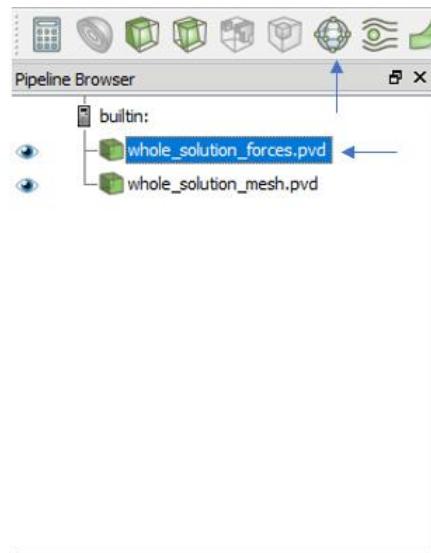


Figure 4-5 – Creation of a Glyph.

After creating the Glyph, we can adjust the scale that rule the size of rendered view of vectors. For that, go to the Pipeline Browser and, first, click the preview icon of Glyph. Then, in Scaling, set the scale mode to vector and change the scale factor to a small number like 0.01. Finally, click Apply.

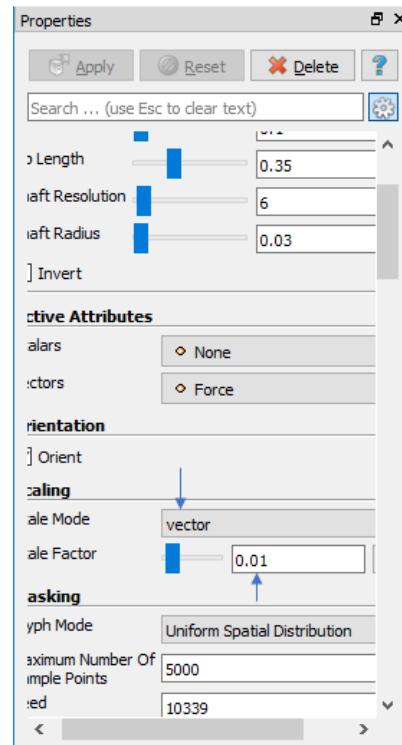


Figure 4-6 – Adjusting the Glyph.

Finally, we can perform an animation to observe how the structure deforms along buckling triggered by internal pressure. When needing to plot time series of displacements, forces or other kinematical quantities, monitors may be helpful.

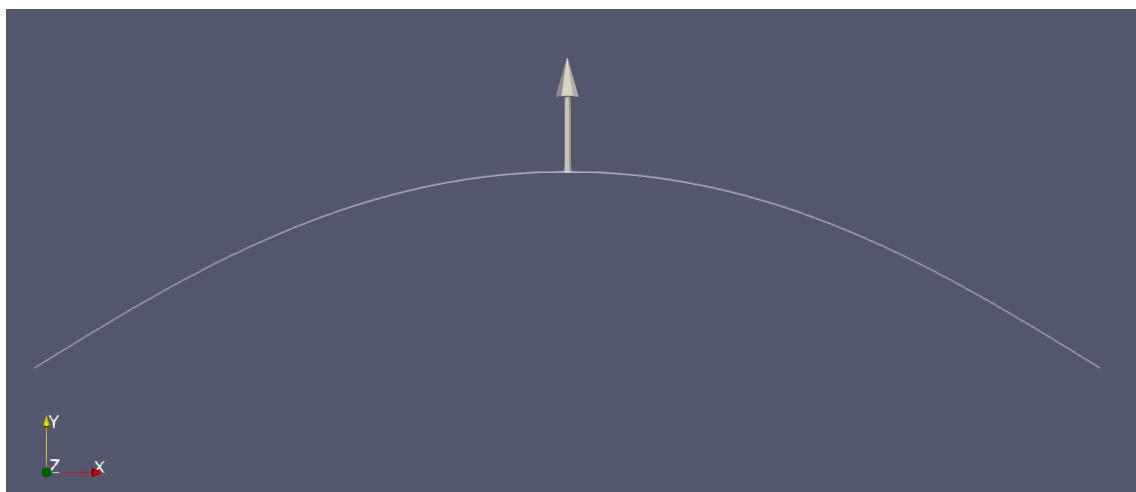


Figure 4-7 – Deformed shape of the pipe at the end of simulation.

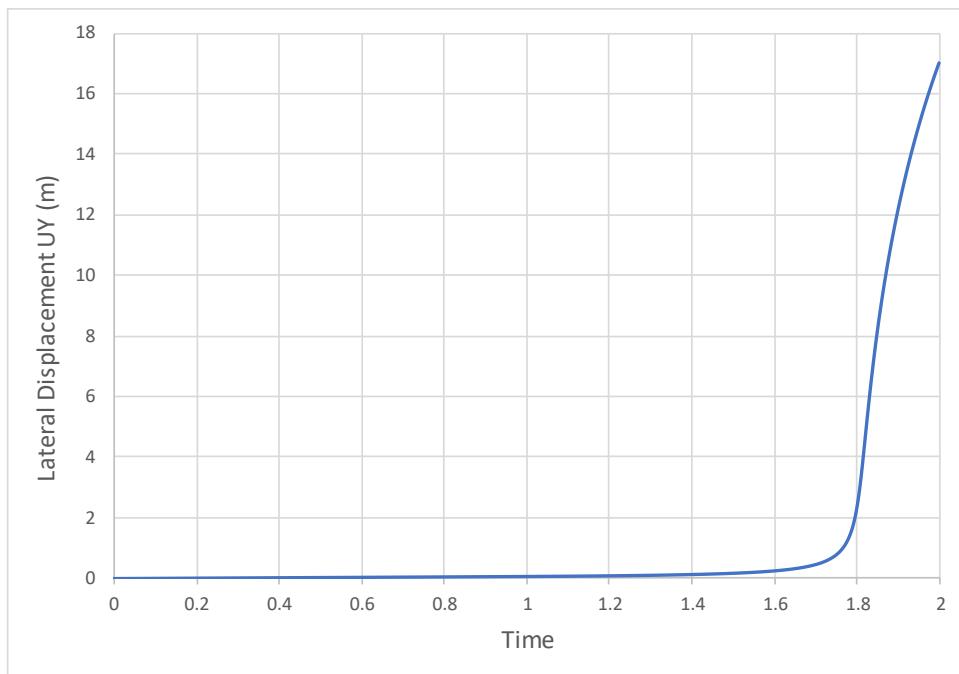


Figure 4-8 – Lateral displacement (UY) of the pipe middle spam vs. time.

Tutorial 5. Static and Dynamic analysis of a cantilever plate

Input file name: tutorial05.inp

Developed by Alfredo Gay Neto & Gabriel Maki

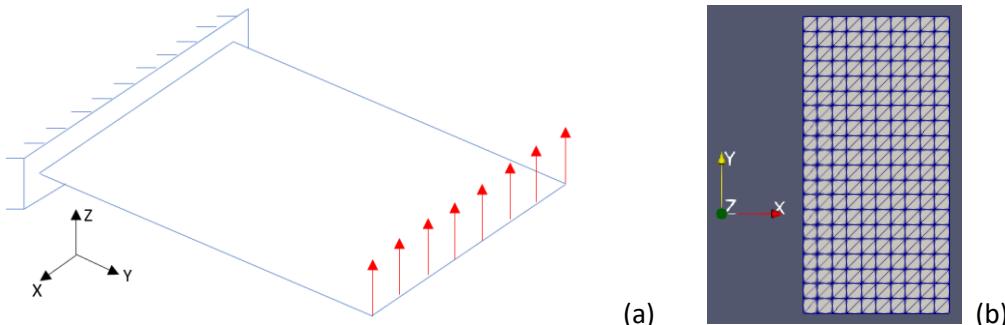


Figure 5-1 - (a) Model of a cantilever plate (b) Mesh used for modeling

Objective

This tutorial presents a geometrically-nonlinear simulation of a cantilever plate, subjected to a uniformly distributed load along a line and clamped in the opposite side, as shown in Figure 5-1(a). The tutorial has the purpose of showing how to solve an analysis that has two steps, one static and one dynamic.

Description

The plate is 0.395 m long and 0.195 m wide. It is meshed considering 861 nodes and 400 elements, using the element Shell_1. There is a clamp boundary condition on one of the edges of the plate. The section has a thickness of 2 mm. Besides, the simulation is solved in two steps. In the first, the load is applied on an edge of the plate. This deforms the whole structure, inducing bending. In the second step, which is dynamic, the nodal load is removed. It is possible to observe how the plate oscillates after dynamically removing the load.

The main difference between static and dynamic solution steps is that dynamics considers not only the internal and external loads on the structure, but also inertial and damping loads. This may be useful for determining a time-series response of a given structure to dynamic phenomena, such as time-varying forces, impacts and other possibilities.

Input file step-by-step**Creation of nodes**

A total of 861 nodes are created to establish a finite element mesh.

```
//Creation of nodes
Nodes 861
Node 1      0      0      0
Node 2      0.0195 0      0
Node 3      0.039000001 0      0
Node 4      0.058499999 0      0
Node 5      0.078000002 0      0
Node 6      0.097499996 0      0
Node 7      0.116999999 0      0
Node 8      0.136500001 0      0
Node 9      0.156000003 0      0
```

Node	10	0.175500005	0	0
Node	11	0.194999993	0	0
...				
Node	850	0.136500001	0.385125011	0
Node	851	0.126749992	0.385125011	0
Node	852	0.126749992	0.395000011	0
Node	853	0.156000003	0.385125011	0
Node	854	0.14625001	0.385125011	0
Node	855	0.14625001	0.395000011	0
Node	856	0.175500005	0.385125011	0
Node	857	0.165749997	0.385125011	0
Node	858	0.165749997	0.395000011	0
Node	859	0.194999993	0.385125011	0
Node	860	0.185249999	0.385125011	0
Node	861	0.185249999	0.395000011	0

Creation of node sets

Two node sets are defined to establish the constraints and the load on the structure. The node sets are represented in Figure 5-2 by red lines. The nodes of node set 1 belong to the clamped line.

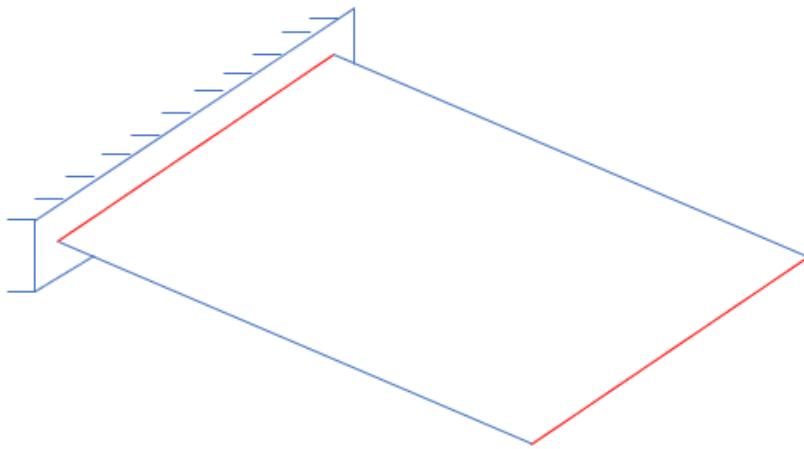


Figure 5-2 - Representation of node sets on the plate.

//Creation of node sets										
NodeSets	2									
//Left tip										
NodeSet	1	Nodes	21	List	1	2	3	4	5	6
7	8	9	10	11	232	237	241	245	249	253
257	261	265	269							
//Right tip										
NodeSet	2	Nodes	21	List	221	222	223	224	225	226
227	228	229	230	231	833	837	840	843	846	849
852	855	858	861							

Creation of elements

A total of 400 elements is created. All the elements have the same material properties and the same shell section.

```
//Creation of elements
//Elements - in this case only Shell_1 elements are created
//Important: the 6 nodes of each shell element MUST lie in a plane at reference
configuration (input file)
//After deformation, shells may deform to out-of-plane configurations.
//Each element has attributes to material (Mat) and section (Sec).
Elements      400
Shell_1       1      Mat    1 Sec   1 Nodes  1     2     13    232    233    234
Shell_1       2      Mat    1 Sec   1 Nodes 13    12     1    235    236    234
Shell_1       3      Mat    1 Sec   1 Nodes  2     3     14    237    238    239
Shell_1       4      Mat    1 Sec   1 Nodes 14    13     2    240    233    239
Shell_1       5      Mat    1 Sec   1 Nodes  3     4     15    241    242    243
...
Shell_1      395      Mat    1 Sec   1 Nodes 217   218    229    824    853    854
Shell_1      396      Mat    1 Sec   1 Nodes 229   228    217    855    850    854
Shell_1      397      Mat    1 Sec   1 Nodes 218   219    230    827    856    857
Shell_1      398      Mat    1 Sec   1 Nodes 230   229    218    858    853    857
Shell_1      399      Mat    1 Sec   1 Nodes 219   220    231    830    859    860
Shell_1      400      Mat    1 Sec   1 Nodes 231   230    219    861    856    860
```

Creation of Materials

One only material model is defined, which identification number is 1.

```
//Creation of Materials
Materials      1
Hooke 1       E      200e9  Nu      0.3   Rho     8000
```

Creation of shell sections

One shell section is defined, homogeneous and with 2 mm thickness. Its identification number is 1.

```
//Creation of ShellSections
ShellSections1
Homogeneous 1      Thickness     0.002
```

Creation of Coordinate Systems

One only coordinate system is created, which is used to define the direction of the nodal load.

```
//Creation of Coordinate Systems
CoordinateSystems 1
CS      1      E1      1      0      0      E3      0      0      1
```

Creation of solution steps

This tutorial has two solutions steps. The first is static and is created establishing the end time equal to "1". The time step is set to "0.1", which means that the load is divided into increments of 10% along the first solution step. The second solution step is dynamic and is created establishing the end time equal to "1.4". Thus, it encompasses simulation from time 1 to 1.4 (from the end of previous solution step until the end time defined for it). In a dynamic analysis, the time step is the physical time, unlike in the static analysis, where time is a tracking scalar parameter, serving only as a guide to define varying loads along the simulation evolution.

```

SolutionSteps2
//First step: statics - from time 0 - 1
Static 1
EndTime      1.0
TimeStep     0.1
MaxTimeStep  0.1
MinTimeStep  0.01
MaxIt       20
MinIt       3
ConvIncrease 4
IncFactor    1.0
Sample 1
//Second step: dynamics - from time 1 - 1.4
Dynamic      2
EndTime      1.4
TimeStep     0.005
MaxTimeStep  0.005
MinTimeStep  0.001
MaxIt       20
MinIt       3
ConvIncrease 5
IncFactor    1.2
Sample 1
RayleighDamping   Alpha 0      Beta 0      Update 0
NewmarkCoefficients Beta 0.3   Gamma 0.5

```

Note that here we set MinTimeStep as 0.001. This permits Giraffe to automatically decrease the time-step in case of difficulties of convergence along the simulation.

The last two lines of the definition of a dynamic solution step refers to damping and time-integration scheme setting. Rayleigh damping permits the user to set Alpha and Beta (multipliers of mass and stiffness matrices of a Rayleigh damping model). Also, it is possible to Update or not such matrices at the beginning of each time-step. In case of the option Update 0, the reference configuration matrices are considered for damping evaluation along the whole simulation. In present example no damping is considered, since both Alpha and Beta parameters are set zero.

Lastly, it is important to mention that Giraffe employs an implicit time-integration scheme of equations of motion. Newmark method is used, in a special version of it, when dealing with rotations (see e.g.: [2]). The values of Beta and Gamma are coefficients to rule the time-integration. The values Beta 0.3 and Gamma 0.5 are recommended to avoid numerical damping.

Creation of loads

A single nodal load is defined. This is done establishing a time varying force in direction Z. Note that, after 1 s, the nodal load is removed very quickly (in 0.0001 s). Giraffe employs linear interpolation between the input tabled values of loads along time.

```

//Creation of loads
Loads 1
NodalLoad   1      NodeSet    2      CS      1      NTimes 3
//Time FX    FY     FZ      MX      MY      MZ
0.00  0      0      0      0      0
1.00  0      0      1000   0      0      0
1.0001 0    0      0      0      0

```

Creation of constraints

A single nodal constraint is created. This is done fixing all the degrees of freedom, representing a clamp boundary condition at node set 1.

```
//Creation of constraints
Constraints 1
//Clamping the left tip
NodalConstraint 1      NodeSet      1
    UX    BoolTable   1
    UY    BoolTable   1
    UZ    BoolTable   1
    ROTX  BoolTable   1
    ROTY  BoolTable   1
    ROTZ  BoolTable   1
```

Creation of solver options

Here it is possible to choose the number of processors used (parallel processing). Besides, we must define the solver type for linear equation systems. In this case, four processors are employed and the direct type of linear system solver is our choice.

```
//Creation of solver options
SolverOptions
Processors 4      LinSys Direct
```

Creation of monitors

We request Giraffe to monitor the two node sets during the simulation. This is useful for further post-processing.

```
//Creation of monitors
Monitor      Sample 1
MonitorNodeSets 1      2
```

Creation of post-files

The options used here are the same as the tutorial 01, where the WriteMesh, WriteRenderMesh and WriteForces are chosen for post-processing using Paraview™ and the MagFactor parameters assume a unit value so the deformed shape of Paraview™ frames will reflect the real scale.

```
//Creation of post files
PostFiles
MagFactor     1
WriteMesh      1
WriteRenderMesh 1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces 0
WriteForces    1
WriteConstraints 0
WriteSpecialConstraints 0
WriteContactForces 0
WriteRenderRigidBodies 0
WriteRenderParticles 0
```

Results and post-processing

To view the post-processing in Paraview™, first we must open the files “whole_solution_forces.pvd” and “whole_solution_mesh.pvd”, which are located at the folder “post” in “tutorial05”.

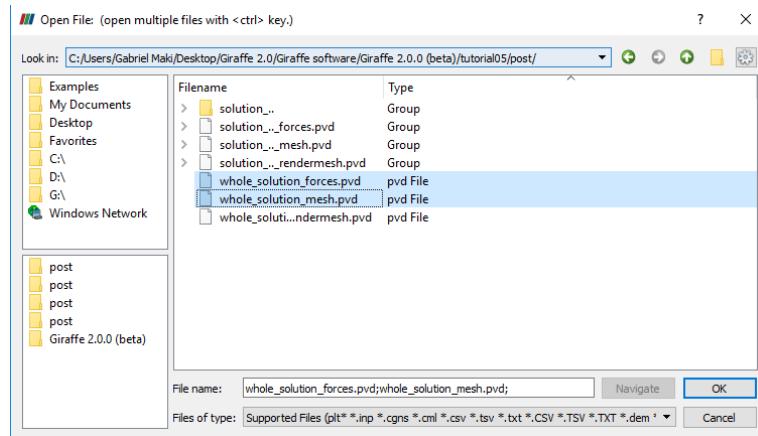


Figure 5-3 - Opening the files of tutorial 05 in Paraview™.

Then, we must click on the preview icon and the Apply button on the Pipeline browser.

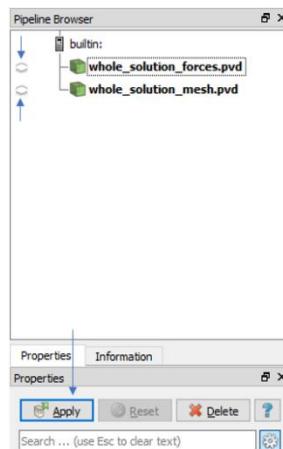


Figure 5-4 - Where to click in Pipeline Browser.

Figure 5-5 shows how Paraview™ displays the results.

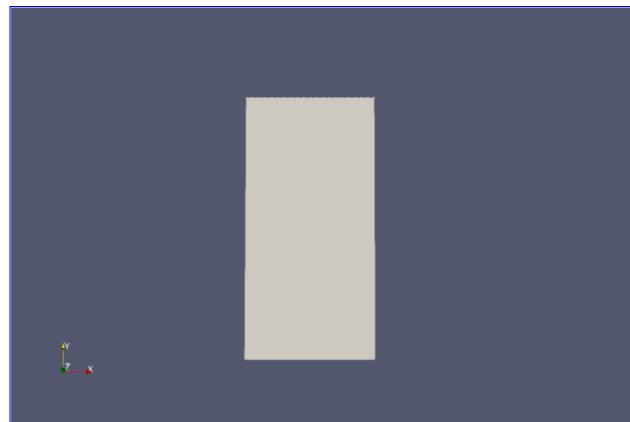


Figure 5-5 - Results using Paraview™.

It is possible to change the view by using the tools of “Set view”. Figure 5-6 depicts the icons that may be used for that.

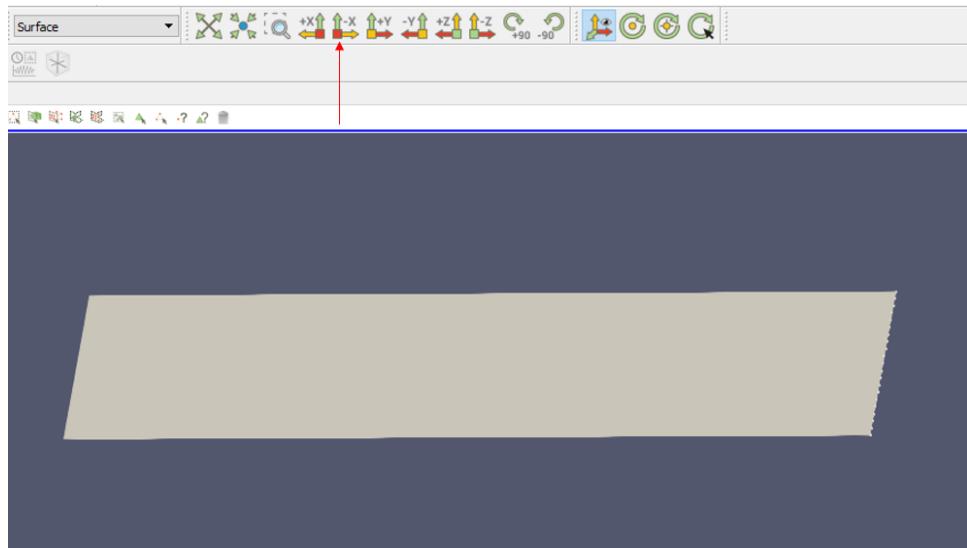


Figure 5-6 - Changing the view in Paraview™.

In addition, to changing the view, it is possible to establish an isometric perspective, instead of a conic one (default in Paraview). For that, we must click on any of Pipeline browser items, go to Properties and check “Camera Parallel Projection in Orientation Axes” option (Figure 5-7).

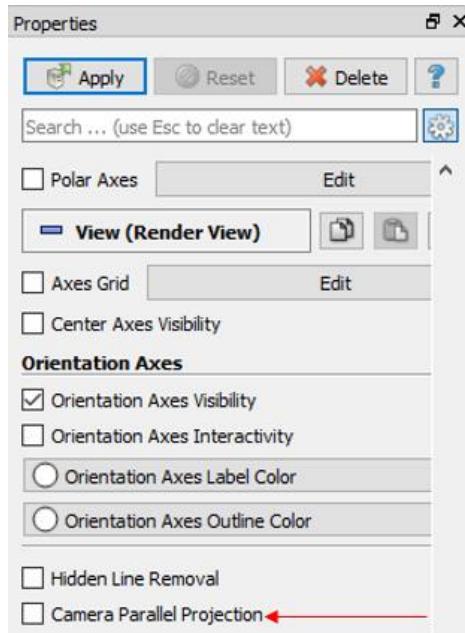


Figure 5-7 - Establishing isometric perspective view.

Next, we can create a Glyph for visualization of the load being applied. For that, go to the Pipeline Browser and click in “whole_solution_forces.pvd”. Then, click the Glyph. The Figure 5-8 shows where to click.

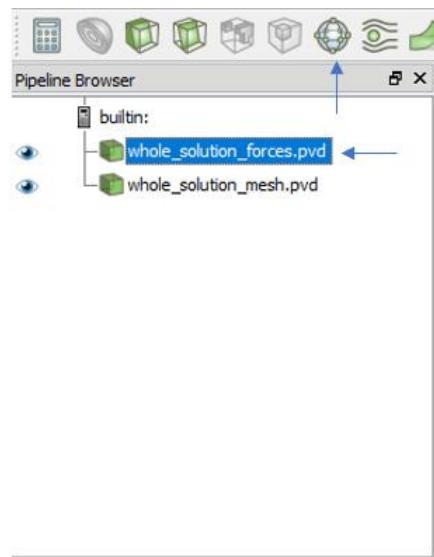


Figure 5-8 - Creating a Glyph.

After creating the Glyph, we have to adjust the scale of vectors. For that, go to the Pipeline Browser and, first, click the preview icon of Glyph. Then, in Scaling, set the scale mode to vector and change the scale factor to a small number like 0.001. Finally, click Apply.

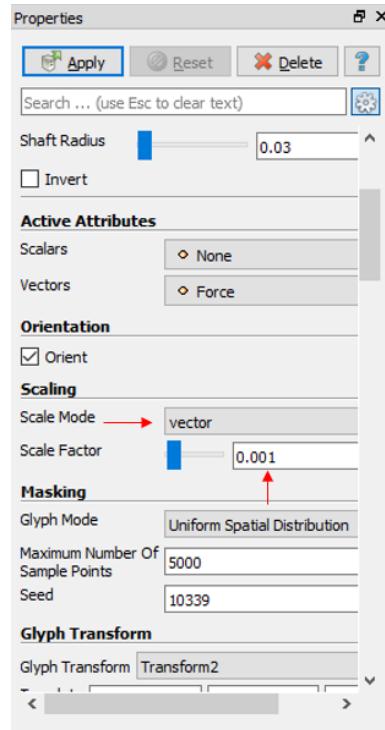


Figure 5-9 - Adjusting the Scaling in Glyph.

Finally, we can perform an animation to observe how the structure deforms along time. Note that this post-processing may be done individually for each solution step, just opening in Paraview, instead of the whole solution results, the desired step results.

When needing to plot time series of displacements, forces or other kinematical quantities, monitors may be helpful.

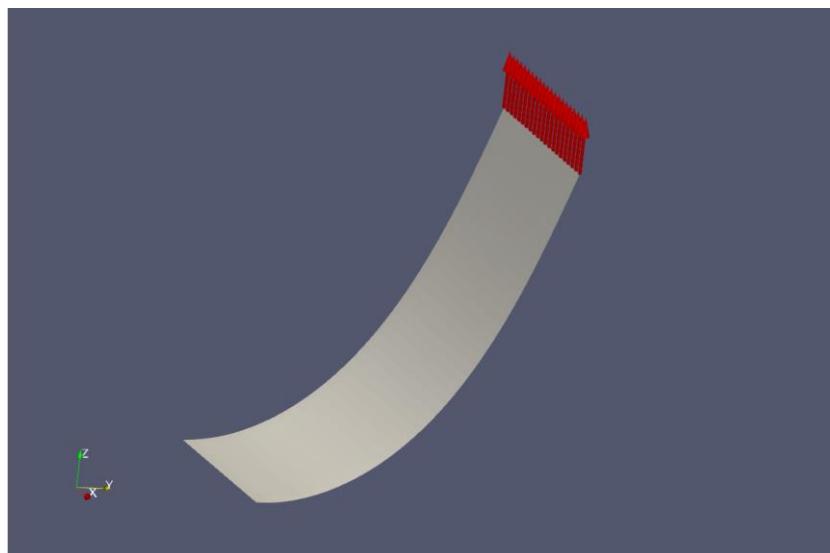


Figure 5-10 - Deformed shape of structure after first solution step.

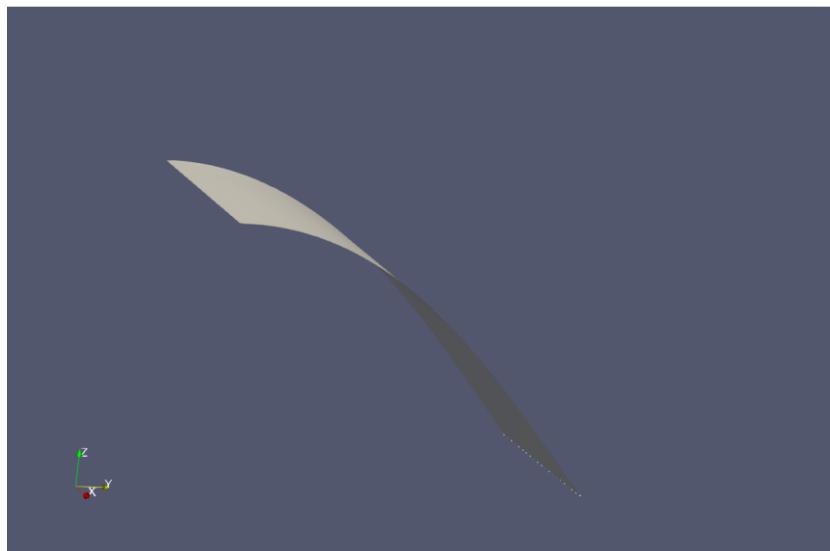


Figure 5-11 – Deformed structure at the end of the simulation.

Next plots refer to node set monitors post-processing. One may find kinematic and load quantities in such monitors.

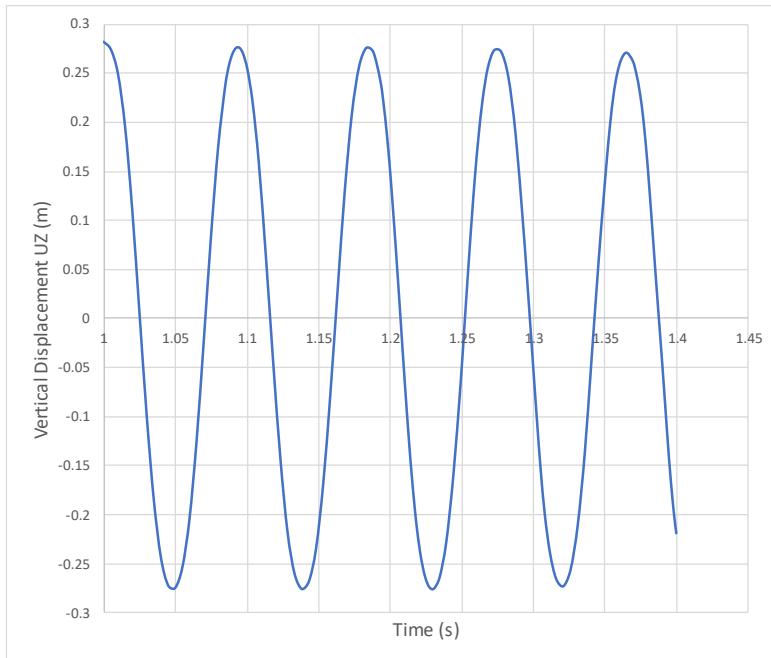


Figure 5-12 - Vertical Displacement (UZ) of the plate tip vs. time.

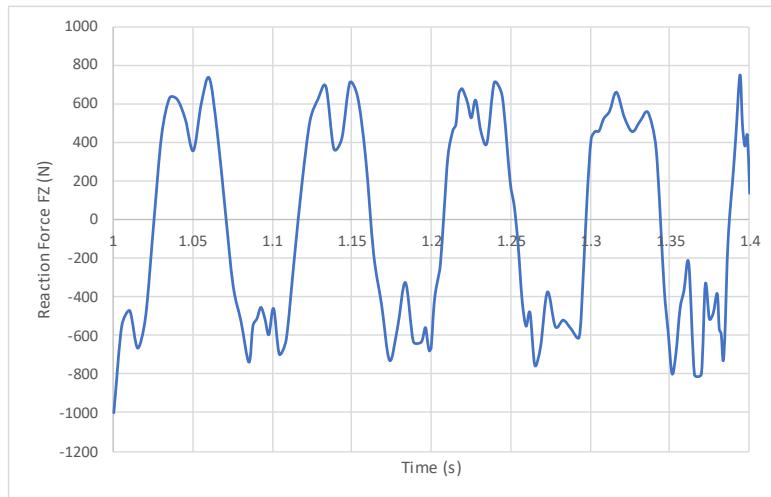


Figure 5-13 - Vertical Reaction Force FZ at the clamp vs. time.

Figure 5-12 shows the oscillatory movement that the structure presents. It is only possible to observe such behavior because the second solution step considers dynamic effects. Furthermore, Figure 5-13 shows the reaction on the plate root vs. time, which is also a useful result.

If we change the second solution step to static, we will see that the plate returns to its initial configuration with no oscillations. To do so, we have to change the second solution step type to "static". Furthermore, it is convenient to decrease the time-step of the simulation to provide Giraffe the possibility of creating a smoother path between the bended plate configuration (the end of the first solution step) and the expected end configuration of the second solution step, corresponding to the plate stress-free configuration. After making that changes and post-processing the model in Paraview™, note that the initial and final configurations are the exactly the same.

Tutorial 6. Static and Dynamic analysis of a cantilever plate with a rigid edge

Input file name: tutorial06.inp

Developed by Alfredo Gay Neto & Gabriel Maki

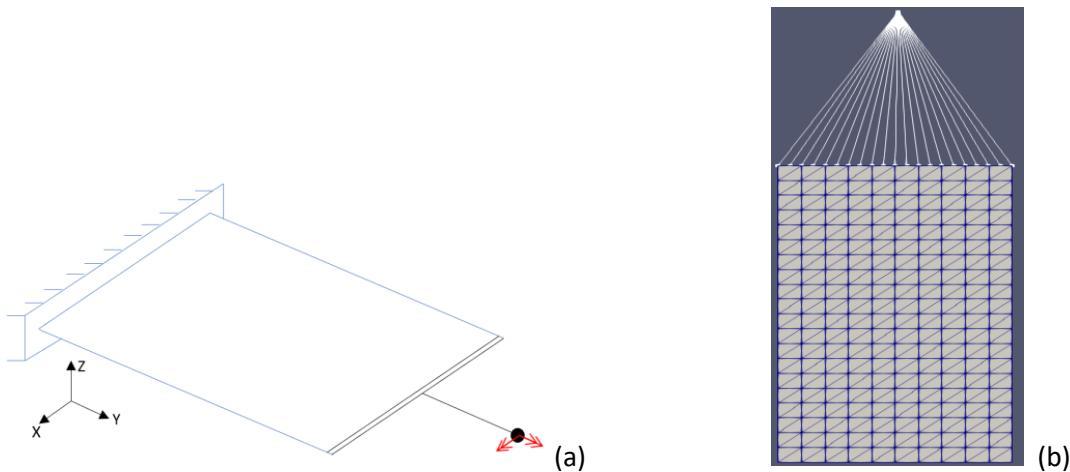


Figure 6-1 - (a) Model of cantilever plate rigidly linked to a node (b) Mesh used for modeling with the rigid node set represented

Objective

The objective of this tutorial is to introduce a new resource for modeling called special constraint. In this case, the structure is exactly the same plate simulated in last tutorial. However, now we establish a rigid region along the edge of the plate by means of a rigid node set, a special constraint available in Giraffe.

Description

The plate size is 0.395 m long and 0.195 m wide. It is meshed considering 862 nodes (one of which is called pilot node to use the special constraint) and 400 elements (Shell_1). There is a clamp boundary condition on one of the edges of the plate. The section has a thickness of 2 mm. As in previous example, the simulation runs in two steps. During the first, a nodal load is applied on the pilot node, which deforms the structure. During the second step, which is dynamic, the load is kept for 0.25 s, then it starts to reduce until zero along more 0.25 s.

Input file step-by-step

Creation of nodes

A total of 862 nodes is created to establish a finite element mesh. Like mentioned before, there is one node that is called pilot node and it is used to define the special constraint.

```
//Creation of nodes
Nodes 862
Node 1 0 0 0
```

```

Node 2    0.0195 0    0
Node 3    0.039000001 0    0
Node 4    0.058499999 0    0
Node 5    0.078000002 0    0
Node 6    0.097499996 0    0
Node 7    0.116999999 0    0
Node 8    0.136500001 0    0
...
Node 855   0.14625001 0.395000011 0
Node 856   0.175500005 0.385125011 0
Node 857   0.165749997 0.385125011 0
Node 858   0.165749997 0.395000011 0
Node 859   0.194999993 0.385125011 0
Node 860   0.185249999 0.385125011 0
Node 861   0.185249999 0.395000011 0
//Node 862 - will be the pilot node of the rigid node set
Node 862   0.1    0.6    0.0

```

Creation of node sets

Three node sets are created to establish the constraints, special constraints and the load applied in the pilot node. The node sets 1 and 2 are the same as the previously tutorial and the node set 3 contains only the pilot node.

```

//Creation of node sets
NodeSets 3
//Left tip
NodeSet 1 Nodes 21 List 1 2 3 4 5 6
7 8 9 10 11 232 237 241 245 249 253
257 261 265 269
//Right tip
NodeSet 2 Nodes 21 List 221 222 223 224 225 226
227 228 229 230 231 833 837 840 843 846 849
852 855 858 861
//Pilot node
NodeSet 3 Nodes 1 List 862

```

Creation of elements

400 elements are created to compose the mesh. All the elements have the same material properties and the same shell section.

```

//Creation of elements
//Elements - in this case only Shell_1 elements are created
//Important: the 6 nodes of each shell element MUST lie in a plane at reference
configuration (input file)
//After deformation, shells may deform to out-of-plane configurations.
//Each element has attributes to material (Mat) and section (Sec).
Elements 400
Shell_1 1 Mat 1 Sec 1 Nodes 1 2 13 232 233
234
Shell_1 2 Mat 1 Sec 1 Nodes 13 12 1 235 236
234
Shell_1 3 Mat 1 Sec 1 Nodes 2 3 14 237 238
239
Shell_1 4 Mat 1 Sec 1 Nodes 14 13 2 240 233
239
Shell_1 5 Mat 1 Sec 1 Nodes 3 4 15 241 242
243
...
Shell_1 395 Mat 1 Sec 1 Nodes 217 218 229 824 853
854

```

Shell_1	396	Mat	1	Sec	1	Nodes	229	228	217	855	850
	854										
Shell_1	397	Mat	1	Sec	1	Nodes	218	219	230	827	856
	857										
Shell_1	398	Mat	1	Sec	1	Nodes	230	229	218	858	853
	857										
Shell_1	399	Mat	1	Sec	1	Nodes	219	220	231	830	859
	860										
Shell_1	400	Mat	1	Sec	1	Nodes	231	230	219	861	856
	860										

Creation of materials

One only material model is defined, which identification number is 1.

```
//Creation of Materials
Materials      1
Hooke 1       E        200e9  Nu      0.3   Rho     8000
```

Creation of shell sections

One shell section is defined, which has a thickness of 2 mm and is homogeneous. Its identification number is 1.

```
//Creation of ShellSections
ShellSections1
Homogeneous 1      Thickness    0.002
```

Creation of coordinate systems

One only coordinate system is created, which is used to define the direction of nodal load.

```
//Creation of Coordinate Systems
CoordinateSystems 1
CS      1      E1      1      0      0      E3      0      0      1
```

Creation of solution steps

This tutorial has two solutions steps. The first is static and is created establishing the end time equal to "1". The time step is "0.05", which means that the load is divided into increments of 5%. The second step is dynamic and is created establishing the end time equal to "2.5".

```
//Creation of the solution steps
SolutionSteps2
//First step: statics - from time 0 - 1
Static 1
EndTime      1.0
TimeStep     0.05
MaxTimeStep  0.05
MinTimeStep  0.01
MaxIt       20
MinIt       3
ConvIncrease 4
IncFactor    1.0
Sample 1
//Second step: dynamics - from time 1 - 2.5
Dynamic      2
EndTime      2.5
TimeStep     0.01
MaxTimeStep  0.01
MinTimeStep  0.0001
MaxIt       20
```

```

MinIt 3
ConvIncrease 5
IncFactor 1.2
Sample 1
RayleighDamping Alpha 0 Beta 0 Update 0
NewmarkCoefficients Beta 0.3 Gamma 0.5

```

Creation of loads

A single nodal load is defined. This is done establishing a time varying moment in direction X and Y. Note that, after 1.25 s, the moments start to reduce until be totally removed at 1.5 s. After that time, even if the user does not input data in the time-series table, Giraffe considers the last established value kept.

```

//Creation of loads
Loads 1
NodalLoad 1 NodeSet 3 CS 1 NTimes 4
//Time FX FY FZ MX MY MZ
0.00 0 0 0 0 0 0
1.00 0 0 0 100 300 0
1.25 0 0 0 100 300 0
1.50 0 0 0 0 0 0

```

Creation of constraints

A single nodal constraint is created. This is done fixing all the degrees of freedom, representing a clamp boundary condition on node set 1.

```

//Creation of constraints
Constraints 1
//Clamping the left tip
NodalConstraint 1 NodeSet 1
UX BoolTable 1
UY BoolTable 1
UZ BoolTable 1
ROTX BoolTable 1
ROTY BoolTable 1
ROTZ BoolTable 1

```

Creation of special constraints

The special constraint used in this tutorial is the Rigid Node Set, which defines a rigid region on a specific node set, in other words, the nodes contained in that node set behaves kinematically as a group of material points within a rigid body. Besides, that node set is rigidly linked to the pilot node as if there were a rigid bar between them. Therefore, the loads applied on the pilot node are transferred to the structure. It is important to mention that, when there is a single "1" on the BoolTable and there are more than one solution steps, it means that the special constraint is activated during all the remaining solution steps.

```

//Definition of a rigid region (rigid node set) on node set 2. This region is
ruled by the DOFs of the pilot node
SpecialConstraints 1
RigidNodeSet 1 PilotNode 862 NodeSet 2 BoolTable 1

```

Creation of solver options

Here it is possible to choose the number of processors used (parallel processing). Besides, we must define the solver type for linear equation systems. In this case, four processors are employed and the direct type of linear system solver is our choice.

```
//Creation of solver options
SolverOptions
Processors    4      LinSys Direct
```

Creation of monitors

We request Giraffe to monitor two node sets during the simulation.

```
//Creation of monitors
Monitor      Sample 1
MonitorNodeSets 1      2
```

Creation of post-files

The options used here are the same as the tutorial 01, where the WriteMesh, WriteRenderMesh and WriteForces are chosen for post-processing using Paraview™ and the MagFactor parameters assume a unit value so the deformed shape of Paraview™ frames will reflect the real scale. It is interesting to choose the “WriteSpecialConstraints” because it allows the user to view the Special Constraints in Paraview™.

```
//Creation of post files
PostFiles
MagFactor    1
WriteMesh    1
WriteRenderMesh    1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces 0
WriteForces    1
WriteConstraints 0
WriteSpecialConstraints    1
WriteContactForces 0
WriteRenderRigidBodies 0
WriteRenderParticles 0
```

Creation of convergence criteria

Here, we can define some convergence tolerances. This is an optional task (for advanced users only).

```
ConvergenceCriteria
ForceTolerance 1e-4
MomentTolerance 1e-4
ForceMinimumReference 1e-4
MomentMinimumReference 1e-4
ConstraintMinimumReference 1e-7
DisplacementTolerance 1e-4
RotationTolerance 1e-4
LagrangeTolerance 1e-4
DisplacementMinimumReference 1e-6
RotationMinimumReference 1e-6
```

LagrangeMinimumReference 1e-6
DivergenceReference 1e+15

Results and post-processing

To view the post-processing in Paraview™, first we must open the files “whole_solution_forces.pvd” and “whole_solution_mesh.pvd”, which are located at the folder “post” in “tutorial06”.

