



# Manual

## Mecway Finite Element Analysis

Version 28  
2025

## Contents

### *Chapter 1*

Welcome to FEA	7
----------------	---

### *Chapter 2*

Overview of Mecway	9
--------------------	---

- 2.1 Mesh, 9
- 2.2 Analysis, 10
- 2.3 Geometry, 10
- 2.4 Components & Materials, 10
- 2.5 Loads & Constraints, 10
- 2.6 Named Selections, 10
- 2.7 Solution, 10
- 2.8 Configurations, 11
- 2.9 Units, 11

### *Chapter 3*

Viewing and Selecting	12
-----------------------	----

- 3.1 Zoom, Pan, Rotate, 12
- 3.2 Display modes, 13
- 3.3 Selection, 14

### *Chapter 4*

Manual Meshing	16
----------------	----

- 4.1 Creating Tools, 16
- 4.2 Editing Tools, 19

- 4.3 Converting a 2D Mesh into a 3D Mesh, 22
- 4.4 Refinement Tools, 25
- 4.5 Tracing an Image, 27
- 4.6 Symmetry, 28
- 4.7 Mesh Information, 28
- 4.8 Modeling Errors, 29

## *Chapter 5*

### **CAD Models**

**33**

- 5.1 Introduction, 33
- 5.2 Loads and Constraints, 33
- 5.3 Meshing, 33
- 5.4 Gmsh, 34
- 5.5 Assemblies, 34
- 5.6 Alibre Design, 34

## *Chapter 6*

### **Analysis Types**

**35**

- 6.1 Static, 35
- 6.2 Nonlinear Static 3D, 36
- 6.3 Frequency, 37
- 6.4 Dynamic Response, 38
- 6.5 Nonlinear Dynamic Response 3D, 39
- 6.6 Buckling, 39
- 6.7 Thermal, 40
- 6.8 DC Current Flow, 41
- 6.9 Acoustic Resonance, 41

## *Chapter 7*

### **Elements**

**43**

- 7.1 Plane Continuum Elements, 43
- 7.2 Axisymmetric Continuum Elements, 44
- 7.3 Solid Continuum Elements, 44
- 7.4 Shell, 45
- 7.5 Beam, 48
- 7.6 Truss, 52
- 7.7 Spring, 52
- 7.8 Damper, 53
- 7.9 Gap, 53
- 7.10 Tension Only, 54
- 7.11 Fin, 54
- 7.12 Fluid Network Element, 55
- 7.13 Resistor, 55

## *Chapter 8*

### **Materials**

57

- 8.1 Library, 57
- 8.2 Defining a New Material, 57
- 8.3 Mixed Materials, 57
- 8.4 Mixed Elements, 57
- 8.5 Anisotropic Materials, 58
- 8.6 Temperature Dependent Properties, 58
- 8.7 Failure Criteria, 58
- 8.8 Nonlinear Materials, 59

## *Chapter 9*

### **Loads and Constraints**

60

- 9.1 Fixed support, 61
- 9.2 Frictionless support, 61
- 9.3 Pinned support, 61
- 9.4 Elastic support, 62
- 9.5 Compression only support, 62
- 9.6 Displacement, 63
- 9.7 Node rotation, 63
- 9.8 Velocity, 63
- 9.9 Contact, 64
- 9.10 Node-surface coupling, 67
- 9.11 Pre-tension section, 68
- 9.12 Flexible joint on beam, 70
- 9.13 Force, 71
- 9.14 Pressure, 71
- 9.15 Traction, 72
- 9.16 Line pressure, 73
- 9.17 Hydrostatic Pressure, 73
- 9.18 Moment, 74
- 9.19 Gravity, 75
- 9.20 Centrifugal force, 75
- 9.21 Mass, 76
- 9.22 Rotational inertia, 76
- 9.23 Temperature, 76
- 9.24 Node temperature, 77
- 9.25 Thermal stress, 77
- 9.26 Rayleigh damping, 77
- 9.27 Heat flow rate, 78
- 9.28 Heat flux, 78
- 9.29 Internal heat generation, 78
- 9.30 Convection, 79
- 9.31 Forced convection, 79
- 9.32 Radiation, 79
- 9.33 Mass flow rate, 79

- 9.34 Fluid pressure, 80
- 9.35 Impedance, 80
- 9.36 Electric potential, 80
- 9.37 Charge, 80
- 9.38 Current, 80
- 9.39 Robin boundary condition, 81
- 9.40 Symmetry, 81
- 9.41 Cyclic symmetry, 82
- 9.42 Constraint equation, 83

## *Chapter 10*

### **Results**

**84**

- 10.1 Display, 84
- 10.2 File Output, 87
- 10.3 Saved Monitor, 87
- 10.4 Stress Linearization, 87
- 10.5 Volume Mean, 89
- 10.6 Surface Mean, 89
- 10.7 Volume Integral, 89
- 10.8 Surface Integral, 89
- 10.9 Surface Normal Integral, 89
- 10.10 Sum, 89
- 10.11 Formula, 89

## *Chapter 11*

### **Samples and Verification**

**91**

- 11.1 BeamBendingAndTwisting.liml, 91
- 11.2 CompositeBeam.liml, 92
- 11.3 CylinderLifting.liml, 94
- 11.4 PressureVesselAxisymmetric.liml, 95
- 11.5 MembraneActionPlate.liml, 96
- 11.6 SaggingCable.liml, 97
- 11.7 BucklingBeam.liml, 98
- 11.8 BucklingPlate.liml, 99
- 11.9 BucklingPlateNonlinear.liml, 100
- 11.10 PipeClip.liml, 101
- 11.11 BoltedJoint.liml, 102
- 11.12 FinConvection.liml, 104
- 11.13 ConductionConvectionRadiation.liml, 105
- 11.14 OscillatingHeatFlow.liml, 106
- 11.15 CoolantChannel.liml, 108
- 11.16 VibratingFreePlate.liml, 109
- 11.17 VibratingCantileverBeam.liml, 110
- 11.18 VibratingCantileverSolid.liml, 111
- 11.19 VibratingTrussTower.liml, 113

- 11.20 VibratingMembrane.liml, 113
- 11.21 Impeller.liml, 115
- 11.22 VibratingString.liml, 117
- 11.23 PandSWaves.liml, 118
- 11.24 DampedVibratingStrip.liml, 120
- 11.25 WheatstoneBridge.liml, 121
- 11.26 Capacitor.liml, 122
- 11.27 PiezoelectricStack.liml, 123

## ***Chapter 12***

### **File Formats**

**127**

- 12.1 .liml, 127
- 12.2 STEP (.step/.stp), 127
- 12.3 .dxf, 127
- 12.4 .stl, 128
- 12.5 Gmsh (.msh), 128
- 12.6 .unv, 128
- 12.7 Netgen (.vol), 129
- 12.8 Tetgen (.mesh), 129
- 12.9 Polygon File Format (.ply), 129
- 12.10 .jpeg, .png, .bmp, 129
- 12.11 .xyz, 129
- 12.12 Paraview (.vtu and .pvd), 129
- 12.13 ANSYS command file (.txt), 129
- 12.14 CalculiX (.inp), 130
- 12.15 CalculiX results (.frd), 134

## ***Chapter 13***

### **CalculiX Solver**

**135**

- 13.1 Custom model definition, 135
- 13.2 Custom step contents, 136
- 13.3 Modify keyword, 136
- 13.4 Don't generate STEP, 136
- 13.5 No extra contact step, 136
- 13.6 Custom element type, 136

## ***Chapter 14***

### **OpenRadioss Solver**

**137**

- 14.1 Setup, 137
- 14.2 Supported features, 137

## ***Chapter 15***

## Automation 140

- 15.1 Command Line Parameters, 140
- 15.2 Scripts, 140

## *Chapter 16*

---

### Tutorials 143

- 16.1 Quick Start, 143
- 16.2 Viewing and Selecting, 144
- 16.3 Manual Meshing, 150
- 16.4 Solution, 166
- 16.5 Using STEP Files from CAD, 168
- 16.6 Mixed Materials, 172
- 16.7 Assembly with Contact, 172
- 16.8 Static Analysis of a Pressurized Cylinder, 176
- 16.9 Thermal Analysis of a Bar Being Cooled, 179
- 16.10 Thermal Stress, 181
- 16.11 Free Vibration of a Cantilever Beam, 183
- 16.12 Dynamic Response of a Solid Part, 184
- 16.13 Acoustic Analysis of an Organ Pipe, 185
- 16.14 Constraint Equations, 187

## *Chapter 17*

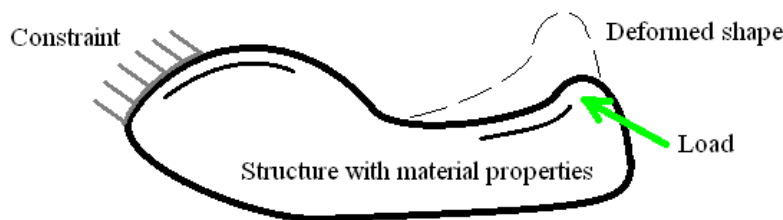
---

### License Agreements 189

- 17.1 Mecway End User License Agreement (EULA), 189
- 17.2 ARPACK, 189
- 17.3 SlimDX, 189
- 17.4 Bouncy Castle, 189
- 17.5 Netgen, Pthreads-win32, ZedGraph, and OCCT version 7.3, 189
- 17.6 GNU Lesser General Public License, 190
- 17.7 OCC CAD Kernel, 192
- 17.8 Turbo Color Map, 194
- 17.9 Intel MKL, 195

## Welcome to FEA

Suppose you want to solve a physical problem such as finding the stresses in an object when some prescribed forces are applied. This is a typical problem for FEA: some type of load is applied to an object and the response calculated subject to specified constraints.



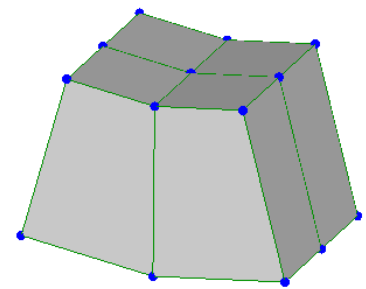
In a mechanics problem, the object might be a gear wheel, the load might be a force applied from another gear, the response might be the stresses throughout the gear wheel. The constraint is that the gear must remain on the shaft.

In a thermal problem, the object might be an electronics enclosure, the load might be heat flow into its interior surface and convective heat flow from its exterior surface. The response would be the temperature distribution in the material.

The finite element method is a numerical technique for gaining an approximate answer to the problem by representing the object by an assembly of simple shapes – the finite elements. Each of these elements is given material properties and is connected to adjacent elements at nodes – special points on the edges of the element. This assemblage of elements connected at their nodes is called a mesh.

Because the elements can easily be assembled into complicated shapes, FEA is a popular and powerful method for realistically predicting the behavior of many engineering structures and components.

The process of using the finite element method is usually iterative – you should solve the model several times to estimate the error in the results and reduce it to an acceptable level. This is called a mesh convergence study.

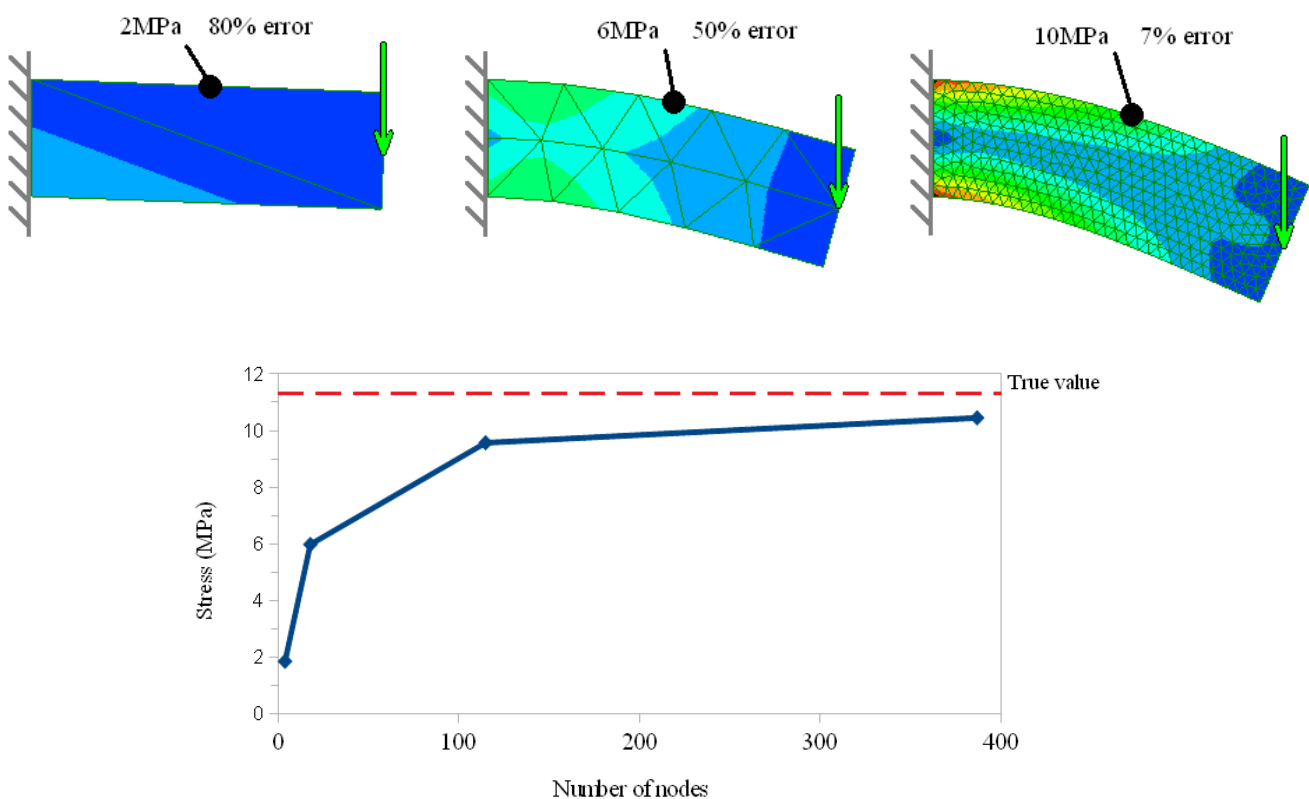


*A mesh consisting of 4 elements and 18 nodes*

### 1. Build the model:

- Either place individual nodes and elements one by one or
- use the mesh creating and editing tools to make it easier or
- import a CAD model and use the automeshing tool to generate a mesh for you.

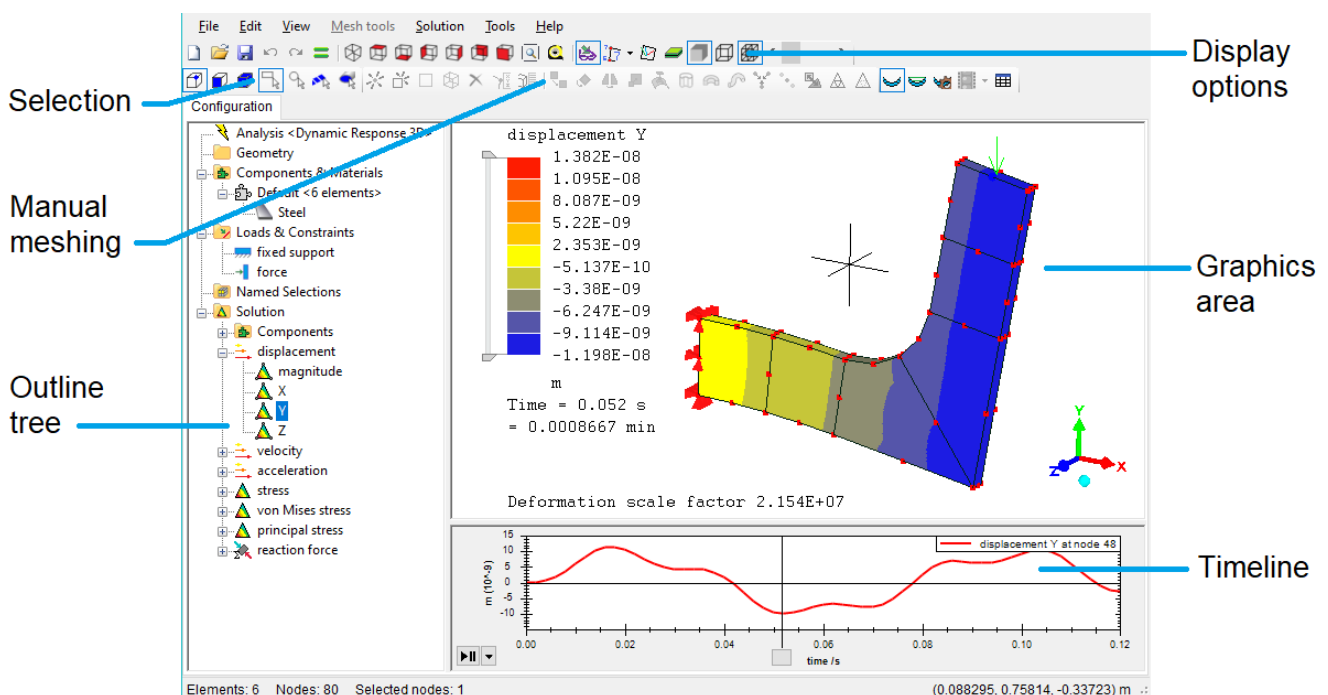
- Assign material properties to the elements and
  - apply loads and constraints to the mesh.
2. Solve the model:
- Define the type of analysis you want e.g. static mechanical, vibrational modes, dynamic response with time, etc.
  - Let Mecway's solver do the work.
3. Refine the mesh and solve again until the results don't change much:
- Record the previous solution values at the points of interest.
  - Refine the mesh, increasing the density of elements in the region of interest.
  - Solve again and repeat until the solution values are similar to the previous solution.



This mesh convergence plot shows a stress component at the top edge of the bar for each of 4 different mesh densities. With around 400 nodes we might consider the error is acceptable. Other element shapes can give faster convergence with fewer nodes.

## Overview of Mecway

This chapter introduces you to the Mecway work-flow for accomplishing your finite element analysis.



### 2.1 Mesh

A finite element mesh consists of nodes (points) and elements (shapes which link the nodes together). Elements represent material so they should fill the volume of the object being modeled. The mesh is displayed in the graphics area which occupies most of the Mecway window.

You can edit a mesh using the **Mesh tools** menu or by selecting parts and using the right-click context menu to access the mesh editing tools.

The other contents of a model are shown in the outline tree on the left hand side of the window. It has several groups containing different types of item listed below. Most of these items can be modified through a context menu which you can access by right-clicking on them.

## 2.2 Analysis

You can change global properties such as analysis type, physical constants, solver settings and output options by editing the Analysis item in the outline tree

## 2.3 Geometry

If you generate a mesh from a STEP file exported from CAD then these files are shown in the Geometry group. Each geometry item can be auto-meshed to generate a mesh.

## 2.4 Components & Materials

A component is an exclusive collection of elements. Every element must belong to exactly one component.

Components are used for assigning materials and controlling the appearance (color and visibility) of elements. All elements in a component share the same material and color. For complex models with logically different parts or features it can be helpful to assign each part to a component to aid in working on the mesh.

Each component containing some elements must have a material assigned to it. The same material can be shared between several components.

You can convert a component to a named selection by selecting its elements then creating a new element selection or adding them to an existing named selection. A named selection containing elements can be converted to a component in a similar way.

## 2.5 Loads & Constraints

This group contains all the loads and constraints in the model. Loads which are applied to named selections show their named selections as child nodes in the outline tree.

## 2.6 Named Selections

A named selection is a non-exclusive collection of nodes, elements or faces. A face is a face or edge of an element. You can use named selections for applying loads and constraints or just to help organize the model. For example, to apply a force to the surface of an object, instead of applying it directly to the faces, you can put all the faces in a named selection and apply the force to the named selection.

## 2.7 Solution

After solving, the results are shown under the **Solution** branch in the outline tree. You can click on a field variable to display a colored contour plot of it. To add more field variables, right click **Solution** or open the **Solution** menu on the menu bar and choose new ones. Some can be generated and displayed immediately while others require you to solve the model again to generate them. If you solve a model when no field variables are listed under Solution, it will produce the minimum set needed to obtain most of the others from without solving again. It also includes von Mises stress where applicable.

## 2.8 Configurations

Configurations allow you to define several different sets of loads and constraints in the same model. For example, you may have one configuration for dead loads and another for dead loads and live loads together. Each configuration has its own separate solution so you can also use them to keep multiple solutions available while changing something in the model.

Use **Edit** → **New configuration** to create a new configuration. Then suppress and unsuppress loads and constraints independently in each configuration or for all configurations at once.

Click the tabs above the outline tree to select the active configuration that will be displayed and solved.

## 2.9 Units

Each numerical value has its own unit. You can change the unit of any quantity by selecting it from the drop-down box next to the number. This will also convert the number into the new unit so it maintains the same physical meaning. An exception is input tables (eg. temperature dependent material properties and time dependent loads) and formulas which are left unchanged.

Density	0.28	lb/in <sup>3</sup> ▼
Thermal expansion coefficient	0	kg/m <sup>3</sup>
Speed of sound	0	g/mm <sup>3</sup>
		lbf.s <sup>2</sup> /in <sup>4</sup>
		lbf.s <sup>2</sup> /ft <sup>4</sup>
		lb/in <sup>3</sup>

The default units for newly created items are determined by previously used units so it learns your preferences.

Pounds-force and pounds-mass are distinguished by the symbols lbf and lb respectively.

## Viewing and Selecting

### 3.1 Zoom, Pan, Rotate

#### 3.1.1 Tool buttons



Rotate to isometric view.



Rotate the view so one set of axes is parallel to the screen.



Fit to window

#### 3.1.2 Keyboard

+	Zoom-in
-	Zoom-out
Arrow Keys	Pan up/down/left/right
Alt + Arrow Keys	Rotate 5° about the horizontal or screen-normal axis
F4 and F5	Rotate 5° about the vertical axis

#### 3.1.3 Mouse

Rotate	Drag with the middle button. Click to position the center of rotation.
Zoom	Rotate the mouse wheel
Pan	Drag with the right button

You can change this default mouse configuration in **Tools** → **Options** → **View control** tab.

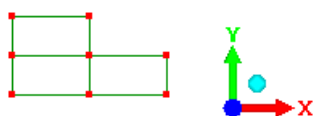
#### 3.1.4 3D mouse

A 3Dconnexion device such as SpaceMouse can be used for: 3D navigation (pan, zoom, rotate), Fit, 6 preset orthogonal views, ISO1 view, and QuickZoom.

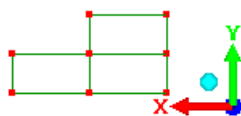
#### 3.1.5 Triad

The triad at the bottom right of the screen can be used to quickly rotate the view parallel to the XY, YZ or ZX planes or isometrically. Left or right click on the arrowheads or the cyan ball to set the view parallel to the axes or to isometric.

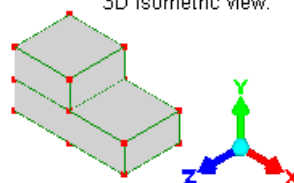
Left click of the Z-axis  
arrowhead



Right click of the  
Z-axis arrowhead



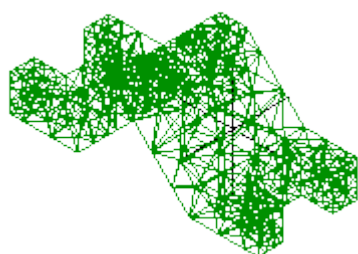
Click the cyan ball for  
3D isometric view.



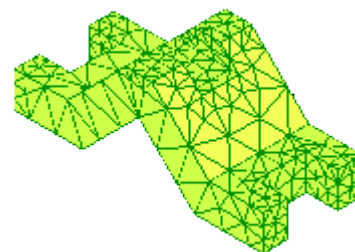
The default orientation has the positive Y direction upwards and positive Z towards the front. You can change to the other popular convention with positive Z upwards and positive Y towards the back using **Tools** → **Options** → **View orientation**. This option determines the orientations obtained by clicking the triad arrowheads, and the 6 default orthogonal views (front, back, etc). It has no effect on the model or solution.

To save the view orientation, pan and zoom settings with the model, use **View** → **Save current view**.

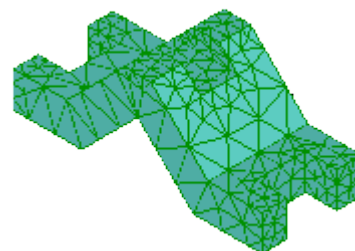
## 3.2 Display modes



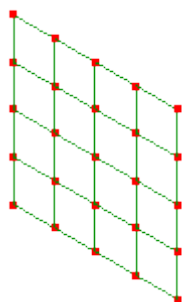
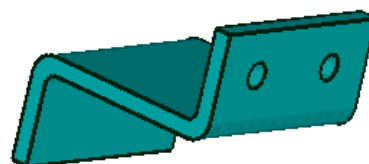
 Show element surfaces



 Show element edges

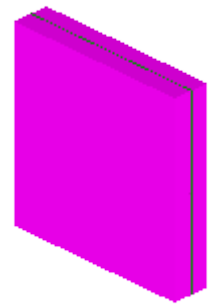


 Show model edges

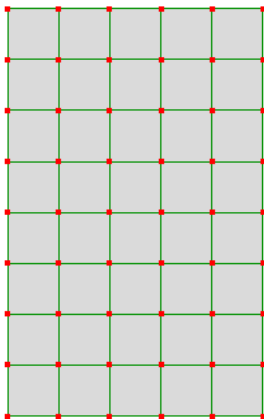
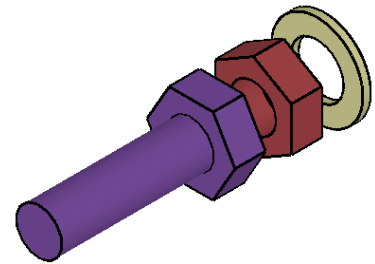
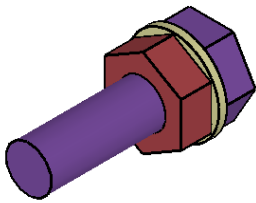




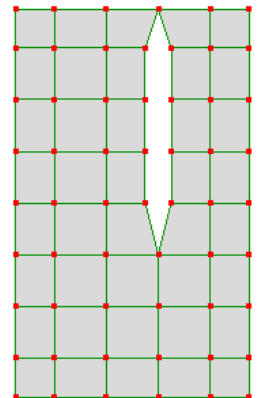
Toggle shell thickness display when element surfaces are displayed.



**Exploded view** in the **View** menu moves disconnected parts away from each other. This is useful for selecting contact surfaces and viewing results in occluded areas.



**Open cracks** in the **View** menu toggles this mode. It helps to show narrow gaps in the mesh where elements appear to be connected but are not sharing the same nodes. When open cracks mode is on, the outside surface of a mesh is shrunk, enlarging any gaps.

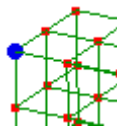


### 3.3 Selection

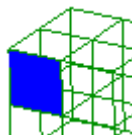
Mecway is selection driven which means to perform most mesh editing tasks you first have to select nodes, element faces or entire elements.



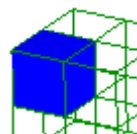
**Select nodes**



**Select faces**




**Select elements**




Changing between these three modes converts any existing selection to the new type.

 **Rectangle selection**

 **Circle selection**

 **Paint selection** Drag the mouse to select faces bounded by sharp edges of the mesh.

 **Edge detecting selection** Click on a face (including shell or beam edges) to quickly select the whole surface bounded by sharp edges of the mesh. If it doesn't detect the correct edges, adjust the **Edge threshold** angle using the slider that appears. This is the largest angle between adjacent face normals that the selection will spread over.

Hold **Ctrl** while selecting to add or remove items from the selection.

Hold the **Shift** key to disable node dragging while selecting nodes with mouse.

**Edit** → **Select nodes by formula** allows you to select nodes according to an inequality of the form

$$f(x, y, z) < g(x, y, z)$$

where  $f()$  and  $g()$  are mathematical functions of position. For example, to select nodes within a sphere of radius 3m, centered on the origin, enter  $x^2+y^2+z^2 < 3^2$  and set the **Position** unit to  $m$ .

For a guided tutorial, see the [Viewing and Selecting](#) tutorial later in this document.

## Manual Meshing

This chapter will explain how to use the tools that are available in Mecway for creating your finite element model. Unlike computer aided design (CAD) software which uses lines, surfaces and solids, finite element analysis software uses only nodes and elements. It is also possible to import CAD models into Mecway and create a suitable mesh with Mecway's automeshing tools, but this is described in Chapter 5 – CAD Models.

A finite element model is a mesh of elements. Each element has nodes which are simply points on the element. Elements can only be connected to other elements node-to-node. An element edge-to-element-node is no connection at all. Elements themselves have very simple shapes like lines, triangles, squares, cubes and pyramids.

Each element is formulated to obey a particular law of science. For example in static analysis, the elements are formulated to relate displacement and stress according to the theory of mechanics of materials. In the case of frequency analysis, the elements are formulated to obey deflection shapes and frequencies according to the theory of structural dynamics. Similarly, in thermal analysis the elements relate temperature and heat according to heat transfer theory. So it is essential that you have an understanding of the underlying physics theory before using finite element analysis software.

When beginning a new model, first check whether or not your choice of element shape is actually supported by the type of analysis. The element shapes that are available to each type of analysis are listed in chapter 6 – Analysis Types.



The manual meshing tools are grouped together by purpose in this manual:

- i. creating tools, that bring into existence a two dimensional mesh
- ii. editing tools, that form and modify the created two dimensional mesh
- iii. tools that will convert a two dimensional mesh into a three dimensional mesh
- iv. refinement tools for converging results


### 4.1 Creating Tools

This section describes in turn each of the tools for creating a mesh. Tools for modifying it are described in Section 4.2.

#### 4.1.1 Quick square / Quick cube

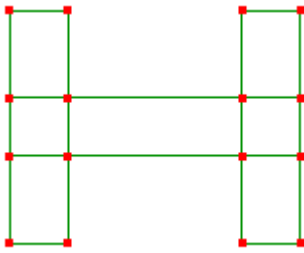
If you're making a simple orthogonal model or want to do a quick test on some feature in Mecway, use the **Mesh tools** → **Create** → **Quick square** or  and the **Mesh tools** → **Create** → **Quick cube** or . They have side lengths of 1 and can be used as building blocks for a model by scaling, re-positioning and refining.

#### 4.1.2 New node


Use the **Mesh tools** → **Create** → **Node...** or  to place nodes by mouse clicks or entering coordinates. If you use mouse clicks, the nodes are placed on a plane parallel to the screen which passes through the origin so it's helpful to use one of the orthogonal views.



#### 4.1.3 New element



To create individual elements

use **Mesh tools** → **Create** → **Element...** or . You can draw elements by clicking on empty space to place each node. If you click an existing node, the element will be linked to that.

The order in which the nodes are clicked will affect the direction in which element subdivisions take effect when using the editing tools. So be consistent in how you are clicking the nodes. For example, you can choose to click the nodes by going counter-clockwise starting at the lower

left corner.

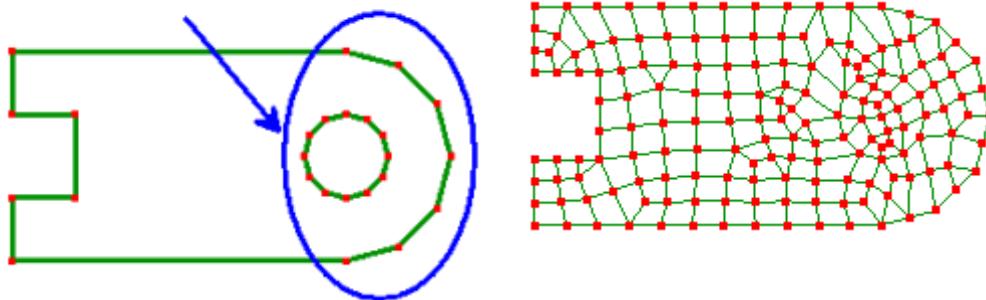
#### 4.1.4 Insert node between

- Select two or more nodes then use the **Mesh tools** → **Insert node between** to create a node at the centroid of the selected nodes. This is useful when laying out nodes for a coarse mesh.

#### 4.1.5 Curve generator

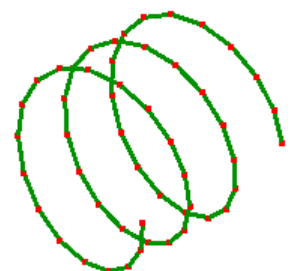
The **Mesh tools** → **Create** → **Curve generator...** creates line element curves defined by 3D parametric equations. If these are boundaries of flat closed shapes, then you can use the two dimensional automesh, **Mesh tools** → **Automesh 2D...** to fill in the shapes.

The circle and the arc was made using the curve generator.



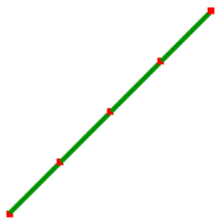
It can also be used to generate 3D shapes made from line2 or curved line3 elements such as a helix using these parameters:

**p initial**      0  
**p final**         $3 \cdot 2 \cdot \pi$       (3 loops)

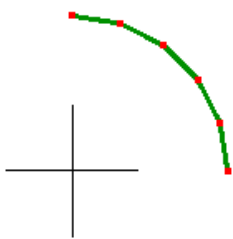


$X = \cos(p) \cdot 1$  (radius is 1)  
 $Y = \sin(p) \cdot 1$  (radius is 1)  
 $Z = p/10$  (pitch is 10)

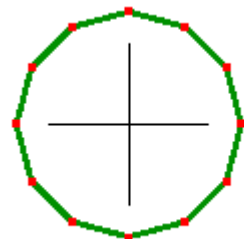
Number of elements 50



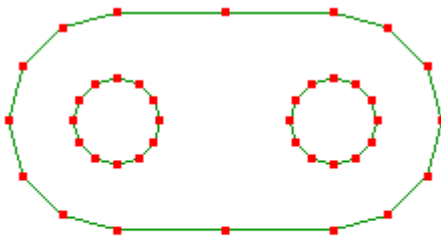
Several shapes have predefined equations so you can create them quickly. These are a straight line, arc, circle, ellipse and parabola.



The most common use of the curve generator is to create arcs and circles. Arcs can be created using either the center, start and end points or by specifying the start, end and any point lying on the arc.

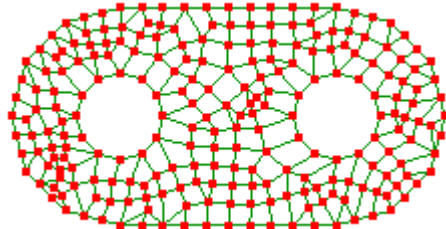


#### 4.1.6 Automesh 2D



The **Mesh tools** → **Automesh 2D...** is used to fill an area bounded by line elements or formed by plane elements with quadrilateral and triangle elements. After a successful automesh the original elements will no longer exist. It can only create elements that lie in a plane, which can be in any orientation.

The automesher will fill the entire bounding area with elements including any holes. You will then have to manually delete the elements in the hole



areas.

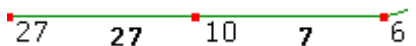
Curved edges can be defined using quadratic elements (line3, tri6, or quad8) so they don't become faceted when the mesh is very fine.

Depending on how you created and edited your model, you may have places where two parts of the bounding lines appear to be joined but are not. For example these two line elements look as though they are connected with each other, but on displaying their node numbers, it's clear that there are actually two overlapping nodes. This means the line elements are not connected to each other.




If there are any unconnected line elements, the automesher will fail.

Therefore, before running the automesher, use the editing tool **Mesh tools** → **Merge nearby nodes...** to replace any overlapping nodes with a shared node, thereby connecting all elements.



If the automesher default values make a mesh with only a few large elements, re-run the automesher using a smaller value for the **Maximum element size**. If you don't know what maximum size value to

specify, use the tape measure tool  to measure the smallest line segment. It will give a dynamic read-out as you click and drag from one node to another.

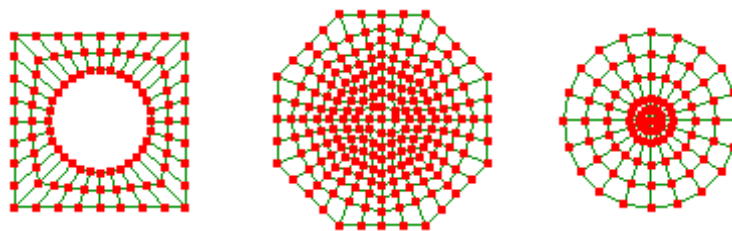
#### 4.1.7 Automesh 3D

**Mesh tools** → **Automesh 3D...** remeshes a solid part. It can also fill in the volume of a closed shell mesh and can include spherical regions of local refinement.

The new mesh is removed from any named selections or loads and all its elements are placed in the same component even if the original mesh belongs to multiple components.

#### 4.1.8 Plate mesh

In **Mesh tools** → **Create** → **Plate mesh** there are templates for creating simple shapes like a circular, square and octagonal plates with or without holes. These templates are simple to use and are self-explanatory.



These shapes can be extruded or revolved to generate three dimensional solids.

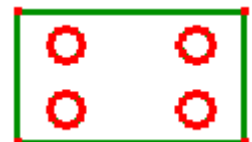
## 4.2 Editing Tools

#### 4.2.1 Move

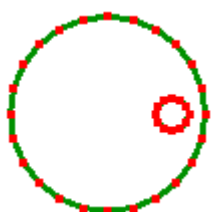
The **Mesh tools** → **Move/copy...** is used to reposition or duplicate nodes or elements. You can choose a coordinate system from Cartesian, cylindrical or spherical polar coordinates. The polar coordinates allow you to change the radial size of an object while keeping its thickness constant.



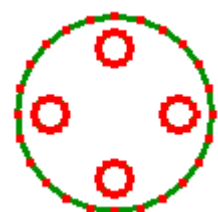
Note the copy check box. If this is ticked, the selection is both moved and duplicated. Bear in mind that the copies are not connected to each other. Use the **Mesh Tools** → **Merge nearby nodes...** to connect them.



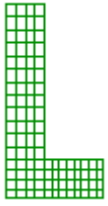
#### 4.2.2 Rotate



The **Mesh tools** → **Rotate/copy...** is used to rotationally reposition nodes or elements. With the copy option selected, nodes and elements can be duplicated.

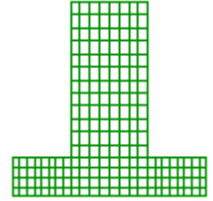


#### 4.2.3 Mirror

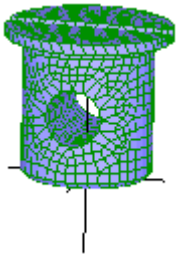


The **Mesh tools** → **Mirror/copy...** is used to reposition nodes or elements by reflecting them about either the XY, YZ or ZX planes. When the **Copy** option is selected, it can be used to mirror meshes.

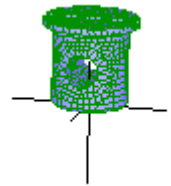
At the mirror joint the elements will not be connected so you will have to use the **Mesh tools** → **Merge nearby nodes...** to make it a continuous mesh.



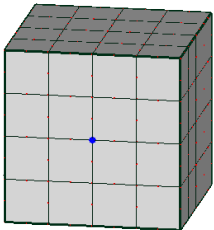
#### 4.2.4 Scale



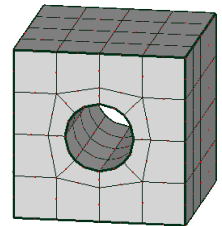
The **Mesh tools** → **Scale/copy...** is used to re-size either the entire mesh or the selected items.



#### 4.2.5 Hole



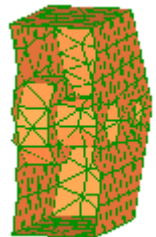
**Mesh tools** → **Hole...** cuts a circular hole through a solid or shell mesh. Define the center point by selecting a node. Choose the direction as either normal to the surface, or parallel to the X, Y or Z axis. Define the diameter by entering its value. It works best on regular hex or quad meshes and may produce some badly shaped elements with more complex meshes.



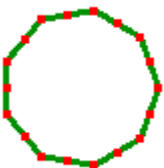
#### 4.2.6 Hollow



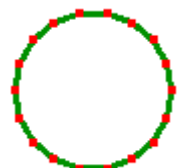
The **Mesh tools** → **Hollow** is used to convert a solid mesh into a shell mesh.



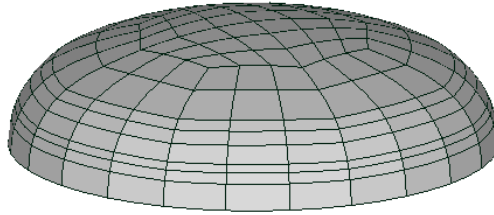
#### 4.2.7 Project onto surface



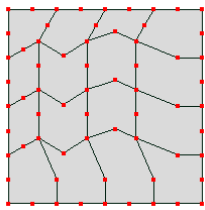
The **Mesh tools** → **Project onto surface...** is used to deform a mesh into a smooth curved shape of sphere, torispherical dome, cylinder or cone or the shape of another mesh.



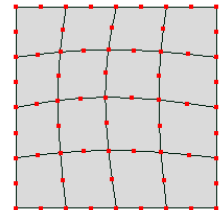
You can use it to create smooth curves on a mesh that has become faceted after refining, to radius a corner or create a pressure vessel end from a flat disk.



#### 4.2.8 Smooth



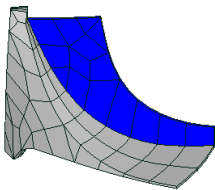
**Mesh tools** → **Smooth** improves the shapes of distorted elements by moving each node towards the mean position of its neighbors. The shapes of existing surfaces are preserved and edges are unchanged. You should avoid repeated use of this tool on coarse curved surfaces because moving the nodes on the surface slightly modifies the shape of the new surface defined by those nodes.



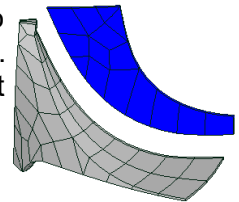
#### 4.2.9 Transfer displacements from solution

**Mesh tools** → **Transfer displacements from solution** deforms the model according to the displacement field variables in the currently selected solution data set. You can specify a scale factor which each displacement value will be multiplied by.

#### 4.2.10 Disconnect elements



To separate part of the mesh from the rest, select the elements to be separated then use **Mesh tools** → **Disconnect elements**. There is no immediate graphical change but you can verify that they are disconnected using **View** → **Exploded view**.




#### 4.2.11 Merge nearby nodes

The **Mesh tools** → **Merge nearby nodes** will ensure that the elements are connected node-to-node by replacing overlapping nodes with a single shared node. Meshing operations such as **Mesh tools** → **Refine** → **Custom...** or **Mesh tools** → **Move/Rotate** with the Copy option will create meshes that are not connected. Separate files assembled using **File** → **Import...** will also not be automatically connected to each other at the mating surface.

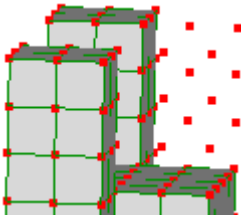
If some nodes are selected before using this tool then only the selected nodes will be considered. However if you check **Merge other nodes into selected nodes** then the selected nodes will not be moved but any other nodes within the tolerance distance will be merged with them.

The **View** → **Open cracks** tool will expose unconnected elements. It shrinks elements slightly to open up any existing gap between adjacent faces of unconnected elements.


To eliminate these gaps use the **Mesh tools** → **Merge nearby nodes** to delete overlapping nodes. You have to specify a radial distance within which two or more nodes will be replaced by a single

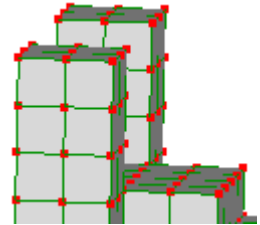
node. Too small a value and some overlapped nodes will not be eliminated. Too large a value and you risk collapsing elements as they lose a node. Use the **Tape Measure** tool  to determine the smallest distance between two nodes in your mesh, then use a value smaller than this so that elements don't collapse. You will notice a reduction in the number of nodes shown in the status bar after using this command if it merged any nodes.

#### 4.2.12 Delete unused nodes



The **Mesh tools** → **Delete unused nodes** will remove nodes not belonging to an element, constraint equation or node-surface coupling as its reference node.

If you use **Edit** → **Delete elements and retain nodes**, the nodes will be left behind. If you can't see them, activate the node select mode .



#### 4.2.13 Invert

Solid elements can sometimes be formed inside-out. This will cause the solver fails with a message about incorrect element topology. You can select the affected elements and use **Mesh tools** → **Invert** to fix their topology. Invert can also be useful for flipping shell elements so their top faces are all on the same side.

#### 4.2.14 Deleting items

To delete elements, first select them or any of their faces then press **Del**. This will also delete any unused nodes left behind. To delete elements without deleting their nodes press **Ctrl + Del**.

To delete a component along with all its elements and their nodes, right click the component in the outline tree and select **Delete**.

To delete nodes, first select the nodes then press **Del**.

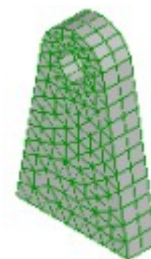
### 4.3 Converting a 2D Mesh into a 3D Mesh

Once a 2D plane mesh has been created it can be extruded, revolved or lofted to create a 3D model.

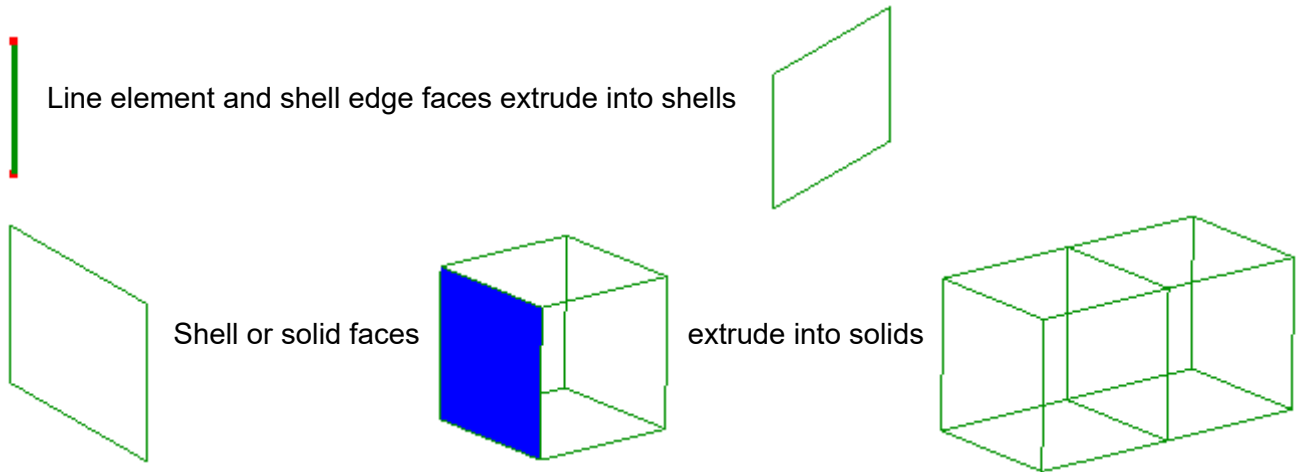
#### 4.3.1 Extrude



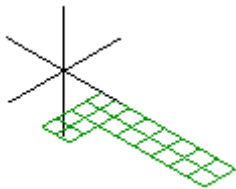
Use **Mesh tools** → **Extrude...** to create a 3D solid mesh.



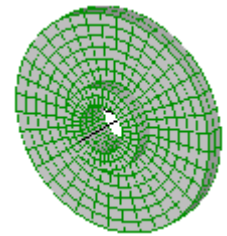
Nodes extrude into line elements.



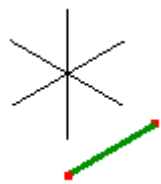
#### 4.3.2 Revolve



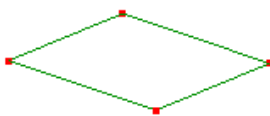
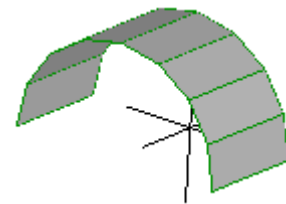
Use **Mesh tools** → **Revolve...** to create a 3D solid mesh. Any nodes on the axis of revolution will be duplicated for each element, making a disconnected mesh. To fix this problem, use **Mesh tools** → **Merge nearby nodes** afterwards.



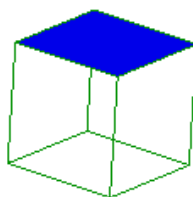
Nodes revolve into line3 elements.



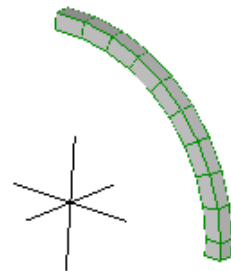
Line elements and shell edge faces revolve into shells



Shell or solid faces



revolve into solids

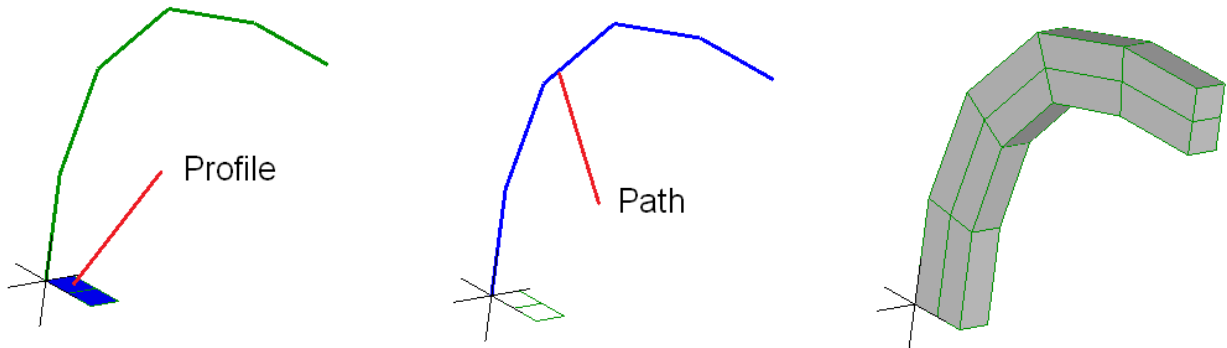


#### 4.3.3 Sweep

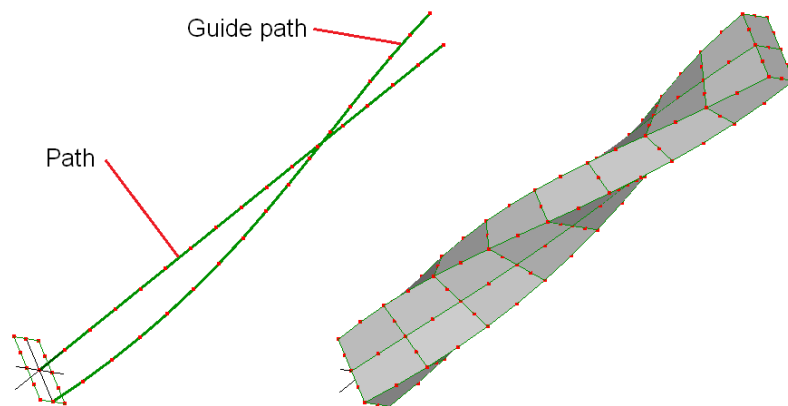
The sweep tool performs extrusion along a defined path which could be curved or have non-uniform element spacing.

After starting the tool with **Mesh tools** → **Sweep...**, select the faces that define the profile then press the corresponding **Use selection** button. Next, select the faces that define the path followed by **Use selection**. If the profile faces are planar they will be swept into solids. If they are the edges of shells or

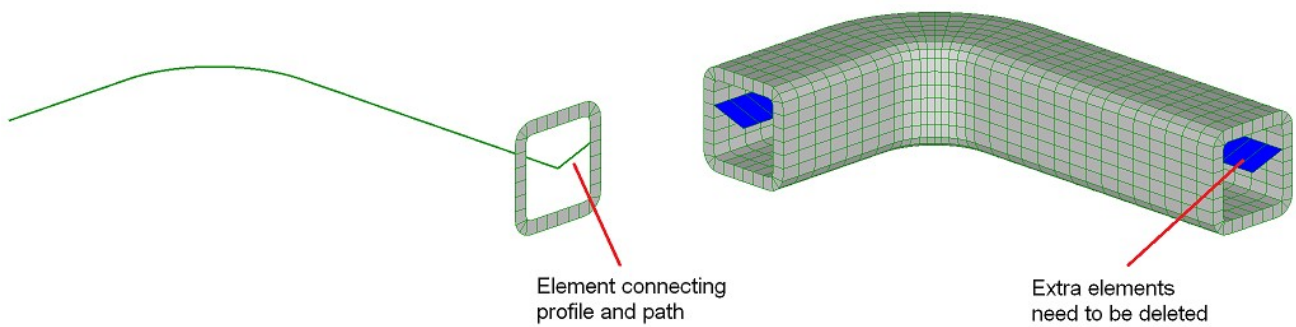
lines, they will be swept into shells. The path must consist of the edge faces of line or shell elements and must have one end node in common with the profile.



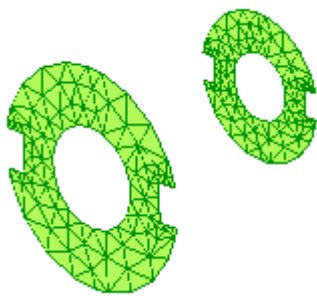
If you need to control the orientation of the profile as it is extruded, for example to make a helix or twisted shaft, you can define a guide path. Each face in the guide path determines the orientation of the profile at the corresponding face of the path. The guide path must have one end node in common with the profile. If it contains more faces than the path, the additional faces are ignored. If it contains fewer faces than the path, the orientation beyond the last guide path face remains the same as it was at the last guide path face.



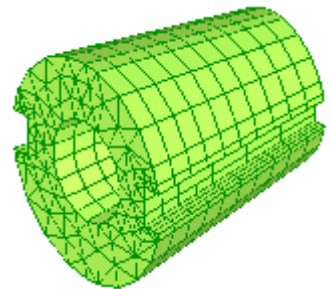
If the path or guide path doesn't intersect the profile, you can add a line element connecting them with its face being part of the profile. This will result in extra elements that you need to delete after performing the sweep.



#### 4.3.4 Loft



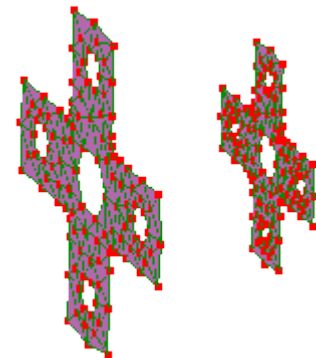
The loft tool fills the gap between two profiles having matching nodal patterns with solid elements. This can be used for creating tapered parts. The order of the node numbers must be identical on each profile with the only difference being a constant offset. You can use **Mesh tools** → **Move/copy** to generate a 2<sup>nd</sup> mesh which satisfies this requirement.



### 4.4 Refinement Tools

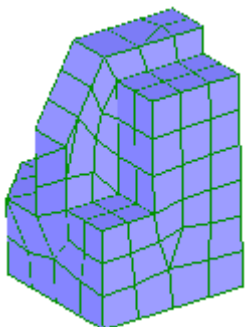
An individual element cannot accurately represent a complicated change in the field values across it. It approximates it as either a constant, linear or quadratic function. Therefore you need smaller elements in regions of the mesh where the results change rapidly.


To determine the areas that need mesh refinement you will first need to solve a coarse mesh and look at the color contour plot of the solution. If you find the field value is changing by a large amount within a single element, that tells you the area may need more elements.

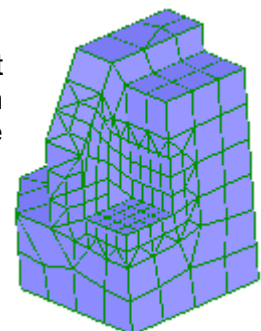


Checking for mesh convergence is a crucial part of finite element analysis because without an estimate of the error, the results may be wildly wrong. To do this, you should solve the same model with several different levels of mesh refinement and note down the important field values for each mesh. If additional refinement leads to only a small change in the results, such as 3%, then it is said to have converged and no further refinement is needed.

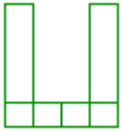
#### 4.4.1 Refine x2



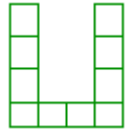
**Mesh tools** → **Refine** → **x2** or  replaces each element with two elements along each edge. It also inserts transition elements so that the refined part of the mesh will be correctly connected to the remaining unrefined part.



#### 4.4.2 Refine Custom



**Mesh tools** → **Refine** → **Custom...** is used to subdivide elements by specifying the number of subdivisions along each of three directions. If no elements are selected, it uses the entire mesh.



#### 4.4.3 Refine x3 (1D/2D elements only)

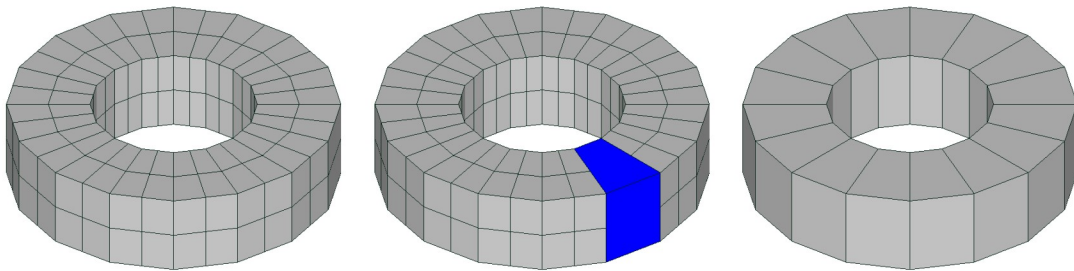
**Mesh tools** → **Refine** → **x3 (1D/2D elements only)** refines the selected nodes or elements by subdividing each element into three elements along each direction. It can be used with line2/3, tri3/6 and quad4/8 elements.

#### 4.4.4 Unrefine x2

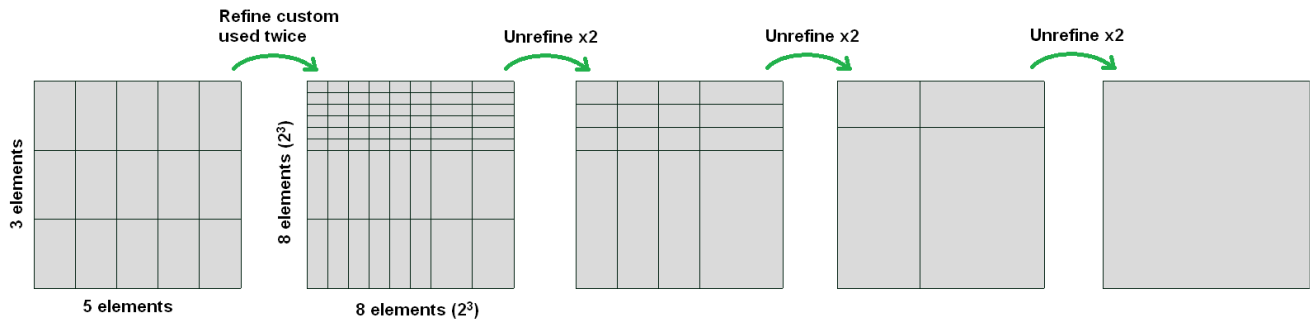
**Mesh tools** → **Refine** → **Unrefine x2** reverses the effect of a globally applied Refine x2 to produce the original coarser mesh. It supports all element shapes except pyramid and wedge (pyr5, pyr13, wedge6, wedge15).

Loads and constraints, named selections and components are maintained where they are the same across all the elements or faces that make up a single unrefined element or face respectively.

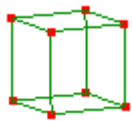
The unrefine algorithm requires a corner node to start from. A corner here is a node that's used by only one element of the same shape. Some meshes have no corners, such as a cylinder or sphere. In those cases, select a subset of the elements which does have a corner and unrefine them first. Then select the remaining elements (**Edit** → **Invert selection**) and unrefine them as shown below.



