Overview of Finite Element Analysis

1. Introduction

This document is intended to give a general description of Finite Element Analysis (FEA) and how it works in practice. It will be based around the ***Calculix*** FEA program, but the description will be relevant for most FEA programs. We will also provide a simple example of an aluminium channel, analysing this using FEA then comparing the theoretical results with actual test data.

2. General Concepts

Analysing a complex machine component for strength can be a very difficult job. Theoretical analysis only really works well for very simple shapes (rectangular or cylindrical blocks, for example). FEA functions by taking a complex shape and dividing it into a very large number of very simple shapes (elements). Each of these shapes can then be analysed using basic stress analysis techniques. If stresses from each element can then be compared with adjacent elements a result for the complete, complex shape can be predicted.

Although the stress calculations for each element are relatively simple, the overall problem is complex due to the very large number of elements involved, and the need to compare the stresses in all the elements with each other. This is where we benefit from computers. They have the ability to perform this comparison efficiently and rapidly. FEA is very much a technique which only exists due to the power and availability of the electronic computer.

In most FEA applications there will be 3 stages in any analysis. These will be Pre-Processing, Processing and Post-Processing.

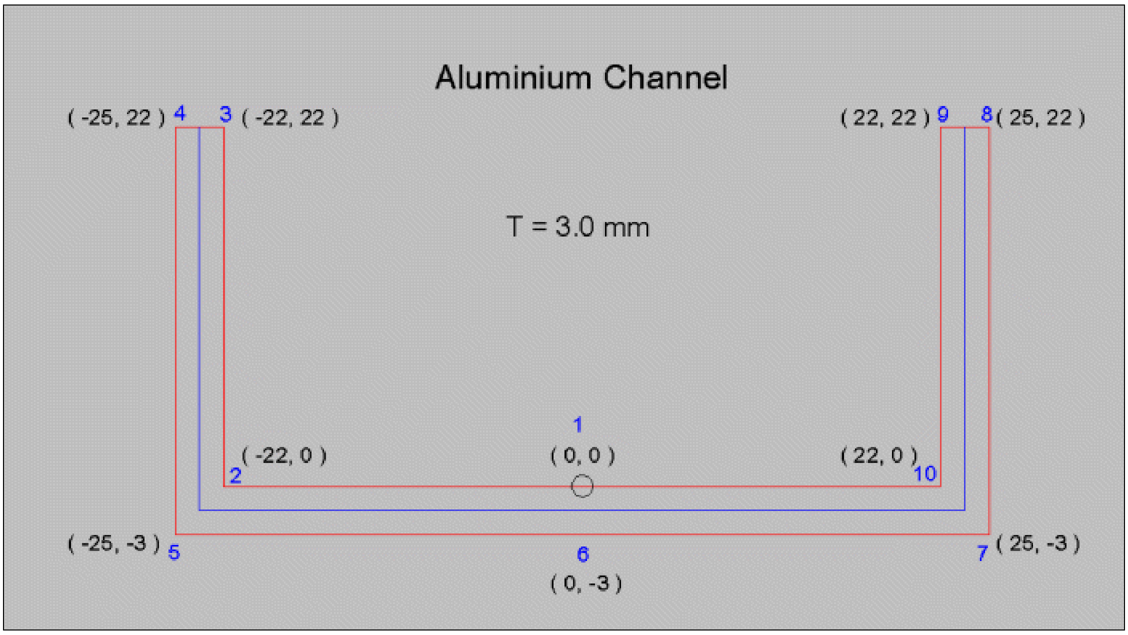
The Pre-Processing phase will take a particular problem and set it up for FEA processing. This will involve defining the object to be analysed as a fine volumetric mesh and applying loads and constraints to represent the conditons acting on the real component. This information will then need to be stored in a format acceptable for the FEA processor (also sometimes called the Solver). Choosing the fineness of the mesh is important as too coarse a mesh will produce inaccurate results and too fine a mesh will take a very long time to process. Loads are the forces acting on an object. Constraints are conditions preventing an object from moving. You need both to be able to determine stresses. A force without any constraints will mean the object being analysed is free to move when the force is applied.

The Processing phase involves running the FEA processor using the input data files created in the Pre-Processing phase. This can take time and the processor will typically just print out occasional lines of information as the processing is carried out. Processing can take minutes or hours depending on the complexity of the object being analysed. During this phase information will be written to output data files for later analysis.

The Post-Processing phase involves taking the data files produced by the FEA processor and viewing them. The FEA data produced is in the form of numbers and it can be very difficult to understand what is happening. The Post-Processor takes these numbers and presents them in various coloured graphical formats to make interpretation very simple.

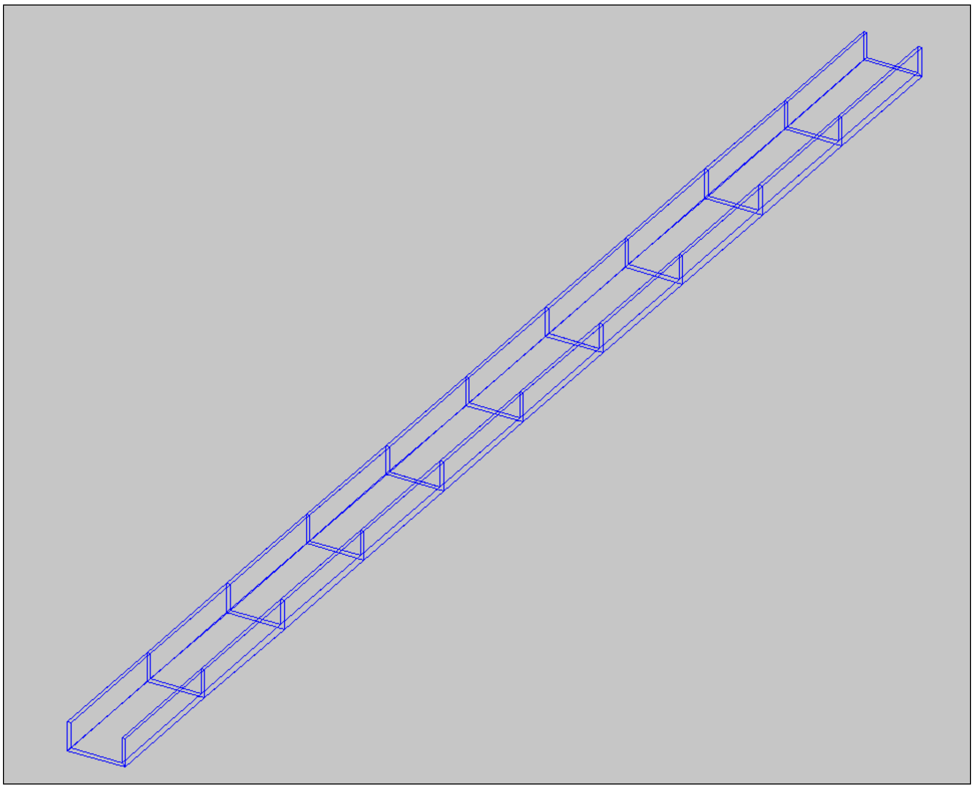
3. Geometry

Having very briefly described the general FEA process we can now look at a simple example to show the steps involved in a typical analysis. Our example will be an aluminium channel. One end will be firmly fixed and a single point load will be applied to the other end.



The above image shows the cross-section shape and dimensions for the aluminium channel. The section is 50 mm wide and 25 mm deep, with a wall thickness of 3.0 mm. We are assuming a length of 1196 mm. This corresponds to a physical length of channel we have available for live testing.

Preparing the data involves 3 steps. We first need to built a 3D model of the channel and convert it to a volumetric mesh. We then need to determine the material properties for the aluminium being used and finally we need to prepare a set of commands to instruct the FEA program what to do.