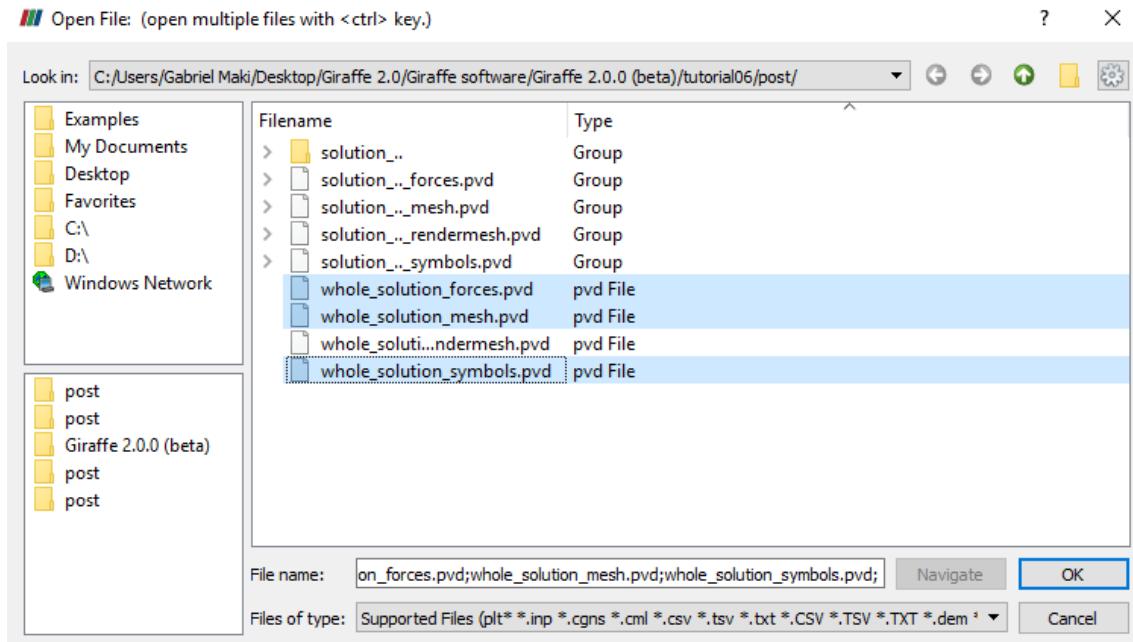


Figure 6-2 - Opening results of tutorial 6 in Paraview™.

Then, we must click the preview icon and the Apply button on the Pipeline browser.

For a better view, we must use the isometric perspective view. To do this, go to Properties and check “Camera Parallel Projection in Orientation Axes.”

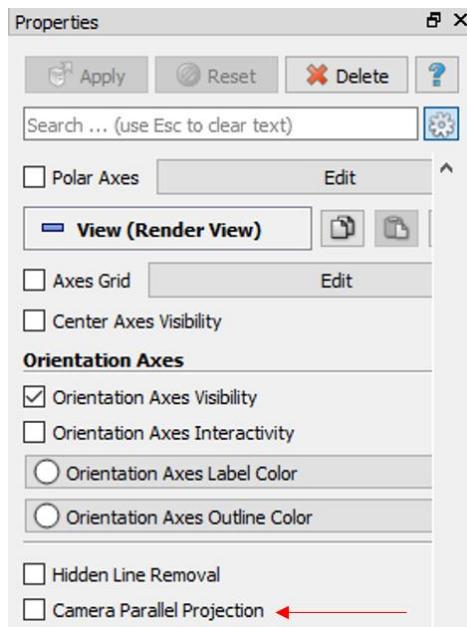


Figure 6-3 - Enabling isometric perspective view in Paraview™.

Next, we must create a Glyph for seeing the load being applied. For that, go to the Pipeline Browser and click in “whole_solution_forces.pvd”. Then, click the Glyph.

After creating the Glyph, we must adjust the scale of vectors. To do this, go to the Pipeline Browser and, first, click the preview icon of Glyph. Then, in Scaling, set the scale mode to vector and change the scale factor to a small number like 0.0005. Besides, we need to change, in Active Attributes, the Vectors to Moment. Finally, click Apply.

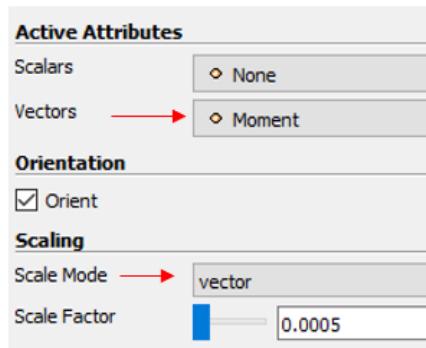


Figure 6-4 - Changing Properties of Glyph.

If the user wants to know the values of displacements of some nodes, node sets or the reaction forces on the clamp at a given instant, monitors resource may be employed.

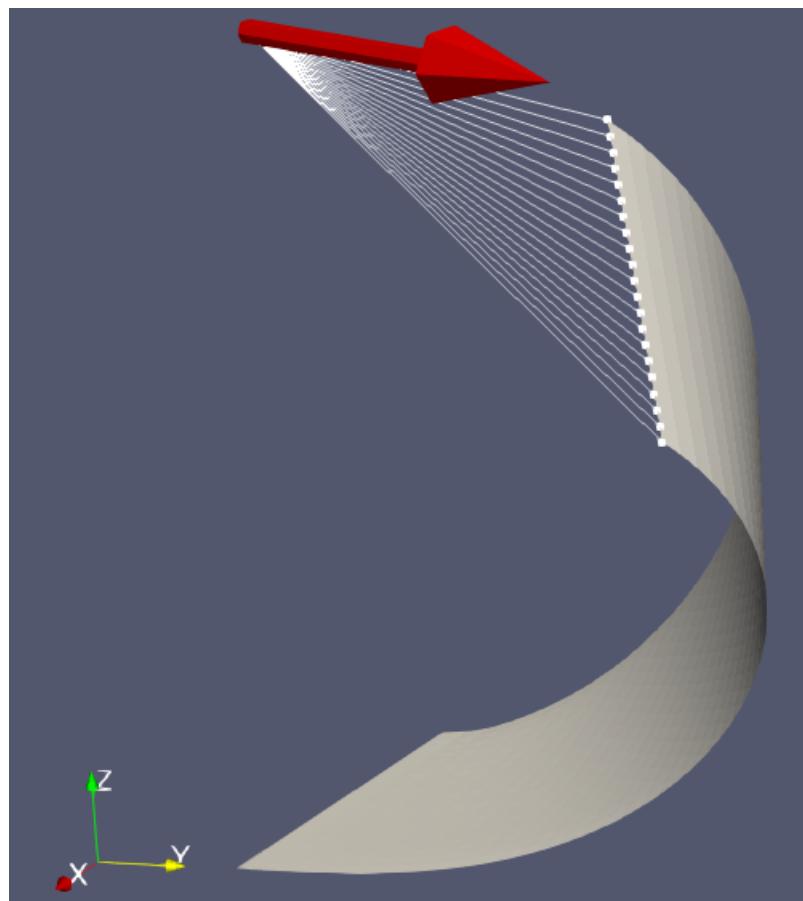


Figure 6-5 - Deformed shape of structure after the first solution step.

Note that the moment is not applied on the plate. It is applied on the pilot node. Besides, it is possible to see that the rigid edge does not deform because of the special constraint imposed which is represented by the white lines in Figure 6-5.

Obs: If the user wants to change the color of the arrow, the one can do it by clicking in Edit, on Coloring at Properties.

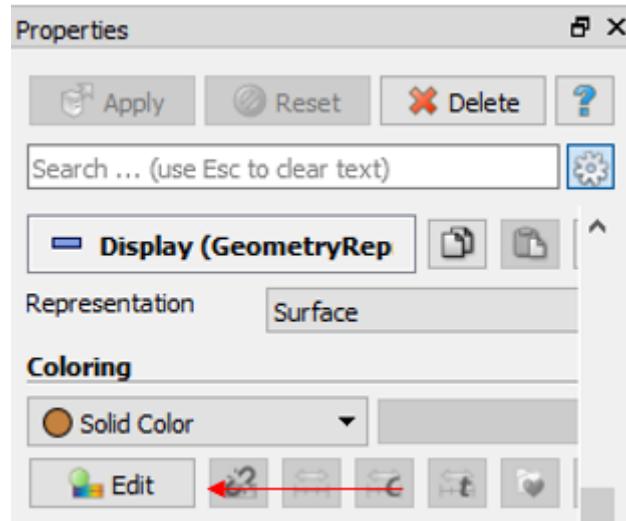


Figure 6-6 - Changing the color of arrows.

Besides, if the user wants to see the Special Constraint on a better way, it is possible to increase the point size. To do this, we have to go to the Properties of “whole_solution_symbols.pvd” and change the Point Size.



Figure 6-7 - Changing the Point Size.

By monitoring the vertical displacement of the structure tip (node set 2), the user can note that it does not move along the first 0.25 s of dynamic solution step because there is no change in the load. When the moments decrease their values, the structure starts to oscillate like observed in the last tutorial. It is possible to see such results in Figure 6-8 and Figure 6-9.

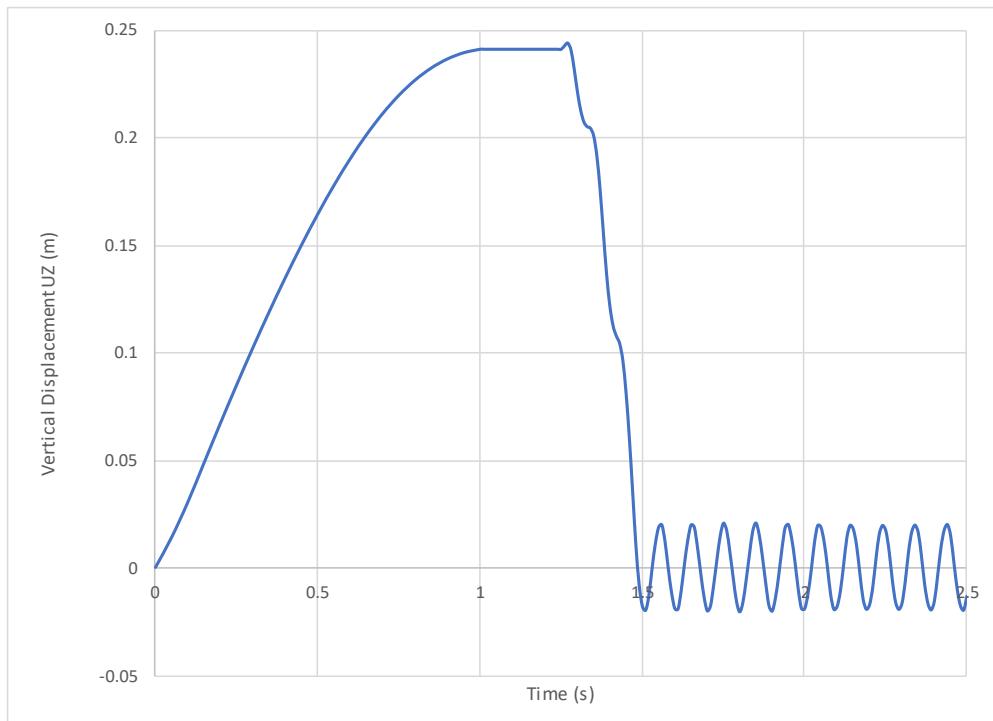


Figure 6-8 – Vertical Displacement UZ vs Time.

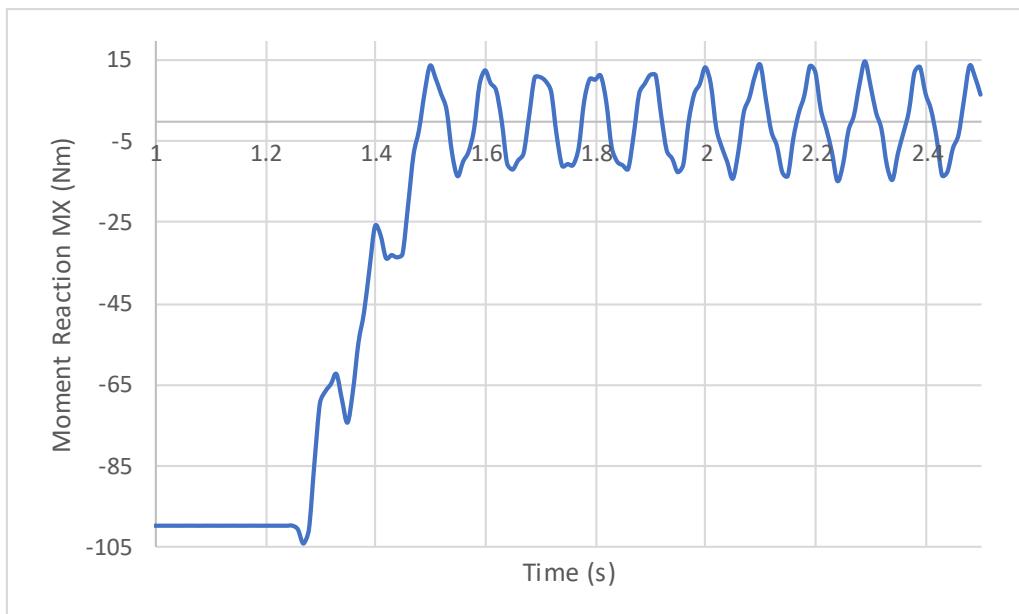


Figure 6-9 - Moment Reaction MX vs time.

Tutorial 7. Using several solution steps in a cantilever beam model

Input file name: tutorial07.inp

Developed by Alfredo Gay Neto & Gabriel Maki

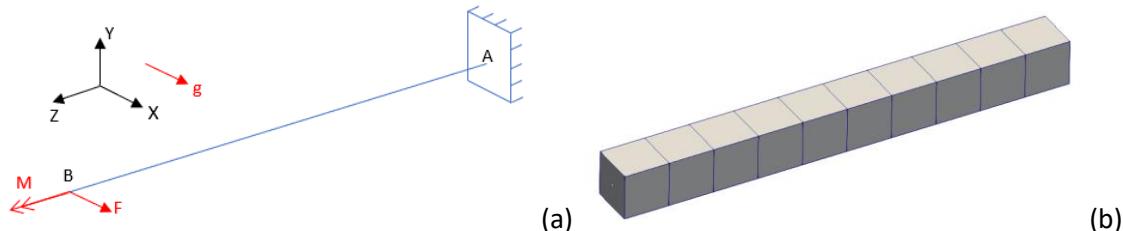


Figure 7-1 - (a) – Cantilever beam model (b) Beam mesh render view

Objective

This tutorial presents a geometrically nonlinear simulation of a cantilever beam. The objective is to introduce the gravitational field in Giraffe and show how to perform a static analysis with many solution steps.

Description

The beam AB (1 m of length) is meshed using 11 nodes and 5 elements, using the element Beam_1. There is a clamp boundary condition at point A. Three loads are applied sequentially. First, a force F is applied at point B. Then, a moment M is applied at point B. The third load is the gravitational field, inducing the weight (distributed along the whole structure volume). Figure 7-1 illustrates the problem. Three loads are applied during three steps. The force and the moment are applied during step 2 to 4 and the gravitational field is applied during step 3 to 5. On the sixth step, all the loads are removed.

Input file step-by-step

Creation of nodes

A total of 11 nodes is created to establish a finite element mesh.

```
//Creation of nodes
Nodes 11
//Number      X      Y      Z
Node  1      0      0      0.0
Node  2      0      0      0.1
Node  3      0      0      0.2
Node  4      0      0      0.3
Node  5      0      0      0.4
Node  6      0      0      0.5
Node  7      0      0      0.6
Node  8      0      0      0.7
Node  9      0      0      0.8
Node 10     0      0      0.9
Node 11     0      0      1.0
```

Creation of node sets

Two node sets are created to establish loads and constraints.

```
//Creation of node sets
NodeSets      2
//Point A
NodeSet       1      Nodes 1      List   1
//Point B
NodeSet       2      Nodes 1      List   11
```

Creation of elements

A total of 5 elements is defined. All the elements are assigned to the same material properties, same cross section and same coordinate system.

```
//Creation of elements
Elements      5
Beam_1 1     Mat    1      Sec    1      CS    1      Nodes 1      2      3
Beam_1 2     Mat    1      Sec    1      CS    1      Nodes 3      4      5
Beam_1 3     Mat    1      Sec    1      CS    1      Nodes 5      6      7
Beam_1 4     Mat    1      Sec    1      CS    1      Nodes 7      8      9
Beam_1 5     Mat    1      Sec    1      CS    1      Nodes 9      10     11
```

Creation of materials

One only material model is defined, which identification number is 1.

```
//Creation of materials
Materials     1
Hooke 1       E      1e7      Nu     0.3      Rho    2000
```

Creation of cross-sections

One only cross-section is defined, that is a rectangle. Both B and H are set to 0.1 m, to represent a square cross-section. The identification number of the cross-section is 1.

```
//Creation of sections
Sections      1
Rectangle    1      B      0.1      H      0.1
```

Creation of coordinate systems

One only coordinate system is created. It is important to mention that, since employing Beam_1 element, the direction E3 must lie in the beam alignment direction. The identification number of the coordinate system is 1.

```
//Creation of coordinate systems
CoordinateSystems 1
CS      1      E1      1      0      0      E3      0      0      1
```

Creation of solution steps

This tutorial has six solution steps. The first solution step is created establishing the end time equal to "1". The second solution step is created establishing the end time equal to "2" and the others follow this pattern. Each solution step has a time step of 0.1, that is, the loads are applied in increments of 10%.

```

//Creation of the solution steps
SolutionSteps6
//First step: FX applied at the tip of the beam
Static 1
EndTime      1
TimeStep     0.1
MaxTimeStep  0.1
MinTimeStep  0.01
MaxIt       20
MinIt       3
ConvIncrease 4
IncFactor    1.0
Sample 2
//Second step: FX kept and MZ applied at the tip of the beam
Static 2
EndTime      2
TimeStep     0.1
MaxTimeStep  0.1
MinTimeStep  0.01
MaxIt       20
MinIt       3
ConvIncrease 4
IncFactor    1.0
Sample 2
//Third step: FX and MZ kept and gravitational field is turned on
Static 3
EndTime      3
TimeStep     0.1
MaxTimeStep  0.1
MinTimeStep  0.01
MaxIt       20
MinIt       3
ConvIncrease 4
IncFactor    1.0
Sample 2
//Fourth step: All kept (FX, MZ and gravity)
Static 4
EndTime      4
TimeStep     0.1
MaxTimeStep  0.1
MinTimeStep  0.01
MaxIt       20
MinIt       3
ConvIncrease 4
IncFactor    1.0
Sample 2
//Fifth step: Only gravity is kept and FX and MZ are removed
Static 5
EndTime      5
TimeStep     0.1
MaxTimeStep  0.1
MinTimeStep  0.01
MaxIt       20
MinIt       3
ConvIncrease 4
IncFactor    1.0
Sample 2
//Sixth step: gravity is removed (no loads at the end)
Static 6
EndTime      6
TimeStep     0.1
MaxTimeStep  0.1
MinTimeStep  0.01
MaxIt       20

```

```
MinIt 3
ConvIncrease 4
IncFactor 1.0
Sample 2
```

Creation of loads

A single nodal load is created. Note that the load application time is associated with the start and end time of the solution steps in this example. This is not strictly necessary. The user may create time-varying loads, according to arbitrary choices.

```
//Creation of loads
Loads 1
NodalLoad 1 NodeSet 2 CS 1 NTimes 5
//Time FX FY FZ MX MY MZ
0 0 0 0 0 0
1 1000 0 0 0 0
2 1000 0 0 0 0 250
4 1000 0 0 0 0 250
5 0 0 0 0 0
```

Creation of constraints

A clamp boundary condition is created. This is done by fixing all the degrees of freedom during the entire simulation. It is important to remember that, in BoolTable, there is a single “1”, but it means that the constraint is applied for all the steps.

```
//Creation of constraints
Constraints 1
NodalConstraint 1 NodeSet 1
UX BoolTable 1
UY BoolTable 1
UZ BoolTable 1
ROTX BoolTable 1
ROTY BoolTable 1
ROTZ BoolTable 1
```

Creation of Environment Data

An environment data is created to introduce the gravity field on the model. Note that it is not kept during the entire simulation. The BoolTable has the objective of establishing in which solution steps such load is going to be considered. In our example, the gravitational field is linearly imposed along solution step 3, but is linearly turned off along solution step 6.

```
//Creation of Environment data
Environment
//Gravity data
GravityData
G 9.81 0 0 BoolTable 0 0 1 1 1 0
```

Creation of solver options

Here we request Giraffe to apply a single thread to solve this case. The linear system solution chosen is Direct.

```
//Creation of solver options
SolverOptions
Processors 1 LinSys Direct
```

Creation of monitors

We request Giraffe to monitor 2 nodes during the simulation: numbers 1 and 11.

```
//Creation of monitors
Monitor      Sample 1
MonitorNodes 1           11
```

Creation of post-files

The options used here are the same as the tutorial 01, where the WriteMesh, WriteRenderMesh and WriteForces are chosen for post-processing using Paraview™ and the MagFactor parameters assume a unit value so the deformed shape of Paraview™ frames will reflect the real scale.

```
//Cration of post files
PostFiles
MagFactor    1
WriteMesh     1
WriteRenderMesh 1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces 0
WriteForces   1
WriteConstraints 0
WriteSpecialConstraints 0
WriteContactForces 0
WriteRenderRigidBodies 0
WriteRenderParticles 0
```

Results and post-processing

To post-process data using Paraview™, first we must open the files “whole_solution_forces.pvd”, “whole_solution_mesh.pvd” and “whole_solution_rendermesh.pvd”, which are located at the folder “post” in “tutorial07”.

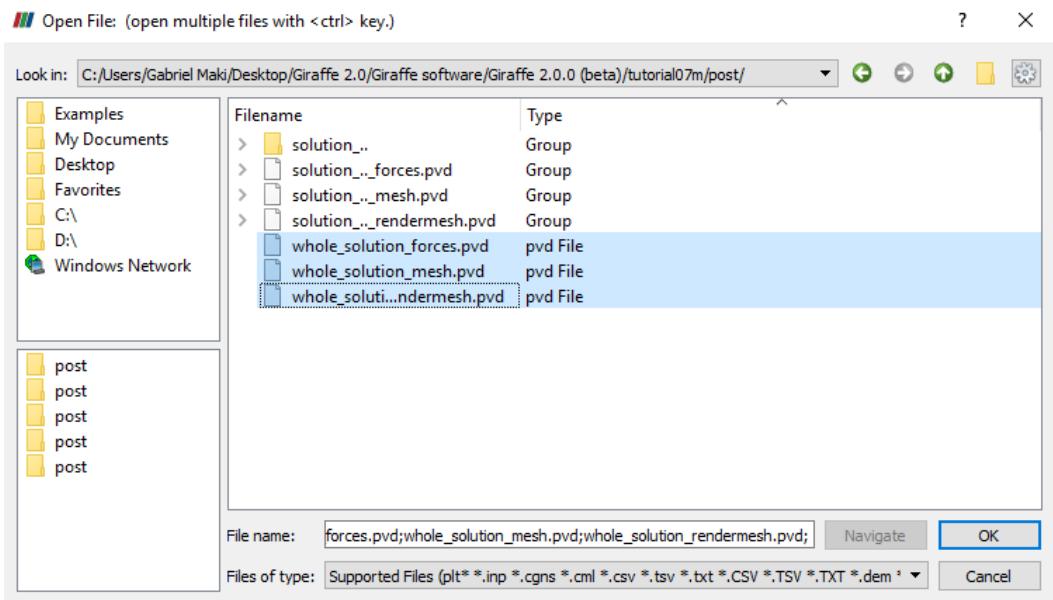


Figure 7-2 - Opening the files in Paraview™.

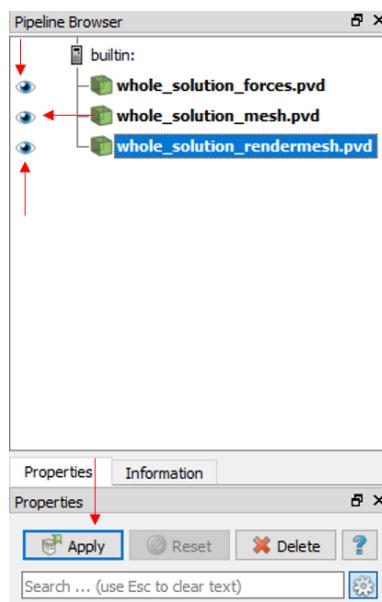


Figure 7-3 - Showing the Pipeline Browser.

For a more convenient view with no distorting dimensions, one may adopt an isometric perspective. For that, go to Properties and check “Camera Parallel Projection in Orientation Axes.

Next, we can create a Glyph to see the symbol of the load being applied. For that, go to the Pipeline Browser and click in “whole_solution_forces.pvd”. Then, click the Glyph twice.

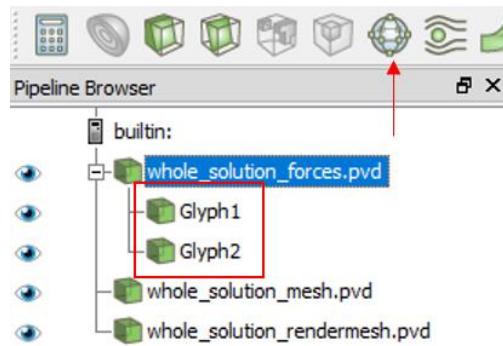


Figure 7-4 - Creating the two Glyphs.

After creating the Glyphs, we have to adjust the scale of vectors. To do this, go to the Pipeline Browser and, first, click the preview icon of Glyph. Then, in Scaling, set the scale mode to vector, change the scale factor to a small number like 0.001 and click Apply. After doing this for both Glyphs, in Glyph 2, we must change, in Active Attributes, the Vectors to Moment and click Apply.

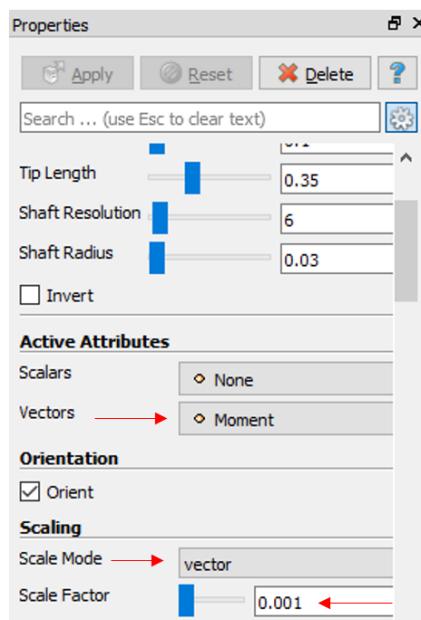


Figure 7-5 - Changing the Properties of Glyph 2.

Finally, we can show an animation of the simulation results. If the user wants to know the values of displacements of some nodes or the reaction forces on the clamp in a certain instant, we can gather such information in the “monitors” folder, since they are requested prior to simulation solving.

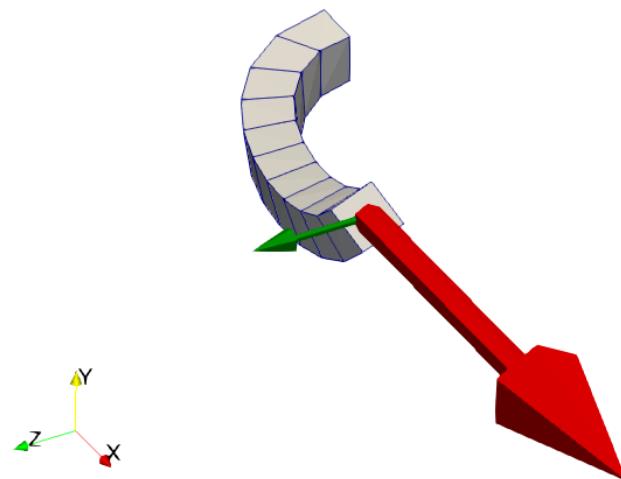


Figure 7-6 - Deformed shape at the end of third solution step.

In Figure 7-6 it is possible to see that the background color has changed. To do this, click the tool indicated in Figure 7-7 and choose the color.

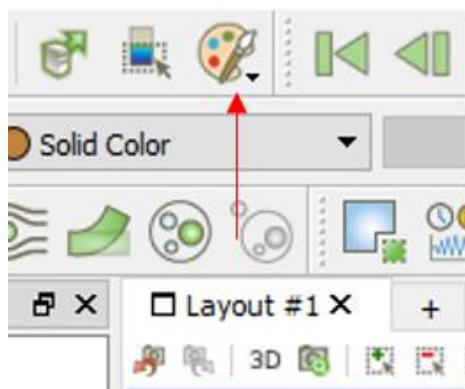


Figure 7-7 - Changing the background color.

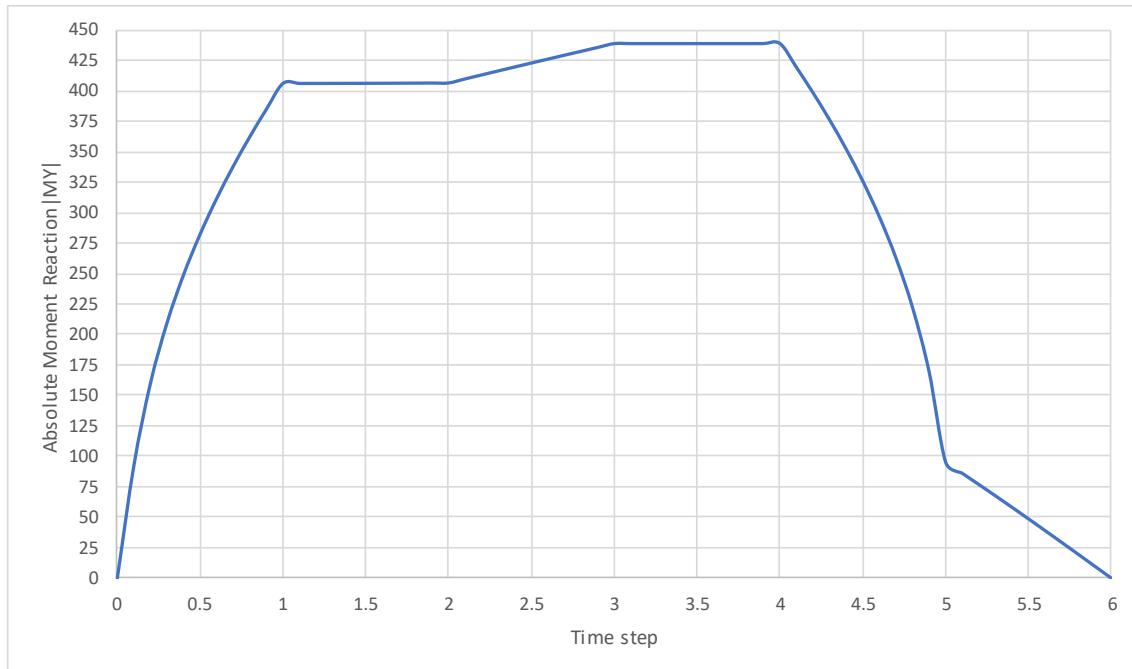


Figure 7-8 – Absolute Moment Reaction |MY| vs Time step

An interesting result can be seen in Figure 7-8. It shows the importance of geometric nonlinearity. Using the reactive moment in direction Y as example, if we calculate this reaction considering the undisturbed structure, it results in 1098 Nm. If we consider the nonlinearity and calculate numerically, it results in 438,4 Nm as shown in Figure 7-8.

Tutorial 8. Evaluation of system modal analysis during a static simulation

Input file name: tutorial08.inp

Developed by Alfredo Gay Neto & Gabriel Maki

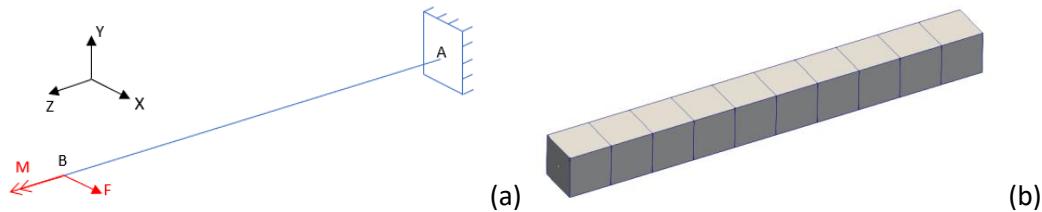


Figure 8-1 - (a) – Beam model (b) Beam mesh render view

Objective

This tutorial presents a geometrically-nonlinear simulation of a cantilever beam. The aim is to give an example on how to work with concomitant solutions.

Description

The beam AB (1 m of length) is meshed using 11 nodes and 5 elements, using the element Beam_1. There is a clamp boundary condition at point A. Two loads are applied sequentially. First, a force F is applied at point B. then, a moment M is applied at the same point. The simulation is done using three static solution steps and, during the first and third solution steps, a concomitant modal analysis is done to evaluate natural frequencies for each deformed (and stressed) configuration.

Input file step-by-step

Creation of nodes

A total of 11 nodes is created to establish a finite element mesh.

```
//Creation of nodes
Nodes 11
//Number      X      Y      Z
Node  1      0      0      0.0
Node  2      0      0      0.1
Node  3      0      0      0.2
Node  4      0      0      0.3
Node  5      0      0      0.4
Node  6      0      0      0.5
Node  7      0      0      0.6
Node  8      0      0      0.7
Node  9      0      0      0.8
Node 10     0      0      0.9
Node 11     0      0      1.0
```

Creation of node sets

Two node sets are created to establish loads and constraints.

```
//Creation of node sets
NodeSets      2
//Point A
NodeSet       1      Nodes 1      List  1
//Point B
NodeSet       2      Nodes 1      List  11
```

Creation of elements

A total of 5 elements is defined. All the elements are assigned to the same material properties, same cross section and same coordinate system.

```
//Creation of elements
Elements      5
Beam_1 1      Mat   1      Sec   1      CS    1      Nodes 1      2      3
Beam_1 2      Mat   1      Sec   1      CS    1      Nodes 3      4      5
Beam_1 3      Mat   1      Sec   1      CS    1      Nodes 5      6      7
Beam_1 4      Mat   1      Sec   1      CS    1      Nodes 7      8      9
Beam_1 5      Mat   1      Sec   1      CS    1      Nodes 9      10     11
```

Creation of materials

One only material model is defined, which identification number is 1.

```
//Creation of materials
Materials      1
Hooke 1        E     1e7      Nu    0.3      Rho   2000
```

Creation of cross-sections

One only cross-section is defined, that is a rectangle. Both B and H are set to 0.1 m, to represent a square cross-section. The identification number of the cross-section is 1.

```
//Creation of sections
Sections      1
Rectangle 1    B     0.1      H     0.1
```

Creation of coordinate systems

One only coordinate system is created. It is important to mention that, since employing Beam_1 element, the direction E3 must lie in the beam alignment direction. The identification number of the coordinate system is 1.

```
//Creation of coordinate systems
CoordinateSystems 1
CS      1      E1      1      0      0      E3      0      0      1
```

Creation of solution steps

This tutorial has three solution steps. All the steps are statics and have a time step of 0.1 which means that the loads are applied or reduced in fractions of 10%.

```
/Creation of the solution steps
SolutionSteps3
//First step: FX applied at the tip of the beam
Static 1
EndTime      1
TimeStep     0.1
MaxTimeStep  0.1
MinTimeStep  0.01
```

```

MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor 1.0
Sample 2
//Second step: FX kept and MZ applied at the tip of the beam
Static 2
EndTime 2
TimeStep 0.1
MaxTimeStep 0.1
MinTimeStep 0.01
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor 1.0
Sample 2
//Third step: all loads are removed (no loads at the end)
Static 3
EndTime 3
TimeStep 0.1
MaxTimeStep 0.1
MinTimeStep 0.01
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor 1.0
Sample 2

```

Creation of concomitant solutions

A concomitant solution is created to run a modal analysis during the first and third solution step. In this case, we request Giraffe to evaluate the smallest ten natural frequencies related eigenvalues. The BoolTable keyword is employed to choose only the first and the third solution steps, to have the concomitant solution evaluated with.

```

//Creation of concomitant solution
ConcomitantSolution
Sample 3      BoolTable 1 0 1
Modal NumberModes 10      Tolerance 1E-6

```

Creation of loads

A single nodal load is created. Note that the load application time is associated with the start and end time of the solution steps.

```

//Creation of loads
Loads 1
NodalLoad 1      NodeSet 2      CS 1      NTimes 4
//Time FX FY FZ MX MY MZ
0      0      0      0      0      0      0
1      1000  0      0      0      0      0
2      1000  0      0      0      0      250
3      0      0      0      0      0      0

```

Creation of constraints

A clamp boundary condition is created. This is done by fixing all the degrees of freedom during the entire simulation. This is done in node set 1.

```
//Creation of constraints
Constraints 1
NodalConstraint 1      NodeSet      1
    UX   BoolTable  1
    UY   BoolTable  1
    UZ   BoolTable  1
    ROTX BoolTable  1
    ROTY BoolTable  1
    ROTZ BoolTable  1
```

Creation of solver options

Here we request Giraffe to apply a single thread to solve this case. The linear system solution chosen is Direct.

```
//Creation of solver options
SolverOptions
Processors 1      LinSys Direct
```

Creation of monitors

We request Giraffe to monitor 2 nodes during the simulation: numbers 1 and 11.

```
//Creation of monitors
Monitor      Sample 1
MonitorNodes 1      11
```

Creation of post-files

The options used here are the same as the tutorial 01, with an exception that is the “WriteForces”. This was turned off to avoid saving many solution files, unnecessary for present objectives.

```
//Creation of post files
PostFiles
MagFactor     1
WriteMesh     1
WriteRenderMesh 1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces 0
WriteForces    0
WriteConstraints 0
WriteSpecialConstraints 0
WriteContactForces 0
WriteRenderRigidBodies 0
WriteRenderParticles 0
```

Results and post-processing

To post-process results in ParaviewTM, first we must open the files “whole_solution_mesh.pvd” and “whole_solution_rendermesh.pvd”, which are located at the folder “post” in “tutorial08”.