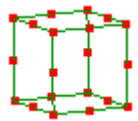
If the mesh has a different density along each edge, such as produced by Refine Custom, use **Mesh tools** → **Refine** → **Custom** on some of the elements so that each edge of a region has  $2^n$  elements then apply the **Mesh tools** → **Unrefine x2** tool  $n$  times as shown below.



#### 4.4.5 Change element shape



Higher order elements are often more efficient. Convert between linear and quadratic elements using **Mesh tools** → **Change element shape...**

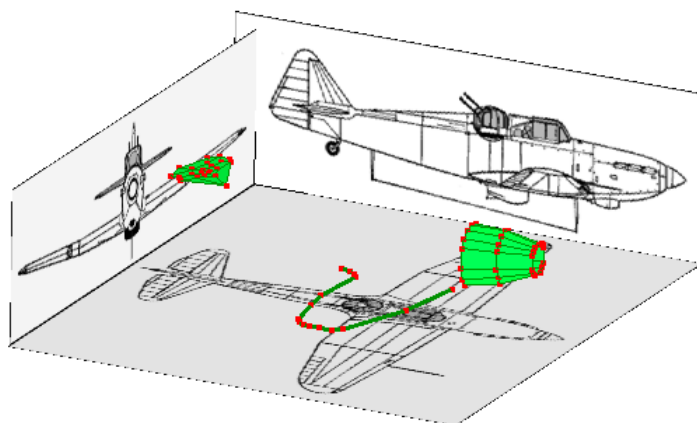
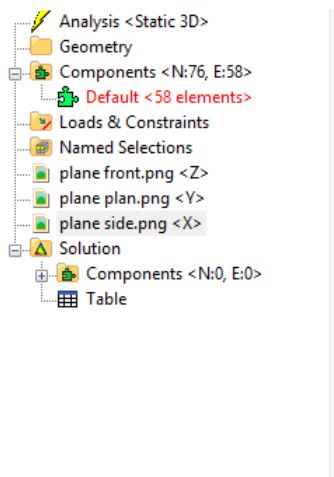


It can also replace some element shapes with templates of different shapes, such as converting a quadrilateral into two triangles. The choice of element shapes will be enabled or disabled according to the element shapes present, so you might have to repeat this step to get the final desired element.

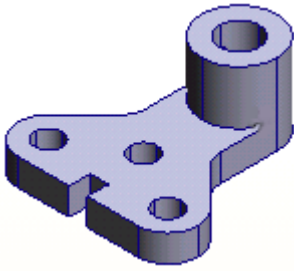
### 4.5 Tracing an Image

If you have a projected image (jpg, png or bmp) of a part, you can open it with Mecway and it will be displayed in the background on one of the 3 coordinate planes so you can manually place nodes in the correct locations. The image file will be linked to the model rather than embedded in it. If you move the liml file, you should move the images with it.

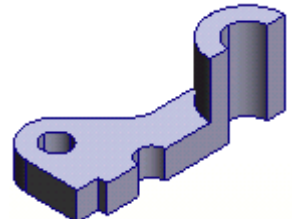
For example, to trace a 3D outline from a 3-view orthographic projection drawing, import (**File** → **Import**) an image file containing each view and place it in a different orientation by choosing **XY plane**, **YZ plane** or **ZX plane** in the dialog box that appears. If an image is misaligned, you can adjust its position by right clicking the file name in the outline tree and entering a distance for **Horizontal offset** (rightward) or **Vertical offset** (upward). After the images are set up, switch to an orthogonal view by clicking an arrowhead of the orientation triad and use the **New element** or **New node** tool to place nodes by mouse clicks.



## 4.6 Symmetry



If the geometry, loads & constraints are symmetric (*mirror symmetry*), a model size can be reduced to half or quarter.



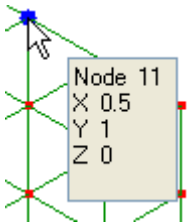
When taking advantage of *mirror symmetry* you must enforce constraints at the plane of symmetry. In static analysis the nodes in the plane of symmetry must be constrained so that they do not move out of that plane otherwise a gap or penetration will occur which in reality is not present in the full model. For elements with rotational degrees of freedom like shells and beams, each node that lies in the plane of symmetry should be constrained to have no rotation about either of the two axes that also lie in the plane of symmetry. In thermal analysis there should be no heat flow across a plane of symmetry, which is a condition that is automatically enforced where no other boundary conditions are specified. The same concept extends to DC current flow analysis.

Take care when assuming mirror symmetry for frequency or buckling problems because non-symmetric modes will be missed.

*Cyclic symmetry* which occurs in turbines, fans, etc can be taken advantage of by modeling only a segment containing the cyclic feature rather than the whole wheel. The node patterns must match on both sides of the segment.

## 4.7 Mesh Information

### 4.7.1 Nodal co-ordinates



Click a node for a readout of its co-ordinates.

### 4.7.2 Length



Activate the tape measure tool, click and drag from one node to another.

In the solution where displacement field values exist, it also reports the deformed distance by adding the displacements to the node coordinates.



### 4.7.3 Surface area

**Tools** → **Surface area** reports the area of selected faces. It also reports the area after deformation if the solution is displayed, contains displacement field variables, and is not a mode shape.

#### 4.7.4 Volume

**Tools** → **Volume** will report the volume of selected elements. If no elements are selected, it will report the full mesh volume. It also reports the volume after deformation if the solution is displayed, contains displacement field variables, and is not a mode shape.

#### 4.7.5 Mass

**Tools** → **Mass** calculates the mass of the model or selected nodes or elements. It includes both density and point mass.

#### 4.7.6 Center of mass

**Tools** → **Center of mass** displays the center of mass (center of gravity) of the model or selected nodes or elements. It includes both density and point mass.

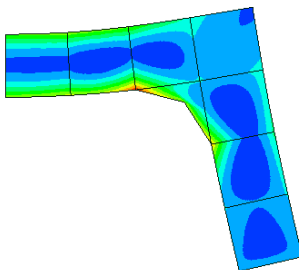
#### 4.7.7 Find duplicate elements

Sometimes two or more elements share the same nodes so they appear as one and cause unexpectedly high stiffness or other errors. **Tools** → **Find duplicate elements** finds elements that mutually have all their corner nodes belonging to another element. It selects all the additional elements so you can see where they are and delete them with the **Del** key.

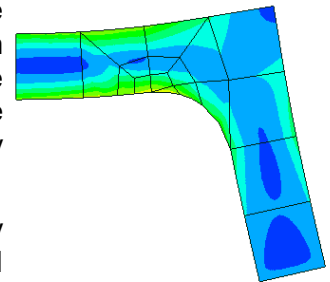
### 4.8 Modeling Errors

Results can only be as accurate as your model. Use rough estimates from hand calculations, experiment or experience to check whether or not the results are reasonable. If the results are not as expected, your model may have serious errors which need to be identified.

#### 4.8.1 Too coarse a mesh



A finite element solution gets more accurate as the mesh is refined. An efficient way of refining a mesh is to concentrate the mesh refinement in those areas where the accuracy can be improved, while leaving unchanged those areas that are already accurate.



You will have to run at least one model to identify the areas where the values are changing a lot and the areas where values are remaining more or less

the same. The second run will be your refined model.

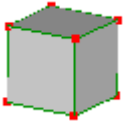
Refine areas that see large changes in value. Do not refine areas where values are more or less the same; it will only bloat the size of the model.

#### 4.8.2 Wrong choice of elements

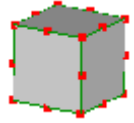
Bending problems with plate-like geometries such as walls, where the thickness is less in comparison to its other dimensions, should be modeled with either shell elements or quadratic solid elements like the 20 node hexahedron or the 10 node tetrahedron. Shell, beam and membrane elements should not be used where their simplified assumptions do not apply. For example beams or shells that are too

thick or membranes that are too thick for plane stress and too thin for plane strain. In each of these cases solid elements should be used.

#### 4.8.3 Linear elements



Linear elements (elements with no mid-side nodes) are too stiff in bending so they typically have to be refined more than quadratic elements (elements with mid-side nodes) for results to converge.

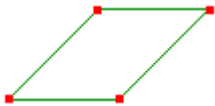


#### 4.8.4 Severely distorted elements

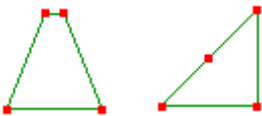
Element shapes that are compact and regular give the greatest accuracy. The ideal triangle is equilateral, the ideal quadrilateral is square, the ideal hexahedron is a cube, etc. Distortions tend to reduce accuracy by making the element stiffer than it would be otherwise, usually degrading stresses more than displacements. However, slight to moderate distortions do not have an appreciable effect on accuracy. The reality is that shape distortions will occur in FE modeling because it is quite impossible to represent structural geometry with perfectly shaped elements. Any deterioration in accuracy will only be in the vicinity of the badly shaped elements and will not propagate through the model (St. Venant's principle).



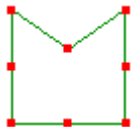
Avoid large aspect ratios. A length to breadth ratio of generally not more than 3.



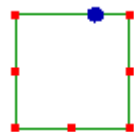
Highly skewed. A skewed angle of generally not more than 30 degrees.



A quadrilateral should not look almost like a triangle.

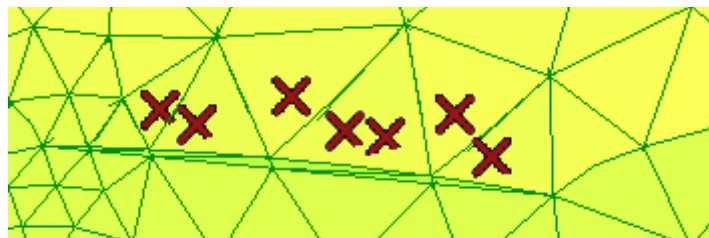


Avoid strongly curved sides in quadratic elements.



Off center mid-side nodes.

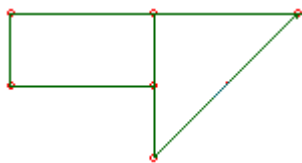
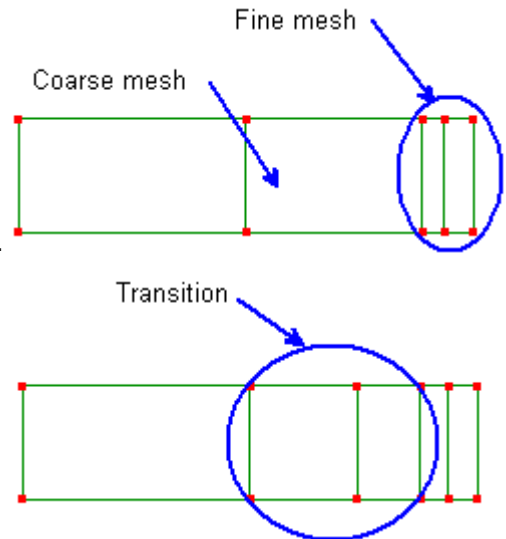
If an element is too severely distorted to solve, it will usually be shown with a red cross in it. This happens if it is inverted, self-intersecting, or collapsed.



If these bad elements appear after meshing a STEP file, they are usually quadratic elements with edges that are too steeply curved. In that case, turn on **Straighten bad elements** in **Geometry** → **file name** → **Meshing Parameters** and generate the mesh again.

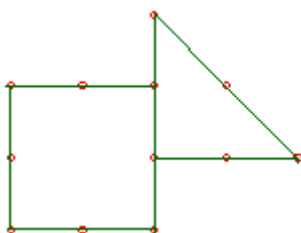
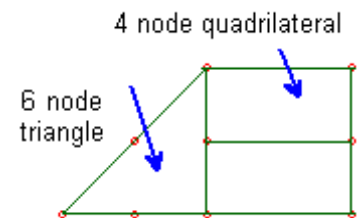
#### 4.8.5 Mesh discontinuities

Element sizes should not change abruptly from fine to coarse. Rather they should make the transition gradually.



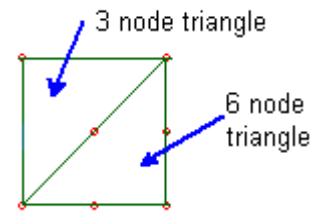
Nodes cannot be connected to element edges. Such arrangements will result in gaps and penetrations that do not occur in reality.

Linear elements (no mid-side node) should not be connected to the mid-side nodes of quadratic elements, because the edge of the quadratic element deforms quadratically whereas the edges of the linear element deform linearly.



Corner nodes of quadratic elements should not be connected to mid-side nodes. Although both edges deform quadratically, they are not deflecting in sync with each other.

Avoid using linear elements with quadratic elements as the mid side node will open a gap or penetrate the linear element.



None of these is a fatal error. Each will simply cause discontinuities in the results which should not be mistaken as being present in the actual part. These effects will be localized and not propagate through the mesh. You can also use bonded contact to connect non-conforming meshes without causing the problems described above.

#### 4.8.6 Improper constraints

Fixed supports will result in less deformation than simple supports which permit material to move within the plane of support.

#### 4.8.7 Rigid body motion

In static analysis, for a structure to be stressed all rigid body motion must be eliminated. For 2D problems there are two translational (along the X- & Y-axes) and one rotational (about the Z-axis) rigid body motions. For 3D problems there are three translational (along the X-, Y- & Z-axes) and three rotational (about the X-, Y- & Z-axes) rigid body motions.

Rigid body motion can be eliminated by applying constraints such as **fixed support**, **displacement** and **node rotation**.

Frequency and dynamic response do not need to have all their rigid body motions eliminated. However the first few modes would be rigid body modes. For example, if you don't apply any constraints in a frequency problem then the first 6 modes would be for the 6 rigid body motions. The 7th mode onwards would be the structure's deformation modes.

## CAD Models

### 5.1 Introduction

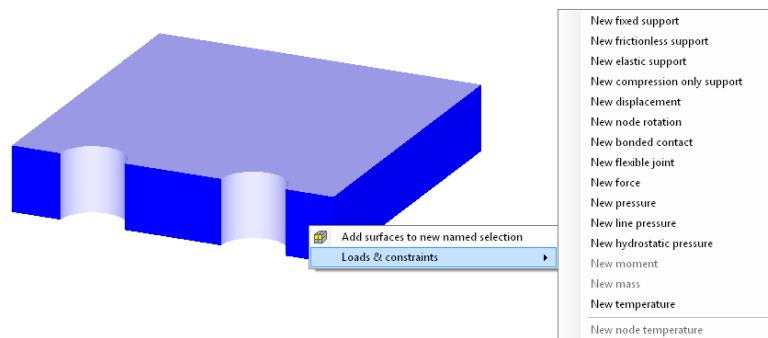
Mecway can open STEP files which can be output by most CAD applications. It displays the parts and can generate a mesh of them (automesh). Links to CAD models appear in the **Geometry** group in the outline tree. Each geometry item in this group should contain a single part so you should split assemblies up into a separate file for each part. There is a tool, **File** → **Split STEP assembly** to do this. If you want to add a CAD file without replacing the currently open model, use **File** → **Import** instead of **Open**.

STL (stereolithography) format files can also be opened and saved by Mecway. An STL file only contains a set of triangles so these are imported as tri3 elements in Mecway without any auto-meshing. Typically, STL files generated by CAD applications contain highly distorted elements so you should use **Automesh 3D** from the **Mesh tools** menu to improve the shape and convert the shells into a solid object.

### 5.2 Loads and Constraints

For imported STEP files, you should apply loads and constraints to surfaces of the geometry rather than the mesh so that they are linked to the geometry and you can regenerate the mesh without losing them. To do this, change to the geometry view by clicking **Geometry** in the outline tree, then right click a surface and choose **Loads & Constraints**.

To select multiple surfaces for a single load or constraint, hold the **Ctrl** key and click the surfaces.

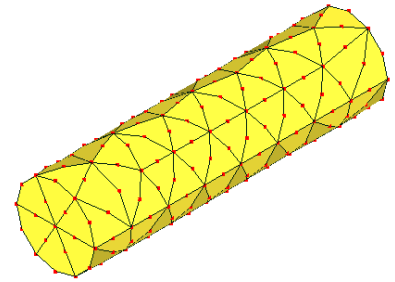
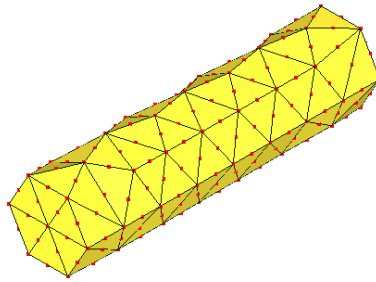


### 5.3 Meshing

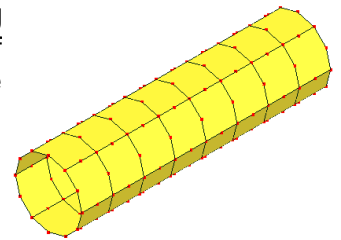
**Meshing parameters** allow local or global refinement by limiting the maximum element size within spherical regions or over the whole geometry. The size gradient of elements can also be controlled. An aggressive size gradient means each element can be much larger or smaller than its immediate

neighbors leading to a low mesh density in large featureless regions and a high density near small details. A gradual size gradient means each element must be a similar size to its immediate neighbors.

The **Fit midside nodes to geometry** option causes the midside nodes of quadratic elements to be located on the surface of the geometry rather than at the midpoints of the corner nodes. This typically improves accuracy but sometimes the midside node will be located too far from the midpoint and lead to a bad element that gives erroneous high stress or fails to solve. To exclude these elements from being fitted to the geometry, turn on **Straighten bad elements**.



To generate a shell mesh, set the **Surface mesh** option under **Meshing parameters**. If the STEP file contains only surfaces, this can work directly. If it contains a solid body, this will produce shell elements in the shape of the solid body's surface.



## 5.4 Gmsh

You can use the open source Gmsh mesher from within Mecway. This is often faster and can mesh some files that the built-in Netgen mesher cannot. Download and install Gmsh, then set the path to it in **Tools** → **Options** → **Gmsh**. You can then use it by selecting the **Gmsh** option in **Meshing parameters**.

## 5.5 Assemblies

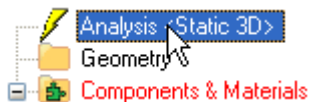
Only a single object is allowed in each CAD file. To model an assembly, import each part as a separate file then use a bonded contact constraint to join their adjacent surfaces. When opening the files with **File** → **Import** or **Geometry** → **Import CAD file**, you can select multiple files at the same time by holding **Ctrl** while selecting them.

## 5.6 Alibre Design

If you have Alibre Design or Alibre Atom3D versions 26-28 or some others installed and licensed, you can open assembly (.AD\_ASM), part (.AD\_PRT), or sheet-metal part (.AD\_SMP) files directly in Mecway. For assemblies, this imports the assembly tree so you can mesh each part or sub-assembly independently or all together. It is only for solid geometry, not surface parts.

To set up Alibre Design support, click **Link to Alibre Design or Atom3D ...** in **Tools** → **Options** → **Alibre**. Choose one of the installed versions shown in the list and click **OK**.

## Analysis Types



The analysis type determines what physical phenomena are modeled. Mecway starts up with Static 3D as the default. Double click **Analysis** or right click it and select **Analysis settings** to switch to another type of analysis such as thermal or frequency.

### 6.1 Static

Static analysis finds the steady state deformation and stress in a structure whose material has a linear stress-strain relationship.

For 3D, it can also model electrostatic fields in insulating dielectric materials. In this case, each node has an electric potential DOF so electric potential should be constrained at at least one node to ensure a unique solution. The solver then uses the electric potential field value to obtain the electric field and flux density.

#### 6.1.1 Static 2D

In Static 2D analysis, all nodes should lie in the XY plane because the Z coordinates are ignored by the solver. Each node has either 2 or 3 DOFs: Nodes of beams have displacement in X, displacement in Y and rotation about Z while nodes of plane, truss and spring elements have only the two displacement DOFs.

#### 6.1.2 Static 3D

In Static 3D analysis, solid, truss and spring elements have 3 DOFs on each node: displacement in X, Y and Z. Shells and beams have 6 DOFs on each node: 3 displacements and also rotation about X, Y and Z. You can combine all the different element types in the same model.

#### Piezoelectric effect

Solid elements can incorporate the piezoelectric effect in static 3D analysis. To use it, you must specify 3 material properties:

- *Elasticity* – either as a 6x6 material stiffness matrix, a 6x6 compliance matrix, 9 orthotropic constants or isotropic Young's modulus and Poisson ratio. These are measured at constant electric field (short circuit).
- *Permittivity* – either a 3x3 matrix, 3 orthotropic constants or 1 isotropic constant. They can be measured at either constant strain or constant stress.
- *Piezoelectric coupling constants* – either a 3x6 stress matrix (e) or 3x6 strain matrix (d).

Matrices with shear terms are in Voigt form, ordered using the popular mapping from tensor indices 11,22,33,23,31,12 to matrix indices 1,2,3,4,5,6 respectively. Caution: Some other software uses different ordering conventions so you may need to swap rows and columns if transcribing data from them.

The piezoelectric effect cannot be used in the same model as thermal stress or cyclic symmetry.

### 6.1.3 Static Axisymmetric

Only plane elements can be used here and they will be treated as axisymmetric elements. The Y axis is the axis of symmetry and the X axis is the radial direction. Each node must lie in the two positive X quadrants of the XY plane and have zero Z coordinates.

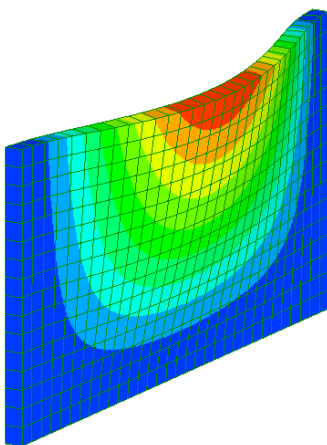
All nodes have two DOFs: displacement in X and displacement in Y.

## 6.2 Nonlinear Static 3D

### 6.2.1 Capabilities

Nonlinear static analysis can model more complex behavior than linear static analysis. It allows displacement to be a nonlinear function of load whereas in linear analysis, the displacement of each node is always in a straight line and proportional to the applied load. This greater capability comes at the expense of a longer solving time. It is only available with the CCX solver. Some examples of where nonlinear analysis is useful are:

**Large displacements and rotations.** For example, a thin strip can be rolled up into a circular shape or members of a mechanical linkage can rotate through various configurations.

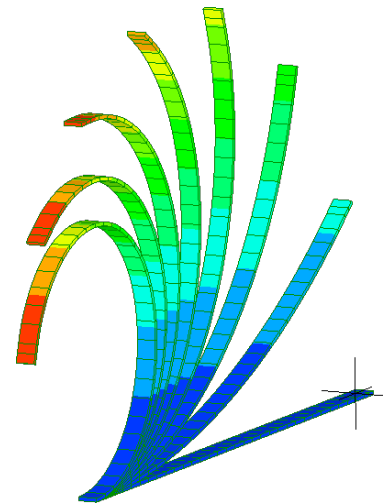


**Membrane action.** An initially flat plate fixed at the edges and bearing a pressure load will deform to a curved shape which becomes stiffer due to membrane stress in the plate.

**Contact between parts.**

**Plastic and hyperelastic materials.**

**Preloaded structures.** For example, a vertical beam hanging under the influence of gravity is in tension. This gives it a greater lateral stiffness than the same beam oriented horizontally.



**Cables.** A cable or rope sagging under the effect of gravity has a bending stiffness which is a function of its tensile force and almost independent of its material properties. This is a similar concept to membrane action. A cable can also have zero or near-zero stiffness in compression as it buckles but a large stiffness in tension. If you use a string of truss elements to model a cable, it must be initially straight and under tension to prevent rigid body motion.

**Buckling.** More general buckling cases are possible than with linear buckling analysis. For example an offset load on an Euler column or an initially bent structure may experience a combination of bending and buckling. A structure may also experience large displacements or rotations before

reaching a configuration in which it becomes unstable and then buckles. Mecway with CCX cannot model the post-buckling response of snap-through buckling.

**Deformation dependent loading (follower loads).** For example, a pressure load applies a constant normal pressure which means the total force changes if surface area or orientation changes.

**Time dependent loading.** Loads can change with time or be turned on and off at different times.

### 6.2.2 Usage

No special settings are needed to use nonlinear analysis in Mecway. Just change the analysis type under **Analysis** → **Edit** and select the **CCX** solver.

If you want time-dependent loads, check the **Quasi-static** box and specify the **Time period** and **Time step** size under **Analysis** → **Edit**. Time here is a pseudo variable which is only used to control the sequence of applied loads. No time dependent effects like inertia or creep are being modeled.

With **Quasi-static** turned off, most loads are automatically ramped up from zero and only the final solution with the full load is shown in the post-processor. With **Quasi-static** turned on, loads that are specified as constant are not ramped and you must explicitly define any required time dependence using formulas or tables.

### 6.2.3 Convergence Failures

When the solver fails to converge on a solution, there are several possible courses of action:

- Ensure there are sufficient constraints to prevent rigid body motion at every time step. Contacts may be initially open so don't rely on them to act as constraints against rigid body motion. For parts that are only supported by contact, add artificial weak springs to support them gently until the contacts close. You can do this using elastic support.
- You can gain insight into how failures occur by turning on **Quasi-static** and defining the loads as formulas that are functions of time ( $t$ ) so that they ramp up gradually over the time period. That will show the converged solution up to the load at which it fails and you may see an unexpected displacement.
- If the model contains many nonlinear features, temporarily remove or suppress some of them to isolate what's causing the problem.
- Turn on **Automatic time stepping** in **Analysis settings**.

## 6.3 Frequency

This was called Modal Vibration 3D in Mecway versions 1-22.

Free vibrations of a structure occur due to its own elastic properties when it is disturbed from its equilibrium state. These vibrations only occur at discrete natural frequencies. The two properties required for vibrational motion are:

- elasticity which returns the disturbed structure back to its equilibrium state, and
- inertia (from the mass) of the structure which makes it overshoot its equilibrium state.

Frequency analysis finds the natural frequencies of a structure and the corresponding deflected shapes (mode shapes). This is done without regard to how the vibration was initiated. All the nodes move with simple harmonic motion in phase with one another at the same frequency. Therefore all the time-dependent displacements reach their maximum magnitudes at the same instant of time.

The magnitudes of all the solution field variables are only relative to the other values in the same mode. Their absolute magnitude has no meaning.

The maximum number of modes is equal to the number of unconstrained DOF in the model. For example, if a model is a single hex8 element with a fixed support applied to one face, the maximum number of modes will be 12, which is the number of DOFs per node (3) multiplied by the number of unconstrained nodes (4). Unless there is shock loading, only the modes of the lowest frequencies are important in the structural response.

Allows 3D elements such as solids, shells, and beams.

#### Stress stiffening

The stress state of a structure can influence its stiffness and vibration modes. This effect, called stress stiffening, is particularly apparent in a tensioned cable or guitar string.

When any loads are present, Mecway first performs a linear static analysis to determine the stress state then proceeds with the frequency analysis using a modified stiffness matrix.

Tensile stress tends to stiffen a structure and increase the natural frequencies. Compressive stress tends to reduce the frequencies. If compressive stress is too high, it may exceed the buckling load of a structure. In these cases Mecway may produce non-physical modes with zero frequencies.

The following features are not allowed with stress stiffening loads:

- Internal solver: Shells using laminate material
- Both solvers: non-zero displacement constraints, non-zero rotation constraints, stress field variable in the solution

## 6.4 Dynamic Response

When a part or structure is subjected to a time varying load, its stresses are amplified by an induced vibration. Dynamic response analysis takes this vibration into account when calculating the stresses & strains. It also calculates the velocities & accelerations in the model's response to the vibrating load.

Default initial conditions are zero displacement and velocity. You can impose an initial acceleration by applying a load at time zero. With the CCX solver, you can specify initial displacements.

The **Time step** affects the accuracy of the solution with smaller time step sizes being more accurate. You can choose a suitable time step size by first performing a frequency analysis to determine the period ( $1/f$ ) of the highest mode of interest, then starting from that value, repeatedly reduce it and solve the problem again until the solution doesn't change significantly. The total **Time period** should be at least the period of the lowest vibration mode. This ensures that all modes oscillate at least once.

Two solvers are available – the default is the Newmark method with constant average acceleration which is unconditionally stable. You can also choose **Mode superposition** which first finds the natural frequencies and mode shapes by solving an eigenvalue problem then generates nodal displacements, rotations and velocities at each time step. There are several other differences of the mode superposition option:

- You must choose the number of modes to include. Typically only the first few modes are important but for shock loading up to 2/3 of them may be needed. Too many modes can slow down the solver and cause it to run out of memory.
- It can be much faster when a large number of time steps are required.
- It is not available in the Axisymmetric analysis type.

- Damper elements cannot be used.
- Any Rayleigh damping must be applied uniformly to the whole model rather than selected elements, and must have a damping ratio less than 1 for every mode that's solved for.
- Rigid body motion is not allowed so the model must be fully constrained.
- Accelerations are not included in the results.

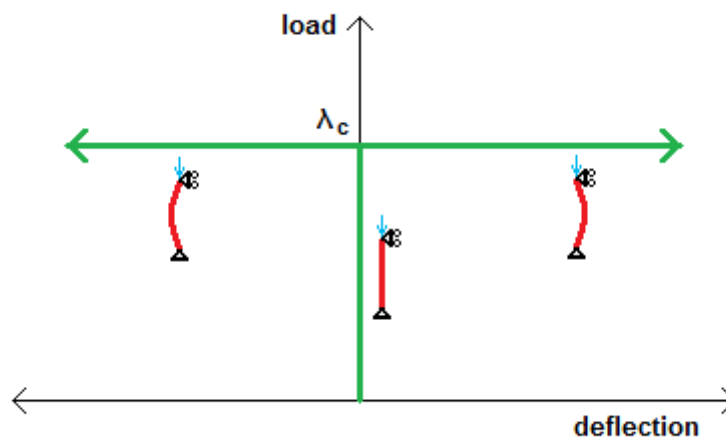
To reduce the number of time steps stored in the results, specify an integer value for **Generate solution at: Every \_\_ time steps**. This reduces memory use and file size.

## 6.5 Nonlinear Dynamic Response 3D

This is similar to Dynamic Response 3D but it allows nonlinear behavior such as large displacements, contact, and plastic and hyperelastic materials. It can only be used with the CCX and OpenRadioss solvers.

## 6.6 Buckling

Linear eigenvalue buckling analysis, such as that provided with Mecway, is only capable of describing bifurcation buckling with a constant, symmetric load-deflection relationship as shown below. An Euler column is used as an example, but the same curve can be applied to other structures. Here, deflection is the displacement perpendicular to the direction of the load. Symmetric means the structure must be equally able to buckle in two opposite directions. There should also be negligible displacement in any direction prior to buckling. If these conditions are not met, you should use nonlinear static analysis instead.



As the load is increased from 0 to the critical load  $\lambda_c$ , the structure remains in its original configuration with no deflection. When the load reaches  $\lambda_c$ , the deflection is indeterminate and increases with no further increase of the load.

Some real structures closely approximate this behavior, while others are so different that eigenvalue buckling analysis is of no use. You should take care to ensure that these assumptions are appropriate to the problem otherwise the buckling factors may be grossly in error even if the mode shapes are reasonable.

An important class of problems for which eigenvalue buckling analysis is usually unsuitable is limit point instability. Here, the structure continuously deflects by a finite amount as load is increased, until a 'limit point' of the load is reached, where it 'snaps through' into a different configuration. An example is a toggle mechanism.

These and other structures which appear to be buckling are in fact general non-linear problems. Another example is a column with an eccentric axial load. The deflection is non-zero for any finite load and there is no bifurcation point.

Eigenvalue buckling analysis assumes no imperfections in the material or loading. For this reason, it is non-conservative and typically overestimates the actual buckling loads.

Each mode has an associated buckling factor. You can think of this as the safety factor. Instability occurs when all the forces are multiplied by the buckling factor. For thermal loads, instability occurs when each node's temperature is

$$T_{cr} = (T_{node} - T_{reference}) \times \text{buckling factor} + T_{reference}.$$

Centrifugal force loads are proportional to the square of the angular velocity. Therefore, the critical angular velocity,  $\omega_{cr}$ , is

$$\omega_{cr} = \omega_{specified} \times \sqrt{(\text{buckling factor})}.$$

It can be convenient to specify unit loads in the model so that the buckling factor is equal to the critical load for forces. If some loads are constant, such as gravity, then you may need to perform several iterations to adjust the unknown loads until the buckling factor becomes 1.

The mode shape represents the relative movement of the nodes immediately after buckling occurs. The actual equilibrium shape of a structure after buckling cannot be found using linear eigenvalue buckling analysis.

You must specify the **Number of modes** and a **Shift point** (internal solver only). The shift point controls the stability of the eigenvalue solver. It must not be zero and should be between zero and the lowest buckling factor. The closer it is to the buckling factors, the greater their accuracy. However, modes with buckling factors below the shift point will not be found.

## 6.7 Thermal

Thermal analysis uses a single temperature DOF for each node. The solver computes heat flux from the temperature field. All thermal analysis types are 3D. However, you can make a 2D model using shell elements of any thickness laid in a plane.

### 6.7.1 Thermal Steady State

This finds the equilibrium temperature distribution in a structure after any transients have dissipated.

### 6.7.2 Thermal Transient

Thermal Transient analysis produces a time history of the temperature field through a structure. You can specify time-dependent loads and temperature constraints as well as an initial temperature distribution. For nodes with no initial temperature specified, Mecway applies a default value of 0 K.

To reduce the number of time steps stored in the results, specify an integer value for **Generate solution at: Every \_\_\_ time steps**. This reduces memory use and file size.

## 6.8 DC Current Flow

This analysis type finds the electric potential (voltage relative to an arbitrary zero) distribution throughout a structure. This electric potential which is the single DOF at each node, is then used to obtain current density, resistive power loss (Joule heating or Ohmic heating) and current in resistor elements.

Temperature dependent conductivity can be used on all types of elements. You must also specify a temperature on every element that use the temperature dependent material. Piecewise constant temperatures be applied with the **Temperature** load or you can obtain a non-uniform temperature field from thermal analysis by first solving the same model as thermal analysis, then right clicking **Loads & Constraints** and choosing **Transfer temperatures from solution**.

To do thermal analysis with Joule heating, first solve it as DC Current Flow, then change to a thermal analysis type and use **Transfer internal heat generation from solution** under **Loads & Constraints**, then solve the thermal model.

## 6.9 Acoustic Resonance

This analysis type can find vibration modes of fluid in an enclosed cavity such as sound resonance in a room or vehicle cabin.

### 6.9.1 Real

If there are no impedance boundary conditions in the model, the modes are solutions to the Helmholtz equation

$$\nabla^2 p + (\omega/c)^2 p = 0$$

where  $p$  is the pressure relative to ambient,  $\omega$  is the angular frequency of the mode, and  $c$  the speed of sound in the medium.

Continuum elements represent the medium inside the cavity. Boundary faces without any constraints behave like hard surfaces with the following Neumann boundary condition

$$\nabla p \cdot \underline{n} = 0$$

where  $\underline{n}$  is the surface normal vector.

The mode shapes (pressure) shown in the solution are the acoustic pressure of the standing waves which will oscillate sinusoidally with time. Their amplitudes are arbitrary.

### 6.9.2 Complex

If there are impedance boundary conditions in the model, Mecway solves a complex quadratic eigenvalue problem and generates frequency, decay rate, and the complex amplitude of pressure ( $P$ ) for each mode. Scaling of  $P$  is arbitrary between modes.

Instantaneous acoustic pressure	$p(t) = \Re(P e^{(i\omega - \tau)t})$
Complex amplitude	$P$
Frequency	$\omega/(2\pi)$
Decay rate	$\tau$

The modes it finds are not, in general, those with the smallest (real) frequencies but with the smallest modulus of  $(i\omega - \tau)$ . These are sometimes equivalent but modes with a high decay rate may be omitted if you don't request enough. It also randomly omits or duplicates 0 Hz rigid body modes due to numerical error.

Bonded contact and constraint equations cannot be used with complex acoustic analysis.

The quadratic eigenvalue problem it solves is:

$$(K + \lambda C + \lambda^2 M)\phi = 0$$

K is the stiffness matrix  $K = \int_V \nabla N^T \nabla N dV$

C is the damping matrix  $C = \int_A \frac{\rho}{Z} N N^T dA$

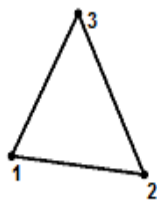
M is the mass matrix  $M = \int_V \frac{1}{c^2} N N^T dV$

$$\lambda = i\omega - \tau$$

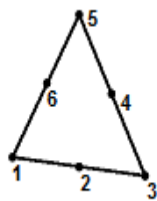
$\phi$  is a vector of P at each node.

## Elements

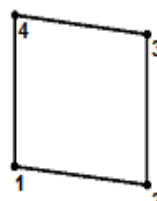
### 7.1 Plane Continuum Elements



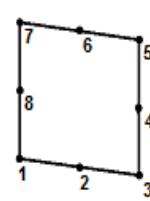
*Tri3*



*Tri6*



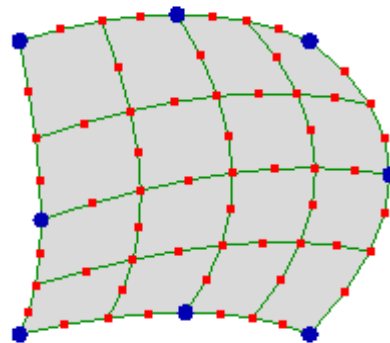
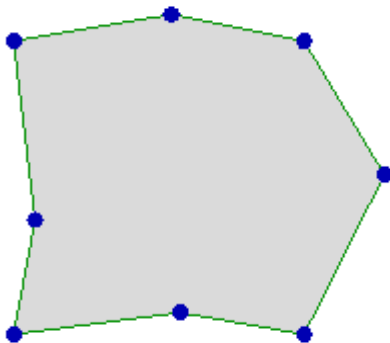
*Quad4*



*Quad8*

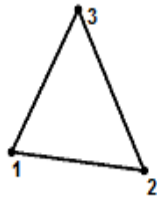
Plane elements can be used for various 2D analysis types to represent structures such as flat plates or prismatic rods. Their nodes should lie in the XY plane.

The quadratic elements (those with midside nodes) can have parabolically curved sides although they are displayed as being straight. You can see the curved shape by refining the element as shown below.

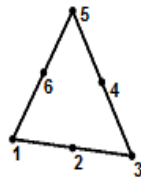


Quadratic elements typically perform better than the linear elements because the DOF field value can vary quadratically across the element whereas the linear elements only allow a linear variation. In mechanical analysis types, the linear elements, especially tri3 (constant strain triangle), have a further limitation of being too stiff in bending. To model bending accurately with linear elements you must refine the mesh so that each individual element experiences mostly tension or compression and less bending.

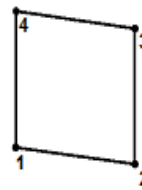
## 7.2 Axisymmetric Continuum Elements



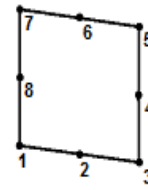
*Tri3*



*Tri6*

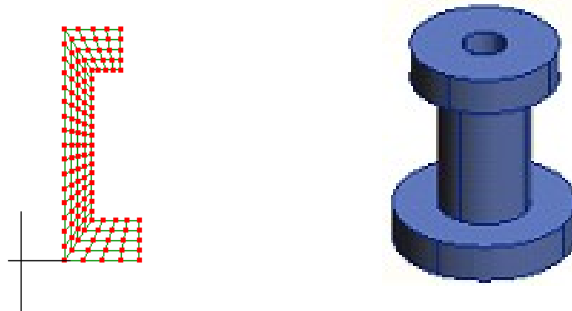


*Quad4*



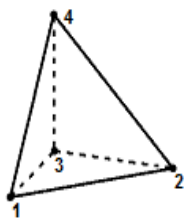
*Quad8*

Although these appear to be the same as plane continuum elements, each axisymmetric element actually represents an entire circular solid as shown below. The axis of symmetry is the Y-axis and elements must be located in the X-Y plane with all X-coordinates non-negative. Axisymmetric elements can only be used in the axisymmetric analysis types.

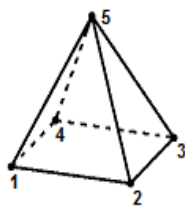


In mechanical analysis types, each node has 2 DOFs: displacement in X(radial) and displacement in Y(axial). Any nodes located at X=0 must be restrained to not be displaced in the X direction because it is physically unreasonable that the material should overlap itself or for a hole to appear. Also, rigid body motion can only occur by translation along the Y-axis, so only translational motion in the Y direction needs to be constrained to prevent rigid body motion.

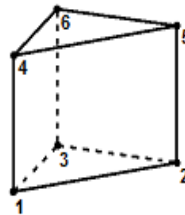
## 7.3 Solid Continuum Elements



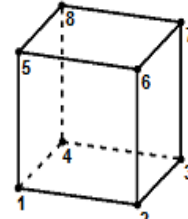
*Tet4*



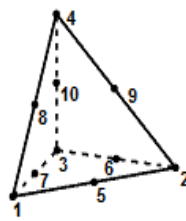
*Pyr5*



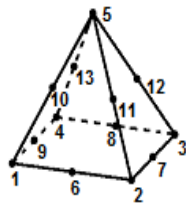
*Wedge6*



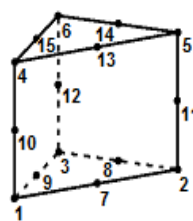
*Hex8*



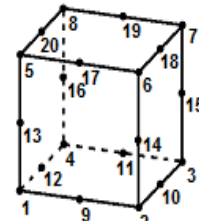
Tet10



Pyr13



Wedge15

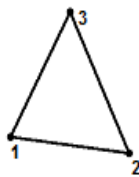


Hex20

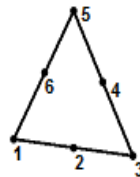
Solid elements are the most general and, in principle, can be used to model any shaped structure. However some geometries such as thin beams or plates can require such a large number of solid elements that the solver runs out of memory or takes too much time. In these cases you can further idealize the model by using shells, beams, fins, resistors, etc. instead of solid elements.

Hex20 is typically the most accurate. However, hex8 also performs well with mechanical analysis types. It uses an incompatible mode formulation which is free from shear-locking and thus has improved accuracy in bending compared to a standard 8-node hex element. It is described in *Three-Dimensional Static and Dynamic Analysis of Structures A Physical Approach With Emphasis On Earthquake Engineering* by Edward L. Wilson.

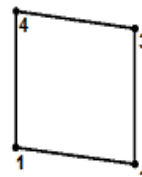
## 7.4 Shell



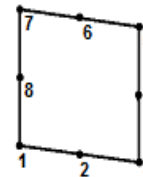
Tri3



Tri6



Quad4



Quad8

Shells are 3D elements that can model stress, heat flow or electric current in the plane of the element but not through the thickness. They are useful for thin parts where solid elements are too computationally expensive or in place of 2D elements where those are not available. Typical structures modeled with shell elements include sheet metal brackets and cabinets, thin metal platforms, pressure vessels, and body parts of motor vehicles.

### 7.4.1 Formulation

In the mechanical analysis types, shell elements are as described in the following papers:

- Tri3: MITC3 (doi:10.1016/j.compstruc.2004.02.004)
- Tri6: MITC6-a (doi:10.1016/j.compstruc.2004.02.004)
- Quad4: MITC4 (doi:10.1108/eb023562)
- Quad8: MITC8 (doi:10.1002/nme.1620220312)

The quadrilateral elements (quad4 and quad8) are the most accurate and triangular elements should be restricted to areas that can't be meshed with quadrilaterals. All four elements avoid shear locking, and the quadratic elements (tri6 and quad8) also avoid membrane locking.

Each node has 6 DOFs – displacement in X, Y and Z and rotation about X, Y and Z. However there is no drilling DOF which means each node is free to rotate about the shell's normal and is only resisted by an arbitrary small stiffness to ensure numerical stability.

The rotational DOFs of shells mean that you can join two shell elements together at a common edge and they will transmit bending moments between each other. This is distinct from solid elements which cannot transmit bending moments when they're only connected by an edge. If the straight edge of a shell is joined to the edge of a solid element, then it will form a hinge joint. To make a stiff joint, use bonded contact.

Shells use consistent mass matrices with fictitious near-zero rotational inertia at the nodes.

### 7.4.2 Orientation of Local Axes

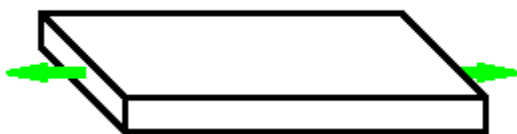
Each element has a local right-handed coordinate system, U,V,W, which is used for orienting orthotropic and laminate materials and for displaying some stress results. The W axis is normal to the element's surface at each point. You can define the direction of the U axis by specifying a vector in global X,Y,Z coordinates. The direction of the U axis is the projection of this vector onto the plane normal to W. If not specified, the default direction of this vector is parallel to the edge from nodes 1 to 2, or 1 to 3 for linear or quadratic elements respectively. The V axis is mutually perpendicular to U and W.

You can specify orientation in two places - **Element properties** and **Loads & Constraints** → **Element orientation**. Specifying orientation through **Loads & Constraints** allows you to enter a formula to define orientation as a function of element center position, and is applied to a selection of elements. Orientation specified through **Element properties** applies to individual elements and is usually not as convenient unless you are importing data with a different orientation for each element.

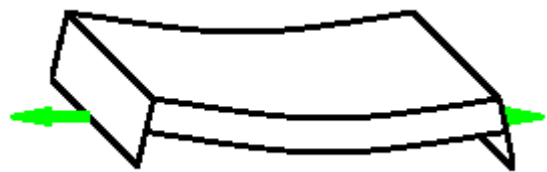
With the CCX solver, element orientations are not available in the solution if any laminate shell elements are present.

### 7.4.3 Shell Offset

By default, the nodes of a shell element are located on its midsurface. However, in static analysis, you can specify a **Shell offset** in **Element properties** to locate the midsurface away from the node locations. This number is a distance the shell is moved in the +W direction in its local coordinate system. Loads are still transmitted to the element at the node locations so extensional forces can result in bending and vice versa.

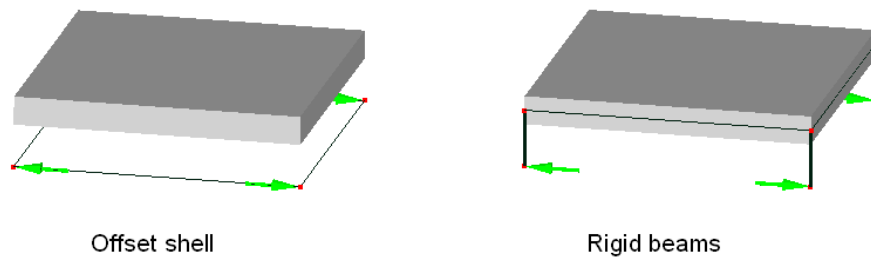


Shell offset = 0



Shell offset > 0

An offset shell behaves the same as if there were rigid beams connected between the nodes and the shell located at the midsurface.



Offsetting a curved shell changes its shape and size because the +W direction varies across the surface of the element. This affects its stiffness and loads that depend on surface area or volume.

With the internal solver, **Displacement**, **Cyclic symmetry**, **Contact** with type **Constraint equations**, and **Node-surface coupling** without **Couple surface node rotational DOFs** can not be applied to offset shell elements or faces. Use their nodes instead, where possible.

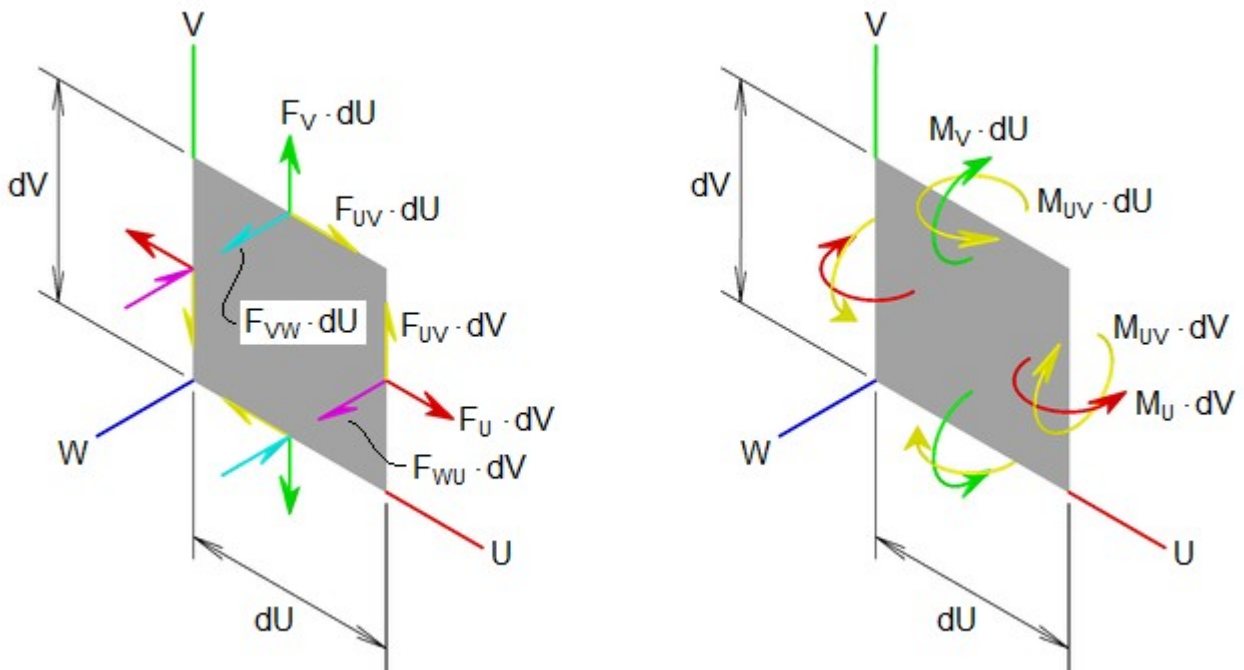
#### 7.4.4 Curved Elements

Shell elements can be curved out of their plane but the radius of curvature should be large compared to the thickness. Only the quadratic elements (tri6 and quad8) can represent curvature well because they have sufficient nodes along each edge to define a quadratic shape.

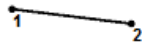
#### 7.4.5 Results

Mecway reports stress and strain at the top, midsurface and bottom of each element or laminate layer. The top surface is the one with a more positive W coordinate. If you select the stress group's node in the outline tree, the display shows all three values simultaneously at their respective locations for non-laminates or just the two outer surfaces for laminates. Select the top, midplane or bottom branch to show only one.

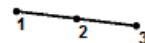
**Moment per Length** and **force per length** are moment and force resultants respectively. They are with respect to the local coordinate system of the element and defined as shown below on differential elements within a shell element.



## 7.5 Beam



Line2



Line3 (CCX)

Mecway's beam elements use Euler beam theory which means they can model bending of slender, uniform, isotropic beams. They also have longitudinal and torsional stiffness allowing them to act as columns or shafts. Each node has both displacement and rotational DOFs so they can transmit bending and twisting moments between elements.

You can apply loads and constraints to the end faces or the side face.

Predefined cross-sectional shapes or an arbitrary shape (**General section**) can be used depending on the analysis type and solver:

#### Internal solver

static 3D frequency buckling dynamic response 3D	I, rectangular bar, rectangular tube, circular bar, circular tube, general section
static 2D	C, T, I, rectangular bar, rectangular tube, circular bar, circular tube, general section
dynamic response 2D	general section

#### CCX solver

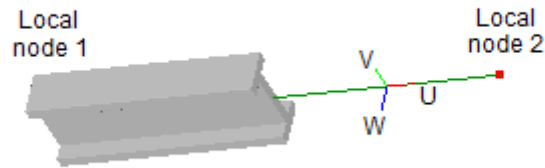
static 3D nonlinear static 3D frequency dynamic response 3D nonlinear dynamic response 3D buckling	rectangular bar, rectangular tube, circular bar, circular tube
---	--

#### OpenRadioss solver

nonlinear dynamic response 3D	I, rectangular bar, rectangular tube, circular bar, circular tube, general section
-------------------------------	---

The orientation of the beam cannot be fully defined by the locations of the nodes alone so there are other options to control how the cross section is rotated about the beam's longitudinal axis. You can specify these in two places - **Element properties** and **Loads & Constraints** → **Element orientation**. Specifying orientation through **Loads & Constraints** allows you to enter a formula to define orientation as a function of element center position, and is applied to a selection of elements. Orientation specified through **Element properties** applies to individual elements and is usually not as convenient unless you are importing data with a different orientation for each element.

In 3D, the element's local coordinate system, U, V, W defines the orientation of the cross-section.



The U axis is parallel to the element's length, the V axis is perpendicular to U and the W axis is mutually perpendicular to both U and V according to  $W=U \times V$ . These constraints leave the V axis free to rotate about the element's longitudinal axis. There are two options for defining the direction of the V axis:

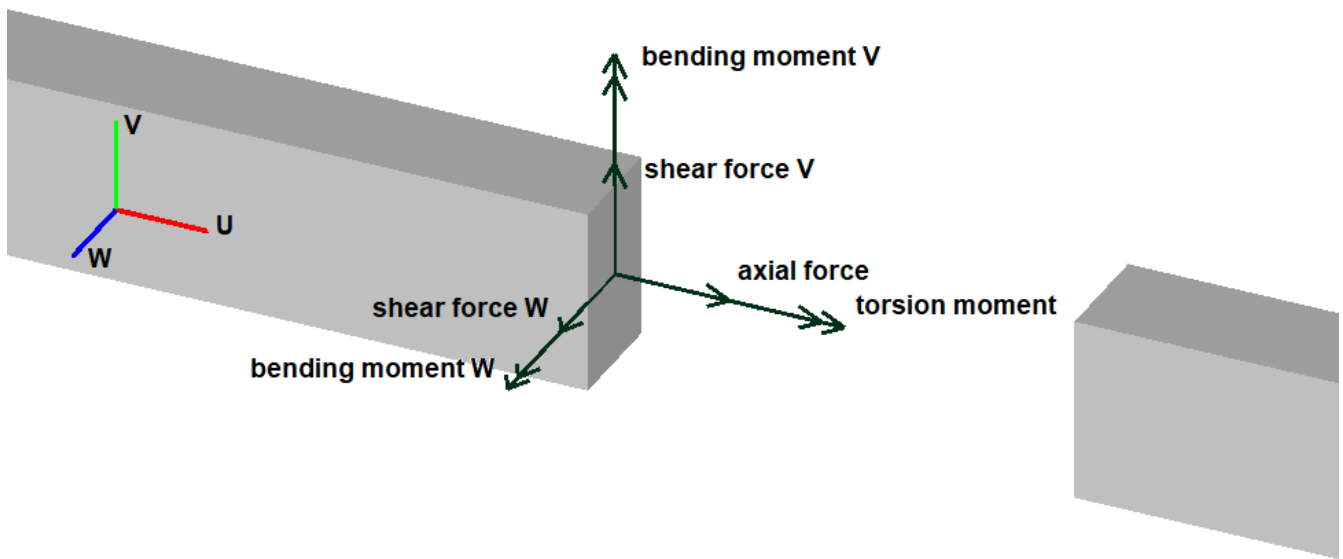
- **Default / none.** The V axis is perpendicular to the global Z axis and parallel to the XY plane. If the element is nearly parallel to the Z axis (gradient < 0.01%) then the V axis is made parallel to the Y axis too.
- **Direction of element V axis.** The V axis can be defined by a vector in the global X, Y, Z coordinate system. The specified vector might not meet the requirement of being perpendicular to U, so the V axis is the projection of this vector onto a plane perpendicular to U. This allows almost any vector to be used as long as it is not zero (0, 0, 0) and not be parallel to U. The magnitude of the vector is not used by Mecway.

After defining the vector you can also apply a further rotation of V about U by a specified angle (**Rotate by angle**). This feature is useful if you know the angle in which you want to orient the beam but don't want to manually convert it to vector form. It can also be helpful if you want to specify a common direction of W for many elements in different orientations. You can set the vector to the direction of W then rotate it by  $-90^\circ$ .

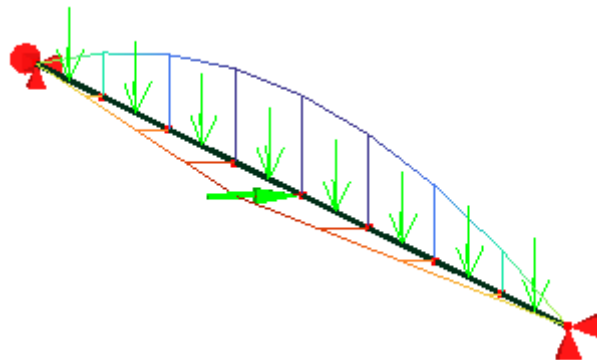
In 2D, the U, V, W element coordinate system is fully defined by the locations of the element's nodes, with W parallel to the global Z axis and in the same direction. You can flip the section upside-down by reversing the element's nodes with **Mesh tools** → **Invert**.

Longitudinal stresses (**stress in element coordinates**) are evaluated at each of 4 predefined points in the cross-section and one user-defined **Stress recovery point in principal coordinates**. These stress values are calculated only from the normal force and bending moments. They do not incorporate shear stress caused by torsion or shear forces.

The solution's internal forces and moments follow the sign convention shown in the diagram below.



For 3D beams, you can display these values as shear force diagrams or bending moment diagrams overlaid on the mesh by clicking the **shear force** or **bending moment** item in the outline tree. These diagrams use element values if available (i.e. internal solver) or node values otherwise (i.e. CCX solver). With CCX, their orientation may be incorrect if there are any laminate shell elements present.



With the CCX solver, beams appear as having a rectangular section in the solution regardless of the actual shape. Stress values are only valid at nodes which lie on the actual section. For circular sections, that is only the midside nodes of the end faces.

The CCX solver (2.21) has some bugs with beam elements. Non-zero displacement constraints on beams can cause incorrect stress, and constrained rotational DOFs can cause incorrect external force. Mecway will show a warning in the outline tree if the conditions for these bugs are met.

### 7.5.1 Beam offset

With **Static 3D** and the internal solver, a beam can be offset relative to its node positions using **Element offset (beams)** in **Loads & Constraints**.

**Displacement**, **Cyclic symmetry**, **Contact** with type **Constraint equations**, and **Node-surface coupling** without **Couple surface node rotational DOFs** can not be applied to offset beam elements or faces. Use their nodes instead, where possible.

## 7.6 Truss



*Line2*

A truss element is a beam hinged at both ends so that it can only resist axial compressive or tensile forces and has no bending or torsional stiffness. Each node has translational degrees of freedom in the X, Y and, if 3D, Z directions.

Truss elements can be used to model truss structures where no bending occurs in individual members, as well as to transmit forces between objects and create hinge joints between beam or shell elements.

To convert a beam element into a truss element, select the desired elements, right click on one of them, click **Element properties** and check **Truss**.

Only 3D models can use all the section shapes (C, T, L, I, rectangular bar, rectangular tube, circular bar, circular tube and general section) for trusses. Static 2D disallows L section, and in other analysis types only the general section can be used. No orientation needs to be specified like for beam elements.

## 7.7 Spring



*Line2*

The spring element has stiffness only in the direction of its length and is hinged at its endpoints. It can be used anywhere that a truss element can. Each node has 2 or 3 translational degrees of freedom along the X, Y and Z axes depending on the number of dimensions (2D or 3D).

You can define a spring by applying the **Spring** mechanical material type to a line2 element.

The **Spring constant**  $k$  is the stiffness in Hooke's law

$$F = -k u$$

where  $F$  is the extending force exerted by the spring and  $u$  is the increase in length from the free length.

A spring element behaves identically to a truss element with zero density. The equivalent truss element satisfies the equation

cross-sectional area x Young's modulus = spring constant x length

## 7.8 Damper



*Line2*

The damper element has damping only in the direction of its length and is hinged at its endpoints. Each node has 2 or 3 translational degrees of freedom along the X, Y and Z axes depending on the number of dimensions (2D or 3D).

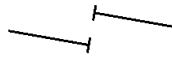
You can define a damper by applying the **Damper** mechanical material type to a line2 element.