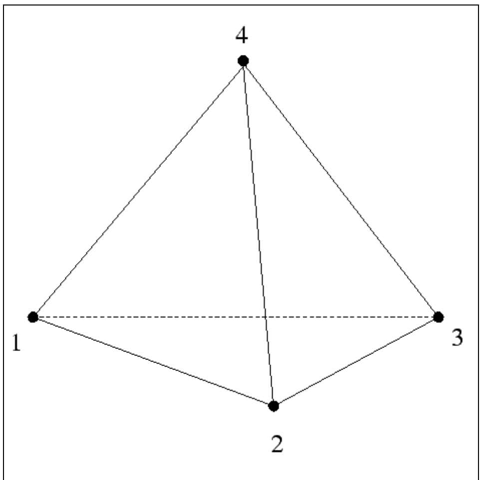
The preceeding image shows the 3D model we started with. This was constructed in ***AutoCAD***. Note that we need to consider the units we are using. The channel cross-section was defined in millimetre units but we need to build the model in metre units so it will be consistent with the defined loads and material properties when we process it.

As a CAD model produced in ***AutoCAD*** this 3D model is defined as a surface and not a volume. If we analysed this now the geometry would be seen as a hollow shell and not a solid block of material. Our first requirement is to produce a volumetric grid. In expensive, high-end FEA programs the Pre-Processing operations are typically provided to cover this requirement. In our case we can find free meshing programs to perform the conversion, but need to take a few steps to achieve the result we are after.

After considerable testing our most convenient meshing procedure (based on the CAD programs available to us) has been found to be to first save the ***AutoCAD*** 3D surface mesh as a DWG or DXF file. This is then loaded into the ***Rhino*** CAD program and saved as an ASCII STL (stereolithography) file. Although later versions of ***AutoCAD*** can also save an STL file they are more limited in what can be produced. This STL file is then converted to a volumetric mesh using a program called ***TetGen.*** The result is a volumetric mesh saved in the required ***Calculix*** format.

The ***TetGen*** program is a program written to produce volumetric meshes. In order to produce the ***Calculix*** formatted mesh we had to modify the ***TetGen*** program for our specific needs. The standard ***TetGen*** program did not perform the specific conversion we required. ***TetGen*** is able to directly convert a mesh, or automatically produce a refined version of the mesh. For this example we produced both. Mesh complexity is usually defined according to the number of nodes and elements making up the mesh. The direct conversion of our CAD data was saved in a file named ***Channel1*** (172 nodes and 376 elements) while the refined mesh was saved in a file named ***Channel2*** (13864 nodes and 42597 elements).

For our volumetric meshes a node is a 3D point in space, and is a point defining one corner of a mesh element. An element here is a tetrahedral prism. The definition of this is shown in the following diagram. The prism is defined by 4 corner points (nodes) and our volumetric mesh is made up of a large number of these elements.



4 node tet element (type 3)

It has been mentioned already that the mesh size chosen is an important consideration. Too simple a mesh will give inaccurate results and too complex a mesh will take a long time to process. The following analysis will show our ***Channel1*** mesh is too coarse and gives poor results while our ***Channel2*** mesh is much better.

Ideally, analysis can be handled by having both a relatively simple mesh and a very fine mesh. For initial testing and setting up of the various parameters the simple mesh can be used. This will process rapidly. When everything is ready the fine mesh can then be used to produce a superior result, though at the cost of much longer computer processing time.

Fortunately the ***TetGen*** program maintains consistent node numbering between the two mesh quality levels it produces, so it is possible to interchange the coarse and fine mesh files without needing to change any other part of the processing sequence.

Having produced the volumetric meshes we now need to define the material properties. The aluminium being used is 6061-T6 grade. Available design data specifies (from two different sources) the following information.

Ultimate Tensile Strength	430 MPa	310 MPa
Tensile Yield Strength	320 Mpa	276 MPa
Modulus of Elasticity	72 GPa	68.9 GPa
Density	2850 kg per cubic metre	
Poission's Ratio	0.34	0.33

Finally we need to define the command sequence for the FEA processor. The following is a sequence of commands to run the FEA process.

```
*HEADING
** Test of TetGen generated 3D geometry
**
*INCLUDE,INPUT=C:\Calculix\Channel1.Ccx
**
*BOUNDARY
4,1
4,2
4,3
71,1
71,2
71,3
147,1
147,2
147,3
158,1
158,2
158,3
**
** Set material for all elements
*MATERIAL,NAME=Aluminium
*ELASTIC
70000.0,0.34
**
*SOLID SECTION,ELSET=Eall,MATERIAL=Aluminium
**
*STEP
*STATIC,SOLVER=SPOOLES
**
*CLOAD
171,3,1.0
**
*NODE PRINT,NSET=Nall
U
**
*EL PRINT,ELSET=Eall
S,E
**
*NODE FILE
U
**
*EL FILE
S,E
**
*END STEP
```

The key items to note are the input of the file ***Channel1.Ccx***. This is the volumetric mesh geometry. By keeping it as a separate file it becomes easy to switch between the fine and coarse mesh files. We also define some boundary conditions, a load and the required material properties.

We define 4 boundary points which are fixed in all directions (X, Y and Z). This firmly attaches the channel in space at one end. We define a single vertical load at the other end of the channel. We define the Modulus of Elasticity for aluminium and its Poission Ratio, to specify the material properties which will be used in the analysis.

Several comment lines are also present in the control sequence to help clarify what is being done. The final commands specify the type of analysis and what data should be recorded to disk files during processing, for later viewing.

At this stage we can run the FEA Processor to analyse our problem.

4. Modelling

It may be helpful to record some general information about how our 3D geometry is processed to make the volumetric meshes needed by ***Calculix***. Producing a suitable mesh can be a major issue.

From our experience to date, a 3D model is normally built in ***AutoCAD*** then imported to ***Rhino*** in the form of a DWG file. This is then exported from ***Rhino*** as an ASCII STL file. It is also possible to save an STL file in a binary format but the later conversion programs cannot handle the binary format. The 3D model should be “closed” (watertight) as it represents a closed volume. ***Rhino*** can save open geometry but offers an output option to only save closed geometry. This option should be used to help ensure good data is produced.

Experience has shown, for unknown reasons, that 3D models created in ***Rhino*** convert poorly. It has also been found that loading an STL file into ***Rhino***, then saving it back out to STL can produce unsuitable files.

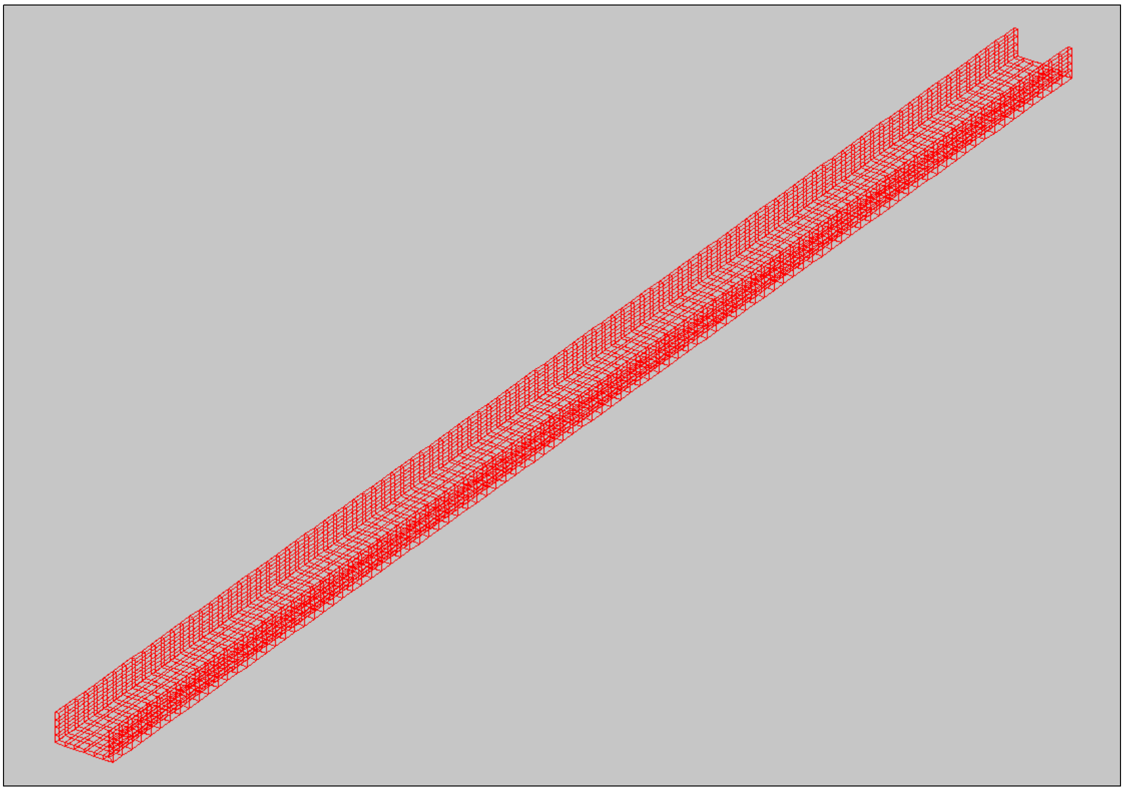
Models exported in STL format from ***AutoCAD*** have positioning limitations and are saved in binary format. The model must exist in the positive octant of the 3D drawing space. This can be limiting as it forces a model location which may not suit the requirements of the FEA analysis. For example, with the aluminium Channel we have specified the origin point (0,0,0) at the centre of one end of the model. This makes the specification of loads and boundary conditions a little easier and more logical.