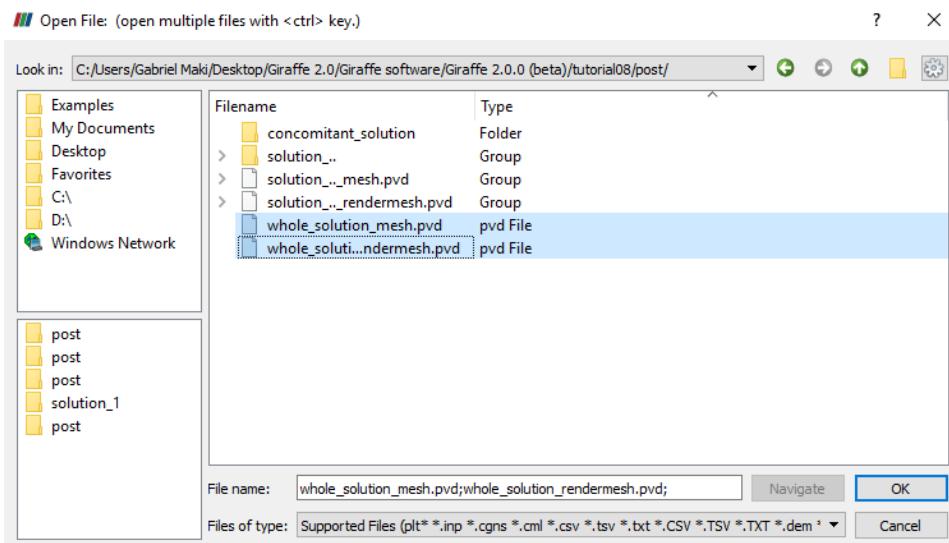


Figure 8-2 - Opening the files of tutorial 7 in Paraview™.

Then, we have to click the preview icon and the Apply button on the Pipeline browser.

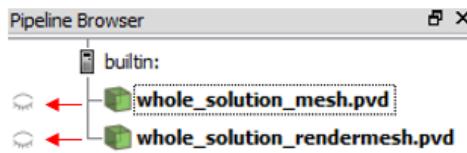


Figure 8-3 - Showing the Pipeline Browser.

The user may note that there is no file for forces, so we do not create Glyphs. But, unlike other tutorials, there is a folder called “concomitant_solution”. Inside this folder, we find a text file with the same name. This file contains all the eigenvalues associated with the vibration modes of the structure, evaluated along time. That is, the eigenvalues are calculated while the structure is deforming.

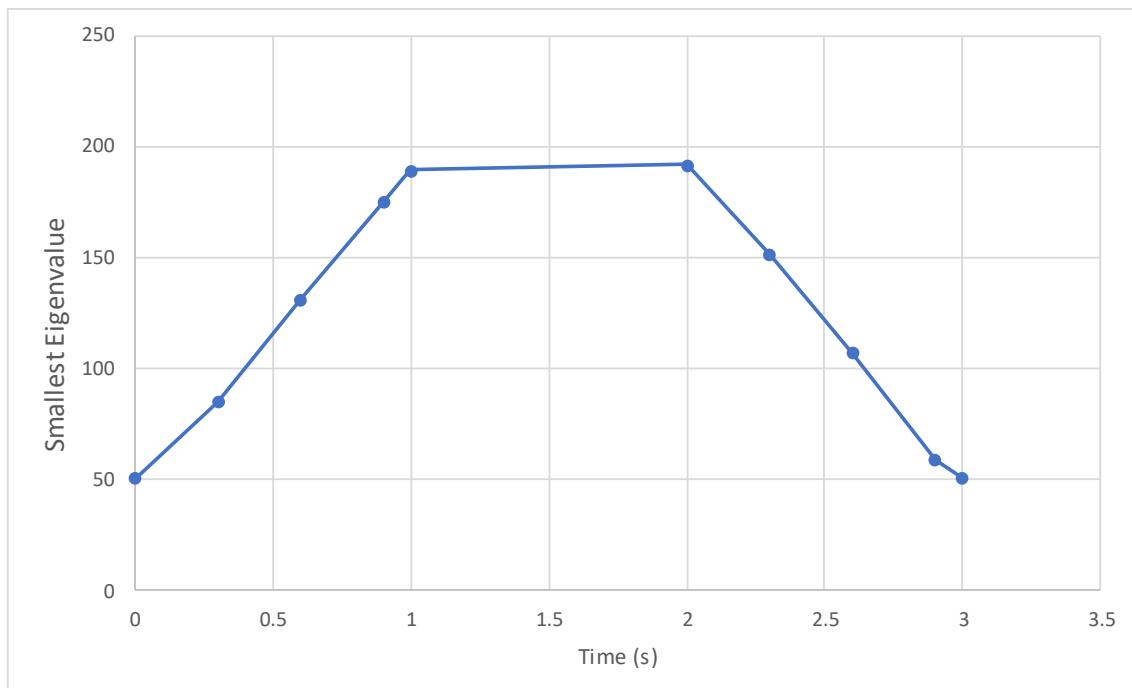


Figure 8-4 – Smallest eigenvalue vs. time.

Tutorial 9. Evaluation of a system modal analysis during buckling of a cantilever beam

Input file name: tutorial09.inp

Developed by Alfredo Gay Neto & Gabriel Maki

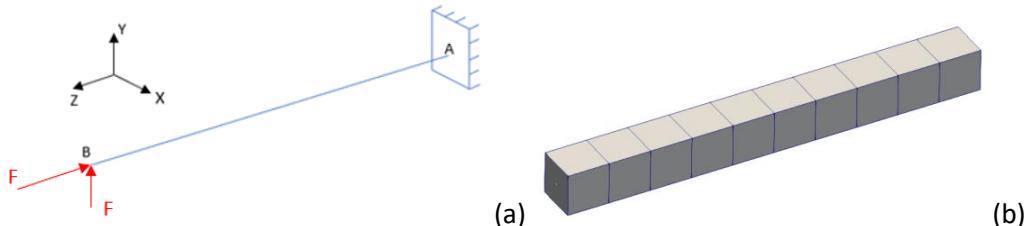


Figure 9-1 - (a) – Beam subjected to buckling (b) Beam mesh render view

Objective

This tutorial presents a geometrically-nonlinear simulation of a cantilever beam. The aim is to perform a modal analysis during the nonlinear buckling simulation of the structure. For that, it is necessary to introduce an imperfection on the beam and the trigger load, as shown in Figure 9-1(a). Besides, the resource of concomitant solution is used on this tutorial too.

The beam AB (1 m of length) is meshed using 51 nodes and 25 elements, using the element Beam_1. There is a clamp boundary condition at point A. Two nodal loads are applied sequentially at point B. First, a lateral load is applied. After, a compressive load is applied. The lateral force is necessary because it is not possible to conduct a nonlinear buckling analysis without an initial imperfection on the structure.

Input file step-by-step

Creation of nodes

A total of 51 nodes is created to establish a finite element mesh.

```
//Creation of nodes
Nodes 51
//Number      X      Y      Z
Node   1      0      0      0.0
Node   2      0      0      0.02
Node   3      0      0      0.04
Node   4      0      0      0.06
Node   5      0      0      0.08
...
Node  45      0      0      0.88
Node  46      0      0      0.9
Node  47      0      0      0.92
Node  48      0      0      0.94
Node  49      0      0      0.96
Node  50      0      0      0.98
Node  51      0      0      1.0
```

Creation of node sets

Two node sets are created to establish loads and constraints.

```
//Creation of node sets
NodeSets      2
//Point A
NodeSet       1      Nodes 1      List   1
//Point B
NodeSet       2      Nodes 1      List   51
```

Creation of elements

A total of 25 elements is defined. All the elements are assigned to the same material properties, same cross section and same coordinate system.

```
//Creation of elements
Elements      25
Beam_1 1     Mat    1      Sec   1      CS    1      Nodes 1      2      3
Beam_1 2     Mat    1      Sec   1      CS    1      Nodes 3      4      5
Beam_1 3     Mat    1      Sec   1      CS    1      Nodes 5      6      7
Beam_1 4     Mat    1      Sec   1      CS    1      Nodes 7      8      9
...
Beam_1 22    Mat    1      Sec   1      CS    1      Nodes 43     44     45
Beam_1 23    Mat    1      Sec   1      CS    1      Nodes 45     46     47
Beam_1 24    Mat    1      Sec   1      CS    1      Nodes 47     48     49
Beam_1 25    Mat    1      Sec   1      CS    1      Nodes 49     50     51
```

Creation of materials

One only material model is defined, which identification number is 1.

```
//Creation of materials
Materials     1
Hooke 1      E      1e7      Nu     0.3      Rho    2000
```

Creation of cross-sections

One only cross-section is defined, that is a rectangle. Both B and H are set to 0.1 m, to represent a square cross-section. The identification number of the cross-section is 1.

```
//Creation of sections
Sections      1
Rectangle    1      B      0.1      H      0.1
```

Creation of coordinate systems

One only coordinate system is created. It is important to mention that, since employing Beam_1 element, the direction E3 must lie in the beam alignment direction. The identification number of the coordinate system is 1.

```
//Creation of coordinate systems
CoordinateSystems 1
CS      1      E1      1      0      0      E3      0      0      1
```

Creation of solution steps

We create only two solution steps, both statics. The first step is associated with the initial imperfection that we must introduce for conduct a nonlinear buckling analysis. The second step is associated with the compression force that leads to buckling.

```
//Creation of the solution steps
SolutionSteps2
//First step: Imperfection force at the tip of the beam
Static 1
EndTime 1
TimeStep      0.1
MaxTimeStep   0.1
MinTimeStep   0.01
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor    1.0
Sample 2
//Second step: Compression force applied at the tip of the beam
Static 2
EndTime      2
TimeStep      0.1
MaxTimeStep   0.1
MinTimeStep   0.01
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor    1.0
Sample 2
```

Creation of loads

Two loads are created. The first is the force applied at the tip of the beam to introduce an imperfection and induce buckling. The second load is the compression force. It is important to mention that the compression force used is a bit larger than the Euler's critical load. That one is calculated by the following expression:

$$F_{cr} = \frac{\pi^2 EI}{(2l)^2}$$

Where F_{cr} is the Euler's critical load, E is the Young's Module, I is the cross section's moment of inertia and l is the length of the beam. Using that expression, we obtain the value of 205.62 N.

```
//Creation of loads
Loads 2
NodalLoad 1      NodeSet      2      CS      1      NTimes 2
//Time FX FY FZ MX MY MZ
0      0      0      0      0      0
1      0      1      0      0      0
NodalLoad 2      NodeSet      2      CS      1      NTimes 2
//Time FX FY FZ MX MY MZ
1      0      0      0      0      0
2      0      0      -206    0      0
```

Creation of concomitant solution

A concomitant solution is created to run a modal analysis during the second solution step. In this case, we request Giraffe to evaluate the 10 smallest eigenvalues, associated with natural frequencies of the system.

```
//Creation of concomitant solution
ConcomitantSolution
Sample 3      BoolTable 0 1
Modal NumberModes 10      Tolerance 1E-6
```

Creation of constraints

A clamp boundary condition is created. This is done by fixing all the degrees of freedom during the entire simulation.

```
//Creation of constraints
Constraints 1
NodalConstraint 1      NodeSet 1
    UX  BoolTable 1
    UY  BoolTable 1
    UZ  BoolTable 1
    ROTX  BoolTable 1
    ROTY  BoolTable 1
    ROTZ  BoolTable 1
```

Creation of solver options

Here we request Giraffe to apply a single thread to solve this case. The linear system solution chosen is Direct.

```
//Creation of solver options
SolverOptions
Processors 1      LinSys Direct
```

Creation of monitors

We request Giraffe to monitor 2 nodes during the simulation: numbers 1 and 11.

```
//Creation of monitors
Monitor Sample 1
MonitorNodes 1      51
```

Creation of post-files

The options used here are the same as the tutorial 01, with an exception that is the “WriteForces”.

```
//Creation of post files
PostFiles
MagFactor 1
WriteMesh 1
WriteRenderMesh 1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces 0
WriteForces 0
WriteConstraints 0
WriteSpecialConstraints 0
WriteContactForces 0
WriteRenderRigidBodies 0
```

Results and post-processing

To post-process results in Paraview™, first we have to open the files “whole_solution_mesh.pvd” and “whole_solution_rendermesh.pvd”, which are located at the folder “post” in “tutorial09”.

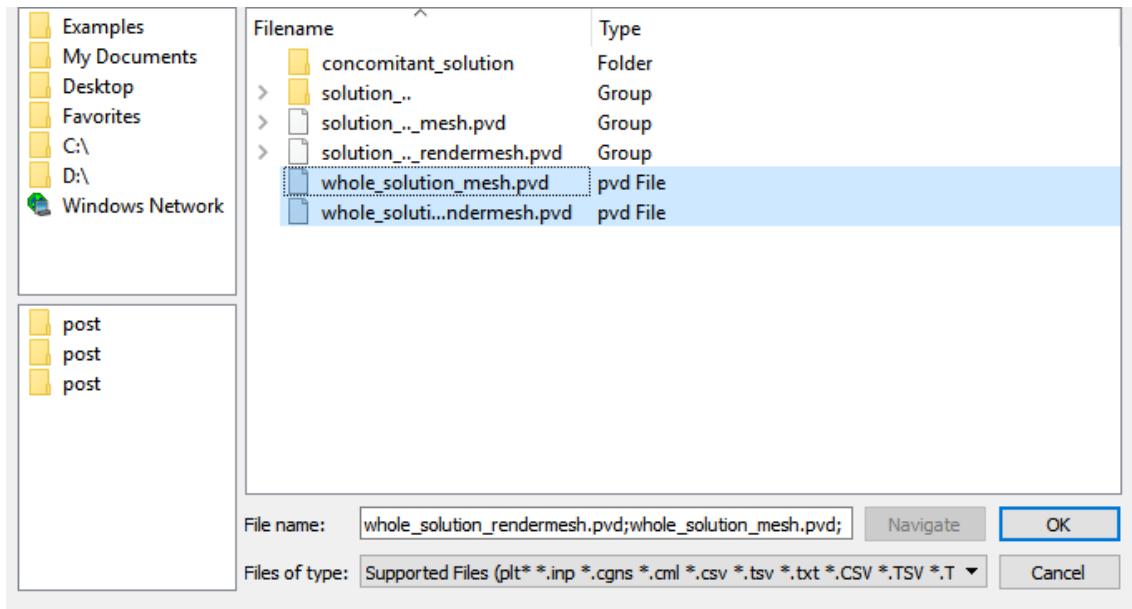


Figure 9-2 - Opening the files of tutorial 09 in Paraview™.

Then, we can click the preview icon and the Apply button on the Pipeline browser. After doing that, we are ready to post-process the results.

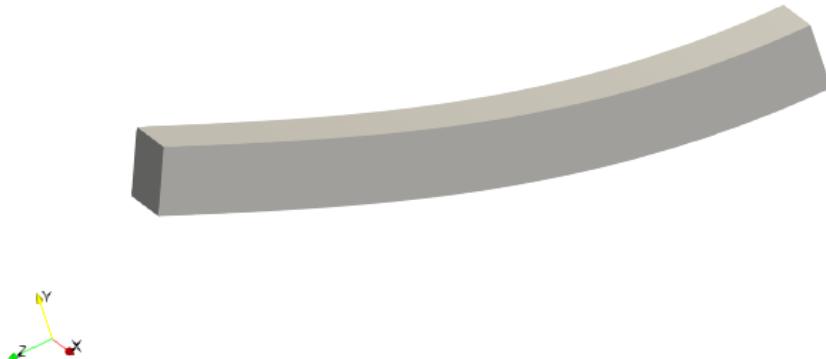


Figure 9-3 - Deformed shape of the structure (render view).

Like mentioned in the previous tutorial, there is a folder called “concomitant_solution”. Inside this folder, we find a text file with the same name. This file contains all the eigenvalues assigned to the vibration modes requested, evaluated over time. One may plot the time-evolution of the smallest eigenvalue real part, leading to Figure 9-4.

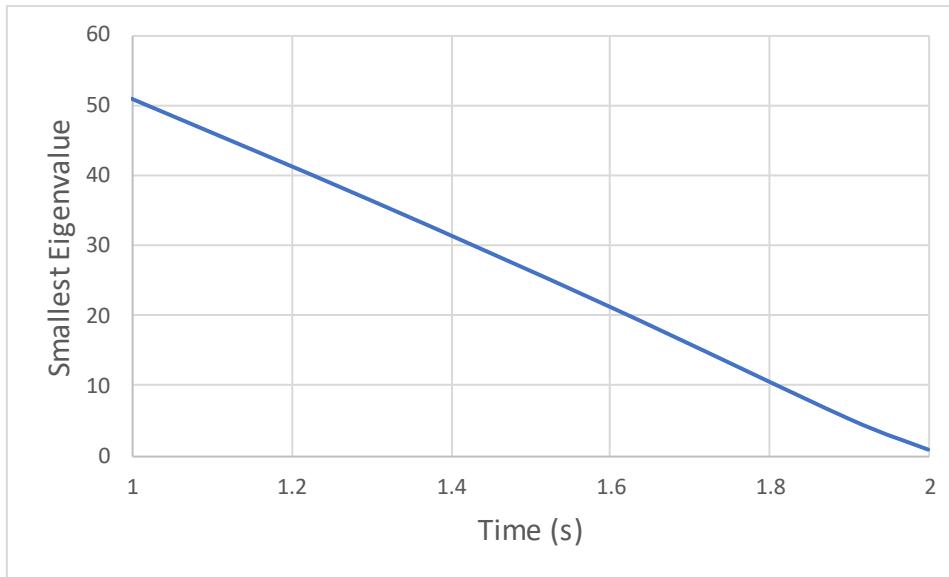


Figure 9-4 - Smallest Eigenvalue vs time.

Note that the smallest eigenvalue approaches zero at the end of simulation, which means that the structure loses stiffness and thus experiences buckling. Besides, there is another way to check it. If we do not introduce an imperfection force, the same eigenvalue analysis may be done. In this case, the eigenvalue does cross the abscissa axis. The instant at which there is the abscissa crossing is related to the Euler's critical load.

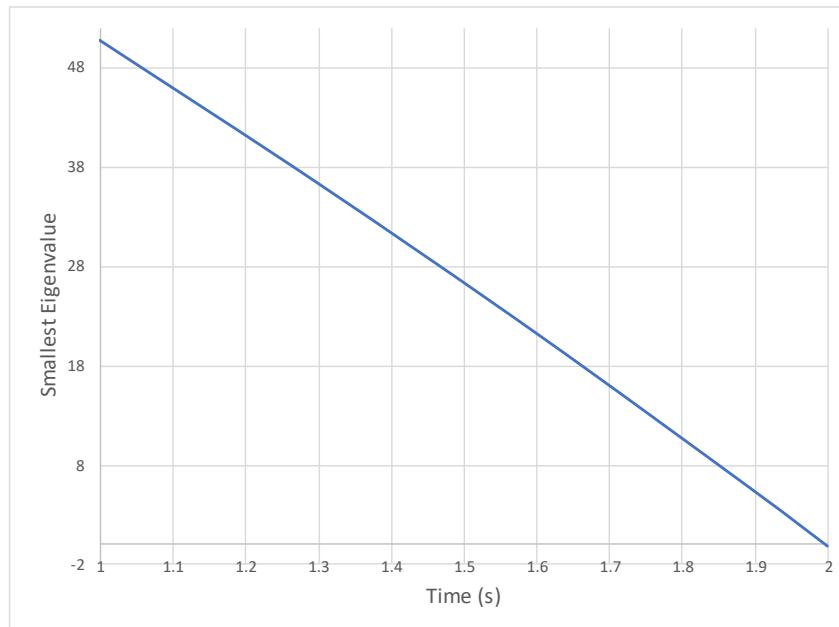


Figure 9-5 - Smallest Eigenvalue vs Time (without imperfection force).

Giraffe generates a monitor text file for each node requested containing the values of displacement and force in each global direction. Therefore, it is possible to obtain the vertical displacement of point B along the simulation evolution. Besides, we can vary the imperfection load in the input file, generating a collection of plots. Then, we can make an equilibrium configuration plot, shown in Figure 9-6.

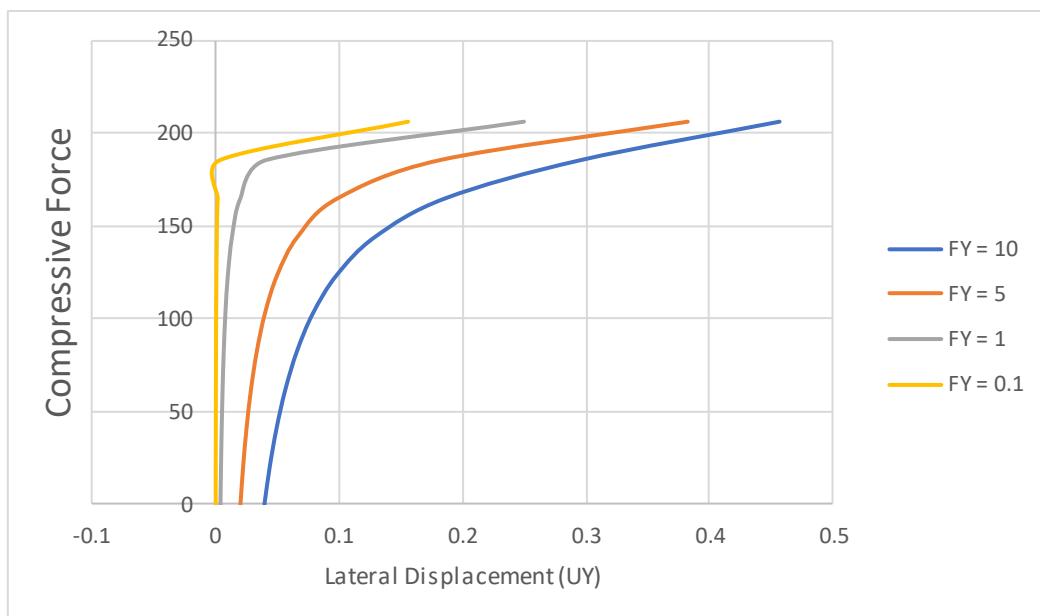


Figure 9-6 - Vertical Displacement UY vs Lateral Force FY.

Tutorial 10. Transient dynamics of a beam

Input file name: tutorial10.inp

Developed by Alfredo Gay Neto & Gabriel Maki

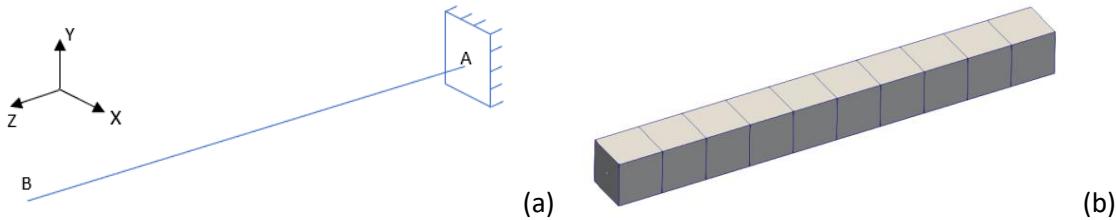


Figure 10-1 - (a) – Beam model (b) – Beam mesh

Objective

This tutorial presents a geometrically nonlinear simulation of a cantilever beam. The objective is to show how to work with prescribed displacements and rotations in Giraffe.

Description

The beam AB (1 m of length) is meshed using 11 nodes and 5 elements, using the element Beam_1. There is a clamp boundary condition at point A. Besides, there is a single solution step, which is dynamic and every 2.5 seconds, a displacement or rotation is prescribed in point A.

Input file step-by-step

Creation of nodes

A total of 11 nodes is created to establish a finite element mesh.

```
//Creation of nodes
Nodes 11
//Number      X      Y      Z
Node  1      0      0      0.0
Node  2      0      0      0.1
Node  3      0      0      0.2
Node  4      0      0      0.3
Node  5      0      0      0.4
Node  6      0      0      0.5
Node  7      0      0      0.6
Node  8      0      0      0.7
Node  9      0      0      0.8
Node 10     0      0      0.9
Node 11     0      0      1.0
```

Creation of node sets

Two node sets are created to establish displacements and constraints.

```
//Creation of node sets
NodeSets    2
```

```
//Point A
NodeSet      1      Nodes  1      List   1
//Point B
NodeSet      2      Nodes  1      List   11
```

Creation of elements

A total of 5 elements is defined. All the elements are assigned to the same material properties, same cross section and same coordinate system.

```
//Creation of elements
Elements      5
Beam_1 1      Mat    1      Sec    1      CS     1      Nodes  1      2      3
Beam_1 2      Mat    1      Sec    1      CS     1      Nodes  3      4      5
Beam_1 3      Mat    1      Sec    1      CS     1      Nodes  5      6      7
Beam_1 4      Mat    1      Sec    1      CS     1      Nodes  7      8      9
Beam_1 5      Mat    1      Sec    1      CS     1      Nodes  9      10     11
```

Creation of materials

One only material model is defined, which identification number is 1.

```
//Creation of materials
Materials      1
Hooke 1       E      1e7      Nu      0.3      Rho      2000
```

Creation of cross-sections

One only cross-section is defined, that is a rectangle. Both B and H are set to 0.1 m, to represent a square cross-section. The identification number of the cross-section is 1.

```
//Creation of sections
Sections      1
Rectangle    1      B      0.1      H      0.1
```

Creation of coordinate systems

One only coordinate system is created. It is important to mention that, since employing Beam_1 element, the direction E3 must lie in the beam alignment direction. The identification number of the coordinate system is 1.

```
//Creation of coordinate systems
CoordinateSystems 1
CS      1      E1      1      0      0      E3      0      0      1
```

Creation of solution steps

This tutorial has a single solution step, which is dynamic. This is created establishing the end time equal to "10".

```
//Creation of the solution steps
SolutionSteps1
//First step: displacement prescribed at point A
```

```

Dynamic      1
EndTime     10.0
TimeStep    0.005
MaxTimeStep 0.005
MinTimeStep 0.001
MaxIt       20
MinIt       3
ConvIncrease 4
IncFactor   1.0
Sample      10
RayleighDamping Alpha 0      Beta 0      Update 0
NewmarkCoefficients Beta 0.3  Gamma 0.5

```

Creation of displacements

A nodal displacement is created at point A. The rotation input is made in radians. The displacement/rotation imposition is done in an incremental way, like the loads. It is important to mention that, **if the degree of freedom is free, the nodal displacement related to this degree is ignored by Giraffe** (there is no way to prescribe a given displacement value to a “free” degree of freedom of the system).

```

//Creation of displacements
Displacements 1
NodalDisplacement 1      NodeSet      1      CS      1      NTimes 5
  //Time UX UY UZ RX RY RZ
  0      0      0      0      0      0      0
  2.5    0      0.5    +0.25  0      0      0
  5.0    0      0      0      0      0      3.14
  7.5    0      -0.5   -0.25  0      0      0
  10     0      0      0      0      0      0

```

Creation of constraints

A clamp boundary condition is created. This is done by fixing all the degrees of freedom during the entire simulation. It is important to remember that, in BoolTable, there is a single “1”, but it means that the constraint is applied for all the steps.

```

//Creation of constraints
Constraints 1
NodalConstraint 1      NodeSet      1
  UX      BoolTable  1
  UY      BoolTable  1
  UZ      BoolTable  1
  ROTX   BoolTable  1
  ROTY   BoolTable  1
  ROTZ   BoolTable  1

```

Creation of solver options

Here we request Giraffe to apply a single thread to solve this case. The linear system solution chosen is Direct.

```

//Creation of solver options
SolverOptions
Processors 1      LinSys Direct

```

Creation of monitors

We request Giraffe to monitor 2 nodes during the simulation: numbers 1 and 11.

```
//Creation of monitors
Monitor      Sample 1
MonitorNodes 1      11
```

Creation of post-files

The options used here are the same as the tutorial 01, with an exception that is the “WriteForces”. On this tutorial, there is no load, so we do not have to create a file for forces.

```
//Creation of post files
PostFiles
MagFactor    1
WriteMesh    1
WriteRenderMesh 1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces 0
WriteForces   0
WriteConstraints 0
WriteSpecialConstraints 0
WriteContactForces 0
WriteRenderRigidBodies 0
WriteRenderParticles 0
```

Results and post-processing

To view the post-processing in Paraview™, first we must open the files “whole_solution_mesh.pvd” and “whole_solution_rendermesh.pvd”, which are located at the folder “post” in “tutorial09”.

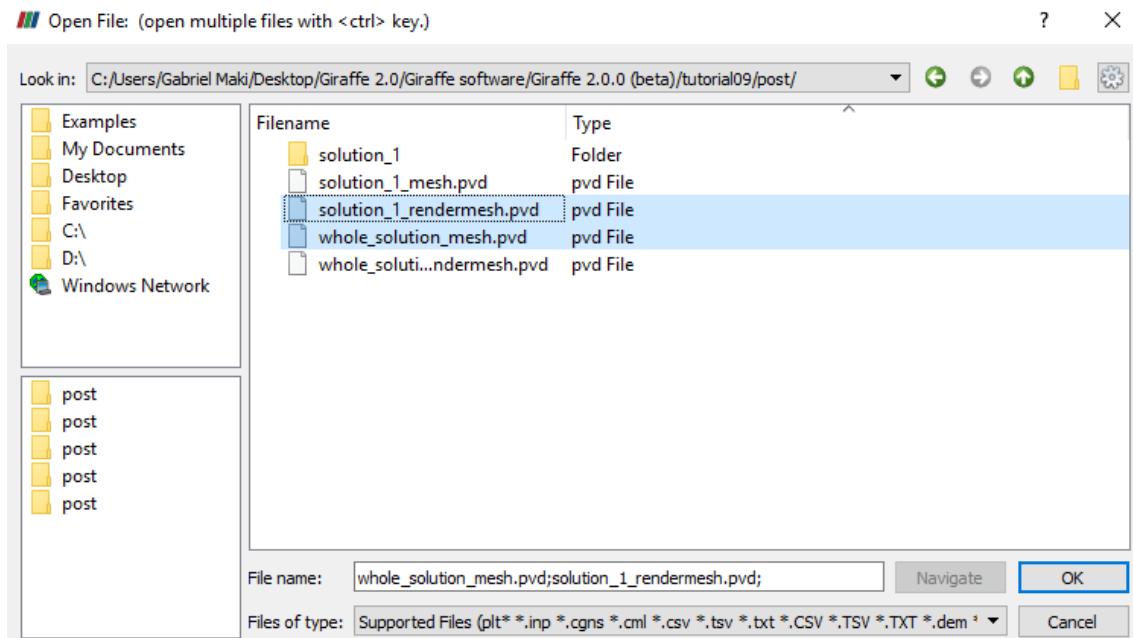


Figure 10-2 - Opening the post-files of tutorial 9 in Paraview™.

Then, we must click the preview icon and the Apply button on the Pipeline browser.

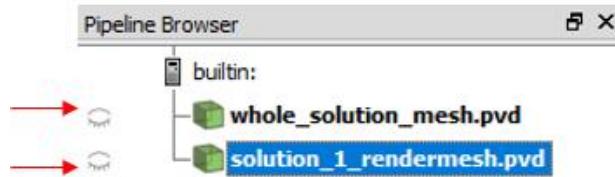


Figure 10-3 - Showing the Pipeline Browser.

For a better view, we must use the isometric projection. To do this, go to Properties and check “Camera Parallel Projection in Orientation Axes”.

Unlike other tutorials, we have no chance to create Glyphs, because there is no load in this model. So, we can run the animation. If the user wants to know the values of displacements of some nodes or the reaction forces on the clamp in a certain instant, we can acquire such information in the “monitors” folder.

In Figure 10-4 we show a comparison between the vertical displacement at the beam tip (node 11) varying Rayleigh stiffness proportional damping, the oscillation clearly decreases when increasing damping.

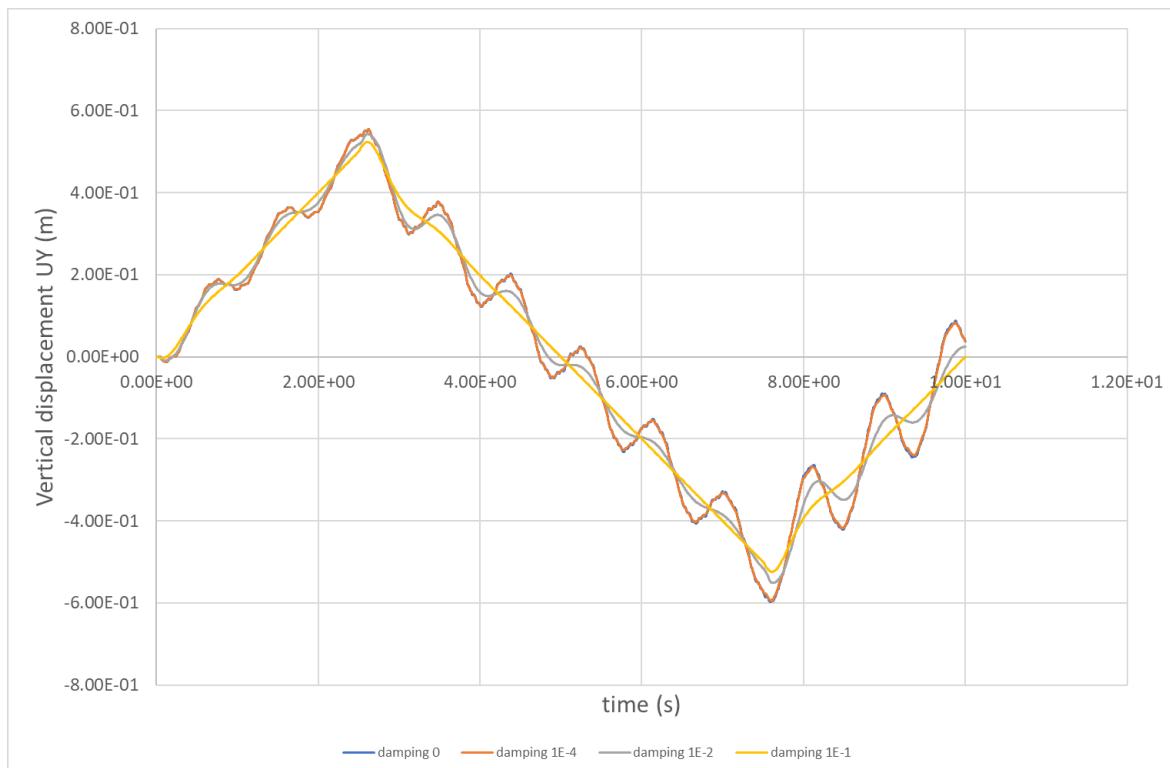


Figure 10-4 – Vertical displacement UY vs. time

Tutorial 11. Two degrees of freedom lumped system

Input file name: tutorial11.inp

Developed by Gabriel Maki and Alfredo Gay Neto

Objective

This tutorial presents a damped mass-spring system with two degrees of freedom. It aims at showing how to create a spring/dashpot element and how to apply a time-varying load. Besides, this tutorial proposes a simple example of a tuned mass damper.

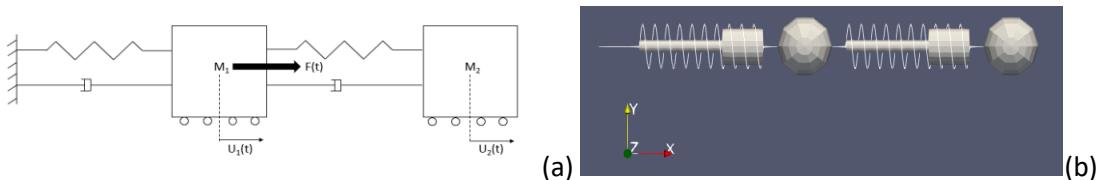


Figure 11-1 - (a) Geometric description of the system (b) 2D rendering for visualization purposes

Description

The system consists of two lumped masses M_1 and M_2 , where $M_1 = 22515.818$ kg and $M_2 = 500$ kg, linked by springs and dashpots. Figure 11-1 shows the model scheme and the degrees of freedom representation. In Giraffe Finite Element environment, the system is represented by three nodes and four elements. The first node is fixed, while the others are free in a single direction (global x). Two elements are created to represent the spring/dashpot and another two are created for the point masses. $U_1(t)$ is defined as the node 2 x-displacement, $U_2(t)$ is defined as the node 3 x-displacement and $F(t)$ is the time varying (harmonic) load, applied at node 2 in direction x.

Input file step-by-step

Creation of nodes

Three nodes are created to represent the system modeled.

```
//Creation of nodes
Nodes 3
//Number      X      Y      Z
Node  1      0      0      0
Node  2      0.5    0      0
Node  3      1.0    0      0
```

Creation of node sets

Four node sets are created to establish loads and constraints.

NodeSets	4					
NodeSet	1	Nodes	1	List	1	
NodeSet	2	Nodes	1	List	2	
NodeSet	3	Nodes	1	List	3	
NodeSet	4	Nodes	2	List	2	3

Creation of elements

Four elements are defined, using two different types of elements. The first is SpringDashpot_1 element. It is used to define a spring/dashpot set. The second element is Mass_1, used to establish a lumped mass.

Elements	4								
SpringDashpot_1	1	1	Stiffness	500000	Damping	5000	Nodes	1	2
SpringDashpot_1	1	2	Stiffness	11103.305	Damping	0	Nodes	2	3
Mass_1	3	Mass	22515.818	Node	2				
Mass_1	4	Mass	500	Node	3				

Creation of coordinate systems

One coordinate system is created, using the same directions of the global coordinate system. It is used to define the nodal load.

CoordinateSystems	1								
CS	1	E1	1	0	0	E3	0	0	1

Creation of solution steps

This tutorial has a single dynamic solution step. The time of simulation is defined as 70 seconds because it should be enough to establish a steady state oscillation pattern.

SolutionSteps1						
Dynamic	1					
EndTime	70					
TimeStep	0.01					
MaxTimeStep	0.01					
MinTimeStep	0.00001					
MaxIt	20					
MinIt	3					
ConvIncrease	4					
IncFactor	1.0					
Sample	10					
RayleighDamping	Alpha	0	Beta	0	Update	0
NewmarkCoefficients	Beta	0.3	Gamma	0.5		

Creation of loads

The external load F(t) applied on the first mass (node 1) is created establishing a time series.

$$F(t) = \rho_0 \sin(\bar{\omega}t + \varphi);$$

Where $\rho_0 = 25,000$ N is the load's amplitude, $\bar{\omega} = 4.712$ rad/s is the load's angular frequency and $\varphi = 0$ rad is the load's initial phase.