**Viscous damping coefficient** ( $c$ ) causes a damping force of

$$F = -c \dot{u}$$

where  $\dot{u}$  is the relative axial velocity of the two nodes.

## 7.9 Gap



*Line2*

The gap element is a unilateral contact between two nodes. To define a gap element, apply the **Gap** mechanical material type to a line2 element.

It only transmits force parallel to the gap direction which is the element's U axis. The default U is the direction from the element's node 1 to node 2 but you can specify an arbitrary direction with **element orientation** instead. In either case, the gap direction doesn't rotate with large displacements.

Specifying the gap direction with **element orientation** allows:

- Coincident nodes.
- A lateral offset (normal to U) between node positions. This offset doesn't affect its behavior but should be small because the gap element doesn't transfer moments.
- Tension-only behavior by specifying a gap direction opposite to the direction from the element's node 1 to node 2.

### 7.9.1 Internal solver

It can only be used in the **Static 3D** analysis type where it causes an iterative solution.

The stiffness has the constant value **Stiffness of closed gap** when it is closed and zero when it is open. To approximate hard contact, **Stiffness of closed gap** should typically be a large value like  $1e12$  N/m.

The Axial force solution variable is the tensile force across the gap. It can not be positive because the gap has no stiffness in tension.

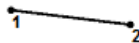
### 7.9.2 CCX

It can only be used in the **Nonlinear Static 3D** and **Nonlinear Dynamic Response 3D** analysis types.

The stiffness is a smooth function that approaches a constant stiffness as penetration increases and constant force as the clearance increases. You must specify these as the two parameters **Stiffness of closed gap** and **Force of open gap**. To approximate hard contact, **Stiffness of closed gap** should be large like  $1e12$  N/m and **Force of open gap** should be small like  $1e-3$  N.

You can also define an **Initial clearance** or a negative value for initial overclosure.

## 7.10 Tension Only



*Line2*

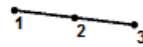
The tension only element is a more limited version of the **Gap** element. Its differences are: The gap direction is always from the element's node 2 to node 1 (opposite to Gap element's default); the nodes cannot be coincident; axial force for a closed gap is positive instead of negative; stiffness of the closed gap is defined by Young's modulus, cross-sectional area and length; axial stress is available in the solution; and it can only be used with the internal solver.

To define a tension only element, apply the **Tension only** mechanical material type to a line2 element and choose one of the section shapes in the geometric material types. You must also identify the element as a truss. To do this, select the desired elements, right click on one of them, click **Element properties** and check **Truss**.

## 7.11 Fin



*Line2*

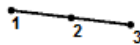


*Line3*

The rate of heat removed by convection from a surface is increased by increasing the surface area for heat transfer using extended surfaces called fins. A simple example is a metal spoon placed in a glass of hot water, heat is conducted through the spoon to cause the handle to become warmer than the surrounding air. Heat is then transferred from the spoon handle to the air by convection. Similarly, industrial applications use fins in car radiators, double pipe heat exchangers, electronic equipment and compressors.

A fin element models heat conduction along its length but not across its width. It can also have convection and radiation heat transfer applied to its side face and/or end faces.

## 7.12 Fluid Network Element



*Line3*

This is a special element for use with forced convection and the CCX solver. It represents flowing fluid which transfers heat downstream along the network as well as by convection with neighboring solid or shell elements. It should not share nodes with any other type of element.

You must specify the fluid's specific heat at constant pressure ( $c_p$ ) as the material property, and its mass flow rate as a constraint on every fluid network element. You must also specify the inlet temperature of the fluid as a temperature constraint on the most upstream node(s) in the network.

The position of the midside node doesn't matter, and the positions of the end nodes are only relevant for determining which faces they are associated with in forced convection.

## 7.13 Resistor



### *Line2*

A resistor element has a linear change in potential over its length according to Ohm's law  $V=IR$ . Its resistance can be defined by either using the material's **Resistor** property **Resistance**, or by its conductivity together with the geometry of a beam. Any of the uniform section shapes (C, T, L, I, rectangular bar, rectangular tube, circular bar, circular tube, general section) can be used. Resistors can be useful for modeling linear electric circuits as well as for measuring current values in 3D continuum or shell models. To find the current flowing through a complex object, you can place a resistor element with near-zero resistance in series with the object. The solution will then show the current flowing in the resistor.

## Materials

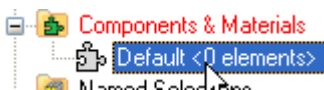
Elements are designed to mathematically represent various types of materials for both their physical properties and their geometric shapes.

### 8.1 Library

Mecway does not include a materials database but you can build one yourself. Create an empty .liml file with only a material defined in it, then when you need that material, open it using **Import** from the **File** menu. This will bring those materials into the currently open model.

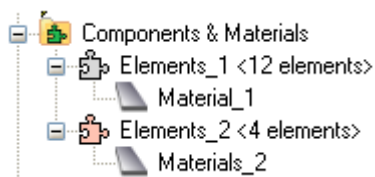
As a convenience for managing such files, you can define a common location in **File** → **Import from library** → **Set library location...** and any .liml files put there will appear on that sub-menu as well as under the **Assign material from library** menu when right clicking on a component. These library files may contain other data such as meshes of common parts or solution formulas. A shortcut to save an individual material to a new library file is the **Save to library...** button in the **Material Properties** window.

### 8.2 Defining a New Material



Right click a component in the outline tree to **Assign new material**

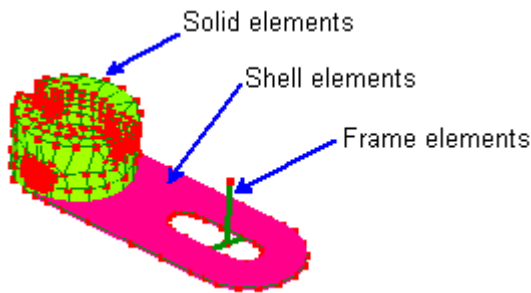
### 8.3 Mixed Materials



In order to have a more than one different material in a model, the elements with different materials need to be in different components.

### 8.4 Mixed Elements

Element types can be mixed in the same model, but each should have its own material definition because their geometrical properties are not the same.




## 8.5 Anisotropic Materials

Orthotropic materials have mechanical properties that differ along directions which are perpendicular with each other. In Mecway you can use orthotropic materials for 3D solid, 2D membrane and shell elements. On shell elements, one axis (W) is always in the through thickness direction (normal to the surface).

There are two conventions for defining the Poisson's ratios of orthotropic materials. Mecway defines  $\nu_{xy}$  to be the negative ratio of strain in the y direction to strain in the x direction due to stress in the x direction. This is sometimes called a major Poisson's ratio. Some older sources of data transpose the subscripts (minor Poisson's ratio). You can check that Mecway's convention is being used by ensuring that this equation is satisfied:

$$\frac{\nu_{xy}}{\nu_{yx}} = \frac{E_x}{E_y}$$

The directions of the orthotropic axes are the local element axes (U, V, W). For solid elements, these are the same as the global X, Y, Z axes by default. You can change this for each element by defining the local U and V axes in either **Element properties** or **Loads & Constraints** → **Element orientation**. The direction you specify for the U axis is the same direction that the element will use. You must also specify the V axis to fully define the orientation in 3D space. It's often not easy to choose a V axis which is perfectly orthogonal to the U axis so Mecway will obtain the orthogonal V axis by projecting the vector you specify onto the plane normal to the U axis.

The local axes of all elements can be displayed graphically using this tool-button . With red, green and blue lines representing the U, V, and W axes respectively.

Anisotropic materials have mechanical properties that differ along arbitrary directions which are not perpendicular with each other. You can use these with solid elements in the Static 3D analysis type by specifying either a material stiffness matrix or compliance matrix (inverse of material stiffness matrix).

## 8.6 Temperature Dependent Properties

You can specify temperature dependent thermal conductivities and specific heat values. The solver linearly interpolates between the specified data points. If a temperature is outside of the range, the solver uses the nearest data point rather than attempting to extrapolate.

## 8.7 Failure Criteria

If you specify a failure criterion for a material with the static or dynamic response analysis types then the solution will include a factor of safety field variable.

The **von Mises** failure criterion can only be used on 2D continuum, axisymmetric, 3D shell, and 3D solid elements.

The **Tsai-Wu** failure criterion can only be used with 3D shell elements and it assumes plane stress, ignoring out-of-plane shear stress. It requires 6 parameters – tensile strength in two directions, compressive strength in two directions, shear strength, and an interaction coefficient  $F^*_{12}$  between -1 and 1 inclusive. For isotropic and orthotropic materials, the factor of safety is evaluated at both sides of the shell (top and bottom) and only the lowest of these two factors of safety is reported for each node. For laminates, the factor of safety is evaluated at both sides of each layer and the minimum factor of safety at each node is reported (first-ply failure) together with the number of the layer (**First Layer to Fail**) and the position within the layer that it occurs on (**Failure Position in Layer**). A position of 1 means the top side (+W direction) and -1 means the bottom side (-W direction) and these are the only two positions where it is evaluated.

The **Factor of Safety** (strength ratio) reported in the solution is the value which all stress components can be multiplied by to cause the stress state to lie on the failure surface. For example, a value between 0 and 1 indicates failure and a value of 2 indicates that the structure can sustain twice the stress before failing. The factor of safety is also the value which all loads can be multiplied by to cause failure. The scale is limited to a maximum of 15 by default but you can change that in **Tools** → **Options** → **Contour plot** → **Factor of safety scale maximum**.

## 8.8 Nonlinear Materials

Several non-linear materials become available with the Nonlinear Static 3D and Nonlinear Dynamic Response 3D analysis types. They are:

- Neo Hooke

- Mooney-Rivlin

- Plastic with isotropic hardening

- Plastic with kinematic hardening

- Ramberg-Osgood

# Loads and Constraints

To apply a load or constraint, right click the **Loads & constraints** branch of the outline tree and choose what to add. Most loads and constraints need to be applied to nodes, faces, elements or named selections. Some are restricted to only nodes, only faces or other combinations. If you select some of these items before you create the new load or constraint then it will automatically use them by default.

Where loads or constraints are distributed over surfaces, apply them to faces rather than nodes. Mesh refinements will automatically transfer face loads and constraints to the newly created elements, whereas loads and constraints applied to nodes remain at the same point locations.

To temporarily disable a load or constraint so that it is ignored by the solver and not displayed in the graphics area, right click it and choose **Suppress**. It's name will then be grayed out until you **Unsuppress**.

Some loads and constraints can vary as a function of time. There are two ways to do this - with a table or a formula. Using a **Table**, you specify a table of time and value pairs which can be entered directly or pasted from a spreadsheet. The solver performs a linear interpolation between data points in the table to determine the value at each time step. If a time step is outside the range of times in the table then constant extrapolation is performed so that the final value will be applied at all subsequent times and the first value will be applied at all preceding times. You can also enter a **Formula** with the variable  $t$  to represent time and it will be evaluated at each time step.

For the CCX and OpenRadioss solvers, formulas are converted to piecewise linear functions. It samples each function at 10001 uniformly spaced time values and retains only those points which define a piecewise linear function whose error is no more than 0.01% of the range of the function. Determination of the error and range are made using the 10001 samples.

You can assign a name to each load or constraint. They are sorted by name in the outline tree and a name allows them to be accessed by scripts using the Python API. To set the name, right click the item in the outline tree and choose **Rename**.

You can organize loads and constraints into groups. Groups can be suppressed or collapsed to make working with large numbers of loads and constraints easier. To create a group and place a load or constraint in it, right click it and choose **Move to group** → **New group**.

## 9.1 Fixed support

Analysis types: Static, Nonlinear Static, Frequency, Dynamic Response, Nonlinear Dynamic Response, Buckling

Solvers: Internal, CCX, OpenRadioss

A fixed support fully constrains the nodes, faces or elements against displacement and rotation in any direction. It can be applied to nodes that have all rotational DOFs (nodes of beam elements and reference nodes of node-surface couplings), or any elements or faces except the end faces of truss, spring, and damper elements. Only those DOFs supported by each face or node will be constrained so a fixed support on the face of a solid element won't constrain the rotational DOF of an adjacent shell element.

## 9.2 Frictionless support

Analysis types: Static 3D, Nonlinear Static, Frequency, Dynamic Response 3D, Nonlinear Dynamic Response, Buckling

Solvers: Internal, CCX, OpenRadioss

A frictionless support constrains displacement in the direction normal to the faces it is applied to. This allows the nodes to slide along the surface but not lift off or penetrate it.

On curved surfaces, the normal direction at each node is the average of the normals of all adjacent faces which the frictionless support is applied to.

On shells and the end faces of beams, frictionless support also constrains node rotation about any of the axes in the plane of the surface.

On the edge faces of beam and truss elements, frictionless support constrains the elements as if they were able to slide inside a tube. They can slide along and rotate about their longitudinal axes, but not move or bend laterally.

In linear static analysis, you can specify a **Normal displacement** which is the distance the surface is displaced. A positive value means displacement in the direction into the surface, as indicated by the red arrowhead constraint symbols.

## 9.3 Pinned support

Analysis types: Static 3D, Nonlinear Static, Frequency, Dynamic Response 3D, Nonlinear Dynamic Response, Buckling

Solvers: Internal, CCX

A pinned support constrains mean displacement to zero in all three directions. This allows the nodes or faces to rotate freely about their centroid.

Faces or multiple nodes with a pinned support cannot have other constraints in some cases.

It can only be applied to faces of solid elements or nodes of any element type.

## 9.4 Elastic support

Analysis types: Static 3D, Nonlinear Static, Frequency, Dynamic Response 3D, Nonlinear Dynamic Response, Buckling

Solvers: Internal, CCX

An elastic support constrains each face through a finite stiffness distributed over the surface. You can independently specify normal and tangential stiffness per unit area. With only a normal stiffness, this is also called a Winkler foundation.

It can only be applied to faces with an area defined by their nodes – that is faces of solid elements and the front and back of shell elements.

## 9.5 Compression only support

Analysis types: Static 3D, Nonlinear Static

Solvers: Internal, CCX

A compression only support behaves similar to a frictionless support at nodes where the support is in compression, and applies no constraint to nodes that would put the support in tension. This is a non-linear feature so the solver will perform several iterations to determine which nodes should be constrained and which are free to displace.

It can only be applied to 2D faces, this includes faces of solid elements and the surfaces of shells.

With the internal solver, convergence is achieved when no constraints change state between successive iterations or when the L2 norm of the change in displacement at all nodes is small compared to the L2 norm of the displacement. The default tolerance is  $10^{-11}$  and you change it in

### **Analysis settings.**

With the CCX solver, the support is provided through nonlinear springs and you must specify the stiffness per unit area in compression (higher is more correct) as well as a nonzero maximum stress in tension (lower is more correct).

Limitations of compression only support with the CCX solver are:

- The solver may fail to converge, especially with quadratic elements or high stiffness per unit area.
- External forces are not available on any of its nodes that are also shared shell, beam or truss elements.
- Not suitable for large rotations of curved surfaces because the spring directions don't rotate with deformation.
- Requires Nonlinear Static 3D analysis type.

## 9.6 Displacement

Analysis types: Static, Nonlinear Static, Frequency, Dynamic Response, Nonlinear Dynamic Response, Buckling

Solvers: Internal, CCX, OpenRadioss

A displacement constraint or initial condition sets the component of the displacement vector in specified direction on every node it is applied to. The components perpendicular to that direction remain free so 3 displacement constraints are required to fully constrain displacement in 3D. When used as an initial condition, the free perpendicular components default to zero so only one displacement initial condition is required.

A displacement can be applied to nodes, faces or elements.

To constrain displacement in an arbitrary direction, specify the X, Y and Z components of a vector describing the direction. Each node can have any number of constraints in any directions. Mecway will internally reduce them to a minimum equivalent set of orthogonal constraints in a rotated coordinate system.

The direction vector's components can be functions of position. This allows, for example, radial or tangential displacement in a cylindrical coordinate system. With nonlinear analysis types, the position is the initial modeled position of each node, even if they move during the solution.

Displacement must be zero when using the internal solver with any analysis type except Static. It can be non-zero and a function of time with the CCX and OpenRadioss solvers.

## 9.7 Node rotation

Analysis types: Static 2D, Static 3D, Nonlinear Static, Dynamic Response 2D, Dynamic Response 3D, Nonlinear Dynamic Response, Frequency, Buckling

Solvers: Internal, CCX, OpenRadioss

Node rotation fixes the rotation angle of each of it's nodes about the specified axis. You can only use it on nodes or faces of elements having the corresponding rotational DOFs. This includes beam and shell elements but not solid, plane continuum, truss, spring or damper elements. If you need to constrain rotation of these types of elements without also constraining displacement, use **Node-surface coupling** and apply node rotation constraints to the reference node.

To constrain rotation about an arbitrary axis, specify the X, Y and Z components of a vector describing the direction of the axis. Each node can have any number of constraints about any axes. Mecway will internally reduce them to a minimum equivalent set of orthogonal constraints in a rotated coordinate system.

Node rotation angle must be zero when using the internal solver with any analysis type except Static.

## 9.8 Velocity

Analysis types: Nonlinear Dynamic Response

Solvers: CCX, OpenRadioss

With the CCX solver, Velocity can only be an initial condition.

A velocity constraint or initial condition sets the component of the velocity vector in specified direction on every node it is applied to. The components perpendicular to that direction remain free so 3 velocity constraints are required to fully constrain velocity in 3D. When used as an initial condition, the free perpendicular components default to zero so only one velocity initial condition is required.

A velocity can be applied to nodes, faces or elements.

Each node can have any number of velocity constraints or initial conditions in any directions.

The direction vector's components can be functions of position. The position is the initial modeled position of each node, even if they move during the solution.

## 9.9 Contact

Analysis types: Static 3D, Nonlinear Static, Frequency, Dynamic Response 3D, Nonlinear Dynamic Response, Buckling, Thermal, DC Current Flow, Acoustic Resonance 3D

Solvers: Internal, CCX, OpenRadioss

Contact between two sets of faces can be either **Bonded** (fixed together) or **Unilateral** (able to separate but not penetrate).

### 9.9.1 Bonded

Bonded contact can be useful for connecting parts of an assembly which have non-conforming meshes or for connecting shells and beams to solid elements without creating a pin joint. The faces in one set (the master) must be 2D such as any face of a 3D solid element or the surface of a shell. The faces in the slave set can be of any type, including the end of a beam or the edge of a shell.

**Position tolerance** is the maximum distance that nodes of the slave surface can be from the master surface for them to be connected. **Automatic** means a small value is chosen based on the element sizes.

Three formulations are available: **Constraint equations**, **Tie**, and **Elastic** which have different limitations in each solver as shown in the following table:

	Solid element face	Shell surface	Shell edge	Beam end	Beam edge	Nonlinear large rotations
Constraint equations	CCX and internal	Internal	Internal (slave only)	Internal (slave only)	Internal (slave only)	None
Tie	CCX and OpenRadioss	OpenRadioss	OpenRadioss (slave only)	OpenRadioss (slave only)	OpenRadioss (slave only)	CCX and OpenRadioss
Elastic	CCX and OpenRadioss	CCX and OpenRadioss	CCX	CCX	None	CCX and OpenRadioss

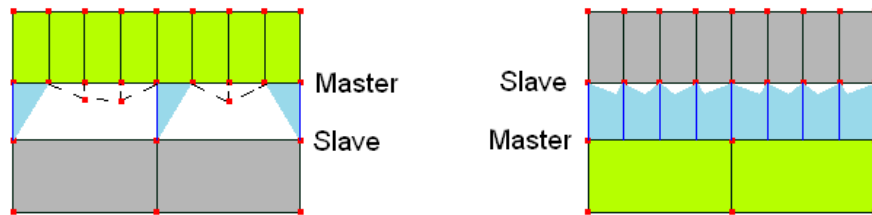
### Constraint equations

It is implemented as a node-to-face contact using constraint equations. One surface is designated the master surface (also called main, independent or target surface) and the other the slave surface (also called secondary, dependent or contact surface). Gaps and penetrations between the surfaces are allowed. Mecway projects each slave node onto at most one master face. The slave node is then constrained to move and rotate along with its projected point. These constraints introduce no artificial stiffness parallel to the master face and they are rigid in the normal direction which prevents the two

surfaces from separating. For other analysis types (thermal, DC current flow, and acoustic resonance), the single DOF of the slave node is constrained to equal the value at the projected point on the master face.

There are several factors to consider when deciding which surface to make the master and which the slave:

- The surface with the coarser mesh should be the master. If the master mesh density is too fine compared to the slave then some element faces of the master surface might not have any slave nodes projected onto them and they will be disconnected.



- The surface with the larger area should be the master. If the slave surface overhangs the edge of the master surface then slave nodes which are too far from the edge won't be connected. However a small tolerance of about 4% of an element face's width in its in-plane direction is allowed. Slave nodes within this distance of the edge of a master face are still connected.



- Dependent DOFs of other constraints are not allowed on nodes of the slave surface. If you want Fixed support or Frictionless support, you can use elastic support with a large stiffness instead.

To help identify disconnected nodes, add the **contact state** field variable to the solution before solving. It shows a value at each node of 1 (in contact), 0 (not in contact), or nothing (unknown). With the CCX solver, nodes may appear as unknown for various reasons, including the presence of Tie contact, the presence of laminates, beam or shell elements not expanded to solids, or time steps.

### Tie

This is similar to **Constraint equations** except that gaps and penetrations should be small because the CCX solver moves nodes of the slave surface in a strain-free way so they lie on the master surface, and the OpenRadioss solver neglects the moment arm due to their separation.

### Elastic, CCX solver

This creates elastic springs between the surfaces. As with **Tie**, the nodes of the slave surface are moved in a strain-free way so they lie on the master surface.

With mechanical analysis types, you must specify **Normal stiffness per unit area** and **Tangential stiffness per unit area**. These define a layer of elastic material between the parts which has no bending or in-plane stiffness and

$$\text{Normal stiffness per unit area} = \frac{E}{t}$$

$$\begin{aligned}\text{Tangential stiffness per unit area} &= \frac{G}{t} \\ &= \frac{E}{2t}\end{aligned}$$

where  $E$  is the Young's modulus and  $t$  is the thickness of the layer.

For ideal hard contact, these would be infinite so choose a finite value that's high enough to keep spurious elastic displacements low but not so high that the solver fails to converge. A rule of thumb to start with for both values is

$$10 \frac{E}{t}$$

where  $E$  and  $t$  here are the Young's modulus and thickness respectively of the parts having contact. Check the solution for excessive displacement and increase by a factor of 10-1000 each time until it's satisfactory.

In thermal analysis, elastic bonded contact puts a thermally resistive layer between the surfaces. You can use this to represent a thermal pad or contact resistance or set **Thermal conductance per unit area** to an arbitrary high value to approximate an ideal connection.

#### Elastic, OpenRadioss solver

This creates elastic springs between the surfaces. Their stiffness is chosen automatically based on the properties of the surface elements. However, you can increase it with a higher **Stiffness factor** to reduce penetration, or reduce it to increase the time step and speed up the solution.

#### **9.9.2 Unilateral**

Only for nonlinear mechanical analysis types, unilateral contact allows contact between two surfaces to be detected so that they don't pass through each other but they can separate and slide, optionally with friction.

#### CCX solver

You must specify **Normal stiffness per unit area** described under **Elastic** above, and, if using friction, also **Tangential stiffness per unit area**.

**Contact stress**, which includes contact pressure, and **contact displacement** are available in the solution. They only appear on slave nodes and are zero everywhere else. The tangential components of contact displacement are only present if friction is defined, otherwise they appear as zero. The normal component of contact displacement (clearance) is only present where penetration occurs and is zero at slave nodes that do not penetrate the master surface. That means only negative clearances appear.

## OpenRadioss solver

Unilateral contact is limited to 2D faces of solid and shell elements and should not have initial penetrations.

## 9.10 Node-surface coupling

Analysis types: Static 3D, Nonlinear Static, Frequency, Dynamic Response 3D, Nonlinear Dynamic Response, Buckling, Thermal, DC Current Flow

Solvers: Internal, CCX

**Node-surface coupling** connects a reference node at an arbitrary location to a surface using multi-point constraints (MPCs) instead of elastic elements. There are two options for how this connection is defined - Rigid (RBE2) and Distributing (RBE3), described below.

Reaction forces and moments include those applied to the nodes by the MPCs. This means you can't typically read reaction forces from constraints on the reference node because they are balanced by the MPCs' forces and appear as zero.

### 9.10.1 Rigid (RBE2)

In mechanical analysis, this fixes all nodes of the surface and a reference node together so that they behave as a single rigid body with 6 degrees of freedom which are the 6 DOF of the reference node.

The surface can be specified as either nodes or faces. In case of nodes, they don't have to define an actual surface.

With the internal solver, there is an option to **Couple surface node rotational DOFs**. This constrains any rotational DOFs of elements in the surface to equal those of the reference node. If the surface includes faces of shell or beam elements, you should turn this on to make it fully rigid or turn it off to leave their rotations about their own nodes free. In either case, rotation of the reference node is still coupled to displacements of the surface nodes.

In thermal, DC current flow, and electrostatic analysis, the single DOF of each node of the surface is constrained to equal that of the reference node so it's a perfect conductor.

### Applications

- Applying remote force, mass, or constraints to a surface.
- Joining two or more parts which may have different geometry, locations and element types.
- Making a surface infinitely stiff without constraining its overall motion.

### Limitations

- The surface nodes are the dependent nodes and must not be the dependent nodes of other constraints.

Additional limitations apply to the CCX solver:

- Constraints on the surface nodes are not allowed.
- Constraints on the reference node should be aligned with the global axes.
- Should not be directly connected to beam or shell elements.

- The reference node must not be shared by more than one node-surface coupling.

### 9.10.2 Distributing (RBE3)

In mechanical analysis, this distributes forces and moments on the reference node to forces on the surface without adding any stiffness to the surface. It also maintains the displacement and rotation DOFs of the reference node to reflect the overall displacement and rotation of the surface.

In thermal, DC current flow, and electrostatic analysis, the single DOF of the reference node is constrained to equal the mean value of that DOF over the surface so it doesn't alter the conductivity of the surface.

#### Applications

- Applying remote force or mass to a surface.
- As a spider to connect a fastener modeled by a beam element to the surface of a plate.
- Connecting solid elements to the end of a beam element without artificially stiffening either part or creating a hinge.
- Although you can apply a moment to the reference node to distribute it over the surface, it can be more convenient to apply the moment directly to the surface instead, especially with quadratic elements. See limitations below.

#### Limitations:

- Not available for the CCX solver.
- The reference node is the dependent node and must not be the dependent node of other constraints.
- Surface nodes must not all be collinear.
- The reference node must not belong to the surface.
- On quadratic elements, the MPC equations contain weighting factors which assume the force is distributed uniformly over each element's face. In case it's not uniform, such as when a moment is applied to the reference node about an axis tangent to the surface, it can cause a spurious zig-zag pattern in the deformed shape. This error reduces with mesh refinement.

## 9.11 Pre-tension section

Analysis types: Static 2D, Static 3D, Nonlinear Static, Frequency, Dynamic Response 3D, Nonlinear Dynamic Response, Buckling

Solvers: Internal, CCX

**Pre-tension section** applies an axial tension or compression to a prismatic mesh made from volume elements (CCX solver only) or a single spring element (internal solver only). A typical application is bolt pre-tension. You can define the load as either a force or a length adjustment:

#### **Force**

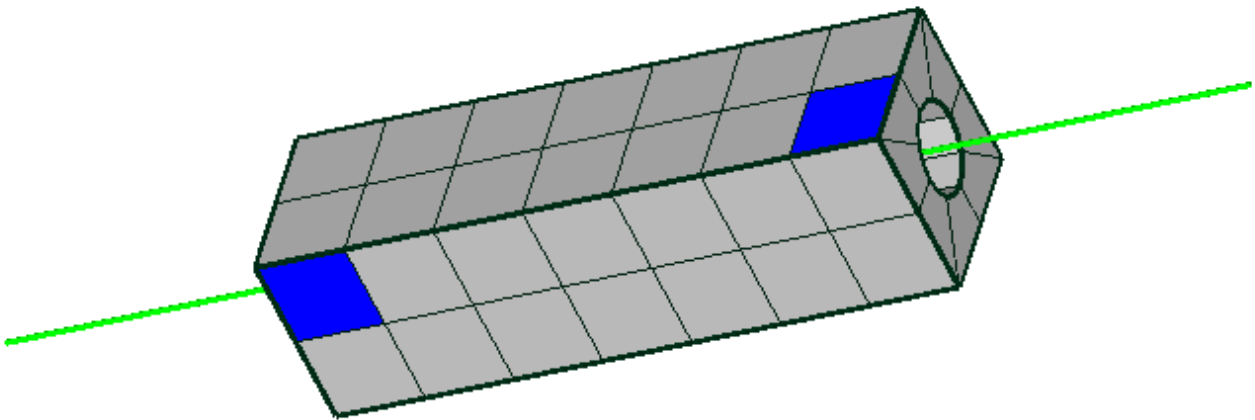
This applies the specified tension (positive) or compression (negative) to the prism. It can be constant or time dependent.

## Length adjustment

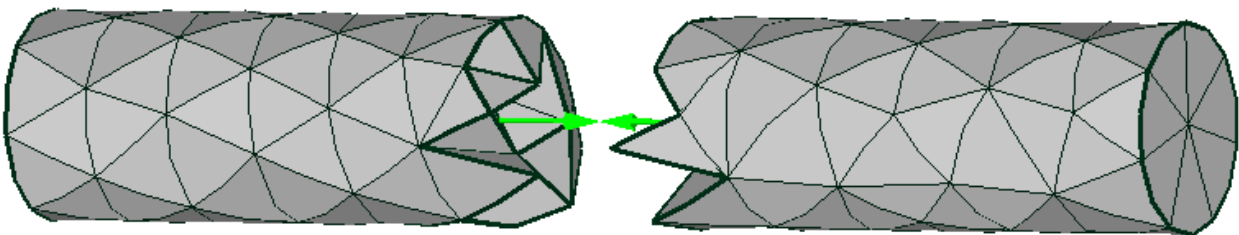
The prism or spring's free length is changed by the specified amount, where a positive value is a shortening. It can be constant or time dependent.

## Prismatic mesh

Apply it to at least two exterior faces oriented at different angles on the surface. These define the orientation of the prism's axis, its extent in the axial direction, and which elements belong to it. The prism's axis is the direction mutually orthogonal to their normal vectors. If you select more than 2 faces, Mecway finds the axes defined by every pair of faces up to a maximum of 1000 faces and uses their median as the axis orientation. It's typically OK to select the entire prismatic surface which can be easier than choosing individual faces. The axial extent of the prism is defined by the maximum axial extent of these faces so you must choose two faces at opposite ends of the region that will contain the pre-tension. The elements within the prism are all the volume elements within its axial extent and connected to any of the selected faces or to any other elements within the prism.



Mecway finds a surface through the cross-section of the prism and CCX uses that to separate the mesh into two parts connected by multi-point constraints. Mecway also inserts a thin layer of hex or wedge elements at the surface to ensure compliance with CCX's requirement that no elements on one part are connected to the surface by only an edge or node.



Limitations include:

- All elements in the prism should belong to the same component, have the same loads and constraints, and have the same element axis orientations specified in Element properties.

- Any loads and constraints on faces or nodes are not extended to the inserted layer of elements. Usually, the error this causes is negligible because the inserted layer is very thin.
- Mecway may be unable to find a surface through the cross-section if the mesh is too irregular. This can happen on unstructured tet meshes with a large number of elements.
- It is not a follower load so if the prism undergoes large rotation, the direction of the pre-tension will not rotate with it.
- Only available with the CCX solver and the following analysis types: Static 3D, Nonlinear Static 3D, Dynamic Response 3D, Nonlinear Dynamic Response 3D.

### Spring element

Apply it to the single edge face of the element. This can only be used with the **Length adjustment** option, the internal solver, and one of the analysis types: Static 2D, Static 3D, Frequency, Buckling.

## 9.12 Flexible joint on beam

Analysis types: Static 3D, Nonlinear Dynamic Response

Solvers: Internal, OpenRadioss

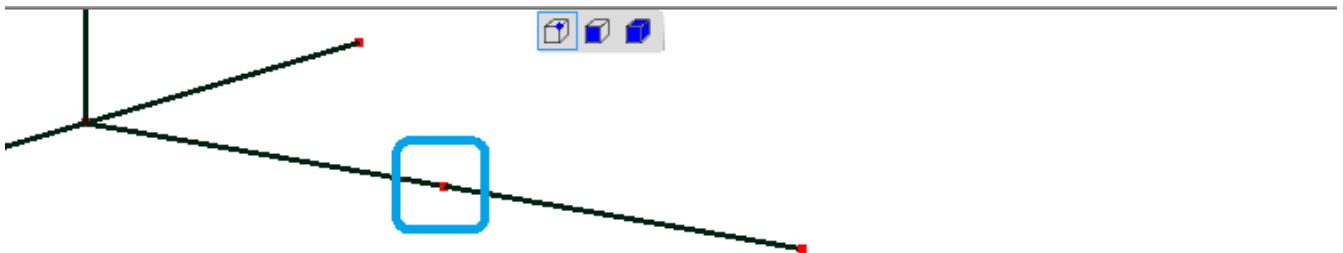
This releases the rotational stiffness of the end of a beam element allowing it to rotate freely even when it's connected to another element or has constrained rotation.

Flexible joint on beam can only be applied to end faces of beam elements.

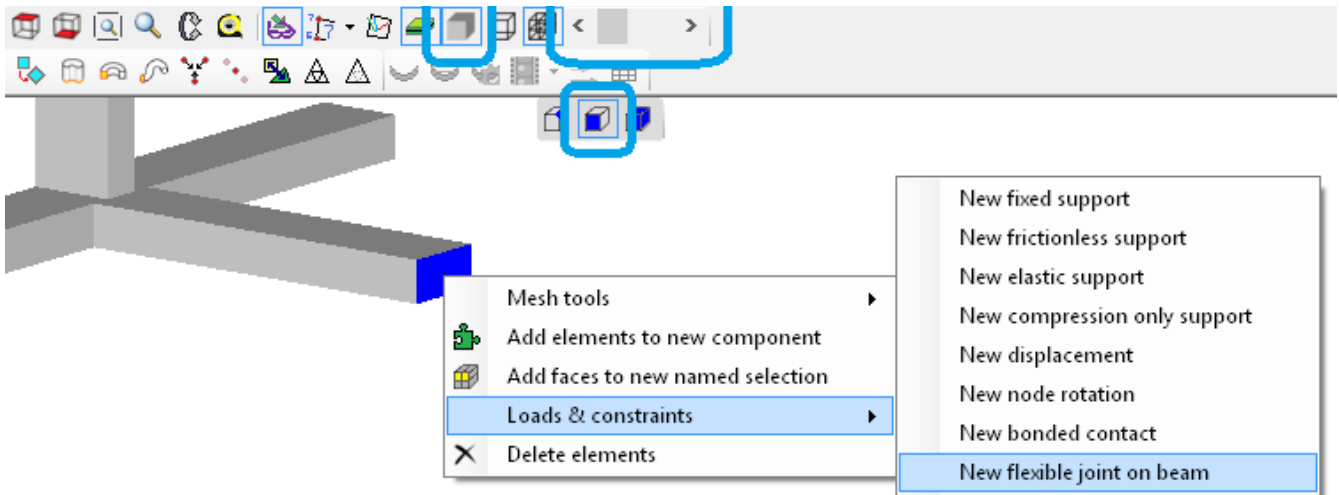
The joint can allow rotation about any combination of the three local axes (U,V,W) of the beam element. The meaning of some combinations are:

- U – The element can rotate about its own axis like being connected to a roller bearing.
- V or W – A pin joint which allows the element to pivot about one axis.
- V and W – A universal joint which transmits torque but not bending moments.
- U, V and W – A ball joint which allows the element to rotate in any direction.

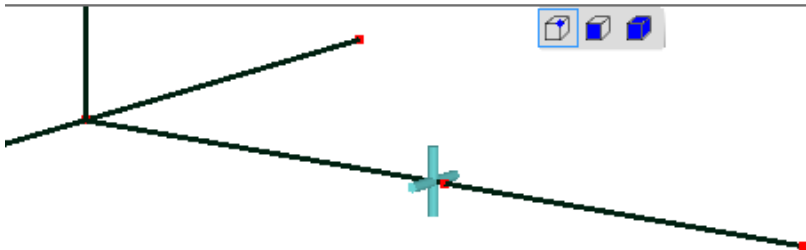
To put a flexible joint at an internal node like this



, turn on **Show element surfaces**, change to **Select faces** mode, and use the **Cutting plane** slider to hide the foreground element(s). Then right click the exposed end face and choose **Loads & Constraints** → **Flexible joint on beam**.



The graphics area then shows pins representing the axes of free rotation.



## 9.13 Force

Analysis types: Static, Nonlinear Static, Frequency, Dynamic Response, Nonlinear Dynamic Response, Buckling

Solvers: Internal, CCX, OpenRadioss

A force applied to faces or elements is uniformly distributed over them. A force applied to nodes is divided equally between the nodes.

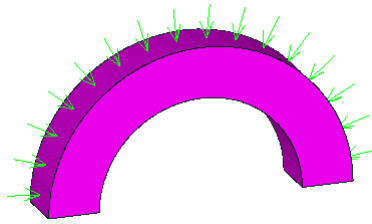
It can be constant or time dependent. The force vector must be specified in global X, Y, Z coordinates.

## 9.14 Pressure

Analysis types: Static, Nonlinear Static, Frequency, Dynamic Response, Nonlinear Dynamic Response, Buckling

Solvers: Internal, CCX, OpenRadioss

Pressure applies a normal force proportional to the area of the face. The sign of the pressure is positive for force directed towards the element.



A pressure can only be applied to faces. It can be constant or time dependent. It can be used with most types of element used in the mechanical analysis types: plane continuum, solid continuum, axisymmetric continuum, shell, beam and truss but not spring or damper elements because they have no surface area.

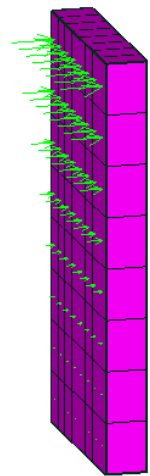
In nonlinear analysis, uniform pressure is a follower load, meaning that if the orientation or area of the faces changes, the total force will change to maintain constant pressure.

With the internal and CCX solvers, pressure can also be defined as a mathematical function of position. To do this, enter a formula for its value using the variables  $x$ ,  $y$  and  $z$  for position. For example, to apply a quadratic pressure load to a vertical (+Y) wall, where the pressure varies from 0 psi at  $Y=0$  to 12 psi at  $Y=2$  ft, enter a pressure of:

$$3*y^2$$

and select units of **ft** for position and **lbf/in<sup>2</sup>** for pressure.

If you define pressure with a formula then on each element face, the pressure function is evaluated at 1, 7 or 9 sampling points for line/point, triangle or quadrilateral faces respectively. This means that rapid changes in pressure might not be captured accurately and may need mesh refinement.

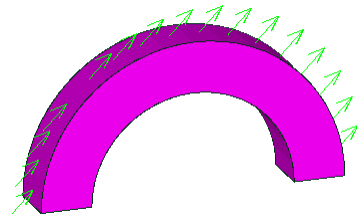


## 9.15 Traction

Analysis types: Static, Nonlinear Static, Frequency, Dynamic Response, Nonlinear Dynamic Response, Buckling

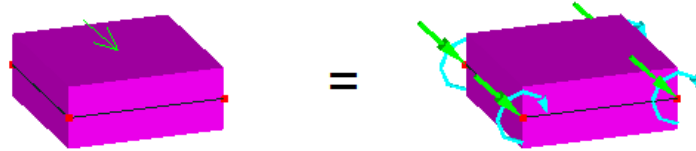
Solvers: Internal, CCX

Traction applies a force proportional to the area of the surface and in a direction specified in global X, Y, Z coordinates.



It can be constant or time dependent and can be used with most types of element used in the mechanical analysis types: plane continuum, solid continuum, axisymmetric continuum, shell, beam and truss but not spring or damper elements because they have no surface area. It can also be defined as a mathematical function of position in the same way as pressure.

On a shell element, the face is offset from the midsurface by half the element thickness so a traction applied to the face generates moments at the nodes as well as forces.



If you don't want these moments, apply it to the elements instead of the faces. Then it acts on the midsurface. You should always do this for curved shells with the CCX solver because those moments usually include a normal component which isn't allowed by CCX.

## 9.16 Line pressure

Analysis types: Static, Frequency, Dynamic Response, Buckling

Solvers: Internal, CCX

Line pressure applies a uniform force per unit length to each edge face or line element. The direction of the force can be specified in either global coordinates (X, Y, Z) or element coordinates (U, V, W) for beam or truss elements.

Line pressure in element coordinates is a follower load for nonlinear analysis types, so the direction and magnitude of the force adjusts with the element's rotation and deformation. Only the V and W components can be used with the CCX solver.

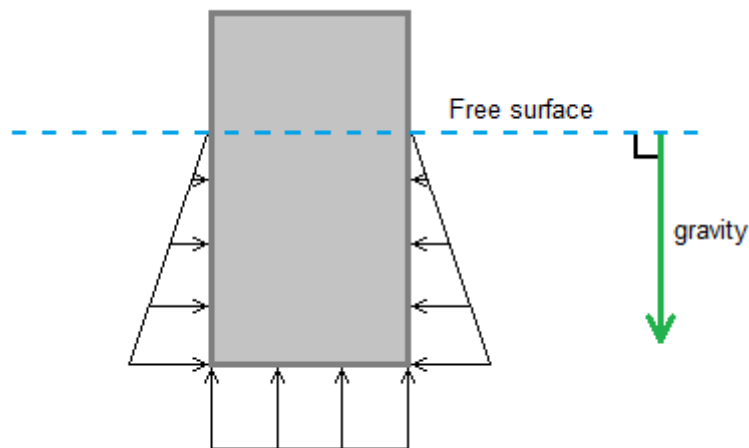
A line pressure can only be applied to line2 or line3 elements or faces that are locally 1-dimensional. This includes the edges of plane and line elements such as shells and beams. It can be constant or time dependent.

## 9.17 Hydrostatic Pressure

Analysis types: Static 3D, Frequency, Dynamic Response 3D, Buckling

Solvers: Internal, CCX

Hydrostatic pressure is a triangular or trapezoidal pressure load that simulates the pressure gradient exerted on a surface by a body of fluid under the influence of gravity.



To use it, you must specify a point on the free surface, the density of the fluid and a separate gravity load.

Mecway converts the pressure into equivalent nodal forces by numerically integrating the pressure function over the area of each face it is applied to. If the entire face is above the free surface then no pressure will be applied to it. If the entire face is below the free surface then the integration will give exact nodal forces. However, if the face intersects the free surface then the integration will typically introduce a small error of less than 1% for that face. To reduce the effect of this error, you can refine the mesh at the free surface or align the edges of elements with the free surface.

A hydrostatic pressure can only be applied to element faces that are locally 2-dimensional. This includes the surfaces of shells and all faces of solid elements.

You can achieve the same effect as hydrostatic pressure by using a pressure load with a formula to define pressure as a function of position. The main difference is that hydrostatic pressure can more accurately capture the abrupt change in pressure that occurs if the free surface intersects an element's face because it uses more sampling points and a piecewise linear pressure function.

## 9.18 Moment

Analysis types: Static 2D, Static 3D, Nonlinear Static, Dynamic Response 2D, Dynamic Response 3D, Frequency, Nonlinear Dynamic Response, Buckling

Solvers: Internal, CCX, OpenRadioss

Moment applies a moment to either a single node with rotational degrees of freedom (beam or shell node), or as forces distributed over a selection of 2D faces (shell or solid faces). The direction of the moment vector is specified in global X, Y, Z coordinates. The axis it act about passes through the position of the node or the centroid of the faces.

Each component can be constant or time dependent.

## 9.19 Gravity

Analysis types: Static, Nonlinear Static, Frequency, Dynamic Response, Nonlinear Dynamic Response, Buckling

Solvers: Internal, CCX, OpenRadioss

Gravity gives a distributed force to all massive parts of a model. This includes elements whose material has density and nodes with mass loads applied to them. It also includes the fluid in any hydrostatic pressure loads.

Specify a vector which is the force per unit mass or acceleration due to gravity. For Earth gravity you can use  $Y=-9.81 \text{ m/s}^2$ . It can be either constant or time-dependent.

## 9.20 Centrifugal force

Analysis types: Static, Nonlinear Static, Frequency, Buckling

Solvers: Internal, CCX

Centrifugal Force puts the model in a rotating reference frame to simulate rotation of the entire model along with the global coordinate system. This gives every massive part (node or element) an outward force whose magnitude is

$$F = m \omega^2 r$$

where  $m$  is the part's mass,  $\omega$  is the angular velocity and  $r$  is the distance from the axis of rotation. The axis of rotation must be one of X, Y or Z.

Angular velocity can be constant or time dependent in nonlinear static analysis. However, since this is a quasi-static simulation, it can't model the forces induced by angular acceleration.

Centrifugal force can increase the radius of rotation of part of a structure, thus further increasing the centrifugal force. This effect is called spin softening and can sometimes have a significant effect on natural frequencies and critical buckling loads. It is automatically taken into account with the Frequency, Buckling and Nonlinear Static analysis types but not linear Static.

You can include a 2<sup>nd</sup> centrifugal force in the Static 3D analysis type with the internal solver. This applies its own loads as described above but the combination of the two also causes the gyroscopic effect of a distributed moment about their mutually perpendicular axis. The overall load at each point with position vector  $\mathbf{r}$  is

$$\mathbf{F} = -m [\boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) + 2 \boldsymbol{\Omega} \times (\boldsymbol{\omega} \times \mathbf{r}) + \boldsymbol{\Omega} \times (\boldsymbol{\Omega} \times \mathbf{r})]$$

where  $\boldsymbol{\Omega}$  is the angular velocity vector of the "outer" centrifugal force load and  $\boldsymbol{\omega}$  is the angular velocity vector of the inner centrifugal force load. You must identify one of them as being outer. The meaning of outer and inner is that the model rotates about the inner rotation axis and the inner rotation axis rotates about the outer rotation axis.

Beam elements that have non-zero density can not be used in the same model as two centrifugal force loads.

## 9.21 Mass

Analysis types: Static, Nonlinear Static, Frequency, Dynamic Response, Nonlinear Dynamic Response, Buckling

Solvers: Internal, CCX

Mass is a lumped mass at a node. It can be used instead of density or to idealize a massive object without modeling it with elements. This mass contributes to dynamic behavior, gravity and centrifugal force. When you apply a single mass load to several nodes, the specified mass is divided equally between each node.

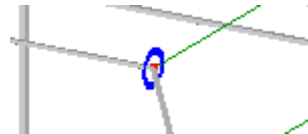


## 9.22 Rotational inertia

Analysis types: Dynamic Response, Frequency

Solvers: Internal

A rotational inertia load is a point rotational inertia about one of the X, Y or Z axes. It can only be applied to the nodes of beam and shell elements. You can use rotational inertia to idealize massive parts such as flywheels without modeling them using elements.



## 9.23 Temperature

Analysis types: Static, Nonlinear Static, Nonlinear Dynamic Response, Frequency, Buckling, Thermal, DC Current Flow

Solvers: Internal, CCX

In a thermal analysis a temperature constraint constrains the temperature to the specified value on the items it is applied to. Heat is allowed to flow into or out of the model at these locations to maintain the specified temperature.

In a mechanical analysis, the temperature is used for thermal stress where a thermal stress load must also be applied.

Temperature can be applied to nodes, faces or elements.

With the internal solver, the temperature can be constrained within a specified time interval in a thermal transient analysis.

Time and position dependent values can be specified as a formula or as a table for time dependence only. With the CCX solver, it cannot be a function of both time and position in the same formula.

## 9.24 Node temperature

Analysis types: Static, Nonlinear Static, Frequency, Buckling, Thermal, DC Current Flow

Solvers: Internal, CCX

Node temperature fixes the temperature of each node individually. It cannot be applied to named selections and is appropriate for a temperature field whose value is different on every node.

A common use for node temperature is thermal stress analysis where the values are generated by the solver in a thermal analysis then used as input for a static analysis.

## 9.25 Thermal stress

Analysis types: Static, Nonlinear Static, Nonlinear Dynamic Response, Frequency, Buckling

Solvers: Internal, CCX

When present, thermal stress enables thermal loading. A single reference temperature must be specified as well as temperatures on parts of the model and thermal expansion coefficients in the materials.

The difference between each element's average specified temperature and the reference temperature is used to determine the thermal expansion/contraction loads. Any nodes that don't have a specified temperature are assumed to be at the reference temperature.

Only one thermal stress can be applied in a model.

## 9.26 Rayleigh damping

Analysis types: Dynamic Response

Solvers: Internal, CCX

Rayleigh damping, also called proportional damping, is a type of viscous damping that's distributed over the elements. It uses a damping matrix (C) formed from a linear combination of the mass (M) and stiffness (K) matrices

$$C = \alpha M + \beta K$$

You can specify values for  $\alpha$  and  $\beta$  to control its behavior. A non-zero  $\alpha$  causes each node to be connected to a fixed external point through a damper whose damping coefficient is related to the mass associated with that node. A non-zero  $\beta$  causes internal damping of deformation of the elements.

You can apply Rayleigh damping to either the whole model or a selection of elements. This allows you to assign different damping properties to different materials or regions of a structure. However, with the CCX solver or mode superposition option, only the whole model is allowed. If you apply more than one Rayleigh damping load to the same element then the damping matrices are summed which is equivalent to summing the damping ratios.

The frequency solver uses a damping ratio for each mode instead of a damping matrix. The damping matrix defined above is equivalent to a set of modal damping ratios

$$\zeta_i = \frac{1}{2} \left( \frac{\alpha}{\omega_i} + \beta \omega_i \right)$$

where  $i$  is the mode number and  $\omega_i$  is the angular frequency of that mode.  $\zeta_i$  must be less than 1 for every mode in the solution.

You can choose values for  $\alpha$  and  $\beta$  by solving the above equation at two points with damping ratios  $\zeta_1$  and  $\zeta_2$  and frequencies  $\omega_1$  and  $\omega_2$  respectively, as shown below. With the mode superposition option,  $\zeta$  for every mode must be less than 1 – i.e. the structure must be underdamped.

$$\alpha = \frac{2 \omega_1 \omega_2 (\zeta_2 \omega_1 - \zeta_1 \omega_2)}{\omega_1^2 - \omega_2^2} \quad \beta = \frac{2 (\zeta_1 \omega_1 - \zeta_2 \omega_2)}{\omega_1^2 - \omega_2^2}$$

## 9.27 Heat flow rate

Analysis types: Thermal

Solvers: Internal, CCX

Heat is added to the model at the specified rate. It is distributed uniformly across the volume or surfaces it is applied to. If it is applied to nodes then the specified heat flow rate is divided equally between each node.

Heat Flow Rate can be applied to nodes, faces or elements.

A time-dependent Heat Flow Rate can be specified as a formula or table of time and value pairs.

## 9.28 Heat flux

Analysis types: Thermal

Solvers: Internal, CCX

Heat is added to the model at the specified rate per unit area. It is distributed uniformly across the surfaces it is applied to.

Heat flux can only be applied to faces.

Time and position dependent values can be specified as a formula or as a table for time dependence only.

## 9.29 Internal heat generation

Analysis types: Thermal

Solvers: Internal, CCX

Heat is added to the model at a uniform rate over the volume it applies to. The value specifies the rate at which heat is added to the model per unit volume.

Internal Heat Generation can only be applied to elements.

Time and position dependent values can be specified as a formula or as a table for time dependence only.

## 9.30 Convection

Analysis types: Thermal

Solvers: Internal, CCX

Convection allows heat flow into or out of a face according to a specified heat transfer coefficient and ambient temperature.

Time and position dependent values can be specified as a formula or as a table for time dependence only. With the CCX solver, it cannot be a function of both time and position in the same formula.

## 9.31 Forced convection

Analysis types: Thermal

Solvers: CCX

This is the same as convection except that instead of transferring heat with a reservoir at an ambient temperature, it transfers heat with a variable temperature moving fluid defined by fluid network elements.

In addition to the faces having convection, you must also specify the fluid network elements to transfer heat with. Mecway automatically associates each face with the nearest element end node among the specified fluid network elements. The fluid network mesh doesn't have to be compatible with the face mesh as each fluid network node can be associated with 0, 1, or many faces.

## 9.32 Radiation

Analysis types: Thermal

Solvers: Internal, CCX

Radiation allows heat flow into or out of a face according to a specified emissivity and ambient temperature. It cannot model heat transfer between two parts of a model.

Time and position dependent values can be specified as a formula or as a table for time dependence only. With the CCX solver, it cannot be a function of both time and position in the same formula.

## 9.33 Mass flow rate

Analysis types: Thermal

Solvers: CCX

This constraint prescribes the mass flow rate through a fluid network element. A positive value is in the direction from node 1 to node 3.

### 9.34 Fluid pressure

Analysis types: Acoustic Resonance

Solvers: Internal

In the Acoustic Resonance analysis type, you can use fluid pressure=0 to enforce a pressure node. Node here means a location where the pressure remains zero throughout the wave's cycle.

### 9.35 Impedance

Analysis types: Acoustic Resonance

Solvers: Internal

You can apply the impedance boundary condition to element faces to model an absorber with real or complex specific acoustic impedance.

It does not support frequency-dependent impedance, however, you can set up a configuration for each of several impedance values and choose modes from the solution of each configuration according to their frequencies.

Impedance cannot be used in the same model as bonded contact or constraint equations.

### 9.36 Electric potential

Analysis types: Static 2D, Static 3D, DC Current Flow

Solvers: Internal

Electric potential applies a uniform potential over every item it is applied to. In DC current flow analysis, current can flow into or out of the model as needed to maintain that potential. This is a voltage source relative to a common ground.

### 9.37 Charge

Analysis types: Static 2D, Static 3D

Solvers: Internal

Charge defines an electric charge uniformly distributed across the volume or surface it is applied to. If it is applied to nodes then the specified charge is divided equally between each node.

Charge can be applied to nodes, faces or elements.

### 9.38 Current

Analysis types: DC Current Flow

Solvers: Internal

Current defines an electric current flowing into the model and uniformly distributed across the surface it is applied to. If it is applied to nodes then the specified current is divided equally between each node.

Current can be applied to faces or nodes.

A positive value on the face of an element represents current flowing into the element.

### 9.39 Robin boundary condition

Analysis types: Static 2D, Static 3D, DC Current Flow

Solvers: Internal

The Robin Boundary Condition imposes the following equation on a face

$$k \partial u / \partial n + c_0 u + c_1 = 0$$

where

- $u$  is electric potential
- $k$  is either the electrical conductivity or absolute permittivity of the material
- $\partial u / \partial n$  is the normal derivative of  $u$  at the face
- $c_0$  and  $c_1$  are user specified parameters. If  $c_0 = 0$  this is a Neumann boundary condition, constraining the field value's gradient normal to the face.

### 9.40 Symmetry

Analysis types: Static 3D, Nonlinear Static, Frequency, Dynamic Response 3D, Nonlinear Dynamic Response

Solvers: Internal, CCX, OpenRadioss

The symmetry constraint enforces a mirror symmetry boundary condition by applying displacement and rotation constraints to the nodes of the faces. The displacement constraints are oriented normal to the symmetry plane and the rotation constraints (for beam and shell nodes) are tangent to it.

It can only be applied to the faces of solid elements, edges of shells, and end faces of beam/truss/spring/damper elements.

You can choose to define the symmetry plane with a vector or use **Symmetry plane defined by nodes**. For **Symmetry plane defined by nodes**, the faces must have at least 3 non-collinear nodes and all nodes should be coplanar.

On the edges of shell elements, only the rotational DOF tangent to the edge is constrained. Constraint of rotation about the other direction (normal to both the symmetry plane normal and the shell edge tangent) is provided by the displacement constraints on adjacent nodes. The reason for omitting this constraint is that it doesn't work reliably with the CCX solver.

Although you can also use **frictionless support** as a symmetry boundary condition, the advantage of **symmetry** is that all the node constraints it generates are aligned with the same plane even if the individual face normals are different. This is particularly useful on the edges of shells and ends of beams where the face normal is often not the same as the symmetry plane normal.

## 9.41 Cyclic symmetry

Analysis types: Static 3D, Nonlinear Static, Frequency, Thermal Steady State

Solvers: Internal, CCX

If geometry and loading are regularly repeated about an axis of rotation, then cyclic symmetry (rotational periodicity) can be used to reduce the model to only one segment, allowing a smaller mesh and improved solver performance.

You must specify each cut surface as either the master or slave using either node or face selections.

In thermal analysis, cyclic symmetry is equivalent to applying constraint equations coupling the corresponding pairs of nodes on the cut surfaces.

In static analysis, cyclic symmetry is equivalent to applying constraint equations which equate the radial, circumferential and axial displacements of corresponding nodes on the cut surfaces.

In frequency analysis, cyclic symmetry applies transformed coordinates and couples the degrees of freedom using special complex constraints described in (*D. L. Thomas, Dynamics of Rotationally Periodic Structures, 1979*). This method can find modes which are not cyclically symmetric. The number of modes specified for the solution is not the total number of modes, but the number of modes having the same number of nodal diameters. For example if 5 modes are specified in a model with 7 segments, the solver will find the lowest 5 modes with each of 0, 1, 2 and 3 modal diameters, leading to a total of 20 modes.

Differences between the internal and CCX solvers:

CCX	Internal
Master and slave meshes don't have to be compatible.	Master and slave node layouts must match each other.
The surface with the finer mesh should be the slave to ensure all master faces have corresponding slave nodes.	
Shells and beams are not allowed in any analysis type.	Shells and beams are not allowed in frequency analysis type.
No constraints allowed on the slave surface.	Constraints on the cut surfaces are limited to: - Displacement in the direction of the symmetry axis - Zero displacement normal to the symmetry axis - Temperature
Any single point constraints allowed on the master surface.	
Solution shows the number of segments requested.	Solution only shows one segment with static and thermal analysis types.

## 9.42 Constraint equation

Analysis types: Static, Frequency, Dynamic Response, Buckling, Thermal Steady State, DC Current Flow, Acoustic Resonance

Solvers: Internal, CCX

A constraint equation (also called multipoint constraint or MPC) is a more general type of constraint which enforces a linear relationship between several DOFs (degrees of freedom). Examples of uses are:

- The X-displacement of one node can be set equal to the Y-displacement of another node so they act as if they are connected by a 90° bell crank such as  $ux_1 - uy_2 = 0$ .
- In DC current flow, a voltage source can be made by setting a constant difference in the potential of two nodes such as  $V_2 - V_1 = 5V$ .

A constraint equation has the form

$$A \times \text{dof}_1 + B \times \text{dof}_2 + \dots = C$$

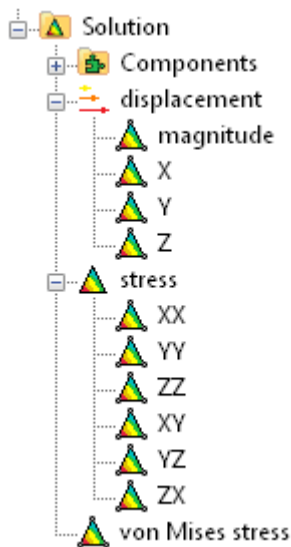
where A, B and C are constants. The first DOF, with coefficient A is the dependent DOF.

With some exceptions, there are restrictions that constraint equations must comply with:

- The dependent DOF must not be a dependent DOF of any other constraint equation or constraint.
- The dependent DOF's coefficient must not be zero.
- Each DOF can only appear once in the same constraint equation.
- The constant (C) must be zero in frequency, dynamic response, buckling, acoustic resonance, and all analysis types with the CCX solver.
- With the CCX solver, they typically cannot refer to the rotational DOFs of beam or shell elements.

