A Primer for performing Static and Dynamic Finite Element Analysis with CalculiX and NETGEN

Ali Sayed
Esar Works, Oakland, CA 94609

February 6, 2010

Contents

Revision History

Initial Release

Chapter 1

Introduction

1.1 Purpose

The purpose of this tutorial is to demonstrate the use of **CalculiX** and **NETGEN** for performing static and dynamic analysis of a simple part, and to introduce the user to the various post-processing capabilities of **CalculiX**. This document is not intended to be an introduction to the use of **CalculiX**, for that the user is referred to introductory tutorials.[1, 2]

This tutorial was prepared using **CalculiX** and **NETGEN** on a Debian Linux (squeeze) system using pre-complied packages[8]. If you do not have access to a Debian Linux installation you may use CAELinux[3] which is a live DVD distribution based on Ubuntu and contains many useful CAE tools. Pre-complied Windows Binaries are also available[2], however each of these options has some minor limitations:

- **NETGEN**[8, 10] package included with CAELinux is not compiled with OpenCascade support, therefore does not support import of IGES for STEP geometry, STL format is supported.
- Windows version of **CalculiX** does not ship with **Gnuplot**[7], hence does not support Time-History Plots. I have not had any interest in trying to install or configure Gnuplot in Windows to see if it works with **CalculiX.**

1.2 Comments and Suggestions

If you have comments or suggestions regarding the content of this tutorial or you need further assistance with installation and configuration of packages on your Linux system please email me via the **CalculiX** user group.[9]

1.3 Problem definition

The example problem used in this tutorial is taken from ABAQUS Documentation[6] (A Student version can be purchased on-line). The reasoning for doing this is quite simple, I wanted to a have a reference to compare my results to and I will present the ABAQUS results throughout this document, not to make comparisons but simply to confirm that we are proceeding correctly. If you have access to the ABAQUS documentation you may refer to it for further details.

A simple 3D lug is modeled as shown in fig 1.1.

The rear end is welded to a massive structure. Loads applied by a bolt placed in the hole will be simulated as pressure loads in the 2-direction (Y-direction). The load is 30kN, using the geometry of the the bolt hole this equates to:

$$\frac{30\,kN}{2 * 0.015\,m * 0.02\,m} = 50\,MPa$$

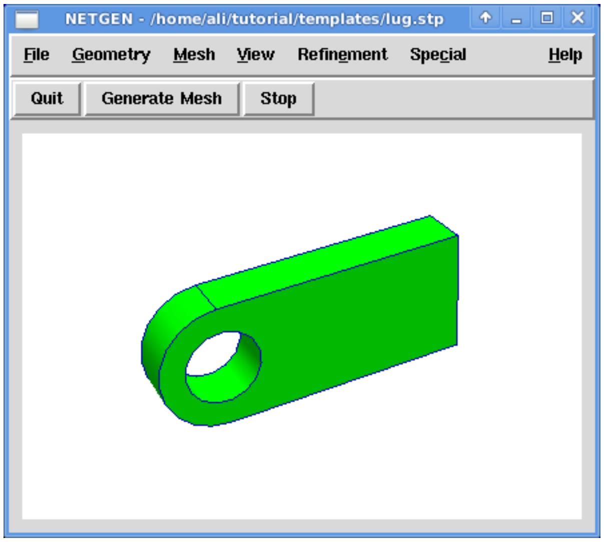

Figure 1.1:

Geometry will initially be modeled using the **CalculiX** pre-processor, **cgx** environment, using SI units, next we will import an IGES model into **NETGEN**[1], and use the auto mesher to mesh it, export the mesh to Abaqus format, and finally use **ccx** to run the analysis.

Five types of analysis will be performed:

1. Linear Static Analysis
   We will observe how the results compare to Abaqus
2. Modal Dynamic Analysis
   Compute the natural frequencies of the structure and evaluate its response to dynamic loading, note that this particular loading condition causes the lug to oscillate and this will be show using Time-History Plots.
3. Modal Dynamic Analysis - Steady State
   This will repeat the above problem except we will use the STEADY STATE parameter and the introduce damping in the structure.
4. Dynamic Analysis - Explicit Procedure
   Direct integration dynamic analysis, this case uses the Explicit approach.
5. Dynamic Analysis - Implicit Procedure
   Direct integration dynamic analysis, this case uses the Implicit approach.