The time step used for time series is 0.05 s. Note that Giraffe performs automatically a linear interpolation between input data in a table form, as shown in the input file extract, next. Thus,

if one needs a higher resolution in a time-series, there is a need for establishing an input file with more data times. Note also that an arbitrary time-series may be input. Our choice, for this tutorial, was a harmonic time-series.

```
Loads 1
NodalLoad 1      NodeSet 2      CS 1      NTimes 1401
//Time FX   FY   FZ   MX   MY   MZ
Time 0       0     0     0     0     0
Time 0.05    5836.134096 0     0     0     0
Time 0.1     11349.76249 0     0     0     0
Time 0.15    16236.20121 0     0     0     0
Time 0.2     20225.42486 0     0     0     0
...
Time 69.85   16236.20121 0     0     0     0
Time 69.9    11349.76249 0     0     0     0
Time 69.95   5836.134097 0     0     0     0
Time 70      1.53355E-07 0     0     0     0
```

Creation of Solver Options

Here it is possible to choose the number of processors to be used. Besides, we must define the solver type for systems of linear equations. In this case, only one processor is employed, and the direct type of linear system solver is chosen.

```
SolverOptions
Processors 1      LinSys Direct
```

Creation of Monitors

Here we request Giraffe to monitor all the nodes during the simulation, for post-processing purposes.

```
Monitor      Sample 10
MonitorNodes 1      2      3
```

Creation of post files

The options used here are the same as the tutorial 01, where the WriteMesh, WriteRenderMesh and WriteForces are chosen for post-processing using Paraview™ and the MagFactor parameters assume a unit value so the deformed shape of Paraview™ frames will reflect the real geometric scale.

```
PostFiles
MagFactor 1
WriteMesh 1
WriteRenderMesh 1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces 0
WriteForces 1
WriteConstraints 0
WriteSpecialConstraints 0
WriteContactForces 0
WriteRenderRigidBodies 0
WriteRenderParticles 0
```

Results and post-processing

To post-process results using Paraview™, first we must open the files “whole_solution_forces.pvd”, “whole_solution_mesh.pvd” and “whole_solution_rendermesh.pvd”, which are located at the folder “post” in “tutorial11”.

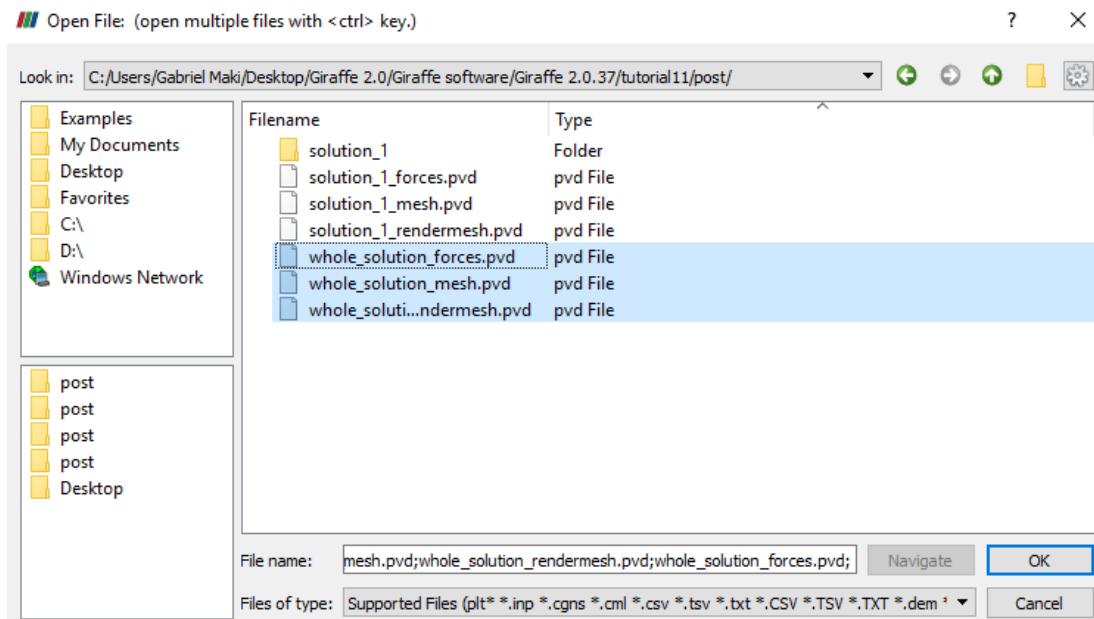


Figure 11-2 - Opening files of tutorial 11 for post-processing

Then, we must click the preview icon and the Apply button on the Pipeline browser.

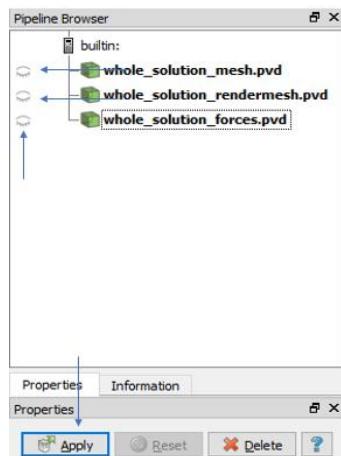


Figure 11-3 - Pipeline browser for tutorial 11.

We may use an isometric perspective and set a two-dimensional view to obtain a better view.

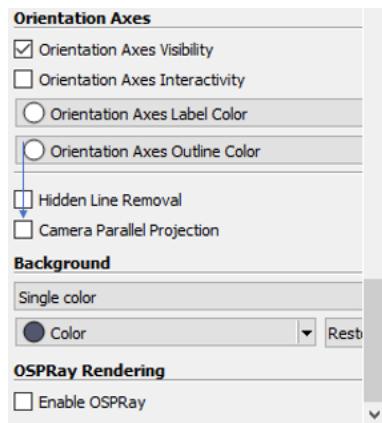


Figure 11-4 - Setting an isometric perspective.



Figure 11-5 - Defining the view direction, for a two-dimensional visualization.

After doing that, the Figure 11-6 shows how would be the view.

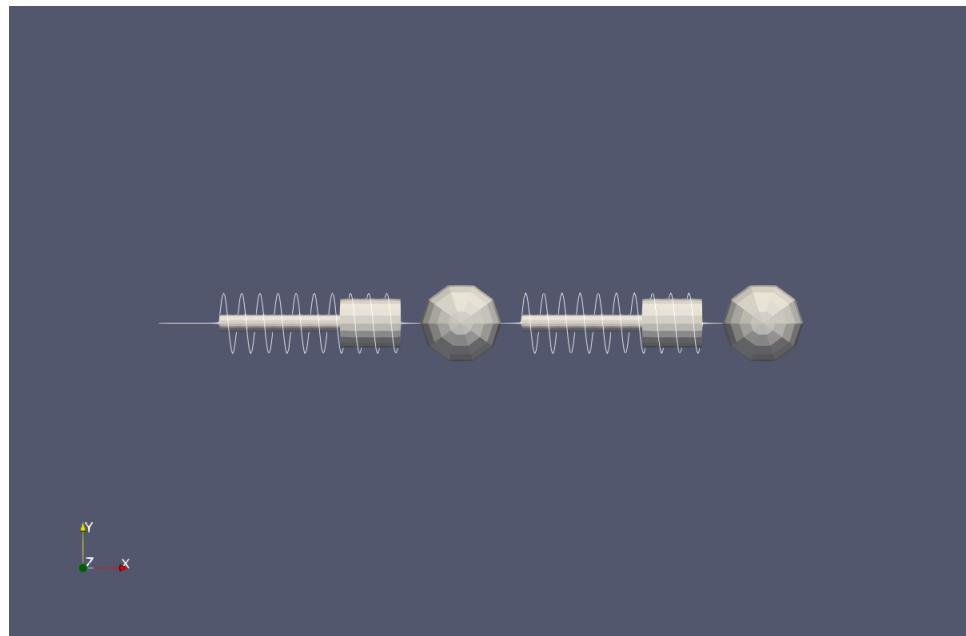


Figure 11-6 - Results of tutorial 11 using Paraview™.

Next, we can create a Glyph for observing the load being applied. For that, go to the Pipeline Browser and click in “whole_solution_forces.pvd”. Then, click the Glyph, as shown in Figure 11-7.

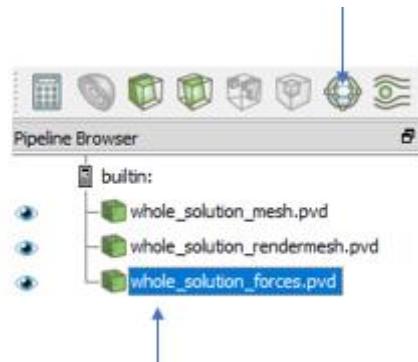


Figure 11-7 - Creation of a Glyph on tutorial 11.

After creating the Glyph, we must adjust the scale of vectors. For that, go to the Pipeline Browser and, first, click the preview icon of Glyph. Then, in Scaling, set the scale mode to vector and change the factor to a small number, e.g.: 0.00002.

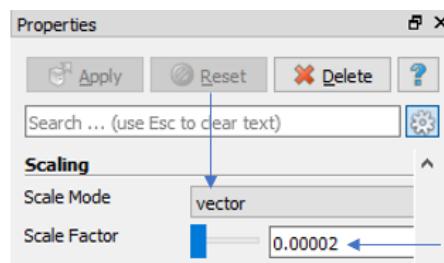


Figure 11-8 - Adjusting the Glyph on tutorial 11.

Obs: If the user wants to change the arrow's color, the one can do it by clicking in Edit, on Coloring at Glyph's Properties.

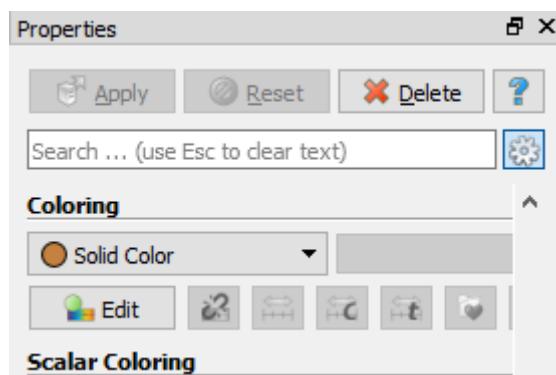
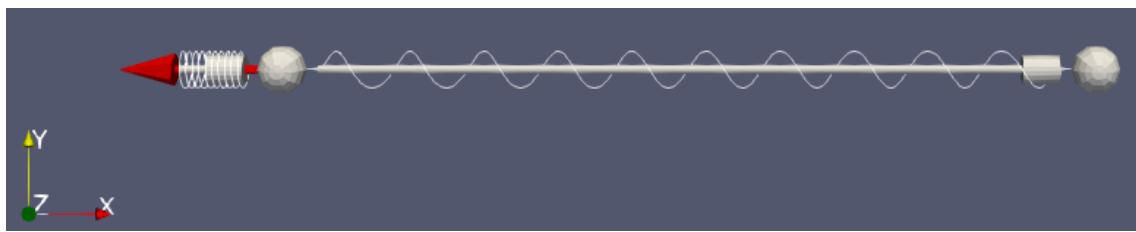
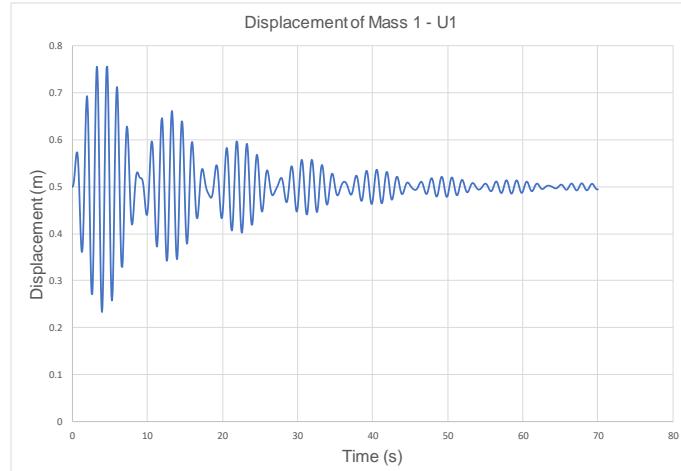
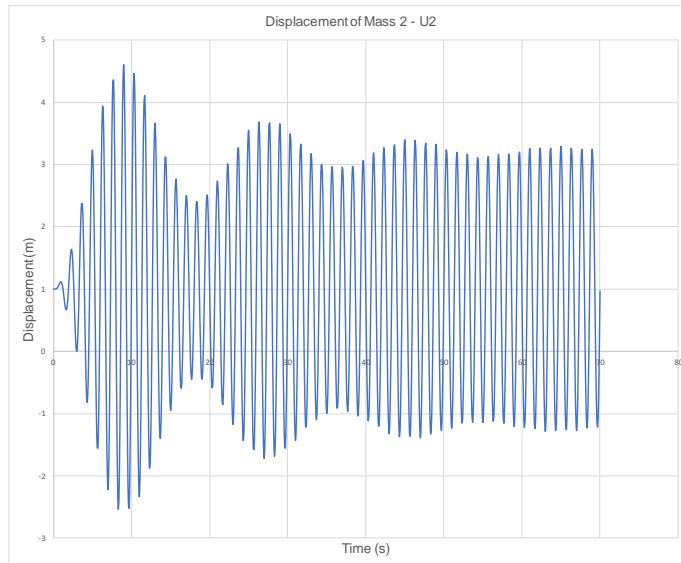


Figure 11-9 - Changing the arrow's color on tutorial 11.

Finally, we can perform an animation to observe the system's movement. When needing to plot time series of displacement, forces or other kinematical quantities, monitors may be helpful.

**Figure 11-10 - System's oscillation.****Figure 11-11 - U₁ (M₁ Displacement).****U₂ (M₂ Displacement)****Figure 11-12 - U₂ (M₂ Displacement).**

One interesting thing to observe is the order of magnitude of displacements. M₂ has displacements much larger than M₁. Besides, it is possible to view that U₁ experiences greater damping, compared to U₂.

If we model only M₁ like shown in Figure 11-13, the system will resonate because the load's frequency is equal to the natural frequency. Figure 11-14 presents the U₁ behavior in this case.

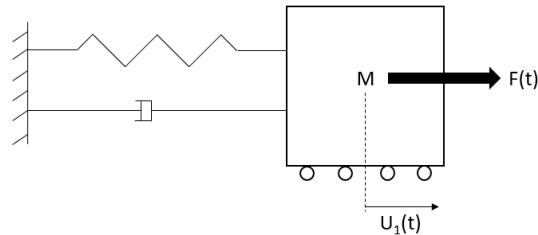


Figure 11-13 - Geometric description of the system of one degree of freedom.

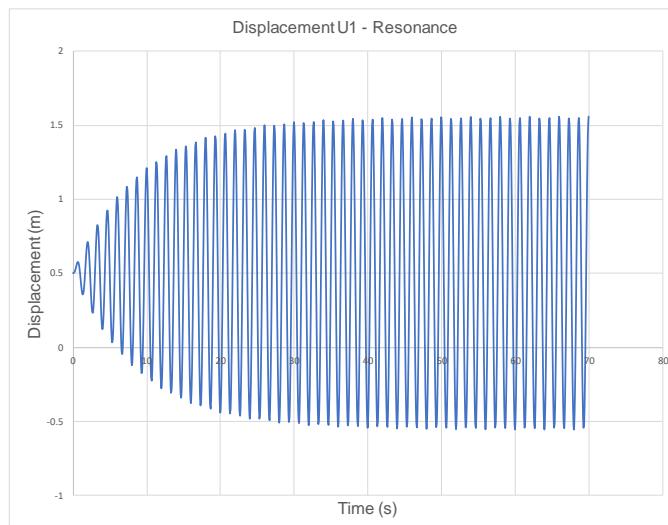


Figure 11-14 - Displacement of M_1 – System of one degree of freedom.

In Figure 11-14, it is possible to see that the displacement increases until establish a steady state (due to damping), which is characteristic of damped 1 degree-of-freedom resonant system. By including the M_2 , M_1 stops vibrating on resonance and the displacements decreases over time, like Figure 11-11. Due to this effect that M_2 has on the system, we call it a tuned mass damper or harmonic absorber. It is an auxiliary device used to reduce the amplitude of vibration of a structure to an acceptable level.

Tutorial 12. Simplified bogie dynamics

Input file name: tutorial12.inp

Developed by Gabriel Maki

Objective

This tutorial presents a dynamic simulation of a bogie. The aim is to show how to model rigid bodies, suspensions and identify the contact between two surfaces. Besides, the tutorial has the objective of study the bogie's movement and the hunting phenomenon in railway systems.

Description

The bogie is meshed considering 34 nodes and 24 elements. It is made up of five rigid bodies, the bolster, the left-side and right-side frames and the frontal and back wheelsets. All the bodies are connected by suspensions. A lumped mass is included in the model to represent half of the wagon mass. It is rigidly linked to the bolster. For the simulation, a lateral nodal load is applied on the bolster barycenter to induce the hunting oscillations. The simulation will be done in 4 solution steps.

Input file step-by-step

Creation of nodes

A total of 34 nodes is created to establish a finite element mesh. The nodes 1 to 13 and 28 to 34 were created to represent the rigid bodies, while the nodes 14 to 27 were created to establish connections between bodies.

```

Nodes 34
//Bolster
Node 1 -1 0 0.5
//Sideframe - left
Node 2 -1.00219 1.2 0.47834
//Sideframe - right
Node 3 -1.00219 -1.2 0.47834
//Wheelset - front
Node 4 0.01347 0 0.4123
//Wheelset - back
Node 5 -1.91533 0 0.4123
//Nodes - Sideframe - left
Node 6 -0.08653 1.2 0.8925
Node 7 -1.81533 1.2 0.8925
//Nodes - Sideframe - right
Node 8 -0.08653 -1.2 0.8925
Node 9 -1.81533 -1.2 0.8925
//Nodes - Wheelset - front
Node 10 0.01347 1.0095 0.4123
Node 11 0.01347 -1.0095 0.4123
//Nodes - Wheelset - back
Node 12 -1.91533 1.0095 0.4123
Node 13 -1.91533 -1.0095 0.4123
//Nodes - Secondary suspension longitudinal direction
Node 14 -1.00093 1.0095 0.5
Node 15 -1.00093 -1.0095 0.5
//Nodes - primary suspension
Node 16 0.01347 1.2 0.8925
Node 17 0.01347 -1.2 0.8925
Node 18 -1.91533 1.2 0.8925
Node 19 -1.91533 -1.2 0.8925

```

```

Node 20 0.01347 1.0095 0.8925
Node 21 -1.91533 1.0095 0.8925
Node 22 0.01347 -1.0095 0.8925
Node 23 -1.91533 -1.0095 0.8925
//Nodes - secondary suspension
Node 24 -1.00093 1.2 0.5
Node 25 -1.00093 -1.2 0.5
Node 26 -1.00093 1.2 0.1952
Node 27 -1.00093 -1.2 0.1952
//Rails
Node 28 -2.5 0.9662 -0.114205
Node 29 2500 0.9662 -0.114205
Node 30 -2.5 -0.9662 -0.114205
Node 31 2500 -0.9662 -0.114205
//Wagon barycenter
Node 32 -1.00093 0 2.1154
//Nodes - Bolster
Node 33 -1 1.0095 0.5
Node 34 -1 -1.0095 0.5

```

Creation of node sets

13 node sets are created to establish the constraints, loads and special constraints.

```

NodeSets 13
//Wheelset - front
NodeSet 1 Nodes 2 List 10 11
//Wheelset - back
NodeSet 2 Nodes 2 List 12 13
//Sideframe - left
NodeSet 3 Nodes 3 List 6 7 26
//Sideframe - right
NodeSet 4 Nodes 3 List 8 9 27
//Bolster
NodeSet 5 Nodes 2 List 33 34
//BC - wheelset front
NodeSet 6 Nodes 1 List 4
//BC - wheelset back
NodeSet 7 Nodes 1 List 5
//Load - bolster
NodeSet 8 Nodes 1 List 1
//Rails
NodeSet 9 Nodes 1 List 28
NodeSet 10 Nodes 1 List 29
NodeSet 11 Nodes 1 List 30
NodeSet 12 Nodes 1 List 31
NodeSet 13 Nodes 1 List 32

```

Creation of special constraints

26 special constraints are defined to represent the rigid node sets and the suspensions. To define suspensions, it is necessary to create a translational joint between nodes that link the bodies. This joint allows that two nodes to have relative displacements in a single direction, defined by the direction E3 of a local coordinate system. Besides, we must define a rotational connection between the bodies, so we use the Same Rotation constraint.

```

SpecialConstraints 26
//Wheelset - front
RigidNodeSet 1 PilotNode 4 NodeSet 1
//Wheelset - back
RigidNodeSet 2 PilotNode 5 NodeSet 2

```

```

//Sideframe - left
RigidNodeSet 3      PilotNode    2      NodeSet      3
//Sideframe - right
RigidNodeSet 4      PilotNode    3      NodeSet      4
//Bolster
RigidNodeSet 5      PilotNode    1      NodeSet      5
//Primary Suspension - vertical
TranslationalJoint 6      Nodes 10    20     RotationNode 2      CS      3
TranslationalJoint 7      Nodes 11    22     RotationNode 3      CS      3
TranslationalJoint 8      Nodes 12    21     RotationNode 2      CS      3
TranslationalJoint 9      Nodes 13    23     RotationNode 3      CS      3
//Primary Suspension - lateral
TranslationalJoint 10     Nodes 20    16     RotationNode 2      CS      4
TranslationalJoint 11     Nodes 22    17     RotationNode 3      CS      4
TranslationalJoint 12     Nodes 21    18     RotationNode 2      CS      4
TranslationalJoint 13     Nodes 23    19     RotationNode 3      CS      4
//Primary Suspension - longitudinal
TranslationalJoint 14     Nodes 6     16     RotationNode 2      CS      5
TranslationalJoint 15     Nodes 8     17     RotationNode 3      CS      5
TranslationalJoint 16     Nodes 7     18     RotationNode 2      CS      5
TranslationalJoint 17     Nodes 9     19     RotationNode 3      CS      5
//Secondary Suspension - vertical
TranslationalJoint 18     Nodes 26    24     RotationNode 2      CS      3
TranslationalJoint 19     Nodes 27    25     RotationNode 3      CS      3
//Secondary Suspension - lateral
TranslationalJoint 20     Nodes 14    24     RotationNode 2      CS      4
TranslationalJoint 21     Nodes 15    25     RotationNode 3      CS      4
//Hinge Joints
SameRotation 22          Nodes 1 2
SameRotation 23          Nodes 1 3
//Wagon barycenter - rigid node set fixed to bolster
RigidNodeSet 24      PilotNode    1      NodeSet      13
//Secondary Suspension - longitudinal
TranslationalJoint 25     Nodes 34    15     RotationNode 3      CS      5
TranslationalJoint 26     Nodes 33    14     RotationNode 2      CS      5

```

Creation of elements

24 elements are created to compose the mesh. Here we use three different types of element. The first type is the RigidBody_1, an element that create a rigid body based on a single node, normally chosen as the barycenter of the body (but not necessarily). The second type is the SpringDashpot_1 element, used to create a spring and damper assembly in parallel. The third type is the Mass_1, which is used to create a lumped mass.

```

Elements 24
RigidBody_1 1 RigidBodyData    1      CS      1      Node    1
RigidBody_1 2 RigidBodyData    2      CS      1      Node    2
RigidBody_1 3 RigidBodyData    3      CS      1      Node    3
RigidBody_1 4 RigidBodyData    4      CS      1      Node    4
RigidBody_1 5 RigidBodyData    4      CS      1      Node    5
//Pad - vertical
SpringDashpot_1 6 Stiffness 7000000 Damping 200000 Nodes 10    20
SpringDashpot_1 7 Stiffness 7000000 Damping 200000 Nodes 11    22
SpringDashpot_1 8 Stiffness 7000000 Damping 200000 Nodes 12    21
SpringDashpot_1 9 Stiffness 7000000 Damping 200000 Nodes 13    23
//Pad - lateral
SpringDashpot_1 10 Stiffness 150000000 Damping 20000 Nodes 20    16
SpringDashpot_1 11 Stiffness 150000000 Damping 20000 Nodes 22    17
SpringDashpot_1 12 Stiffness 150000000 Damping 20000 Nodes 21    18
SpringDashpot_1 13 Stiffness 150000000 Damping 20000 Nodes 23    19
//Pad - longitudinal
SpringDashpot_1 14 Stiffness 140000000 Damping 20000 Nodes 16    6
SpringDashpot_1 15 Stiffness 140000000 Damping 20000 Nodes 17    8
SpringDashpot_1 16 Stiffness 140000000 Damping 20000 Nodes 18    7

```

```

SpringDashpot_1 17 Stiffness 140000000 Damping 20000 Nodes 19      9
//Secondary - vertical
SpringDashpot_1 18 Stiffness 1700000 Damping 0 Nodes 26      24
SpringDashpot_1 19 Stiffness 1700000 Damping 0 Nodes 27      25
//Secondary - lateral
SpringDashpot_1 20 Stiffness 130000000 Damping 0 Nodes 14      24
SpringDashpot_1 21 Stiffness 130000000 Damping 0 Nodes 15      25
Mass_1 22 Mass 55000 Node 32
//Secondary - longitudinal
SpringDashpot_1      23 Stiffness 170000000      Damping 200000 Nodes 34
      15
SpringDashpot_1      24 Stiffness 170000000      Damping 200000 Nodes 33
      14

```

Creation of rigid body data

Here we insert the mass and inertia properties of each kind of rigid body. Besides, we can request Giraffe to use a CAD file in the simulation to represent the rigid body (for post-processing purposes). For that, one may use any CAD which is able to export in .stl ASCII file to develop the geometry.

```

RBDATA 1
Mass 800
J11 930 J22 22 J33 480 J12 0.0 J13      0.0 J23 0.0
Barycenter 0.0 0.0 0.0
CADFile BolsterTutorial.stl
RBDATA 2
Mass 700
J11 60 J22 340 J33      300 J12 0.0 J13      0.0 J23 0.0
Barycenter 0.0 0.0 0.0
CADFile SideleftTutorial.stl
RBDATA 3
Mass 700
J11 60 J22 340 J33      300 J12 0.0 J13      0.0 J23 0.0
Barycenter 0.0 0.0 0.0
CADFile SiderightTutorial.stl
RBDATA 4
Mass 1550
J11 1050 J22 120 J33 1050 J12 0.0 J13  0.0 J23 0.0
Barycenter 0.0 0.0 0.0
CADFile EixoTutorial.stl

```

Creation of coordinate systems

Nine coordinate systems are defined considering all the constraints and surfaces that will be created after. Note that CS 6 and CS 7 have some different directions because they are used to define the rail cant (inclination). They are made so that the direction E2 stays parallel to rail cross sections. In this case, the cant used is 1:20.

```

CoordinateSystems 9
//Global CS
CS    1     E1    1     0     0     E3    0     0     1
//Wheelset CS - E1 aligned with wheelset axle
CS    2     E1    0     1     0     E3    0     0     1
//Suspension - vertical
CS    3     E1    1     0     0     E3    0     0     1
//Suspension - lateral
CS    4     E1    1     0     0     E3    0     1     0
//Suspension - longitudinal
CS    5     E1    0     1     0     E3    1     0     0

```

```
//Rails
CS 6 E1 0 0.998752339179 4.99376109278e-2    E3     1     0     0
CS 7 E1 0 -0.998752339179 4.99376109278e-2    E3    -1     0     0

//Wheels
CS 8 E1      0      0      -1 E3 1      0      0
CS 9 E1      0      0      -1 E3 -1      0      0
```

Creation of arcs

Here we insert the circumference arcs for the creation of contact surfaces. Note that input data include: initial, final and center points of the arc in a plan. The coordinate system and the node established as the system origin are defined on the creation of each surface.

```
//Creation of arcs
Arcs 14
//Wheel arcs
Arc  1   InitialPoint 0.425796370582452  0.882054430691546
FinalPoint 0.443208542157583  0.889979653581409
CenterPoint 0.421579962014195  0.914408507213130
Arc  2   InitialPoint 0.443208542157585  0.889979653581408
FinalPoint 0.446742276607354  0.897460714881545
CenterPoint 0.436246087964145  0.897843543549290
Arc  3   InitialPoint 0.446742276607329  0.897460714881546
FinalPoint 0.442401755983930  0.908021749519942
CenterPoint 0.432948005503375  0.897963834881460
Arc  4   InitialPoint 0.442401755983959  0.908021749519973
FinalPoint 0.434723623779321  0.912588822523442
CenterPoint 0.428246647673200  0.892962025722420
Arc  5   InitialPoint 0.426919157227811  0.918935831513070
FinalPoint 0.434723623779279  0.912588822523315
CenterPoint 0.439150899172595  0.926004532741030
Arc  6   InitialPoint 0.421791943301005  0.935976728599755
FinalPoint 0.426919157227897  0.918935831513120
CenterPoint 0.460460922049591  0.938319555863530
Arc  7   InitialPoint 0.421791943301296  0.935976728599773
FinalPoint 0.415483689711215  1.00785311080860
CenterPoint -0.910691689680170  0.855245892541840
Arc  8   InitialPoint 0.415483689711318  1.00785311080870
FinalPoint 0.413972014807964  1.019925467434800
CenterPoint -0.188853650451404  0.938310091375720
Arc  9   InitialPoint 0.413972014807492  1.019925467434730
FinalPoint 0.400805033000113  1.027311927799740
CenterPoint 0.404605274257736  1.018657322933800
//Rails arcs
Arc  10  InitialPoint 0.03685759800334810  0.08526439800008370
FinalPoint 0.02779624699522180  0.09820644298783040
CenterPoint 0.022574560946790100  0.084907320972669800
Arc  11  InitialPoint 0.02779624699875580  0.09820644299683120
FinalPoint 0.01777999999465240  0.10036312589318200
CenterPoint 0.01619249977804300  0.068652837381108900
Arc  12  InitialPoint 0.01777999999754500  0.10036312595096100
FinalPoint -0.01778000000000110  0.10036312600002300
CenterPoint -0.000000000491233737  -0.254792066634406000
Arc  13  InitialPoint -0.01778000000076080  0.10036312601519700
FinalPoint -0.02779624703571790  0.09820644309097000
CenterPoint -0.016192499912296400  0.068652838315372000
Arc  14  InitialPoint -0.02779624701583830  0.09820644304033850
FinalPoint -0.03685759803196000  0.08526439802029900
CenterPoint -0.022574560818925100  0.084907320913286000
```

Creation of surfaces

Here we request Giraffe to create surfaces based on the arcs defined previously. There are resources in Giraffe that uses arc definitions to establish extruded/revolved surfaces. For the wheel, we must use the RigidArcRevolution_1 (revolution surface). For the rail, we must use the FlexibleArcExtrusion_1 (extrusion surface).

Surfaces	46							
//Left front wheel								
RigidArcRevolution_1	1	Arc	1	CS	8	Node	4	
Convex //interior								
RigidArcRevolution_1	2	Arc	2	CS	8	Node	4	
Convex								
RigidArcRevolution_1	3	Arc	3	CS	8	Node	4	
Convex								
RigidArcRevolution_1	4	Arc	4	CS	8	Node	4	
Convex								
RigidArcRevolution_1	5	Arc	5	CS	8	Node	4	
Concave								
RigidArcRevolution_1	6	Arc	6	CS	8	Node	4	
Concave								
RigidArcRevolution_1	7	Arc	7	CS	8	Node	4	
Convex								
RigidArcRevolution_1	8	Arc	8	CS	8	Node	4	
Convex								
RigidArcRevolution_1	9	Arc	9	CS	8	Node	4	
Convex //exterior								
//Right front wheel								
RigidArcRevolution_1	10	Arc	1	CS	9	Node	4	
Convex //interior								
RigidArcRevolution_1	11	Arc	2	CS	9	Node	4	
Convex								
RigidArcRevolution_1	12	Arc	3	CS	9	Node	4	
Convex								
RigidArcRevolution_1	13	Arc	4	CS	9	Node	4	
Convex								
RigidArcRevolution_1	14	Arc	5	CS	9	Node	4	
Concave								
RigidArcRevolution_1	15	Arc	6	CS	9	Node	4	
Concave								
RigidArcRevolution_1	16	Arc	7	CS	9	Node	4	
Convex								
RigidArcRevolution_1	17	Arc	8	CS	9	Node	4	
Convex								
RigidArcRevolution_1	18	Arc	9	CS	9	Node	4	
Convex //exterior								
//Left rail								
FlexibleArcExtrusion_1	19	Arc	10	CS	6	Nodes	28	29
Convex								
FlexibleArcExtrusion_1	20	Arc	11	CS	6	Nodes	28	29
Convex								
FlexibleArcExtrusion_1	21	Arc	12	CS	6	Nodes	28	29
Convex								
FlexibleArcExtrusion_1	22	Arc	13	CS	6	Nodes	28	29
Convex								
FlexibleArcExtrusion_1	23	Arc	14	CS	6	Nodes	28	29
Convex								
//Right rail								
FlexibleArcExtrusion_1	24	Arc	10	CS	7	Nodes	31	30
Convex								

FlexibleArcExtrusion_1 Convex	25	Arc	11	CS	7	Nodes	31	30
FlexibleArcExtrusion_1 Convex	26	Arc	12	CS	7	Nodes	31	30
FlexibleArcExtrusion_1 Convex	27	Arc	13	CS	7	Nodes	31	30
FlexibleArcExtrusion_1 Convex	28	Arc	14	CS	7	Nodes	31	30
//Left back wheel								
RigidArcRevolution_1 Convex //interior	29	Arc	1	CS	8	Node	5	
RigidArcRevolution_1 Convex	30	Arc	2	CS	8	Node	5	
RigidArcRevolution_1 Convex	31	Arc	3	CS	8	Node	5	
RigidArcRevolution_1 Convex	32	Arc	4	CS	8	Node	5	
RigidArcRevolution_1 Concave	33	Arc	5	CS	8	Node	5	
RigidArcRevolution_1 Concave	34	Arc	6	CS	8	Node	5	
RigidArcRevolution_1 Convex	35	Arc	7	CS	8	Node	5	
RigidArcRevolution_1 Convex	36	Arc	8	CS	8	Node	5	
RigidArcRevolution_1 Convex //exterior	37	Arc	9	CS	8	Node	5	
//Right back wheel								
RigidArcRevolution_1 Convex //interior	38	Arc	1	CS	9	Node	5	
RigidArcRevolution_1 Convex	39	Arc	2	CS	9	Node	5	
RigidArcRevolution_1 Convex	40	Arc	3	CS	9	Node	5	
RigidArcRevolution_1 Convex	41	Arc	4	CS	9	Node	5	
RigidArcRevolution_1 Concave	42	Arc	5	CS	9	Node	5	
RigidArcRevolution_1 Concave	43	Arc	6	CS	9	Node	5	
RigidArcRevolution_1 Convex	44	Arc	7	CS	9	Node	5	
RigidArcRevolution_1 Convex	45	Arc	8	CS	9	Node	5	
RigidArcRevolution_1 Convex //exterior	46	Arc	9	CS	9	Node	5	

Creation of surface sets

After the creation of surfaces, we must define surface sets to be used in contact pairs establishment. On this tutorial, we have six different surface sets (four wheels and two rails).

SurfaceSets	6							
SurfaceSet	1	Surfaces	5	List	19	20	21	22
SurfaceSet	2	Surfaces	9	List	1	2	3	4
	6	8	9					5
SurfaceSet	3	Surfaces	5	List	24	25	26	27
SurfaceSet	4	Surfaces	9	List	10	11	12	13
	15	17	18					14
SurfaceSet	5	Surfaces	9	List	29	30	31	32
	34	36	37					33
SurfaceSet	6	Surfaces	9	List	38	39	40	41
	43	45	46					42

Creation of contacts

Here we define contact pairs. These are composed by combination of surface sets for which wheel/rail possibility. The total is four contact pairs.

Contacts	4
SSSS	1 SurfaceSet1 1 SurfaceSet2 2 MU 0.3 EPN 5e8
CN	5e5 EPT 5e7 CT 0 Pinball 1.0e6
BoolTable	1
SSSS	2 SurfaceSet1 3 SurfaceSet2 4 MU 0.3 EPN 5e8
CN	5e5 EPT 5e7 CT 0 Pinball 1.0e6
BoolTable	1
SSSS	3 SurfaceSet1 1 SurfaceSet2 5 MU 0.3 EPN 5e8
CN	5e5 EPT 5e7 CT 0 Pinball 1.0e6
BoolTable	1
SSSS	4 SurfaceSet1 3 SurfaceSet2 6 MU 0.3 EPN 5e8
CN	5e5 EPT 5e7 CT 0 Pinball 1.0e6
BoolTable	1

Creation of solution steps

This tutorial has four solution steps. All the steps are dynamics. The first, ends at 0.5 s and the time step is 0.1 s because it is the beginning of simulation where the bogie's movement is a free fall (prior to contact interaction with the rails). From the second step on the contact begins to occur and the time step is reduced to 0.002 s. During the second step, the bogie accommodates on the rail. At the beginning of the third step, an initial velocity is imposed on the bogie and the simulation finishes at the fourth step after 150 s of movement.

```

SolutionSteps 4
Dynamic      1
EndTime 0.5
TimeStep 0.1
MaxTimeStep 0.1
MinTimeStep 1e-7
MaxIt 15
MinIt 3
ConvIncrease 2
IncFactor 1.4
Sample 100
RayleighDamping   Alpha 0     Beta 0      Update 0
NewmarkCoefficients Beta 0.3 Gamma 0.5
Dynamic      2
EndTime 6
TimeStep 0.002
MaxTimeStep 0.02
MinTimeStep 1e-7
MaxIt 15
MinIt 3
ConvIncrease 2
IncFactor 1.4
Sample 100
RayleighDamping   Alpha 0     Beta 0      Update 0
NewmarkCoefficients Beta 0.3 Gamma 0.5
Dynamic      3
EndTime 16.2
TimeStep 0.002
MaxTimeStep 0.002
MinTimeStep 1e-9
MaxIt 15
MinIt 3

```

```

ConvIncrease 2
IncFactor 1.4
Sample 100
RayleighDamping      Alpha 0      Beta 0      Update 0
NewmarkCoefficients Beta 0.3 Gamma 0.5
Dynamic 4
EndTime 150
TimeStep 0.002
MaxTimeStep 0.002
MinTimeStep 1e-9
MaxIt 15
MinIt 3
ConvIncrease 2
IncFactor 1.4
Sample 10
RayleighDamping      Alpha 0      Beta 0      Update 0
NewmarkCoefficients Beta 0.3 Gamma 0.5

```

Creation of initial conditions

At the beginning of the third solution step, the bogie starts to move. We define initial conditions of translational velocity for all the nodes of the model, such as angular velocities to the wheelsets.

```

//Creation of initial conditions
InitialConditions 30
InitialCondition    1      Node 1      DU      1.6      0      0      OMEGA 0
      0      0      SolutionStep 3
InitialCondition    2      Node 2      DU      1.6      0      0      OMEGA 0
      0      0      SolutionStep 3
InitialCondition    3      Node 3      DU      1.6      0      0      OMEGA 0
      0      0      SolutionStep 3
...