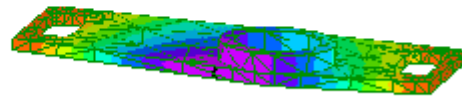
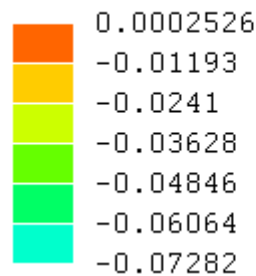
## Results

### 10.1 Display



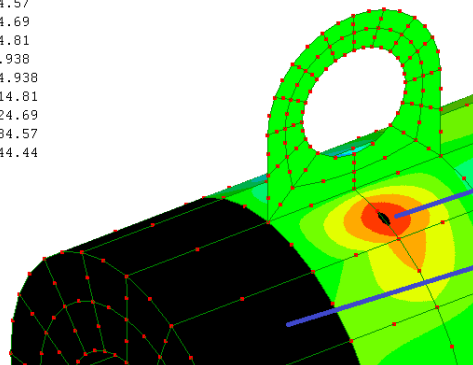
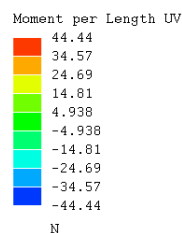
Select an item below **Solution** to display a field variable as a color coded contour plot. These are only available after solving the model. You can add additional field variables by right clicking the **Solution** branch.

Displacement in Z



The contour plot can be configured using by selecting **Options** then **Contour plot** from the **Tools** menu. Here you can change the color scheme, number of colors and resolution. A higher resolution is slower to render but gives smoother boundaries between colors so it can be useful for presentation. The **Gradient** option gives a smoother image which is also faster to generate with large meshes however it doesn't accurately show changes of color within a single element face.

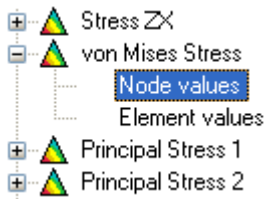
Sometimes black regions appear in the contour plot. This happens when there's no correct color that can be used. One reason is the value is outside the range of values that map to colors. It can also occur if the field value being displayed doesn't exist on that element. For example, solid elements have no bending moment so bending moments appear black. You can change this from black to



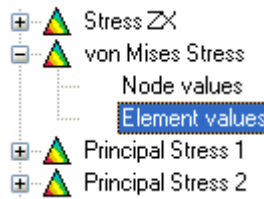
Value is greater than 44.44 N

Moment per Length UV does not exist on solid elements

another color using **Tools** → **Options** → **Contour plot** and **Above range color**, **Below range color**, and **No value color**.



**Node values** will display the results at each node as the average of the values from all its adjacent elements. This produces a continuous plot.

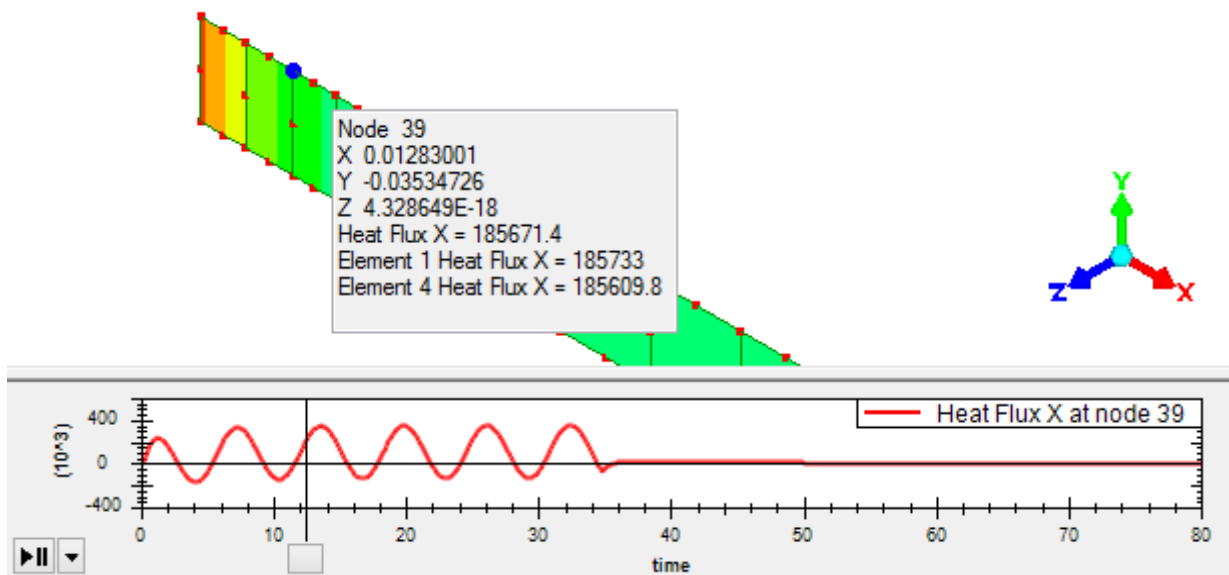


**Element values** will display the results without nodal averaging. Beam and shell values that are expressed in local element coordinates (U, V, W) should be displayed this way because the meaning of values on adjacent elements may be different. Not available with the CCX solver.

To turn off the contour plot, click the **Components** branch under **Solution** in the outline tree.

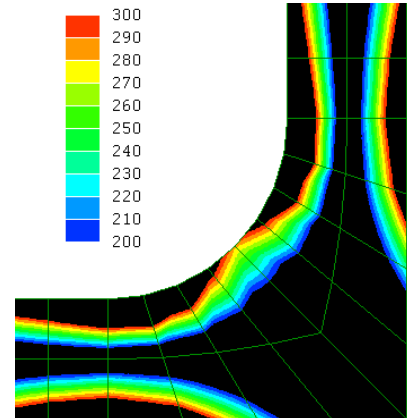
Click on a node to display both its averaged node value and the non-averaged values of all its adjacent elements. If there are time steps in the solution, the timeline will also show a graph of the node's value.

You can export the graph's data to a text file by right clicking on the graph and clicking **Export to CSV file...** or copy it to the clipboard with **Copy data**.

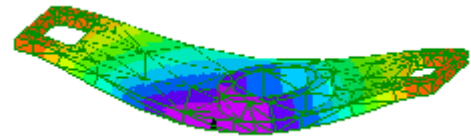


You can control the range of values displayed in the contour plot by moving the sliders to the left of the color key. Parts of the model outside the specified range will be displayed using the **Above range color** or **Below range color** which default to black.

An automatic way to set the range of values is to right click some selected elements and click **Fit contour plot max/min to selection**. This adjusts the scale to match the selected elements and ignore everything else.



**Deformed view** shows exaggerated deformation. A default scale factor is chosen automatically to give a clearly visible change. The minimum automatic scale factor is 1 except for mode shapes.



**Undeformed shape** superimposes the undeformed profile on the exaggerated deformation. The dropdown menu allow you to display it as either element edges or model edges.



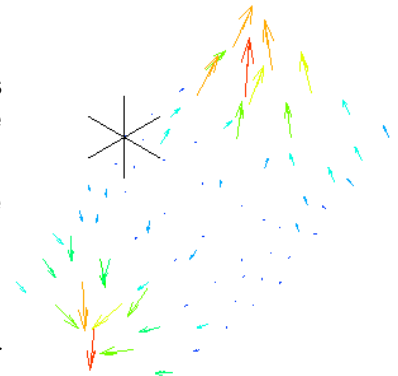
Vector fields are displayed as arrows if you select the group name in the outline tree, such as **displacement** or **velocity**.



**Animation** animates the deformation. This only works for results without time steps such as static and frequency. For results with time steps such as dynamic response, use the timeline described below.

Time steps in transient problems are displayed on the timeline, the length of which is the total time period of the analysis. The result in the graphic display will correspond to the slider's position.

To animate the display, click the play/pause button. The drop-down menu can be used to control the speed. You can use this together with **Deformed view** to show movement.



To move the slider in single steps, click its handle then press the left and right arrow keys.

To save the animation as an MPEG file, click the down arrow next to the animation button and choose **Save video to file**.



**Table** displays the results in a table which can be copied and pasted into a spreadsheet. The options you set here are saved with the model so you can easily generate the same table after solving again. If you enter a file name under **Write to csv file after solving** Mecway will save the table to the specified text file every time it solves the model. This file can be opened with a spreadsheet application or other software for further processing. If you omit the extension, no extension will be added to the name and the file will be saved without an extension. You can add multiple tables to

output different parts of the solution to different files and you can suppress different tables in different configurations.

**Solution** → **Find max/min** shows lines pointing to the maximum and minimum values in whole mesh if nothing is selected, or the selected nodes or elements if any are selected.

## 10.2 File Output

Various results can be written to files when the solver runs. Most of these options are available by right click on the **Analysis** item at the top of the outline tree. However, to write the solution data in tabulated form to a csv file, use the **Table** item under the **Solution** branch of the outline tree. This is described in the previous section.

If you include the special code <f> in the output file's filename, it will be replaced by the current file's filename without extension. For example, a model called *example.liml* with **save after solve** set to <f> *solved.liml* will save its solution in the file *example solved.liml*.

**Save after solve** saves the model and solution to the specified file.

**Export stiffness matrix** saves the stiffness matrix to a text file. It can be stored in either a sparse or dense format. The **Dense** format contains the entire square matrix with one row on each line of the file. The **Sparse** format doesn't contain most of the zero elements and has only the upper triangle of the symmetric matrix. Each line of the file contains 3 numbers: the row number, the column number and the value in that location of the matrix.

**Export mass matrix** saves the mass matrix with the same two format options as the stiffness matrix.

**Export load vector** saves the global load vector to a file with the value at each DOF on a separate line.

**Export DOF ordering** saves the relationship between node numbers and DOFs and the row numbers of the matrix. Each line of the file contains the node number, DOF name and matrix row number separated by commas. The matrix row number may be shown as 0 where the global DOF has been condensed out of the matrix due to constraints.

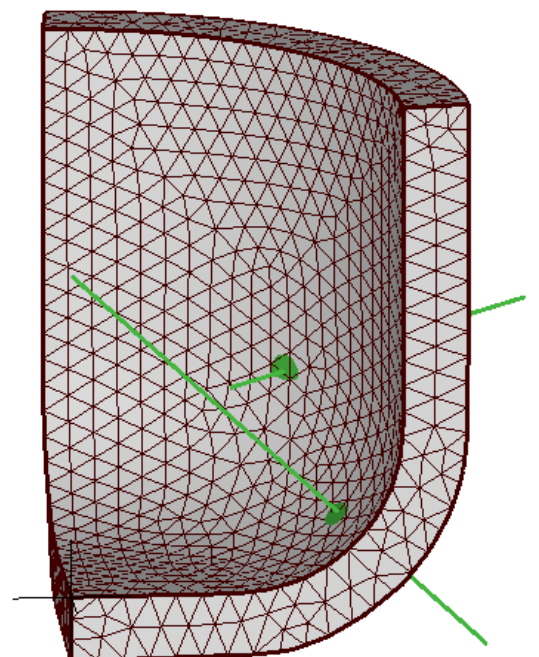
## 10.3 Saved Monitor

Adding **Saved monitor** to the solution keeps the solver's progress messages stored with the model. You can add it either before solving or after solving and before closing the solver window or monitor.

## 10.4 Stress Linearization

This tool finds linearized stresses along a straight line (stress classification line) in a solid model. It can be used for pressure vessel design with ASME or other codes. It interpolates 101 uniformly spaced points between the two surfaces of the mesh along a Stress Classification Line (SCL) defined by two endpoints. It then produces a graph and table showing linearized von Mises, maximum shear stress  $\times 2$  (stress intensity), principal stresses, and/or stress components transformed to the SCL's coordinates.

There is no need to have mesh nodes on the SCL. Mecway interpolates stress values from the elements that the line passes through.



To create a Stress Linearization item, click **Solution** → **Stress linearization**. You can specify the SCL endpoints either by entering 2 sets of coordinates or by selecting two nodes in the solution before opening the Stress linearization tool. The endpoints can be outside the mesh, in which case, only the section inside the mesh will be used as the SCL.

#### 10.4.1 SCL coordinates

Stress is first transformed into the SCL coordinates x, y, z. x is parallel to the SCL, in the direction away from the global origin. y and z form an orthogonal right-handed coordinate system with x. By default, y and z are chosen automatically and this choice doesn't affect the results.

If you want to isolate the stress components, you must specify **Direction of y axis**. Mecway obtains the y axis by projecting the direction you set onto the plane normal to the x axis. The only available directions are the 3 global axes so you may have to orient your model to be aligned with them before solving.

#### 10.4.2 Linearization

The 1-dimensional stress field along the SCL is decomposed into constant, linear, and higher-order terms:

$$\sigma = \sigma^m + \sigma^b + \sigma^{\text{peak}}$$

Total Stress  $\sigma$  is the non-linearized stress tensor interpolated from the values at the nodes.

Membrane stress ( $\sigma^m$ ) is constant along the SCL and is determined by evaluating the following integral using the trapezoidal method:

$$\sigma^m = \frac{1}{L} \int_{-L/2}^{L/2} \sigma dx$$

L is the length of the SCL.

Each component of bending stress ( $\sigma^b$ ) varies linearly along the SCL, passing through zero at the center. The bending stress at the beginning of the SCL is determined by evaluating this integral using the trapezoidal method:

$$\sigma^b(0) = \frac{-6}{L^2} \int_{-L/2}^{L/2} \sigma x dx$$

However, if **Exclude  $\sigma_{xx}$ ,  $\sigma_{xy}$ ,  $\sigma_{zx}$  from bending stress** is checked, then the components of bending stress parallel to the SCL are set to zero.

$$\sigma_{xx}^b = 0$$

$$\sigma_{xy}^b = 0$$

$$\sigma_{zx}^b = 0$$

Mecway doesn't show bending stress directly, but the sum membrane + bending stress instead:

$$\sigma^{m+b} = \sigma^m + \sigma^b$$

Peak stress is the remaining stress besides membrane and bending:

$$\sigma^{\text{peak}} = \sigma - \sigma^{m+b}$$

## 10.5 Volume Mean

**Solution** → **Volume Mean** evaluates the mean of a scalar field over a domain defined by 3D solid or 2D plane elements. An example use is to find the average temperature of a non-uniformly heated object.

If you want to use a domain which isn't the complete model, you must define it as an element selection or component before solving.

## 10.6 Surface Mean

**Solution** → **Surface Mean** evaluates the mean of a scalar field over a surface defined by face selections. An example use is to find the average temperature of a surface.

## 10.7 Volume Integral

**Solution** → **Volume Integral** integrates a scalar field value over a domain defined by 3D solid or 2D plane elements. An example use is to find the total power dissipated in a solid conductor by integrating power density.

## 10.8 Surface Integral

**Solution** → **Surface Integral** integrates a scalar field over a surface defined by face selections. An example use is to calculate current from current density.

## 10.9 Surface Normal Integral

**Solution** → **Surface Normal Integral** integrates the dot product of a vector field and the outward surface normal vector. An example use is to calculate heat flow rate from heat flux.

## 10.10 Sum

**Solution** → **Sum** sums the values of solution variables on the nodes of predefined named selections. It can only be used for reaction force components, external force components, formulas, and unknown field variables imported from external solvers.

## 10.11 Formula

**Solution** → **Formula** generates a new field variable that's a function of existing field variables, position, time, and some material properties. The available variables and operators are listed in the formula window. Some examples:

### 10.11.1 Reaction moment

Reaction moments composed of force couples are not included in the default reaction moment output. However, you can calculate the moment of each reaction force about the origin using the following formulas and then use **Solution** → **Sum** to sum them to a single value.

Reaction moment about X:  $FR.z * y - FR.y * z + MR.x$

Reaction moment about Y:  $FR.x * z - FR.z * x + MR.y$

Reaction moment about Z:  $FR.y * x - FR.x * y + MR.z$

The  $MR.x$ ,  $MR.y$ , and  $MR.z$  terms must not be included if there are no elements with rotational DOFs.

### 10.11.2 Cylindrical coordinate system

The files *DisplacementCylindrical.liml* and *StressCylindrical.liml* in the samples folder contain formulas which transform the displacement and stress fields of a solution into cylindrical coordinates with Z as the axis of symmetry. You can use **File** → **Import...** to add these formulas to an existing model.

## Samples and Verification

### 11.1 BeamBendingAndTwisting.liml

Analysis type: Static 3D

Elements: Beam (line2)

Loads and constraints: Fixed support, Force, Moment

This sample shows Mecway finding the longitudinal stresses due to bending of a beam using a predefined cross-section. It also shows torsional stiffness and that torsion does not influence longitudinal stress. The beam is made from a hollow rectangular tube with properties:

Width:  $a=23\text{mm}$

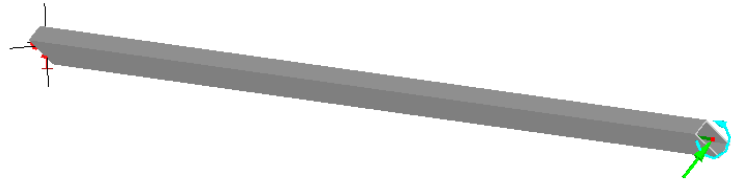
Height:  $b=50\text{mm}$

Wall thickness:  $t=2\text{mm}$

Length:  $l=2\text{m}$

Young's modulus:  $E=200\text{GPa}$

Poisson's ratio: 0.3

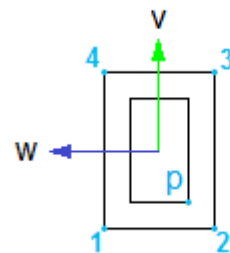


The beam is oriented at 45 degrees to the Y axis and is fully constrained at one end and loaded at the other end with a lateral force of 1N and a twisting moment of 2Nm.

Mecway reports stress at 5 locations on the cross-section: each of the four outer corners and a user defined point p on an inside corner. These results are compared to hand calculations below.

The longitudinal stress at the base of the beam at point 1 ( $W_1 = 12.5\text{mm}$ ) should be

$$\sigma_1 = \frac{M_v W_1}{I_{vv}} = 0.844490 \text{ MPa}$$



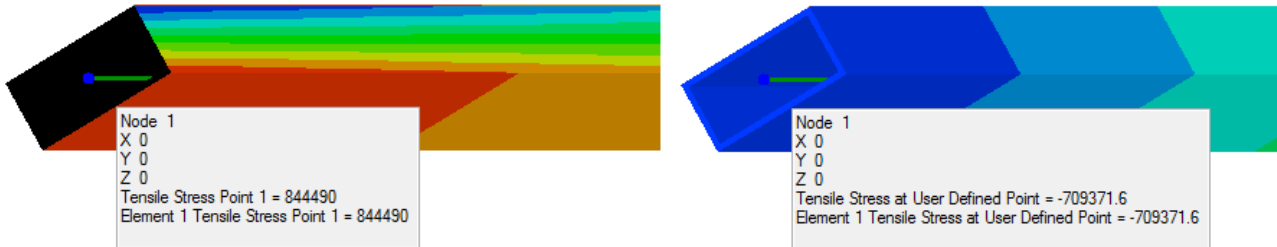
The longitudinal stress at the user defined point p ( $W_p = -10.5\text{mm}$ ) should be

$$\sigma_p = \frac{M_v W_p}{I_{vv}} = -0.709372 \text{ MPa}$$

The bending moment about the V axis ( $M_V$ ) is  $2\text{m} \times 1\text{N} = 2\text{Nm}$ . The 2nd moment of area about the V axis ( $I_{VV}$ ) can be found using a readily available formula and the parallel axis theorem:

$$I_{VV} = 29603 \frac{2}{3} \text{mm}^4$$

Mecway shows the same stress values:



The twisting angle can be found using

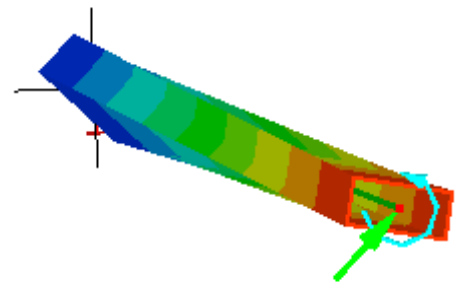
$$\theta = \frac{LT}{GJ}$$

where  $L$  is the beam length (2m),  $T$  is the applied torque (2Nm),  $G$  is the modulus of rigidity ( $\frac{200}{2.6} \text{GPa}$ ), and  $J$  is the torsion constant which is approximated by

$$J \approx \frac{2t(a-t)^2(b-t)^2}{a+b-2t} = 6.86657 \times 10^{-8} \text{m}^4$$

(Warren C. Young, *Roark's Formulas for Stress & Strain*)

This gives  $\theta = 7.5729 \times 10^{-4}$  radians which is the same as the twist angle found by Mecway because Mecway uses the same approximate formula for  $J$ .



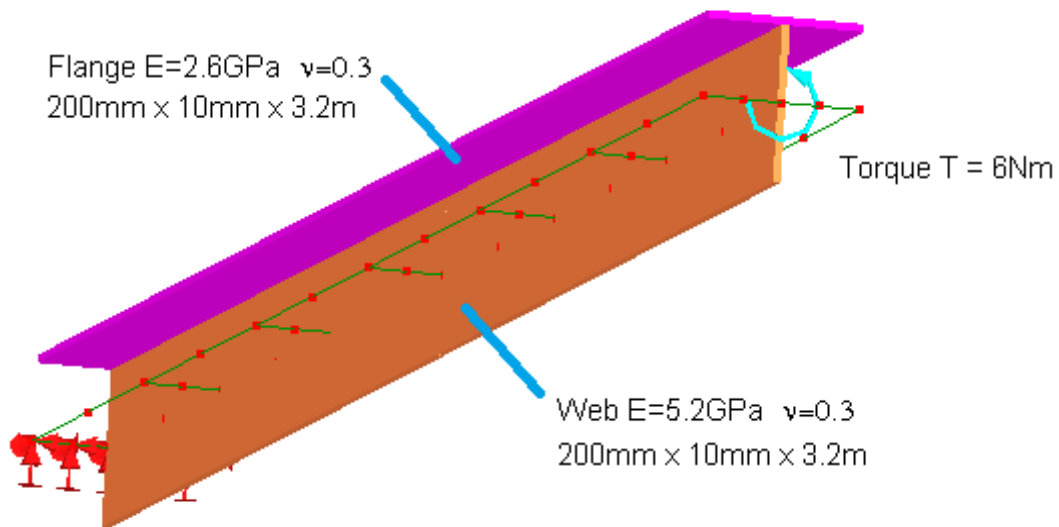
## 11.2 CompositeBeam.liml

Analysis type: Static 3D

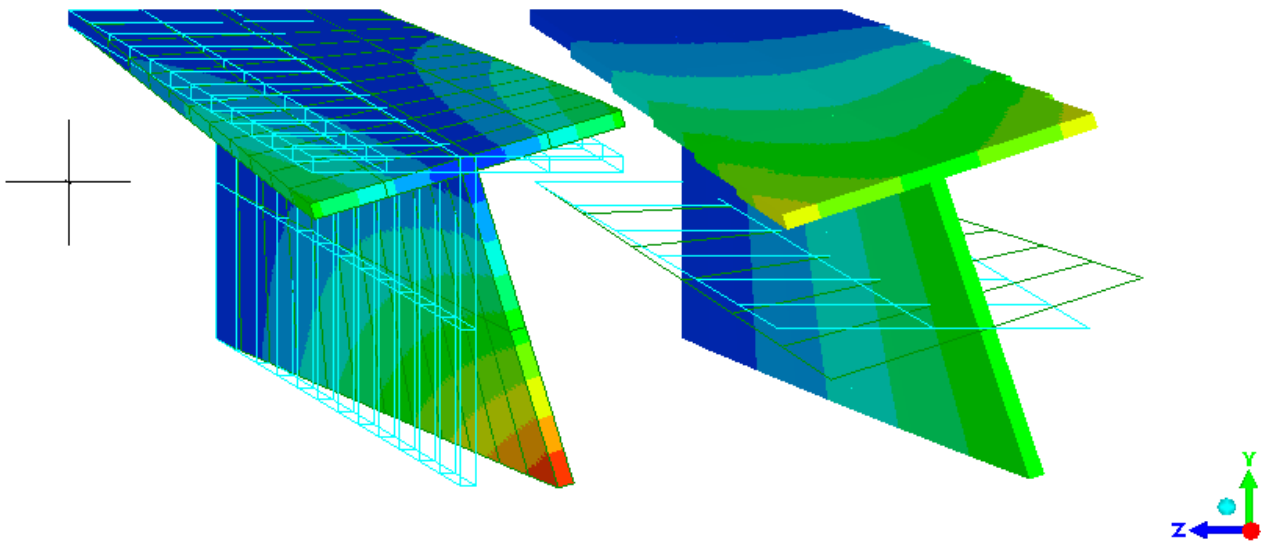
Elements: Beam (line2), Shell (quad8), Solid continuum (hex20)

Loads and constraints: Fixed support, Force, Moment

This beam is made from two different materials by using a combination of beam and shell elements. The web is made from a line of beam elements whereas the flange is made from shells connected to the beams elements. A 0.105m **Shell offset** is specified in **Element properties** to locate the shell elements at the edge of the web rather than at the center.



An equivalent solid model is included in the same file for comparison. The solutions to both models show the beam rotating about a point at the intersection of the web and flange because this is the location of the shear center.



We will now calculate the angle of twist by treating the web and flange stiffnesses independently:

Torsion constant of either the flange or the web

$$J = \frac{Bt^3}{3} = \frac{0.2 \times 0.01^3}{3} = 6.6667 \times 10^{-8} m^4$$

Shear modulus of flange

$$G_{flange} = \frac{E_{flange}}{2(1+0.3)} = 1 \times 10^9 Pa$$

Shear modulus of web

$$G_{web} = \frac{E_{web}}{2(1+0.3)} = 2 \times 10^9 Pa$$

Angle of twist

$$\theta = \frac{IT}{J(G_{flange} + G_{web})} = \frac{3.2m \times 6Nm}{6.6667 \times 10^{-8} m^4 (1 \times 10^9 Pa + 2 \times 10^9 Pa)} = 0.108 \text{ radians}$$

This closely matches the angle of 0.109 radians given by Mecway for rotation about X at the tip of the beam.

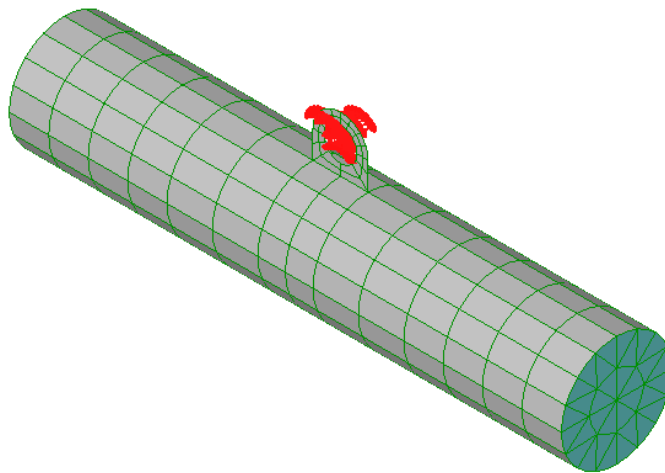
### 11.3 CylinderLifting.liml

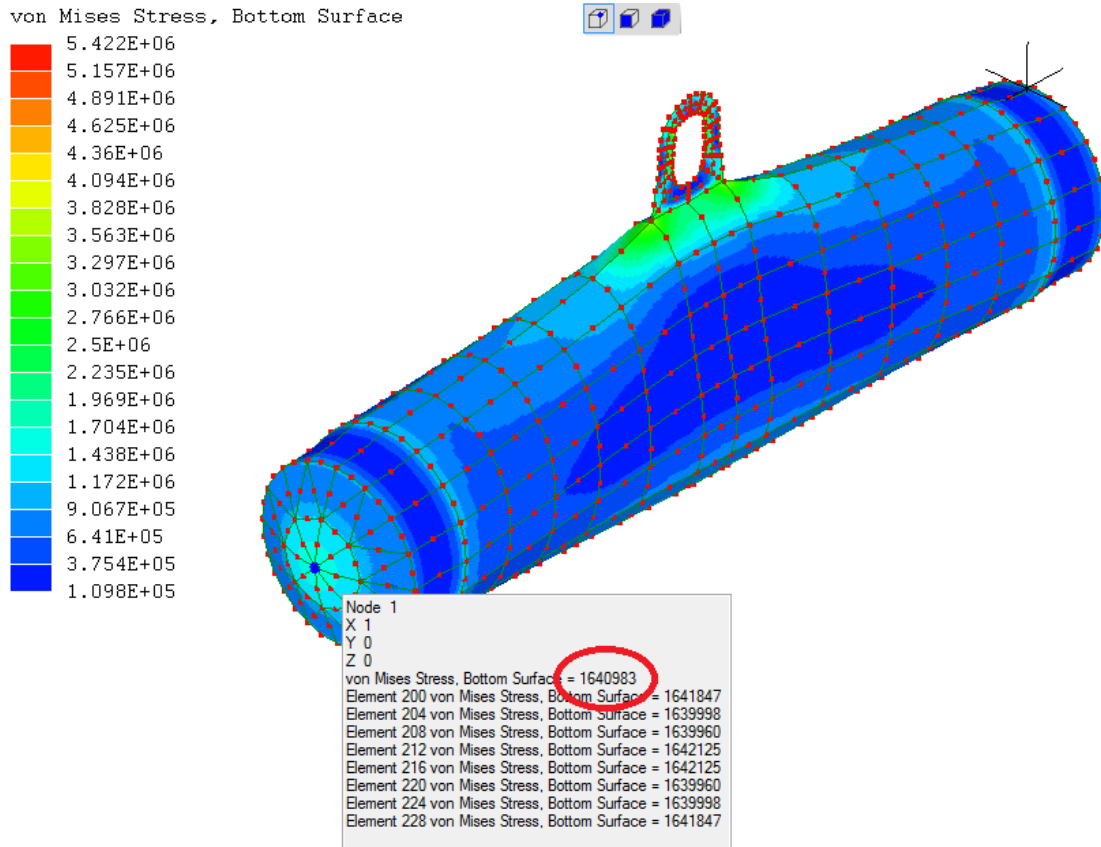
Analysis type: Static 3D

Elements: Shell (tri6, quad8)

Loads and constraints: Fixed support, Gravity, Pressure

This sample shows the use of shell elements for a pressure vessel in two different configurations – self weight under gravity, and the same with an additional internal pressure load. It also uses material of different thickness for the end caps.





In the With\_pressure configuration, the von Mises equivalent stress on the outside surface at the center of each end cap is  $1.64 \times 10^6 \text{ Pa}$ . Mesh refinement to 16 times the number of elements shows this stress to be in error by about 2%. The outside surface is called the bottom surface in the solution. You can see this by turning on **Show element axes** to show the direction of the blue W axis of each element.

## 11.4 PressureVesselAxisymmetric.liml

Analysis type: Static axisymmetric

Elements: Axisymmetric continuum (quad8)

Loads and constraints: Displacement, Pressure

This is a cylindrical pressure vessel with semi-ellipsoidal ends. It is modeled as axi-symmetric because the geometry and loads all have axial symmetry about the longitudinal axis. Only one half is modeled due to mirror symmetry and a displacement constraint in the Y direction is applied to enforce this boundary condition.

It is made from steel ( $E=200\text{GPa}$ ,  $\nu=0.3$ ) with wall thickness  $t=20\text{mm}$ , inside radius  $r=500\text{mm}$  and internal pressure  $P=1\text{MPa}$ .

Three tests are used to validate the results:

The longitudinal reaction force in the cylindrical wall at the mirror symmetry boundary is

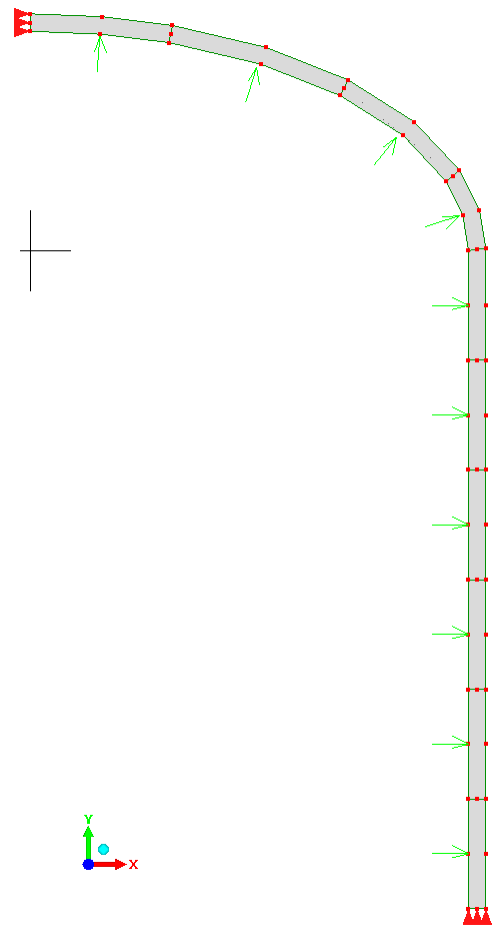
$$F_Y = -P \times \pi r^2 = -785\text{kN} \text{ compared to Mecway's } -798\text{ kN}$$

The axial stress in the cylindrical wall is

$$\sigma_{YY} = \frac{F_Y}{\pi((r+t)^2 - r^2)} = 12.25\text{MPa} \text{ compared to Mecway's } 12.44\text{ MPa}$$

The hoop stress in the middle of the cylindrical wall is

$$\sigma_{ZZ} = \frac{Pr}{t} = 25.00\text{MPa} \text{ compared to Mecway's } 25.00\text{ Mpa}$$



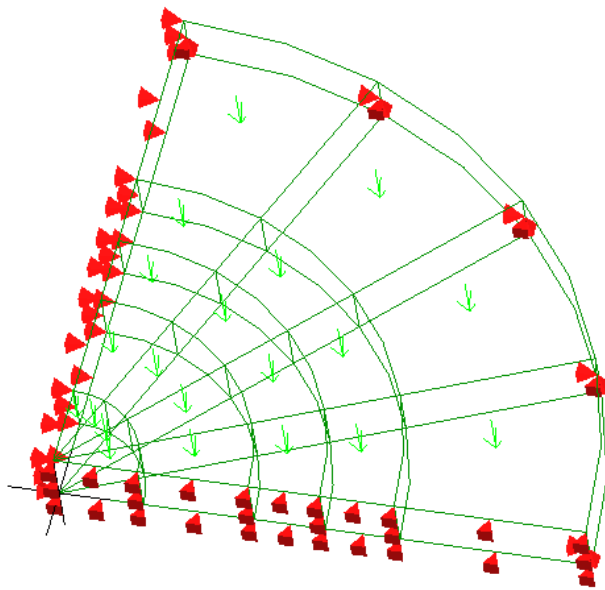
## 11.5 MembraneActionPlate.liml

Analysis type: Nonlinear Static

Elements: Solid continuum (hex20)

Loads and constraints: Frictionless support, Displacement, Pressure

This sample demonstrates the nonlinear effect of membrane action to stiffen a circular plate supported at the edges. The supports allow rotation about the edge but not displacement in any direction. Because radial displacement is constrained, when a pressure load is applied to the surface and the plate deflects, a tensile membrane stress is developed within it and further deflection is resisted.



Only one quarter of the plate is modeled due to symmetry. Frictionless supports are used to enforce mirror symmetry on two surfaces.

The solution shows the Z-displacement at the center of the circle to be -1.5mm which is significant compared to the 1mm thickness of the plate.

Changing the analysis type from Nonlinear Static 3D to Static 3D then solving it again shows a much greater center displacement of -7.8mm. This linear solution is wrong because it is made with the assumption of small displacements so it ignores the dominant membrane stiffness which only appears after the plate begins to deflect.

## 11.6 SaggingCable.liml

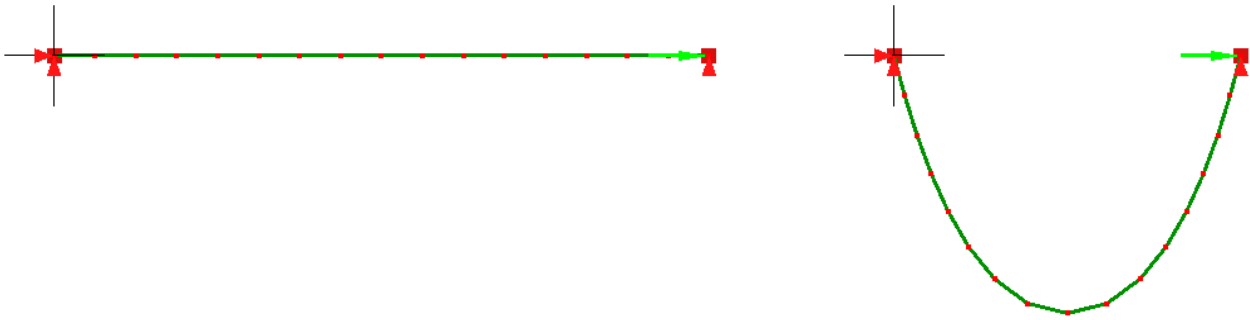
Analysis type: Nonlinear Static

Elements: Truss (line2)

Loads and constraints: Displacement, Force, Gravity

A flexible cable or rope is suspended between a fixed point and a horizontally sliding point. A force is applied at the sliding point to keep the cable in tension. We will find the vertical displacement at the center.

Because the untensioned cable has no bending stiffness and cannot support any lateral load, the tensile force is applied first. This gives it stiffness to support the gravity load which is applied in the 2<sup>nd</sup> time step.



At time=1s the cable is tensioned (left). at time=2s gravity is applied causing it to sag (right).  
The deformed shape is a catenary, which, with the vertex at the origin has the formula

$$y = a \cosh(x/a) - a$$

$$a = F / (\rho A g)$$

Tensile force  $F = 2000\text{N}$

Density  $\rho = 8000\text{kg/m}^3$

Cross sectional area  $A = 0.01\text{m}^2$

Gravity  $g = 9.8\text{N/kg}$

The solution shows that half the horizontal length of the deformed cable is  $x = 3.63792\text{m}$ . Using this value, the formula gives  $y = 3.0645\text{m}$  which is within 0.1% of distance in the solution (3.0680m).

## 11.7 BucklingBeam.liml

Analysis type: Buckling

Elements: Beam (line2)

Loads and constraints: Fixed Support, Force

Consider a column fixed at the base and free at the top, subject to an axial compressive force. We will find the value of the force at which it buckles, the critical load.

The column is of length  $L=100\text{mm}$ , the cross section is a  $10\text{mm}\times 10\text{mm}$  square and the Young's modulus is  $E=200\text{ GPa}$ .

The model is made with line2 elements. The bottom face has a fixed support to constrain displacement and rotation. A point load of  $1\text{N}$  has been applied at the top.

The Euler buckling equation is used to calculate the critical load:

$$P_{cr} = \pi^2 E I / (K L)^2$$

$K = 2$  for these boundary conditions

$E = 2 \times 10^5 \text{ N/m}^2$

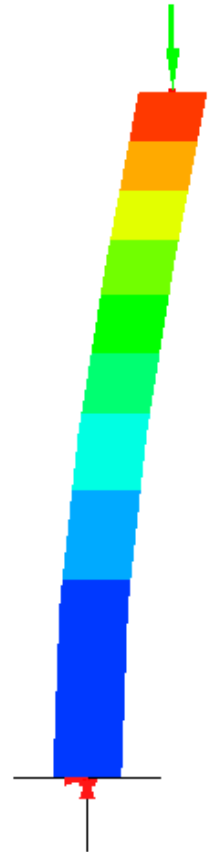
$I = 833.333 \text{ mm}^4$

$L = 100 \text{ mm}$

Critical load,  $P_{cr} = 41123 \text{ N}$

The buckling factor for mode 1 found by Mecway is 41125 and approaches the theoretical value with mesh refinement. This is equal to the buckling load in newtons because a force of  $1\text{N}$  was applied in the model. The mode shape shows that this is indeed the first buckling mode.

The shift point is set to 1000 which is typically reasonable being an order of magnitude below the lowest buckling factor.



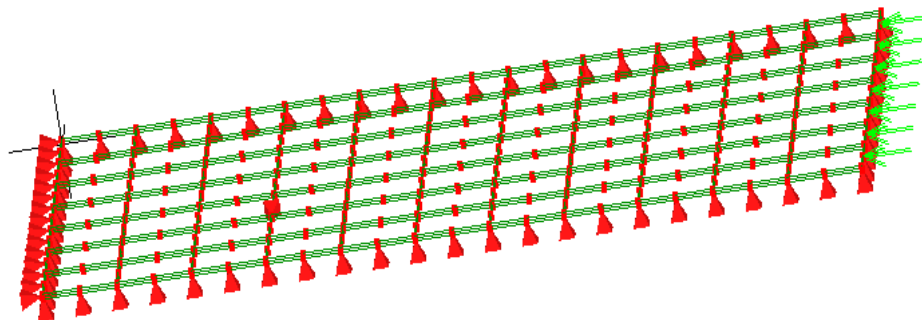
## 11.8 BucklingPlate.liml

Analysis type: Buckling

Elements: Solid continuum (hex20)

Loads and constraints: Displacement, Pressure

Consider a simply supported plate with compressive edge loading on two opposite edges. We will determine the critical loads of lowest few buckling modes.



The translational motions along the  $Z$  axis are constrained for all the edges, but they are free to rotate about the edge directions. Translational motion in the  $X$ - $Y$  plane is allowed, except the bottom edge is

restrained to provide a reaction to the applied force. A single arbitrarily chosen node is constrained in the x direction to prevent rigid body motion.

The total applied load is equal to 1, so the buckling factors produced by the solver are equal to the critical loads.

For thin plates such as the one used above, the following analytical equation can be used to calculate the first four critical loads:

$$F_{cr} = b\pi^2 a^2 D ([m/a]^2 + [n/b]^2) / m^2$$

$F_{cr}$ : The critical load applied to one edge

$a=2$ : Length in direction of load

$b=1$ : Width

$D=E t^3 / [12 (1-\nu^2)]$  : Plate rigidity

$t=0.0125$  : Thickness

$\nu=0.3$ : Poisson's ratio

$E=1e6$ : Young's modulus

$m$ : Number of half-waves of the buckle in the direction of the load

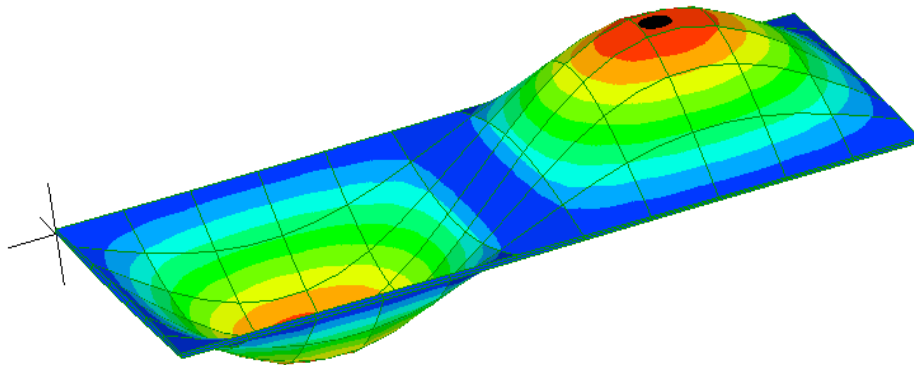
$n$ : Number of half-waves of the buckle perpendicular to the load

For  $m=1$  and  $n=1$ ,  $F_{cr}=11.0$ . The Mecway model gives 11.1

For  $m=2$  and  $n=1$ ,  $F_{cr}=7.06$ . The Mecway model gives 7.14 (pictured below)

For  $m=3$  and  $n=1$ ,  $F_{cr}=8.29$ . The Mecway model gives 8.60

For  $m=4$  and  $n=1$ ,  $F_{cr}=11.0$ . The Mecway model gives 12.0



Notice that the lowest critical load is for the mode with 2 half-waves so this mode will develop first under increasing load.

## 11.9 BucklingPlateNonlinear.liml

Analysis type: Nonlinear Static

Elements: Solid continuum (hex20)

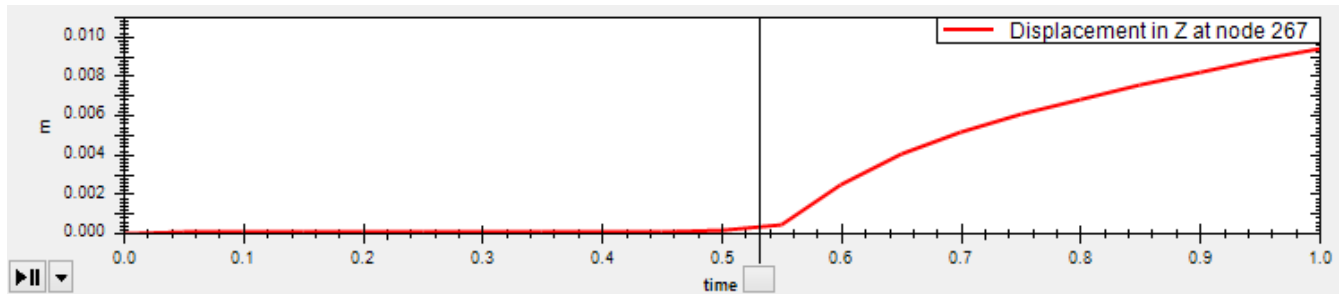
Loads and constraints: Displacement, Force

This is the same case as BucklingPlate.liml above but is solved using nonlinear static analysis instead of linear buckling analysis.

The applied load is ramped up from 6N to 8N which includes the expected critical load for the 1<sup>st</sup> buckling mode. If you didn't know the critical load in advance, you would ramp up the load from zero.

A perfectly symmetric model won't buckle in nonlinear analysis so we must apply an imperfection to initiate it. A small out of plane force of 0.0001N is applied at an arbitrary location that's not on a nodal line of the mode shape.

The solution for out-of-plane displacement shows that buckling is initiated at a load of 7.0-7.1N which contains the theoretical critical load of 7.06N. In this graph, time=0 corresponds to a load of 6N and time=1 corresponds to 8N.



## 11.10 PipeClip.liml

Analysis type: Nonlinear Static 3D

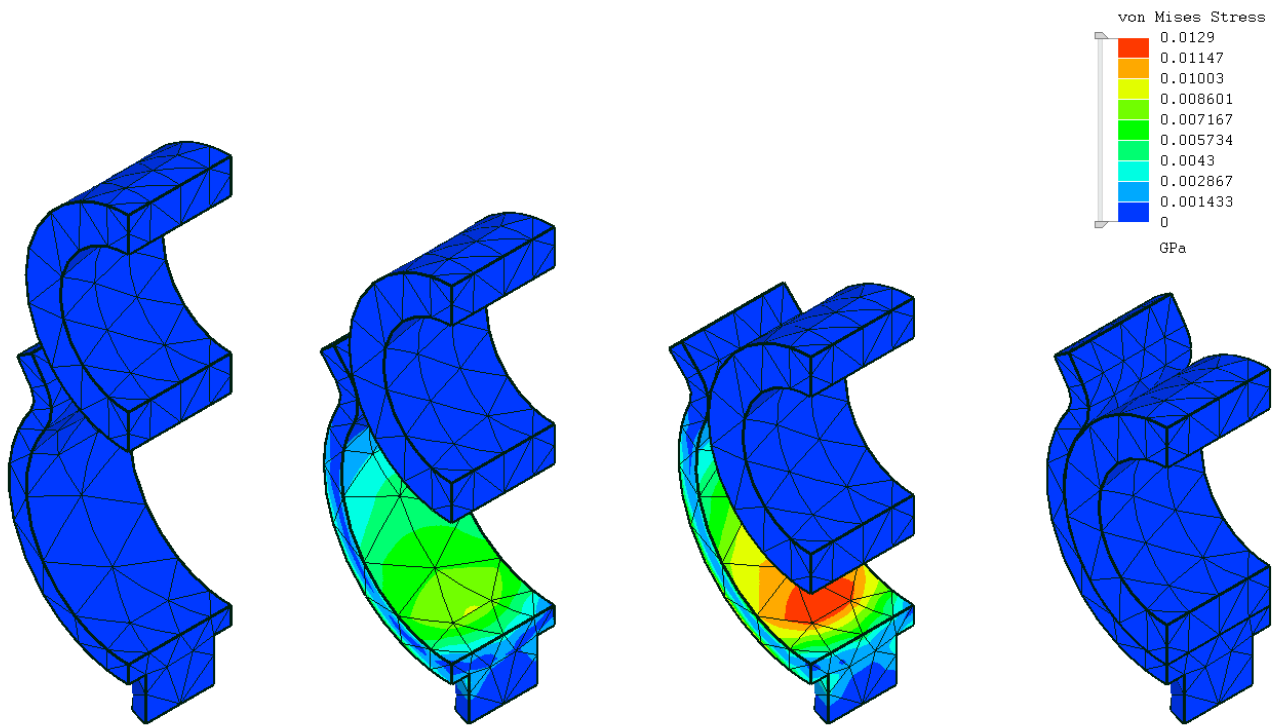
Elements: Solid continuum (tet10)

Loads and constraints: Fixed Support, Frictionless Support, Displacement, Contact

The sample shows non-linear contact between two different parts - a pipe and a clip using the CCX solver.

The two parts are defined by CAD geometry in two .step files clip.step and pipe.step which are only needed to rebuild the mesh.

Boundary conditions include frictionless support to enforce mirror symmetry, fixed support at the base of the clip where it would be mounted to another part, and contact between the outside of the pipe and the inside of the clip. The pipe is moved into the clip using a time-dependent constraint defined by the displacement function  $u_y = -12t$  mm/s.



The solution shows maximum stress when the clip is fully opened by the pipe, as expected.

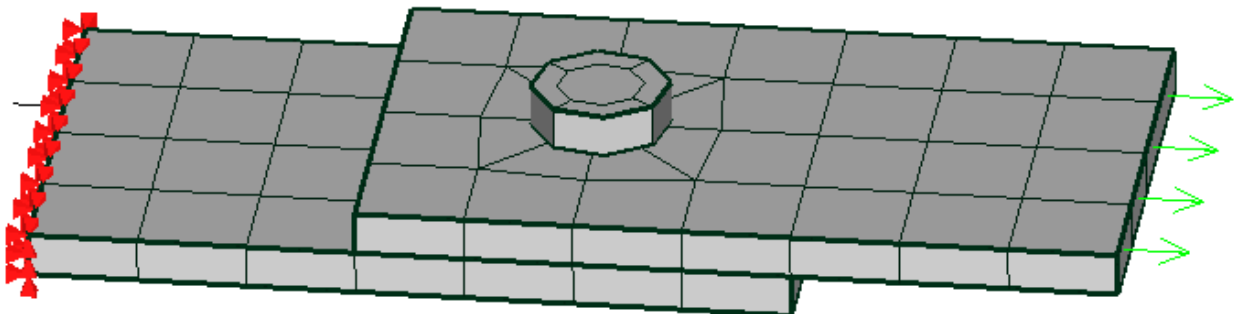
## 11.11 BoltedJoint.liml

Analysis type: Nonlinear Static 3D

Elements: Solid continuum (hex20)

Loads and constraints: Fixed Support, Bonded Contact, Contact, Pre-tension Section, Force

This sample shows two plates fixed together by a preloaded bolt with friction between the plates and an applied force that is increased until the connection fails.



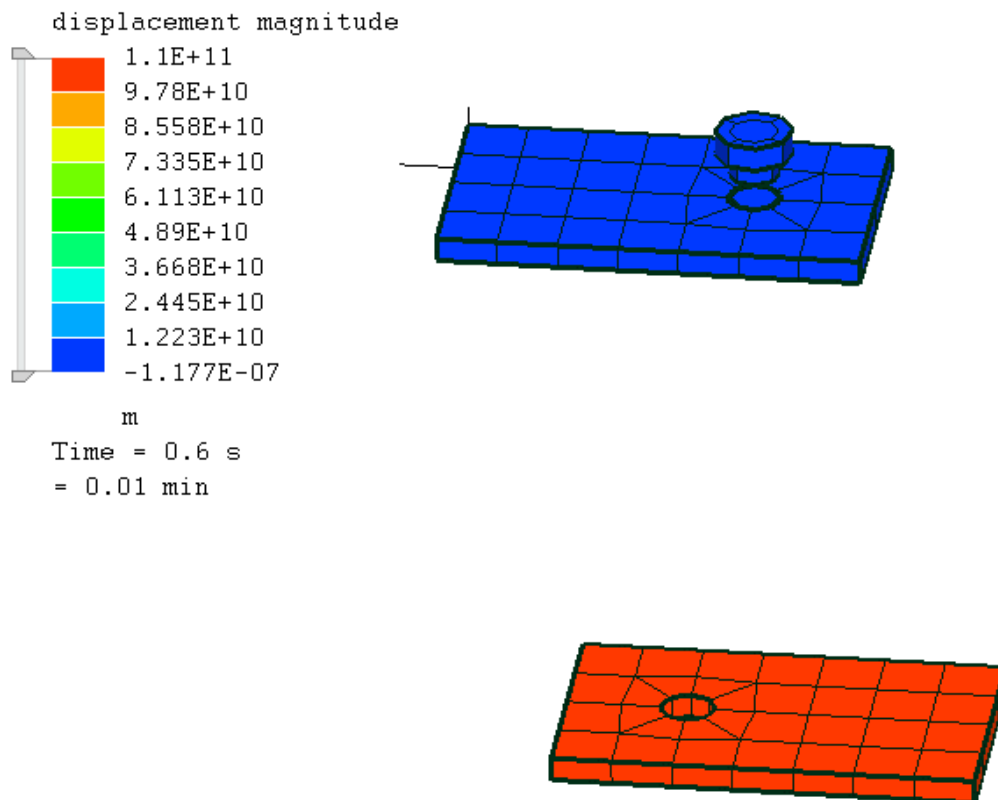
The bolt head is attached to one plate with bonded contact to make a robust connection. You may be able to use contact instead to allow it to slide or separate. The other surfaces are connected by contact so they can separate.

The preload force on the bolt is a constant 10 kN and the friction coefficient between the plates is 0.5 so we expect the maximum load the joint can carry is  $0.5 \times 10 \text{ kN} = 5 \text{ kN}$ .

The solver fails after time = 0.5 s with the message

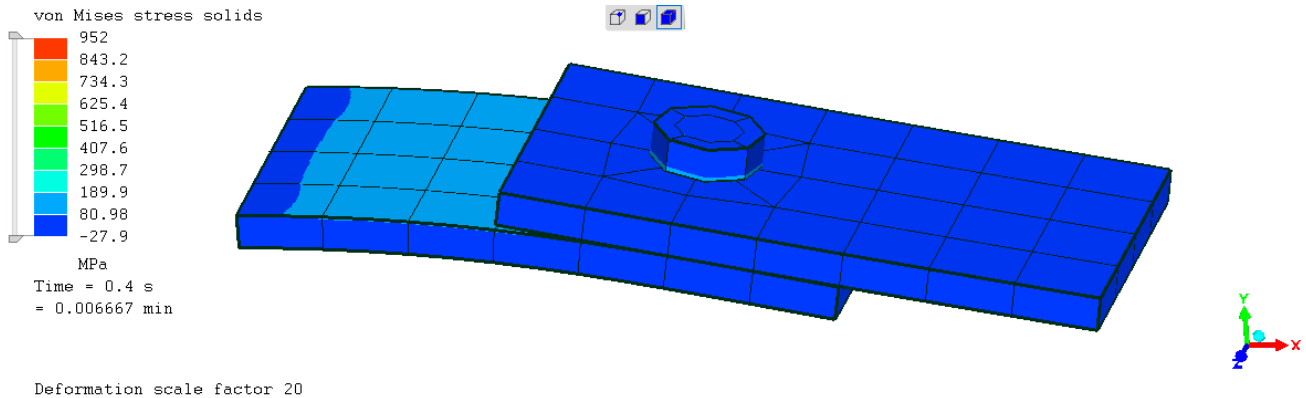
```
*ERROR: solution seems to diverge; please try
automatic incrementation; program stops
best solution and residuals are in the frd file
```

0.5 s corresponds to an applied force of 5 kN as expected. We can get some insight into the reason for the failure by looking at the final unconverged deformed shape. Set the timeline slider to 0.6 s and notice that displacement magnitude of the loaded plate is very large, in this case, on the order of  $10^{11}$  m. Set **Solution** → **Deformed view settings** → **Scale factor** to a reasonable scale of about 10% of the inverse of that, which is  $1\text{e-}12$  in this case. We can now see that there was rigid body motion of the loaded plate, indicating that it broke free of the friction as expected.



Deformation scale factor 1E-12

We can also see the deflection at lower loads such as  $t = 0.4$  s with a deformation scale factor of 20.



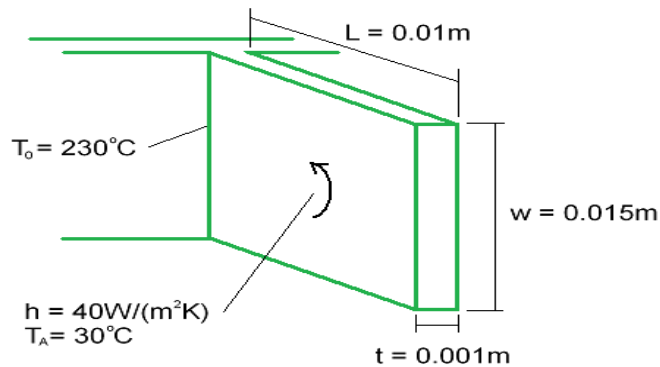
## 11.12 FinConvection.liml

Analysis type: Thermal Steady State

Elements: Fin (line2, line3)

Loads and constraints: Temperature, Convection

Mecway will find the temperature distribution along the length of the thermal fin shown below. Heat enters the base of the fin which is held at a fixed temperature of  $230^{\circ}\text{C}$  and is lost through the 4 sides by convection. The material has thermal conductivity of  $k=380\text{W}/(\text{mK})$ . The model contains two alternative meshes for the same fin, one using a line3 element and the other using two line2 elements.



The temperature ( $T_x$ ) a distance  $x$  from the base of the fin is given by

$$\frac{T_x - T_a}{T_0 - T_a} = \frac{\cosh(mL - mx)}{\cosh(mL)}$$

where  $m = \sqrt{\frac{hP}{kA}}$

$P$  is the perimeter of the cross-section,  $0.032\text{m}$

$A$  is the area of the cross-section,  $0.000015\text{m}^2$

The table below shows close agreement between Mecway's results and the temperatures given by the above formula.

Location	$T_x$ from the formula	2×line2 elements	1×line3 element
Midpoint ( $x=0.005\text{m}$ )	228.3306°C	228.3299°C	228.3306°C
End ( $x=0.01\text{m}$ )	227.7752°C	227.7742°C	228.7752°C

### 11.13 ConductionConvectionRadiation.liml

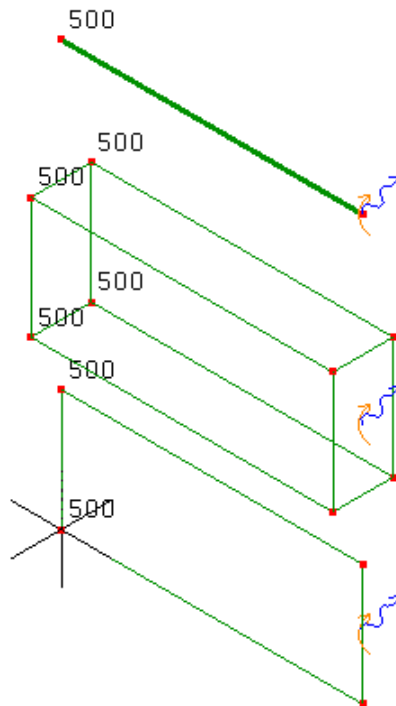
Analysis type: Thermal Steady State

Elements: Shell (quad4), Solid continuum (hex8), fin (line2)

Loads and constraints: Temperature, Convection, Radiation

This model shows one-dimensional steady state heat transfer by conduction, convection and radiation through a slab of thickness 0.05 m. Heat flows in through the left hand surface which is held at 500 K and heat flows out through the right hand surface by both convection (heat transfer coefficient  $h = 37.5 \text{ W/(m}^2\text{K)}$ ) and radiation (emissivity  $\epsilon = 0.8$ ) to an ambient temperature of 300 K.

Three versions of the model are included in the same file, using shell, solid and fin elements. All three give the same results except heat flux is not available for the fin element.