1.4 Using a Text Editor to prepare input

Your FEA toolkit must include a good text editor, one you are comfortable with. I will use **SciTE**[11] frequently to edit or combine files and prepare input for the analysis. This in my opinion is the most productive why to run several test cases.

I have provided templates and complete input files for the problems covered in this tutorial, the templates can be used as starting points for other problems as well.

It is recommend, but not required, that you become familiar with **SciTE**. It can be installed on any Linux system[2]. If you are using the Windows version of **CalculiX** then **SciTE** is included in your installation.

My preference for using **SciTE** is that it offers syntax highlighting for Abaqus files, although this is not enabled by default on most systems. I believe **vim** and **Emacs** also support highlighting

[1] The model is provided as lug.iges for those who don't have access to a CAD package to build there own

[2] For Debian systems it is part of the repositories, installation instructions are beyond the scope of this document

for ABAQUS format, I have only experimented with vim so far, it works but its a bit hard to read on the screen.

1.4.1 SciTE and Syntax Highlighting

This is my hack, some of the changes might be unnecessary but I find that this works well for me.

Open a terminal window, and either become root or use sudo. Launch **SciTE** as root, else you will not be able to modify the Global Properties file.

Options ▷ Open Global Options File

and add the following lines in the appropriate sections: [3]

```
$(filter.abaqus)\
$(filter.apdl)\
$(filter.fbd)\
ABAQUS|inp||\
ANSYS|apdl||\
FBD|fbd||\
import abaqus
import apdl
import fbd
```

Save the file and quit **SciTE**, you may log out as root. Now when you open an ABAQUS syntax file the keywords will be highlighted.

1.5 Units [4]

Geometry modeled within the **CalculiX** environment uses SI units, however while writing the tutorial I discovered that **NETGEN** (OpenCascade specifically) converts length units from meter to mm no matter what options I set in my CAD package before exporting the model. These differences are summarized as follows:

Type or Symbol	Value (m,kg,s,K)	Value (mm,N,s,K)
Youngs Modulus (E)	2.0 E+11 $\frac{N}{m^2}$	2.0 E+5 $\frac{N}{mm^2}$
Input Coordinates (length)	as meters	as mm
Density (ρ)	7850 $\frac{kg}{m^3}$	7.85E-9 $\frac{g}{mm^3}$
Applied Pressure	50. E+6 $\frac{N}{m^2}$	50.0 $\frac{N}{mm^2}$
Calculated Stress	MPa $\frac{N}{m^2}$	Pa $\frac{N}{mm^2}$
Displacement	as meters	as mm

1.6 Reading Input files:

Since this is not meant to be a first introduction to **CalculiX,** one should already be familiar with how to read and write the various files needed to perform basic analysis. Here is a quick summary:

- To read in a geometry file (.fbd)

  ```
  cgx -b <file_name.fbd>
  ```

- To read an ABAQUS format file (from a **NETGEN** export)

  ```
  cgx -c <file_name.inp>
  ```

[3]note: the backslash (\) is necessary

[4]Obviously this document could have been written using only one set of units, however I thought it would be interesting to write it in this manner since it would help explain the importance of switching from one to the other. It took me a while to get used to it and I'm hoping that by doing it in this manner I will help those who might still be struggling with the notion the FEA code does not know or care for Units. Its the user responsibility.

- To read results along with the model file (so that sets can be referenced)

  `cgx -v <file_name.frd> <file_name.inp>` [5]

1.7 Template listing in Detail

As mentioned earlier, the input to the solver will be prepared using a text editor. All modeling and meshing is done graphically, however various plain text files need to be combined to prepare the complete analysis job file and this is where use of a text editor and a basic template for the particular type of analysis can really make this a painless process.

The supplied templates are almost identical to the point where the analysis step begins (keyword *STEP). All that is required is to populate the file with the data (for nodes below *NODE, NSET=Nall, elements below *ELEMENT, type= ..., etc, etc.) make some other minor adjustment (using **SciTE** in this case) and submit the job to **ccx** for analysis.

CalculiX keywords are always preceded by * (e.g. *DLOAD) comments begin with ** (**any questions ?)

Here is the listing of the five template files:

[5] Without the second <file_name.inp> one cannot do Time-History Plots since no data is available regarding set definitions.