```

Creation of constraints

Four nodal constraints are created to fix all the degrees of freedom of rail nodes.

```

//Constraints
Constraints 4
NodalConstraint    1      NodeSet 9
      UX  BoolTable 1
      UY  BoolTable 1
      UZ  BoolTable 1
      ROTX  BoolTable 1
      ROTY  BoolTable 1
      ROTZ  BoolTable 1
NodalConstraint    2      NodeSet 10
      UX  BoolTable 1
      UY  BoolTable 1
      UZ  BoolTable 1
      ROTX  BoolTable 1
      ROTY  BoolTable 1
      ROTZ  BoolTable 1
NodalConstraint    3      NodeSet 11
      UX  BoolTable 1
      UY  BoolTable 1
      UZ  BoolTable 1
      ROTX  BoolTable 1
      ROTY  BoolTable 1
      ROTZ  BoolTable 1
NodalConstraint    4      NodeSet 12

```

UX	BoolTable	1
UY	BoolTable	1
UZ	BoolTable	1
ROTX	BoolTable	1
ROTY	BoolTable	1
ROTZ	BoolTable	1

Creation of loads

A single nodal load is created. This is done establishing a time varying force in direction Y. This lateral force is applied to induce an oscillatory movement of the bogie. The force is applied on the bolster barycenter.

```
//Creation of loads
Loads 1
NodalLoad 1 NodeSet 8 CS 1 NTimes 3
//Time FX FY FZ MX MY MZ
Time 10.0 0 0 0 0 0
Time 20.0 0 20000 0 0 0
Time 30.0 0 0 0 0 0
```

Creation of environment data

An environment data is created to introduce the gravity field on the model.

```
Environment
GravityData
G 0 0 -9.81 BoolTable 1
```

Creation of solver options

Here we request Giraffe to apply six threads to solve this case. The linear system solution chosen is Direct.

```
SolverOptions
Processors 6 LinSys Direct
```

Creation of monitors

We request Giraffe to monitor three nodes (Bolster and wheelsets barycenter) and the four contacts.

```
Monitor Sample 10
MonitorNodes 1 4 5
MonitorContacts 1 2 3 4
```

Creation of post files

We choose the MagFactor parameter to be a unit value so that the Paraview™ frames reflect the real scale. Besides, the WriteRigidContactSurfaces, WriteFlexibleContactSurface, WriteSpecialConstraints, WriteContactForces and WriteRenderRigidBodies are chosen for post-

processing. If the user wants to reduce Giraffe's runtime, she/he may not choose the WriteRenderRigidBodies because it has a high computational cost.

```
PostFiles
MagFactor      1
WriteMesh       0
WriteRenderMesh 0
WriteRigidContactSurfaces 1
WriteFlexibleContactSurfaces     1
WriteForces     0
WriteConstraints 0
WriteSpecialConstraints 1
WriteContactForces 1
WriteRenderRigidBodies    1
WriteRenderParticles   0
```

Results and post-processing

To post-process data using Paraview™, first we must open the files "whole_solution_contactforces.pvd", "whole_solution_contactsurfaces.pvd" and "whole_solution_rb_particles.pvd".

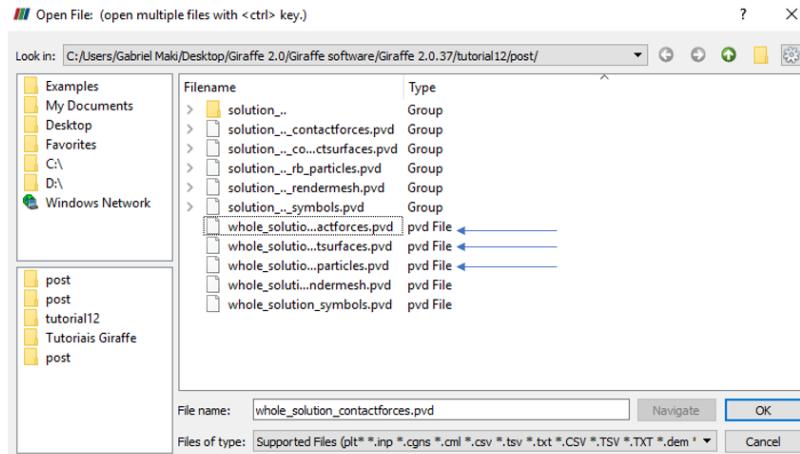


Figure 12-1 - Opening the files of tutorial 12 in Paraview™.

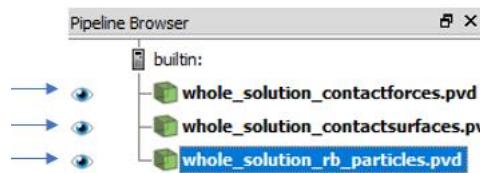


Figure 12-2 - Showing the Pipeline Browser.

For a more convenient view with no distorting dimensions, one may adopt an isometric perspective. For that, go to Properties and check "Camera Parallel Projection in Orientation Axes.

An interesting thing to observe during the simulation is the contact point. To get a better view of that point, the user may select the "whole_solution_contactsurfaces.pvd" on the Pipeline Browser. Then, select "Surface with Edges" like Figure 12-3.

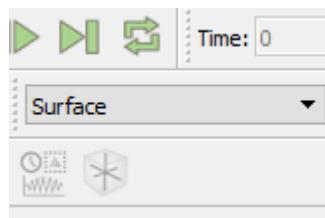


Figure 12-3 - Changing the view to surface with edges.

The Paraview™ will show the model like the Figure 12-4.

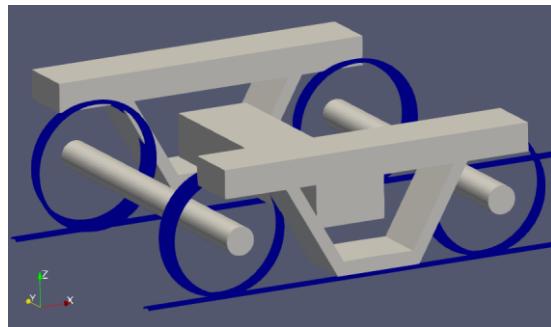


Figure 12-4 - Visualization of results of tutorial 12 in ParaviewTM.

It may be helpful to increase the size point. For that, select “whole_solution_contactforces.pvd” on Pipeline Browser and change the point size at the Properties.

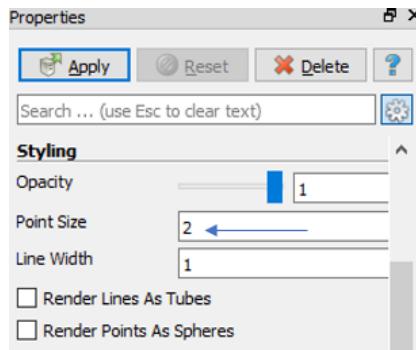


Figure 12-5 - Changing the point size on tutorial 12.

Next, we can create a Glyph to see the symbol of the contact forces. For that, go to the Pipeline Browser, click in “whole_solution_contactforces.pvd” and click in Apply. Then, click the Glyph.

After creating the Glyph, we have to select what kind of force we want to see (normal or friction force). To do this, go to the Glyph Properties and change the Vectors in Active Attributes to normal or friction.

After that, we must adjust the scale of vectors. To do this, go to Scaling, at the Properties, and set the scale mode to vector, change the scale factor to a small number like 0.000001 and click Apply.

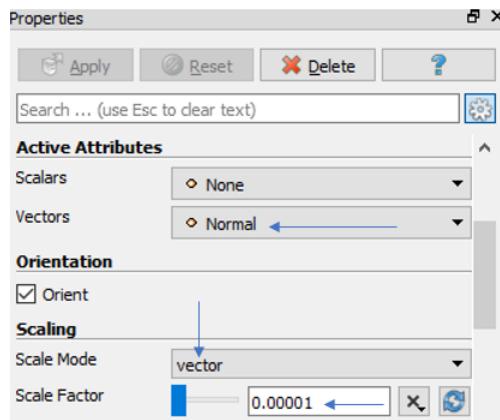


Figure 12-6 - Adjusting the Glyph of tutorial 12.

Finally, we can show an animation of the simulation results. If the user wants to know the values of displacements of some nodes or contact forces in a certain instant, we can gather such information in the “monitors” folder, since they are requested prior to simulation solving.

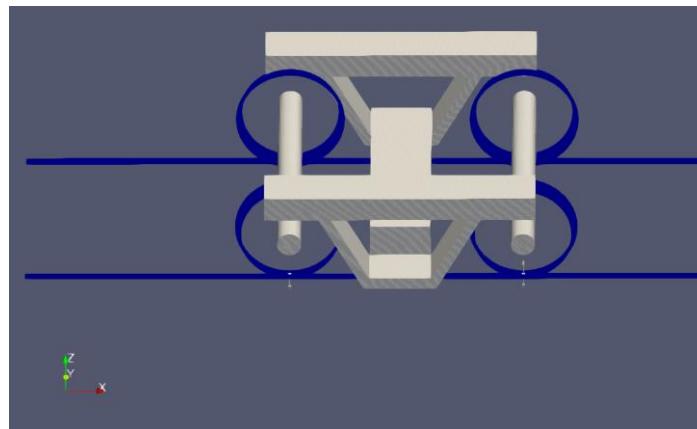


Figure 12-7 - Bogie simulation.

Figure 12-8 shows the lateral displacement of bolster barycenter over time. Note that, after 30 s, when the lateral force stops being applied, the bolster presents an oscillatory movement, which characterizes the hunting phenomenon.

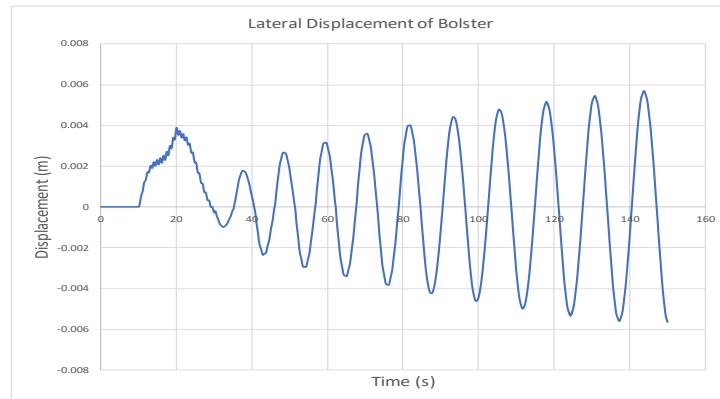


Figure 12-8 - Lateral displacement (UY) of bolster barycenter.

Tutorial 13. Baseball bat hitting the ball

Input file name: tutorial13.inp

Developed by Lucas da Silva

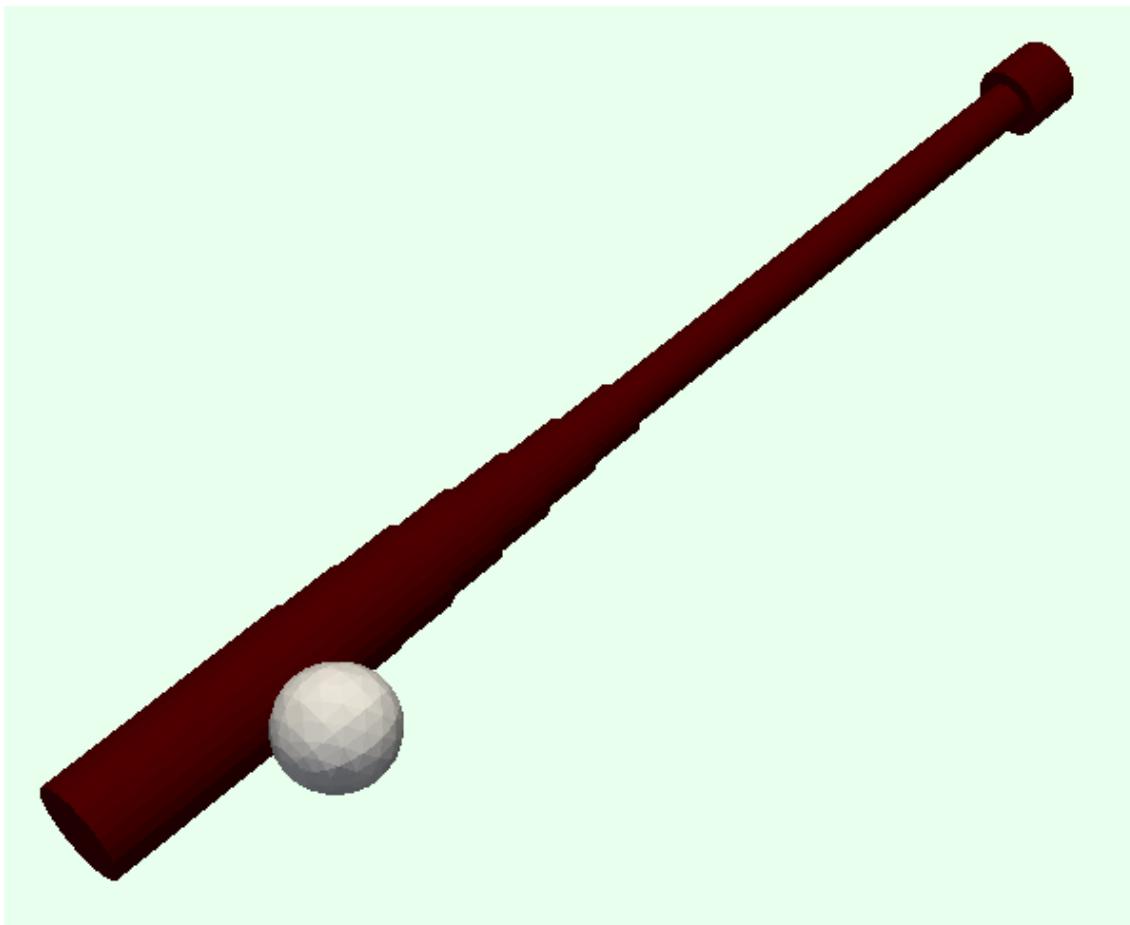


Figure 13-1 – Baseball bat and ball visualization

Objective

This tutorial presents a geometrically-nonlinear simulation of a baseball bat hitting a rigid ball, the two objects have given initial velocities, the bat has initial rotation velocity and the ball has initial linear velocity and spin. The tutorial has the purpose of showing how to set and solve an impact simulation.

Description

The bat (0.84 m of length) is meshed using 41 nodes and 20 elements, using the element Beam_1. The cross-sections of the elements are circular with varying radius to represent the geometry shown in Figure 13-1. The base node of the bat is constrained, only the rotation about the y axis is permitted, our intention is that the motion takes place in the xz plane. The ball is meshed with a single node used in the creation of a rigid spherical particle with given radius.

Input file step-by-step

Creation of nodes

A total of 42 nodes is created to establish a finite element mesh.

```
//Creation of nodes
Nodes 42
//Number   X      Y      Z
//Bat nodes
Node  1    0      0      0.000
Node  2    0      0      0.0021
Node  3    0      0      0.0042
Node  4    0      0      0.0063
Node  5    0      0      0.0084
...
Node  38   0      0      0.777
Node  39   0      0      0.798
Node  40   0      0      0.819
Node  41   0      0      0.840
//Ball node
Node  42   -0.08  0.000  0.71
```

Creation of node sets

Two node sets are created: the first one to establish the base constraints, the second one to establish contact.

```
//Creation of node sets
NodeSets 2
//Base of the bat
NodeSet 1     Nodes 1      List  1
//Ball node
NodeSet 2     Nodes 1      List  42
```

Creation of elements

A total of 20 elements is defined for the bat. All the elements are assigned to the same material properties and same coordinate system, the cross-section is different for each part of the bat.

```
//Creation of elements
Elements 20
//Knob"
Beam_1 1   Mat  1      Sec  1      CS   1      Nodes 1    2    3
//Handle"
Beam_1 2   Mat  1      Sec  2      CS   1      Nodes 3    4    5
Beam_1 3   Mat  1      Sec  2      CS   1      Nodes 5    6    7
Beam_1 4   Mat  1      Sec  2      CS   1      Nodes 7    8    9
...
Beam_1 10  Mat  1      Sec  2      CS   1      Nodes 19   20   21
//Transition"
Beam_1 11  Mat  1      Sec  3      CS   1      Nodes 21   22   23
Beam_1 12  Mat  1      Sec  4      CS   1      Nodes 23   24   25
Beam_1 13  Mat  1      Sec  5      CS   1      Nodes 25   26   27
Beam_1 14  Mat  1      Sec  6      CS   1      Nodes 27   28   29
Beam_1 15  Mat  1      Sec  7      CS   1      Nodes 29   30   31
Beam_1 16  Mat  1      Sec  8      CS   1      Nodes 31   32   33
//Barrel"
Beam_1 17  Mat  1      Sec  9      CS   1      Nodes 33   34   35
...
Beam_1 20  Mat  1      Sec  9      CS   1      Nodes 39   40   41
```

Creation of particles

The ball (node 42) is represented in the model with a rigid spherical particle of radius 0.037 (m). Giraffe evaluates the ball mass automatically, given the properties of the assigned material (2).

```
//Creation of particles
Particles 1
Sphere 1 Mat 2 CS 1 Radius 0.037 Node 42
```

Creation of materials

Two elastic materials are defined: the first one is assigned to the bat elements, it has properties of wood. The second material is assigned to the particle. Since the particle is modeled as a rigid body, its elastic properties are ignored.

```
//Creation of materials
Materials 2
//Wood
Hooke 1 E 18.14e9 Nu 0.3 Rho 650
//Ball
Hooke 2 E 18.14e9 Nu 0.3 Rho 670
```

Creation of cross-sections

A total of 9 cross sections is defined. All of them are SuperEllipse type sections (semi-axes A and B, exponent N) reduced to circular sections (A=B, N=2).

```
//Creation of sections
Sections 9
SuperEllipse 1 A 0.025 B 0.025 N 2 AMeshFDM 100
SuperEllipse 2 A 0.015 B 0.015 N 2 AMeshFDM 100
SuperEllipse 3 A 0.018 B 0.018 N 2 AMeshFDM 100
SuperEllipse 4 A 0.021 B 0.021 N 2 AMeshFDM 100
SuperEllipse 5 A 0.024 B 0.024 N 2 AMeshFDM 100
SuperEllipse 6 A 0.027 B 0.027 N 2 AMeshFDM 100
SuperEllipse 7 A 0.030 B 0.030 N 2 AMeshFDM 100
SuperEllipse 8 A 0.0315 B 0.0315 N 2 AMeshFDM 100
SuperEllipse 9 A 0.033 B 0.033 N 2 AMeshFDM 100
```

Creation of coordinate systems

A single coordinate system is created. It is important to mention that, since employing Beam_1 element, the direction E3 must lie in the beam alignment direction. The identification number of the coordinate system is 1.

```
//Creation of coordinate systems
CoordinateSystems 1
CS 1 E1 1 0 0 E3 0 0 1
```

Creation of solution steps

There are two solution steps, both dynamics. The first step is associated with the impact between the bat and the ball, we force the time steps to be 0.035 ms or less, because the contact time is of about 0.7 ms, so the simulation gets nearly 20 time-steps to model the impact.

The second step takes place right after the first one, it is associated with the system evolution after the collision, the purpose of this step is to permit a greater time step after the impact, when we needed a really small time-step.

```
//Creation of solution steps
/*
Two steps:
1st. Bat with given rotation velocity hitting a ball with given
initial linear velocity and spin
2nd. Evolution of the system post collision
*/
SolutionSteps2
SolutionSteps 2
Dynamic 1
EndTime 0.001
TimeStep 0.000035
MaxTimeStep 0.000035
MinTimeStep 0.00000035
MaxIt 20
MinIt 3
ConvIncrease 5
IncFactor 1.2
Sample 1
RayleighDamping Alpha 0 Beta 0 Update 0
NewmarkCoefficients Beta 0.3 Gamma 0.5
Dynamic 2
EndTime 0.025
TimeStep 0.000035
MaxTimeStep 0.00035
MinTimeStep 0.00000175
MaxIt 20
MinIt 3
ConvIncrease 5
IncFactor 1.2
Sample 1
RayleighDamping Alpha 0 Beta 0 Update 0
NewmarkCoefficients Beta 0.3 Gamma 0.5
```

Creation of initial conditions

A total of 42 initial conditions is created, one for each node. The first initial condition is associated with the ball (node 42), it has an initial velocity of 40.0 (m/s) on the x direction and a rotation of 230 (rad/s) on the y direction.

The following 41 conditions are associated with the bat. To all nodes is assigned the same rotation of $\omega=-50$ (rad/s) on the y direction, and a velocity given by:

$$v = \omega \cdot r,$$

where r is the distance between the node and the base of the bat. With these velocities the bat has rigid body motion of rotation about the Oy axis.

```
//Creation of initial conditions
InitialConditions 42
// Ball initial conditions
InitialCondition 1 Node 42 DU 40.0 0 0 OMEGA 0 230 0 SolutionStep 1
// Bat initial conditions
InitialCondition 2 Node 1 DU -0.00 0 0 OMEGA 0 -50 0 SolutionStep 1
InitialCondition 3 Node 2 DU -1.05 0 0 OMEGA 0 -50 0 SolutionStep 1
InitialCondition 4 Node 3 DU -2.10 0 0 OMEGA 0 -50 0 SolutionStep 1
InitialCondition 5 Node 4 DU -3.15 0 0 OMEGA 0 -50 0 SolutionStep 1
...
InitialCondition 39 Node 38 DU -38.85 0 0 OMEGA 0 -50 0 SolutionStep 1
InitialCondition 40 Node 39 DU -39.90 0 0 OMEGA 0 -50 0 SolutionStep 1
InitialCondition 41 Node 40 DU -40.95 0 0 OMEGA 0 -50 0 SolutionStep 1
InitialCondition 42 Node 41 DU -42.00 0 0 OMEGA 0 -50 0 SolutionStep 1
```

Creation of constraints

The single constraint is applied to node set 1 (the base of the bat). It restricts all motion except for the rotation about the y axis.

```
//Creation of constraints
Constraints 1
NodalConstraint 1 NodeSet 1
UX BoolTable 1
UY BoolTable 1
UZ BoolTable 1
ROTX BoolTable 1
ROTY BoolTable 0
ROTZ BoolTable 1
```

Creation of contact surfaces

One contact surface is defined to enclose the barrel of the bat, the region the ball is supposed to hit. It has similar geometrical properties as the cross-section 9 assigned to the barrel part. The normal to the surface is set to be exterior, pointing to the region where the ball is supposed to be. Just like the barrel elements, the surface goes from node 33 to node 41.

```
//Creation of contact surfaces
Surfaces 1
FlexibleSECylinder 1 1 A 0.033 B 0.033 N 2 CS 1 NormalExterior Nodes 33 41
```

Creation of surface sets

One surface set is required to establish contact, it contains the only surface.

```
//Creation of surface sets
SurfaceSets 1
SurfaceSet 1 Surfaces 1 List 1
```

Creation of contact pairs

There are two interacting bodies, one of them is a spherical object. The Node Set to Surface Set (NSSS) contact is perfect for this problem because it considers that the nodes in the given node set are centers of spherical surfaces of given radius, therefore the only surface to be concerned is the barrel surface that was already defined and linked to the surface set.

The node set 2 (associated with the ball) and the surface set 1 are chosen. Then, some contact parameters are defined:

- friction coefficient (MU)
- normal penalty parameter (EPN)
- normal dissipative coefficient (CN)
- tangential penalty parameter (EPT)
- tangential dissipative coefficient (CT)

The EPN parameter is strongly related with the collision time, the greater its value the smaller the period of contact. The CN parameter is also related with the collision time, but more importantly, with the coefficient of restitution (COR) of the collision, since it establishes a dissipation coefficient, proportional to the relative velocity of contacting bodies' surfaces.

The values of these two parameters were set to establish a contact time about 0.0007 (s) and a COR about 0.33.

The value of EPT is the same as the EPN value with one order of magnitude less. The CT parameter is set equal to the CN parameter.

The pinball radius just needs to be large enough for contact detection to occur before the penetration of the sphere and the barrel surface.

The Radius parameter is the ball radius of 0.037 (m).

```
//Creation of contact pairs
Contacts 1
NSSS 1 NodeSet 2 SurfaceSet 1 MU 0.3 EPN 3.2e6
CN 2e2 EPT 3.2e5 CT 2e2 Pinball 1 Radius 0.037
MaxPointwiseInt 1 BoolTable 1
```

Creation of solver options

Here we request Giraffe to apply a single thread to solve this case. The linear system solution chosen is Direct.

```
//Creation of solver options
SolverOptions
Processors 1 LinSys Direct
```

Creation of monitors

We request Giraffe to monitor node 42 (the ball) and the contact pair during the simulation.

```
//Creation of monitors
Monitor Sample 1
MonitorNodes 42
MonitorContacts 1
```

Creation of post-files

We choose WriteRenderMesh, WriteFlexibleContactSurfaces, WriteContactForces and WriteRenderParticles options for post-processing using Paraview™.

```
//Creation of post files
PostFiles
MagFactor 1
WriteMesh 0
WriteRenderMesh 1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces 1
WriteForces 0
WriteConstraints 0
WriteSpecialConstraints 0
WriteContactForces 1
WriteRenderRigidBodies 0
WriteRenderParticles 1
```

Results and post-processing

To visualize our model in Paraview™, first we have to open the files “whole_solution_contactsurfaces.pvd”, “whole_solution_rb_particles.pvd” and “whole_solution_rendermesh.pvd”, which are located at the folder “post” in “tutorial14”.

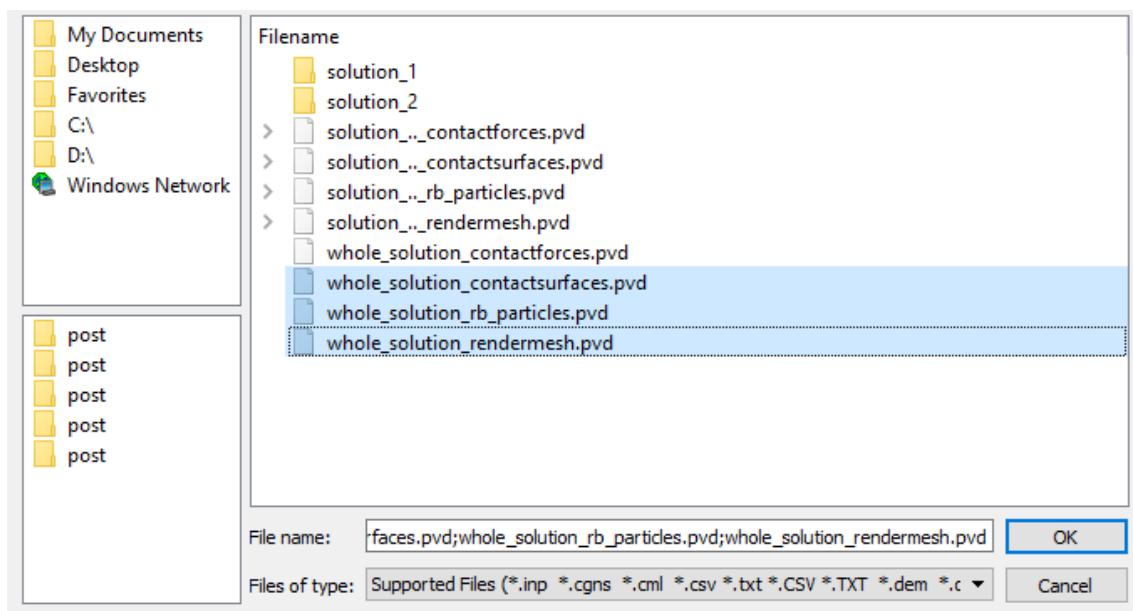


Figure 13-2 - Opening the files of tutorial 14 in Paraview™.

Then, we can click the preview icon and the Apply button on the Pipeline browser. After doing that, we are ready to post-process the results.

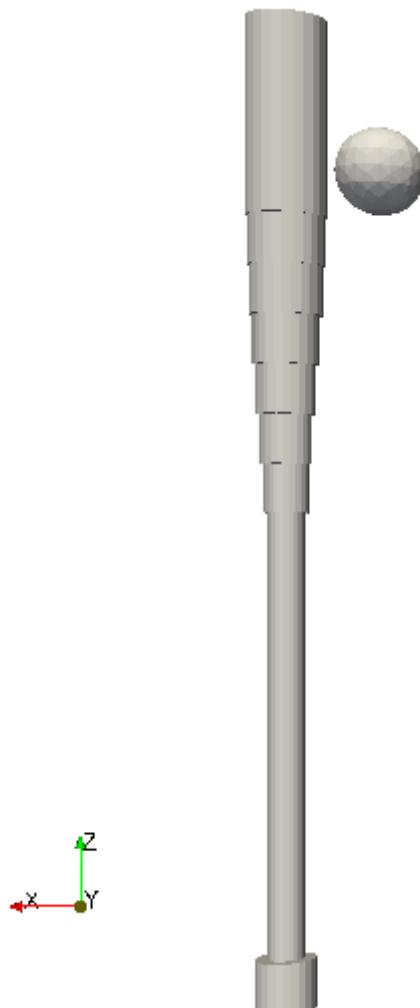
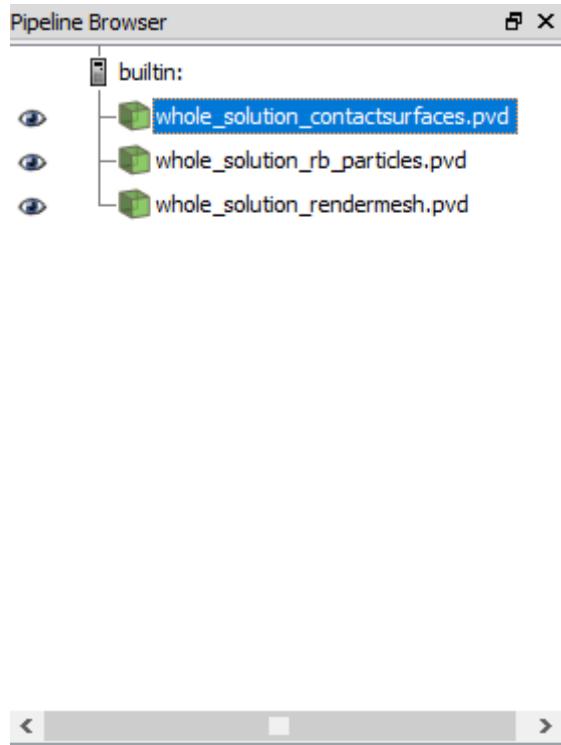
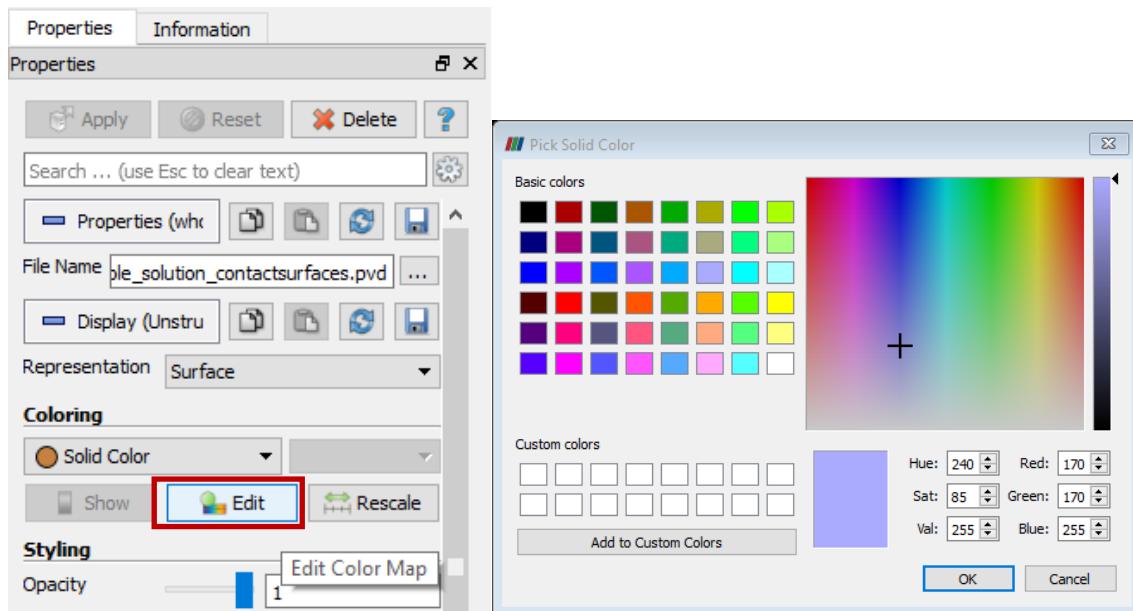


Figure 13-3 – Initial position of the system (render view).

It is possible to color the different elements of the model improving the visualization. To do that, we first select the object on the Pipeline browser (Figure 13-4), then we go to Properties tab, click edit in the Coloring section and pick a color from the Basic colors or from the color panel or typing the RGB or HSV coordinates of the color (Figure 13-5).

**Figure 13-4 – Choosing object on Pipeline browser.****Figure 13-5 – Changing object color property.**

We can also change the Opacity of the object on the Styling section. After some changes, we get the following result.

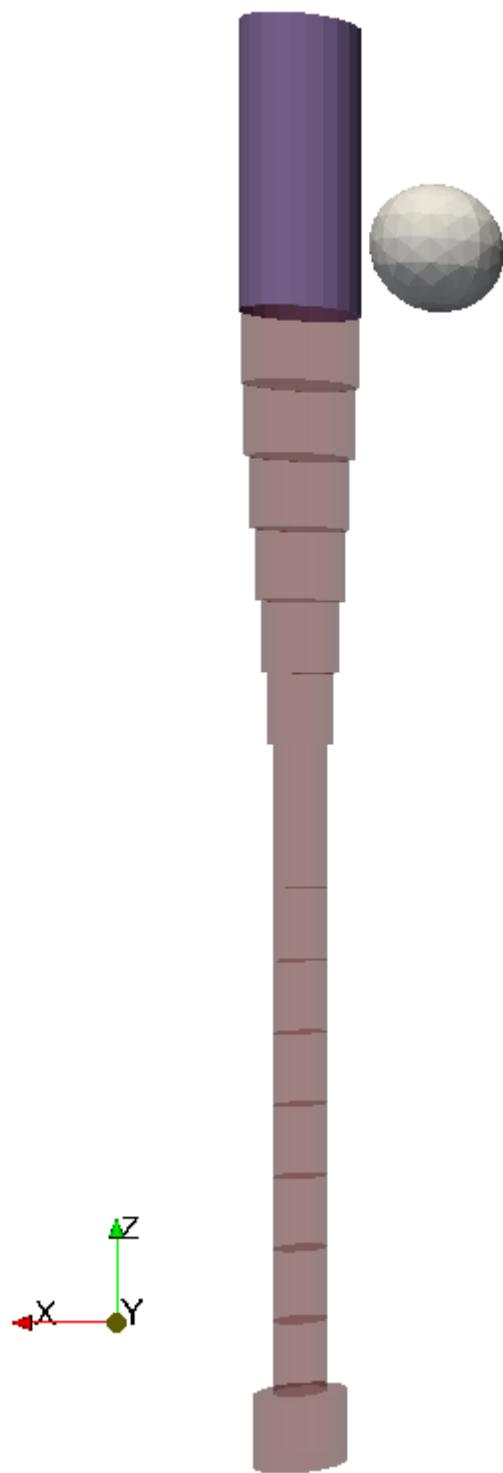


Figure 13-6 – Choosing object on Pipeline browser.

Notice that the contact surface is present only on the top part of the bat.

We can now visualize the animation clicking on the play button on the Main Controls toolbar. In Figure 13-7, we can see some frames of the animation, arrows indicating the velocity of the ball were added.

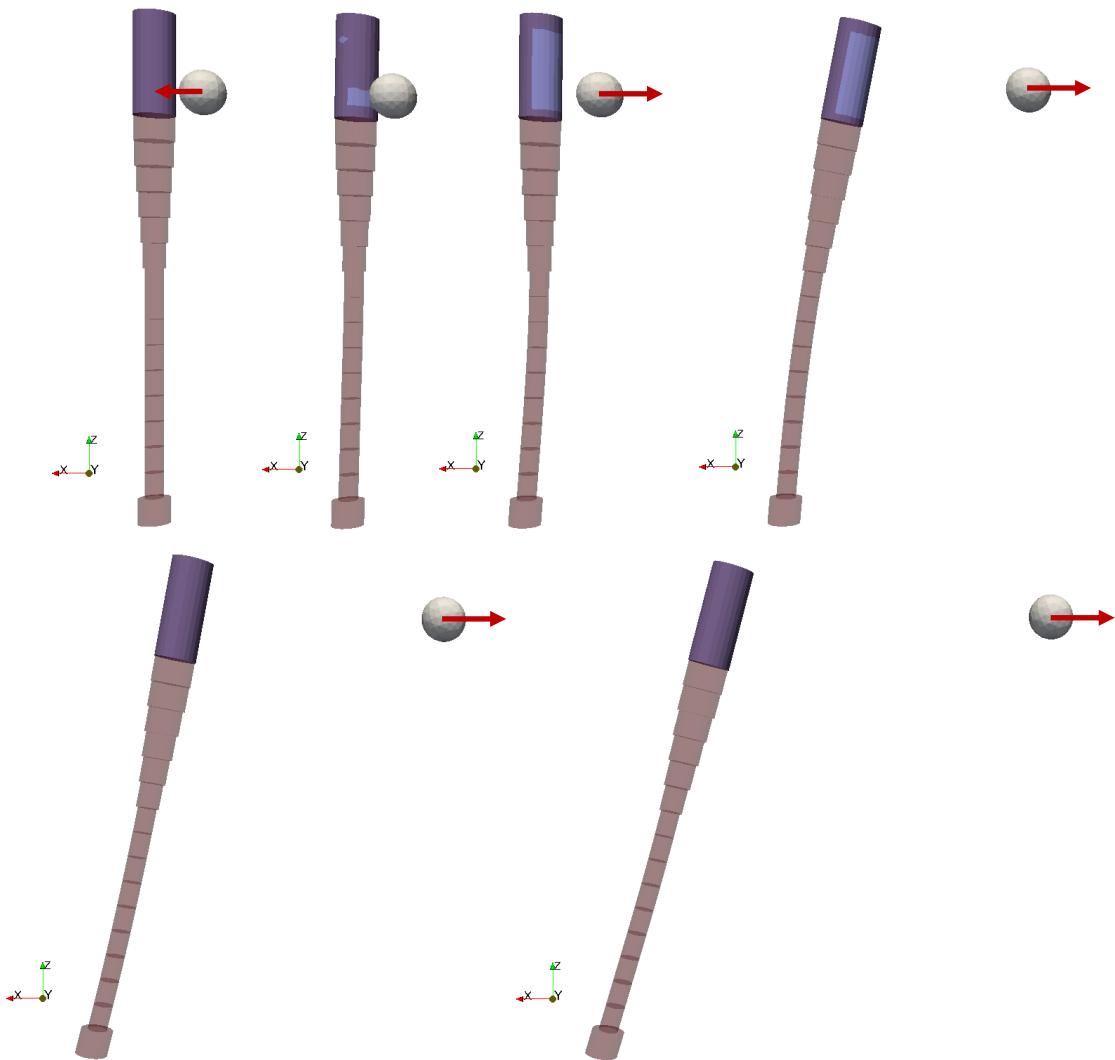
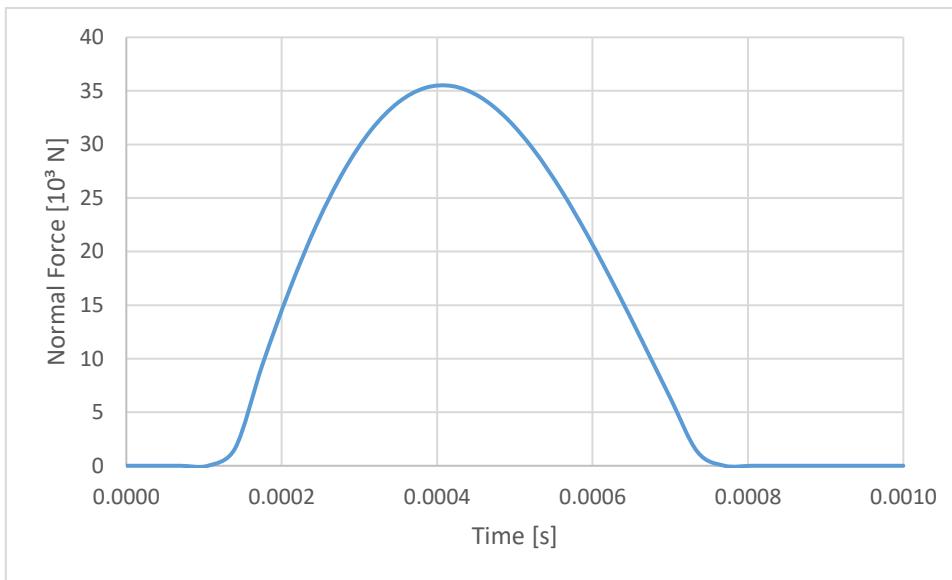
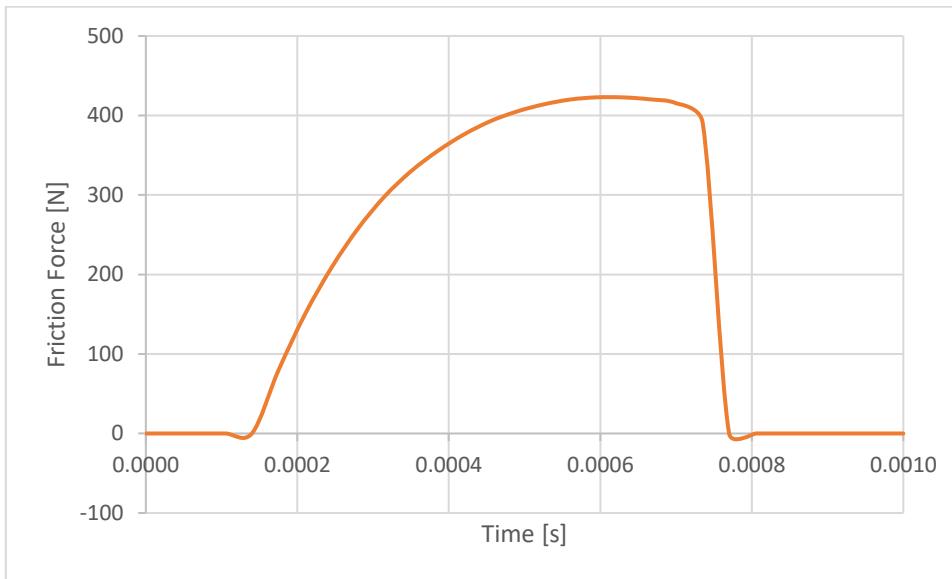


Figure 13-7 – Some selected frames of the simulation.