To verify the analysis we will perform a simple heat balance:

$$q_{\text{conduction}} = q_{\text{convection}} + q_{\text{radiation}}$$

Heat flux by conduction ( $q_{\text{conduction}}$ ) is given by Mecway as the x component of heat flux which is the same over the entire model

$$q_{\text{conduction}} = 3591.65 \text{ W/m}^2$$

Heat flux by convection is found using the temperature Mecway calculated for the right hand face ( $T_{\text{surface}}=380.2783\text{K}$ )

$$q_{\text{convection}} = h(T_{\text{surface}} - T_{\text{ambient}}) = 3010.44 \text{ W/m}^2$$

Heat flux by radiation is found in a similar way but using the Stefan-Boltzmann law

$$q_{\text{radiation}} = \epsilon \sigma (T_{\text{surface}}^4 - T_{\text{ambient}}^4) = 581.21 \text{ W/m}^2$$

where  $\sigma$  is the Stefan-Boltzmann constant,  $5.67037 \times 10^{-8} \text{ Wm}^{-2}\text{K}^{-4}$

The heat balance equation is satisfied:

$$3591.65 \text{ W/m}^2 = 3010.44 \text{ W/m}^2 + 581.21 \text{ W/m}^2$$

## 11.14 OscillatingHeatFlow.liml

Analysis type: Thermal Transient

Elements: Shell (quad8)

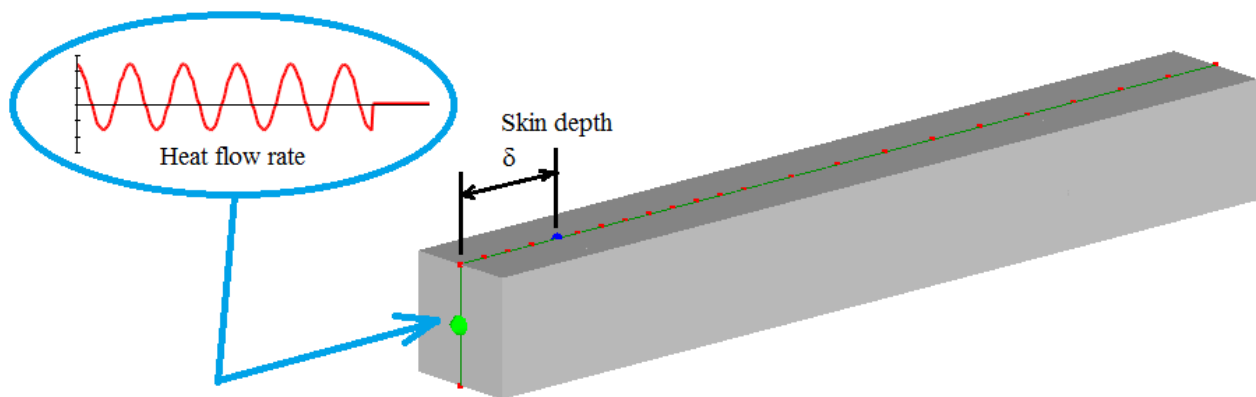
Loads and constraints: Heat flow rate

An insulated aluminum bar is heated by an oscillating thermal load applied to one end then removed. Mecway will find the temperature in the bar as a function of both time and distance along the bar. The material properties are:

Specific heat capacity,  $c_p = 900 \text{ K/(kg.K)}$

Density,  $\rho = 2700 \text{ kg/m}^3$

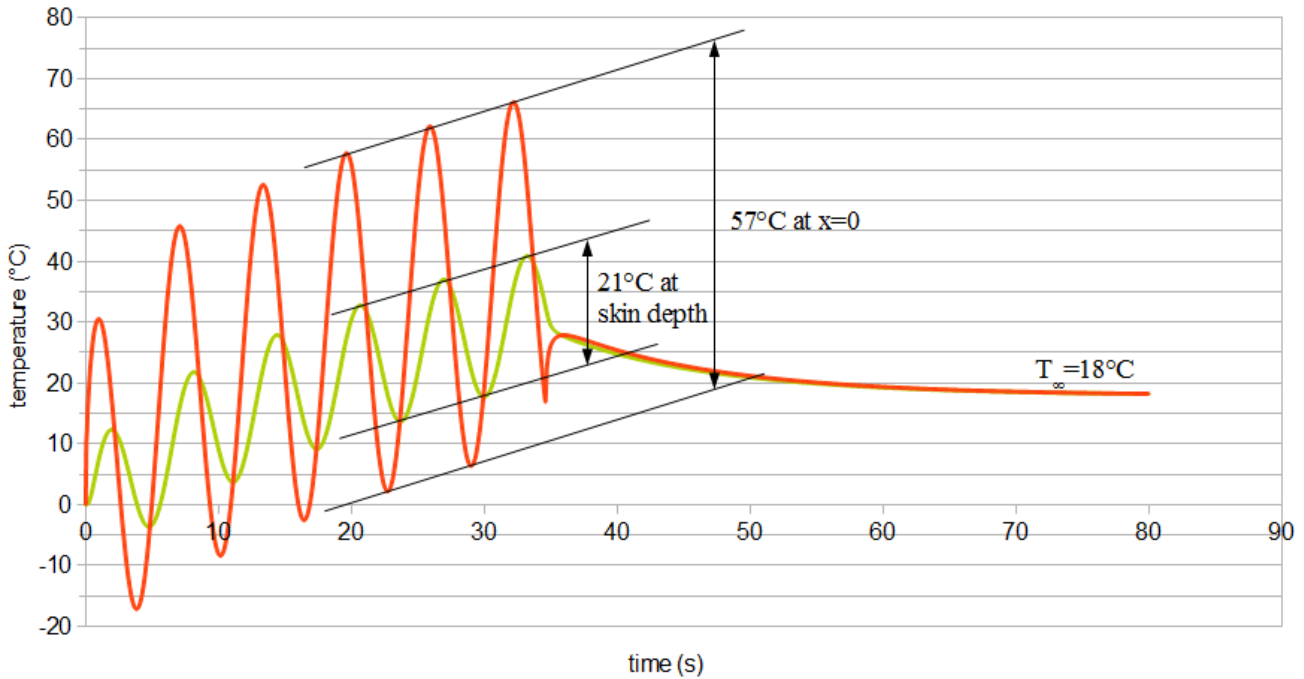
Thermal conductivity,  $k = 200 \text{ W/(m.K)}$



The heat load  $Q(t)$  is input into Mecway as a table of heat flow rate values every 0.2s according to this function:

$$Q(t) = \begin{cases} (100\cos(t) + 20)W & t \leq 34.6s \\ 0 & t > 34.6s \end{cases}$$

The graph below shows the temperature variation with time at two points. The red curve is measured at the end of the bar ( $x=0$ ) and the green curve is temperature measured at the skin depth ( $x=\delta$ ).



These results are verified in two ways, first by comparing the ratio of the temperature wave amplitudes at two points to their theoretical ratio, and second by comparing the net heat input to the total temperature rise after it has reached equilibrium.

The skin depth  $\delta$  is the distance at which the amplitude of the temperature variation is  $1/e$  of the amplitude at  $x=0$ . It is given by

$$\delta = \sqrt{\frac{kP}{\rho c_p \pi}}$$

where  $P$  is the period of the temperature wave.  $P=2\pi$

$$\delta = 0.01283001m$$

A mesh node is located at  $x=\delta$  for convenient measurement. The graph above shows the peak-peak amplitude of temperature at this node is  $21^\circ\text{C}$ . It also shows the peak-peak amplitude of temperature at  $x=0$  is  $57^\circ\text{C}$ .  $57/20 = 2.85 \approx e$  as required.

The expected temperature rise from  $t=0$  to  $t=\infty$  is

$$\Delta T = \frac{Q_{net}}{\rho c_p V}$$

$V$  = volume of the bar =  $1.5625 \times 10^{-5} \text{m}^3$

$$\begin{aligned}
 Q_{net} &= \int_0^{34.6} Q(t) dt \\
 &= \int_0^{34.6} (100\cos(t) + 20) W dt \\
 &= [100\sin(t) + 20t]_a^b J \\
 &= 687.75 J
 \end{aligned}$$

$$\Delta T = 18^\circ C$$

This agrees with the final value of temperature given by Mecway across the whole bar which is also  $18^\circ C$ .

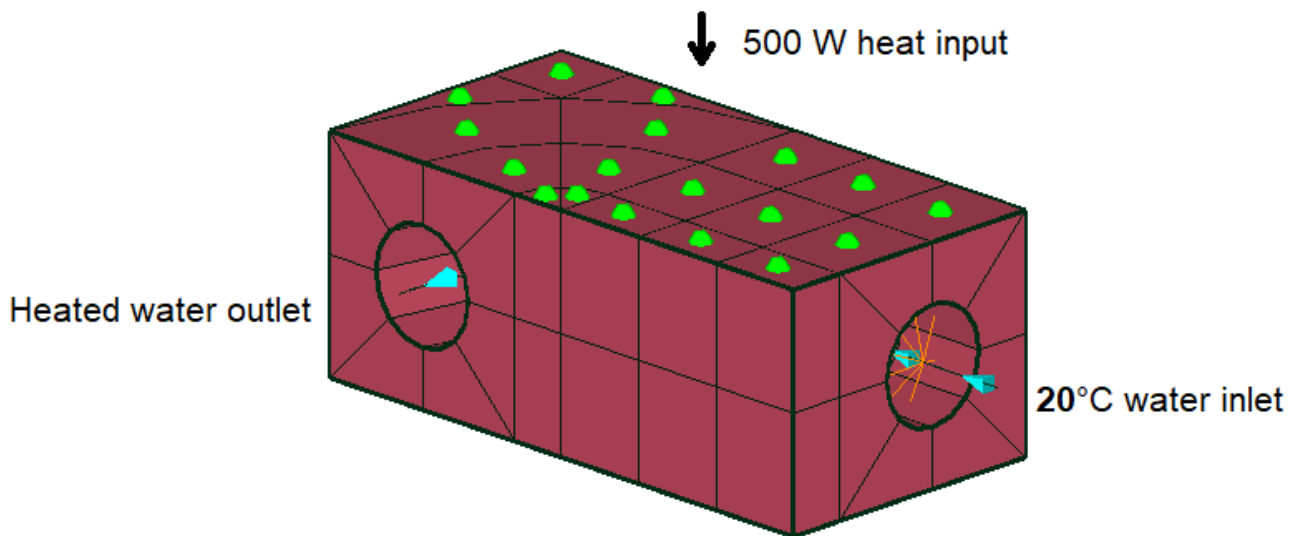
### 11.15 CoolantChannel.liml

Analysis type: Thermal Steady State

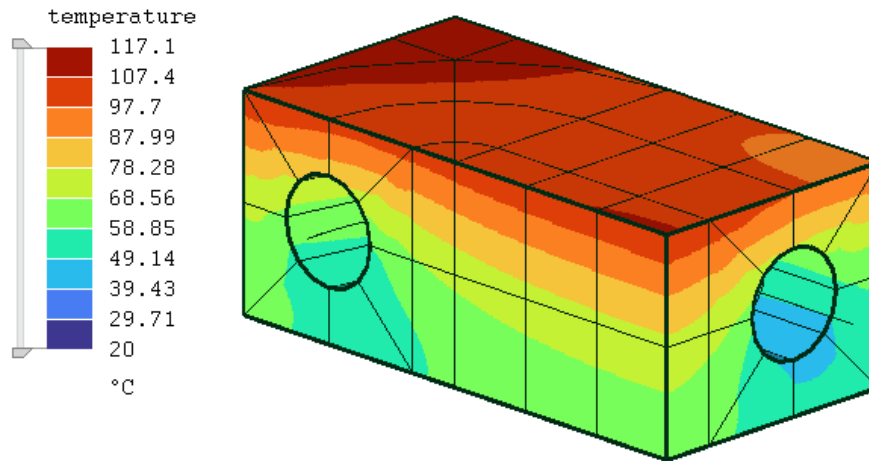
Elements: Fluid network element (line3), Solid continuum (wedge15, hex20)

Loads and constraints: Temperature, Forced convection, Mass flow rate, Heat flow rate

A heat source is being cooled by water pumped through a channel in a casting. We will find the temperature distribution within the casting and confirm that the heat is carried away by the water at the same rate that it is generated by the heat source.



After solving, we can see that the water channel cools the casting.



The heat flow rate carried away by the water is

$$\dot{Q} = \dot{m} c_p \Delta T$$

Mass flow rate,  $\dot{m}$ , is specified as 0.04 kg/s.

Specific heat at constant pressure for water,  $c_p = 4180 \text{ J/kg/K}$

Temperature difference between the inlet and outlet,  $\Delta T = 22.990^\circ\text{C} - 20.000^\circ\text{C} = 2.990 \text{ K}$

So  $\dot{Q} = 499.9 \text{ W}$  which closely matches the specified input heat flow rate of 500.0 W.

## 11.16 VibratingFreePlate.liml

Analysis type: Frequency

Elements: Shell (quad8)

Loads and constraints: none

This sample shows the undamped natural vibration modes of a square plate without any supports. Its properties are:

Side length,  $a = 0.1\text{m}$

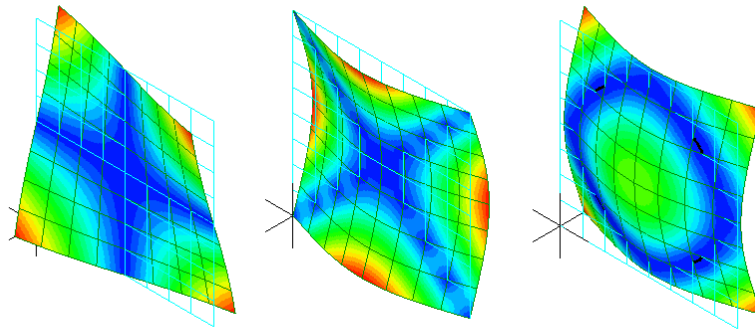
Thickness,  $h = 1\text{mm}$

Young's modulus  $E = 200 \text{ GPa}$

Poisson's ratio,  $\nu = 0.225$

Density,  $\rho = 8000 \text{ kg/m}^3$

The first three deformation modes found by Mecway are shown below along with their angular frequencies. Mecway also produces 6 rigid body modes because the plate is unconstrained.



$$\omega_1 = 2114 \text{ rad/s}$$

$$\omega_2 = 3122 \text{ rad/s}$$

$$\omega_3 = 3662 \text{ rad/s}$$

These results are verified by comparison with analytical formulas from (William Weaver, Jr., Stephen P. Timoshenko, Donovan H. Young, *Vibration Problems in Engineering*, 1990):

$$\omega = \frac{\alpha}{a^2} \sqrt{\frac{D}{\rho h}}$$

$$D = \frac{Eh^3}{12(1-\nu^2)}$$

$$\alpha_1 = 14.10, \omega_1 = 2089 \text{ rad/s}$$

$$\alpha_2 = 20.56, \omega_2 = 3046 \text{ rad/s}$$

$$\alpha_3 = 23.91, \omega_3 = 3542 \text{ rad/s}$$

For all three modes the error is less than 4% and can be reduced to less than 1% with mesh refinement.

## 11.17 VibratingCantileverBeam.liml

Analysis type: Frequency

Elements: Beam (line2)

Loads and constraints: Fixed support

Beam elements are used to model a vibrating cantilever fixed at one end and free at the other. At least four elements are needed to capture the higher bending modes. The beam's properties are:

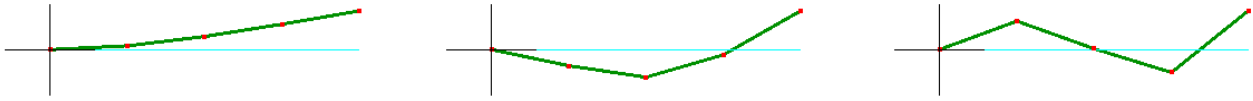
Length,  $L = 2.4\text{m}$

Width,  $b = 0.1\text{m}$

Height,  $h = 0.05\text{m}$

Young's modulus,  $E = 200 \text{ GPa}$

Density,  $\rho = 7860 \text{ kg/m}^3$



The frequencies of the first 3 modes for bending in the XY plane are verified by the formula

$$f_n = \frac{\alpha_n^2}{2\pi} \sqrt{\frac{EI}{\rho b h L^4}}$$

$$\alpha_1 = 1.875$$

$$\alpha_2 = 4.694$$

$$\alpha_3 = 7.885$$

$$I = \frac{bh^3}{12} = 2.0833 \times 10^{-6} m^4$$

which gives these frequencies compared to Mecway

Mode	Frequency from formula	Frequency from Mecway
n=1	7.073 Hz	7.074 Hz
n=2	44.33 Hz	44.38 Hz
n=3	125.1 Hz	125.1 Hz

Mecway's solution is 3D so it also includes modes for bending in the Z direction which are neglected in this 2D calculation.

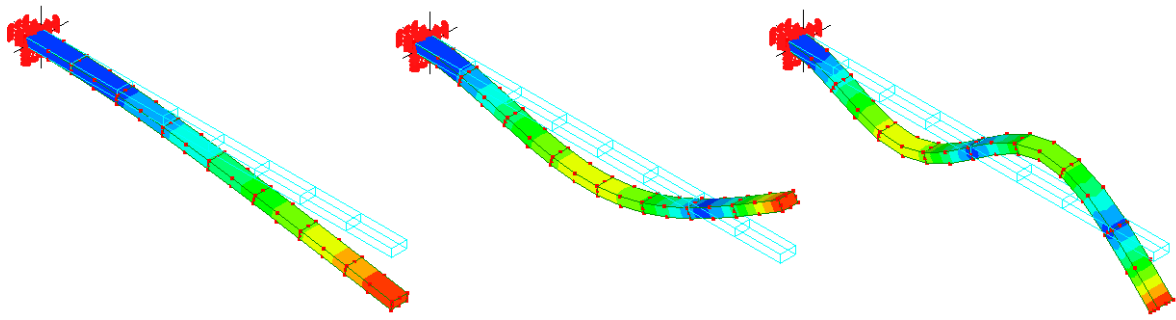
## 11.18 VibratingCantileverSolid.liml

Analysis type: Frequency

Elements: Solid continuum (hex20)

Loads and constraints: Fixed support

This is a model of the same beam described in VibratingCantileverBeam.liml but it is made using hex20 solid elements. The picture below shows the same mode shapes that are shown for VibratingCantileverBeam.liml.



Mode	Frequency from formula	Frequency from Mecway
n=1	7.073 Hz	7.089 Hz
n=2	44.33 Hz	44.99 Hz
n=3	125.1 Hz	128.7 Hz

## 11.19 VibratingTrussTower.liml

Analysis type: Frequency, Static 3D

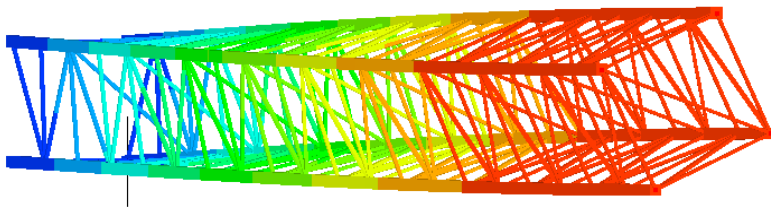
Elements: Truss (line2)

Loads and constraints: Displacement, Force

A truss tower will be analyzed to find the fundamental torsional vibration mode. It is 20m high and made from 10 identical 1m×1m×2m sections connected end to end. Each section uses two types of steel beam – the chords at the corners have 0.01m<sup>2</sup> cross sectional area and the bracing members have 0.001m<sup>2</sup> cross sections. All joints are free to rotate in any direction. The steel's Young's modulus is 200GPa and density is 7800kg/m<sup>3</sup>. The 4 nodes at the base are constrained against displacement in all directions.

The first two modes found by Mecway are bending modes and the 3rd is the fundamental torsion mode shown below with frequency

$$f_{\text{Mecway}} = 7.64 \text{ Hz.}$$



This result is validated using the formula for the fundamental torsion mode of a uniform shaft with distributed mass

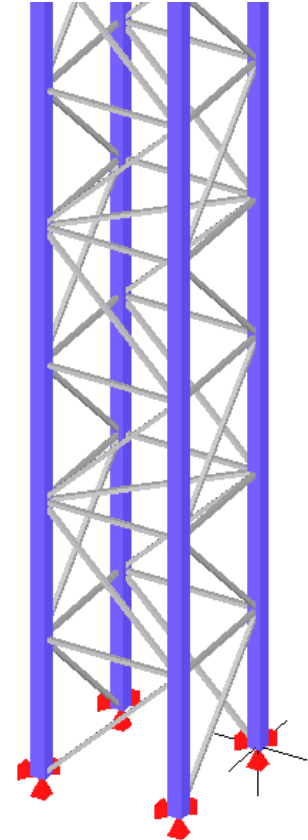
$$f = \frac{1}{4} \sqrt{\frac{K}{I}}$$

$I = 3640.54 \text{ kgm}^2$  is the rotational inertia of the whole structure about its longitudinal axis. It can be found by summing the rotational inertia of all 250 truss elements.

$K$  is the torsional stiffness of the structure. This is found by using Mecway to perform a static analysis with an applied moment of 1 Nm then measuring the resulting rotation at the end. The moment is applied using 4 forces on the top 4 nodes. If you unsuppress the group named *Moment* and change the analysis type to Static 3D then solve it, it will show the top nodes each rotate by  $\theta = 2.92843 \times 10^{-7}$  radians about the longitudinal axis of the tower. This gives  $K = 1\text{Nm} / \theta = 3.41480 \times 10^6 \text{ Nm/radian}$ .

The frequency given by this formula matches Mecway's result to within 0.3%.

$$f = 7.66 \text{ Hz}$$



## 11.20 VibratingMembrane.liml

Analysis type: Frequency

Elements: Shell (tri6)

Loads and constraints: Fixed Support, Displacement, Thermal Stress, Temperature

A hexagonal sheet of fabric is fixed around its outer edge and held in a state of uniform isotropic bi-axial tension. It can vibrate like a drum head due to the stiffness caused by its tension. We will find its first 3 natural frequencies and mode shapes.

Density: 1000kg/m<sup>3</sup>

Thickness: 1mm

Distance between opposite corners: 600mm

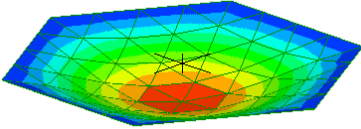
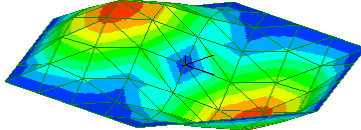
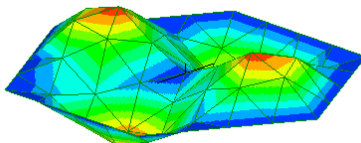
Tension per unit length: 2N/m

The uniform stress is applied using thermal strain. An artificial thermal expansion coefficient of 1K<sup>-1</sup> is used with an arbitrary small Young's modulus of 0.001N/m<sup>2</sup> and a temperature change of -2×10<sup>6</sup>K. This generates a stress of 2000N/m<sup>2</sup> which corresponds to a tension of

$$2000\text{N/m}^2 \times 1\text{mm} = 2\text{N/m}.$$

You can confirm the 2000N/m<sup>2</sup> stress by changing the analysis type to Static 3D and solving.

All elements are constrained against movement in the in-plane directions to speed up solving and prevent spurious in-plane modes caused by the very low stiffness of the material.

Mode	Mode shape	Frequency
Mode 1 0 nodal diameters		2.007 Hz
Modes 2 and 3 1 nodal diameter		3.197 Hz
Modes 4 and 5 2 nodal diameters		4.282 Hz

## 11.21 Impeller.liml

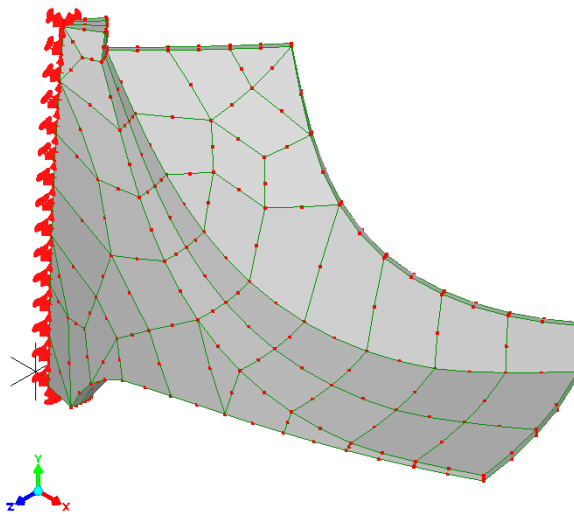
Analysis type: Frequency

Elements: Solid continuum (wedge15, hex20)

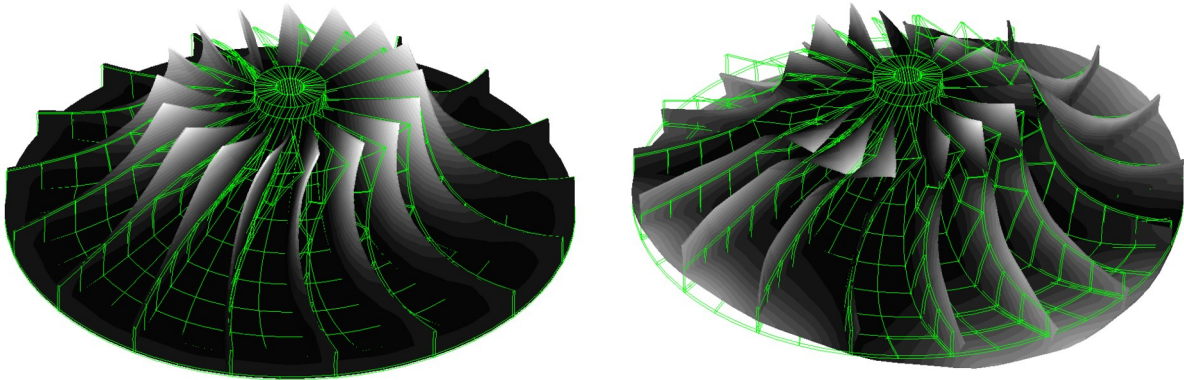
Loads and constraints: Fixed Support, Centrifugal Force, Cyclic Symmetry

Vibration modes of a 17-blade, 100,000 rpm impeller are found. The frequencies of the modes depend on the speed of rotation because centrifugal force alters the stiffness through the effects of stress stiffening and spin softening.

The structure is composed of 17 identical segments so only one segment is modeled by taking advantage of cyclic symmetry. Since it uses the internal solver, the node patterns on both surfaces of the segment are the same. The CCX solver doesn't have this limitation.

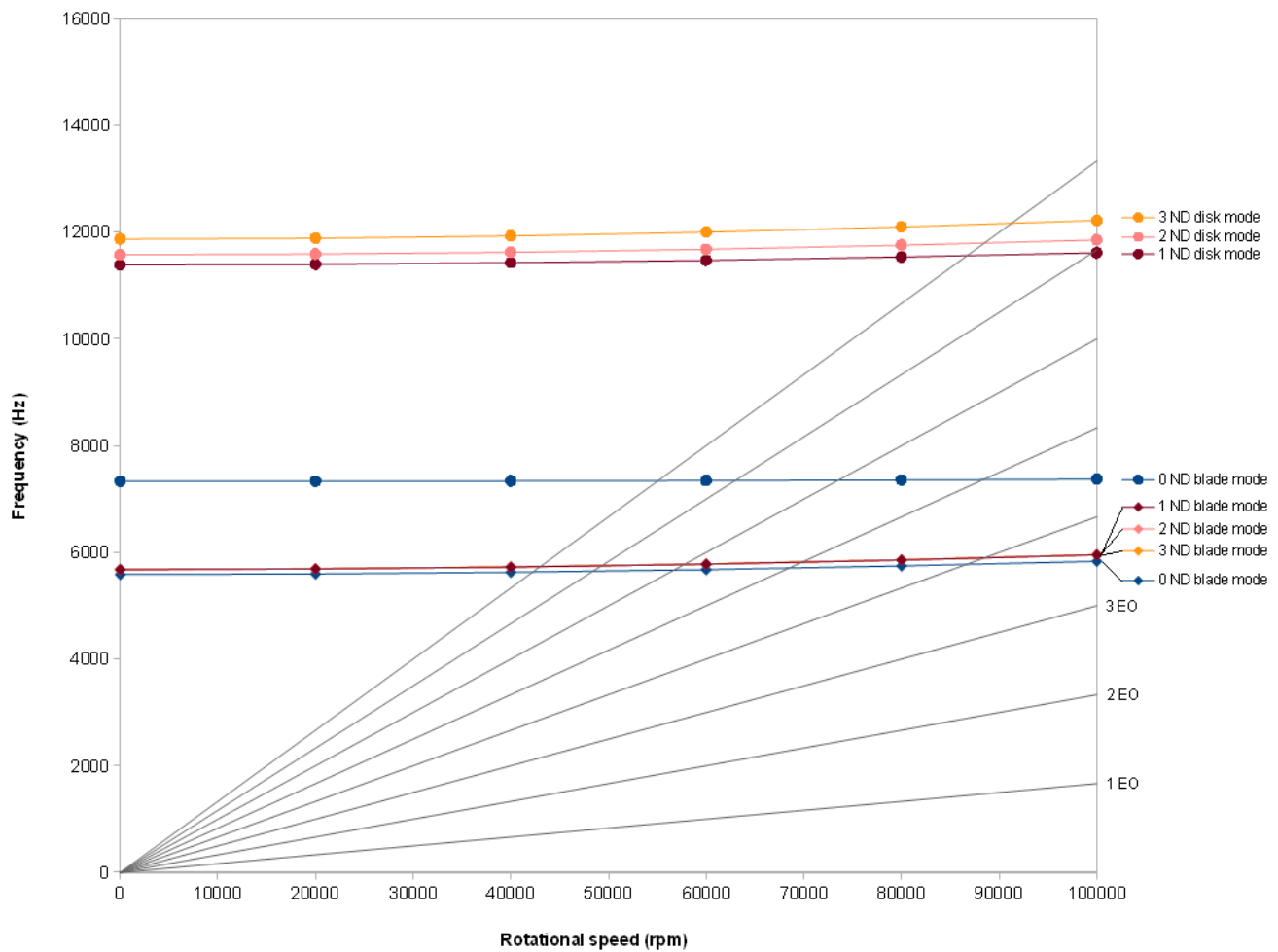


The material is steel ( $E=210\text{GPa}$ ,  $\nu=0.3$ ,  $\rho=7800\text{kg/m}^3$ ) and the inside surface is constrained in the rotating reference frame to represent a rigid shaft. Results may be inaccurate for a flexible shaft due to the Coriolis effect. Two mode shapes are shown below.



*The 1<sup>st</sup> blade mode. Deformation of each A disk mode with 2 nodal diameters segment is identical with 0 nodal diameters.*

The Campbell diagram below shows the first 2 modes for each of 0, 1, 2 and 3 nodal diameters (ND). It was made by solving the model 6 times, at 6 different rotational speeds.



## 11.22 VibratingString.liml

Analysis type: Frequency

Elements: Solid continuum (hex8)

Loads and constraints: Displacement, Force

We will model the transverse vibration of a straight flexible cable held in tension. The cable has the following properties:

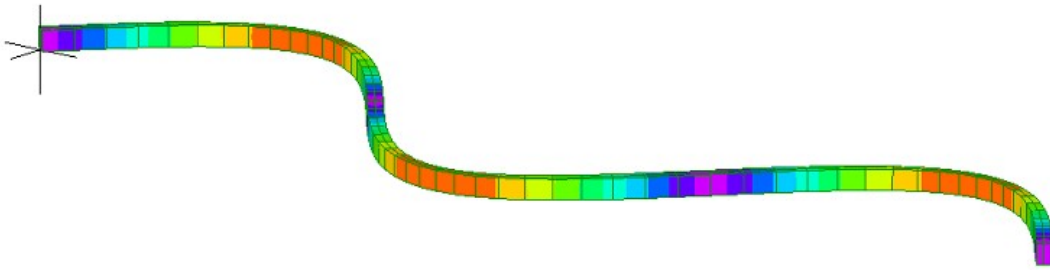
- Length:  $L=60\text{m}$
- Linear density:  $\rho=3\text{kg/m}$
- Tensile force:  $F=40\text{N}$
- Bending stiffness: Negligible

The solution is compared to hand calculation using the formula

$$f = n/2 (F/\rho)^{1/2}/L$$

where  $f$  is the frequency and  $n$  is the mode number of the standing wave. The Mecway model includes axial modes which are ignored here because they are a consequence of the arbitrary material stiffness.

Mode	$f$ (hand calculation)	$f$ (Mecway)
$n=1$	0.03043 Hz	0.03042 Hz
$n=2$	0.06086 Hz	0.06088 Hz
$n=3$	0.09129 Hz	0.09141 Hz
$n=4$	0.1217 Hz	0.1221 Hz



## 11.23 PandSWaves.liml

Analysis type: Dynamic Response 2D

Elements: Plane continuum (quad4)

Loads and constraints: Displacement, Force

Dynamic response analysis is used to model wave propagation through an elastic medium (rock). A sinusoidal force is applied at the center of an infinite domain. This generates pressure waves (P-waves) traveling outward in the direction of the force, and shear waves (S-waves) traveling outward perpendicular to the force.

Only one quadrant of the domain is modeled due to symmetry. There is mirror symmetry about the X axis which is enforced by constraining displacement in Y along the axes. There is also anti-symmetry in load and displacement about the Y axis which is enforced by constraining displacement in Y along the Y axis too. Mecway has no infinite boundary condition so a finite boundary far from the source of the waves is used instead. The analysis is only run until the first waves reach the boundary so any reflections from the boundary are excluded.

The material's properties are:

Young's modulus  $E = 7 \text{ GPa}$

Poisson's ratio  $\nu = 0.2$

Density  $\rho = 3000 \text{ kg/m}^3$

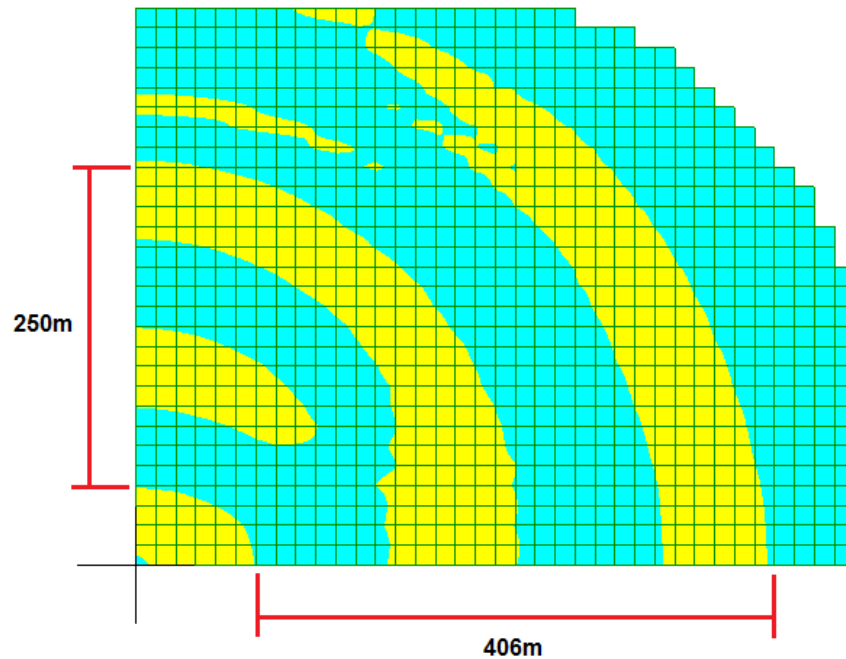
Shear modulus  $G = \frac{E}{2(1+\nu)} = 2.917 \text{ GPa}$

Bulk modulus  $K = \frac{E}{3(1-2\nu)} = 3.889 \text{ GPa}$

Displacement in X

2.221E-10  
1.527E-12  
-2.19E-10

Time = 0.3  
= 0.005 × 60



The contour plot above at time = 0.3s shows displacement in the X direction using only 2 colors (**Tools** → **Options** → **Contour plot**) so the wavelengths can be easily read off it. The tape measure tool is used to measure the average wavelength of two P-waves along the X axis as  $406\text{m}/2 = 203\text{m}$ . In the same way, the average wavelength of 2 S-waves measured along the Y axis is  $250\text{m}/2 = 125\text{m}$ .

The solution is verified by comparing the wavelengths of the P-waves and S-waves to their theoretical values.

The theoretical wavelengths are

$$\lambda_p = \frac{c_p}{f} \quad \text{and} \quad \lambda_s = \frac{c_s}{f}$$

where

$$f = \frac{50}{2\pi} \text{ Hz} \quad \text{is the frequency of the applied force}$$

$$c_p = \sqrt{\frac{\frac{4}{3}G + K}{\rho}} = 1610 \text{ m/s} \quad \text{is the speed of P-waves.}$$

$$c_s = \sqrt{\frac{G}{\rho}} = 986.0 \text{ m/s} \quad \text{is the speed of S-waves.}$$

This gives

$$\lambda_p = 202 \text{ m} \quad \text{and} \quad \lambda_s = 124 \text{ m}$$

which is within 5% of Mecway's results of 203m and 125m respectively.

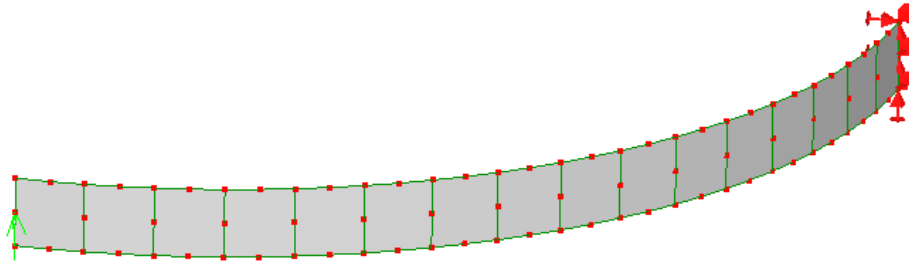
## 11.24 DampedVibratingStrip.liml

Analysis type: Dynamic Response 3D

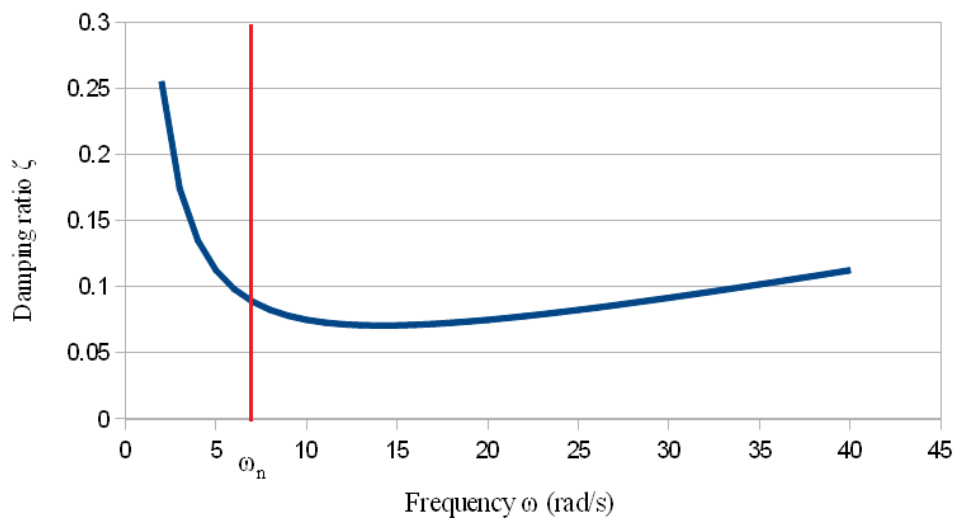
Elements: Shell (quad8)

Loads and constraints: Fixed Support, Force, Rayleigh damping

This sample shows Rayleigh damping applied to the vibration of a curved strip.



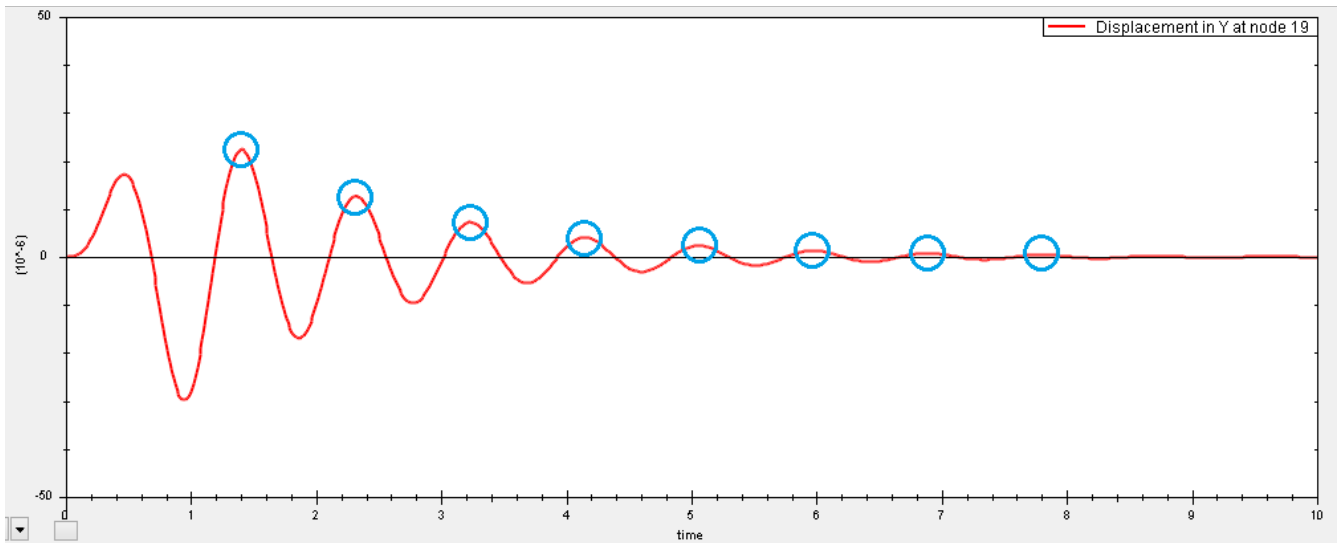
The load excites the lateral bending and twisting mode which has a natural frequency of  $\omega_n=6.91\text{rad/s}$ . The vibration is then allowed to decay by viscous damping. The Rayleigh damping constants are set to  $\alpha=1$  and  $\beta=0.005$  which give a damping ratio of  $\zeta=0.08978$  at  $\omega=\omega_n$ . Alternatively, this can be expressed as a Q factor of  $Q=\frac{1}{2\zeta}=5.569$ .



This damping ratio tells us that the amplitude of oscillation should decay with a time constant of

$$\tau = \frac{1}{\omega \zeta} = \frac{1}{6.91 \text{ rad/s} \times 0.08978} = 1.61 \text{ s}$$

Measurement of the time constant from the peaks of the displacement curve generated by Mecway also gives 1.61s.



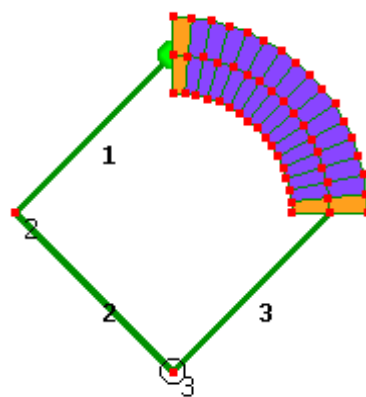
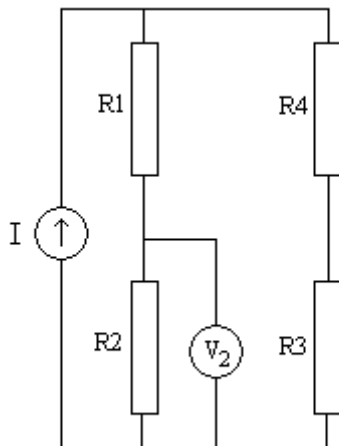
## 11.25 WheatstoneBridge.liml

Analysis type: DC Current Flow

Elements: Resistor (line2), Solid continuum (hex8)

Loads and constraints: Electric potential, Current

The Wheatstone bridge circuit shown below is modeled in Mecway to find the voltage  $V_2$  across  $R_2$  with an applied current of  $I=100\text{mA}$ . Three of the resistors are line2 resistor elements and  $R_4$  is modeled using solid elements having high-conductivity terminals to connect to the resistor elements. Node 3 is constrained to 0V and is the reference point for voltages at the other nodes.



$$R1=750\Omega$$

$$R2=1000\Omega$$

$$R3=500\Omega$$

$R4=1169.826\Omega$  which was found by integration of the curved shape.

Hand calculation shows that the voltage at node 2 is

$$V_2 = I \frac{(R1 + R2)(R3 + R4)}{R1 + R2 + R3 + R4} \frac{R2}{R1 + R2} = 48.828V$$

which matches Mecway's result of 48.823V. Most of the error is due to coarseness of the solid element mesh of R4 and can be eliminated by replacing it with a line2 resistor element having the resistance listed above.

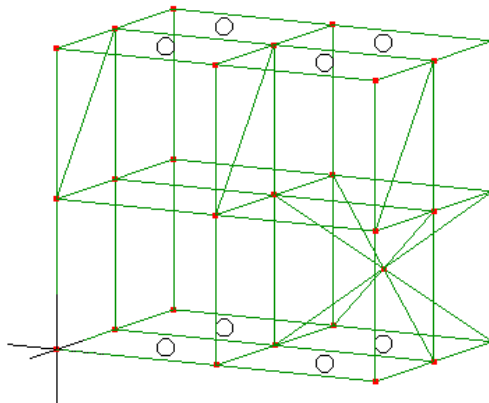
## 11.26 Capacitor.liml

Analysis type: Static 3D

Elements: Solid continuum (pyr5, wedge6, hex8)

Loads and constraints: Electric potential

The electric field between two plates of a capacitor will be modeled. The plates are  $d=0.25m$  apart with a dielectric ( $\epsilon_r=2.5$ ) in-between and a potential difference of  $V=1.5V$  across them. Each plate has an area of  $A=0.2m \times 0.3m$ .



The expected electric field is

$$E = V / d \\ = 1.5V / 0.25m = 6 \text{ V/m}$$

which is consistent with the results from Mecway of 6.000000 V/m on each node.

The expected energy stored in the capacitor is

$$U = 1/2 C V^2 \\ = 5.9766 \times 10^{-12} \text{ J}$$

where

$$C = A \epsilon_r \epsilon_0 / d$$

$\epsilon_0 = 8.854187818 \times 10^{-12} \text{ F/m}$ , the permittivity of free space

This corresponds to an energy density of  $U \times A \times d = 3.9844 \times 10^{-10} \text{ J/m}^3$ . The energy density given by Mecway is also  $3.9844 \times 10^{-10} \text{ J/m}^3$ .

Edge effects do not appear in this model because the mesh has no elements outside the area between the plates.

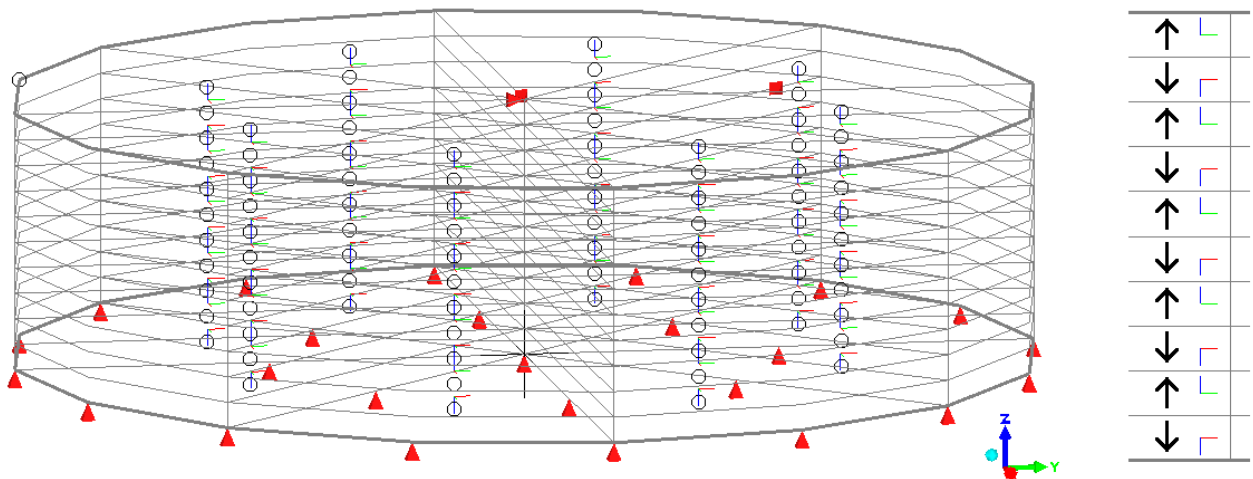
## 11.27 PiezoelectricStack.liml

Analysis type: Static 3D

Elements: Solid continuum (wedge15)

Loads and constraints: Frictionless support, Displacement, Pressure, Electric potential

A stack of 10 layers of piezoelectric ceramic disks is analyzed as both an actuator and a force sensor. The layers are oriented with alternating polarization directions of +Z and -Z. These are defined by the direction of the local W axis of each element. The base of the stack has a frictionless support to support the load on the sensor and keep the behavior 1-dimensional so it can be easily compared to hand calculations.



*Mesh. The black arrows indicate the polarization directions.*

Material properties for PZT-5A are:

Compliance matrix at constant electric field (short circuit)

$$s^E = \begin{bmatrix} 16.4 & -5.74 & -7.22 & 0 & 0 & 0 \\ -5.74 & 16.4 & -7.22 & 0 & 0 & 0 \\ -7.22 & -7.22 & 18.8 & 0 & 0 & 0 \\ 0 & 0 & 0 & 47.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 47.5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 44.3 \end{bmatrix} \times 10^{-12} \frac{\text{m}^2}{\text{N}}$$

This shows that the material is transversely isotropic with the 3 axis being the axis of symmetry. In Mecway, this corresponds to the element's local W axis.

#### Coupling coefficient matrix d

$$\mathbf{d} = \begin{bmatrix} 0 & 0 & 0 & 0 & 584 & 0 \\ 0 & 0 & 0 & 584 & 0 & 0 \\ -171 & -171 & 374 & 0 & 0 & 0 \end{bmatrix} \times 10^{-12} \frac{\text{m}}{\text{V}}$$

Due to the transverse isotropy,  $d_{32}=d_{31}$  and  $d_{24}=d_{15}$  so some sources of material data may only include the 3 unique values  $d_{31}$ ,  $d_{33}$ , and  $d_{15}$ .

#### Permittivity matrix at constant stress (clamped)

$$\boldsymbol{\epsilon}^T = \begin{bmatrix} 1730 & 0 & 0 \\ 0 & 1730 & 0 \\ 0 & 0 & 1700 \end{bmatrix} \times \epsilon_0$$

#### **Actuator**

The sample file is configured to act as an actuator driven by a voltage source by having two sets of electric potential constraints on the faces between the layers. The strain is given by the actuator equation:

$$\mathbf{S} = \mathbf{s}^E \mathbf{T} + \mathbf{d} \mathbf{E}$$

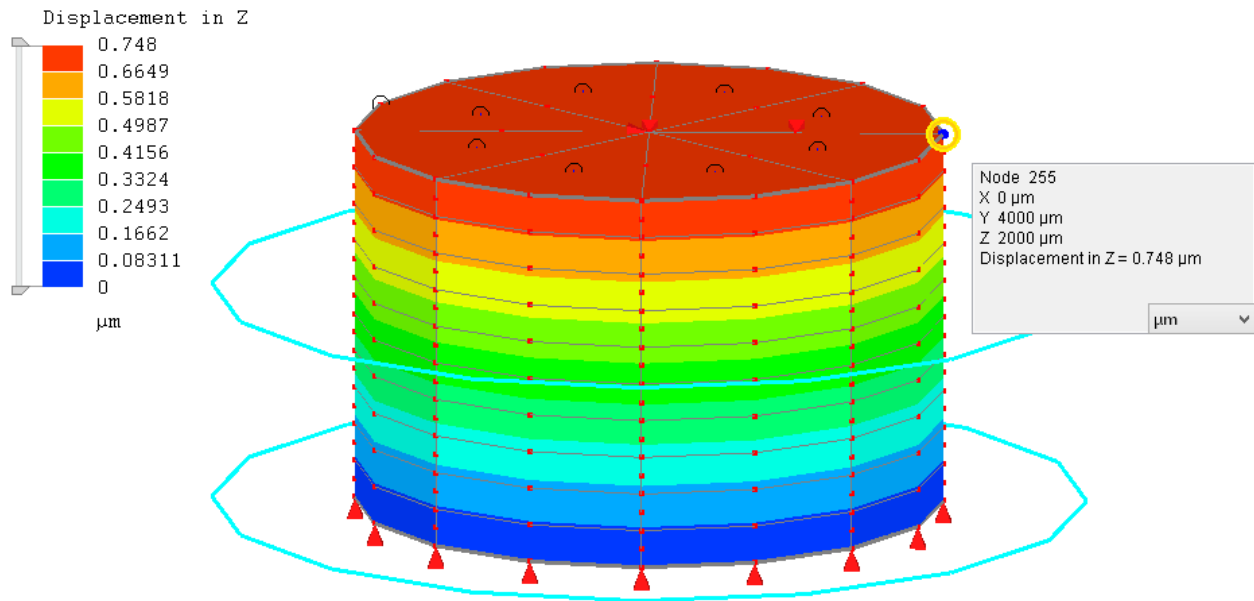
For 1D behavior, this becomes a scalar equation with

$$\text{Electric field } \mathbf{E} = E_w = 200 \text{ V} / 200 \text{ } \mu\text{m} = 10^6 \text{ V/m}$$

$$\text{Coupling coefficient } \mathbf{d} = d_{33} = 3.74 \times 10^{-10} \text{ m/V}$$

Stress  $\mathbf{T} = 0$  because there are no mechanical loads or constraints.

This gives strain  $S = 0.000374$ . The total length of the stack is 2 mm so its change in length should be  $2 \text{ mm} \times 0.000374 = 0.748 \text{ } \mu\text{m}$ . Mecway's solution agrees with this with Displacement in Z = 0.748  $\mu\text{m}$  at the top surface.



## Sensor

If you suppress the first two electric potential constraints and unsuppress the pressure load, the model becomes a sensor that produces an electric voltage across each layer in response to the applied pressure. The electric field is given by the sensor equation:

$$\mathbf{D} = \mathbf{d} \mathbf{T} + \boldsymbol{\epsilon}^T \mathbf{E}$$

For 1D behavior, this also becomes a scalar equation with

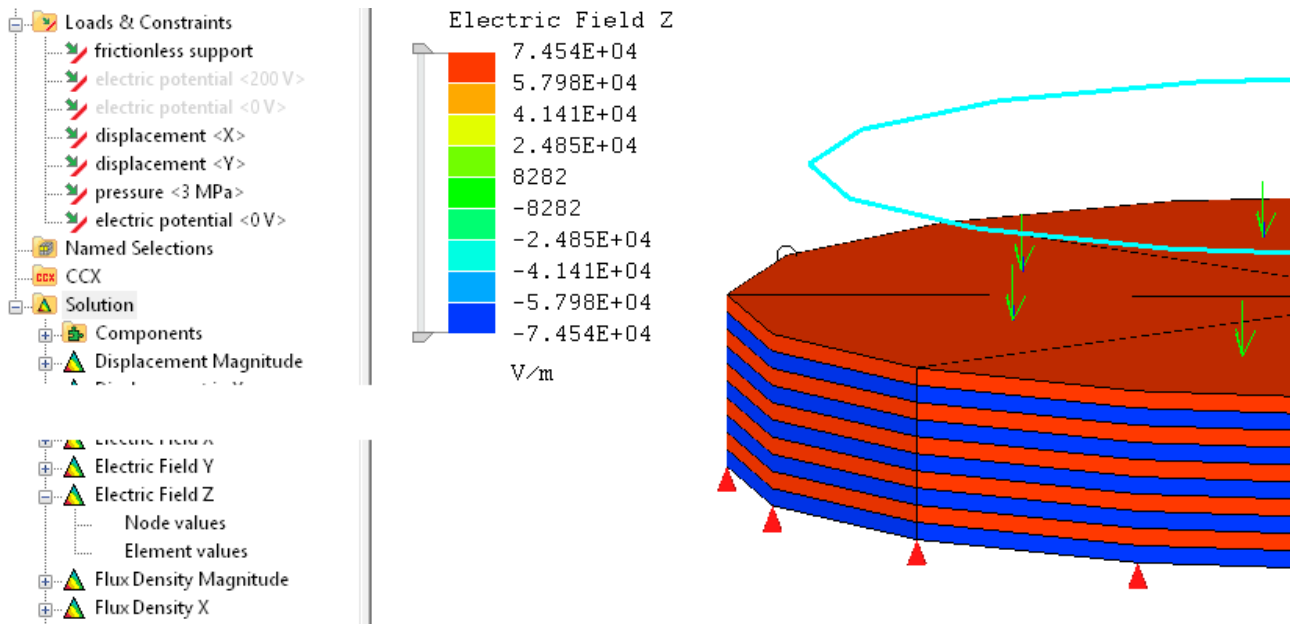
Electric displacement  $\mathbf{D} = 0$  because it's open circuit.

Coupling coefficient  $\mathbf{d} = d_{33} = 3.74 \times 10^{-10}$  m/V

Stress  $\mathbf{T} = -3$  MPa

Permittivity at constant stress  $\boldsymbol{\epsilon}^T = \boldsymbol{\epsilon}_{33}^T = 1700 \times \epsilon_0 = 1.505 \times 10^{-8}$  C/(Vm)

This gives electric field  $\mathbf{E} = 74500$  N/C (or V/m) in the 3 direction. Mecway's solution agrees with this with Electric Field Z = 74541 V/m on elements where the 3 direction is the same as the global Z axis and -74541 V/m on reversed elements.



From the Z component of the electric field, we can find the potential difference across one layer as

$$V = -E d$$

On the top layer with thickness  $d = 0.2$  mm and positive  $E$ ,  $V = -14.9$  V which agrees with Mecway's result of 14.91 V at the bottom of the layer and 0 V at the top.

## File Formats

Mecway can open and save a range of file formats to inter-operate with other software. Some of these (dxf, msh, stl, vol and xyz) do not contain physical units so you will be asked to choose the length unit when opening them.

To export the deformed mesh from a solution, use **Mesh tools** → **Transfer displacements from solution...** to deform the modeler's mesh, then export it.

### 12.1 .liml

This is the native file format for Mecway. It uses XML which is a human readable text format. A simple example of a file containing two nodes and one beam element is

```
<liml version="3">
  <node nid="1" x="0" y="0" z="0" />
  <node nid="2" x="1" y="1" z="0" />
  <elset>
    <elem eid="1" shape="line2" nodes="1 2" />
  </elset>
</liml>
```

Numbers without units are either dimensionless, such as node numbers shown above, or are in SI units, such as the node coordinates shown above which are in meters regardless of the unit used to display them in Mecway. The unit symbols are either included with after the number or, in the case of formulas, appear as separate XML attributes.

### 12.2 STEP (.step/.stp)

Mecway can open a STEP file containing a single solid body. When you open a STEP file, it appears in the **Geometry** branch of the outline tree. You can right click this item for the menu to generate a mesh. See Chapter 5 - *CAD models* for more details. Mecway cannot save to this format.

### 12.3 .dxf

Mecway can read some features from the ASCII version of DXF R14 files. Nodes at the same locations are automatically merged together. Curves are converted to line3 elements which are parabolic curves so they cannot exactly represent the original geometry but are close enough for many purposes. If you refine these curved line3 elements or use **Automesh 2D** with them, they retain their curvature. Layers are converted to components with the 0 layer becoming the default component. Mecway cannot save to DXF format.

POINT – becomes a node

LINE – becomes a line2 element

LWPOLYLINE – straight segments become line2 elements and arc segments become strings of line3 elements in the same way as ARC.

CIRCLE – becomes 8 line3 elements.

ELLIPSE – becomes a string of line3 elements with each element spanning approximately 45 degrees.

ARC – becomes a string of uniformly sized line3 elements with each element representing an arc of approximately 45 degrees.

3DFACE – becomes a tri3 or quad4 element.

INSERT – imports the contents of a BLOCK. Rows and columns are ignored.