Linear Static Analysis

```
**
** Structure: Abaqus LUG example with C3D10 elements
** Test objective: Netgen for Meshing, length units in mm
** OpenCascade converts IGES files to mm for length
**
*HEADING
Static Analysis
**
*NODE, NSET=Nall
*ELEMENT, type=C3D10, ELSET=Eall
** Names based on fix1
*NSET,NSET=Nfix1
** Location Bottom of hole
*NSET,NSET=N1
** Location Attach point
*NSET,NSET=N2
*BOUNDARY
Nfix1,1,3
**
** Material Properties converted to suit length units in mm.
*MATERIAL, name=steel
*DENSITY
7.85E-9,
*ELASTIC
200000, 0.30
*SOLID SECTION,ELSET=Eall,MATERIAL=steel
**
**
*STEP
*STATIC
*DLOAD
** Pressure based on load (50 Pa)
*NODE FILE
U
*EL FILE
S, ZZS
*NODE PRINT, NSET=Nall
U
*EL PRINT, ELSET=Eall
S
*END STEP
```

Modal Dynamic Analysis

```
**
**   Structure: Abaqus LUG example with C3D10 elements
**   Test objective: Netgen for Meshing length units in mm
**   OpenCascade converts IGES files to mm for length
**
**   Test objective: dynamic response to a constant impact; no damping.
*HEADING
Modal Dynamic Analysis
**
**
*NODE, NSET=Nall
*ELEMENT, type=C3D10, ELSET=Eall
** Names based on fix1
*NSET,NSET=Nfix1

** Location Bottom of hole
*NSET,NSET=N1
** Location Attach point
*NSET,NSET=N2
*BOUNDARY
Nfix1,1,3
**
**  Material Properties converted to suit length units in mm.
*MATERIAL, name=steel
*DENSITY
7.85E-9,
*ELASTIC
200000, 0.30
*SOLID SECTION,ELSET=Eall,MATERIAL=steel
**
**
*AMPLITUDE,NAME=A1
0.,1.,5.E-4,1.
*STEP
*FREQUENCY,SOLVER=ARPACK,STORAGE=YES
10,0.01
*NODE FILE
U
*EL FILE,POSITION=AVERAGED AT NODES
S,SINV
*END STEP
*STEP
*MODAL DYNAMIC
** 1.E-5,1.E-4       this value was too small to produce desired results
1.E-5,1.E-3
*DLOAD,AMPLITUDE=A1
** Pressure based on load (50 Pa)

*NODE PRINT,NSET=N1
U
*END STEP
```

Modal Dynamic Analysis with Rayleigh Damping - Steady State Solution

```
**
**   Structure: Abaqus LUG example with C3D10 elements
**   Test objective: Netgen for Meshing length units in mm
**   OpenCascade converts IGES files to mm for length
**
**   Test objective:  dynamic response to a constant impact; check of STEADY STATE parameter
*HEADING
Modal Dynamic Analysis with STEADY STATE
**
**
*NODE, NSET=Nall
*ELEMENT, type=C3D10, ELSET=Eall
** Names based on fix1
*NSET,NSET=Nfix1

** Location Bottom of hole
*NSET,NSET=N1
** Location Attach point
*NSET,NSET=N2
*BOUNDARY
Nfix1,1,3
**
**  Material Properties converted to suit length units in mm.
*MATERIAL, name=steel
*DENSITY
7.85E-9,
*ELASTIC
200000, 0.30
*SOLID SECTION,ELSET=Eall,MATERIAL=steel
**
**
*AMPLITUDE,NAME=A1
0.,1.,5.E-4,1.
*STEP
*FREQUENCY,SOLVER=ARPACK,STORAGE=YES
10,0.01
**NODE FILE
**U
**EL FILE,POSITION=AVERAGED AT NODES
**S,SINV
*END STEP
*STEP
** Second line parametes are different when STEADY STATE is used, see docs
*MODAL DYNAMIC,STEADY STATE
** Play with the Error prameter
1.E-5,1.E-3
*MODAL DAMPING,RAYLEIGH
,,20000.,2.e-4
*DLOAD,AMPLITUDE=A1
** Pressure based on load (50 Pa)
*NODE PRINT,NSET=N1
U
*NODE FILE
U
*END STEP
```