We could also plot a Glyph representing the contact force, but this is only for visualization purposes since we will inspect the forces by means of the contact monitor. The monitor files can be found in folder monitors in tutorial14.

Giraffe generates a monitor text file for the requested contact containing the time series of the normal force and friction force (Cartesian components and magnitude). Therefore, we can plot the time series of the forces and verify the impact time.

**Figure 13-8 – Normal Force vs Time.****Figure 13-9 – Friction Force vs Time.**

Giraffe also generates a file for the requested node containing the values of displacement and velocity in each global direction. Therefore, it is possible to plot the components of the velocity and obtain the exit velocity of the ball after the collision.

Figure 13-10 shows the x component of the velocity of the ball, the dominant coordinate. The ball approaches the bat with speed 40 m/s and exits with a larger speed.

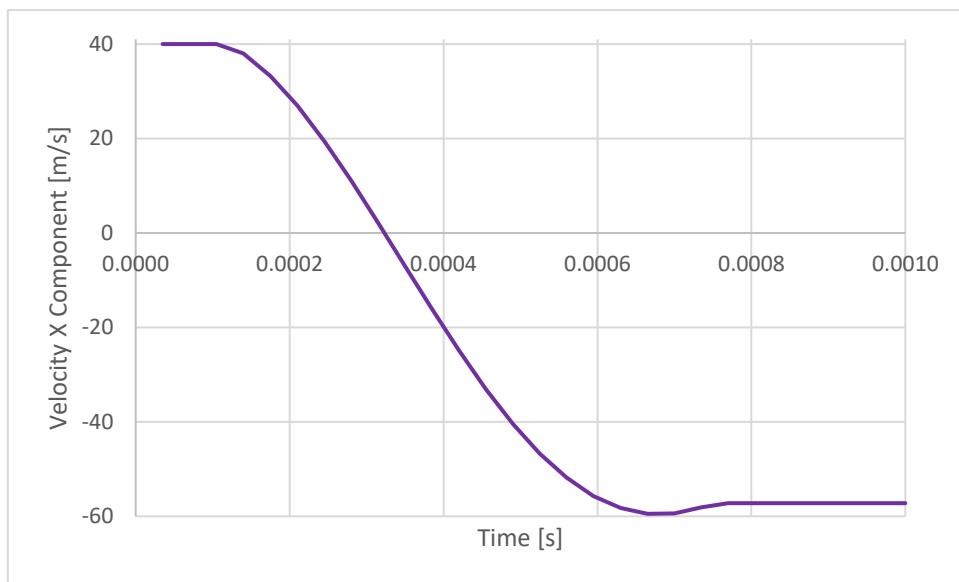


Figure 13-10 – Velocity x component vs Time.

There is some exit velocity component in the z direction, gained because of the friction, which transforms the rotation energy from the spin into kinetic energy, as show in Figure 13-11.

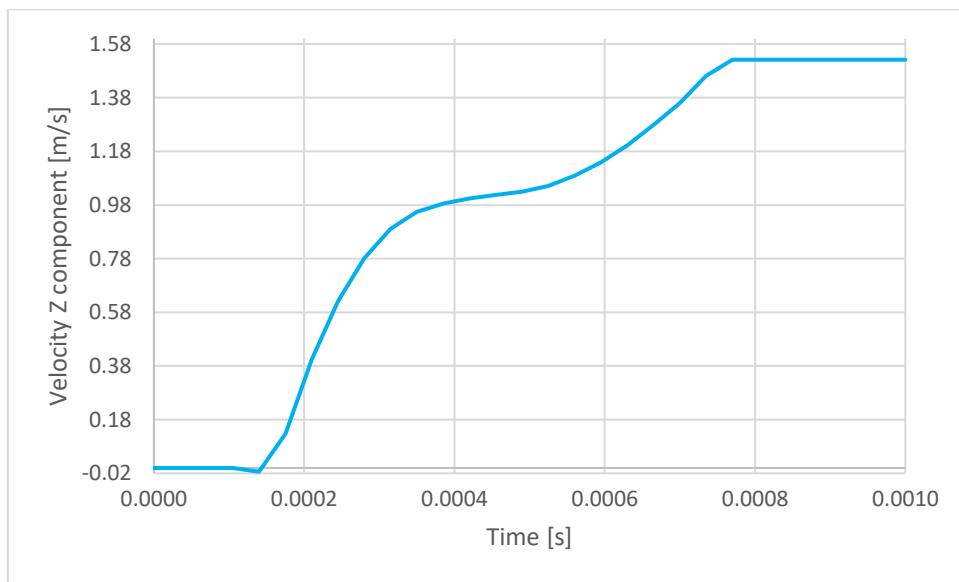


Figure 13-11 – Velocity z component vs Time.

Tutorial 14. Contact between cantilever beams

Input file name: tutorial14.inp

Developed by Debora Naomi Higa & Alfredo Gay Neto

Objective

This tutorial presents a simulation of contact between cantilever beams. The aim is to simulate the process of contact between beams with super-elliptical cross section subject to a prescribed axial rotation - applying nodal displacements and boundary conditions and solving a static geometrically-nonlinear analysis.

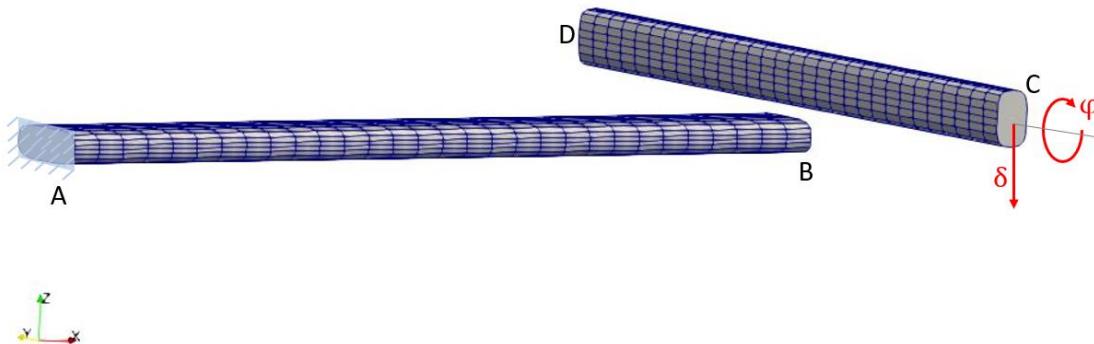


Figure 14-1 – Beams rendering view.

Description

Beams AB and CD (both with 3.0 m length) are meshed using 31 nodes and 15 elements each, using the element Beam_1. Beam AB is aligned to X direction and fixed at point A. Beam CD is aligned to Y direction.

In a first step, a vertical displacement is imposed at point C in order to establish contact between beams. In the sequence, a rotation is prescribed about Y axis at point C, thus creating an oscillating contact force pattern.

Input file step-by-step

Creation of nodes

62 nodes are created to establish a finite element mesh (31 nodes for each beam).

```
//Creation of nodes
Nodes 62
//Number      X      Y      Z
Node   1      0      2.5    0
Node   2      0.1    2.5    0
Node   3      0.2    2.5    0
Node   4      0.3    2.5    0
Node   5      0.4    2.5    0
...
Node  58     2.5    2.6    0.3
Node  59     2.5    2.7    0.3
Node  60     2.5    2.8    0.3
Node  61     2.5    2.9    0.3
Node  62     2.5    3       0.3
```

Creation of node sets

Two node sets are created to establish loads and constraints.

```
//Creation of node sets
NodeSets      2
//Point A
NodeSet       1      Nodes 1      List  1
//Point B
NodeSet       2      Nodes 1      List  32
```

Creation of materials

Two material models are defined, which identification numbers are 1 and 2.

```
//Creation of materials
Materials      2
//Material 1
Hooke 1      E      2e9      Nu      0.30      Rho      8000
//Material 2
Hooke 2      E      2e9      Nu      0.30      Rho      8000
```

Creation of cross-sections

Two cross-sections are defined, both super ellipses. The semi-axis A and B are defined lying in direction E1 and E2, respectively. Both super ellipse exponent value N are set by 3. The parameter AMeshFDM establishes the discretization employed by Giraffe to evaluate the St. Venant torsion constant (by the finite-difference method). This is done as a pre-processing. The identification numbers of the cross-sections are 1 and 2.

```
//Creation of sections
Sections      2
//Sesction 1
SuperEllipse 1      A      0.2      B      0.06      N      3      AMeshFDM      200
//Section 2
SuperEllipse 2      A      0.06      B      0.1      N      3      AMeshFDM      200
```

Creation of coordinate systems

Three coordinate systems are input. These are used to define the element properties and the nodal displacements. The identification number of the coordinate systems are 1, 2 and 3. Note that, since employing Beam_1 element (shown in sequence), the direction E3 must lie in the beam alignment direction

```
//Creation of coordinate systems
CoordinateSystems 3
CS    1      E1      0      1      0      E3      1      0      0
CS    2      E1      1      0      0      E3      0      1      0
CS    3      E1      1      0      0      E3      0      0      1
```

Creation of elements

Thirty elements are defined (fifteen for each beam). The elements are assigned to different material properties (identification number 1 or 2), different cross section (identification number

1 or 2) and different coordinate system (identification number 1 or 2). Each Beam_1 element is defined by three nodes, which **must** be equally spaced and contained in a straight line.

Note: In case of creation of not-aligned beam elements (for example, for representing an approximation of a curved beam or a frame), each beam element must be associated with a defined coordinate system, which direction E3 is aligned with the element direction. This is done in order to permit creation of arbitrary alignment between successive cross sections of a beam-like structure, composed possibly by many cross-sections or alignments of these.

```
//Creation of elements
Elements      30
Beam_1 1     Mat   1     Sec   1     CS    1     Nodes  1     2     3
Beam_1 2     Mat   1     Sec   1     CS    1     Nodes  3     4     5
Beam_1 3     Mat   1     Sec   1     CS    1     Nodes  5     6     7
Beam_1 4     Mat   1     Sec   1     CS    1     Nodes  7     8     9
Beam_1 5     Mat   1     Sec   1     CS    1     Nodes  9     10    11
Beam_1 6     Mat   1     Sec   1     CS    1     Nodes  11    12    13
Beam_1 7     Mat   1     Sec   1     CS    1     Nodes  13    14    15
Beam_1 8     Mat   1     Sec   1     CS    1     Nodes  15    16    17
Beam_1 9     Mat   1     Sec   1     CS    1     Nodes  17    18    19
Beam_1 10    Mat   1     Sec   1     CS    1     Nodes  19    20    21
Beam_1 11    Mat   1     Sec   1     CS    1     Nodes  21    22    23
Beam_1 12    Mat   1     Sec   1     CS    1     Nodes  23    24    25
Beam_1 13    Mat   1     Sec   1     CS    1     Nodes  25    26    27
Beam_1 14    Mat   1     Sec   1     CS    1     Nodes  27    28    29
Beam_1 15    Mat   1     Sec   1     CS    1     Nodes  29    30    31
Beam_1 16    Mat   2     Sec   2     CS    2     Nodes  32    33    34
Beam_1 17    Mat   2     Sec   2     CS    2     Nodes  34    35    36
Beam_1 18    Mat   2     Sec   2     CS    2     Nodes  36    37    38
Beam_1 19    Mat   2     Sec   2     CS    2     Nodes  38    39    40
Beam_1 20    Mat   2     Sec   2     CS    2     Nodes  40    41    42
Beam_1 21    Mat   2     Sec   2     CS    2     Nodes  42    43    44
Beam_1 22    Mat   2     Sec   2     CS    2     Nodes  44    45    46
Beam_1 23    Mat   2     Sec   2     CS    2     Nodes  46    47    48
Beam_1 24    Mat   2     Sec   2     CS    2     Nodes  48    49    50
Beam_1 25    Mat   2     Sec   2     CS    2     Nodes  50    51    52
Beam_1 26    Mat   2     Sec   2     CS    2     Nodes  52    53    54
Beam_1 27    Mat   2     Sec   2     CS    2     Nodes  54    55    56
Beam_1 28    Mat   2     Sec   2     CS    2     Nodes  56    57    58
Beam_1 29    Mat   2     Sec   2     CS    2     Nodes  58    59    60
Beam_1 30    Mat   2     Sec   2     CS    2     Nodes  60    61    62
```

Creation of contact surfaces

Thirty contact surfaces are defined (fifteen for each beam), all flexible super elliptical cylinder surfaces. The cylinder surfaces semi-axes (A and B) and exponent (N) are set in a similar way from the creation of cross-sections. Coordinate systems (CS) chosen are same employed during the creation of elements (identification number 1 or 2). The normal direction of the surfaces points outwards of the super elliptical cylinder. Each cylinder surface is defined by two nodes.

```
//Creation of contact surfaces
Surfaces      30
FlexibleSECylinder_1    1     A     0.2     B     0.06   N     3     CS
                           1     NormalExterior      Nodes  1     3
FlexibleSECylinder_1    2     A     0.2     B     0.06   N     3     CS
                           1     NormalExterior      Nodes  3     5
FlexibleSECylinder_1    3     A     0.2     B     0.06   N     3     CS
                           1     NormalExterior      Nodes  5     7
...
```

Creation of surface sets

Surface sets are created to set the potential contact bodies. Two surface sets are defined (one for each beam) input in a sequence. Both surfaces sets have 15 surfaces, which are 1 to 15 (SurfaceSet 1) and 16 to 30 (SurfaceSet 2).

```
//Creation of surface sets
SurfaceSets 2
SurfaceSet 1 Surfaces 15 Sequence Initial 1 Increment 1
SurfaceSet 2 Surfaces 15 Sequence Initial 16 Increment 1
```

Creation of displacements

Two nodal displacements are created (identification numbers 1 and 2), both with three time steps (NTimes) and coordinate system identification 3. The first prescribes no displacement to node set 1. The second prescribes displacements and rotations to node set 2. From time 0 to time 1 a displacement is linearly prescribed in Z direction (-0.5). From time 1 to time 2 a linearly varying displacement in Y direction of two units is prescribed. The displacement in Z direction (-0.5) is kept. From time 1 to time 2 a large rotation in Y axis is also linearly prescribed (25 rad).

```
//Creation of displacements
Displacements 2
NodalDisplacement 1 NodeSet 1 CS 3 NTimes 3
//Time UX UY UZ ROTX ROTY ROTZ
0 0 0 0 0 0
1 0 0 0 0 0
2 0 0 0 0 0
NodalDisplacement 2 NodeSet 2 CS 3 NTimes 3
//Time UX UY UZ ROTX ROTY ROTZ
0 0 0 0 0 0
1 0 0 -0.5 0 0 0
2 0 2.0 -0.5 0 25 0
```

Creation of constraints

Two nodal constraints are created. Both are done fixing the following DOFs for the node sets 1 and 2: UX, UY, UZ, ROTX, ROTY and ROTZ.

```
//Creation of constraints
Constraints 2
NodalConstraint      1      NodeSet      1
UX                  BoolTable    1
UY                  BoolTable    1
UZ                  BoolTable    1
ROTX                BoolTable    1
ROTY                BoolTable    1
ROTZ                BoolTable    1
NodalConstraint      2      NodeSet      2
UX                  BoolTable    1
UY                  BoolTable    1
UZ                  BoolTable    1
ROTX                BoolTable    1
ROTY                BoolTable    1
ROTZ                BoolTable    1
```

Creation of contacts

One contact constraint for the interaction between two surface sets (SSSS) is defined. The surface sets are identified by numbers 1 and 2. The coefficient of friction value (MU) is set by 0.3.

Constraints enforcements are done by the Penalty Method. Thus, it is necessary for the user to input penalty parameters data. Usually these may be calibrated based on physical information related to the desired scenario, basing on equivalent local stiffness, leading to allowable penetration on each contact zone. In this case, penalty coefficient to enforce normal and tangential contact constraints (EPN and EPT) are respectively set to 1e7 and 1e6.

Normal and tangential damping parameter coefficients (CN and CT) are useful for dissipation of energy during impact simulations, avoiding high frequency oscillations on contact forces. In this case, there is no damping on contact since the coefficients are set to 0.

Pinball radius value is a rough search geometrical parameter used by Giraffe to establish probable and not probable contact interactions. The pinball radius value is set to 1.

The maximum number of contact pointwise interactions between surfaces (MaxPointwiseInt) is set to 1. Note that on non-convexity scenarios, surfaces have the possibility of seeking for more than one pointwise contact solution.

Bool table data for current contact constraint (BoolTable) is set to 1. It is an optional keyword and it permits to create a scenario in which the contact constraint is turned on/off along solution steps. If BoolTable is not included, Giraffe assumes that the contact constraint will be considered “on” for all solution steps.

```
//Creation of contacts
Contacts      1
SSSS    1      SurfaceSet1   1      SurfaceSet2   2      MU      0.3      EPN      1e7
        CN      0      EPT      1e6      CT      0      Pinball      1.0
        MaxPointwiseInt      1      BoolTable      1
```

Creation of solution steps

This tutorial has two solution steps (both are statics). The first step is created establishing the end time equal to “1”. The time-step is set to “0.05”, which means that the load is divided into increments of 5% along the first solution step. The maximum time-step is set by “0.05” and the minimum time-step is set by “1e-4” which permits Giraffe to automatically decrease the time-step in case of difficulties of convergence along the simulation. The second step is created establishing the end time equal to “2”, time-step “0.005”, maximum time-step “0.1” and minimum time-step “1e-6”. The sampling for saving post-processing files is set to “1” which ensures saving post-processing files for each converged configuration.

```
//Creation of the solution steps
SolutionSteps2
Static 1
EndTime 1
TimeStep 0.05
MaxTimeStep 0.05
MinTimeStep 1e-4
MaxIt 15
```

```

MinIt 3
ConvIncrease 2
IncFactor 1.4
Sample 1

Static 2
EndTime 2
TimeStep 0.005
MaxTimeStep 0.1
MinTimeStep 1e-6
MaxIt 15
MinIt 3
ConvIncrease 2
IncFactor 1.4
Sample 1

```

Creation of post-files

The options WriteMesh, WriteRenderMesh, WriteFlexibleContactSurfaces and WriteContactForces are chosen for post-processing using Paraview™. The parameter MagFactor assumes a unit value so the deformed shape of Paraview™ frames reflects the real scale.

```

//Creation of post files
PostFiles
MagFactor      1
WriteMesh      1
WriteRenderMesh    1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces     1
WriteForces     0
WriteConstraints   0
WriteSpecialConstraints 0
WriteContactForces 1
WriteRenderRigidBodies 0
WriteRenderParticles 0

```

Convergence criteria

Here, we define some convergence and divergence tolerances. For detailed explanation on the meaning of each of these quantities, refer to Giraffe users' manual. Altering default convergence criteria is recommended only for advanced users.

```

//Convergence criteria
ConvergenceCriteria
ForceTolerance 1e-4
MomentTolerance 1e-4
ForceMinimumReference 1e-3
MomentMinimumReference 1e-3
ConstraintMinimumReference 1e-7
DisplacementTolerance 1e-4
RotationTolerance 1e-4
LagrangeTolerance 1e-4
DisplacementMinimumReference 1e-5
RotationMinimumReference 1e-5
LagrangeMinimumReference 1e-6
DivergenceReference 1e+15

```

Creation of monitors

It is requested Giraffe to monitor the contact 1 (SSSS) created.

```
//Creation of monitors
Monitor Sample 1
MonitorContacts 1
```

Results and post-processing

Result files are automatically saved during the solving process on the folder “/tutorial15/post”.

Paraview™ post-processing

To view the post-processing in Paraview™, first we must open the files “whole_solution_contact_forces.pvd”, “whole_solution_contact_forces.pvd”, “whole_solution_mesh.pvd” and “whole_solution_rendermesh.pvd”, which are located at the folder “post” in “tutorial15”.

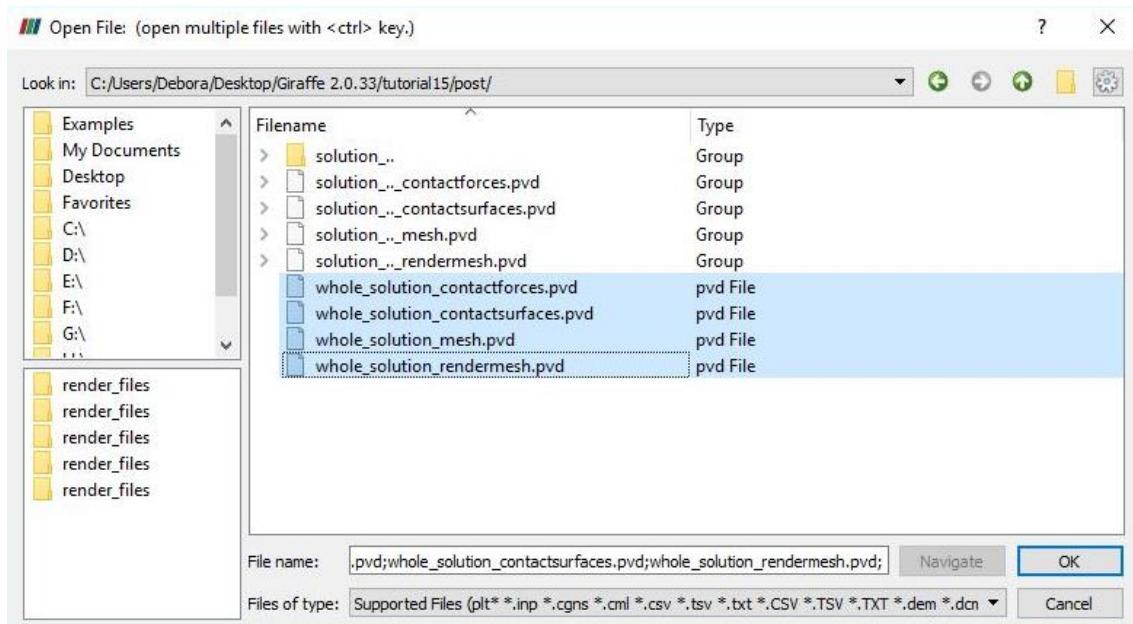


Figure 14-2 – Opening the files in Paraview™

Then, we must click the preview icon and the Apply button on the Pipeline browser.

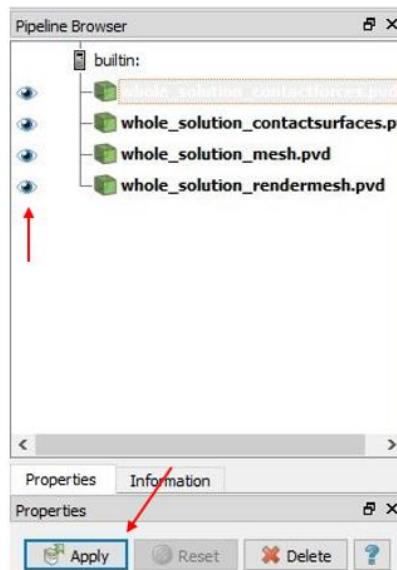


Figure 14-3 – Pipeline browser in Paraview™

As a particular feature, forces output may be used in Paraview™ together with Glyphs in order to establish special visualizations. For that, select “whole_solution_contactforces.pvd” in the Pipeline Browser. Then, click in the Glyph feature button and in the Apply button to create a Glyph.

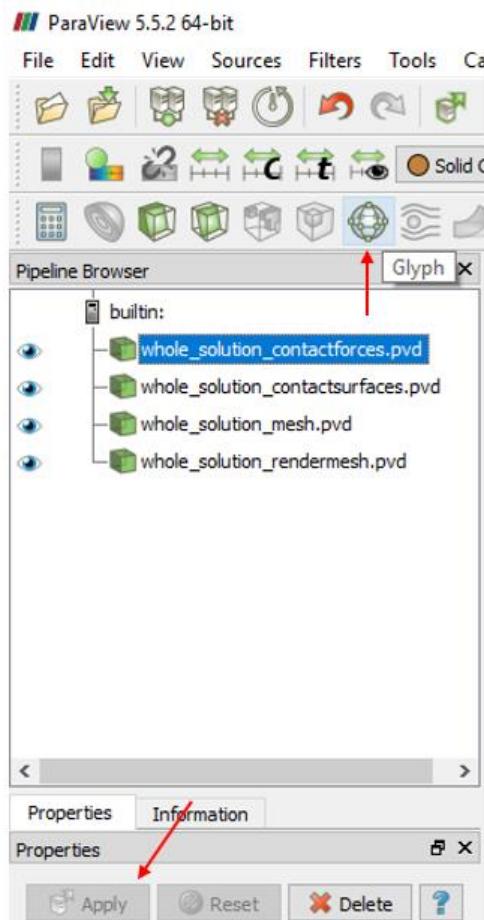


Figure 14-4 – Using Glyphs in Paraview™

After creating the Glyph, select it on Pipeline Browser to modify its default options as depicted next. Glyph Type is set to “Arrow”, Vectors is set to “normal” and Glyph Mode is set to “All Points”. Scale Mode is set to “vector” and scale factor is set to 0.0001. This value is arbitrary and may vary from model to model and controls the size of the created Glyph feature. It should be chosen based on the desired size of the Glyph feature on the visualization of the model.

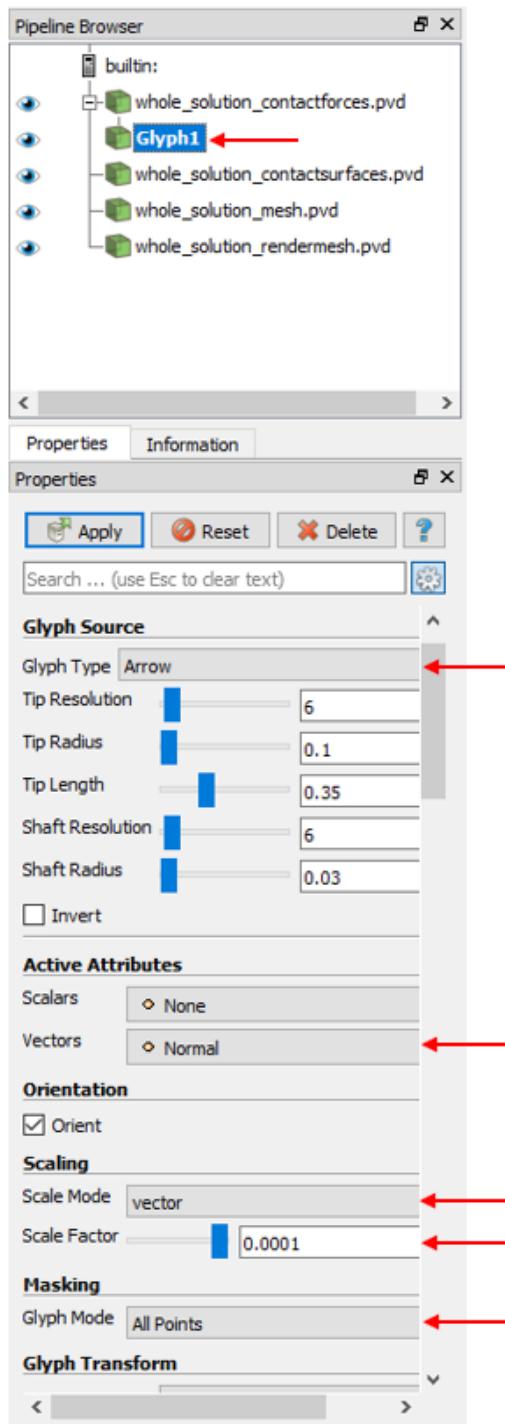
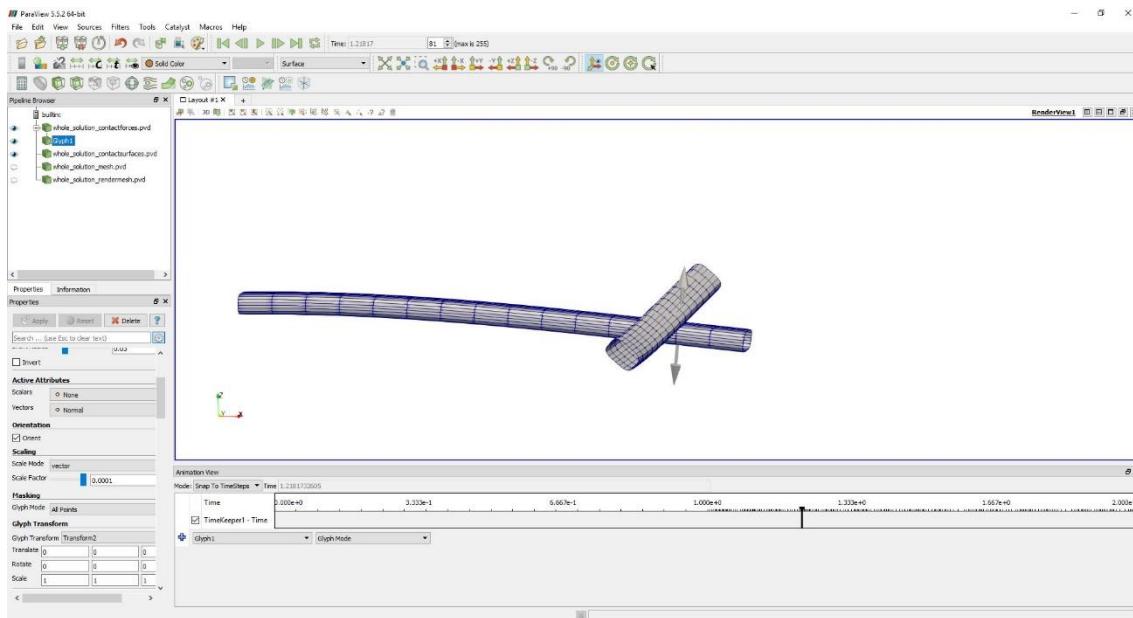


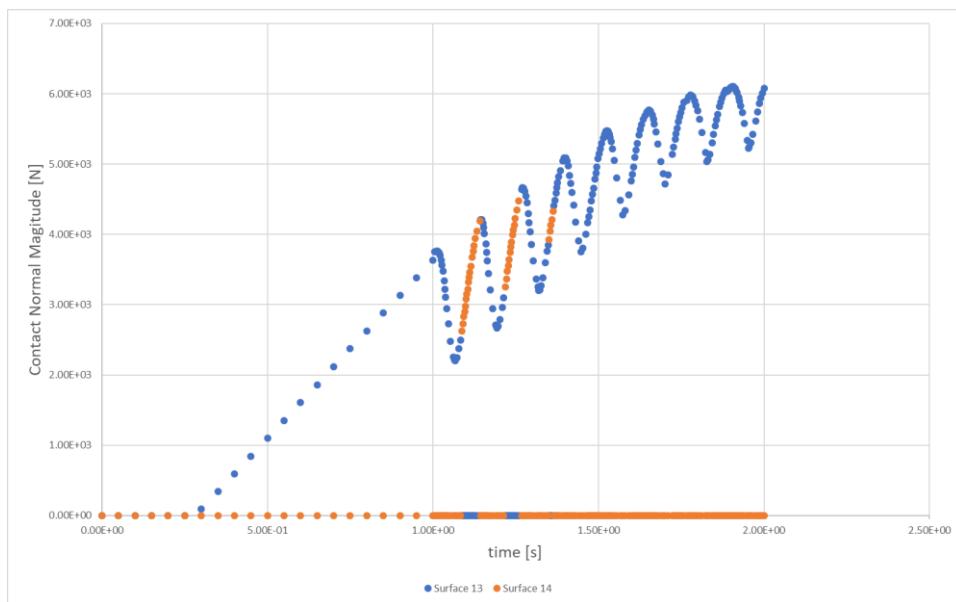
Figure 14-5 – Setting Glyph attributes in Paraview™

Visualization will look like the shown in Figure 14-6:

**Figure 14-6 – Using Glyphs in Paraview™**

The user may animate results to see the Glyph feature changing its size, according to the force magnitude along time.

Giraffe generates a monitor text file for the requested contact containing the time series of the forces. For example we can plot the time series of the normal forces.

**Figure 14-7 – Normal force vs time.**

What to try now?

- change the magnitude and the direction of the force
- change the beams cross section
- change material properties of one beam
- change material properties of both beams
- create different displacement compositions
- change the boundary conditions (fixed and free nodes)

Tutorial 15. Lateral buckling of a pipeline laid on a flat rigid surface

Input file name: tutorial15.inp

Developed by Marina Vendl Craveiro & Alfredo Gay Neto

Objective

This tutorial presents a geometrically-nonlinear simulation of a pipeline laid on a flat rigid surface with an isolated oscillatory imperfection. The aim is to simulate the laying process of the pipeline on the surface and evaluate the buckling of the structure when it is subjected to an internal pressure load.

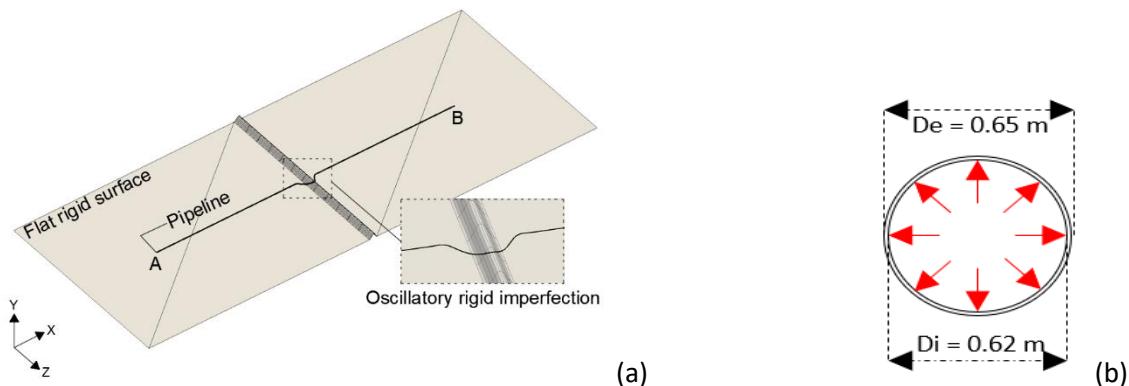


Figure 15-1 – (a) Model of the pipeline laid on a surface with imperfection (b) Cross section of the pipeline subjected to internal pressure field.

Description

The pipeline AB (2000 m length) is meshed considering 2001 equally-spaced nodes and 1000 Pipe_1 elements. The flat surfaces and the oscillatory imperfection are modeled using rigid oscillatory surfaces. The pipeline is initially positioned at the level of the top of the oscillatory imperfection. In order to establish contact with the rigid surface, the pipeline is released under the action of its self-weight. The interaction between the pipeline and its support is described by the NSSS contact model. Note that laying the pipeline on the oscillatory imperfection generates a vertical imperfection. Here, in addition to the vertical imperfection, a lateral imperfection is also imposed by means of a prescribed displacement at the central node of the pipeline. The laying process and the imposition of the lateral imperfection are done through static solution steps and lead to an initial imperfect pipeline configuration that is then subjected dynamically to the internal pressure load. The internal pressure is responsible for triggering the buckling of the pipeline.

Input file step-by-step

Creation of nodes

The finite element mesh contains 2001 nodes. Other 3 nodes are created to be pilot nodes of rigid oscillatory surfaces. So, the total of nodes is 2004.

```
//Creation of nodes
Nodes 2004
Node 1      -1000  0.25  0
Node 2      -999   0.25  0
```

Node	3	-998	0.25	0
Node	4	-997	0.25	0
Node	5	-996	0.25	0
Node	6	-995	0.25	0
Node	7	-994	0.25	0
Node	8	-993	0.25	0
Node	9	-992	0.25	0
Node	10	-991	0.25	0
Node	11	-990	0.25	0
Node	12	-989	0.25	0
Node	13	-988	0.25	0
Node	14	-987	0.25	0
Node	15	-986	0.25	0
Node	16	-985	0.25	0
Node	17	-984	0.25	0
Node	18	-983	0.25	0
Node	19	-982	0.25	0
Node	20	-981	0.25	0
...				
Node	1982	981	0.25	0
Node	1983	982	0.25	0
Node	1984	983	0.25	0
Node	1985	984	0.25	0
Node	1986	985	0.25	0
Node	1987	986	0.25	0
Node	1988	987	0.25	0
Node	1989	988	0.25	0
Node	1990	989	0.25	0
Node	1991	990	0.25	0
Node	1992	991	0.25	0
Node	1993	992	0.25	0
Node	1994	993	0.25	0
Node	1995	994	0.25	0
Node	1996	995	0.25	0
Node	1997	996	0.25	0
Node	1998	997	0.25	0
Node	1999	998	0.25	0
Node	2000	999	0.25	0
Node	2001	1000	0.25	0
Node	2002	0	0.125	0
Node	2003	-753.125	0	0
Node	2004	753.125	0	0

Creation of node sets

Ten node sets are defined to establish loads, displacements and constraints. Node set 1 groups all the pipeline nodes and is used to create the NSSS contact between the pipeline and the surfaces. Node sets 2 and 4 are associated with the pipeline end nodes. Node set 3 contains the central node of the pipeline, which establishes contact with the top of the oscillatory imperfection and at which the lateral displacement is prescribed. Node sets 5, 6 and 7 contain pilot nodes of rigid oscillatory surfaces.