BLOCK – the contents are imported into Mecway with each associated INSERT entity but the block definition is discarded.

## 12.4 .stl

An STL file contains a list of triangles in 3D space. Each triangle is imported to Mecway as a tri3 element. Coincident triangle vertices are automatically merged into the same node to form a continuous mesh. You can also convert these surface triangles into a volume mesh using **Mesh tools** → **Automesh 3D**.

The STL files generated by CAD applications often contain triangles with very large aspect ratios (long and thin). These are usually unsuitable for use as finite elements so you should use **Automesh 3D** to improve their shape.

## 12.5 Gmsh (.msh)

Gmsh is an open source automeshing tool. You can use it to generate meshes then import them into Mecway. Mecway can read the ASCII file format versions 2.0, 2.1, 2.2, 4.0, and 4.1. If the file contains no solution data then Mecway assumes it is a mesh generated by gmsh and imports only the elements with the highest dimension – for example if there are both volume and surface elements, only the 3D elements appear in Mecway. This is because Gmsh uses surface and line elements to label the surfaces and edges of solids.

Geometric entities of surface elements are used to create named selections for surfaces in Mecway. If a volume element has a surface element on its face, then the geometric entity ID of that surface element is used to assign the volume element's face to a corresponding named selection. This preserves the distinct surfaces from the original geometry.

Geometric entities of volume elements are converted to components in Mecway. This allows you to use multiple materials in the same mesh with clearly defined boundaries between them.

Mecway can also write a mesh without solution data to Gmsh files. Components are stored as physical and geometric entities. 2D faces in face selections are stored as surface elements with physical and geometric entities to identify them.

## 12.6 .unv

You can open .unv mesh files generated by software such as Salome, Gmsh, and NX. Mecway reads nodes (dataset 2411), elements (dataset 2412), permanent groups (datasets 2435, 2452, 2467, 2477), units (dataset 164), and physical property table names (dataset 2470).

## 12.7 Netgen (.vol)

Netgen is an open source automeshing tool. You can use it to generate meshes then import them into Mecway. Most of the functionality of Netgen is already included with Mecway so you typically don't need to use this. Mecway only reads the elements with the highest dimension – either volume or surface elements. This is because Netgen uses surface elements to label the surfaces of solid parts.

## 12.8 Tetgen (.mesh)

Tetgen is an open source automeshing tool. You can use it to generate meshes then import them into Mecway.

## 12.9 Polygon File Format (.ply)

Mecway can open surface meshes consisting of tri3 and quad4 elements in both the ASCII and binary versions of this file format.

## 12.10 .jpeg, .png, .bmp

You can import images to display alongside the model. See the section *Tracing an Image* for more details. You can also save screenshots as PNG files.

## 12.11 .xyz

An XYZ file is a text file containing a list of points in 3D space. Each line has the 3 coordinates of a point. The coordinates can be delimited by whitespace or commas. Lines which contain text or not exactly 3 numbers are usually ignored. An exception is a single word followed by 3 numbers, which allows molecule geometry XYZ files to be opened. Mecway can open XYZ files but not save them.

## 12.12 Paraview (.vtu and .pvd)

You can save the solution in VTU format for post-processing with Paraview or compatible software. If there are multiple modes or time steps then each one is saved in a separate VTU file. A PVD file is also created which references the VTU file(s).

## 12.13 ANSYS command file (.txt)

Mecway can save some parts of the model as ANSYS APDL commands. These are:

- Nodes: N
- 3D solid mechanical elements: SOLID185 and SOLID186
- 3D solid piezoelectric elements: SOLID5 and SOLID226
- Displacement constraints: D and NANG
- Electric potential constraints: D
- Isotropic material: MP,EX and MP,PRXY
- Anisotropic stiffness matrix: TB,ANEL and TB,DATA
- Anisotropic compliance matrix: TB,ANEL and TB,DATA
- Anisotropic permittivity matrix: TB,DPER and TB,DATA

- Piezoelectric stress matrix (e): TB,PIEZ and TBDATA
- Piezoelectric strain matrix (d): TB,PIEZ and TBDATA
- Pressure (normal, uniform and constant): SFE

## 12.14 CalculiX (.inp)

### 12.14.1 Opening .inp files

The following keywords and parameters are supported.

Keyword	Parameters	Notes
*BEAM GENERAL SECTION	ELSET MATERIAL SECTION=PIPE	
*BEAM SECTION	ELSET MATERIAL SECTION=BOX / CIRC / I / L / PIPE / THICKPIPE / RECT	
*BOUNDARY		Displacement, rotation, fluid pressure, electric potential and temperature
*CFLUX		
*CLOAD		
*CONDUCTIVITY	TYPE=ISO / ORTHO	Can be temperature dependent with TYPE=ISO
*DASHPOT	ELSET	
*DENSITY		
*DLOAD		P, Pn load types for 2D/3D solids and shells. Not beams.
*DYNAMIC	DIRECT DIRECT USER CONTROL	
*ELASTIC	TYPE=ENGINEERING CONSTANTS TYPE=ISOTROPIC	
*ELEMENT	TYPE	

	ELSET INPUT	
*ELGEN	ELSET	
*ELSET	ELSET GENERATE	
*END STEP		
*FILM		Face numbers on shells can use the Abaqus (FPOS, FNEG) or CalculiX (F1-F6) forms.
*FREQUENCY		Number of eigenvalues on data line
*HEAT TRANSFER	STEADY STATE	
*INCLUDE	INPUT	It cannot appear within another keyword block.
*MATERIAL	NAME	
*MEMBRANE SECTION	MATERIAL ELSET	
*NCOPY	OLDSET CHANGENUMBER NEWSET SHIFT MULTIPLE	
*NFILL	NSET BIAS TWO STEP	
*NGEN	NSET LINE=P / C	
*NODE	NSET SYSTEM=C / S / R INPUT	
*NSET	NSET	

	ELSET GENERATE UNSORTED	
*ORIENTATION	NAME	Only a rectangular system defined by two or three points' coordinates and on 3D solid and shell elements. For shell elements, Abaqus and CalculiX assign different meanings to the coordinates. If either a 3rd point or a 2nd data line is specified then Mecway will interpret it as Abaqus format, otherwise it will interpret it as CalculiX format.
*PHYSICAL CONSTANTS	ABSOLUTE ZERO	
*RADIATE		Face numbers on shells can use the Abaqus (FPOS, FNEG) or CalculiX (F1-F6) forms.
*SHELL SECTION	MATERIAL ELSET	
*SOLID SECTION	MATERIAL ELSET ORIENTATION	
*SPECIFIC HEAT		Can be constant or temperature dependent
*SPRING	ELSET	
*STATIC	DIRECT	
*STEP	NLGEOM	
*SURFACE	NAME TYPE=NODE / ELEMENT	
*TRANSFORM	NSET TYPE=C / S / R	

Element types are converted to various element shapes, material properties and analysis types according to the table below. If there are conflicts such as a mixture of 2D and 3D elements or no

compatible analysis type then only one will be chosen. Some element types are not supported by Mecway's solver but will still be imported. These include 3-node beams, hexahedra collapsed into prisms, and axisymmetric shells. Other element shapes don't exist in Mecway, such as some of the cohesive (COH), gasket (GK) and cylindrical (CCL) element types. These will be converted to nearest available element shape without adding any extra nodes.

Element type in inp file	How it appears in Mecway
T2D2,...	2D truss
T3D2,...	3D truss, 2D beam, 3D beam or fin depending on the analysis procedure
SPRINGA	3D beam with spring material property
DASHPOTA	3D beam with damper material property
CPS3,...	2D continuum element with plane stress material property
CPE3,...	2D continuum element with plane strain material property
CAX3,...	Axisymmetric
S8R,...	3D shell. The local element axes are determined by an associated *ORIENTATION block if present. Otherwise the default is U is parallel to X unless the element's normal direction is within 0.1 degrees of the X axis, in which case U is parallel to Z.
C3D8,...	3D solid

### General Limitations

- All text is case insensitive, including labels in quotation marks.
- All spaces are ignored. This means filenames and labels in quotes may not be read correctly.
- Keywords and most parameter values must be complete although parameter names can be shortened, for example ISO instead of ISOTROPIC.
- If node or element numbers are not contiguous, they'll be renumbered so that they are.
- All physical quantities are assumed to be in SI units (m, kg, s, K) except angle which may be in degrees or radians according to the file format's specification. An exception is that you can choose the units of node coordinates when opening the file.

### **12.14.2 Saving .inp files**

This file is intended for use with the CCX solver and while it is also mostly compatible with Abaqus, some data is interpreted differently.

All quantities are saved in SI units (kg, m, s, K) and radians. So if a length is 1 inch in Mecway, it will appear as 0.0254 in the .inp file.

## **12.15 CalculiX results (.frd)**

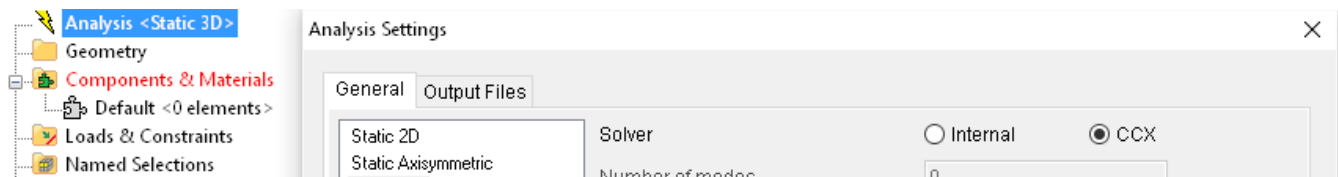
Mecway can read most forms of this file as generated by CCX versions 2.8-2.21. It assumes all quantities are in SI units (m, kg, s). This requirement will be met if the .frd file is the solution to a .inp file generated by Mecway regardless of the units used to define the model in Mecway.

## CalculiX Solver

In addition to Mecway's built in solver, you can also use the open source CalculiX solver (CCX). This gives access to more advanced features like frictional contact and plastic material behavior.

CCX 2.21 is installed along with Mecway and ready to use. If you prefer to install another version then install it and go to **Tools** → **Options** → **CalculiX** and **Browse** to the location of the solver's executable **ccx.exe**. In either case, if you prefer to keep the automatically generated files in your own working directory, check **Use the same directory as the LIML file**, otherwise temporary files will be created in a temporary folder and deleted when you close Mecway.

Change the model's solver option from **Internal** to **CCX** under **Analysis settings**. When solving, Mecway will generate a .inp file, run the CCX solver with this filename as a parameter then read the .frd file it generates and display the solution.



If the solve fails, click the **CCX output** button to show CCX's standard output and standard error messages as well as error messages generated by Mecway while writing the .inp file. You can also inspect the .inp file's contents with the **Input file** button.

Units are converted to a consistent system (m, kg, s) as required by CCX. This means you can use any combination of units while building the model in Mecway and the solution will also have correct values which can be displayed in any of the available units.

Mecway sets the `MKL_PARDISO_OOC_MAX_CORE_SIZE` and `MKL_PARDISO_OOC_PATH` environment variables for the CCX process. These enable the out-of-core feature of MKL PARDISO with the version of CCX that is included as source code.

A branch called **CCX** appears in the outline tree when the **Solver** is set to **CCX**. By right clicking on this, you can control how the .inp file is generated. The options are:

### 13.1 Custom model definition

Here you can enter text that will be inserted into the .inp file at the end of the model definition section immediately before the first step section (identified by a \*STEP keyword). You can use it to manually add different elements, materials, homogeneous (zero) boundary conditions, etc. that are not available in Mecway. See the CCX documentation for details of the keywords.

For example, to define a compression-only material with Youngs modulus of 30 GPa and maximum tensile stress of 3 MPa, enter the following text in the **custom model definition**:

```
*SOLID SECTION, ELSET=Default, MATERIAL=COMPRESSION_ONLY_1
*MATERIAL, NAME=COMPRESSION_ONLY_1
*USER MATERIAL, CONSTANTS=2
30e9, 3e6
```

The elements must belong to a component named *Default* with no material assigned to it in Mecway.

## 13.2 Custom step contents

**Custom step contents** is similar to **custom model definition** but the text is placed at the end of a step block (identified by \*STEP). You can use it to define loads, output variables, non-homogeneous boundary conditions or additional step sections.

For analysis types that generate two step blocks, the text is placed in the step block containing loads.

## 13.3 Modify keyword

Sometimes the .inp file cards generated by Mecway may not have the same options that you want. If this happens, you can use **modify keyword** to alter all instances where that keyword would normally be generated by Mecway. Available modifications are:

**Omit key word block.** Prevents the keyword card and its associated data lines from being generated. This was called **Don't generate keyword** in earlier versions of Mecway.

**Set parameter.** Sets a parameter to a value. For example, to use mortar contact, set the parameter with **Name TYPE** to the **Value MORTAR** on the \*CONTACT PAIR keyword.

**Omit parameter.** Prevents the specified parameter being generated.

## 13.4 Don't generate STEP

Add this item to prevent the \*STEP section(s) being generated by Mecway. It's useful if you're replacing it with your own \*STEP section defined in **custom model definition**.

## 13.5 No extra contact step

Using this compatibility item may cause incorrect results for quasi-static contact. For more details, open it in Mecway.

## 13.6 Custom element type

This allows you to change the element type used for the CCX solver to something which isn't otherwise supported by Mecway. The specified element type is applied to all elements that match the specified shape and are in the selection that **custom element type** is applied to.

For example, to use reduced integration for all hex8 elements, add a **custom element type** with:

<b>Apply to</b>	<Whole model>
<b>Element shape</b>	Hex8
<b>CCX element type</b>	C3D8R

## OpenRadioss Solver

Mecway can be used as a pre- and post-processor for the open source explicit dynamics solver OpenRadioss.

The Altair Radioss documentation <https://help.altair.com/hwsolvers/rad/index.htm> mostly applies to OpenRadioss.

### 14.1 Setup

1. Download OpenRadioss for Windows and extract it to an installation directory. <https://github.com/OpenRadioss/OpenRadioss/releases>
2. In Mecway, click **Find OpenRadioss root directory** in **Tools** → **Options** → **OpenRadioss** and choose the installation directory. This will automatically identify the starter and engine executables and define some environment variables.
3. In **Analysis settings**, set the **Solver** to **OpenRadioss**.

### 14.2 Supported features

#### 14.2.1 Analysis type

Nonlinear Dynamic Response 3D

#### 14.2.2 Elements

Beam, truss, spring, damper: line2

Shell: tri3, quad4

Solid: tet4, pyr5, wedge6, hex8, tet10, hex20

#### 14.2.3 Materials

Isotropic elastic

Custom materials defined by /MAT/ . . . keywords

#### 14.2.4 Initial Conditions

Velocity

#### 14.2.5 Loads & Constraints

Fixed support

Frictionless support  
Displacement  
Node rotation  
Velocity  
Contact  
Flexible joint on beam  
Force  
Pressure (uniform only)  
Moment  
Element orientation (beams only)  
Symmetry  
Gravity

#### 14.2.6 Named selections

Node selections are exported as node groups (`/GRNOD/NODE`)

Face selections are exported as surface segments (`/SURF/SEG`) containing 2D faces and line segments (`/LINE/SEG`) containing 1D faces (edges). Quadratic element faces (tet10, hex20) are not supported.

Element selections are exported as groups (`/GRBRIC/BRIC`, `/GRSHEL/SHEL`, `/GRSH3N/SH3N`, `/GRBEAM/BEAM`, `/GRSPRI/SPRI`, `/GRTRUS/TRUS`) containing their respective element types.

The group and segment IDs are numbered sequentially from 1 in alphabetical order of the names of their named selections. This allows you to refer to them in other keywords in custom model definition (see below).

Named selections can contain a mixture of face or element types which are exported as multiple segments or groups sharing the same ID. For example, if the 5th named selection contains tri3 and quad4 elements, it is exported as `/GRSH3N/SH3N/5` and `/GRSHEL/SHEL/5` which have *grsh3n\_ID* = 5 and *grshell\_ID* = 5 respectively. This allows you to refer to either of them in other keywords as needed.

#### 14.2.7 Solution variables

Displacement  
Velocity  
Acceleration  
Axial force (beam, truss, spring, damper)  
Shear force (beam)  
Bending moment (beam)  
Torsion moment (beam)  
Reaction force

Reaction moment

Stress (solid)

Von Mises stress (solid)

Element time step

Time step (global)

Internal energy (global)

Kinetic energy (global)

#### **14.2.8 Special features**

These are available in the **OpenRadioss** branch of the outline tree.

##### Custom model definition

Here, you can enter starter keywords for features that aren't supported by Mecway's interface, such as contact, other types of loads and elements, etc.

##### Custom engine input

Here, you can enter engine keywords for features that aren't supported by Mecway's interface, such as for outputting different solution variables.

## Automation

### 15.1 Command Line Parameters

Command line parameters can be used to run Mecway from another application, a batch file or a shell script.

```
mecway [<filename> [solve | clearsolutions | script <script filename>]]
```

The optional <filename> is a file of any of the mesh or geometry types that Mecway can open. It causes Mecway to start with that file open. The file name must include the extension such as .liml, .stp, etc. For example:

```
mecway mymodel.liml
```

The optional **solve** parameter causes Mecway to solve the file then exit without displaying the main window. To obtain the solution, set an output file in the model, such as **Write solution to liml file** or a table with **Write to csv file after solving**. When you use the **solve** parameter, <filename> can contain wildcards. For example to solve all liml files in the current folder, run:

```
mecway *.liml solve
```

**clearsolutions** is useful to reduce the size of a large file by removing its solution data.

**script** opens the file <filename>, runs the script <script filename>, then exits. If <filename> defines multiple files by using wildcards then it runs the script after open each file.

### 15.2 Scripts

You can write scripts that appear as menu items and are executed by a built-in implementation of Python 2.7 (IronPython). The approximate scope of the functionality is currently:

- Simple operations on nodes and elements
- Mesh refinement, smoothing, and automesh
- Material properties
- Numerical values in most loads and constraints
- Named selections
- Solution field variables
- Measurement tools that integrate over elements or faces
- Read the current selection
- Read and modify the elements in components

- Solve (with no progress indication)
- File Open and Save
- Python standard library
- Importing modules written in pure Python

It does not include:

- Most meshing tools
- Importing modules that require CPython, such as NumPy

To create a script, click **Tools** → **Scripts...** and press the **New** button. Enter a non-existent file name to be created or choose an existing one. The editor (default is Notepad) then opens where you can write and save the script. Here is a simple example that creates a node and displays a message in a pop-up box:

```
node_id = mw.new_node(Vector(1.2, 3.4, 5.6))
mw.message("Node " + str(node_id) + " created.")
```

To run a script, click its menu item in the **Tools** menu. You can also activate it from the keyboard using **Alt+T** followed by the first letter of the script's name or run it with **File** → **Open...**, **File** → **Import...** or **File** → **Import from library**.

The API to Mecway consists of the following objects and classes which are automatically available in every script:

- mw** Object containing functions that operate on the application and the model in the modeler
- solution** Object containing functions that operate on the solution in the active configuration
- FaceId** Class for identifying a face of an element
- Vector** Class for representing a 3D vector

Details are listed in a separate document accessible through **Help** → **API reference**.

Two sample scripts are provided in the **Samples** folder where Mecway is installed. They are **generate\_mesh.py** which creates an element, and **transfer\_displacements\_from\_solution.py** which reads displacement data from the solution and uses it to deform the mesh.

All physical quantities are expressed in the SI unit system (m, kg, s, K, A, radian) regardless of how they were specified in the model. For example, if your model has a node with a coordinate of 1 inch, then reading that using a script will return the value 0.0254.

Cautions:

- Scripts can contain malware so avoid running unknown scripts from untrusted sources.
- The application will be unresponsive while a script is running so avoid infinite loops.
- The API, Python version, and Python implementation may change in future versions of Mecway, breaking compatibility with existing scripts. However, you can continue to use old major versions of Mecway alongside newer ones if you don't want to update your scripts.



## Tutorials

These tutorials assume that you are new to Mecway and indeed may be new to finite element analysis. Here you will find step by step instructions to get you started using the program.

### 16.1 Quick Start

This is a very simple tutorial showing you how to create a working model from scratch. You can skip this if you prefer a more in depth introduction from the other tutorials.

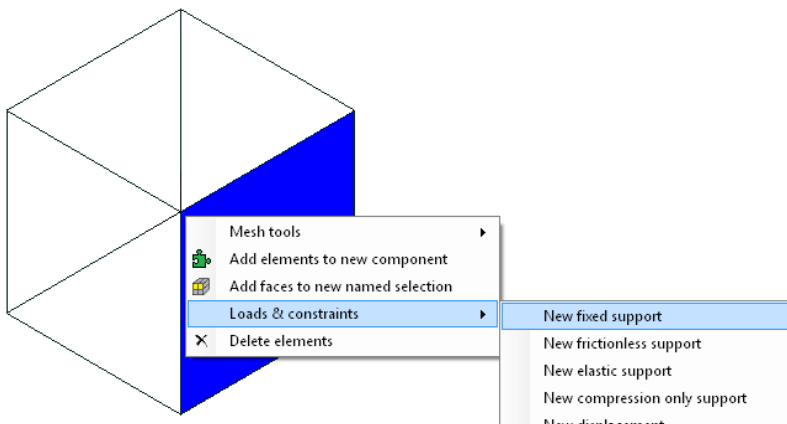
#### Step 1

Change to Select faces mode.



#### Step 2

Click the Quick cube button  to create a hexahedron element.

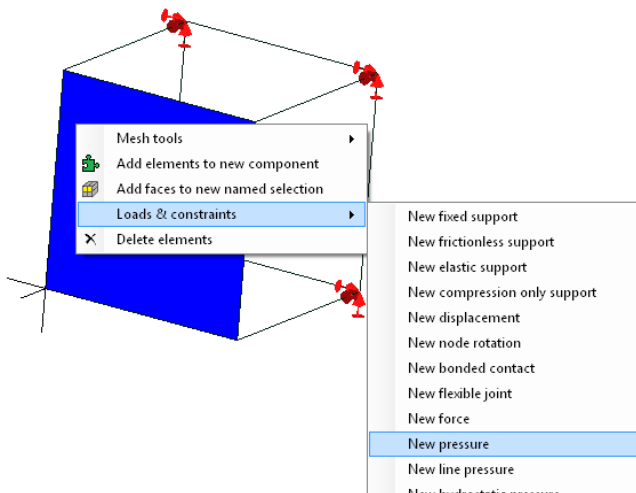


#### Step 3

Right click the right-most face and click **Loads & constraints** then **Fixed support**. Click **OK** in the box that appears.

#### Step 4

Drag with the middle mouse button to rotate the model so the opposite face is towards you.

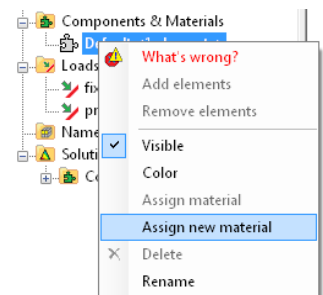


### Step 5


Right click the opposite face and click **Loads & constraints** then **Pressure**. Enter the value **5** and press **OK**.

### Step 6

Right click the **Default** component in the outline tree and click **Assign new material**. Choose **Isotropic** and enter a **Young's modulus** of 50000. Press **OK**.

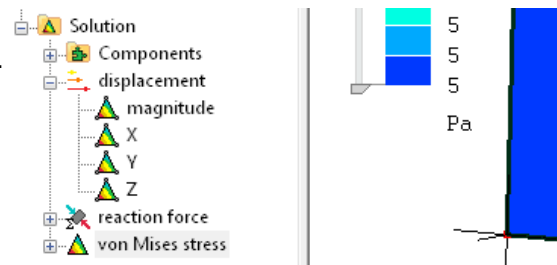


### Step 7

Press solve. 

### Step 8

Click the **von Mises stress** field variable in the outline tree. The color key shows a 5 Pa uniform stress due to the pressure load.



## 16.2 Viewing and Selecting

This tutorial shows you how to do most common operations for changing the view and selecting items. It refers the menus but you may prefer to use the toolbar buttons instead, where they exist.

### Step 1

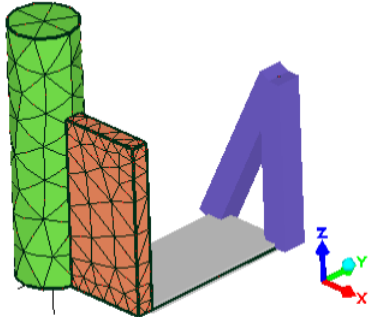
From the **File** menu, Click **Open** and open the file *ViewingAndSelecting.liml* in the tutorials folder where Mecway has been installed.

### Step 2

Use the middle mouse button to rotate the model, the right button to pan, and the scroll wheel to zoom.

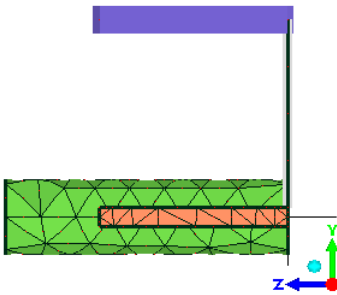
### Step 3

From the **View** menu, click **Fit to window** to re-center the model.



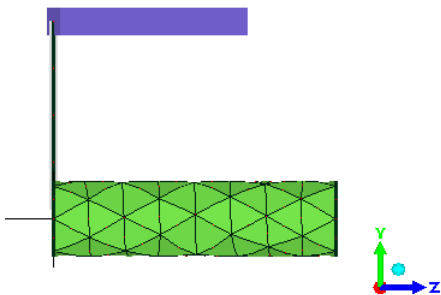
### Step 4

Click the X axis arrowhead on the triad to rotate to an orthogonal view.



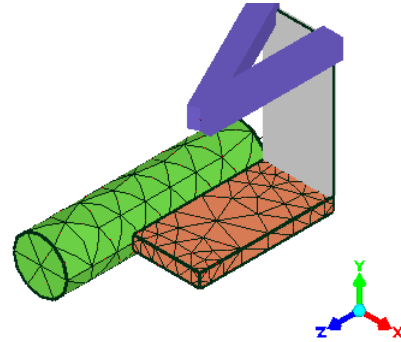
### Step 5

Right click the X axis arrowhead for the view from the -X direction.



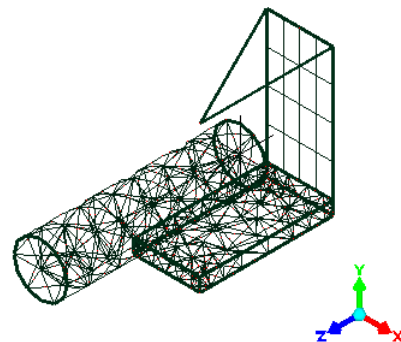
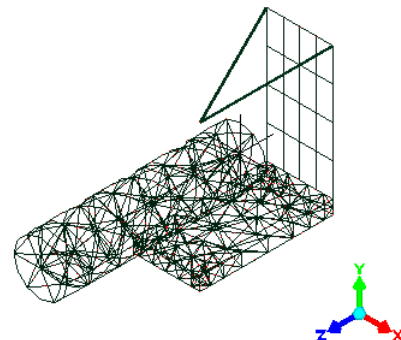
### Step 6

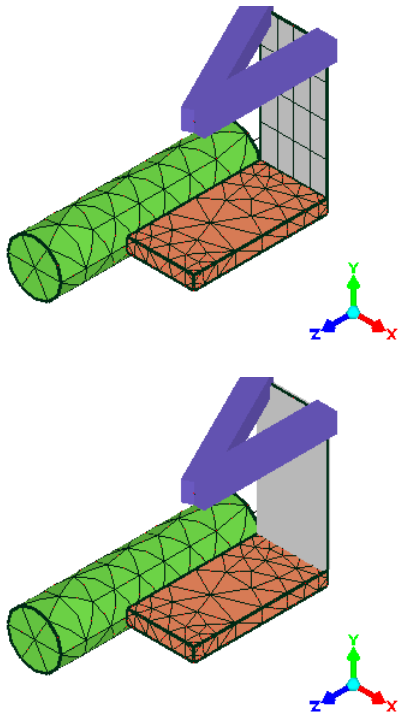
Click the cyan ball in the triad to rotate to isometric view.



### Step 7

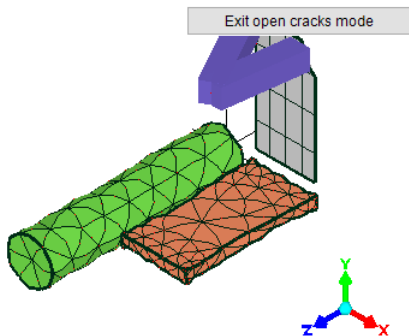
Turn each of these options in the **View** menu on and off to see its effect: **Show element edges**, **Show model edges**, **Show element surfaces**, **Show thickness**.





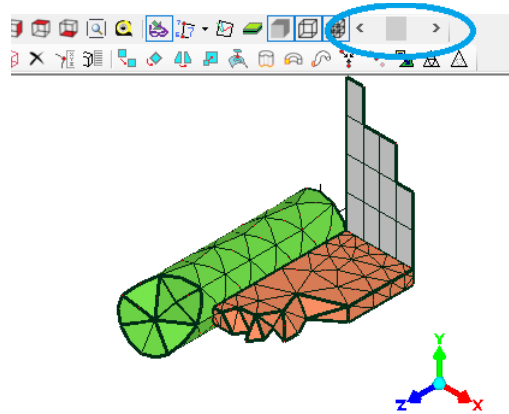
#### Step 8

From the **View** menu, click **Open cracks**. This shrinks the elements and reveals that the shells are not connected to the solids. Press the **Esc** key to exit from Open cracks mode.



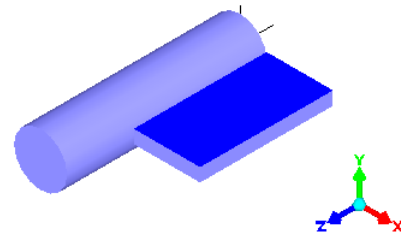
#### Step 9

Move the cutting plane slider to hide the foreground elements and see inside the mesh. Then return it to the far left.



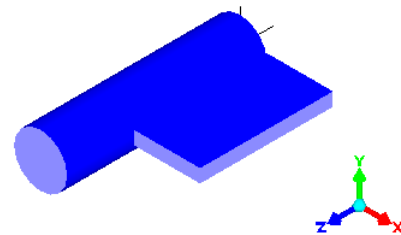
#### Step 10

Click **Geometry** in the outline tree to display the STEP files and click on a surface to select it.



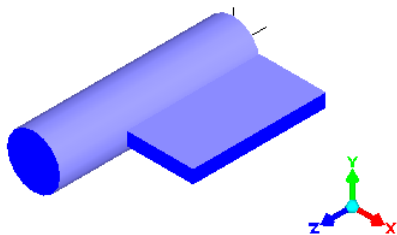
#### Step 11

Hold the **Ctrl** key and click on another surface to also select it.



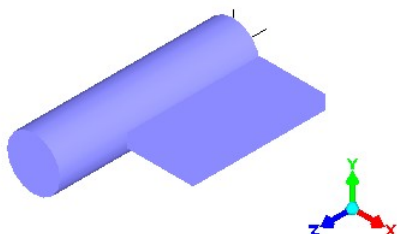
#### Step 12

From the **Edit** menu, click **Invert selection** to select all other surfaces except those two.



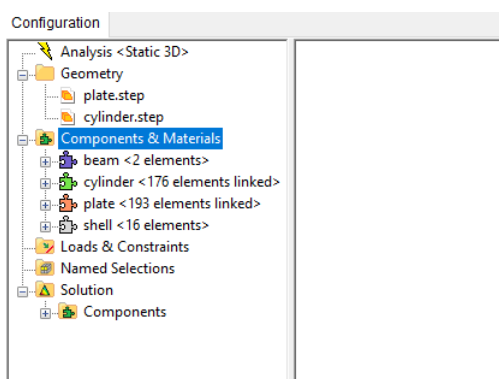
### Step 13

Click on a blank place beside the model to clear the selection.



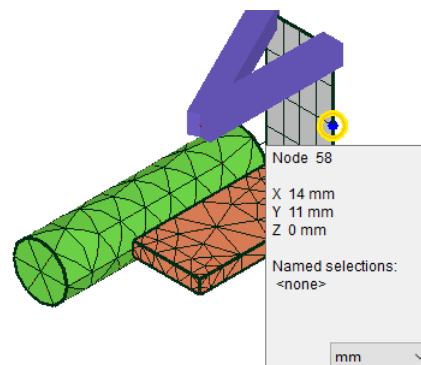
### Step 14

Click **Components & Materials** in the outline tree to display the mesh.



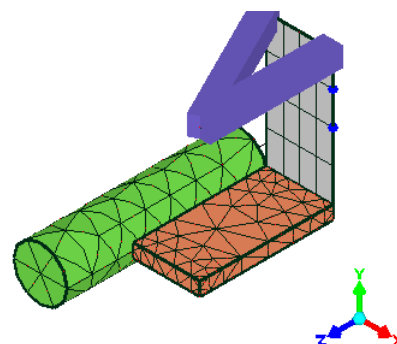
### Step 15

From the **Edit** menu, click **Select nodes**, then click on a node to select it.



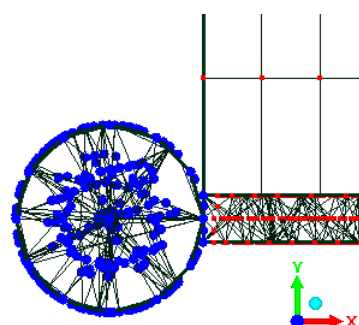
### Step 16

Hold the **Ctrl** key and click on another node to also select it.



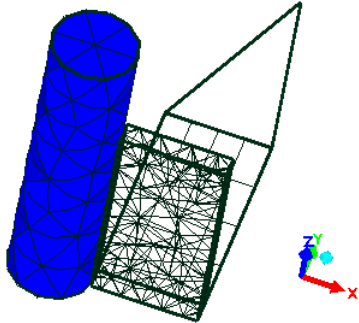
### Step 17

With **Show element surfaces** turned off, click the Z axis arrowhead on the triad, and drag a rectangle around the nodes of the cylinder to select them.



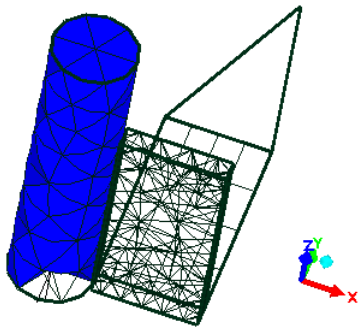
#### Step 18

From the **Edit** menu, click **Select elements** to convert the node selection into an element selection containing all elements of the cylinder.



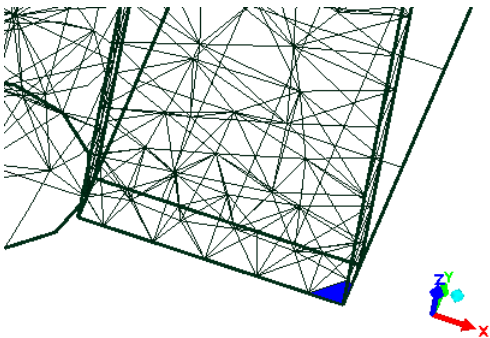
#### Step 19

Hold the **Ctrl** key and click on some of the selected elements to deselect them.



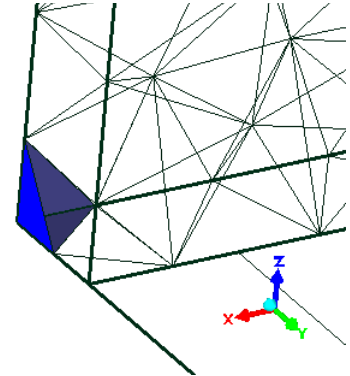
#### Step 20

Click on a corner element to select it alone.



#### Step 21

From the **Edit** menu, click **Select faces** to convert the element selection into a face selection. Notice that only the exterior faces are selected.

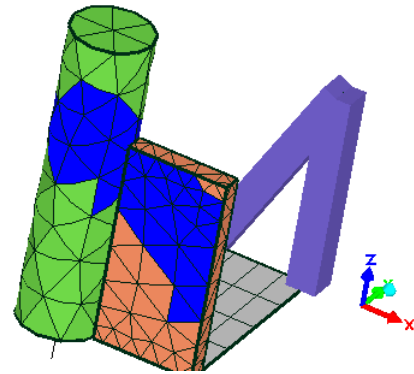


#### Step 22

Click on a blank place beside the model to clear the selection.

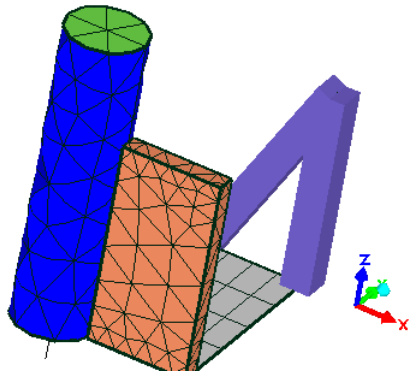
#### Step 23

From the **Edit** menu, click **Paint selection** and drag the mouse over the mesh to select many faces.



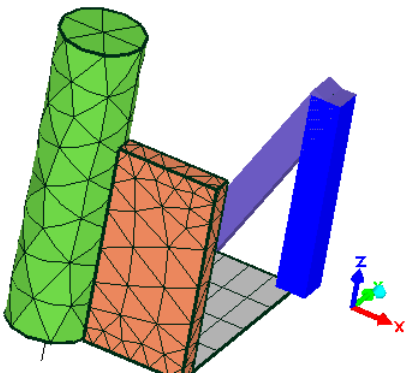
#### Step 24

From the **Edit** menu, click **Edge detecting selection** and click on the cylinder to select all faces bounded by a steep angle between adjacent faces.



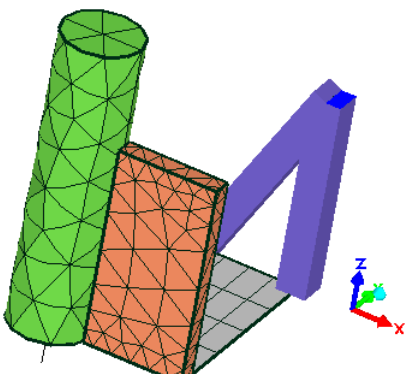
#### Step 25

With **Show element surfaces** turned on, click on one of the beam elements to select its edge "face" which is the entire perimeter.



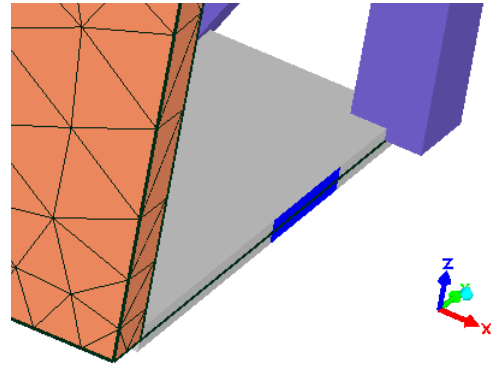
#### Step 26

Click on the end of a beam element to select its end face.



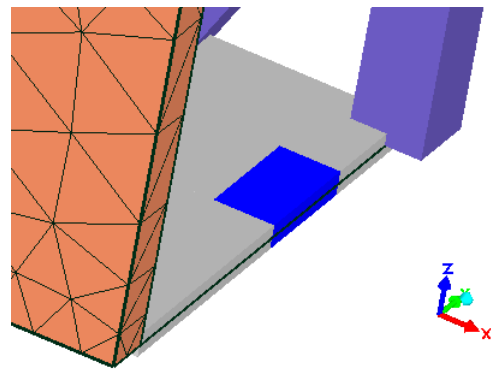
#### Step 27

With **Show thickness** turned on, click on an edge face of the shell element to select it.



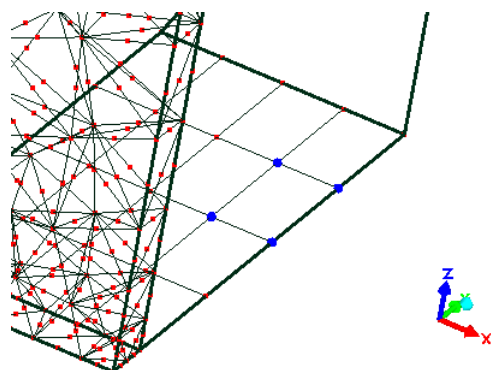
#### Step 28

From the **Edit** menu, click **Select elements** and the element whose face was selected will become selected.



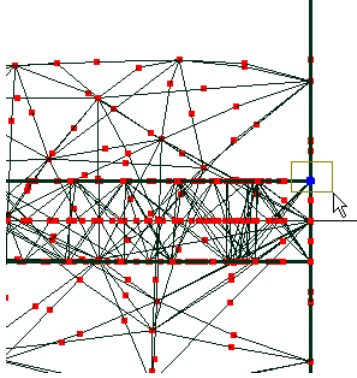
#### Step 29

From the **Edit** menu, click **Select nodes** and all the nodes of the selected element will become selected.



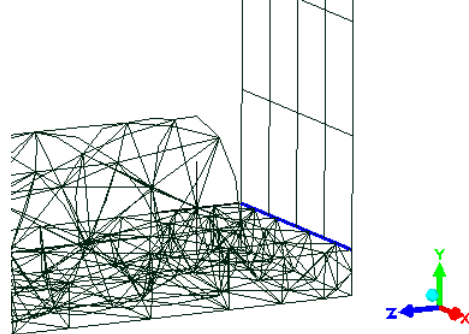
### Step 30

With **Show element surfaces** turned off, click the X axis arrowhead on the triad, and drag a rectangle around the nodes shown below.



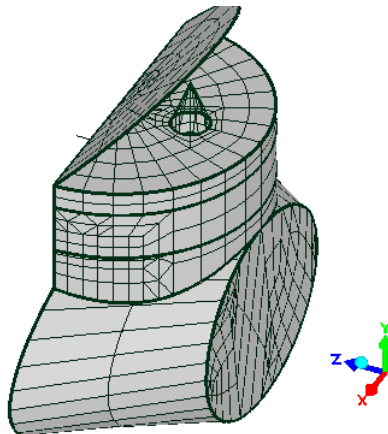
### Step 31

From the **Edit** menu, click **Select faces** to convert the node selection to the shell faces along that edge.



## 16.3 Manual Meshing

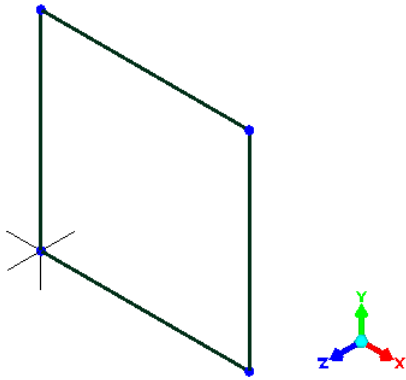
This tutorial shows you how to use the most common manual meshing tools to build an arbitrary 3D structure from scratch that includes, solid, shell and beam elements.



It assumes you are already familiar with content of the Viewing and Selecting tutorial above. If you make a mistake, click **Undo** in the **Edit** menu to revert to the previous step. You may prefer to close each tool's window after use if they're cluttering the screen.

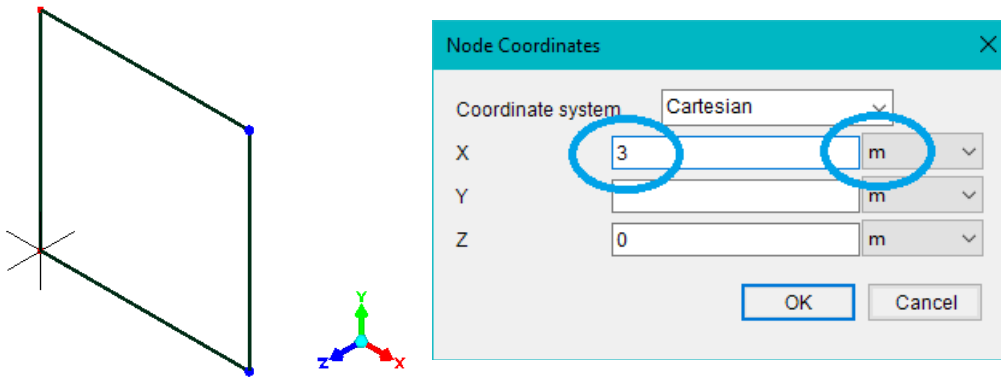
### Step 1

From the **Mesh tools** menu, open **Create** and click **Quick square**.



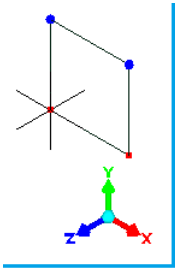
### Step 2

Select the two nodes on the +X edge as shown below. From the **Mesh tools** menu, click **Node coordinates**, set the parameters shown below and click **OK**.



### Step 3

Select the two nodes on the +Y edge as shown below. If the square has become a slender rectangle, you may need to click the X axis arrowhead on the triad to view it end-on as shown below and select the two nodes by dragging a rectangle around them. From the **Mesh tools** menu, click **Node coordinates**, set the parameters shown below and click **OK**.



**Node Coordinates**

Coordinate system:

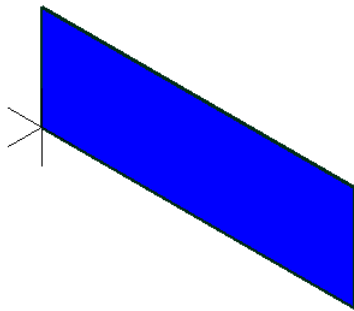
X:  m

Y:  m

Z:  m

#### Step 4

Select the element. From the **Mesh tools** menu, open **Refine** and click **Custom**, set the parameters shown below and click **Apply**.



**Refine Custom**

Line - Only use R  
 Quad - Use R and S  
 Wedge - Only use T  
 Hex - Use R, S and T

Number of subdivisions on internal element axes

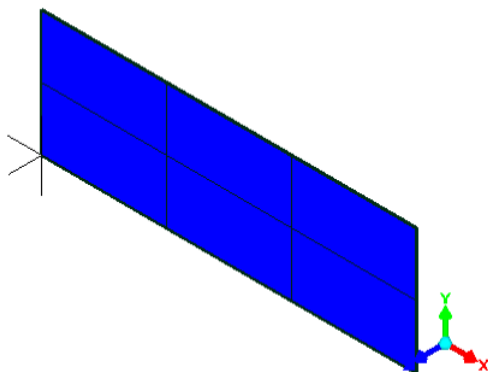
R:

S:

T:

#### Step 5

Change to **Select faces** mode to select the 6 faces shown below. From the **Mesh tools** menu, click **Revolve**, set the parameters shown below and click **Apply**.



**Revolve**

Axis direction: ☐ X ☒ Y ☐ Z

Origin X:  m

Origin Y:  m

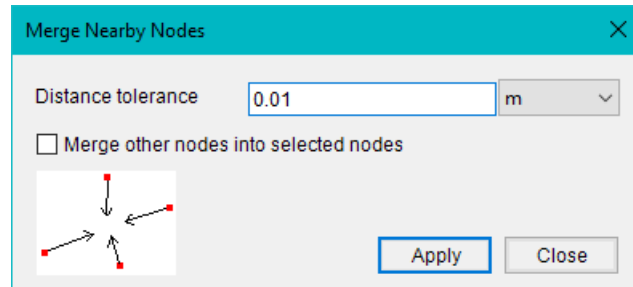
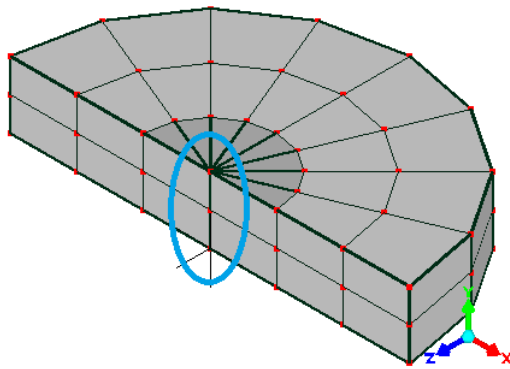
Origin Z:  m

Angle:  °

Subdivisions:

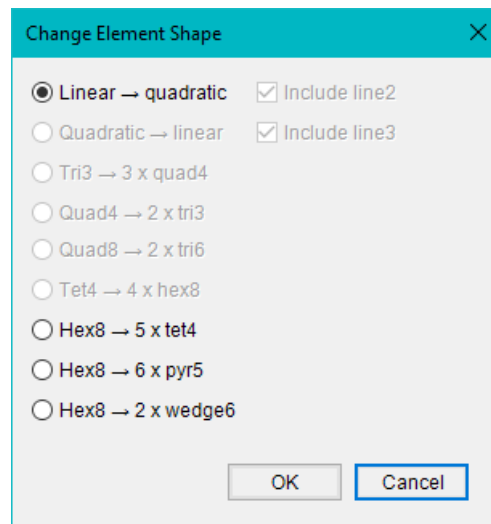
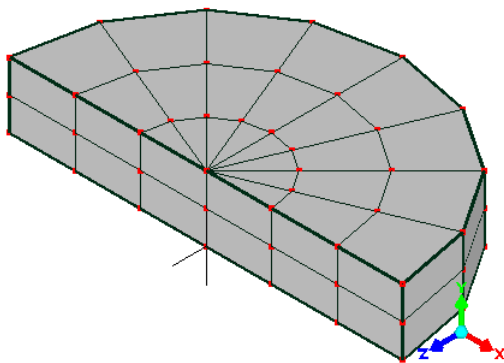
### Step 6

The previous step generated many coincident nodes at the axis where the elements aren't connected to each other. To correct this, with nothing selected, click **Merge nearby nodes** in the **Mesh tools** menu, set the parameters shown below and click **Apply**.



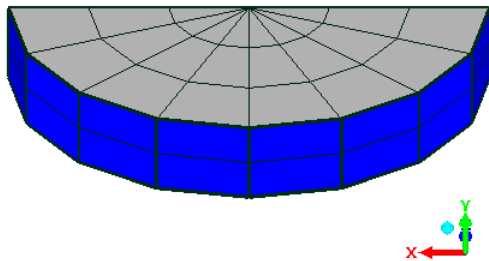
### Step 7

From the **Mesh tools** menu, Click **Change element shape** and click **OK**.



### Step 8

The faces on the circular arc are now flat instead of curved. To correct this, select the 16 faces shown below, then from the **Mesh tools** menu, click **Project onto surface**, set the parameters shown below and click **Apply**.



**Project Onto Surface**

Source: Use selection 16 faces

Target:

- ☐ Mesh
- ☒ Cylinder
- ☐ Sphere
- ☐ Cone
- ☐ Torispherical dome

Axis direction: ☐ X ☒ Y ☐ Z

Origin X: 0 m

Origin Y: 0 m

Origin Z: 0 m

Mouse pick

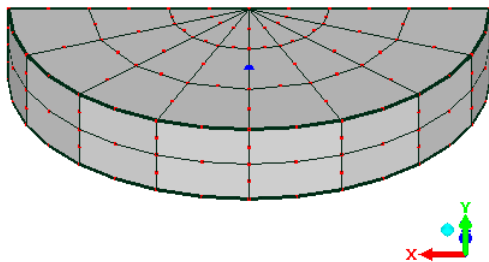
Radius: 3 m

Direction to move: ☐ X ☐ Y ☐ Z ☐ Source normal ☒ Target normal

Apply Close

### Step 9

Select the node on the +Y surface shown below. From the **Mesh tools** menu, click **Move/copy**, set the parameters shown below, and click **Apply**. This creates a new isolated node above the part.



**Move/Copy**

Coordinate system: Cartesian

$\Delta X$ : m

$\Delta Y$ : 1 m

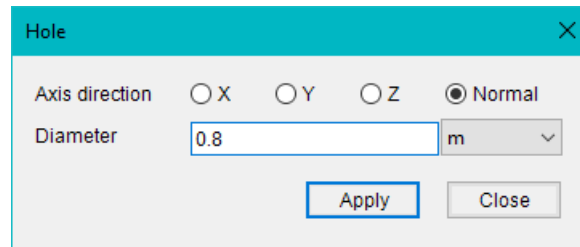
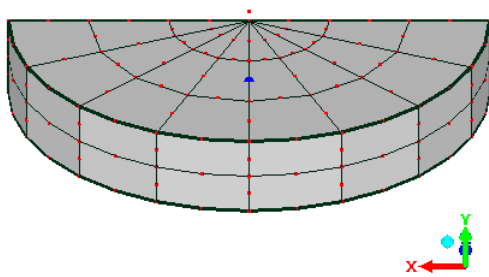
$\Delta Z$ : m

☒ Copy

Apply Close

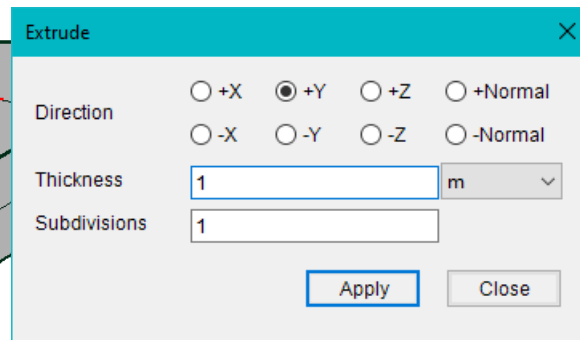
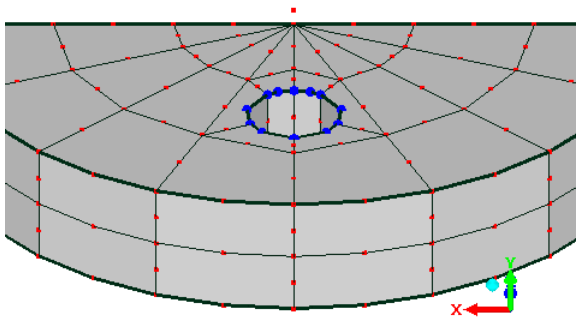
### Step 10

Select the same node again. From the **Mesh tools** menu, click **Hole**, set the parameters shown below and click **Apply**.



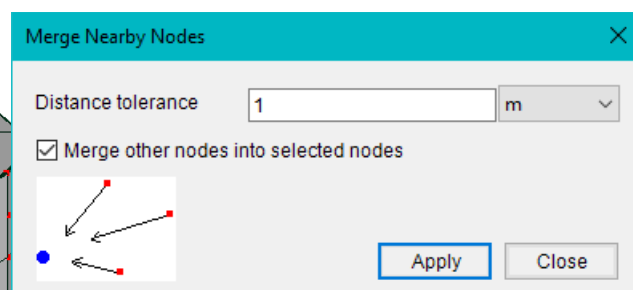
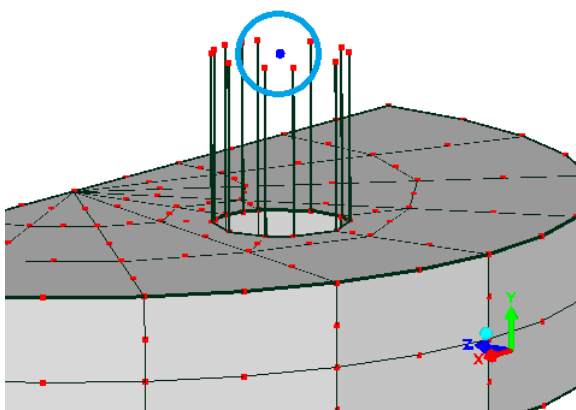
### Step 11

Select the 12 nodes around the edge of the hole as shown below. From the **Mesh tools** menu, click **Extrude**, set the parameters shown below and click **Apply**.



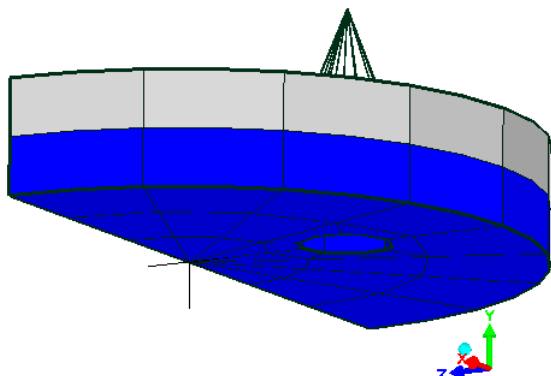
### Step 12

Change to **Select nodes** mode and select the node shown below. From the **Mesh tools** menu, click **Merge nearby nodes**, set the parameters shown below and click **Apply**.



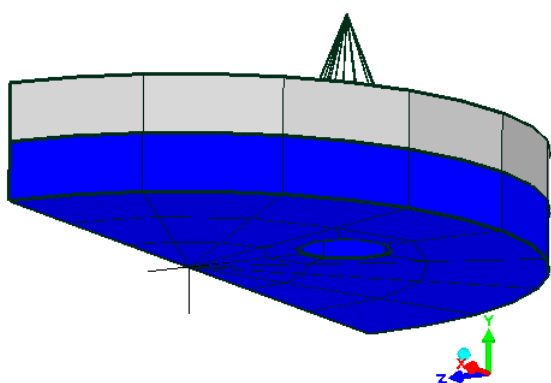
### Step 13

Select the elements shown below. From the **Mesh tools** menu, click **Disconnect elements**.



### Step 14

With the same elements still selected, click **Move/copy** in the **Mesh tools** menu, set the parameters shown below and click **Apply**.



Move/Copy

Coordinate system

Cartesian

$\Delta X$

m

$\Delta Y$

-1

m

$\Delta Z$

m

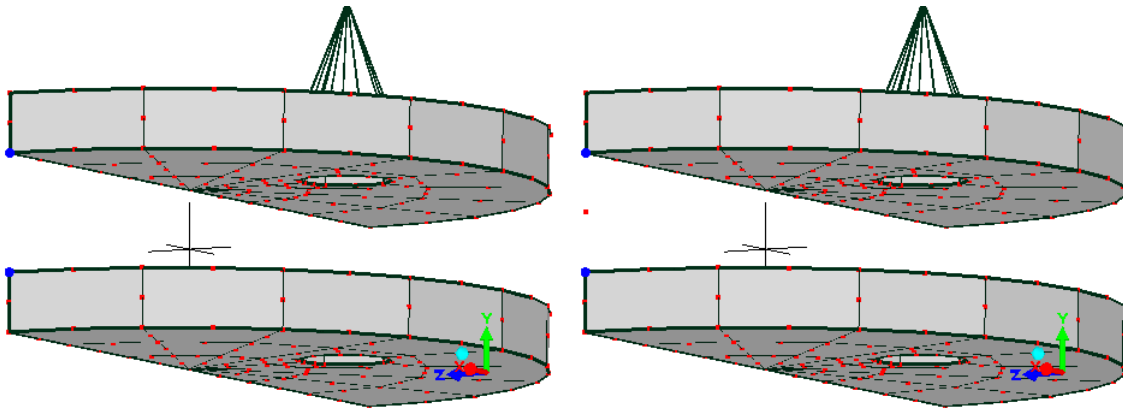
☐ Copy

Apply

Close

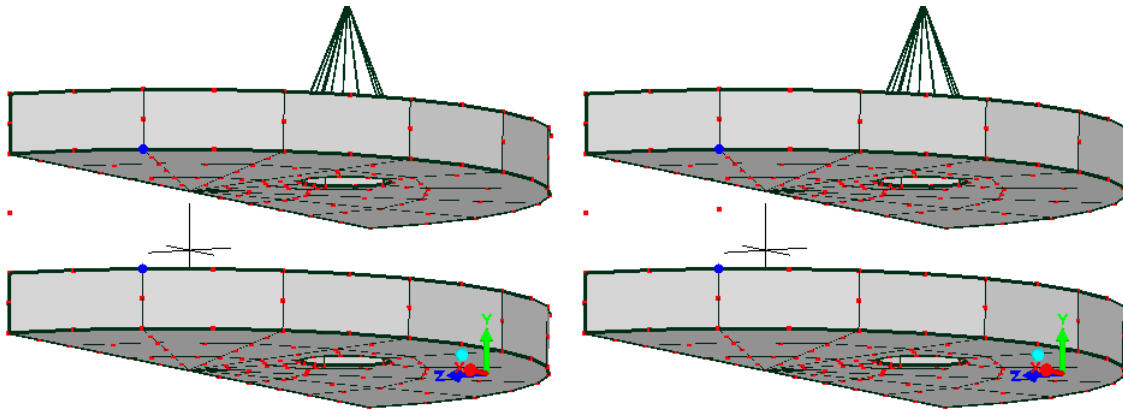
### Step 15

Select the two nodes shown below. From the **Mesh tools** menu, click **Insert node between**.