The orientation of the facets defining a 3D surface must be done consistently. They must all have consistent definition of the points defining each face (either clockwise or anti-clockwise orientation). If a more detailed mesh is required then one approach to maintain consistency is to build a simple model then use a mesh manipulation program to increase the mesh density. One such program is called ***MeshLab***. This has the ability to sub-divide meshes, so each single facet can be subdivided into 4 facets. Repeated application can increase the number of faces making up a surface without needing to manually create them. If using ***MeshLab***, do NOT fix the vertices when asked as this will cause problems with the STL file later.

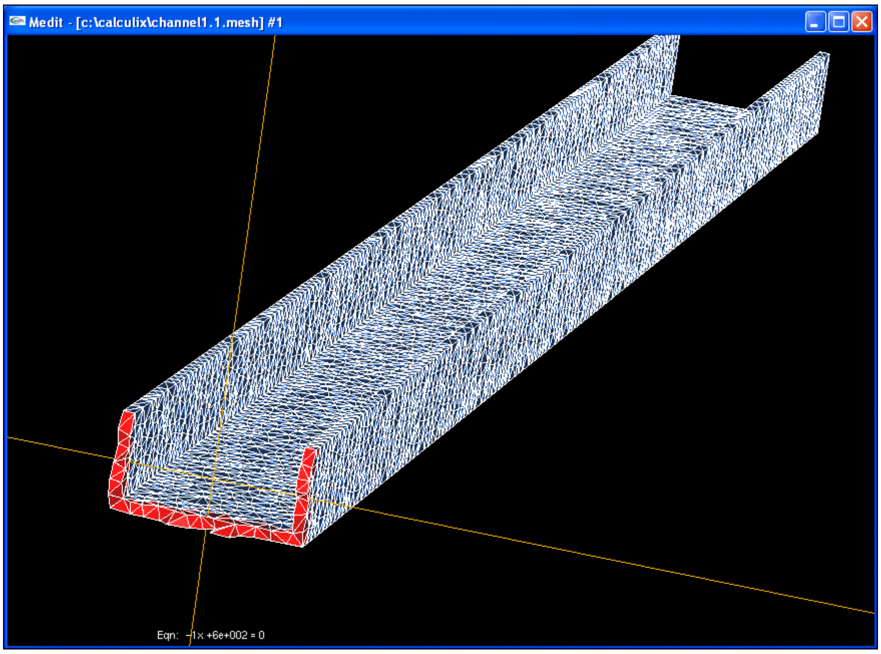
In a specific situation described later, the simplest model of the Channel was saved as an ASCII STL file. It was then loaded into ***MeshLab***. The mesh was filtered using the “midpoint sub-division” option. The mesh was saved as a DXF file to disk. This DXF file was loaded into ***Rhino***, then saved as a new ASCII STL file. If we saved the STL file directly from ***MeshLab*** it was saved in binary format.

If we had used ***Rhino*** to directly convert the binary STL file to ASCII format (as required by the ***TetGen*** program) the resulting file was rejected by ***TetGen***. The DXF conversion procedure produced files acceptable to ***TetGen***. As some of this information indicates, successful conversion of CAD data to a suitable FEA format can be a bit tricky.

Once we have produced a suitable STL file we then always convert it to the required FEA format using ***TetGen***.



3D CAD Geometry



Volumetric Mesh (tetrahedral)

5. Analysis

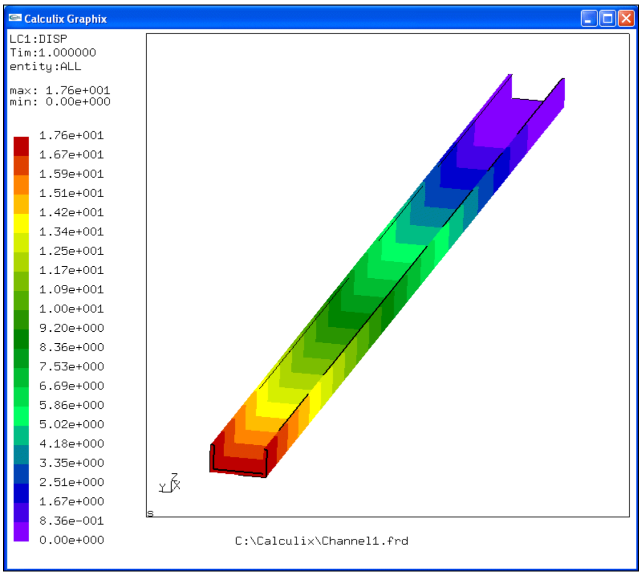
As previously mentioned, the FEA Processor will take the geometry, loads and material properties and process these. The immediate output will be some brief text displayed on the computer screen during processing and the creation of one or more quite large data files on disk. These files are used by the Post-Processor program for viewing and detailed analysis.

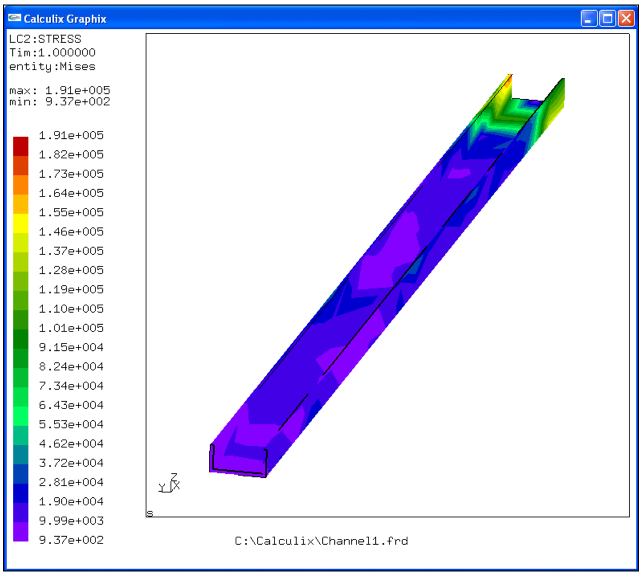
The screen output from the FEA Processor run with our channel geometry appears as follows.