In order to avoid rigid body movements during the laying process of the pipeline, the release of the pipeline nodes on the rigid surfaces is done in more than one step, also requiring the grouping of the pipeline nodes into three different node sets. Node set 8 contains 101 central nodes and node sets 9 and 10 contain the other nodes adjacent to node set 8, except for the pipeline end nodes (which are included in node sets 2 and 4).

```
//Creation of node sets
NodeSets 10
NodeSet 1 Nodes 2001 Sequence Initial 1 Increment 1
```

```

NodeSet 2 Nodes 1 List 1
NodeSet 3 Nodes 1 List 1001
NodeSet 4 Nodes 1 List 2001
NodeSet 5 Nodes 1 List 2002
NodeSet 6 Nodes 1 List 2003
NodeSet 7 Nodes 1 List 2004
NodeSet 8 Nodes 101 Sequence Initial 951 Increment 1
NodeSet 9 Nodes 949 Sequence Initial 2 Increment 1
NodeSet 10 Nodes 949 Sequence Initial 1052 Increment 1

```

Creation of elements

A total of 1000 elements is created. All the elements have the same cross section (identification number 1) and the same coordinate system (identification number 1). Each Pipe_1 element is defined by 3 nodes, which must be equally spaced and contained in a straight line.

```

//Creation of elements
Elements      1000
Pipe_1 1      PipeSec    1      CS     1      Nodes  1      2      3
Pipe_1 2      PipeSec    1      CS     1      Nodes  3      4      5
Pipe_1 3      PipeSec    1      CS     1      Nodes  5      6      7
Pipe_1 4      PipeSec    1      CS     1      Nodes  7      8      9
Pipe_1 5      PipeSec    1      CS     1      Nodes  9      10     11
Pipe_1 6      PipeSec    1      CS     1      Nodes  11     12     13
Pipe_1 7      PipeSec    1      CS     1      Nodes  13     14     15
Pipe_1 8      PipeSec    1      CS     1      Nodes  15     16     17
Pipe_1 9      PipeSec    1      CS     1      Nodes  17     18     19
Pipe_1 10     PipeSec    1      CS     1      Nodes  19     20     21
Pipe_1 11     PipeSec    1      CS     1      Nodes  21     22     23
Pipe_1 12     PipeSec    1      CS     1      Nodes  23     24     25
Pipe_1 13     PipeSec    1      CS     1      Nodes  25     26     27
Pipe_1 14     PipeSec    1      CS     1      Nodes  27     28     29
Pipe_1 15     PipeSec    1      CS     1      Nodes  29     30     31
...
Pipe_1 986    PipeSec    1      CS     1      Nodes  1971   1972   1973
Pipe_1 987    PipeSec    1      CS     1      Nodes  1973   1974   1975
Pipe_1 988    PipeSec    1      CS     1      Nodes  1975   1976   1977
Pipe_1 989    PipeSec    1      CS     1      Nodes  1977   1978   1979
Pipe_1 990    PipeSec    1      CS     1      Nodes  1979   1980   1981
Pipe_1 991    PipeSec    1      CS     1      Nodes  1981   1982   1983
Pipe_1 992    PipeSec    1      CS     1      Nodes  1983   1984   1985
Pipe_1 993    PipeSec    1      CS     1      Nodes  1985   1986   1987
Pipe_1 994    PipeSec    1      CS     1      Nodes  1987   1988   1989
Pipe_1 995    PipeSec    1      CS     1      Nodes  1989   1990   1991
Pipe_1 996    PipeSec    1      CS     1      Nodes  1991   1992   1993
Pipe_1 997    PipeSec    1      CS     1      Nodes  1993   1994   1995
Pipe_1 998    PipeSec    1      CS     1      Nodes  1995   1996   1997
Pipe_1 999    PipeSec    1      CS     1      Nodes  1997   1998   1999
Pipe_1 1000   PipeSec    1      CS     1      Nodes  1999   2000   2001

```

Creation of element sets

One element set is defined, which contains all the elements created previously. It is used to create the internal pressure load field.

```

//Creation of element sets
ElementSets 1
ElementSet 1 Elements 1000 Sequence Initial 1 Increment 1

```

Creation of surfaces

Both the flat surfaces and the oscillatory imperfection are modeled using rigid oscillatory surfaces. So, 3 surfaces are created: the first represents the central imperfection and the other 2 represent the adjacent flat surfaces. Note that rigid triangular surfaces could also be used to represent the flat surfaces.

To create the oscillatory imperfection it is necessary to provide information on amplitude, wavelength, phase and number of waves for each direction of the surface. Such parameters must be adjusted to create the desired geometry of the imperfection, besides establishing a smooth connection with the adjacent flat surfaces. Using the coordinate system with identification number 2, the oscillatory imperfection illustrated in Figure 15-1 (a) can be created by using one wave in each direction of the surface. Amplitude A1 must be equal to half the desired height for the imperfection and Lambda1 must be equal to the desired wavelength for the imperfection. Phi1 must be $-\pi/2$. Once the imperfection height along direction E2 does not vary, amplitudes A2 and A12 must be zero and Lambda2 must be equal to the desired width for the surface. Note that it is also necessary to specify a pilot node to rule the surface movement along the model evolution. In this tutorial, the pilot node must be located in the center of the surface, at half the height of the imperfection.

The flat surfaces can be created using the same procedure. For a surface to be flat, all amplitudes must be zero.

```
//Creation of surfaces
Surfaces      3
RigidOscillatorySurface_1 1 A1 -0.125 A2 0 A12 0 Lambda1 12.5 Lambda2 1000
Phi1 -1.57 Phi2 1.57 Waves1 1 Waves2 1 CS 2 PilotNode 2002
RigidOscillatorySurface_1 2 A1 0 A2 0 A12 0 Lambda1 1493.75 Lambda2 1000 Phi1
-1.57 Phi2 1.57 Waves1 1 Waves2 1 CS 2 PilotNode 2003
RigidOscillatorySurface_1 3 A1 0 A2 0 A12 0 Lambda1 1493.75 Lambda2 1000 Phi1
-1.57 Phi2 1.57 Waves1 1 Waves2 1 CS 2 PilotNode 2004
```

Creation of surface sets

Two surface sets are created: the first groups the flat surfaces and the second contains the oscillatory imperfection. The surfaces sets are used to create the NSSS contact between the pipeline and the surfaces.

```
//Creation of surface sets
SurfaceSets 2
SurfaceSet 1 Surfaces 1 List 1
SurfaceSet 2 Surfaces 2 Sequence Initial 2 Increment 1
```

Creation of contacts

Two NSSS contacts are created to test the interaction between the pipeline and the rigid surfaces along the simulation. The NSSS contact with identification number 1 tests the interaction between the pipeline nodes (node set 1) and the oscillatory imperfection (surface set 1) while the NSSS contact with identification number 2 tests the interaction between the pipeline nodes (node set 1) and the flat surfaces (surface set 2). The NSSS contact model uses a node to surface description in which constraint enforcements are done by penalty method. It is necessary to input the coefficient of friction, the normal and tangential penalty coefficients,

besides the normal and tangential damping. It is assumed here that the surfaces are rigid. So, large values of penalty coefficients are used (within the possibilities of numerical convergence) in order to minimize the vertical penetrations of the pipeline into the surfaces and to approximate the tangential constitutive relation to the Coulomb's law. The coefficient of friction is 0.4 and no damping is considered.

Once the pipeline axis is at the level of the surfaces, the radius is zero. Neither the rigid body rotation that occurs in rolling scenarios nor the moment of the friction force are evaluated. A pinball radius of 2500 m ensures that whenever it exists, the contact is detected by the model. Besides this, only a single pointwise contact is allowed between the pipeline and the surfaces.

The BoolTable is not included so the contact constraint is considered turned on for all solution steps.

```
//Creation of contacts
Contacts 2
NSSS 1 NodeSet 1 SurfaceSet 1 MU 0.4 EPN 1e9 CN 0 EPT 1e8 CT 0 Pinball 2500
Radius 0 MaxPointwiseInt 1
NSSS 2 NodeSet 1 SurfaceSet 2 MU 0.4 EPN 1e7 CN 0 EPT 1e6 CT 0 Pinball 2500
Radius 0 MaxPointwiseInt 1
```

Creation of cross-sections

One only cross section is created, which is shown in the Figure 15-1 (b).

```
//Creation of pipeline sections
PipeSections 1
PS      1     EA      5.9847E9     EI      3.0182E8     GJ      2.4146E8     GA
          2.3939E9   Rho      234.9    CDT      0.1     CDn      1.2     CAT      0     CAN
          1     De      0.65     Di      0.62
```

Creation of coordinate systems

Three coordinate systems are created. The first one is used to define the element properties. Note again that, since employing Pipe_1 element, direction E3 must lie in the pipeline alignment direction. The second coordinate system is used to define the rigid oscillatory surfaces. Directions E1 and E2 must be the horizontal directions of the oscillatory surfaces while direction E3 must be related to their amplitudes. Finally, the third coordinate system is used to define the direction of application of the nodal loads and prescription of the nodal displacements.

```
//Creation of coordinate systems
CoordinateSystems 3
CS      1     E1      0      1      0      E3      1      0      0
CS      2     E1      1      0      0      E3      0      1      0
CS      3     E1      1      0      0      E3      0      0      1
```

Creation of solution steps

This tutorial has eight solution steps. The first seven solution steps are static and are used to simulate the laying process of the pipeline on the rigid surfaces and to impose a lateral imperfection at the central node of the pipeline. The laying process is done in more than one step to avoid rigid body movements that can occur due to the low initial stiffness of the structure. As already mentioned, the pipeline is initially positioned at the level of the top of the

oscillatory imperfection and has all nodes fixed. The first solution step consists in prescribing a vertical displacement of 0.25 m (imperfection height) at node sets 9 and 10 in order to position the pipeline at the level of the flat rigid surfaces. Note that the pipeline self-weight is not applied yet. So, no contact is established between the pipeline and the surfaces. Such step establishes the end time equal to 1. The time step used is 0.04, that is, the prescribed displacement is divided into increments of 4%. In the second solution step, the self-weight is applied to the structure and the node set 8 is released in order to establish contact with the rigid surfaces. The solution step 2 establishes the end time equal to 2. The time step used is 0.004, that is, the load is divided into increments of 0.4%. The third solution step releases the node sets 9 and 10 under the action of the pipeline self-weight in order to establish contact with the rigid surfaces. The solution step 3 establishes the end time equal to 3. Again, the time step used is 0.004. The fourth solution step releases the node sets 2 and 4 in order to establish contact with the rigid surfaces. Such node sets are fixed again in the fifth solution step. The solution steps 4 and 5 establish the end times equal to 4 and 5, respectively. The time step used is 1, that is, the load is applied in a single step.

Once finished the laying process of the pipeline, a lateral displacement of 0.501 m is imposed at the central node in the sixth solution step. Such node is released in the seventh solution step and, due to the friction between the pipeline and the surface, the lateral imperfection of the pipeline is maintained. The solution steps 6 and 7 establish the end times equal to 6 and 7, respectively. The time step used is 1, that is, the load is applied in a single step.

The eighth solution step is dynamic and is used to apply an internal pressure of 20 MPa to the pipeline. The internal pressure is applied over 11240 s. Such time is approximately eleven times the largest natural period of the structure (a modal analysis can be performed previously to determine the natural period of the structure) and avoids the excitation of any natural vibration mode of the structure. The eighth step establishes the end time equal to 11247. The time step used is 1, that is, the internal pressure is initially divided into increments of approximately 0.0089%.

```
//Creation of solution steps
SolutionSteps 8
Static 1
EndTime 1
TimeStep 0.04
MaxTimeStep 1
MinTimeStep 1.0E-10
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor 1.1
Sample 10
Static 2
EndTime 2
TimeStep 0.004
MaxTimeStep 1
MinTimeStep 1.0E-10
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor 1.1
Sample 10
Static 3
EndTime 3
TimeStep 0.004
MaxTimeStep 1
MinTimeStep 1.0E-10
```

```
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor 1.1
Sample 10
Static 4
EndTime 4
TimeStep 1
MaxTimeStep 1
MinTimeStep 1.0E-10
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor 1.1
Sample 10
Static 5
EndTime 5
TimeStep 1
MaxTimeStep 1
MinTimeStep 1.0E-10
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor 1.1
Sample 10
Static 6
EndTime 6
TimeStep 1
MaxTimeStep 1
MinTimeStep 1.0E-10
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor 1.1
Sample 10
Dynamic 7
EndTime 7
TimeStep 1
MaxTimeStep 1
MinTimeStep 1.0E-10
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor 1.1
Sample 1
RayleighDamping Alpha 0 Beta 0 Update 0
NewmarkCoefficients Beta 0.3 Gamma 0.5
Dynamic 8
EndTime 11247
TimeStep 1
MaxTimeStep 50
MinTimeStep 1e-8
MaxIt 15
MinIt 3
ConvIncrease 10
IncFactor 1.1
Sample 1
RayleighDamping Alpha 0 Beta 0 Update 0
NewmarkCoefficients Beta 0.3 Gamma 0.5
```

Creation of loads

An internal pressure of 20 MPa is applied to the element set 1 over 11240 s. Such load is applied as a ramp-load in the solution step 8.

```
//Creation of loads
Loads 1
PipeLoad 1 ElementSet 1 NTimes 2
//Time POI POE RhoI RhoE
7 0 0 0 0
11247 20000000 0 0 0
```

Creation of displacements

Three nodal displacements are prescribed along the simulation. The nodal displacements 1 and 2 consist in vertical displacements of 0.25 m prescribed at node sets 9 and 10, respectively. The idea is to place the pipeline closer to the flat surfaces in order to avoid rigid body movements. Such displacements are prescribed as ramp-displacements in the first solution step. The nodal displacement 3 consists in the creation of the lateral imperfection of 0.501 m at node set 3 and is prescribed as a ramp-displacement in the sixth solution step.

```
//Creation of displacements
Displacements      3
NodalDisplacement 1 NodeSet 9 CS 3 NTimes 2
//Time UX UY UZ ROTX ROTY ROTZ
0 0 0 0.000 0 0 0
1 0 -0.250 0 0 0 0
NodalDisplacement 2 NodeSet 10 CS 3 NTimes 2
//Time UX UY UZ ROTX ROTY ROTZ
0 0 0 0.000 0 0 0
1 0 -0.250 0 0 0 0
NodalDisplacement 3 NodeSet 3 CS 3 NTimes 2
//Time UX UY UZ ROTX ROTY ROTZ
5 0 0 0.000 0 0 0
6 0 0 0.501 0 0 0
```

Creation of constraints

Nine nodal constraints are created. The first three constraints refer to the pilot nodes and fix their DOFs UX, UY, UZ, ROTX, ROTY and ROTZ throughout the simulation. The fourth constraint refers to the node set 8 and fixes its DOFs UX, UY, UZ, ROTX, ROTY and ROTZ in the first solution step. Such DOFs are released in the second solution step, which allows to the central region of the pipeline to move under the action of its self-weight. The fifth and sixth constraints refer to the node sets 9 and 10, respectively, and fix their DOFs UX, UY, UZ, ROTX, ROTY and ROTZ in the first solution step (to prescribe the vertical displacement of 0.25 m) as well as in the second solution step (while the node set 8 is released). Such DOFs are released in the third solution step, which allows to the end regions of the pipeline to move under the action of their self-weights. The seventh and eight constraints refer to the node sets 2 and 4, respectively. Such node sets are maintained fixed throughout the simulation, except for the solution step 4, in which the end nodes are released to establish contact with the surfaces. Finally, the last constraint refers to the node set 3 whose DOFs UX, UY, UZ, ROTX, ROTY and ROTZ are fixed in the solution steps 1 and 6. In the solution step 6, a lateral displacement of 0.501 m is prescribed at node set 3, requiring its constraints to be turned on.

```
//Creation of constraint
Constraints 9
NodalConstraint    1      NodeSet      5
UX           BoolTable   1
UY           BoolTable   1
UZ           BoolTable   1
```

ROTX	BoolTable	1
ROTY	BoolTable	1
ROTZ	BoolTable	1
NodalConstraint	2	NodeSet 6
UX	BoolTable	1
UY	BoolTable	1
UZ	BoolTable	1
ROTX	BoolTable	1
ROTY	BoolTable	1
ROTZ	BoolTable	1
NodalConstraint	3	NodeSet 7
UX	BoolTable	1
UY	BoolTable	1
UZ	BoolTable	1
ROTX	BoolTable	1
ROTY	BoolTable	1
ROTZ	BoolTable	1
NodalConstraint	4	NodeSet 8
UX	BoolTable	1 0 0 0 0 0 0 0
UY	BoolTable	1 0 0 0 0 0 0 0
UZ	BoolTable	1 0 0 0 0 0 0 0
ROTX	BoolTable	1 0 0 0 0 0 0 0
ROTY	BoolTable	1 0 0 0 0 0 0 0
ROTZ	BoolTable	1 0 0 0 0 0 0 0
NodalConstraint	5	NodeSet 9
UX	BoolTable	1 1 0 0 0 0 0 0
UY	BoolTable	1 1 0 0 0 0 0 0
UZ	BoolTable	1 1 0 0 0 0 0 0
ROTX	BoolTable	1 1 0 0 0 0 0 0
ROTY	BoolTable	1 1 0 0 0 0 0 0
ROTZ	BoolTable	1 1 0 0 0 0 0 0
NodalConstraint	6	NodeSet 10
UX	BoolTable	1 1 0 0 0 0 0 0
UY	BoolTable	1 1 0 0 0 0 0 0
UZ	BoolTable	1 1 0 0 0 0 0 0
ROTX	BoolTable	1 1 0 0 0 0 0 0
ROTY	BoolTable	1 1 0 0 0 0 0 0
ROTZ	BoolTable	1 1 0 0 0 0 0 0
NodalConstraint	7	NodeSet 2
UX	BoolTable	1 1 1 0 1 1 1 1
UY	BoolTable	1 1 1 0 1 1 1 1
UZ	BoolTable	1 1 1 0 1 1 1 1
ROTX	BoolTable	1 1 1 0 1 1 1 1
ROTY	BoolTable	1 1 1 0 1 1 1 1
ROTZ	BoolTable	1 1 1 0 1 1 1 1
NodalConstraint	8	NodeSet 4
UX	BoolTable	1 1 1 0 1 1 1 1
UY	BoolTable	1 1 1 0 1 1 1 1
UZ	BoolTable	1 1 1 0 1 1 1 1
ROTX	BoolTable	1 1 1 0 1 1 1 1
ROTY	BoolTable	1 1 1 0 1 1 1 1
ROTZ	BoolTable	1 1 1 0 1 1 1 1
NodalConstraint	9	NodeSet 3
UX	BoolTable	1 0 0 0 0 1 0 0
UY	BoolTable	1 0 0 0 0 1 0 0
UZ	BoolTable	1 0 0 0 0 1 0 0
ROTX	BoolTable	1 0 0 0 0 1 0 0
ROTY	BoolTable	1 0 0 0 0 1 0 0
ROTZ	BoolTable	1 0 0 0 0 1 0 0

Creation of environment data

Environment data are created to introduce the gravity field on the model. In this example, the gravity field is linearly imposed along solution step 2 and is maintained throughout the simulation.

```
//Creation of environment data
Environment
GravityData
G 0 -9.81 0 BoolTable 0 1 1 1 1 1 1 1
```

Creation of solver options

It is requested Giraffe to apply three threads to solve this case. The linear system solution chosen is Direct.

```
//Creation of solver options
SolverOptions
Processors 3 LinSys Direct
```

Creation of monitors

It is requested Giraffe to monitor 3 nodes along the simulation: node 1 (point A), node 1001 (midspan) and node 2001 (point B). It is also requested Giraffe to monitor the two NSSS contacts created.

```
//Creation of monitors
Monitor Sample 1
MonitorNodes 1 1001 2001
MonitorContacts 1 2
```

Creation of post files

The options WriteMesh, WriteRigidContactSurfaces and WriteContactForces are chosen for post-processing using Paraview™. The parameter MagFactor assumes a unit value so the deformed shape of Paraview™ frames reflects the real scale.

```
//Creation of post files
PostFiles
MagFactor 1.0
WriteMesh 1
WriteRenderMesh 0
WriteRigidContactSurfaces 1
WriteFlexibleContactSurfaces 0
WriteForces 0
WriteConstraints 0
WriteSpecialConstraints 0
WriteContactForces 1
WriteRenderRigidBodies 0
WriteRenderParticles 0
```

Creation of convergence criteria

Here, it is defined the convergence criteria. This is an optional task, which may be done only by advanced users. For this case, however, it is necessary to re-establish such criteria according to here provided data in order to ensure convergence.

```
//Creation of convergence criteria
ConvergenceCriteria
ForceTolerance 1e-5
```

MomentTolerance	1e-5
ForceMinimumReference	1e0
MomentMinimumReference	1e0
ConstraintMinimumReference	1e-7
DisplacementTolerance	1e-4
RotationTolerance	1e-4
LagrangeTolerance	1e-4
DisplacementMinimumReference	1e-4
RotationMinimumReference	1e-4
LagrangeMinimumReference	1e-6
DivergenceReference	1e+15

Results and post-processing

Paraview™ post-processing

To post-process results using Paraview™, first we must open the files “whole_solution_contactsurfaces.pvd” and “whole_solution_mesh.pvd”, which are located at the folder “post” in “tutorial16”.

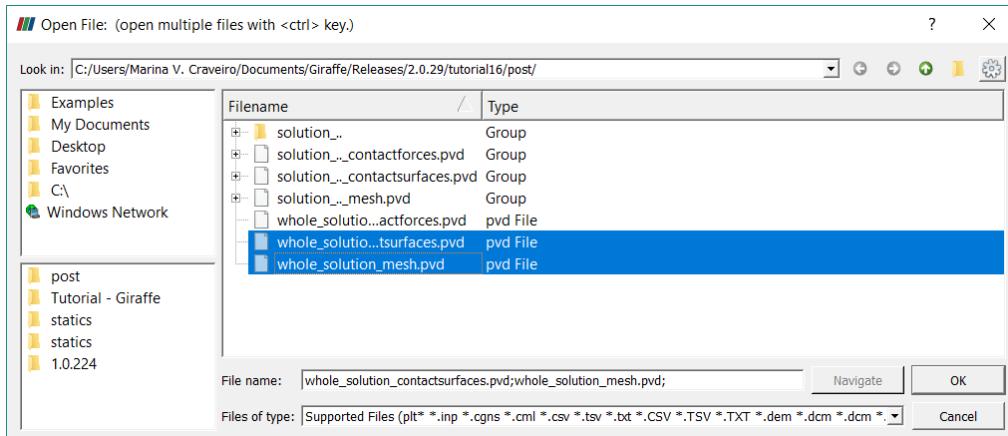


Figure 15-2 – Opening the files in Paraview™.

Then, we must click the preview icon and the Apply button on the Pipeline browser.

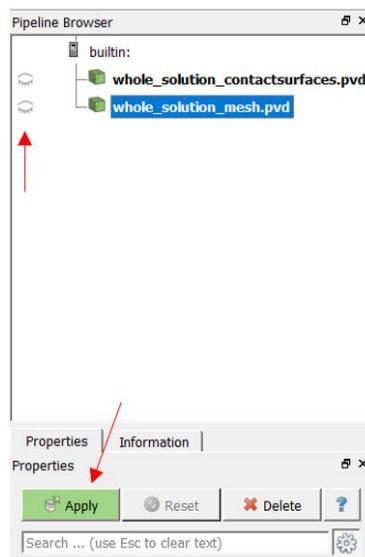


Figure 15-3 – Pipeline Browser.

Figure 15-4 shows how Paraview™ displays the results.

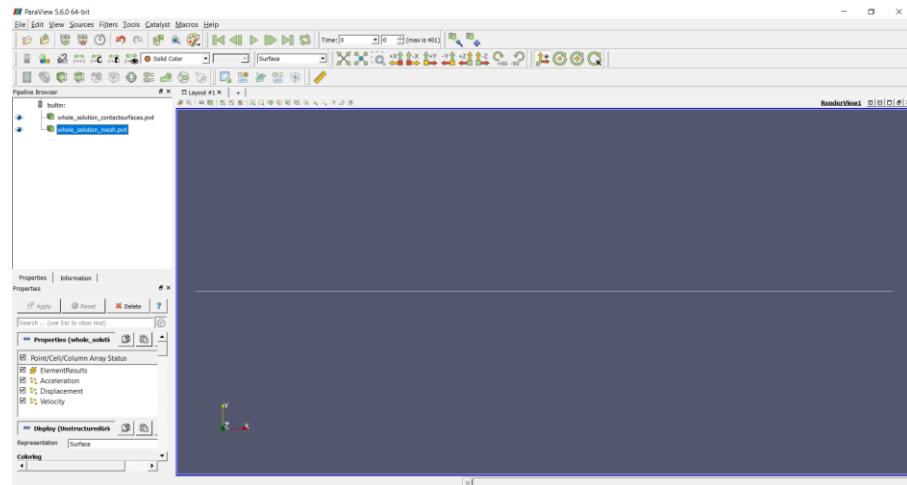


Figure 15-4 – Results using Paraview™.

It is interesting to perform an animation to observe how the structure deforms during the laying process and during the application of the internal pressure. Note that the model is three-dimensional and a better visualization of the deformation of the structure requires multiple views. This can be done by dividing the screen into multiple views. First, you can divide the screen horizontally into two parts. For that, use the menu located in the upper-right corner of the current view (Figure 15-5). The left-hand view can be used, for example, to visualize the perspective of the model (RenderView1). The right-hand view, in its turn, can also be divided vertically into two parts. For that, use the menu located in the upper-right corner of the right-hand view (Figure 15-6). Choose, then, two new render views. The top view, for example, can be used to visualize the xy plane (RenderView2) and the bottom view can be used to visualize the xz plane (RenderView3).

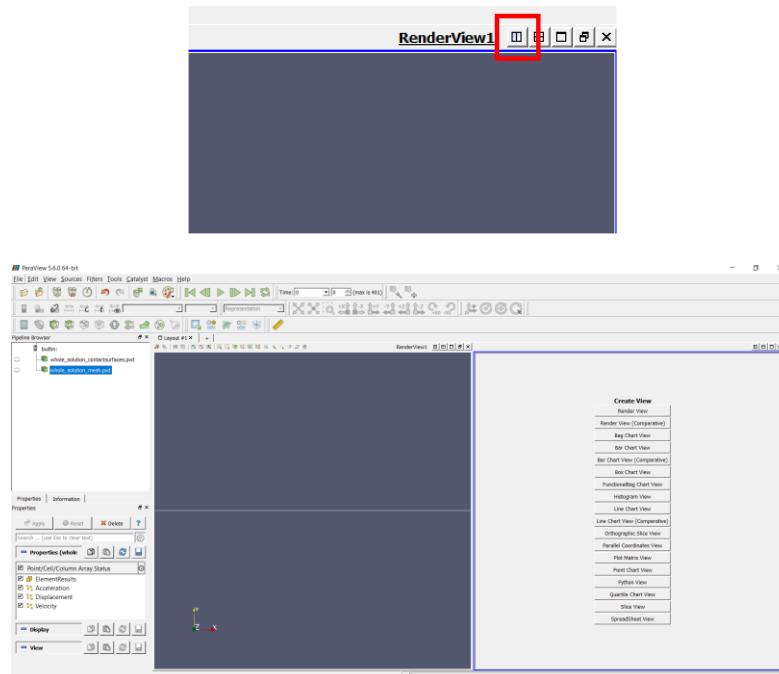


Figure 15-5 – Dividing the screen horizontally into two views.

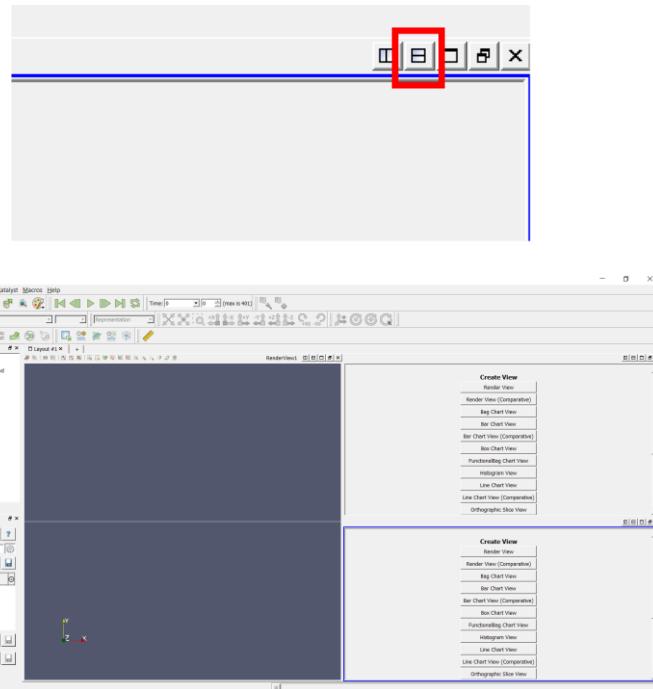


Figure 15-6 – Dividing the screen vertically into two views.

It is possible to adjust the direction and the zoom of each of the views for a better visualization. Texts can also be inserted by clicking, in sequence, Sources, Alphabetical and Text. Figure 15-7 shows the final layout. The animation can be performed by clicking the Play button located in the VCR controls toolbar.

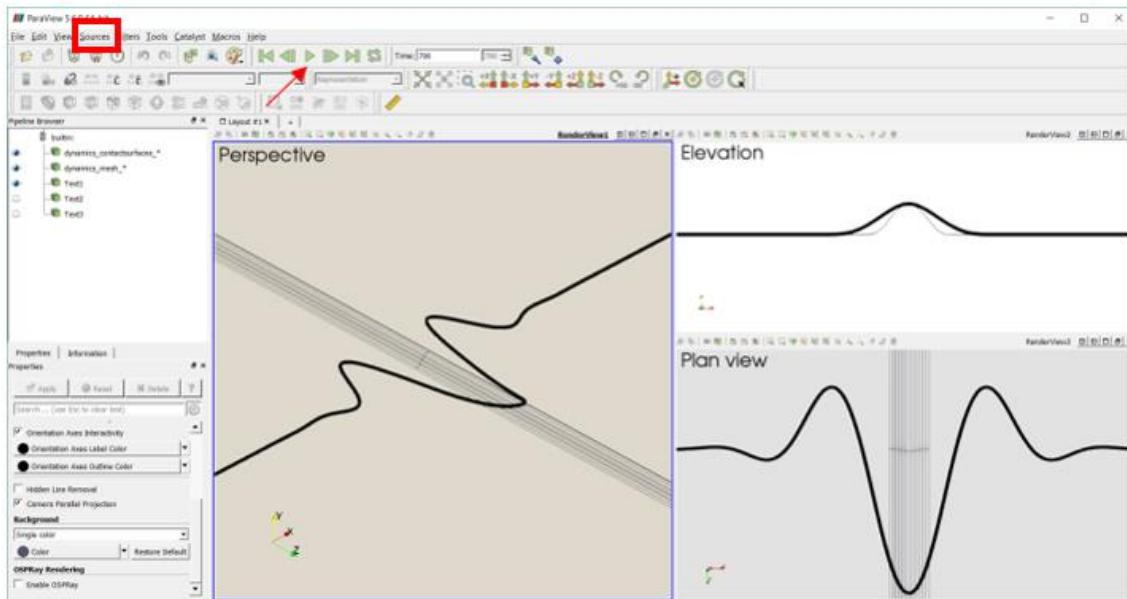


Figure 15-7 – Animation layout.

Monitors post-processing

Monitors are useful, for example, to plot the time series of node coordinates and displacements. It allows to evaluate the buckling process.

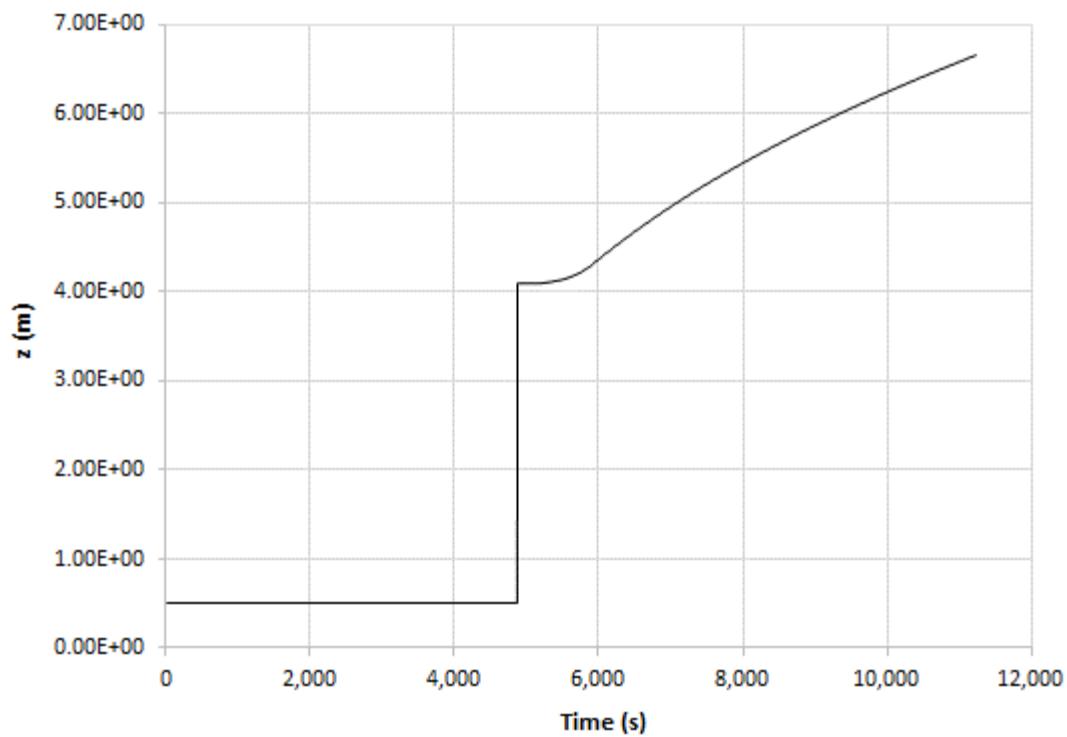


Figure 15-8 – Coordinate z of the central node of the pipeline versus time.

Tutorial 16. Suspension of a catenary mooring line

Input file name: tutorial16.inp

Developed by Guilherme Rocha Martins

Objective

This tutorial presents a geometrically-nonlinear simulation of a catenary polymeric and homogeneous mooring line. The objective is to show how to suspend a catenary mooring line (or a catenary riser).

Description

The mooring line (835.35 m length) is meshed using 269 nodes and 134 elements, using the element Pipe_1. There is an additional node, the pilot node attached to the seabed (to rule its behavior). The anchor (A) is fixed and has all your degrees of freedom blocked during all the analysis. The fairlead (B) is established along the simulation, while moving the mooring line tip to the desired location, at the floating unit. The mooring line is initially lying on the seabed, aligned in direction x. We consider here environment data, such as gravity and sea current loads. Next, a force of 1,000,000 N is applied at the point B in direction x and, lastly, a displacement of 186 m in direction z is prescribed to the point B to locate it close to surface level, attached to the floating unit. In this example, this point is located at 14 m depth.

Input file step-by-step

Creation of nodes

The mooring line and the pilot node of the seabed add up to 270 nodes. Node 1 is the anchor, node 269 is the fairlead and node 270 will be set, after, as the pilot node of the seabed. We made the choice here of considering the mooring line initially located at level z=0 and aligned in direction x.

Note that the discretization is more refined in the future expected touchdown zone of the catenary (between nodes 36 and 171).

```
//Creation of nodes
Nodes 270
//Number      X      Y      Z
//Anchor mooring line
Node   1      0.000000  0      0
Node   2      4.176750  0      0
Node   3      8.353500  0      0
Node   4     12.530250  0      0
Node   5     16.707000  0      0
Node   6     20.883750  0      0
Node   7     25.060500  0      0
Node   8     29.237250  0      0
Node   9     33.414000  0      0
Node  10     37.590750  0      0
...
Node  34    137.832750  0      0
Node  35    142.009500  0      0
Node  36    144.097880  0      0
Node  37    146.186250  0      0
Node  38    148.274630  0      0
Node  39    150.363000  0      0
Node  40    152.451380  0      0
```

```

...
Node 260    797.75925   0     0
Node 261    801.93600   0     0
Node 262    806.11275   0     0
Node 263    810.28950   0     0
Node 264    814.46625   0     0
Node 265    818.64300   0     0
Node 266    822.81975   0     0
Node 267    826.99650   0     0
Node 268    831.17325   0     0
//Fairlead node
Node 269    835.35000   0     0
//Pilot node
Node 270    0         0     0

```

Creation of elements

A total of 134 elements is defined. All the elements are assigned to the same pipe cross-section (identification number 1) and same coordinate system (identification number 1). Each Pipe_1 element is defined by three nodes, which must be equally spaced and contained in a straight line.

```

//Creation of elements
Elements      134
Pipe_1 1      PipeSec      1      CS      1      Nodes  1      2      3
Pipe_1 2      PipeSec      1      CS      1      Nodes  3      4      5
Pipe_1 3      PipeSec      1      CS      1      Nodes  5      6      7
Pipe_1 4      PipeSec      1      CS      1      Nodes  7      8      9
Pipe_1 5      PipeSec      1      CS      1      Nodes  9      10     11
Pipe_1 6      PipeSec      1      CS      1      Nodes  11     12     13
Pipe_1 7      PipeSec      1      CS      1      Nodes  13     14     15
Pipe_1 8      PipeSec      1      CS      1      Nodes  15     16     17
Pipe_1 9      PipeSec      1      CS      1      Nodes  17     18     19
Pipe_1 10     PipeSec      1      CS      1      Nodes  19     20     21
...
Pipe_1 125    PipeSec      1      CS      1      Nodes  249    250    251
Pipe_1 126    PipeSec      1      CS      1      Nodes  251    252    253
Pipe_1 127    PipeSec      1      CS      1      Nodes  253    254    255
Pipe_1 128    PipeSec      1      CS      1      Nodes  255    256    257
Pipe_1 129    PipeSec      1      CS      1      Nodes  257    258    259
Pipe_1 130    PipeSec      1      CS      1      Nodes  259    260    261
Pipe_1 131    PipeSec      1      CS      1      Nodes  261    262    263
Pipe_1 132    PipeSec      1      CS      1      Nodes  263    264    265
Pipe_1 133    PipeSec      1      CS      1      Nodes  265    266    267
Pipe_1 134    PipeSec      1      CS      1      Nodes  267    268    269

```

Creation of pipe sections

One only cross-section is defined. There is no need to define the material properties, because the pipe section already encompasses such data, together with cross section geometric information.

```

//Creation of pipe sections
PipeSections 1
PS      1      EA      7.536000e+08 EI      2.763621e+05 GJ      1.905945e+05 GA
          2.598621e+08 Rho     1.133500e+02 CDT      4.000e-01 CDn     2.000e+00
          CAT     2.500e-01 CAn     8.000e-01 De      7.660e-02 Di
          0.000e+00

```

Creation of coordinate systems

Two coordinate systems are created. The first one is used to define the element properties. Note that, since employing Pipe_1 element, the direction E3 must lie in the pipe alignment direction. The second coordinate system is the global one, used to define the direction of the application of the loads.

```
//Creation of coordinate systems
CoordinateSystems 2
//Pipe_1 Element CS (1)
CS    1      E1      0      1      0      E3      1      0      0
//Global aligned CS (2)
CS    2      E1      1      0      0      E3      0      0      1
```

Creation of solution steps

This tutorial has four solution steps (all statics). The choices of all end-times of solution steps are arbitrary, since there is no direct meaning of physical time in statics.

The first solution step is established to impose the self-weight loading on the mooring line and its contact with the seabed. The end-time is equal to “1” and a time-step is set to “0.1”, which means that the load is divided into increments of 10%. IncFactor 2.0 is considered, which permits Giraffe to increase the time-step automatically, in case of easy convergence rate.

The second solution step is established to provide stress-stiffness to the line, by applying an axial force at point B (the node that will become the fair lead, at the end of the simulation). This step is defined with the end-time equal to “2” with time-step equal to “0.1”, which means that the load is also divided into increments of 10%. IncFactor 2.0 is considered, which permits Giraffe to increase the time-step automatically, in case of easy convergence rate.

The third load step is established to prescribe a displacement at point B, from the seabed level to the desired level of the fair lead. For that, the end-time choice is equal to “3” and the time-step is considered “0.01”, with IncFactor 1.5.