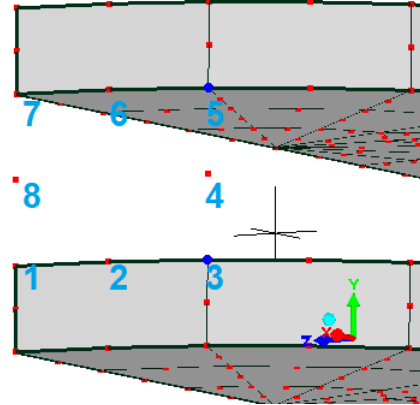
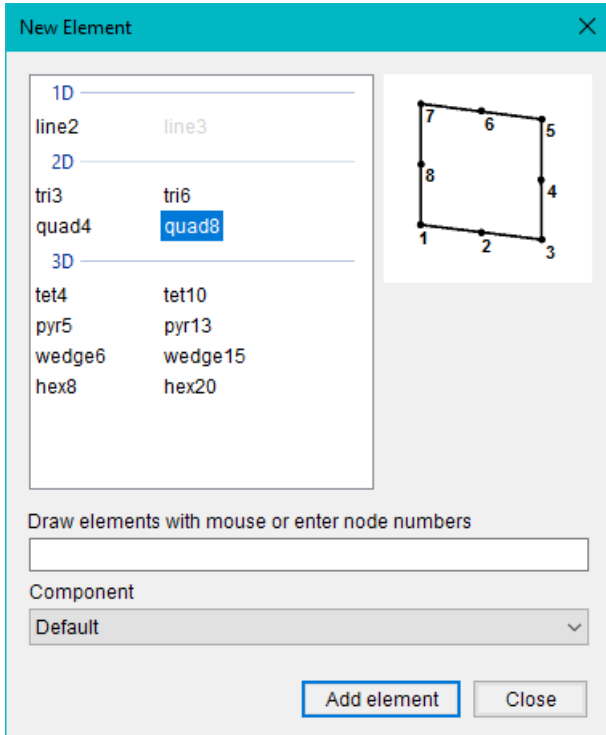
### Step 16

Repeat the previous step using the two nodes shown below.



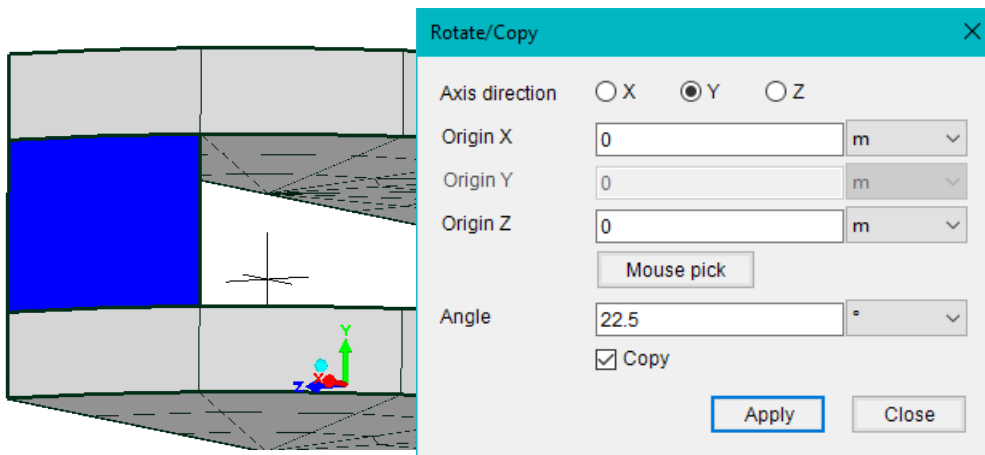
### Step 17

From the **Mesh tools** menu, open **Create** and click **Element**. Select **quad8** in the list, then click each of the 8 nodes shown below in the order indicated. A new element will appear immediately after clicking the 8<sup>th</sup> node.



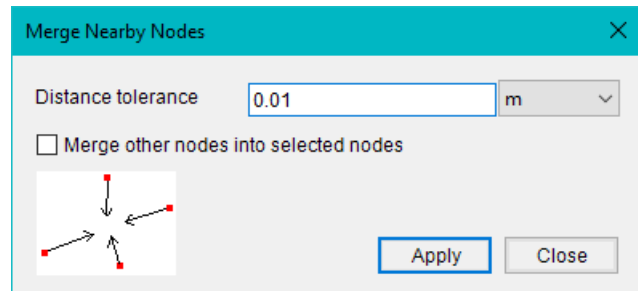
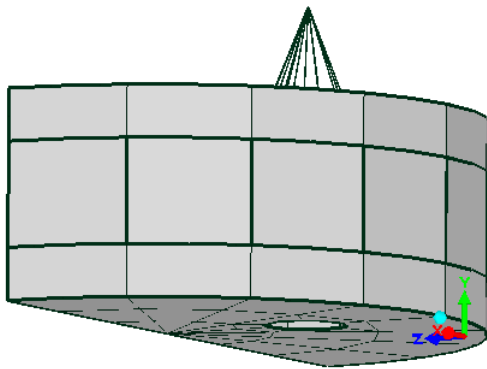
### Step 18

Select the new element as shown below. From the **Mesh tools** menu, click **Rotate/copy**, set the parameters shown below and click **Apply** 7 times.



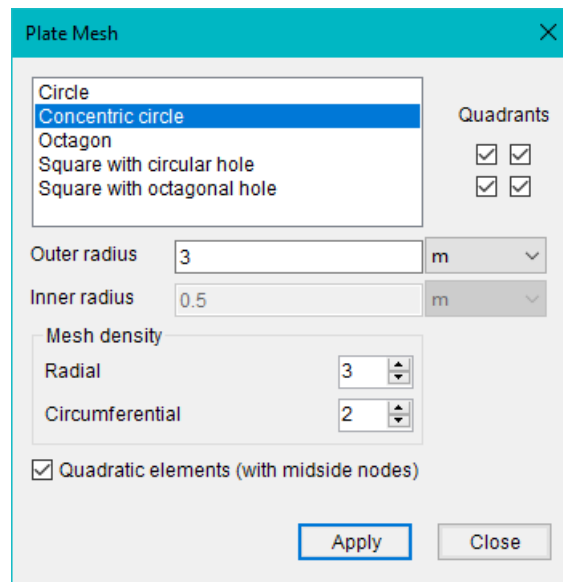
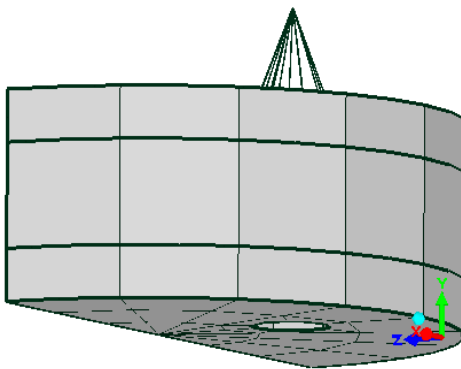
### Step 19

The previous operation leaves the new elements disconnected. To correct this, with nothing selected, click **Merge nearby nodes** in the **Mesh tools** menu, set the parameters shown below and click **Apply**.



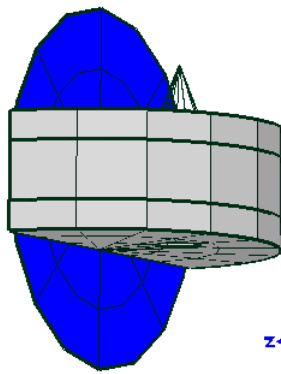
### Step 20

From the **Mesh tools** menu, open **Create** and click **Plate mesh**, set the parameters shown below and click **Apply**.



### Step 21

With the new elements still selected, click **Rotate/copy** in the **Mesh tools** menu, set the parameters shown below and click **Apply**.



Rotate/Copy

Axis direction

☒ X
 ☐ Y
 ☐ Z

Origin X

0

m

Origin Y

0

m

Origin Z

0

m

Mouse pick

Angle

-45

°

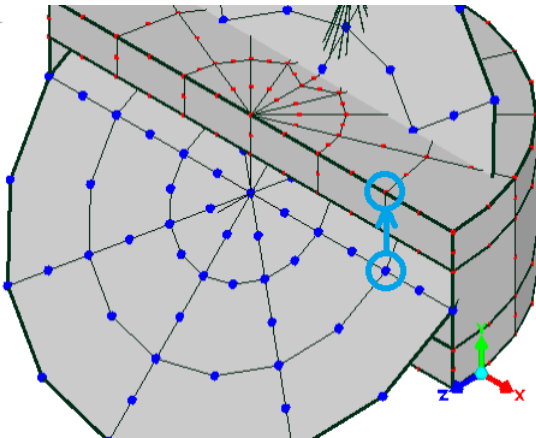
☐ Copy

Apply

Close

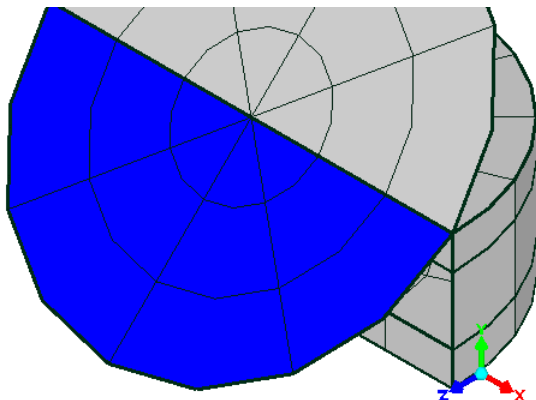
### Step 22

Change to **Select nodes** mode so that the nodes of the disk are selected, then drag one of them onto the corresponding node of the semi-circle as shown below. When the **Merge nodes?** prompt appears, click **Yes**.



### Step 23

Select the elements shown below and press the **delete** key to delete them.



## Step 24

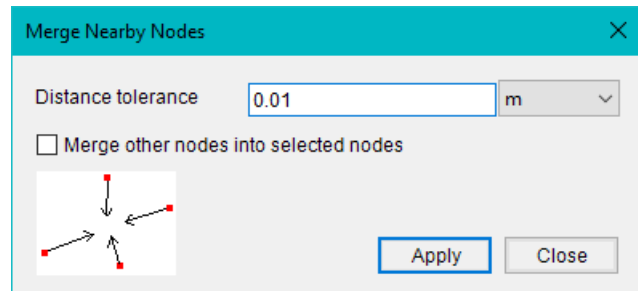
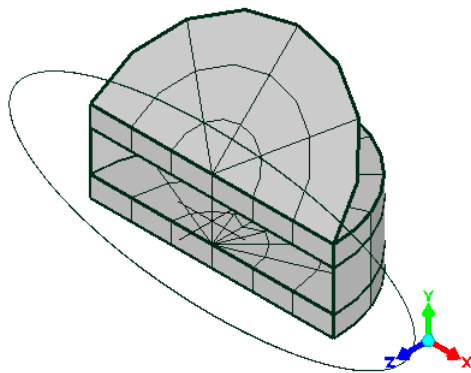
From the **Mesh tools** menu, open **Create** and click **Curve generator**. Click **Circle/ellipse** from the Predefined curves list, set the parameters shown below, and click **OK**. Next, set the other parameters as shown below and click **Apply**.

The first screenshot shows the 'Circle/ellipse' dialog box. It has two input fields: 'D1' with the value '10' and a unit dropdown set to 'm', and 'D2' with the value '4' and a unit dropdown set to 'm'. To the right is a diagram of an ellipse with a coordinate system (X and Y axes) and dimension lines indicating D1 as the horizontal width and D2 as the vertical height. At the bottom are 'OK' and 'Cancel' buttons.

The second screenshot shows the 'Curve Generator' dialog box. The title is 'Specify a strip of line elements parametrically as functions of p.' Under the 'Geometry' section, 'p initial' is '0' and 'p final' is '2 \* pi'. The parametric equations are: X = 5\*cos(p), Y = 2\*sin(p), and Z = 0. A unit dropdown is set to 'm'. Under the 'Mesh' section, 'Number of elements' is '16'. The 'Quadratic elements (line3)' checkbox is checked and circled in blue. Below it are two radio button options: 'Distance along curve is constant between adjacent nodes (uniform element size)' (unselected) and 'Change in p is constant between adjacent nodes' (selected). At the bottom, a 'Predefined curves' list shows: 'Straight line', 'Center and start point arc', 'Three point arc', 'Circle/ellipse' (highlighted), and 'Parabola'. 'Apply' and 'Close' buttons are at the bottom right.

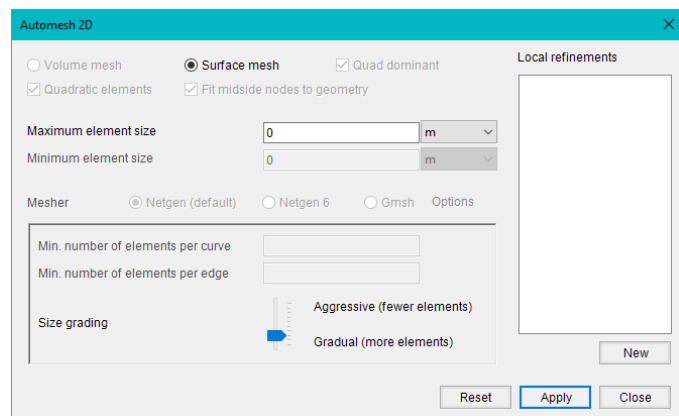
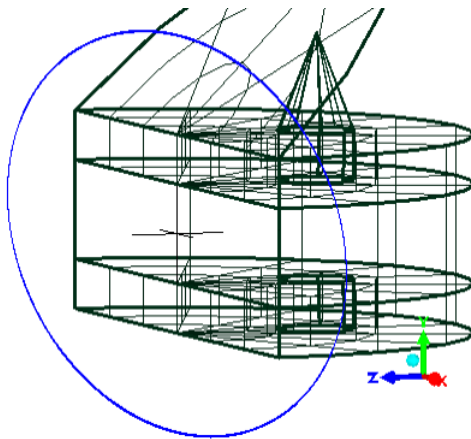
## Step 25

The previous step generated an open curve with two coincident end nodes not connected. To correct this, with nothing selected, click **Merge nearby nodes** in the **Mesh tools** menu, set the parameters shown below and click **Apply**.



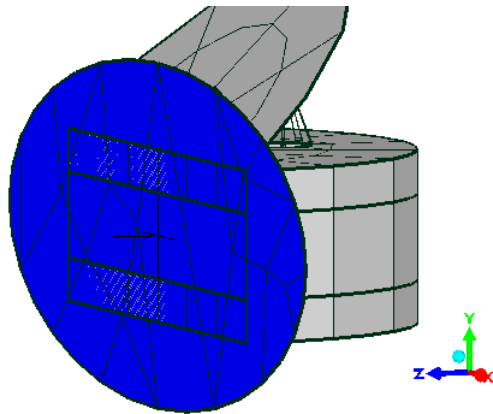
### Step 26

Select the 16 elements of the ellipse as shown below. A quick way to do this is turn off **Show element surfaces**, click the X arrowhead on the triad and use **Select nodes** mode to select them while viewed edge-on, then change to **Select elements** mode. Next, from the **Mesh tools** menu, click **Automesh 2D** and click **Apply**.



### Step 27

Select the meshed elements as shown below. A quick way to do this is using **Edge detecting selection**. From the **Mesh tools** menu, click **Move/copy**, set the parameters shown below and click **Apply**.



**Move/Copy** [X]

Coordinate system: Cartesian

$\Delta X$ : [ ] m

$\Delta Y$ : [ ] m

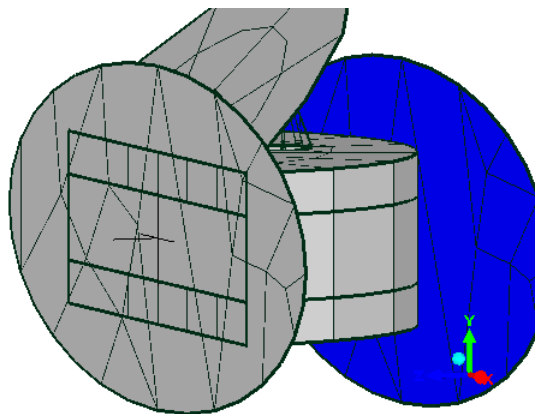
$\Delta Z$ : -3 m

☒ Copy

[Apply] [Close]

### Step 28

With the new elements selected, click **Scale/copy** in the **Mesh tools** menu, set the parameters shown below and click **Apply**.



**Scale/Copy** [X]

Origin X: 0 m

Origin Y: 0 m

Origin Z: 0 m

[Mouse pick]

Scale factor X: 0.5

Scale factor Y: 1

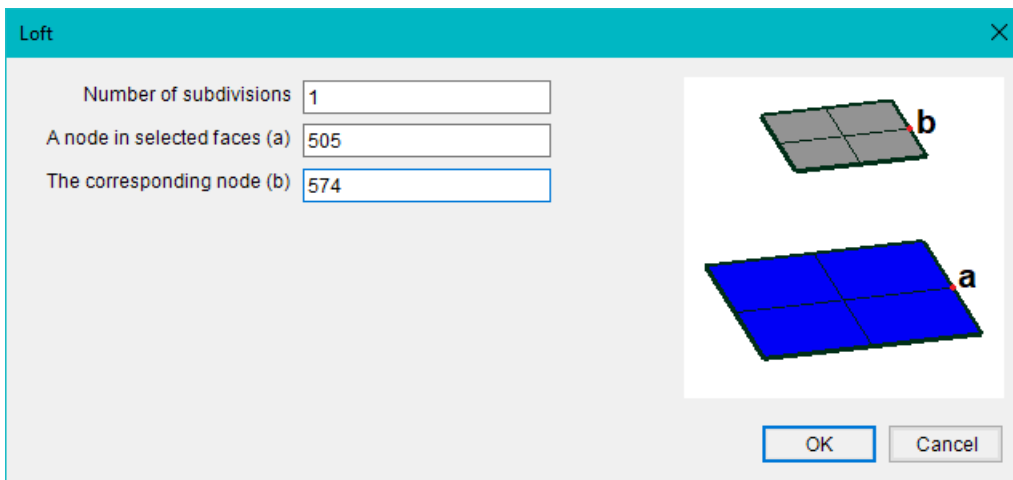
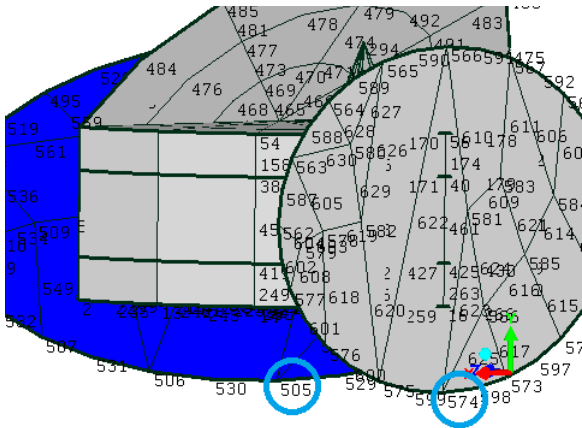
Scale factor Z: 1

☐ Copy

[Apply] [Close]

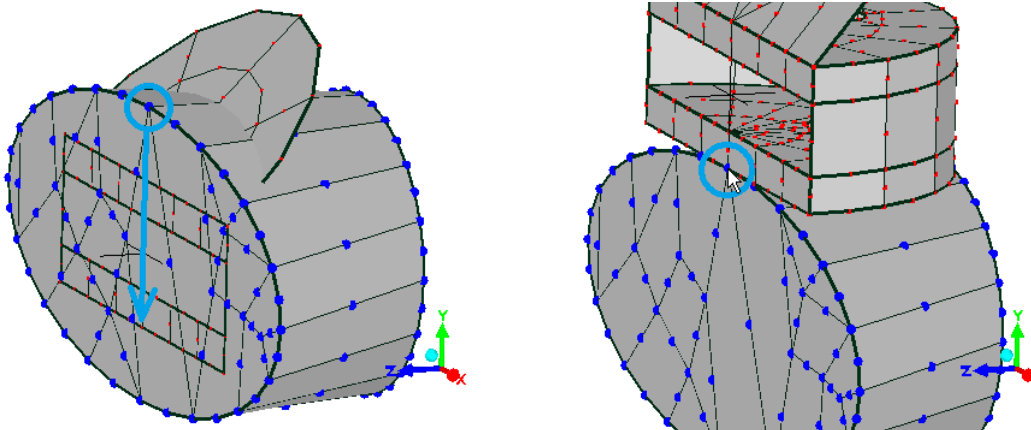
### Step 29

Select the faces on one side of the first ellipse as shown. Turn on **Show node numbers** in the **View** menu and orient the view so some numbers from both ellipses are visible. Next, click **Loft** in the **Mesh tools** menu and enter the number of any node in the first ellipse for **(a)**, and the number of its corresponding node in the second ellipse **(b)**. The node numbers shown below may be different in your model. Finally, click **OK** to generate a frustum. Note that the loft tool requires both surfaces to have the same mesh topology and the same difference between each pair of corresponding node numbers so you should always create one as a copy of the other.



### Step 30

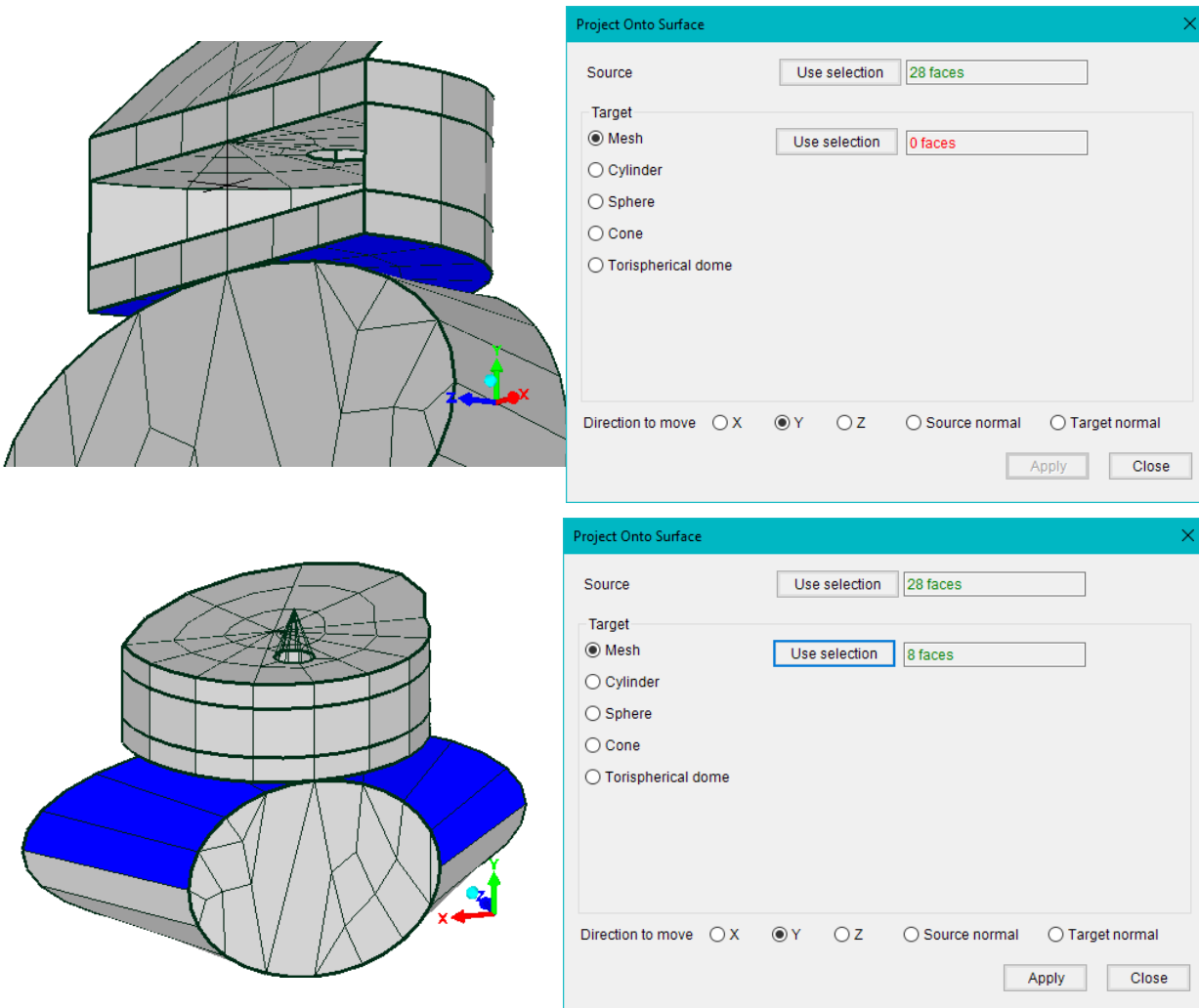
Select all the nodes of the frustum as shown below, then drag the top (+Y) center node onto the center node of the semi-circle as shown below. When the **Merge nodes?** prompt appears, click **No**.



### Step 31

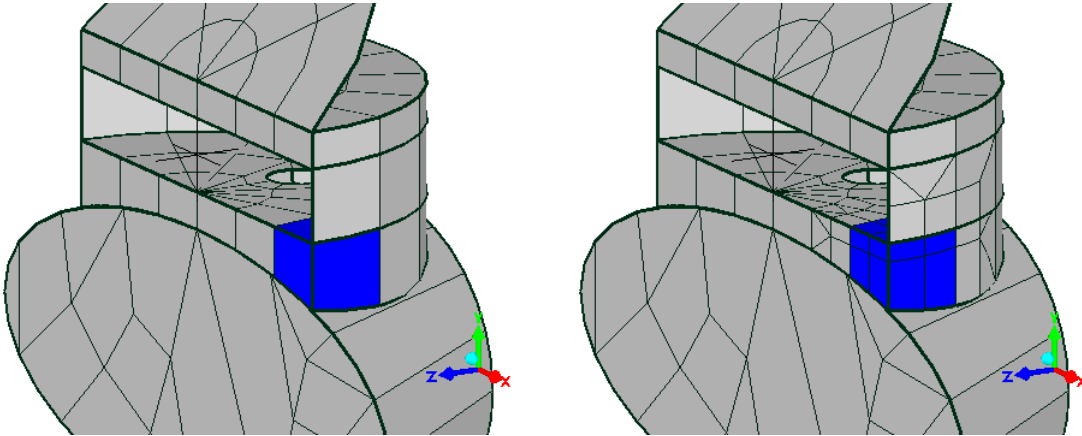
Select the 28 faces on the surface of the semi-circle as shown below. From the **Mesh tools** menu, click **Project onto surface** and set the **Mesh** and **Y** options as shown below. Leaving the window

open, select the 8 faces of the upper surface of the frustum as shown below and click **Use selection** in the **Target** box. Finally, click **Apply**. Although the two parts now have matching shapes, they are not connected to each other.



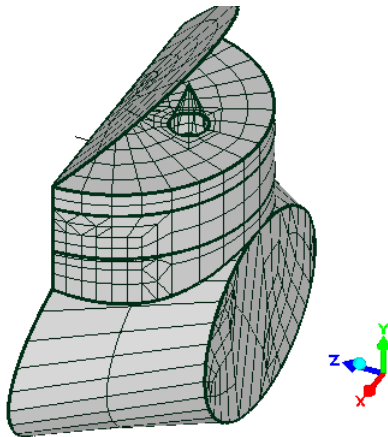
### Step 32

Select the element shown below. From the **Mesh tools** menu open **Refine** and click **x2**. This is local refinement.



### Step 33

With nothing selected, open **Refine** in the **Mesh tools** menu and click **x2** again. This is global refinement. If you are using the free version of Mecway, you will not be able to save the model after this step because it has more than 1000 nodes. In that case, use **Undo** first if you want to save it.




## 16.4 Solution


### Step 1

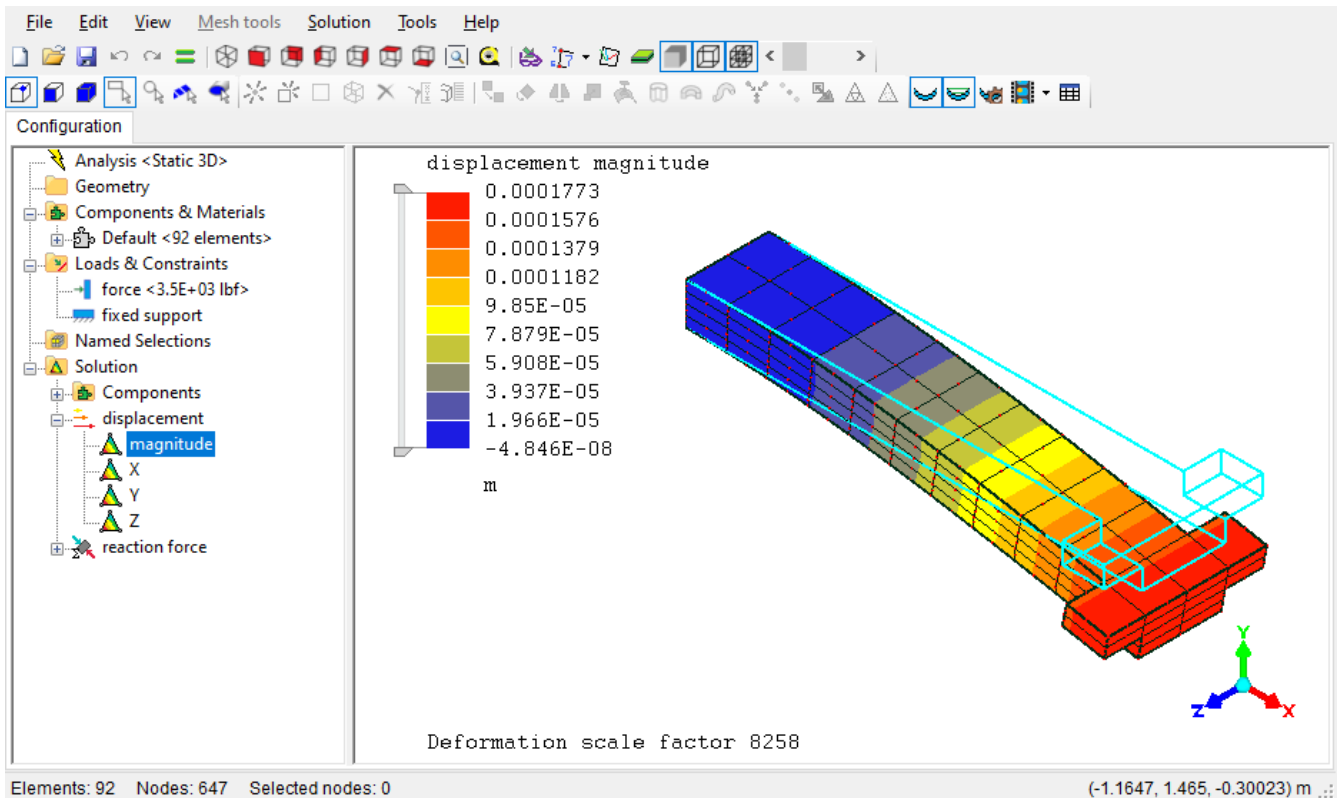
Open the file *Solution.liml* in the tutorials folder where Mecway has been installed.

In the solution section of the outline tree click on **displacement** → **magnitude**. The graphics display will update to show the results of the solved model. The units used in the legend on the display scale can be changed by clicking on them.

### Step 2

Click the  **Deformed view** tool-button to visualize an exaggerated displacement of the structure.


Click the  **Undeformed shape** to superimpose an outline of the undeformed shape onto the deformed mesh.




### Step 3

Right click **Solution** in the outline tree and choose **Stress and strain** → **stress** to add the 6 stress components to the solution.

### Step 4

Click the  table tool-button to display the displacements and stresses at each node in spreadsheet-style cells which you can then copy and paste into your own spreadsheet.

### Step 5

Use the  animation tool-button to animate the deflection. Smooth deformation between the two extremes of movement is simulated. You can choose the scale factor. This tool is especially useful for visualizing solutions in vibration analysis.

### Step 6

To return to the model that does not display the results, click any of the items in the outline tree that is not below **Solution**.

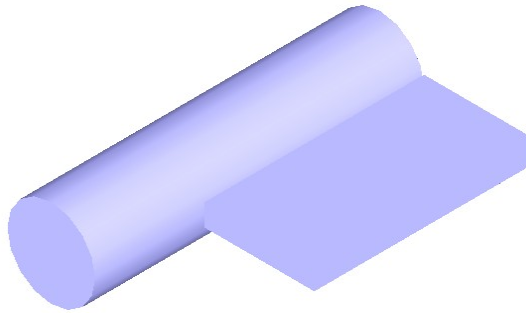
Now that you have some experience in manipulating the graphics of the model, it's time to learn how to create them from scratch. The next section will walk you through the case we have just studied.

## 16.5 Using STEP Files from CAD

This tutorial shows you how to make a model from multiple parts with different materials. To use your own models, you must first export each part from your CAD software as a separate STEP file.

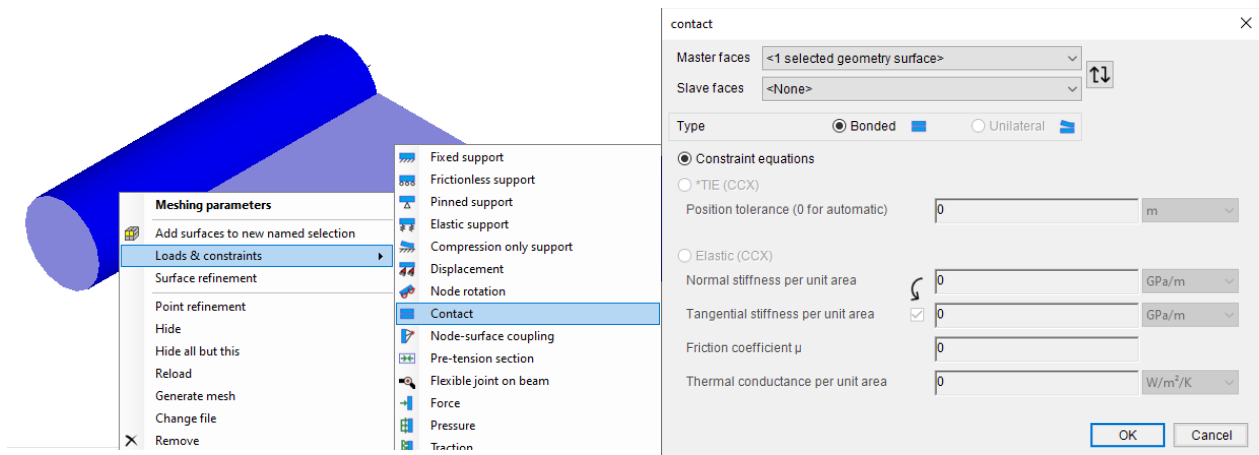
### Step 1

Use **File** → **Import** to import **plate.step** and **cylinder.step** from the tutorials folder where Mecway has been installed. For convenience, you can open both files at the same time by holding **Ctrl** while selecting them.



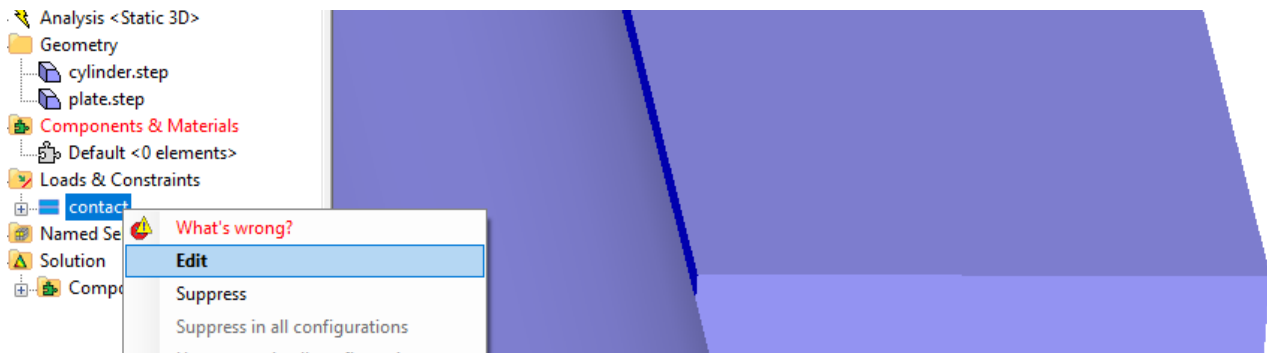
### Step 2

Right click the cylindrical surface, then click **Loads & constraints** → **Contact** and click **OK**.



### Step 3

Rotate the view to find the thin surface of the plate adjacent to the cylinder. Click this surface to select it. Right click **contact** in the outline tree and click **Edit**. Set Slave faces to **<1 selected geometry surface>** and click **OK**.

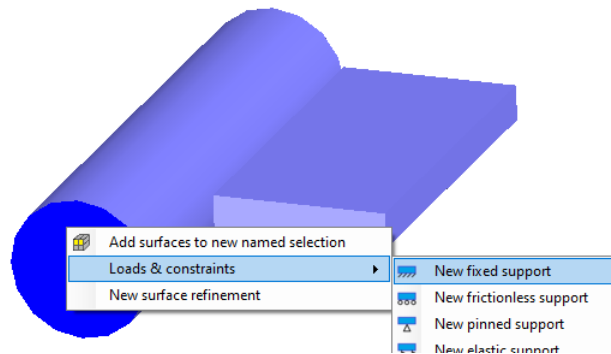


#### Step 4

Right click **Geometry** in the outline tree and click **Generate all meshes**.

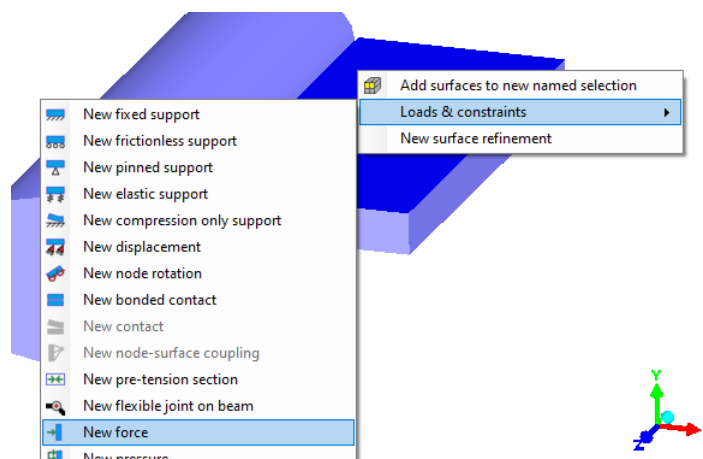
#### Step 5

Click the **Geometry** branch of the outline tree to return to the geometry view. Right click the circular end of the cylinder and click **Loads & constraints** → **Fixed support**, then click **OK**.



#### Step 6

Right click the large surface of the plate and click **Loads & constraints** → **Force**. Specify a Y component of -100 lbf, then click **OK**.



### Step 7

Right click the **plate <### elements linked>** component in the outline tree and click **Assign new material**. Select **Isotropic** and specify a **Young's modulus** of 70 GPa, then click **OK**.


### Step 8

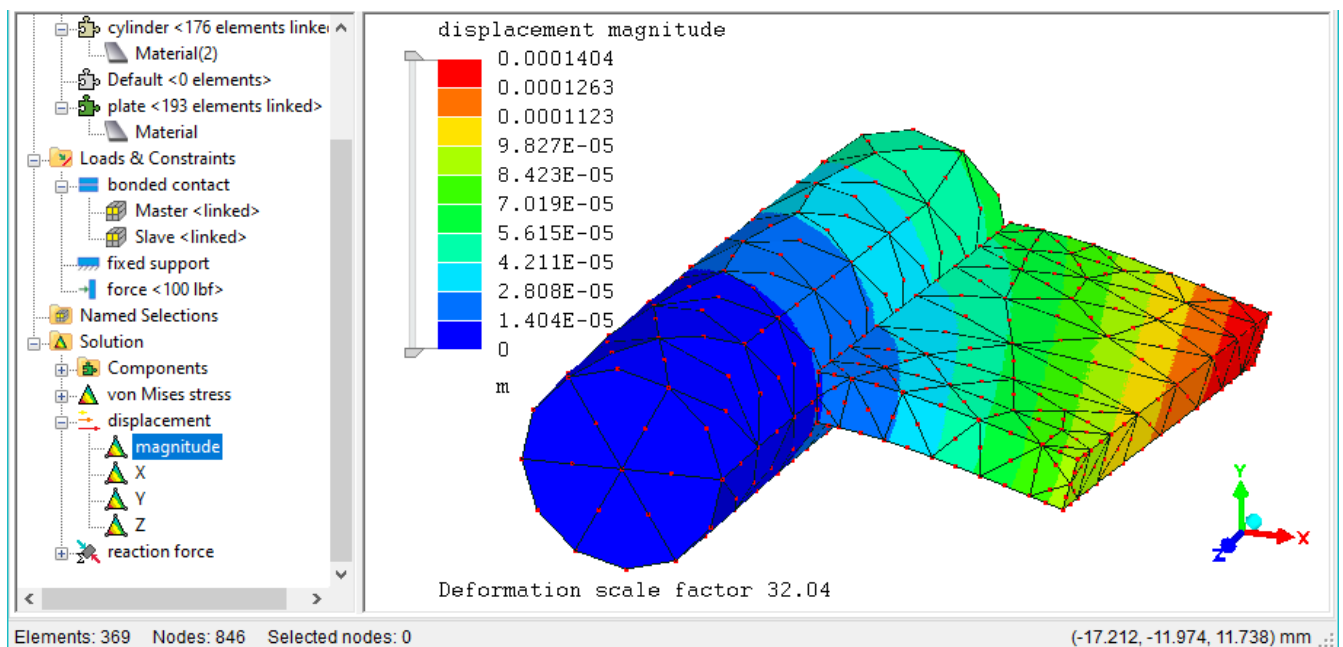
Right click the **cylinder <### elements linked>** component in the outline tree and click **Assign new material**. Select **Isotropic** and specify a **Young's modulus** of 210 GPa, then click **OK**.

### Step 9

Click solve .

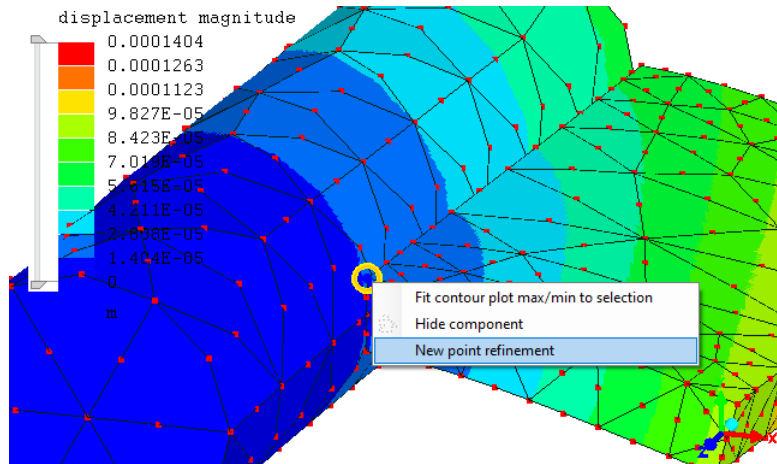
### Step 10

Click **magnitude** under **displacement** in the **Solution** branch of the outline tree then the **Animate** button  to show the deformation caused by the load.



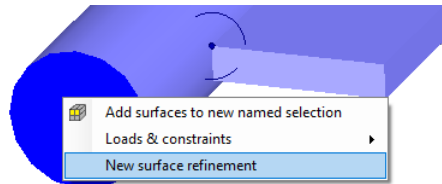
### Step 11

In **Select nodes** mode, right click a node as shown and click **Point refinement**, then click **OK**.



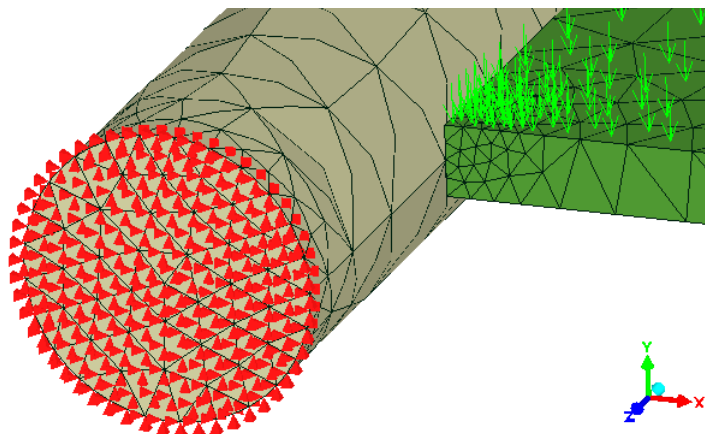
### Step 12

Return to the Geometry view and right click on the end of the cylinder and click **Surface refinement**. Enter a **Maximum element size** of 1 mm and click **OK**.



### Step 13

Right click **Geometry** in the outline tree and click **Generate all meshes**. Notice that those two areas now have a finer mesh.



If you are using the free version of Mecway, you will not be able to save the model after this step because it has more than 1000 nodes. In that case, use **Undo** first if you want to save it.

## 16.6 Mixed Materials

### Step 1

Open a new instance of Mecway.

### Step 2

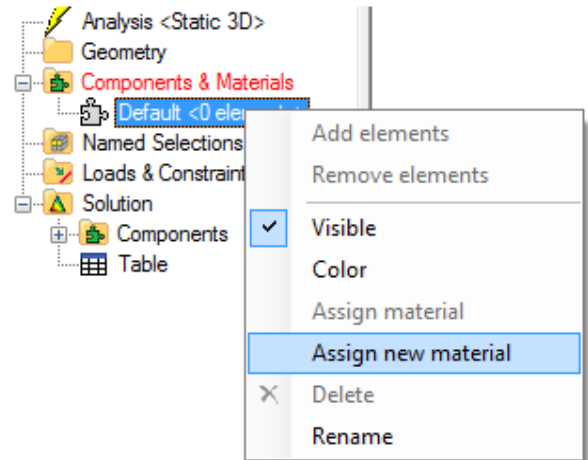
Click **Quick cube** 

### Step 3


Click **Refine x2** 

### Step 4

Right click the **Default** component in the outline tree and select **Assign new material**. Choose **Isotropic** and specify **Young's modulus** as 200 GPa, then click **OK**.

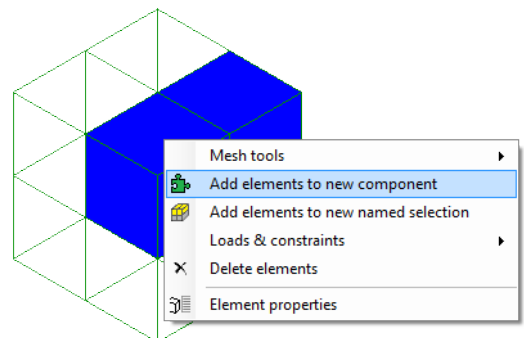


### Step 5


Change to **Select elements** mode  and select two of the 8 elements. Hold the **Ctrl** key while clicking on each element.

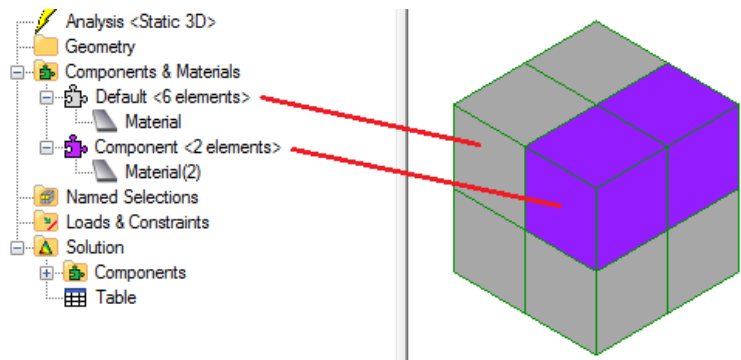
### Step 6

Right click one of the selected elements and choose **Add elements to new component**. The newly created Material(2) is a duplicate of the first material but you can change its properties independently.



### Step 7

Click on an empty space in the graphics area to clear the selection then click **Show element surfaces** . Notice that each material's component has a different color so you can easily distinguish them.



## 16.7 Assembly with Contact

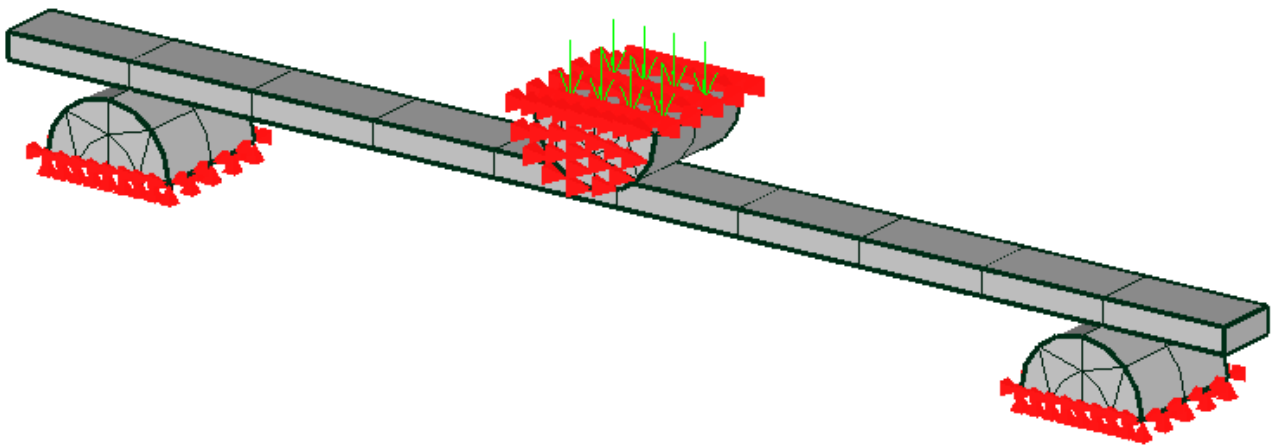
This tutorial is to help get a static solution for an assembly of parts that are held in place by contact with each other. It assumes you're familiar with general use of Mecway and contact.

All parts of an assembly must be initially in static equilibrium otherwise the solver won't converge. That means all loaded parts must have constraints to provide reaction forces. Contact is not reliable as an initial constraint and can leave some parts unconstrained or otherwise prevent the first time step from converging.

We will look at a model of a three-point bending test that fails and try various adjustments until it solves.

### Step 1

Open *AssemblyWithContact.liml* from the tutorials folder where Mecway has been installed.



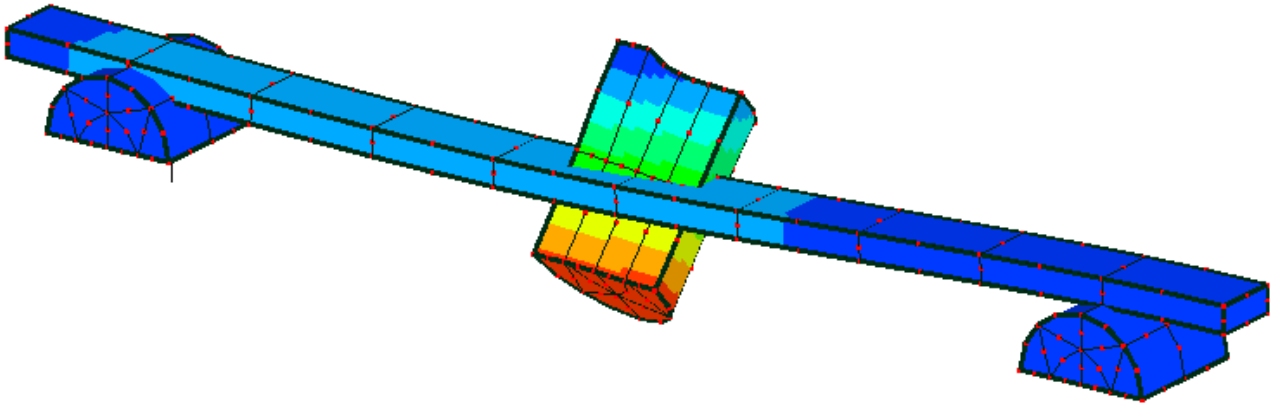
### Step 2

Solving the *Naive* configuration. The solver fails with the message

```
*ERROR: too many cutbacks best solution and residuals are in the frd file
```

### Step 3

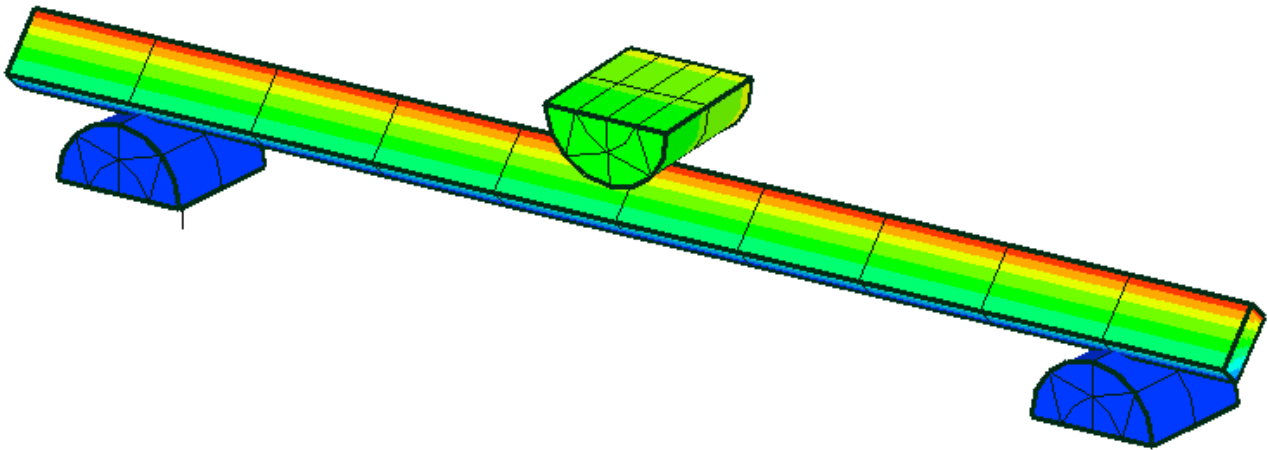
Make sure the problem is caused by rigid body motion and that the contact is otherwise working correctly. We do this by adding a **Fixed support** to each free part. Try this by solving the *Fixed\_support* configuration.



It solves quickly and the sample bends into the expected shape. The deformed shape looks strange at first because it's exaggerated due to the high deformation scale factor and low displacement magnitude. This is not the correct solution because of the artificial fixed support but it shows that the failure was likely due to lack of constraints.

#### Step 4

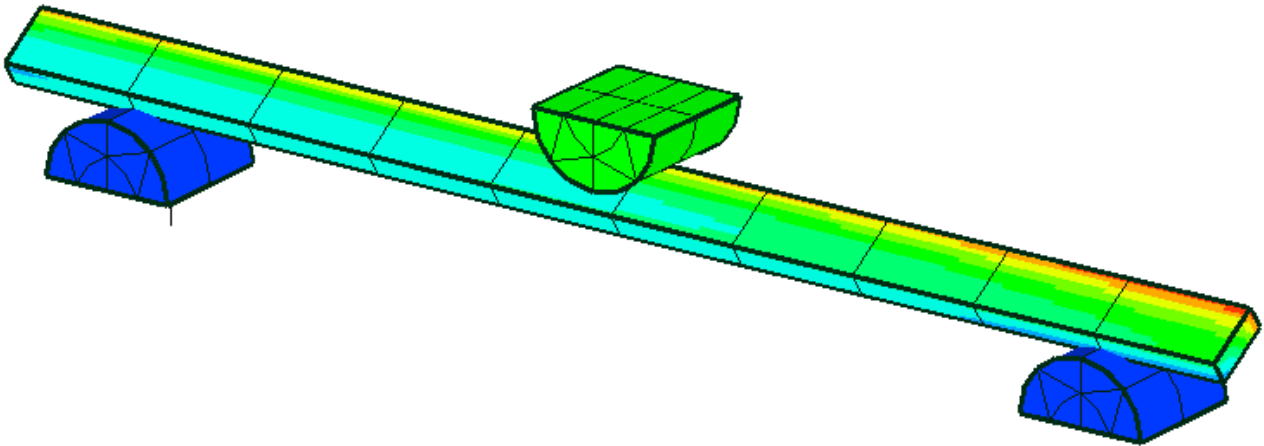
A failure at the first step is physically reasonable because the contacting surfaces are frictionless so the sample is free to slide out of place. We correct this by adding friction to constrain it. Try this by solving the *Friction* configuration.



Unfortunately, friction is not sufficient because the sample and/or loading pin are not being properly constrained by the contacts. You can see this in the non-converged solution. If displacements are too high, set a suitable deformation scale factor in **Solution** → **Deformed view settings**.

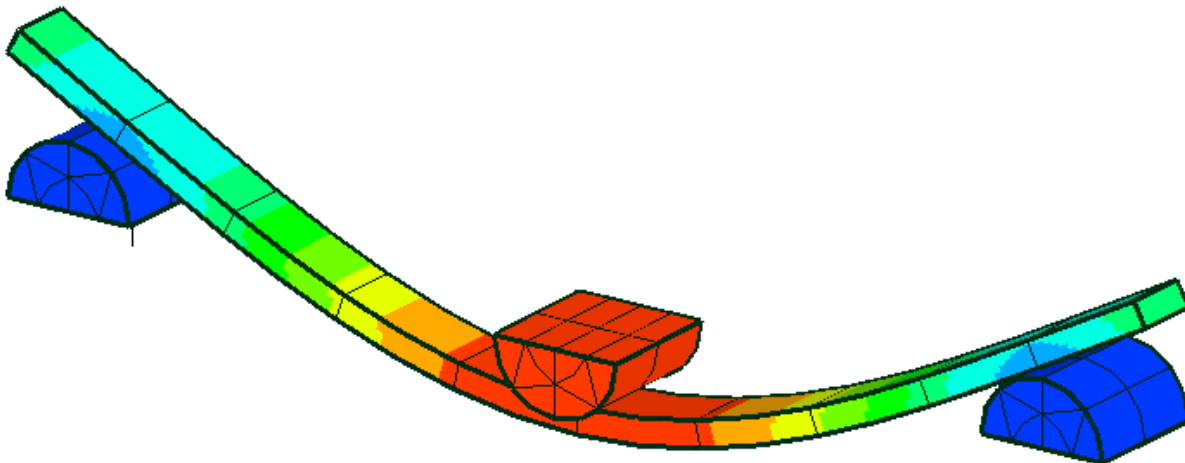
### Step 5

We next add **Elastic support** to the sample and loading pin to weakly constrain them. The elastic support's stiffness should be low enough that it won't significantly interfere with the model but high enough to prevent extreme displacements. Try this by solving the *Elastic\_support* configuration. It still fails.



### Step 6

A final way to help it converge is by starting with a smaller initial force. This is easy to accomplish without affecting the final force by ramping it quadratically ( $F_y = -1000 \times t^2$ ) instead of linearly ( $F_y = -1000 \times t$ ). Try this by solving the *Quadratic\_ramp* configuration.



Now it solves successfully. Notice that the displacement is symmetric despite the unsymmetrical locations of the elastic supports. That indicates they are weak enough not to have an obvious effect on the solution. To measure this more accurately we should check that the external forces at the elastic supports are negligible. Unfortunately, external force is not available with elastic support so you would need to replace it with fixed support connected to the model through elements with a low stiffness material.

## 16.8 Static Analysis of a Pressurized Cylinder

A cylinder of 2m radius, 10m length, 0.2m thickness, Young's modulus 200 GPa and Poisson ratio 0.285 will be analyzed to determine its hoop stress caused by an internal pressure of 100N/m<sup>2</sup>.