CCX C:\Calculix\Channel1

```
************************************************************
CalculiX version 1.7, Copyright(C) 1998-2007 Guido Dhondt
CalculiX comes with ABSOLUTELY NO WARRANTY. This is free
software, and you are welcome to redistribute it under
certain conditions, see gpl.htm
************************************************************

You are using an executable made on  Aug 9 13:17:29 PST 2007

  The numbers below are estimated upper bounds

  number of:
   nodes:  172
   elements:  376
   one-dimensional elements:  0
   two-dimensional elements:  0
   integration points per element:  1

   distributed facial loads:  0
   distributed volumetric loads:  0
   concentrated loads:  1
   single point constraints:  12
   multiple point constraints:  1
   terms in all multiple point constraints:  1
   tie constraints:  0
   dependent nodes tied by cyclic constraints:  0

   sets:  2
   terms in all sets:  924

   materials:  1
   constants per material and temperature:  2
   temperature points per material:  1
   plastic data points per material:  0

   orientations:  0
   amplitudes:  2
   data points in all amplitudes:  2
   print requests:  3
   transformations:  0
   property cards:  0

 STEP  1
 Static analysis was selected

 Decascading the MPC's


 Renumbering the nodes to decrease the profile:
 old profile =  0*2147483647+ 11799
 new profile =  0*2147483647+ 1477

 Determining the structure of the matrix:
 number of equations
 504
 number of nonzero matrix elements
 7218

 Factoring the system of equations using spooles

 Job finished
```

For the Calculix FEA program there are two physical programs involved. The CCX program is the FEA Processor while they also provide a CGX program which performs both Pre and Post-Processing functions. Although we've done our pre-processing using CAD software we can use the CGX program to graphically display the results of this analysis. The following two images show the deflections and stresses in the channel under the applied load.

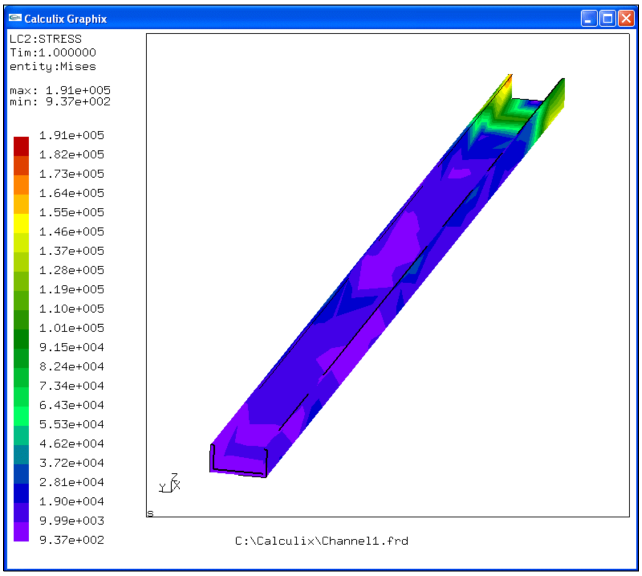


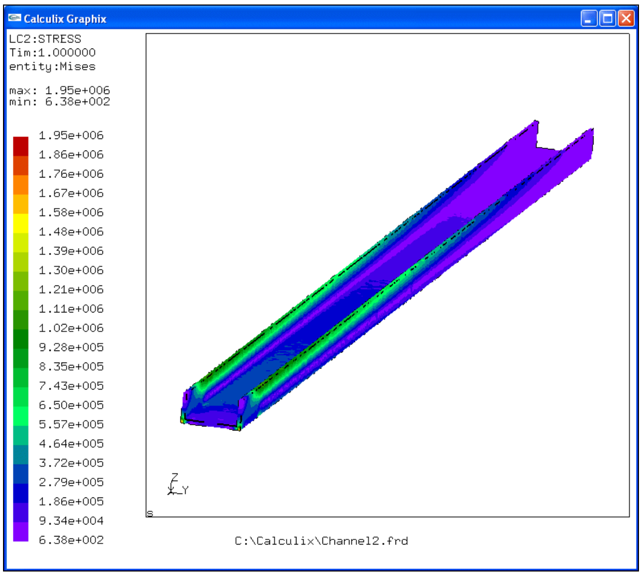


The easiest way to show the results of the analysis is by colour, where various shades of colour show different stresses or deflections. In the left image red shows the greatest deflection while in the right image red shows the highest stress. The vertical scales give a reference of what level of stresse/ deflection corresponds to which colours.

6. Mesh Quality

When we processed both the coarse mesh and the fine mesh we obtained quite different results. A colour display of the stresses for each of these mesh files is shown below. Note that the left image is the result from the coarse mesh and the right image is from the fine mesh. Also note the fine mesh result is shown looking from the opposite direction to the coarse mesh image.



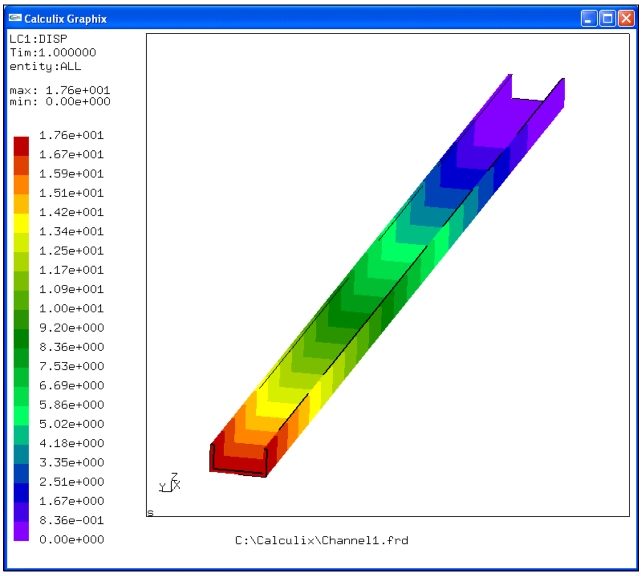


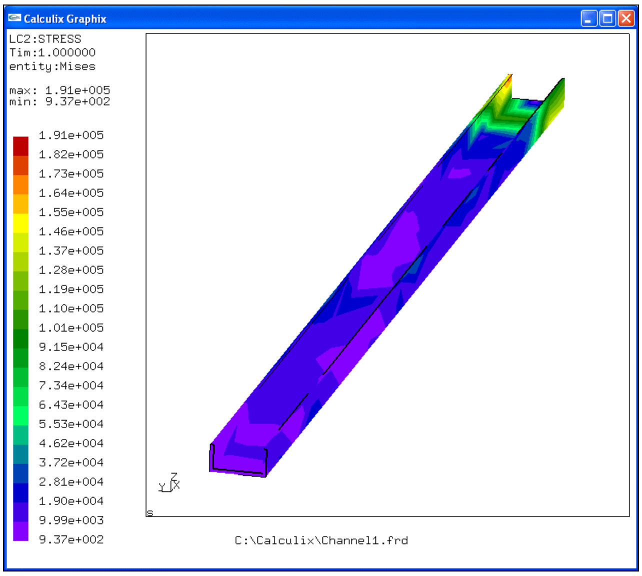
For the fine mesh we see higher stress levels along the tops of each vertical flange of the channel. For the coarse mesh we find the whole region at the end of the channel is under stress. This demonstrates that the quality of the mesh used is very important. Poor choice will give poor results.

To demonstrate this more clearly we produced two more mesh files mid-way between our coarse (Channel1) mesh and our fine (Channel2) mesh. We took the original Channel1 geometry and sub-divided the mesh using *MeshLab*, to create a file we named Channel3. We then sub-divided this again to produce a file named Channel4. These new files were then also run through the FEA Processor.

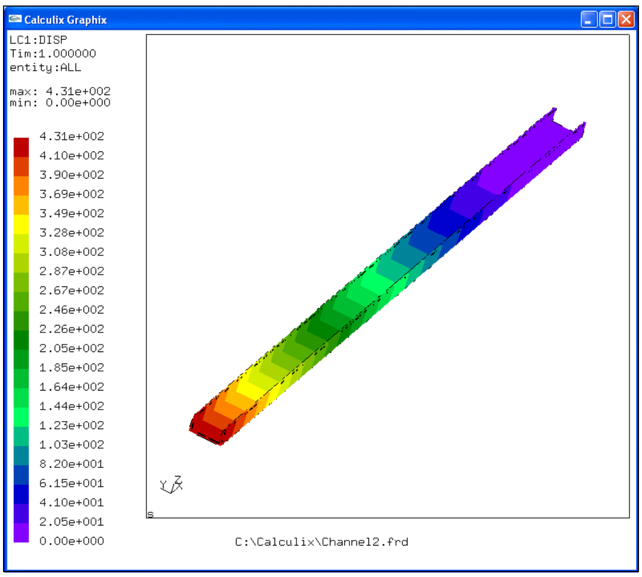
The comparative results from all four runs are shown below. For each run we've displayed the stress and displacement colour coded images from the CGX Post-Processor and also shown the number of nodes and mesh elements comprising each mesh.