Explicit Dynamic Analysis

```
**
**   Structure: Abaqus LUG example with C3D10 elements
**   Test objective: Netgen for Meshing length units in mm
**   OpenCascade converts IGES files to mm for length
**
**   Test objective: nonlinear dynamic response to a constant
**                    impact; no damping; explicit procedure.
*HEADING
Explicit Dynamic Analysis
**
**
*NODE, NSET=Nall
*ELEMENT, type=C3D10, ELSET=Eall
** Names based on fix1
*NSET,NSET=Nfix1
** Location Bottom of hole
*NSET,NSET=N1
** Location Attach point
*NSET,NSET=N2
*BOUNDARY
Nfix1,1,3
**
**  Material Properties converted to suit length units in mm.
*MATERIAL, name=steel
*DENSITY
7.85E-9,
*ELASTIC
 200000, 0.30
*SOLID SECTION,ELSET=Eall,MATERIAL=steel
**
**
*AMPLITUDE,NAME=A1
0.,1.,5.E-4,1.
*STEP,INC=100000
*DYNAMIC,EXPLICIT
1.E-7,5.E-3
*DLOAD,AMPLITUDE=A1
** Pressure based on load (50 Pa)
*NODE PRINT,NSET=N1
U
*NODE FILE
U
*EL FILE
S
*END STEP
```

Implicit Dynamic Analysis

```
**
** Structure: Abaqus LUG example with C3D10 elements
** Test objective: Netgen for Meshing length units in mm
** OpenCascade converts IGES files to mm for length
**
** Test objective: nonlinear dynamic response to a constant
** impact; no damping; implicit procedure.
*HEADING
Implicit Dynamic Analysis
**
**
*NODE, NSET=Nall
*ELEMENT, type=C3D10, ELSET=Eall
** Names based on fix1
*NSET,NSET=Nfix1
** Location Bottom of hole
*NSET,NSET=N1
** Location Attach point
*NSET,NSET=N2
*BOUNDARY
Nfix1,1,3
**
** Material Properties converted to suit length units in mm.
*MATERIAL, name=steel
*DENSITY
7.85E-9,
*ELASTIC
200000, 0.30
*SOLID SECTION,ELSET=Eall,MATERIAL=steel
**
**
*AMPLITUDE,NAME=A1
0.,1.,5.E-4,1.
*STEP,INC=100000
*DYNAMIC,DIRECT
1.E-5,5.E-3
*DLOAD,AMPLITUDE=A1
** Pressure based on load (50 Pa)
*NODE PRINT,NSET=N1
U
*NODE FILE
U
*EL FILE
S
*END STEP
```

Chapter 2

Model Building and Mesh Generation

2.1 Building the Basic Model of the Lug in CGX

I will use the supplied file *point-data.fbd* as the starting point for the model. You may open the file in a text editor to examine its contents, the command PNT ! X Y Z simply means create a new point at coordinates X, Y, Z and automatically number the point. If you prefer to control the point numbers, simply replace the PNT ! with the next number

e.g PNT 001 X Y Z[5]

Copy the file *point-data.fbd* to a new directory, lets call the directory *cgx-model*, and launch **cgx** to read the file in:

```
cgx -b point-data.fbd
```

You should see the **cgx** graphics window with 13 blue dots (the points) as shown in Figure 2.1

```
rot -z
```

Will rotate and orient the view to the X-Y plane.

```
plot pa all
```

Will draw the points with numbering on

Now let's draw the lines to represent the profile of the lug, you should be familiar with the basic commands to select entities individually or as a selection window.[1, 2, 5]

Before drawing the lines, type:

```
plus l all
```

To make sure the lines appear in the window as they are being drawn.

Use the **qlin** command to draw the profile, note the number of line segments in Figure 2.2.

Next let us draw the arcs to complete the profile. Arcs are drawn with the **qlin** command, the only difference is that you define a start point, a center and an end point. In our case the center of all the arcs is the point P009 as shown in Figure 2.3. You might want to zoom in before drawing the arcs. type:

```
qlin
```

move mouse focus to point P006 and type **b** to begin selection

move mouse focus P009 and type **c** to define the center of the arc

finally move mouse focus to P00G and type **g** to generate the arc as shown in the Figure 2.3

Line segment L009 is the newly created arc.

Draw the remaining arcs and the connecting segments (L00H, L00I, L00J, L00K) to complete the profile as shown in Figure 2.4. A defined surface cannot have a hole in it, hence we need those four segments to later define the surfaces.