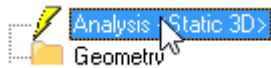
From shell theory, the circumferential or hoop stress for a thin cylinder of constant radius and uniform internal pressure is given by:

$$\sigma = (\text{pressure} \times \text{radius}) / \text{thickness}$$

$$\sigma = (100 \times 2) / 0.2$$

$$\sigma = 1000 \text{ N/m}^2$$

### Step 1

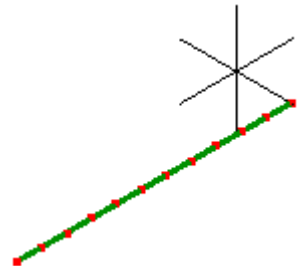


Check that the default analysis type is Static 3D.

### Step 2

**Mesh tools** → **Create** → **Curve generator...** click **Straight line** and enter the following values:

<b>X1</b>	2 m
<b>Y1</b>	0
<b>Z1</b>	0
<b>X2</b>	2 m
<b>Y2</b>	0
<b>Z2</b>	10 m
<b>Number of nodes</b>	12

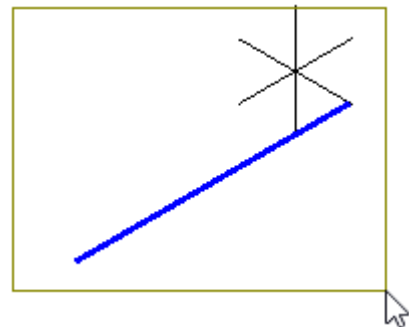


Click **OK** then **Apply** to generate the elements, and use **Fit to window**  to display them.

### Step 3

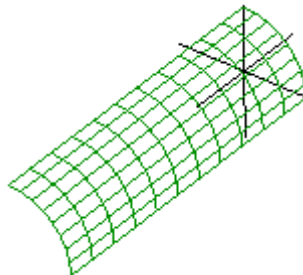
Due to the axial symmetry, only one quadrant will be modeled.

Activate **Edit** → **Edge detecting selection** and click on the line to select all the edge faces.

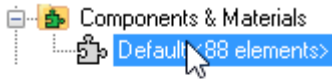


**Mesh tools** → **Revolve...**

<b>Axis direction</b>	Z
<b>Angle</b>	90 °
<b>Subdivisions</b>	8



#### Step 4



Right click, **Assign new material**

**Geometric tab**

select **Shell/membrane**

**Thickness** 0.2 m

**Mechanical tab**

select **Isotropic**

**Young's modulus** 200E9 Pa

**Poisson's ratio** 0.285

#### Step 5

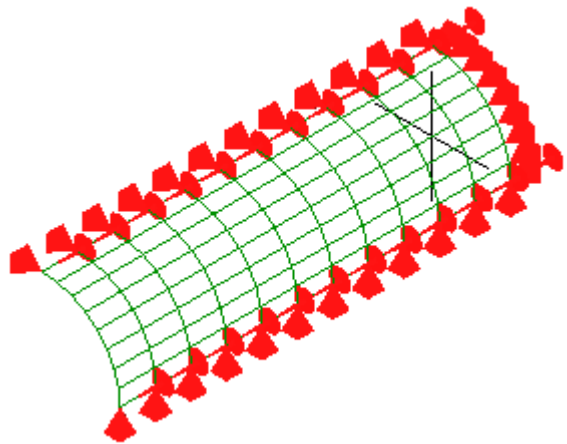
Because of the mirror symmetry, only one quadrant of the cylinder has been modeled. At the planes of mirror symmetry, the nodes must be constrained so that they do not move out of the plane. Also, no bending must occur in that plane of symmetry.

Activate **Edit** → **Edge detecting selection** again.

To enforce mirror symmetry at the edge in the YZ plane, click on that edge of the shell to select it, right click **Loads & Constraints** in the outline tree, click **Symmetry**, uncheck **Symmetry plane defined by nodes** then click **OK**.

Repeat for the edge in the ZX plane but this time, select the **Y** option in the **Symmetry** window.


To eliminate rigid body translation motion along the Z axis, click the edge in the XY plane, right click **Loads & Constraints** in the outline tree, click **Displacement**, select the **Z** option, then click **OK**.



### Step 6


Click the outer surface to select it. Right click **Loads & Constraints** in the outline tree, click **Pressure**, and set it to -100 Pa.

### Step 7


Click  to solve the model.

The results are listed in the outline tree below **Solution**. Click **von Mises stress**. to view the hoop stress.

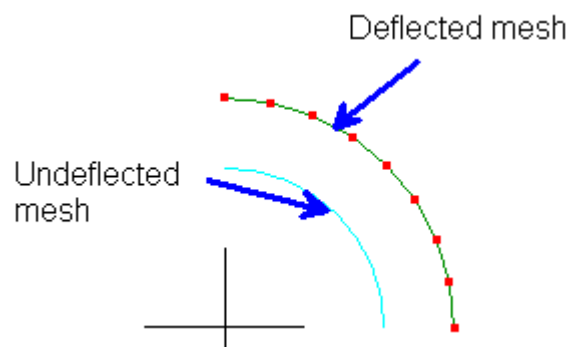
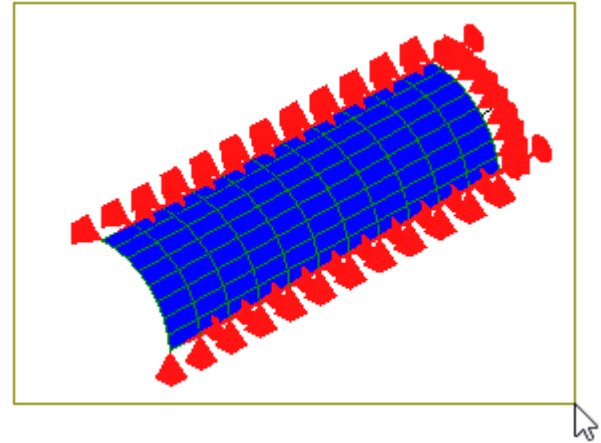
The first thing to check is the deformation. Errors in the applied constraints or loads can show up in the deformation shape.

Click the Z arrowhead  to view the XY plane parallel to the screen.

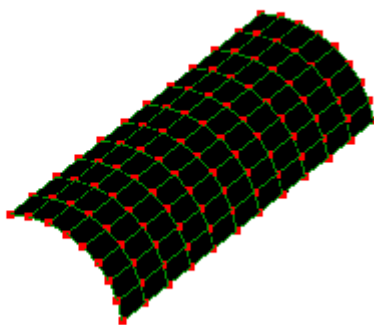
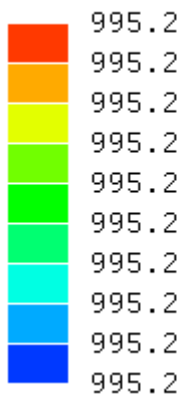
Click the **Deformed view** tool-button .

Click the **Undeformed shape** tool-button  to superimpose the undeformed geometry onto the deformed geometry.

Observe that the nodes in the YZ and ZX plane remain in these planes and no bending has occurred, this indicates that the constraints applied to enforce symmetry are working. The fact the deformation expands outward radially indicates that the internal pressure has been correctly applied.



von Mises Stress, Membrane



The computed hoop stress is only 0.48% different from the hand calculations. This is close enough not to need further mesh refinement.

## 16.9 Thermal Analysis of a Bar Being Cooled

A steel bar at an initial temperature of 250°C is suddenly immersed in an oil bath of temperature 50°C. We will find the time taken for the bar to cool to a temperature of 200°C.

### Step 1

Open *BarMesh.liml* from the tutorials folder where Mecway has been installed.

### Step 2

Double click **Analysis** at the top of the outline tree and specify:

<b>Analysis type</b>	<b>Thermal Transient</b>
<b>Time period</b>	400 s
<b>Time step</b>	10 s

### Step 3

Right click the *Default* component in the outline tree and click **Assign new material**. On the **Density** tab, select **Density** and enter 7860 kg/m<sup>3</sup>. On the **Thermal** tab, select **Isotropic** and specify **Thermal conductivity** as 40 W/m/K, and **Specific heat** as 500 J/kg/K.

### Step 4

Select all 16 elements, right click **Initial Conditions** in the outline tree, and click **Temperature**. Specify a value of 250°C.

### Step 5

Convective heat transfer takes place over the entire surface of the bar. Select all the outside faces, right click one of them, click **Convection** from the **Loads & Constraints** sub-menu, and specify the following values, then click **OK**.

<b>Ambient temperature</b>	50°C
<b>Heat transfer coefficient</b>	80 W/m <sup>2</sup> /K

convection

Apply to
<66 faces>

Ambient temperature

☒ Formula
☐ Table

50

? fx
°C

Time, t
s

Position, x,y,z
m

Heat transfer coefficient

☒ Formula
☐ Table

80


? fx
W/m²/K

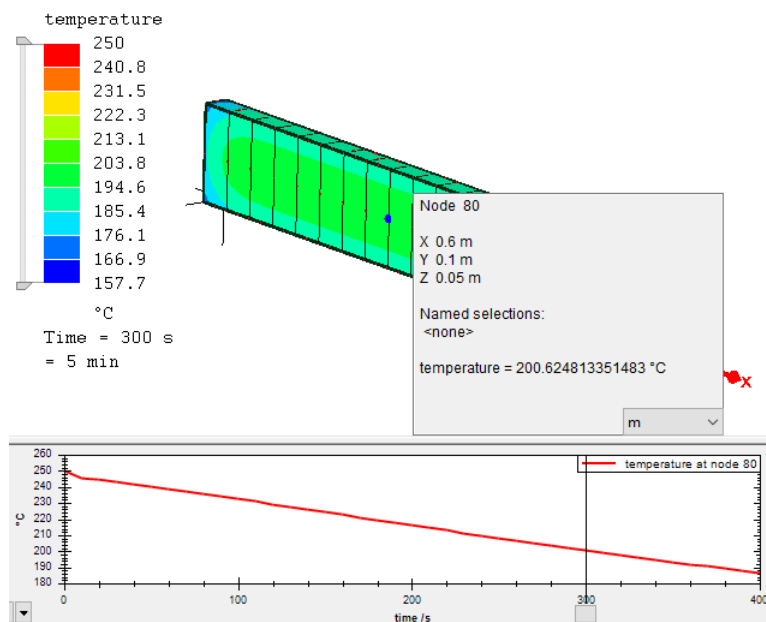
Time, t
s

Position, x,y,z
m

OK
Cancel

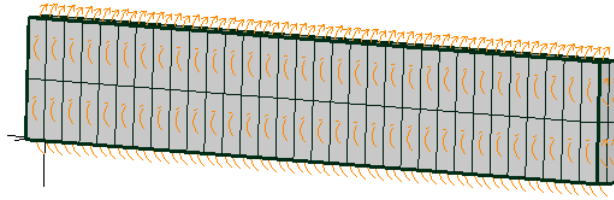
## Step 6

Click  to solve the model. The results are listed in the outline tree below **Solution**. Click **temperature** and, in **Select nodes** mode, click a node near the middle of the bar. A graph of its temperature history will be displayed in the timeline. You can see that the middle of the bar reaches 200°C at around 300 s.




### Step 7

There may be error due to the coarseness of the mesh so refine the mesh using **Mesh tools** → **Refine** → **x2**.



### Step 8

Click  to solve the model again. Click **temperature** under **Solution** in the outline tree and, in **Select nodes** mode, click the same node again. This shows that the temperature is still within about half a degree of what it was with the coarse mesh so the mesh is probably fine enough.

## 16.10 Thermal Stress

### Step 1

Start a new model. **File** → **New**

Double click **Analysis** in the outline tree and select **Thermal Steady State** then press **OK**.



### Step 2

Create a new hex8 element by clicking the Quick cube tool-button. 

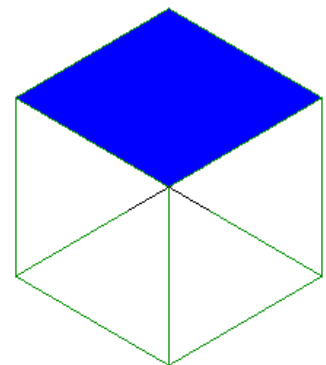
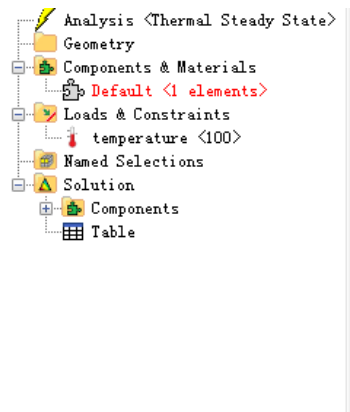
### Step 3

Activate select faces 

Click anywhere in the graphics area to clear the selection.

Right click any face and select **Loads & constraints** → **Temperature**

Enter the value 100. This constrains the temperature of that face to 100°C.



### Step 4

Right click another face and select **Loads & constraints** → **Convection**

**Ambient temperature**      20°C

**Heat transfer coefficient**      400 W/m<sup>2</sup>/K

### Step 5

Right click the component **Default <1 elements>** in the outline tree and choose **Assign new material**.

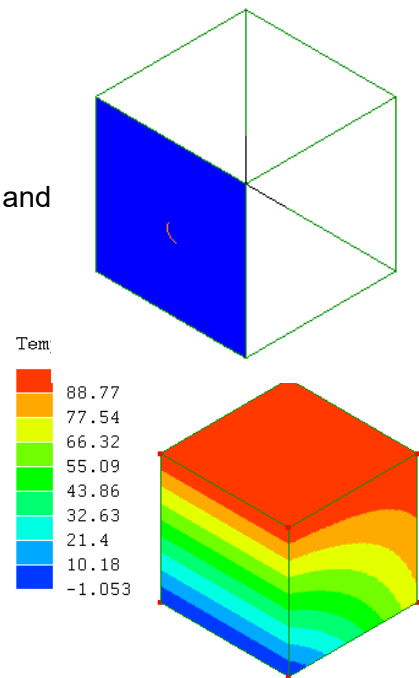
Select **Isotropic** and enter a **Thermal conductivity** of 40 W/m/K.

### Step 6

 **Solve.**

Click **Temperature** under the **Solution** branch of the outline tree. Your temperature values may be different depending on the cube's size.

Notice that some nodes may have an unrealistic temperature lower than the 20°C ambient. This is because of the extremely coarse mesh. In a practical problem, you would refine the mesh to improve accuracy.



### Step 7

Double click **Analysis**. Select **Static 3D**  **Analysis <Static 3D>**

### Step 8

 **Loads & Constraints** Right click **Loads & constraints** and select **Transfer temperatures from solution**

### Step 9

 **Loads & Constraints** Right click **Loads & constraints** and select **Thermal stress**  
**Reference temperature** 20°C

The reference temperature is the initial temperature of the nodes that are now at the temperatures which were transferred from the thermal analysis solution.

### Step 10

Right click **Material** and select **Edit**

**Mechanical tab**

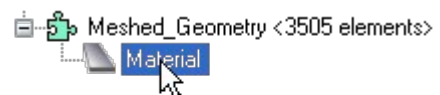
**Isotropic**

**Young's modulus**

200 GPa

**Thermal expansion coefficient**

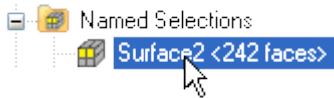
11E-06 1/K



**Density** tab  
**Density**

7800 kg/m<sup>3</sup>

### Step 11



Right click any face and choose **Loads & constraints** → **Fixed support**

### Step 12



Solve

The solution now shows the cube deformed by thermal expansion.

## 16.11 Free Vibration of a Cantilever Beam

We will find the lowest natural frequency of a cantilever beam of length 1.2 m, cross-section 0.2 m × 0.05 m, Young's modulus 200 GPa, and density 7860 kg/m<sup>3</sup>.

For slender beams, the following analytical equation is used to calculate the first natural frequency:

$$f = \frac{K_1}{2\pi} \sqrt{\frac{k}{3m}}$$

$$K_1 = 3.52$$

$$m = \text{density} \times \text{volume} \\ = 94.32 \text{ kg}$$

$$k = 3 \times E \times I / L^3$$

$$I = \text{moment of inertia of the cross-section} \\ = bh^3/12 \\ = 2.083 \times 10^{-6} \text{ m}^4$$

$$E = \text{Young's modulus}$$

$$L = \text{length}$$

$$f = 28.3 \text{ Hz}$$

### Step 1

Open *BarMesh.liml* from the tutorials folder where Mecway has been installed.

### Step 2

Double click **Analysis** at the top of the outline tree and specify:

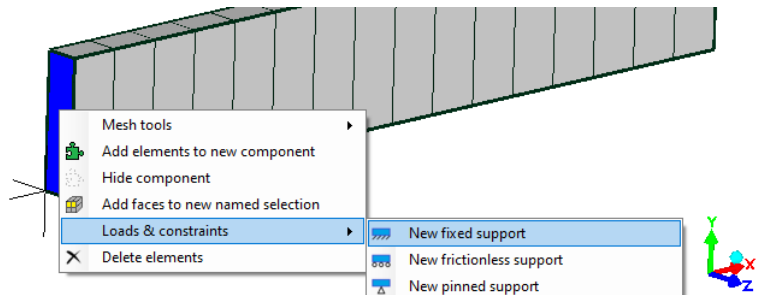
<b>Analysis type</b>	<b>Frequency</b>
<b>Number of modes</b>	3

### Step 3



Right click the *Default* component in the outline tree and click **Assign new material**. Select **Isotropic** and specify **Young's modulus** as 200 GPa. On the **Density** tab, select **Density** and enter 7860 kg/m<sup>3</sup>.

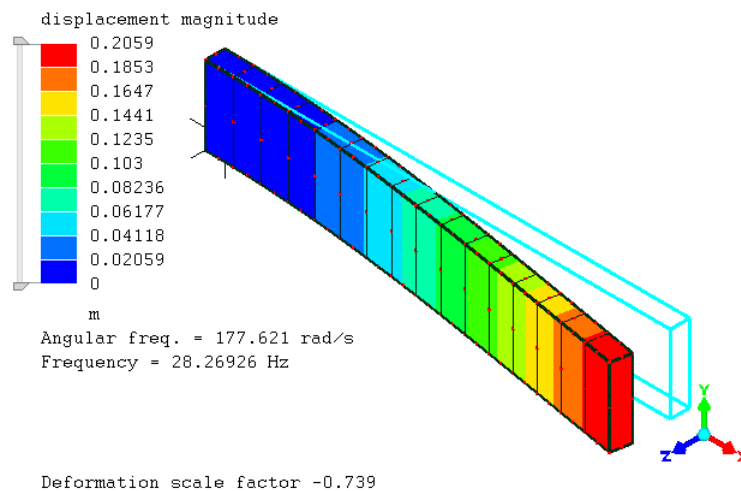
### Step 4

Right click one of the end faces, click **Fixed support** from the **Loads & Constraints** sub-menu, then click **OK**.



### Step 5

Click  to solve the model. The results are listed in the outline tree below **Solution**. The frequency for the lowest mode (**Mode 1**) matches the hand calculation result of 28.3 Hz. Click **Mode 1**, then the Animate button  to view the mode shape.



Note that the displacement and stress values of a modal analysis are only meaningful relative to each other. The scale of the values of a real structure depends on load history and damping which are not modeled by frequency analysis.

## 16.12 Dynamic Response of a Solid Part

### Step 1

Open *DynamicResponse.liml* from the tutorials folder where Mecway has been installed.

### Step 2

Double click **Analysis** at the top of the outline tree and specify:

**Time period** 0.003 s  
**Time step** 0.0004 s

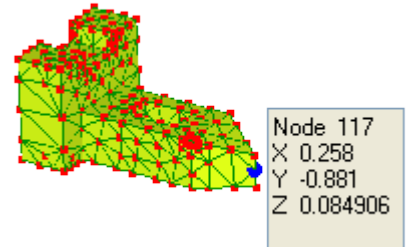
### Step 3

 Activate the **Select nodes** mode and select this node.

An applied force will ramp up linearly from 0 to a maximum 500 N then down to 0. At each time step Mecway will determine the force by interpolating between the specified values.

Right click **Loads & Constraints** and select **Force**. Under **X**, select **Table** and enter the following 3 rows:

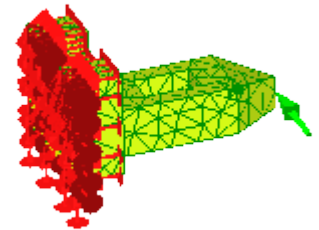
0	0
0.0015	500
0.003	0




### Step 4

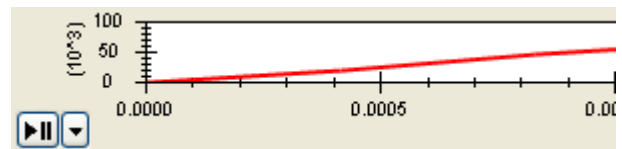


Right click Surface15 and select **Loads & constraints** → **Fixed support**



### Step 5

 Solve then select any of the results below the Solution group in the outline tree. Drag the slider on the timeline to display the results at the various time steps. Select a node to display a graph of its displacement.



## 16.13 Acoustic Analysis of an Organ Pipe

We will predict the musical note produced by an 8 ft long 'open wood' organ pipe. An actual pipe would be rectangular in section, but here we make only a 2D model. The pipe will be 120 mm wide with a 30 mm mouth at one end where air is blown in, and an opening at the other end for the air to exit. The wood is 10 mm thick.

In acoustic analysis the air within the pipe needs to be meshed and not the pipe itself. The mesh needs to include a large volume of the air in the room outside the pipe which is connected to the mouth and the open end. This scheme allows Mecway to model leakages and openings. We are interested only in the resonant modes of the organ pipe, but Mecway will also predict resonances in the air outside. An important part of this case study, therefore, is to show how some understanding of the physics is required in order to interpret the results and identify the appropriate modes.

A rough calculation will be made for the expected frequency.

An end correction factor must be added to the length of the pipe. This correction factor is equal to the width of the pipe. So the effective length of the pipe is  $2.4384 \text{ m} + 0.12 \text{ m} = 2.5584 \text{ m}$

When  $1/2$  a wavelength fits into the pipe it will be of wavelength  $\lambda = 2 \times 2.5584 \text{ m} = 5.1168 \text{ m}$

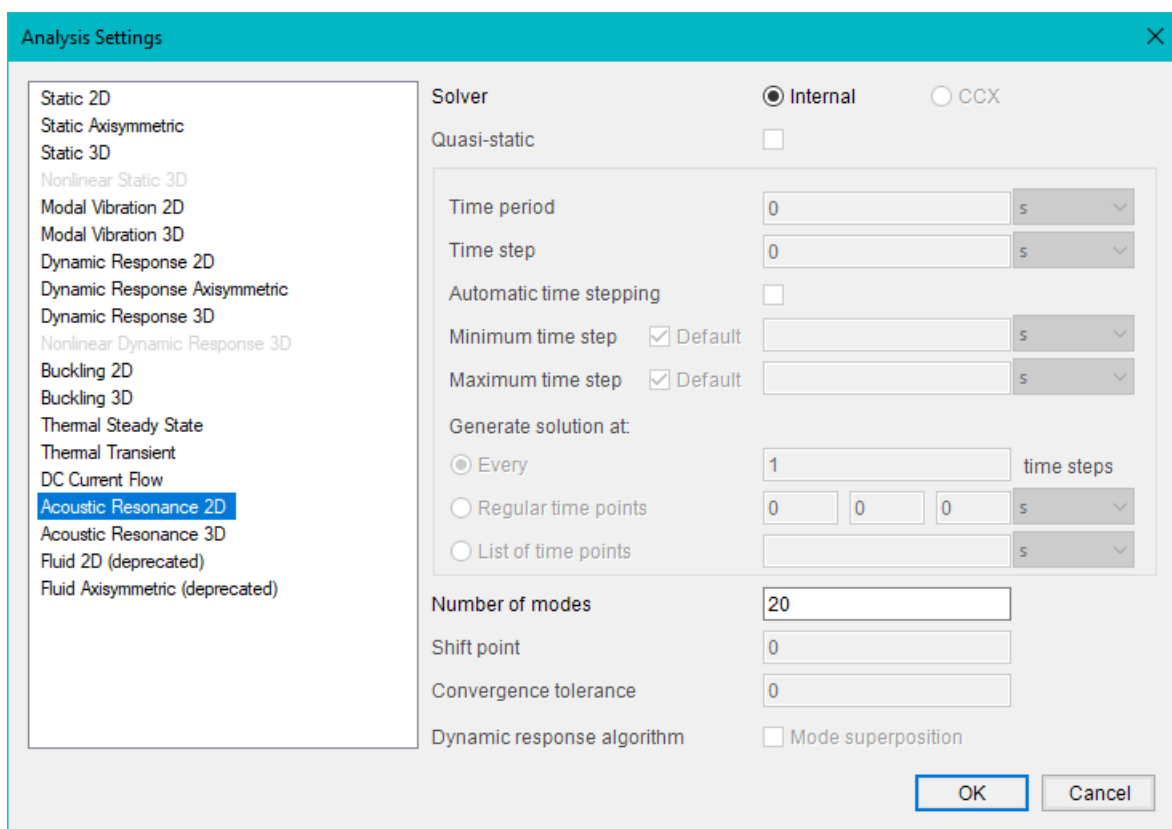
The speed of sound in air at  $15^\circ\text{C}$  is  $340 \text{ m/s}$ , so the expected frequency will be  $f = v/\lambda = 340/5.1168 = 66.45 \text{ Hz}$ .

### Step 1

Open the file *AcousticAnalysisOfAnOrganPipe.liml* in the tutorials folder where Mecway has been installed. This is just the mesh.

### Step 2

Double click **Analysis** at the top of the outline tree, then select **Acoustic Resonance 2D** and set **Number of modes** to 20.




To ascertain the fundamental pitch we need to examine the results for a number of pitches. We will also look at harmonics.

### Step 3

Right click the **Default** component in the outline tree and click **Assign new material**. Select **Isotropic** and specify **Speed of sound** as  $340 \text{ m/s}$ .

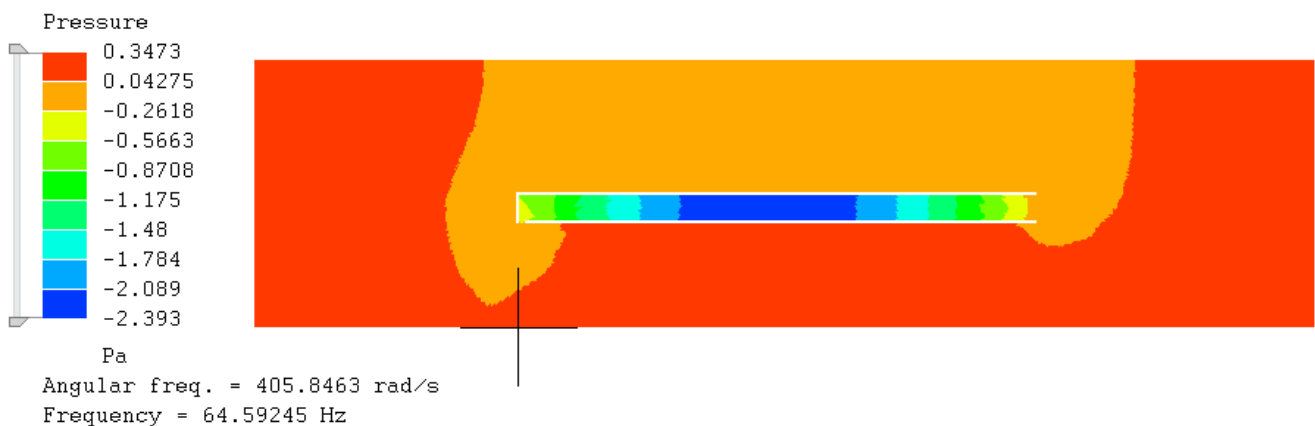
#### Step 4

No special boundary conditions are required because the edge of the mesh represents a hard surface. Click  to solve the model.

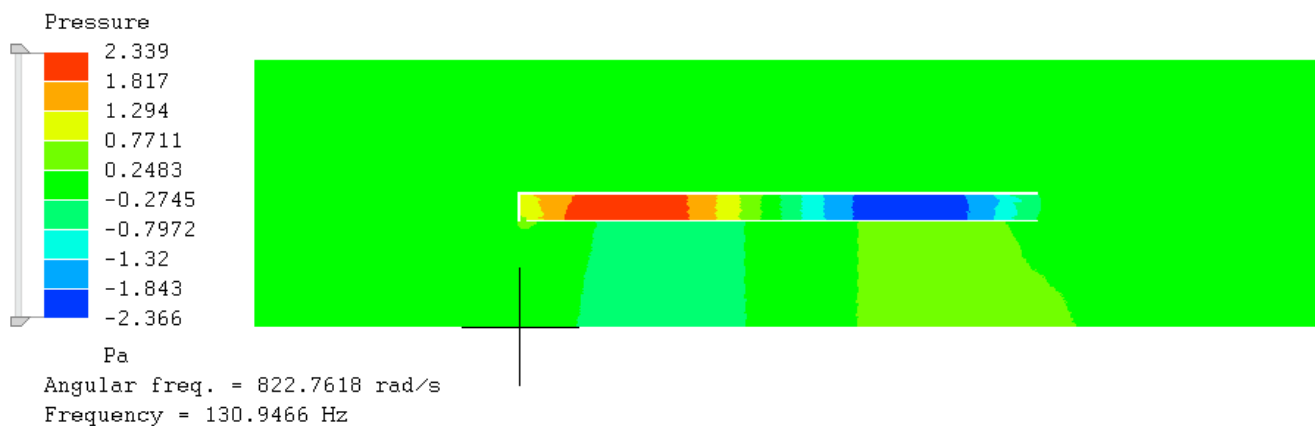
The modal frequencies are listed in the outline tree below Solution. Click the nodes for a read-out of the field variable there.

The subtlety in this case study is in interpreting the results. From the 20 calculated modes we want to pick out the fundamental pitch of the pipe itself. Click through the various modes and observe that most show red, yellow and blue patches for the room air elements.

The 4th mode at 64.59 Hz is different because the 'room' is almost at constant pressure (color). Zoom in on the pipe and notice that the magnitude of the maximum pressure inside the pipe is much higher than anywhere outside the pipe. This identifies mode 4 as the fundamental pitch of the organ pipe. Half a wavelength fits inside the pipe. The frequency matches the hand calculated value very well.



Similarly Mode 8 at 131 Hz has an almost constant pressure in the room and high, variable pressure in the pipe. This is the first harmonic at twice the frequency of mode 4, sounding one octave higher. A full wavelength fits inside the pipe.



## 16.14 Constraint Equations

#### Step 1

Open *ConstraintEquations.liml* from the tutorials folder where Mecway has been installed.

### Step 2

Right click **Loads & Constraints** in the outline tree and click **Constraint equation**

Click **New term** and specify:

<b>Coefficient</b>	1
<b>DOF</b>	Displacement in X
<b>Node</b>	201

Click **OK**

Click **New term** again and specify:

<b>Coefficient</b>	-1
<b>DOF</b>	Displacement in X
<b>Node</b>	36


Click **OK**

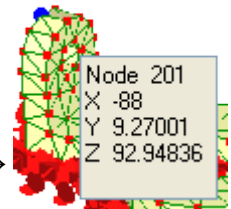
The constraint equation should now read

$$1 u_{x_{201}} - 1 u_{x_{36}} = 0$$


Click **OK**.

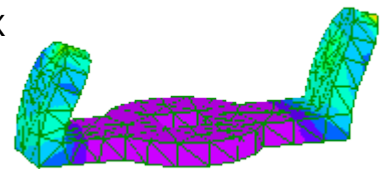
### Step 3

In **Select nodes** mode (  ), right click this node and select **Loads & constraints** → **Force**. Specify the **X** component to be 100 N.



### Step 4

Click  to solve the model. The two nodes have the same X displacement as if they are connected by a rigid, non-rotating bar.



## License Agreements

### 17.1 Mecway End User License Agreement (EULA)

1. This agreement is between you and Mecway Limited (New Zealand) and relates to the Mecway finite element analysis package including third party software libraries and any internet and support services provided for it, collectively referred to as "the software".

2. THE SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES FOR MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

3. You must indemnify and defend Mecway Limited, its directors, staff, contractors and suppliers and the developers of and contributors to any portion of the software from any claims related to your use of the software.

4. You may not redistribute, reverse engineer or disassemble the Intel Math Kernel Library included with the software.

5. Parts of the software are used under license from their respective copyright holders.

6. Your use of Open CASCADE Technology 6.3 used by the software is covered by the Open CASCADE Technology Public License version 6.3.

7. Open CASCADE Technology 7.3 used by the software is covered by the GNU Lesser General Public License Version 2.1 with with additional exception.

8. The Netgen, Pthreads-win32 and ZedGraph libraries used by the software are covered by the GNU Lesser General Public License Version 2.1.

### 17.2 ARPACK

The ARPACK software library included with Mecway is used under the following license.

Copyright (c) 1996-2008 Rice University.

Developed by D.C. Sorensen, R.B. Lehoucq, C. Yang, and K. Maschhoff.

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer listed in this license in the documentation and/or other materials provided with the distribution.

- Neither the name of the copyright holders nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

### 17.3 SlimDX

The SlimDX graphics library included with Mecway is used under the following license.

Copyright (c) 2007-2010 SlimDX Group

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 17.4 Bouncy Castle

The Bouncy Castle software library included with Mecway is used under the following license.

Copyright (c) 2000 - 2015 The Legion of the Bouncy Castle Inc. (<http://www.bouncycastle.org>)

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

### 17.5 Netgen, Pthreads-win32, ZedGraph, and OCCT version 7.3

The Netgen software library included with Mecway is copyright by Joachim Schoeberl. Its use is covered by the GNU Lesser General Public License Version 2.1.

The Pthreads-win32 software library release 2.8.0 included with Mecway is subject to the following copyright:

Pthreads-win32 - POSIX Threads Library for Win32

Copyright(C) 1998 John E. Bossom

Copyright(C) 1999,2006 Pthreads-win32 contributors

Copying, distribution and modifications of it must be done in accordance with the GNU Lesser General Public License version 2.1.

The ZedGraph software library included with Mecway is copyright by John Champion. Its use is covered by the GNU Lesser General Public License Version 2.1.

Open CASCADE Technology version 7.3 is included with Mecway. The source code is available from <https://www.opencascade.com/>. Its use is covered by the GNU Lesser General Public License Version 2.1 with additional exception. The exception is:

*Open CASCADE Exception (version 1.0) to GNU LGPL version 2.1.*

*The object code (i.e. not a source) form of a "work that uses the Library" can incorporate material from a header file that is part of the Library. As a special exception to the GNU Lesser General Public License version 2.1, you may distribute such object code incorporating material from header files provided with the Open CASCADE Technology libraries (including code of CDL generic classes) under terms of your choice, provided that you give prominent notice in supporting documentation to this code that it makes use of or is based on facilities provided by the Open CASCADE Technology software.*

Netgen has been modified for use with Mecway. You can obtain a complete machine-readable copy of the modified source code by sending a written request to the address for service of Mecway Limited (NZBN/GLN 9429030014299). Your request must include a return email address and be sent within 3 years of the date you received Netgen. For your convenience, you may also be able to obtain this source code by sending a request to [support@mecway.com](mailto:support@mecway.com)

The original source code is available from:

<http://sourceforge.net/projects/netgen-mesher/files/>

If you wish to modify and/or recompile the Pthreads-win32 library, the source code can be obtained from:

<http://sourceware.org/pthreads-win32/>

If you wish to modify and/or recompile the ZedGraph library, you can download the source code from:

<http://sourceforge.net/projects/zedgraph>

## 17.6 GNU Lesser General Public License

### GNU LESSER GENERAL PUBLIC LICENSE

Version 2.1, February 1999

Copyright (C) 1991, 1999 Free Software Foundation, Inc.

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

[This is the first released version of the Lesser GPL. It also counts

as the successor of the GNU Library Public License, version 2, hence

the version number 2.1.]

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public Licenses are intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users.

This license, the Lesser General Public License, applies to some specially designated software packages—typically libraries—of the Free Software Foundation and other authors who decide to use it. You can use it too, but we suggest you first think carefully about whether this license or the ordinary General Public License is the better strategy to use in any particular case, based on the explanations below.

When we speak of free software, we are referring to freedom of use, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish); that you receive source code or can get it if you want it; that you can change the software and use pieces of it in new free programs; and that you are informed that you can do these things.

To protect your rights, we need to make restrictions that forbid distributors to deny you these rights or to ask you to surrender these rights. These restrictions translate to certain responsibilities for you if you distribute copies of the library or if you modify it.

For example, if you distribute copies of the library, whether gratis or for a fee, you must give the recipients all the rights that we gave you. You must make sure that they, too, receive or can get the source code. If you link other code with the library, you must provide complete object files to the recipients, so that they can relink them with the library after making changes to the library and recompiling it. And you must show them these terms so they know their rights.

We protect your rights with a two-step method: (1) we copyright the library, and (2) we offer you this license, which gives you legal permission to copy, distribute and/or modify the library.

To protect each distributor, we want to make it very clear that there is no warranty for the free library. Also, if the library is modified by someone else and passed on, the recipients should know that what they have is not the original version, so that the original author's reputation will not be affected by problems that might be introduced by others.

Finally, software patents pose a constant threat to the existence of any free program. We wish to make sure that a company cannot effectively restrict the users of a free program by obtaining a restrictive license from a patent holder. Therefore, we insist that any patent license obtained for a version of the library must be consistent with the full freedom of use specified in this license.

Most GNU software, including some libraries, is covered by the ordinary GNU General Public License. This license, the GNU Lesser General Public License, applies to certain designated libraries, and is quite different from the ordinary General Public License. We use this license for certain libraries in order to permit linking those libraries into non-free programs.

When a program is linked with a library, whether statically or using a shared library, the combination of the two is legally speaking a combined work, a derivative of the original library. The ordinary General Public License therefore permits such linking only if the entire combination fits its criteria of freedom. The Lesser General Public License permits more lax criteria for linking other code with the library.

We call this license the "Lesser" General Public License because it does Less to protect the user's freedom than the ordinary General Public License. It also provides other free software developers Less of an advantage over competing non-free programs. These disadvantages are the reason we use the ordinary General Public License for many libraries. However, the Lesser license provides advantages in certain special circumstances.

For example, on rare occasions, there may be a special need to encourage the widest possible use of a certain library, so that it becomes a de-facto standard. To achieve this, non-free programs must be allowed to use the library. A more frequent case is that a free library does the same job as widely used non-free libraries. In this case, there is little to gain by limiting the free library to free software only, so we use the Lesser General Public License.

In other cases, permission to use a particular library in non-free programs enables a greater number of people to use a large body of free software. For example, permission to use the GNU C Library in non-free programs enables many more people to use the whole GNU operating system, as well as its variant, the GNU/Linux operating system.

Although the Lesser General Public License is Less protective of the users' freedom, it does ensure that the user of a program that is linked with the Library has the freedom and the wherewithal to run that program using a modified version of the Library.

The precise terms and conditions for copying, distribution and modification follow. Pay close attention to the difference between a "work based on the library" and a "work that uses the library". The former contains code derived from the library, whereas the latter must be combined with the library in order to run.

#### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License Agreement applies to any software library or other program which contains a notice placed by the copyright holder or other authorized party saying it may be distributed under the terms of this Lesser General Public License (also called "this License"). Each licensee is addressed as "you".

A "library" means a collection of software functions and/or data prepared so as to be conveniently linked with application programs (which use some of those functions and data) to form executables.

The "Library", below, refers to any such software library or work which has been distributed under these terms. A "work based on the Library" means either the Library or any derivative work under copyright law: that is to say, a work containing the Library or a portion of it, either verbatim or with modifications and/or translated straightforwardly into another language. (Hereinafter, translation is included without limitation in the term "modification".)

"Source code" for a work means the preferred form of the work for making modifications to it. For a library, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the library.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running a program using the Library is not restricted, and output from such a program is covered only if its contents constitute a work based on the Library (independent of the use of the Library in a tool for writing it). Whether that is true depends on what the Library does and what the program that uses the Library does.

1. You may copy and distribute verbatim copies of the Library's complete source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and distribute a copy of this License along with the Library.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Library or any portion of it, thus forming a work based on the Library, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) The modified work must itself be a software library.

b) You must cause the files modified to carry prominent notices stating that you changed the files and the date of any change.

c) You must cause the whole of the work to be licensed at no charge to all third parties under the terms of this License.

d) If a facility in the modified Library refers to a function or a table of data to be supplied by an application program that uses the facility, other than as an argument passed when the facility is invoked, then you must make a good faith effort to ensure that, in the event an application does not supply such function or table, the facility still operates, and performs whatever part of its purpose remains meaningful.

(For example, a function in a library to compute square roots has a purpose that is entirely well-defined independent of the application. Therefore, Subsection 2d requires that any application-supplied function or table used by this function must be optional: if the application does not supply it, the square root function must still compute square roots.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Library, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Library, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Library.

In addition, mere aggregation of another work not based on the Library with the Library (or with a work based on the Library) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may opt to apply the terms of the ordinary GNU General Public License instead of this License to a given copy of the Library. To do this, you must alter all the notices that refer to this License, so that they refer to the ordinary GNU General Public License, version 2, instead of to this License. (If a newer version than version 2 of the ordinary GNU General Public License has appeared, then you can specify that version instead if you wish.) Do not make any other change in these notices.

Once this change is made in a given copy, it is irreversible for that copy, so the ordinary GNU General Public License applies to all subsequent copies and derivative works made from that copy.

This option is useful when you wish to copy part of the code of the Library into a program that is not a library.

4. You may copy and distribute the Library (or a portion or derivative of it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange.

If distribution of object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place satisfies the requirement to distribute the source code, even though third parties are not compelled to copy the source along with the object code.

5. A program that contains no derivative of any portion of the Library, but is designed to work with the Library by being compiled or linked with it, is called a "work that uses the Library". Such a work, in isolation, is not a derivative work of the Library, and therefore falls outside the scope of this License.

However, linking a "work that uses the Library" with the Library creates an executable that is a derivative of the Library (because it contains portions of the Library), rather than a "work that uses the library". The executable is therefore covered by this License. Section 6 states terms for distribution of such executables.

When a "work that uses the Library" uses material from a header file that is part of the Library, the object code for the work may be a derivative work of the Library even though the source code is not. Whether this is true is especially significant if the work can be linked without the Library, or if the work is itself a library. The threshold for this to be true is not precisely defined by law.

If such an object file uses only numerical parameters, data structure layouts and accessors, and small macros and small inline functions (ten lines or less in length), then the use of the object file is unrestricted, regardless of whether it is legally a derivative work. (Executables containing this object code plus portions of the Library will still fall under Section 6.)

Otherwise, if the work is a derivative of the Library, you may distribute the object code for the work under the terms of Section 6. Any executables containing that work also fall under Section 6, whether or not they are linked directly with the Library itself.

6. As an exception to the Sections above, you may also combine or link a "work that uses the Library" with the Library to produce a work containing portions of the Library, and distribute that work under terms of your choice, provided that the terms permit modification of the work for the customer's own use and reverse engineering for debugging such modifications.

You must give prominent notice with each copy of the work that the Library is used in it and that the Library and its use are covered by this License. You must supply a copy of this License. If the work during execution displays copyright notices, you must include the copyright notice for the Library among them, as well as a reference directing the user to the copy of this License. Also, you must do one of these things:

a) Accompany the work with the complete corresponding machine-readable source code for the Library including whatever changes were used in the work (which must be distributed under Sections 1 and 2 above); and, if the work is an executable linked with the Library, with the complete machine-readable "work that uses the Library", as object code and/or source code, so that the user can modify the Library and then relink to produce a modified executable containing the modified Library. (It is understood that the user who changes the contents of definitions files in the Library will not necessarily be able to recompile the application to use the modified definitions.)

b) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (1) uses at run time a copy of the library already present on the user's computer system, rather than copying library functions into the executable, and (2) will operate properly with a modified version of the library, if the user installs one, as long as the modified version is interface-compatible with the version that the work was made with.

c) Accompany the work with a written offer, valid for at least three years, to give the same user the materials specified in Subsection 6a, above, for a charge no more than the cost of performing this distribution.

d) If distribution of the work is made by offering access to copy from a designated place, offer equivalent access to copy the above specified materials from the same place.

e) Verify that the user has already received a copy of these materials or that you have already sent this user a copy.

For an executable, the required form of the "work that uses the Library" must include any data and utility programs needed for reproducing the executable from it. However, as a special exception, the materials to be distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

It may happen that this requirement contradicts the license restrictions of other proprietary libraries that do not normally accompany the operating system. Such a contradiction means you cannot use both them and the Library together in an executable that you distribute.

7. You may place library facilities that are a work based on the Library side-by-side in a single library together with other library facilities not covered by this License, and distribute such a combined library, provided that the separate distribution of the work based on the Library and of the other library facilities is otherwise permitted, and provided that you do these two things:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities. This must be distributed under the terms of the Sections above.

b) Give prominent notice with the combined library of the fact that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

8. You may not copy, modify, sublicense, link with, or distribute the Library except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, link with, or distribute the Library is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

9. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Library or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Library (or any work based on the Library), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Library or works based on it.

10. Each time you redistribute the Library (or any work based on the Library), the recipient automatically receives a license from the original licensor to copy, distribute, link with or modify the Library subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties with this License.

11. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Library at all. For example, if a patent license

would not permit royalty-free redistribution of the Library by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Library.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply, and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

12. If the distribution and/or use of the Library is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Library under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

13. The Free Software Foundation may publish revised and/or new versions of the Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Library does not specify a license version number, you may choose any version ever published by the Free Software Foundation.

14. If you wish to incorporate parts of the Library into other free programs whose distribution conditions are incompatible with these, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

15. BECAUSE THE LIBRARY IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE LIBRARY, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE LIBRARY "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE LIBRARY IS WITH YOU. SHOULD THE LIBRARY PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

16. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE LIBRARY AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE LIBRARY (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE LIBRARY TO OPERATE WITH ANY OTHER SOFTWARE), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### END OF TERMS AND CONDITIONS

##### How to Apply These Terms to Your New Libraries

If you develop a new library, and you want it to be of the greatest possible use to the public, we recommend making it free software that everyone can redistribute and change. You can do so by permitting redistribution under these terms (or, alternatively, under the terms of the ordinary General Public License).

To apply these terms, attach the following notices to the library. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

one line to give the library's name and an idea of what it does.

Copyright (C) year name of author

This library is free software; you can redistribute it and/or  
modify it under the terms of the GNU Lesser General Public  
License as published by the Free Software Foundation; either  
version 2.1 of the License, or (at your option) any later version.

This library is distributed in the hope that it will be useful,  
but WITHOUT ANY WARRANTY; without even the implied warranty of  
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU  
Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public  
License along with this library; if not, write to the Free Software  
Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA  
02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the library, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in  
the library `Frob' (a library for tweaking knobs) written  
by James Random Hacker.

signature of Ty Coon, 1 April 1990

Ty Coon, President of Vice

That's all there is to it!

## 17.7 OCC CAD Kernel

Parts of version 6.3 of OCC are included with Mecway. The source code is available from <http://www.opencascade.org/>. It is used under the following license agreement.

#### Open CASCADE Technology Public License

License version : 6.3 August, 2008

Open CASCADE S.A.S. releases and makes publicly available the source code of the software Open CASCADE Technology to the free software development community under the terms and conditions of this license.

It is not the purpose of this license to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this license has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

Please read this license carefully and completely before downloading this software. By downloading, using, modifying, distributing and sublicensing this software, you indicate your acceptance to be bound by the terms and conditions of this license. If you do not want to accept or cannot accept for any reasons the terms and conditions of this license, please do not download or use in any manner this software.

#### 1. Definitions

Unless there is something in the subject matter or in the context inconsistent therewith, the capitalized terms used in this License shall have the following meaning.

"Applicable Intellectual Property Rights" means (a) with respect to the Initial Developer, any rights under patents or patents applications or other intellectual property rights that are now or hereafter acquired, owned by or assigned to the Initial Developer and that cover subject matter contained in the Original Code, but only to the extent necessary to use, reproduce, modify, distribute or sublicense the Original Code without infringement; and (b) with respect to You or any Contributor, any rights under patents or patents applications or other intellectual property rights that are now or hereafter acquired, owned by or assigned to You or to such Contributor and that cover subject matter contained in Your Modifications or in such Contributor's Modifications, taken alone or in combination with Original Code

"Contributor" means each individual or legal entity that creates or contributes to the creation of any Modification, including the Initial Developer.

"Derivative Program": means a new program combining the Software or portions thereof with other source code not governed by the terms of this License.

"Initial Developer": means Open CASCADE S.A.S., with main offices at Tour Opus 12, 77, Esplanade du Général de Gaulle, 92914, Paris, La Défense Cedex, France.

"Modifications": mean any addition to, deletion from or change to the substance or the structure of the Software. When source code of the Software is released as a series of

files, a Modification is: (a) any addition to, deletion from or change to the contents of a file containing the Software or (b) any new file or other representation of computer program statements that contains any part of the Software. By way of example, Modifications include any debug of, or improvement to, the Original Code or any of its components or portions as well as its next versions or releases thereof.

"Original Code": means (a) the source code of the software Open CASCADE Technology originally made available by the Initial Developer under this License, including the source code of any updates or upgrades of the Original Code and (b) the object code compiled from such source code and originally made available by Initial Developer under this License.

"Software": means the Original Code, the Modifications, the combination of Original Code and any Modifications or any respective portions thereof.

"You" or "Your": means an individual or a legal entity exercising rights under this License

## 2. Acceptance of license

By using, reproducing, modifying, distributing or sublicensing the Software or any portion thereof, You expressly indicate Your acceptance of the terms and conditions of this License and undertake to act in accordance with all the provisions of this License applicable to You.

## 3. Scope and purpose

This License applies to the Software and You may not use, reproduce, modify, distribute, sublicense or circulate the Software, or any portion thereof, except as expressly provided under this License. Any attempt to otherwise use, reproduce, modify, distribute or sublicense the Software is void and will automatically terminate Your rights under this License.

## 4. Contributor license

Subject to the terms and conditions of this License, the Initial Developer and each of the Contributors hereby grant You a world-wide, royalty-free, irrevocable and non-exclusive license under the Applicable Intellectual Property Rights they own or control, to use, reproduce, modify, distribute and sublicense the Software provided that:

You reproduce in all copies of the Software the copyright and other proprietary notices and disclaimers of the Initial Developer as they appear in the Original Code and attached hereto as Schedule "A" and any other notices or disclaimers attached to the Software and keep intact all notices in the Original Code that refer to this License and to the absence of any warranty;

You include a copy of this License with every copy of the Software You distribute;

If you distribute or sublicense the Software (as modified by You or on Your behalf as the case may be), You cause such Software to be licensed as a whole, at no charge, to all third parties, under the terms and conditions of the License, making in particular available to all third parties the source code of the Software;

You document all Your Modifications, indicate the date of each such Modifications, designate the version of the Software You used, prominently include a file carrying such information with respect to the Modifications and duplicate the copyright and other proprietary notices and disclaimers attached hereto as Schedule "B" or any other notices or disclaimers attached to the Software with your Modifications.

For greater certainty, it is expressly understood that You may freely create Derivative Programs (without any obligation to publish such Derivative Program) and distribute same as a single product. In such case, You must ensure that all the requirements of this License are fulfilled for the Software or any portion thereof.

## 5. Your license

You hereby grant all Contributors and anyone who becomes a party under this License a world-wide, non-exclusive, royalty-free and irrevocable license under the Applicable Intellectual Property Rights owned or controlled by You, to use, reproduce, modify, distribute and sublicense all Your Modifications under the terms and conditions of this License.

## 6. Software subject to license

Your Modifications shall be governed by the terms and conditions of this License. You are not authorized to impose any other terms or conditions than those prevailing under this License when You distribute and/or sublicense the Software, save and except as permitted under Section 7 hereof.

## 7. Additional terms

You may choose to offer, on a non-exclusive basis, and to charge a fee for any warranty, support, maintenance, liability obligations or other rights consistent with the scope of this License with respect to the Software (the "Additional Terms") to the recipients of the Software. However, You may do so only on Your own behalf and on Your sole and exclusive responsibility. You must obtain the recipient's agreement that any such Additional Terms are offered by You alone, and You hereby agree to indemnify, defend and hold the Initial Developer and any Contributor harmless for any liability incurred by or claims asserted against the Initial Developer or any Contributors with respect to any such Additional Terms.

## 8. Disclaimer of warranty

The Software is provided under this License on an "as is" basis, without warranty of any kind, including without limitation, warranties that the Software is free of defects,

merchantable, fit for a particular purpose or non-infringing. The entire risk as to the quality and performance of the Software is with You.

## 9. Liability

Under no circumstances shall You, the Initial Developer or any Contributor be liable to any person for any direct or indirect damages of any kind including, without limitation, damages for loss of goodwill, loss of data, work stoppage, computer failure or malfunction or any and all other commercial damages or losses resulting from or relating to this License or indirectly to the use of the Software.

## 10. Trademark

This License does not grant any rights to use the trademarks, trade names and domain names "MATRA", "EADS Matra Datavision", "CAS.CADE", "Open CASCADE", "opencascade.com" and "opencascade.org" or any other trademarks, trade names or domain names used or owned by the Initial Developer.

## 11. Copyright

The Initial Developer retains all rights, title and interest in and to the Original Code. You may not remove the copyright © notice which appears when You download the Software.

## 12. Term

This License is granted to You for a term equal to the remaining period of protection covered by the intellectual property rights applicable to the Original Code.

## 13. Termination

In case of termination, as provided in Section 3 above, You agree to immediately stop any further use, reproduction, modification, distribution and sublicensing of the Software and to destroy all copies of the Software that are in Your possession or control. All sublicenses of the Software which have been properly granted prior to termination shall survive any termination of this License. In addition, Sections 5, 8 to 11, 13.2 and 15.2 of this License, in reason of their nature, shall survive the termination of this License for a period of fifteen (15) years.

## 14. Versions of the license

The Initial Developer may publish new versions of this License from time to time. Once Original Code has been published under a particular version of this License, You may choose to continue to use it under the terms and conditions of that version or use the Original Code under the terms of any subsequent version of this License published by the Initial Developer.

## 15. Miscellaneous

### 15.1 Relationship of the Parties

This License will not be construed as creating an agency, partnership, joint venture or any other form of legal association between You and the Initial Developer, and You will not represent to the contrary, whether expressly, by implication or otherwise.

### 15.2 Independent Development

Nothing in this License will impair the Initial Developer's right to acquire, license, develop, have others develop for it, market or distribute technology or products that perform the same or similar functions as, or otherwise compete with, Modifications, Derivative Programs, technology or products that You may develop, produce, market or distribute.

### 15.3 Severability

If for any reason a court of competent jurisdiction finds any provision of this License, or portion thereof, to be unenforceable, that provision of the License will be enforced to the maximum extent permissible so as to effect the economic benefits and intent of the parties, and the remainder of this License will continue in full force and extent.

## END OF THE TERMS AND CONDITIONS

### OF THIS LICENSE

Open CASCADE S.A.S. is a French société par actions simplifiée having its registered head office at 31, avenue de la Baltique , 91954 Les Ulis Cedex, France and main offices at Tour Opus 12, 77, Esplanade du Général de Gaulle, 92914, Paris, La Défense Cedex, France. Its web site is located at the following address [www.opencascade.com](http://www.opencascade.com)

### Open CASCADE Technology Public License

#### Schedule "A"

The content of this file is subject to the Open CASCADE Technology Public License Version 6.2 (the "License"). You may not use the content of this file except in compliance with the License. Please obtain a copy of the License at <http://www.opencascade.org> and read it completely before using this file.

The Initial Developer of the Original Code is Open CASCADE S.A.S., with main offices at Tour Opus 12, 77, Esplanade du Général de Gaulle, 92914, Paris, La Défense Cedex, France. The Original Code is copyright © Open CASCADE S.A.S., 2001. All rights reserved.

"The Original Code and all software distributed under the License are distributed on an "AS IS" basis, without warranty of any kind, and the Initial Developer hereby disclaims all such warranties, including without limitation, any warranties of merchantability, fitness for a particular purpose or non-infringing.

Please see the License for the specific terms and conditions governing rights and limitations under the License".

#### End of Schedule "A"

### Open CASCADE Technology Public License

#### Schedule "B"

"The content of this file is subject to the Open CASCADE Technology Public License Version 6.2 (the "License"). You may not use the content of this file except in compliance with the License. Please obtain a copy of the License at <http://www.opencascade.org> and read it completely before using this file.

The Initial Developer of the Original Code is Open CASCADE S.A.S., with main offices at Tour Opus 12, 77, Esplanade du Général de Gaulle, 92914, Paris, La Défense Cedex, France. The Original Code is copyright © Open CASCADE S.A.S., 2001. All rights reserved.

Modifications to the Original Code have been made by \_\_\_\_\_.  
Modifications are copyright © [Year to be included]. All rights reserved.

The software Open CASCADE Technology and all software distributed under the License are distributed on an "AS IS" basis, without warranty of any kind, and the Initial Developer hereby disclaims all such warranties, including without limitation, any warranties of merchantability, fitness for a particular purpose or non-infringement. Please see the License for the specific terms and conditions governing rights and limitations under the License"

#### End of Schedule "B"

## 17.8 Turbo Color Map

Apache License

Version 2.0, January 2004

<http://www.apache.org/licenses/>

### TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

#### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");

you may not use this file except in compliance with the License.

You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and

limitations under the License.

## 17.9 Intel MKL

Intel Simplified Software License (Version October 2022)

Use and Redistribution. You may use and redistribute the software, which is provided in binary form only, (the "Software"), without modification, provided the following conditions are met:

\* Redistributions must reproduce the above copyright notice and these terms of use in the Software and in the documentation and/or other materials provided with the distribution.

\* Neither the name of Intel nor the names of its suppliers may be used to endorse or promote products derived from this Software without specific prior written permission.

\* No reverse engineering, decompilation, or disassembly of the Software is permitted, nor any modification or alteration of the Software or its operation at any time, including during execution.

No other licenses. Except as provided in the preceding section, Intel grants no licenses or other rights by implication, estoppel or otherwise to, patent, copyright, trademark, trade name, service mark or other intellectual property licenses or rights of Intel.

Third party software. "Third Party Software" means the files (if any) listed in the "third-party-software.txt" or other similarly-named text file that may be included with the Software. Third Party Software, even if included with the distribution of the Software, may be governed by separate license terms, including without limitation, third party license terms, open source software notices and terms, and/or other Intel software license terms. These separate license terms solely govern Your use of the Third Party Software.

DISCLAIMER. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT ARE DISCLAIMED. THIS SOFTWARE IS NOT INTENDED FOR USE IN SYSTEMS OR APPLICATIONS WHERE FAILURE OF THE SOFTWARE MAY CAUSE PERSONAL INJURY OR DEATH AND YOU AGREE THAT YOU ARE FULLY RESPONSIBLE FOR ANY CLAIMS, COSTS, DAMAGES, EXPENSES, AND ATTORNEYS' FEES ARISING OUT OF ANY SUCH USE, EVEN IF ANY CLAIM ALLEGES THAT INTEL WAS NEGLIGENT REGARDING THE DESIGN OR MANUFACTURE OF THE SOFTWARE.

LIMITATION OF LIABILITY. IN NO EVENT WILL INTEL BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

No support. Intel may make changes to the Software, at any time without notice, and is not obligated to support, update or provide training for the Software.

Termination. Your right to use the Software is terminated in the event of your breach of this license.

Feedback. Should you provide Intel with comments, modifications, corrections, enhancements or other input ("Feedback") related to the Software, Intel will be free to use, disclose, reproduce, license or otherwise distribute or exploit the Feedback in its sole discretion without any obligations or restrictions of any kind, including without limitation, intellectual property rights or licensing obligations.

Compliance with laws. You agree to comply with all relevant laws and regulations governing your use, transfer, import or export (or prohibition thereof) of the Software.

Governing law. All disputes will be governed by the laws of the United States of America and the State of Delaware without reference to conflict of law principles and subject to the exclusive jurisdiction of the state or federal courts sitting in the State of Delaware, and each party agrees that it submits to the personal jurisdiction and venue of those courts and waives any objections. THE UNITED NATIONS CONVENTION ON CONTRACTS FOR THE INTERNATIONAL SALE OF GOODS (1980) IS SPECIFICALLY EXCLUDED AND WILL NOT APPLY TO THE SOFTWARE.