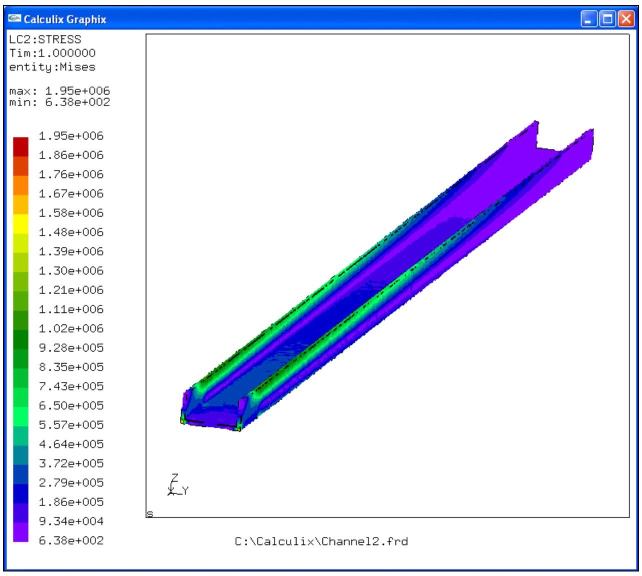
Channel1 172 nodes 376 elements



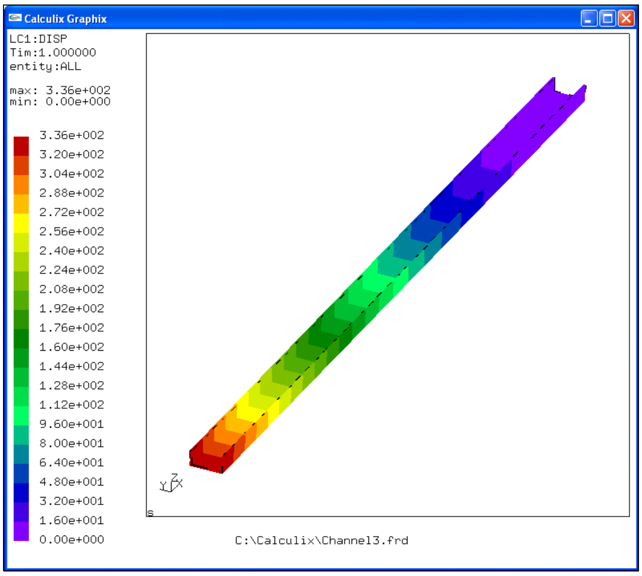


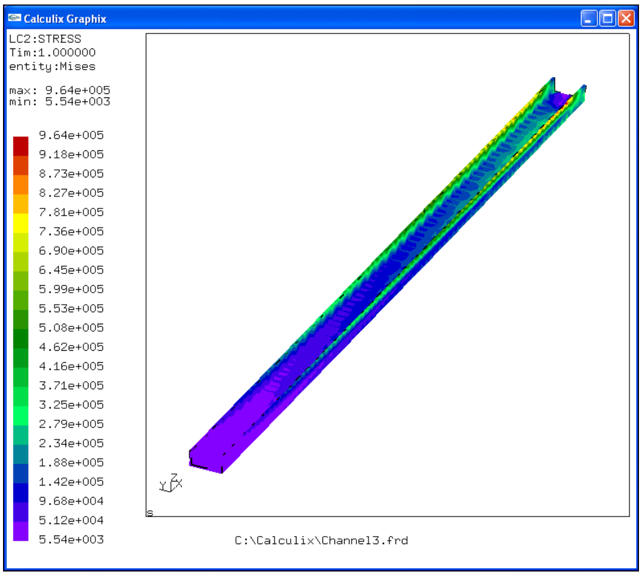
Channel2 13864 nodes 42597 elements



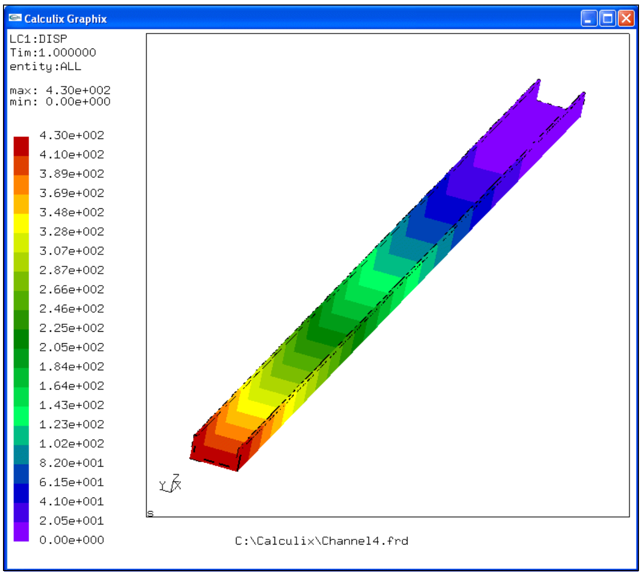


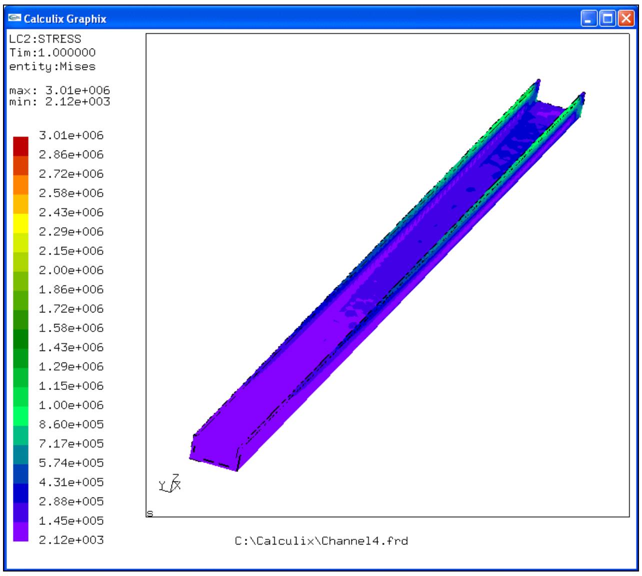
Channel3 2022 nodes 5683 elements





Channel4 6565 nodes 18780 elements





For the stress levels there is reasonable agreement for the three finer meshes. This suggests that our coarse mesh was too coarse and as a result would give invalid results. For any analysis there is a minimum mesh quality level below which the results will be poor. Provided the mesh is above this threshold the results will be acceptable and they should improve slightly as the mesh fineness is increased.

In the situation here we would probably suggest our coarse mesh (for initial model setting up) should have at least 5000 mesh elements and the fine mesh (for the final analysis) could have 50,000 or more.

7. Validation

Having set up our theoretical FEA analysis we now need to do a “Real World” check by applying a load to the actual aluminium beam. It’s always important to validate the computer results with actual tests to maintain the integrity of the analysis.

We’ve also, until now, not worried too much about the specific units being used in the analysis, but just concentrated on getting everything working. This now needs to be addressed. With *Calculix* you can use nearly any units you wish, as long as they form a consistent set. For our application we’ve decided to use metres as the dimensional units and Newtons as the force units.