The fourth load step is established to consider the sea current load. The end-time is equal to “4” and the time-step is considered “0.01”, with IncFactor 1.5.

```
//Creation of the solution steps
SolutionSteps 4
//Step 1
Static 1
EndTime      1
TimeStep     0.1
MaxTimeStep  1.0
MinTimeStep  0.0001
MaxIt       20
MinIt       3
ConvIncrease 1
IncFactor    2
Sample 10
//Step 2
Static 2
EndTime      2
TimeStep     0.1
MaxTimeStep  1.0
MinTimeStep  0.0001
MaxIt       30
```

```

MinIt 3
ConvIncrease 4
IncFactor 1.5
Sample 10
//Step 3
Static 3
EndTime 3
TimeStep 0.01
MaxTimeStep 0.1
MinTimeStep 0.0001
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor 1.5
Sample 10
//Step 4
Static 4
EndTime 4
TimeStep 0.01
MaxTimeStep 0.1
MinTimeStep 0.00001
MaxIt 20
MinIt 3
ConvIncrease 4
IncFactor 1.5
Sample 10

```

Creation of points

Four points are created to establish the seabed, through two triangular surfaces.

```

//Creation of points
Points 4
Point 1 -100 -200 0.01
Point 2 1100 -200 0.01
Point 3 1100 +200 0.01
Point 4 -100 +200 0.01

```

Creation of surfaces

Two triangular surfaces are created to compose a rectangular surface. Both surfaces are considered attached to the same pilot node (node 270). The pilot node movement rules the movement of the surface. Both translational and rotational movements are transferred from the pilot node to all surface points, as a rigid region.

```

//Creation of surfaces
Surfaces 2
RigidTriangularSurface_1 1 Points 1 2 3 PilotNode 270
RigidTriangularSurface_1 2 Points 1 3 4 PilotNode 270

```

Creation of node sets

Four node sets are created to establish loads and constraints. The first one represents the anchor and the second represents the fairlead. The third one contains all the mooring line nodes. Lastly, the fourth node set is the pilot node, on the seabed.

```
//Creation of node sets
NodeSets      4
//Tip - Point A
NodeSet       1      Nodes 1      List  1
//Tip - Point B
NodeSet       2      Nodes 1      List  269
//Mooring line nodes (all - except the pilot node)
NodeSet       3      Nodes 269    Sequence Initial 1 Increment 1
//Tip - Pilot node (seabed)
NodeSet       4      Nodes 1      List  270
```

Creation of the surface sets

Surface sets are created to establish contacts. Only one surface set is created, to represent the seabed.

```
//Creation of surface sets
SurfaceSets 1
//Seabed (ground)
SurfaceSet 1 Surfaces 2 List 1  2
```

Creation of contacts

Only one contact region is defined: between the mooring line (node set identification number 3) and the seabed (surface set identification number 1). The contact is between a node set and a surface set, indicated in Giraffe as “NSSL”. A friction coefficient of 0.2 is considered.

```
//Creation of contacts
Contacts      1
NSSL 1 NodeSet 3 SurfaceSet      1      MU      2.000000e-01 EPN      1.000000e+06
      CN      1.000000e+02 EPT      1.000000e+05 CT      0.000000e+00 Pinball
      1.762983e+03 Radius 0      MaxPointwiseInt      1      BoolTable 1
```

Creation of constraints

Three nodal constraints are created. The BoolTable feature is here employed, to consider each constraint turned “on” or “off”, according to desire in each sequential solution step.

The first nodal constraint is applied to the anchor node (node set 1). All translations are fixed, as well as the rotation around direction x. Other rotations are free. This condition is kept along the whole simulation.

The second nodal constraint is applied to the fairlead node (node set 2). It is equal to the first nodal constraint during the first solution step. Next, then the translation in direction x is released in the steps two and three (when the axial forces and vertical translation are imposed). In the last step (when the sea current is imposed), all degrees of freedom are fixed – since it is considered fixed at a floating unit.

The last nodal constraint concerns the seabed (pilot node). All degrees of freedom are fixed along the whole the simulation.

```
//Creation of constraints
Constraints 3
NodalConstraint    1      NodeSet      1
    UX           BoolTable   1
    UY           BoolTable   1
    UZ           BoolTable   1
    ROTX         BoolTable   1
NodalConstraint    2      NodeSet      2
    UX           BoolTable   1 0 0 1
    UY           BoolTable   1 1 1 1
    UZ           BoolTable   1 1 1 1
    ROTX         BoolTable   1 1 1 1
    ROTY         BoolTable   0 0 0 1
    ROTZ         BoolTable   0 0 0 1
NodalConstraint    3      NodeSet      4
    UX           BoolTable   1
    UY           BoolTable   1
    UZ           BoolTable   1
    ROTX         BoolTable   1
    ROTY         BoolTable   1
    ROTZ         BoolTable   1
```

Creation of displacements

A single prescribed nodal displacement is created. It is a vertical translation at the fairlead (node set identification number 2). It is a time-series data. Giraffe automatically performs linear interpolation between the given data, if needed.

```
//Creation of displacements
Displacements1
NodalDisplacement 1      NodeSet      2      CS      2      NTimes 2
    //Time UX UY UZ RX RY RZ
    2      0      0      0      0      0      0
    3      0      0      186     0      0      0
```

Creation of loads

A single nodal load in direction x (global) is created. This is done establishing a time-varying force on the fairlead (node set identification number 2). Giraffe automatically performs linear interpolation for undefined time values following input data.

```
//Creation of loads
Loads 1
NodalLoad    1      NodeSet      2      CS      2      NTimes 3
    //Time FX FY FZ MX MY MZ
    1      0      0      0      0      0      0
    2      1000000 0      0      0      0      0
    3      1000000 0      0      0      0      0
```

Creation of environment data

Environment data here includes gravity and ocean data. Both are considered according to the BoolTable input to turn “on” or “off” the effects, as desired and described here in solution steps section. Gravity data is considered during all the simulation and the sea current load is considered only in the last solution step. This can be noted in the BoolTable. We consider here

the surface position at 200 m level (z). The sea current is a 1.5 m/s speed aligned in 90 deg azimuth direction.

```
//Creation of environment
Environment
//Gravity data
GravityData
G      0      0      -9.81  BoolTable    1
OceanData
RhoFluid   1024   SurfacePosition    0      0      200
SeaCurrent  N      2      BoolTable    0 0 0 1
Depth     0      Speed  1.500  Angle  90
Depth    100    Speed  1.500  Angle  90
Depth    200    Speed  1.500  Angle  90
```

Creation of post-files

We choose WriteMesh, WriteRenderMesh and WriteRigidContactSurfaces options for post-processing using Paraview™. The MagFactor parameters can be used to generate plots with exaggerated displacement patterns, to help to visualize in case of small displacements. This is not the case of this example. Then, we will assume a unit value, then, requesting that the deformed shape of Paraview™ frames will reflect the real scale. WriteForces and WriteContactForces are also considered for post-processing.

```
//Creation of post files
PostFiles
MagFactor    1
WriteMesh    1
WriteRenderMesh  1
WriteRigidContactSurfaces 0
WriteFlexibleContactSurfaces 1
WriteForces   1
WriteConstraints 0
WriteSpecialConstraints 0
WriteContactForces 1
WriteRenderRigidBodies 0
WriteRenderParticles 0
```

Creation of monitors

We request Giraffe to monitor 6 nodes during the simulation (numbers 2, 36, 88, 90, 170, 268) and 6 elements (numbers 1, 18, 44, 45, 85, 134). Note that we entered a sampling of 1 for saving data for each converged time-step. In case we desire smaller files, we less data, we could set larger sample rates.

```
//Creation of monitors
Monitor          Sample 1
MonitorNodes 10  1      2      36      88      90      170      268
MonitorElements 6      1      18      44      45      85      134
```

Creation of convergence criteria

Some convergence and divergences tolerances are defined.

```
//Creation of convergence criteria
ConvergenceCriteria
ForceTolerance 1.0e-4
MomentTolerance 1.0e-4
ForceMinimumReference 1.0e-3
MomentMinimumReference 1.0e-3
ConstraintMinimumReference 1.0e-7
DisplacementTolerance 1.0e-4
RotationTolerance 1.0e-4
LagrangeTolerance 1.0e-4
DisplacementMinimumReference 1.0e-6
RotationMinimumReference 1.0e-6
LagrangeMinimumReference 1.0e-6
DivergenceReference 1.0e+15
```

Creation of solver options

Here we request Giraffe to apply four cores to solve this case. The linear system solution chosen is Direct.

```
//Creation of solver options
SolverOptions
Processors    4      LinSys Direct
```

Note that the Giraffe input file commands not necessarily has to follow the here proposed sequence. The important is that all the commands here presented are included.

Results and post-processing

Result files are automatically saved during the solving process on the folder “/tutorial17/post/” and “/tutorial17/monitors/”

To post-process results in Paraview™, first we have to open the files “whole_solution_mesh.pvd”, “whole_solution_rendermesh.pvd”, “whole_solution_forces.pvd” “whole_solution_contactsurfaces.pvd”, which are located at the folder “post” in “tutorial17”.

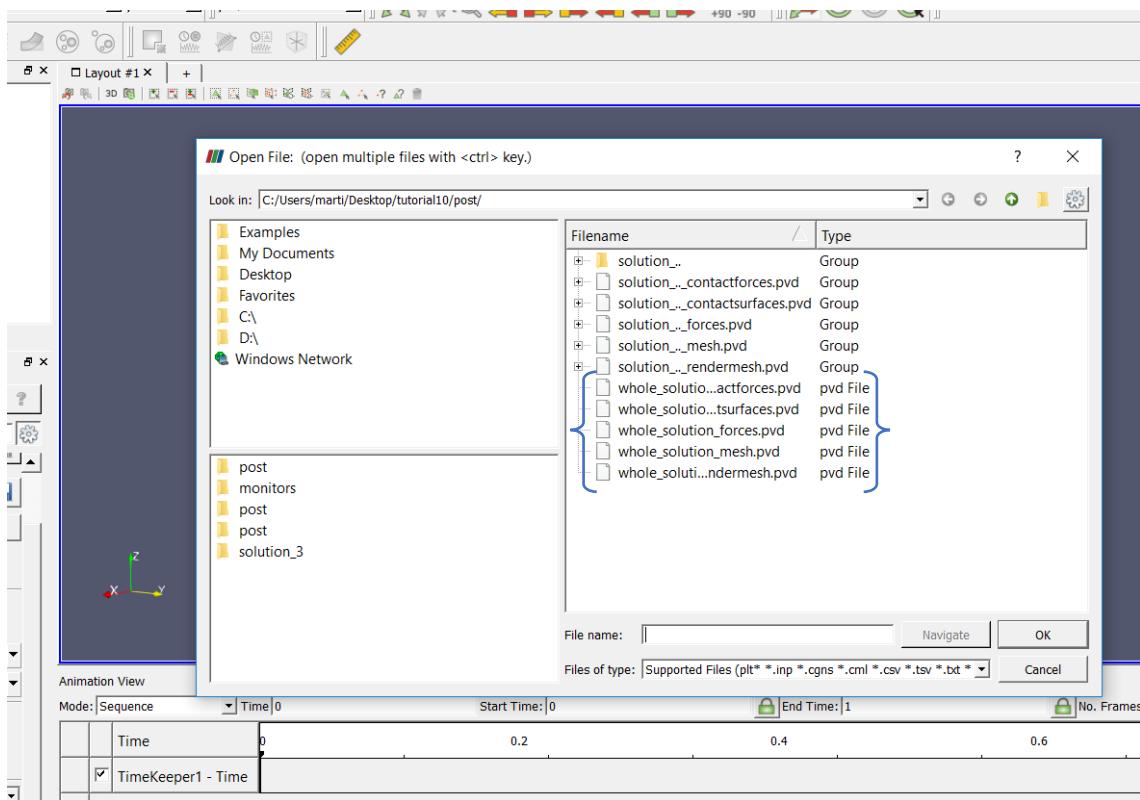


Figure 16-1 - Opening the files of tutorial 17 in Paraview™

Then we can click the preview icon and the “Apply button” on the Pipeline browser. Then, click in the Glyph feature button to create a Glyph, after doing that, we are ready to post-process the results.

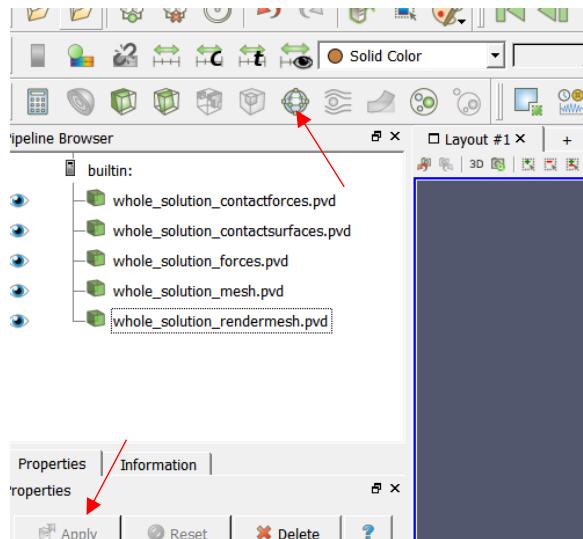


Figure 16-2 - Using Glyphs in Paraview™

Paraview™ post-processing

As we have done, in order to avoid dimensions distortions, we must check the “Camera Parallel Projection” in “Orientation Axes”.

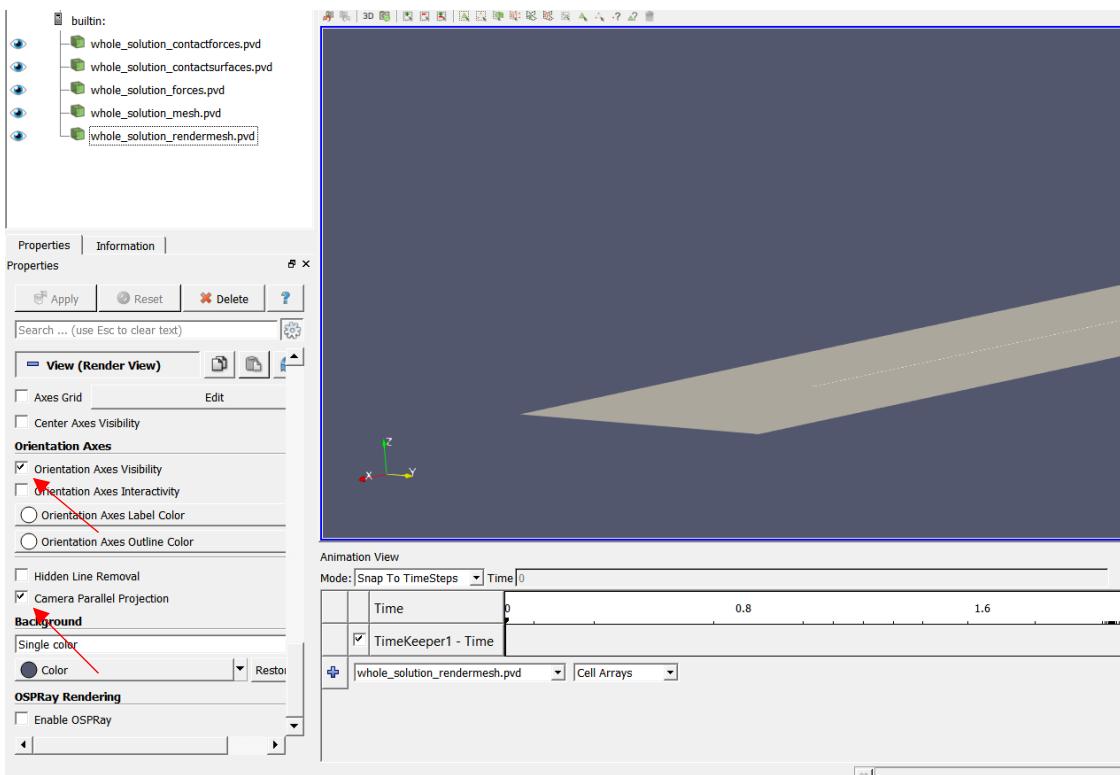


Figure 16-3 - Checking the "Camera Parallel Projection" in Paraview™

Then we able to create glyphs. It is interesting to visualize the contact forces between seabed and mooring line and the force applied to the fairlead node. To do that, go to the pipeline browser and click in “whole_solution_contactforces.pvd” than click in the glyph button. Do the same for the forces, click in “whole_solution_forces.pvd” than click in the glyph button.

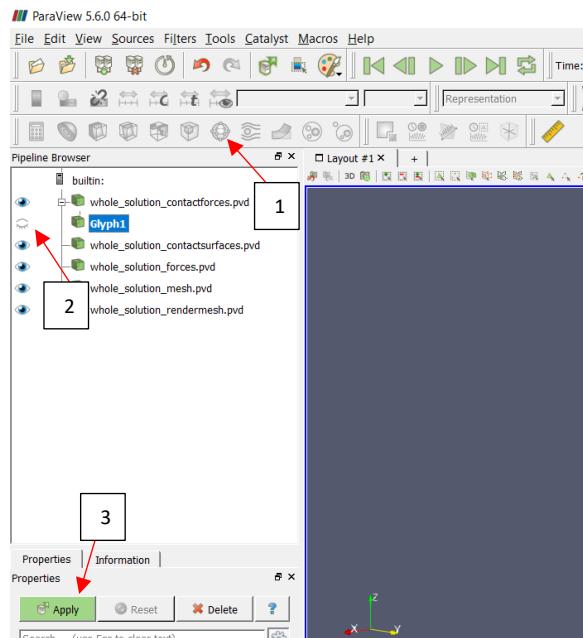


Figure 16-4 - Creating the contact force's glyph

Note that the size of each glyph may be set as proportional to the magnitude of force. If this is the case, the contact forces in the more refined region of the mesh (nearby touchdown point)

are smaller than in the coarser region. This reflects the contact modeling, which considers nodal detection and pointwise (nodal) action.

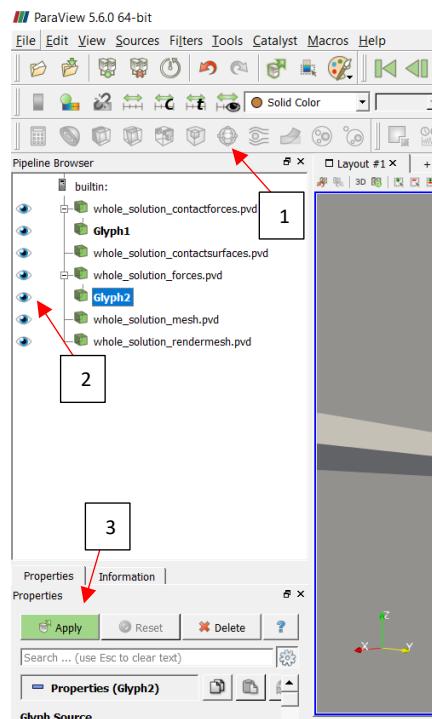


Figure 16-5 - Creating the force's glyph

After creating the glyphs, we have to adjust the scale of the vectors. To do this, first click in the glyph of interest preview icon and. Than in “Orientation” set the orientation array to “normal” and in “Scale” set the scale array also to “normal”. Finally, set the vector scale mode to “scale by magnitude” and change the scale factor to a small number like 0.005 and click Apply. For the force’s glyph the process is analogous, but we choose “force” in the orientation array and scale array, and we can set a smallest scale factor, like 0.0002, as indicated in Figure 16-7.

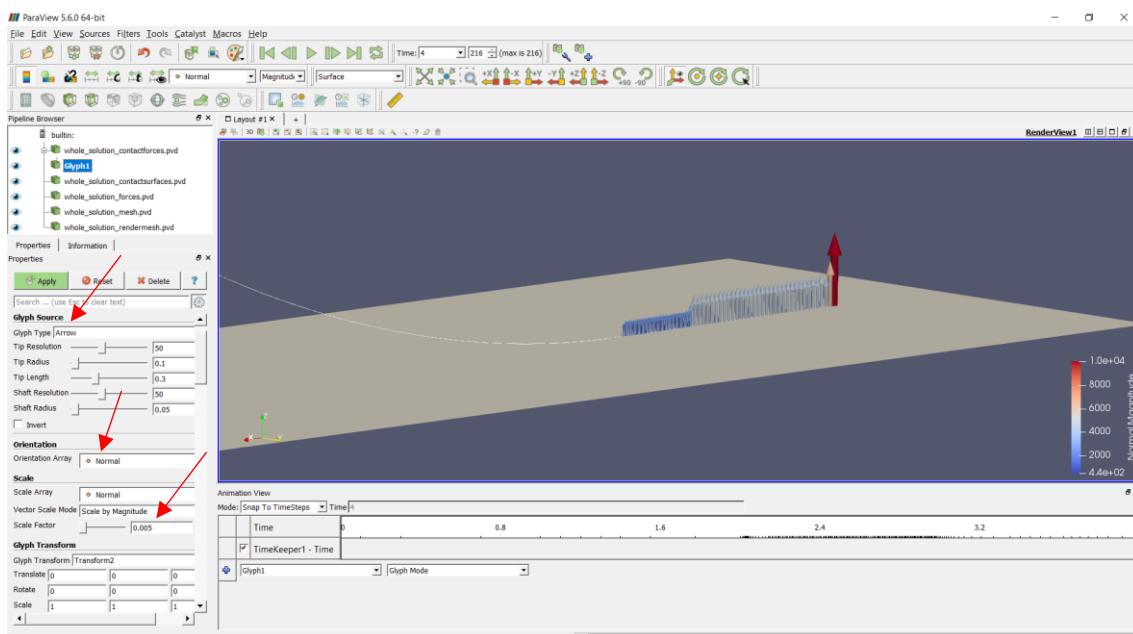


Figure 16-6 - Changing the properties of contact force's glyph

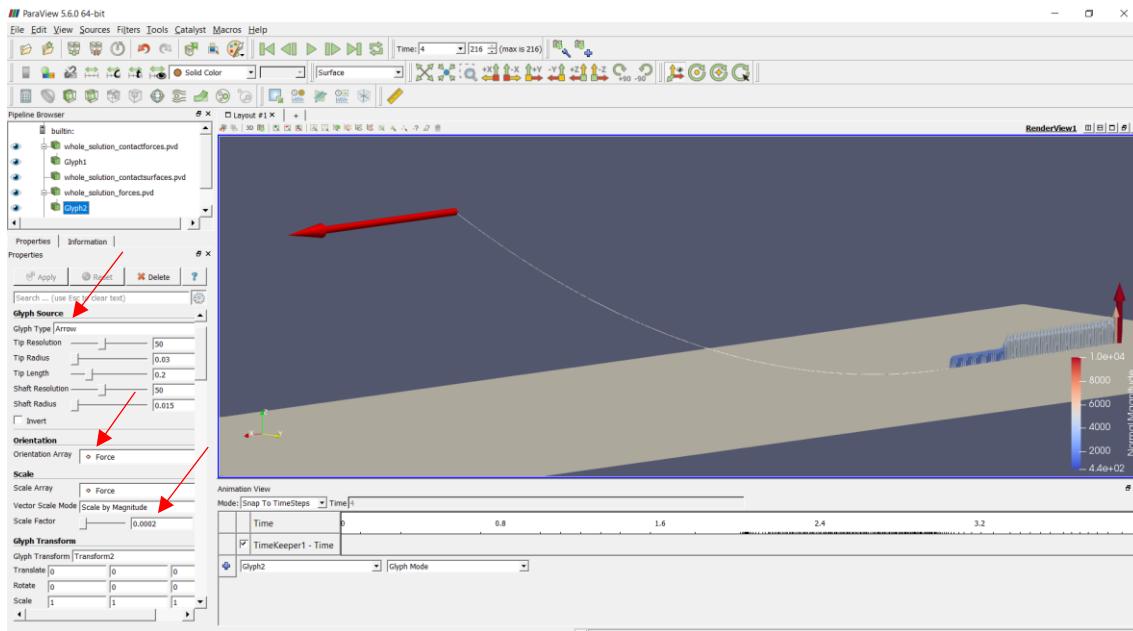


Figure 16-7 - Changing the properties of force's glyph

For a better visualization, we can change the background color. To do that, go to “Background”, in view. Here, we apply a gradient, observing that the color from below it's best to be darker, to simulate the seabed.

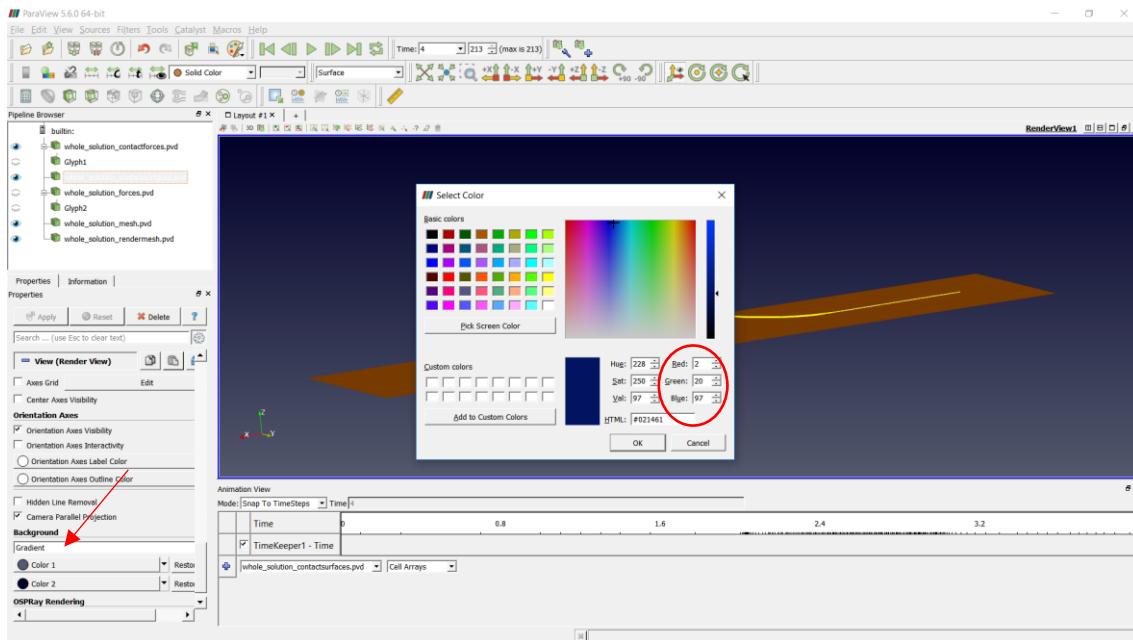


Figure 16-8 - Changing the background color

For a better view, we can also enhance the line width and chance the color. To do that, click in “whole_solution_mesh.pvd” icon and go to properties. The color can be changing in “coloring” and the line width in “Styling”, as showing in Figure 16-9.

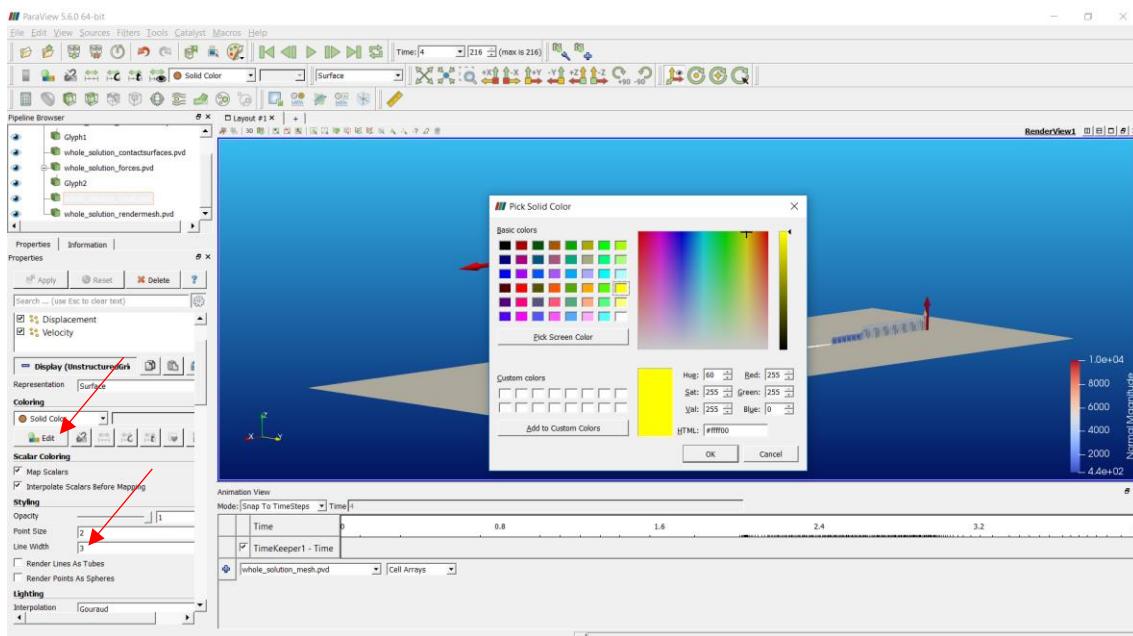


Figure 16-9 - Changing the mesh properties

Finally, we can show an animation of the simulation results. If the user wants to know the values of forces and moments of some elements in a certain instant, we can gather such information in the “monitors” folder, since they are requested prior to simulation solving.

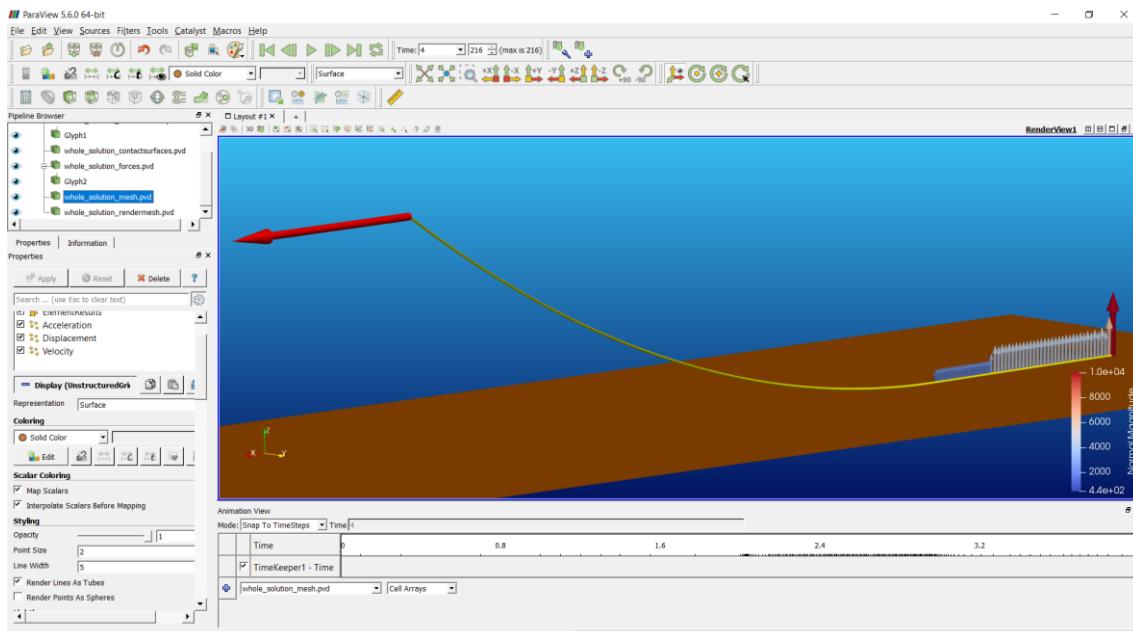


Figure 16-10 - Configuration established before apply sea current

Other post-processing

It is interesting to evaluate the bending moment and the shear force along the mooring line length, especially close to the touchdown zone. First, we have to open the last solution file of the third solution step (before applying the sea current) and copy it to a spreadsheet. The same analysis can be done after sea current is considered, here we opted to analyze before, to avoid

the three-dimensional analysis. So, we can now go to the folder “post”, then to “solution_3” and, finally, open the last converged solution.

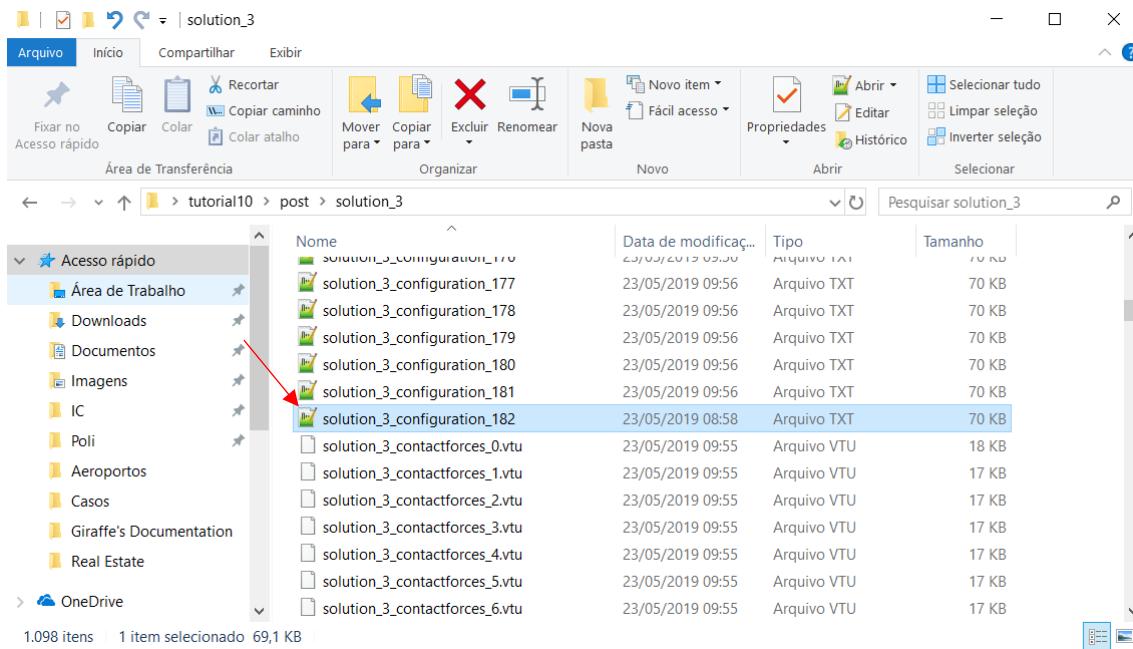


Figure 16-11 - Opening the solution report

To plot the graphics, attempt to the coordinate system and remember that the elements results are given in the local coordinate system (of the element).

Said that, it is our interest here to evaluate the shear force in local direction e_2 and the bending moment around the local axis e_1 .

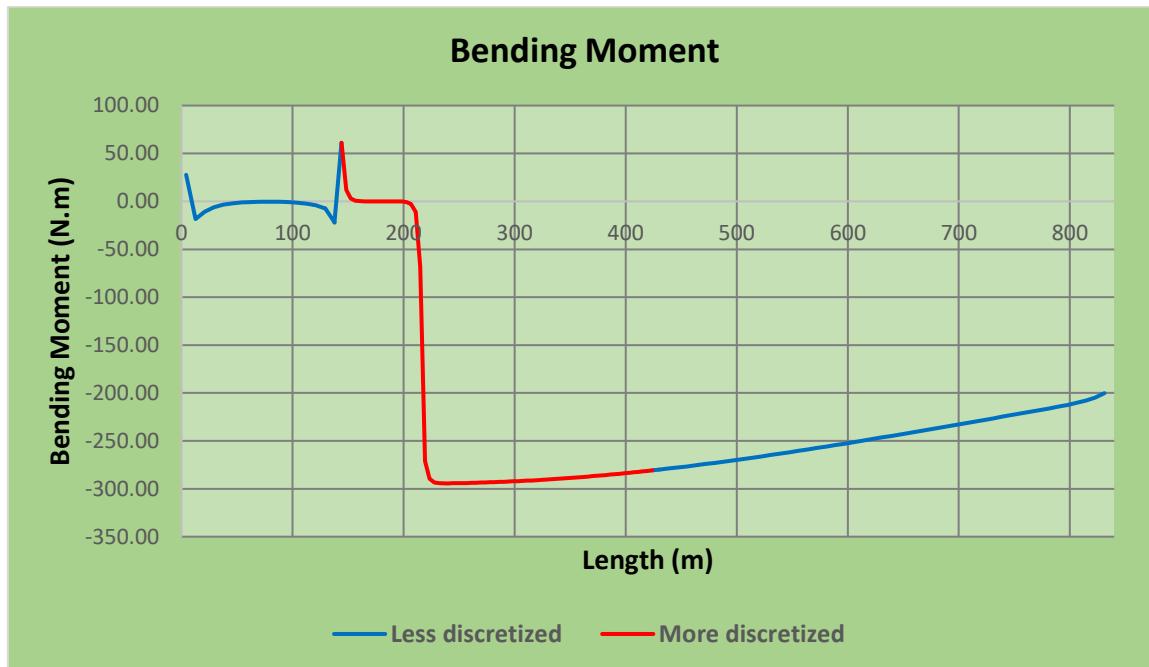


Figure 16-12 - Bending moment along the mooring line length

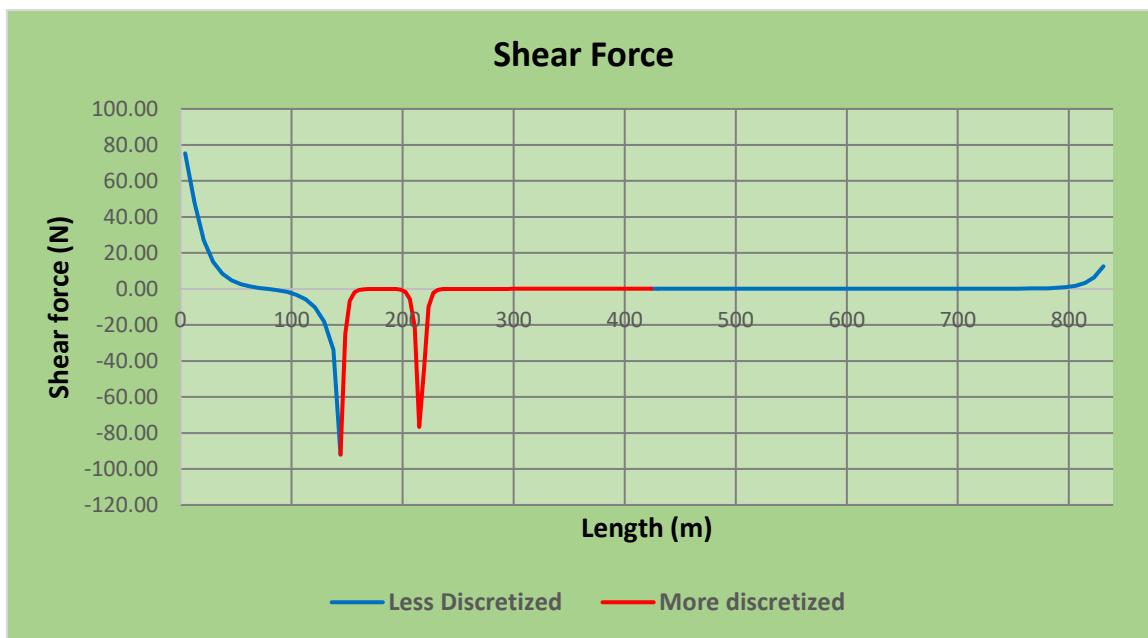


Figure 16-13 - Shear force along the mooring line length

Acknowledgements

Alfredo Gay Neto would like to express his gratitude to all students that collaborated on the development of tutorials and on the elaboration of this document.

References

- [1] A. Gay Neto, P. Pimenta e C. Martins, “Hydrostatic Pressure Load in Pipes Modeled using Beam Finite Elements: Theoretical Discussions and Applications,” *J. Eng. Mech.*, vol. 143, nº 4, p. 04017003, 2017.
- [2] A. Ibrahimbegovic e M. Mikdad, “Finite rotations in dynamics of beams and implicit time-stepping schemes,” *Int. J. for Num. Meth. in Eng.*, vol. 41, p. 781–814, 1998.