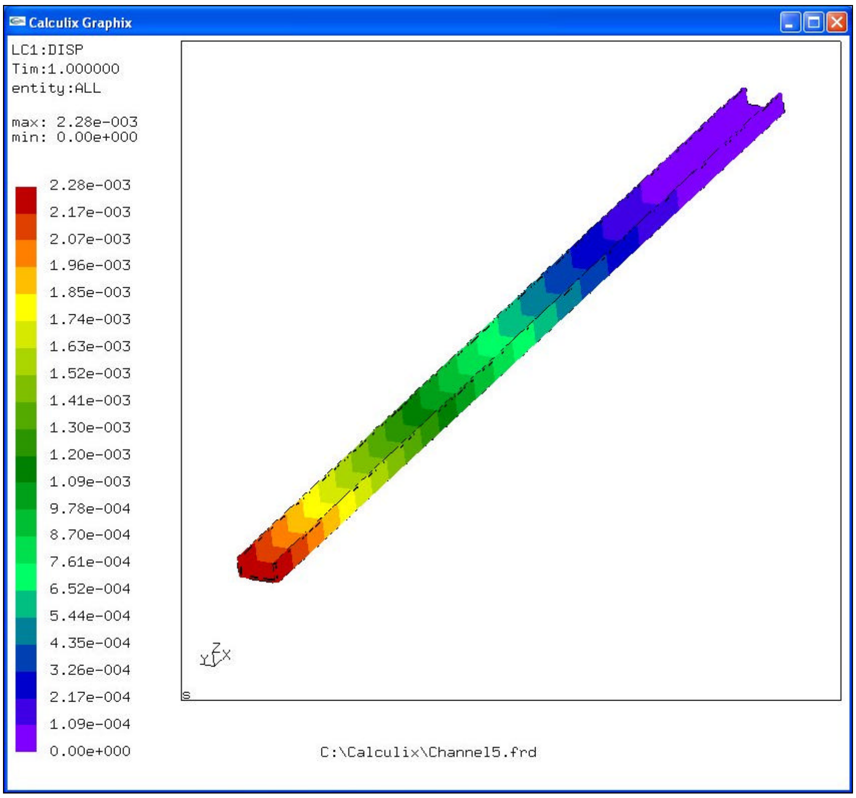
The 3D geometry has already been constructed in metre units. We plan to apply an approximately 0.5 kilogram load to the end of the beam. Our selected aluminium material has a Young’s Modulus of 70,000 MPa, where one Pascal (Pa) is one Newton per square metre.

We have performed a live test by applying a load to the end of the channel while the other end was firmly anchored to a fixed location. The measured deflection at the end of the beam was 6.35 mm under a measured weight of 540 grams. Some items of interest were noted during this test. There was slight distortion visible at the mounted end of the channel, of perhaps 0.35 mm. This was due to a slight deflection of the surface the channel was fixed to. It was also noted there was about a 2.5 mm deflection of the channel under its own weight.

The concentrated load applied to the channel was 0.54 kg x 9.81 = 5.3 Newtons.

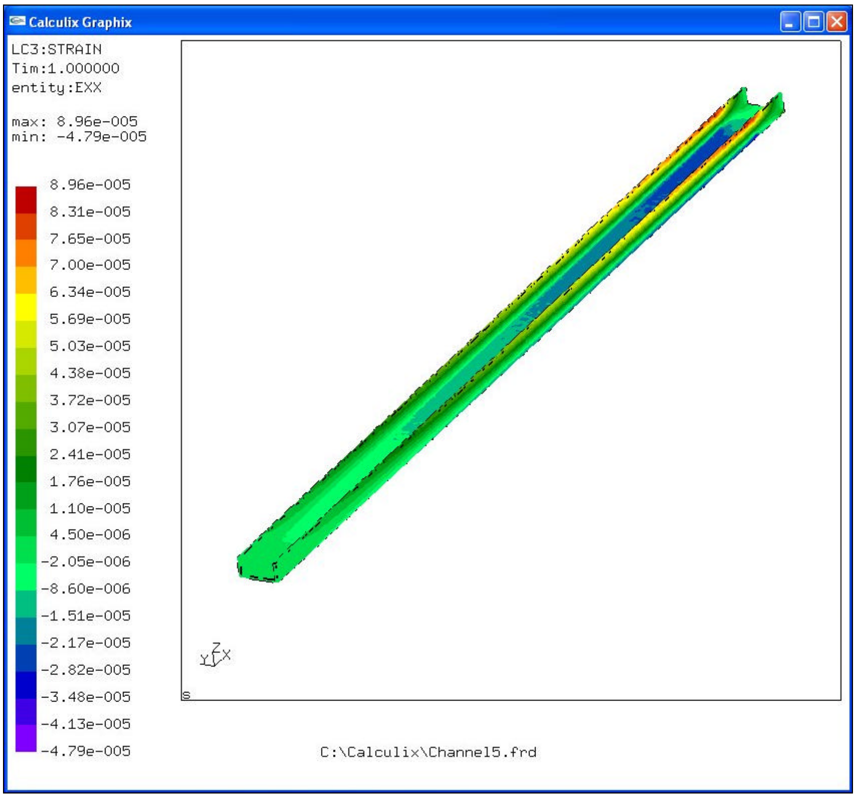
We can now enter these values in our FEA data file to calculate a theoretical value. We will use our most refined mesh geometry (Channel2) but produce a new set of data files, named as Channel5. In order to manage the size of the numbers being used we will define loads in units of kiloPascals (kPa) rather than just Pascals. So our applied load will be 0.0053 kPa and the material Young’s Modulus will now be 70,000,000 kPa. These values will then give us consistent units.

The data was run and the following image shows the calculated displacements along the channel.

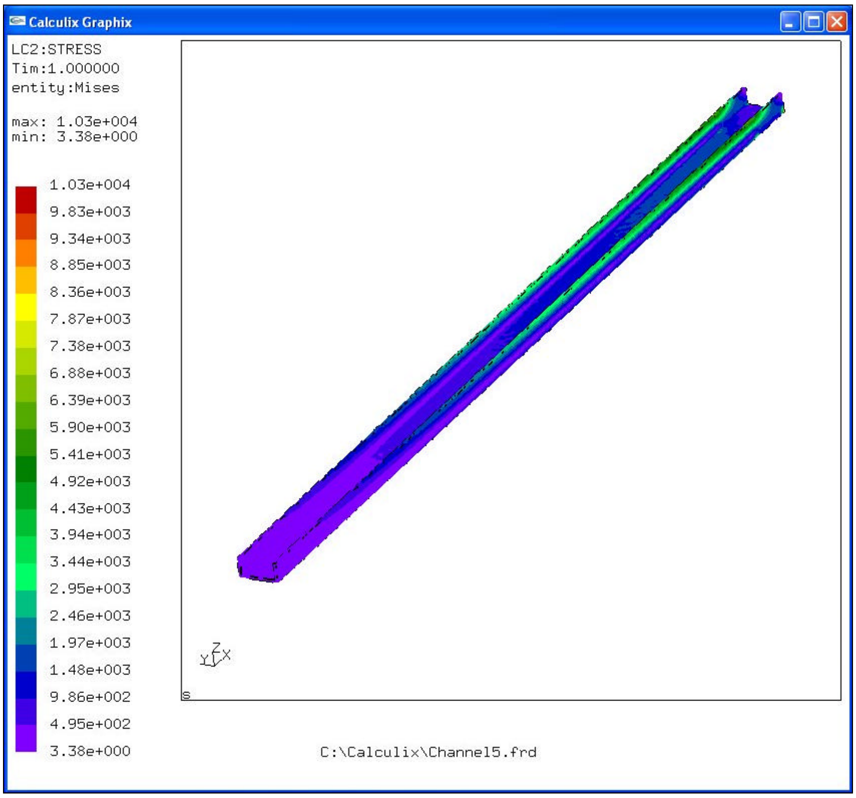


This shows a calculated maximum deflection of 0.00228. where the units are metres, or 2.28 mm at the end of the channel (where the load was applied).

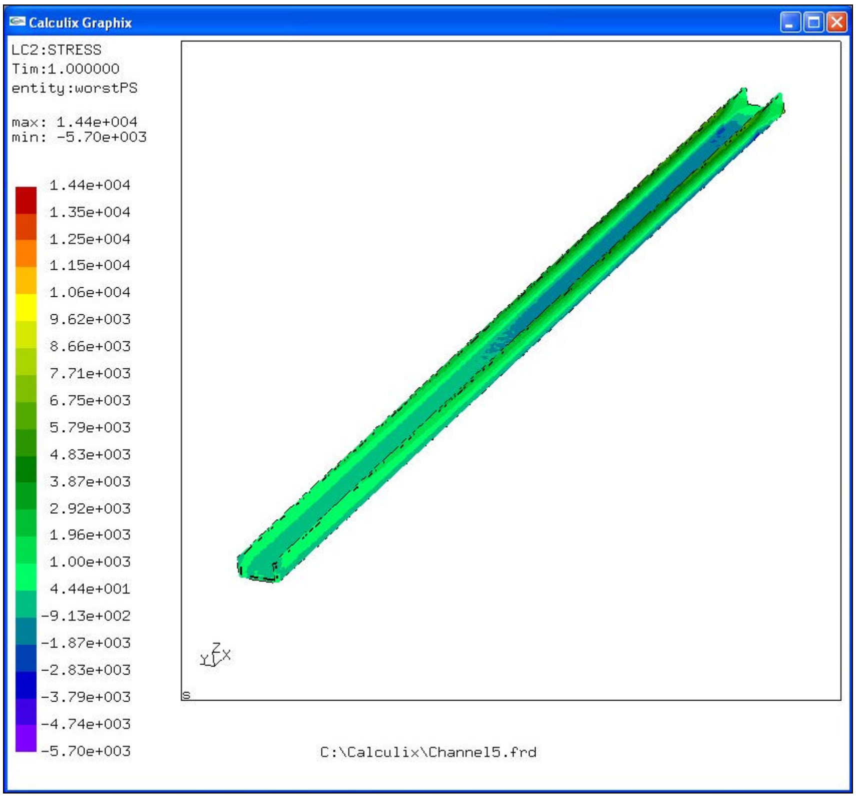
The following image shows the strain in the channel.



The following image shows the Von Mises stress in the channel.



The following image shows the Worst PS stress in the channel.



As a separate validation exercise we also ran a simple beam deflection calculation, for a point load applied to a simple cantilever beam. There is a standard equation which gives the maximum deflection for such a beam.

$$Ymax = (W \times L^3) / (3 \times E \times I)$$

where

W	=	Applied Load	5.3	Newtons
L	=	Beam Length	1196	mm
E	=	Young's Modulus	70,000	Newtons per square millimetre
I	=	Moment of Inertia	21,746	mm^4

This gives a calculated maximum deflection at the end of the beam of 1.99 mm.

The simple beam theory calculated deflection of 1.99 mm compares very well with the FEA calculated deflection of 2.28 mm. The measured deflection of about 6 mm differs substantially from the other values but a number of aspects of the testing of this beam could easily account for the variation. It was apparent the beam was not rigidly fixed and the deflection of the mount could easily have been much greater than estimated. The specific way in which the beam was mounted would also have some effect on the deflection.

The FEA analysis gives a good insight into the behaviour of the channel under a point load. It also provides good information about where the highest stresses are located in the channel. The live test confirms the FEA results are of the correct magnitude and so provides some confidence in the FEA analysis.

Appendix A: Sample Screen Output for Channel 5 FEA Run

This shows the screen output for the final FEA run using the most detailed 3D mesh geometry.

CCX C:\Calculix\Channel5.Inp

```
************************************************************
CalculiX version 1.7, Copyright(C) 1998-2007 Guido Dhondt
CalculiX comes with ABSOLUTELY NO WARRANTY. This is free
software, and you are welcome to redistribute it under
certain conditions, see gpl.htm
************************************************************

You are using an executable made on  Aug 9 13:17:29 PST 2007

  The numbers below are estimated upper bounds

  number of:

   nodes:  13864
   elements:  42597
   one-dimensional elements:  0
   two-dimensional elements:  0
   integration points per element:  1

   distributed facial loads:  0
   distributed volumetric loads:  0
   concentrated loads:  1
   single point constraints:  12
   multiple point constraints:  1
   terms in all multiple point constraints:  1
   tie constraints:  0
   dependent nodes tied by cyclic constraints:  0

   sets:  2
   terms in all sets:  99058

   materials:  1
   constants per material and temperature:  2
   temperature points per material:  1
   plastic data points per material:  0

   orientations:  0
   amplitudes:  2
   data points in all amplitudes:  2
   print requests:  3
   transformations:  0
   property cards:  0

 STEP  1

 Static analysis was selected

 Decascading the MPC's

 Renumbering the nodes to decrease the profile:
 old profile =  0*2147483647+ 77897113
 new profile =  0*2147483647+ 676698

 Determining the structure of the matrix:
 number of equations
 41580
 number of nonzero matrix elements
 715860

 Factoring the system of equations using spooles

 Job finished
```

Appendix B: Sample Screen Output for a TETGEN Run

This shows the screen output for the conversion of a 3D STL geometry file to the volumetric 3D mesh geometry required for *Calculix*.

C:\Calculix\tetgen -xq channel1.stl

```
Opening Channel1.stl.
Constructing Delaunay tetrahedralization.
Delaunay seconds:  0.24
Creating surface mesh.
Jettisoning redundants points.
Perturbing vertices.
Delaunizing segments.
Constraining facets.
Segment and facet seconds:  0.25
Removing unwanted tetrahedra.
Hole seconds:  0.01
Repairing mesh.
Repair seconds:  0

Writing Channel1.1.node.
Writing Channel1.1.ele.
Writing Channel1.1.face.
Writing Channel1.1.smesh.
Writing Channel1.1.mesh.
Writing Channel1.1.ccx.
Writing Channel1.1.ccx.

Output seconds:  0.1
Total running seconds:  0.6

Statistics:

 Input points: 516
  Input facets: 172
  Input segments: 258
  Input holes: 0
  Input regions: 0

  Mesh points: 170
  Mesh tetrahedra: 372
  Mesh triangles: 912
  Mesh subfaces: 336
  Mesh subsegments: 178
```

Appendix C: References

The following references relate to the various software components mentioned in this document.

Calculix

http://www.calculix.de/	main site and Linux implementation
http://bConverged.com/	Windows implementation

The solver, CCX, was written and is maintained by Guido Dhondt. The pre and post processor, CGX, was written and is maintained by Klaus Wittig. The source code, Linux builds, documentation, tests and other resources for CalculiX are available for download at http://www.calculix.de/

Convergent Mechanical Solutions has bundled this build for CCX and CGX as well as the documentation, examples and test suite. There have been some changes from the original source, and those modified source files are included. Some scripts were also added. These are available at http://bConverged.com/

TetGen

http://tetgen.berlios.de/index.html	main Tetgen web site
http://www.car-stuff.info/tetgen	Tetgen version modified for Calculix use

Tetgen is a tetrahedral meshing program.

Rhino

www.rhino3d.com

Rhino is a NURBS 3D Modeller for Windows.

MeshLab

http://meshlab.sourceforge.net

MeshLab is an extensible mesh processing system intended for the cleaning, filtering and rendering of unstructured triangular meshes.

AutoCAD

www.autodesk.com

AutoCad is a widely used 2D/3D CAD program.

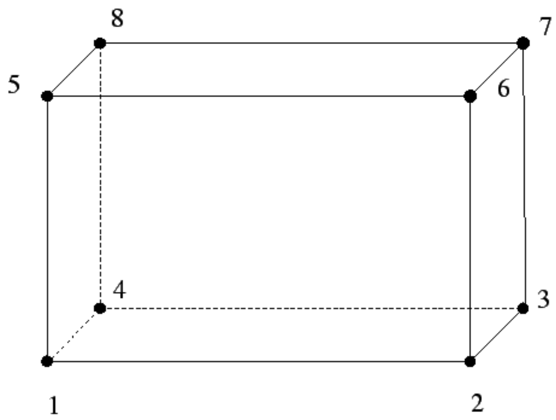
This document was produced by

John McIver

Temporal Images
www.temporal.com.au

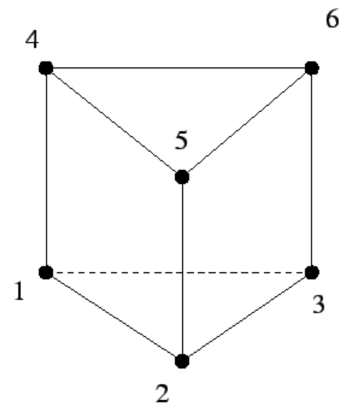
30th April 2008

Appendix D: Standard 2D and 3D Geometry Elements Available in Calculix



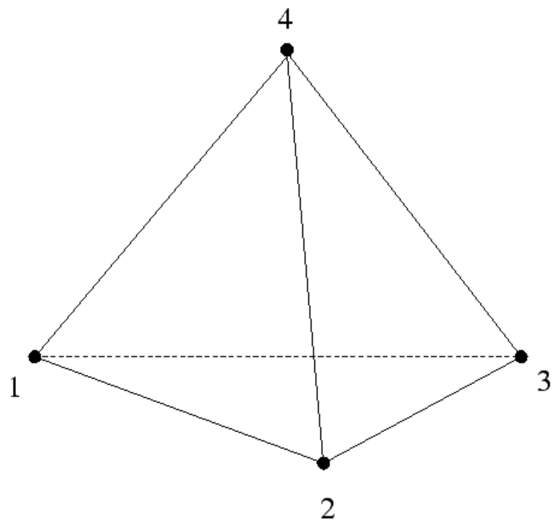
8 node brick element (he8, type 1)

Type 1



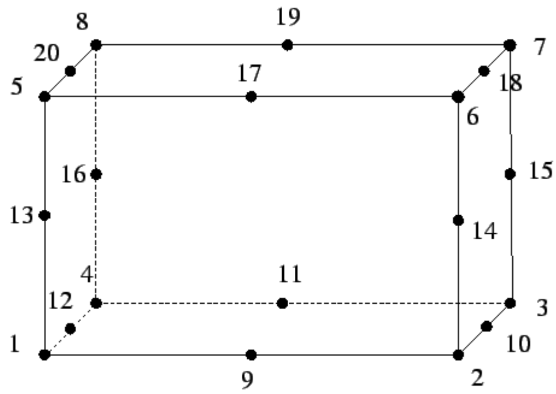
6 node penta element (pe6, type 2)

Type 2



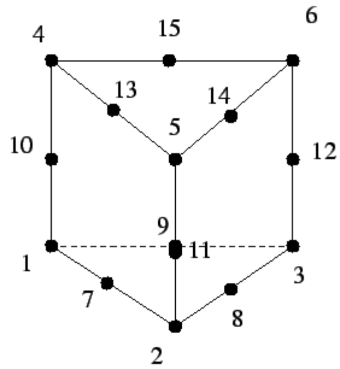
4 node tet element (type 3)

Type 3



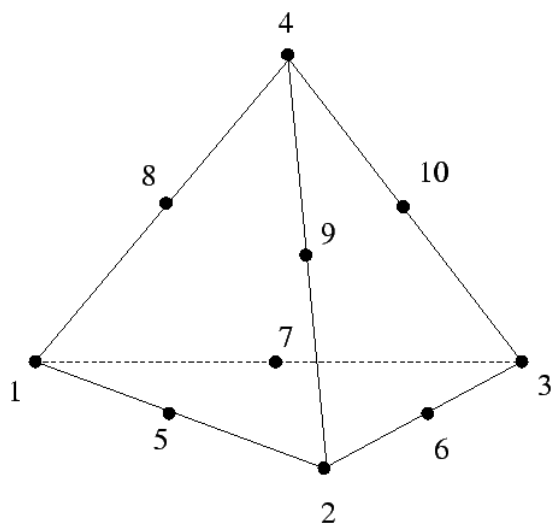
20 node brick element (he20, type 4)

Type 4



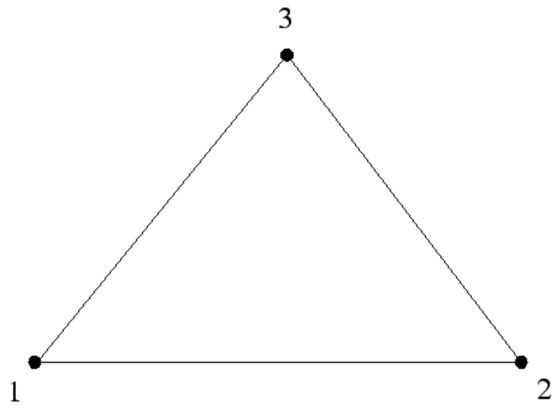
15 node penta element (pe15, type 5)

Type 5



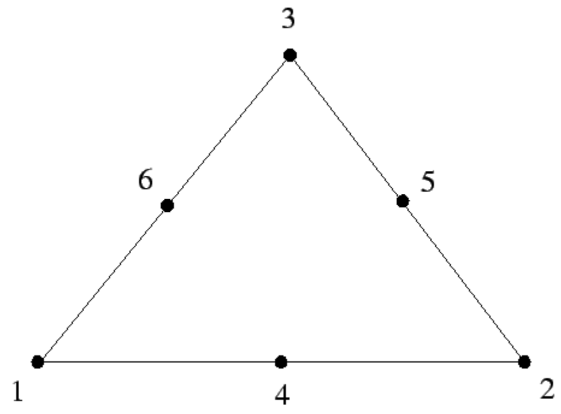
10 node tet element (type 6)

Type 6



3 node shell element (tr3, tr3u, type 7)

Type 7

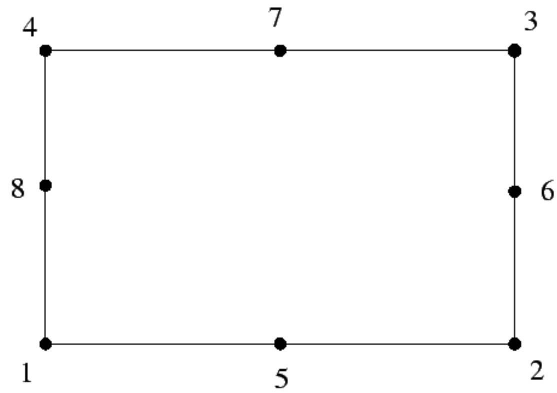


6 node shell element (tr6, type 8)

Type 8

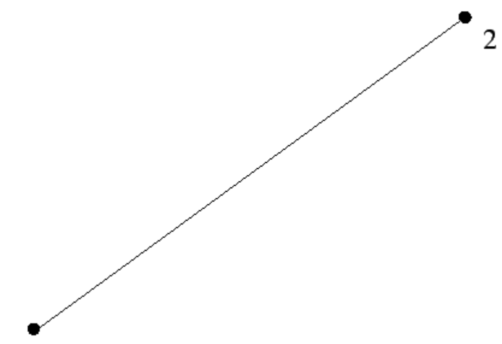
4 node shell element (qu4, type 9)

Type 9



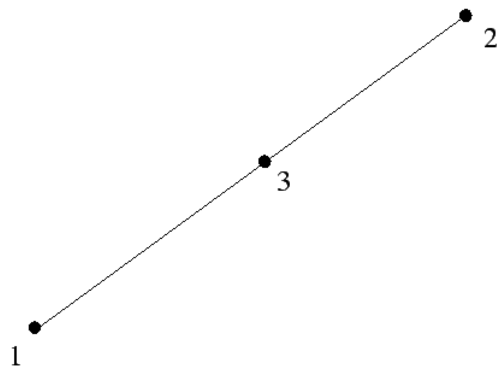
8 node shell element (qu8, type 10)

Type 10



2 node beam element (be2, type 11)

Type 11



3 node beam element (be3, type 12)

Type 12