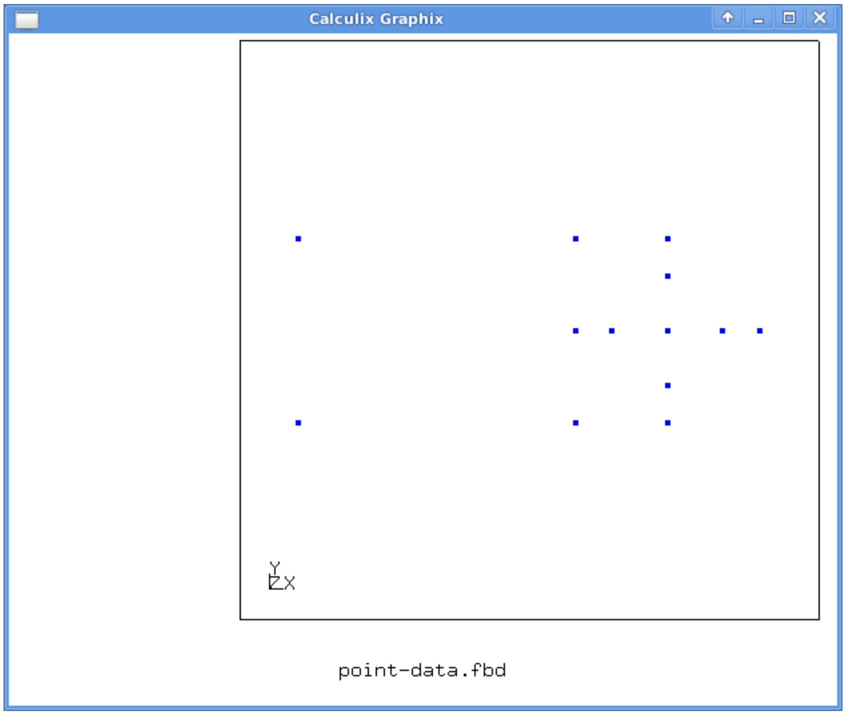

Figure 2.1:

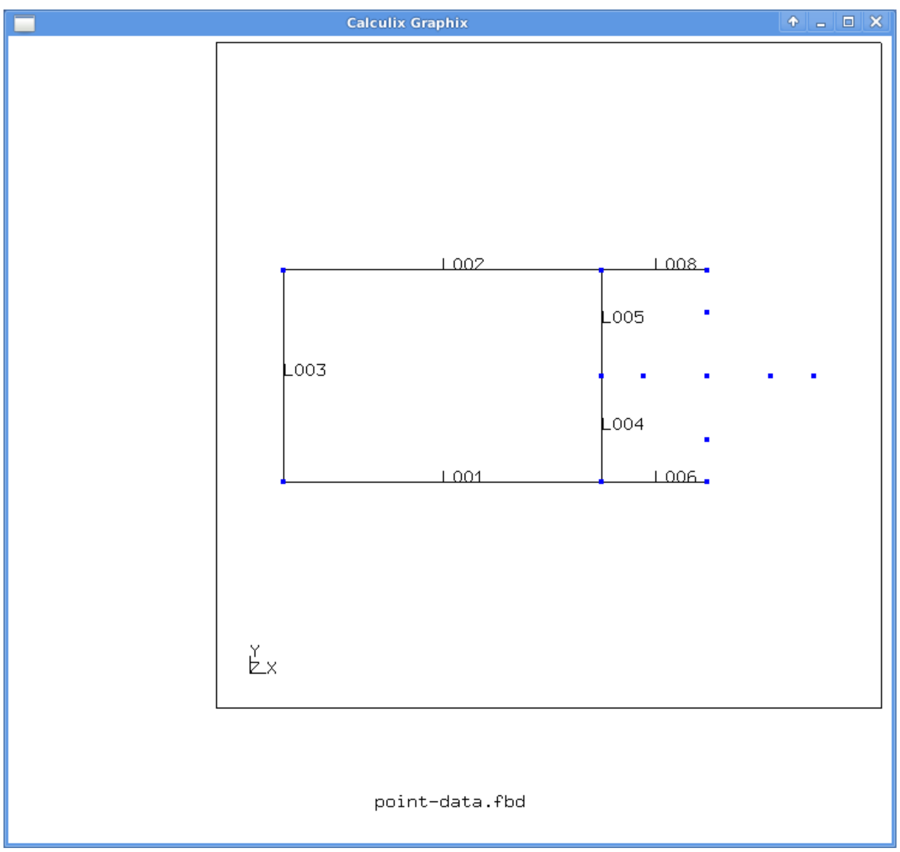

Figure 2.2:

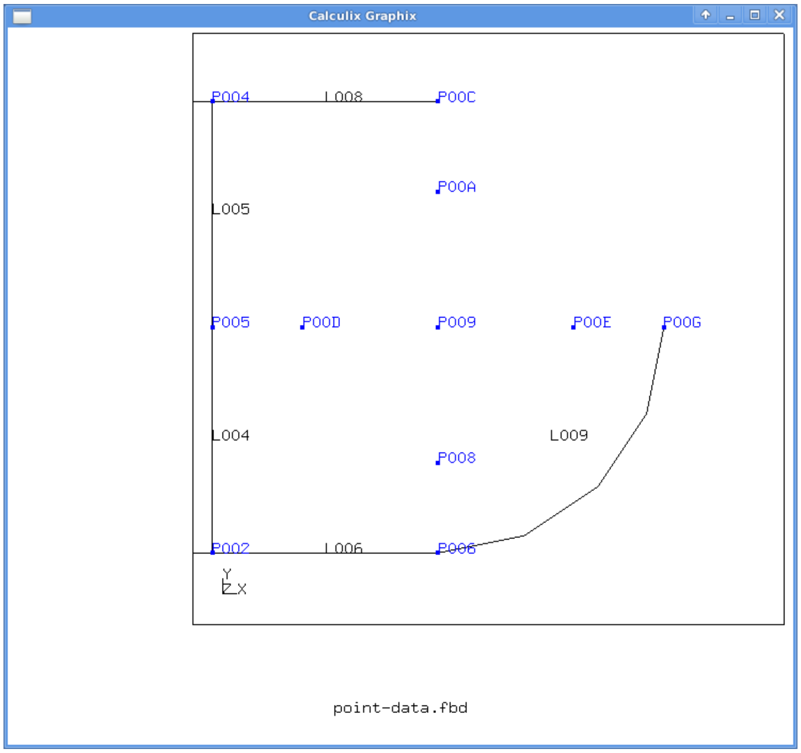

Figure 2.3:

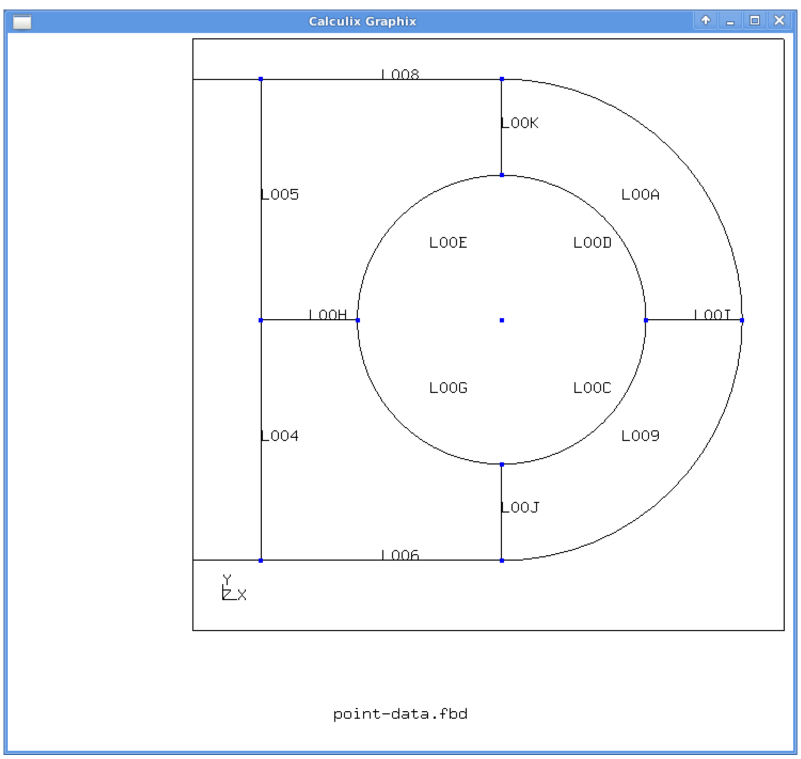

Figure 2.4:

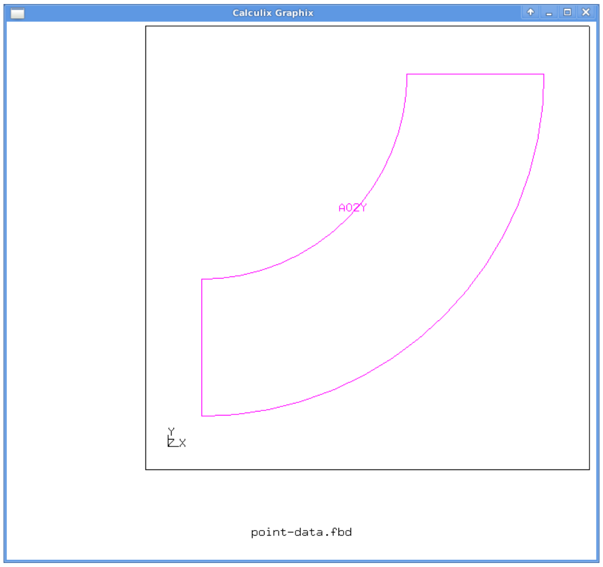

Figure 2.5:

Now let us construct the surfaces (total of 5 surfaces). Surfaces A030 and A032 and A033 are made up from five line segments and surface A02Y and A02Z have four segments each as seen in Figure 2.9. Let us start with the easy ones first. Use Figure 2.4 to locate the line callout below. Type:

`plus sa all` [1]

1. qsur
2. focus mouse on L009 and type 1
3. focus mouse on L00I and type 2
4. focus mouse on L00C and type 3
5. focus mouse on L00J and type 4
6. type g to generate the first surface

Your first surface should look like Figure 2.5 if defined correctly[2].

Repeat the **qsur** procedure to draw the adjacent surface using segments L00A, L00K, L00D, L00I.

If you don't get them right the first time, type:

```
plot sa all
```

And then **qdel** (enable window select) and select the surface to delete and start over, the names are not important since **cgx** auto numbers and we are only going to be concerned with set lists and not the names of each item. Figure 2.6 show the completed surfaces.

Type:

```
plot la all
```

to draw all the lines. Next Type:

```
plus s all
```

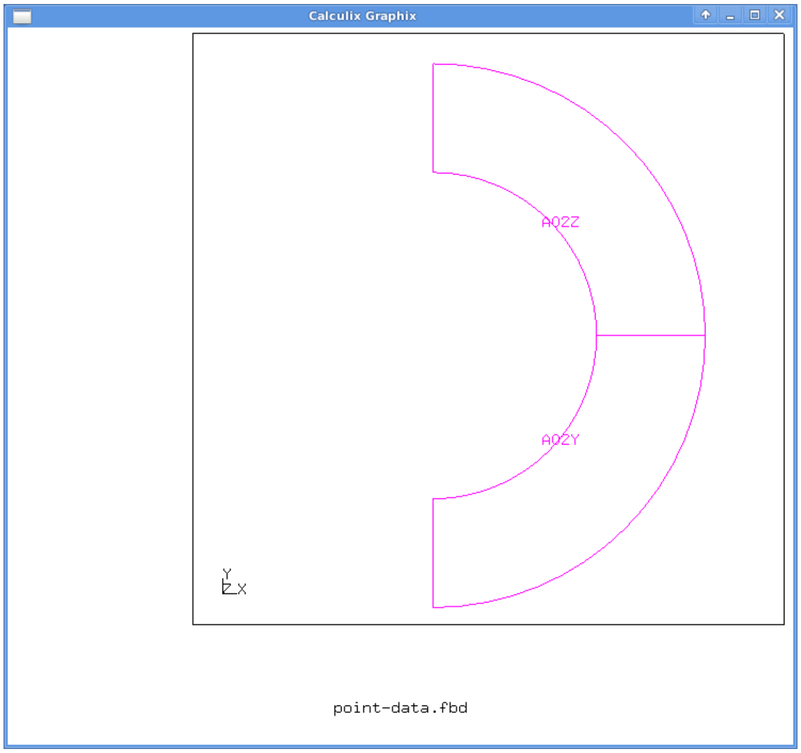

Figure 2.6:

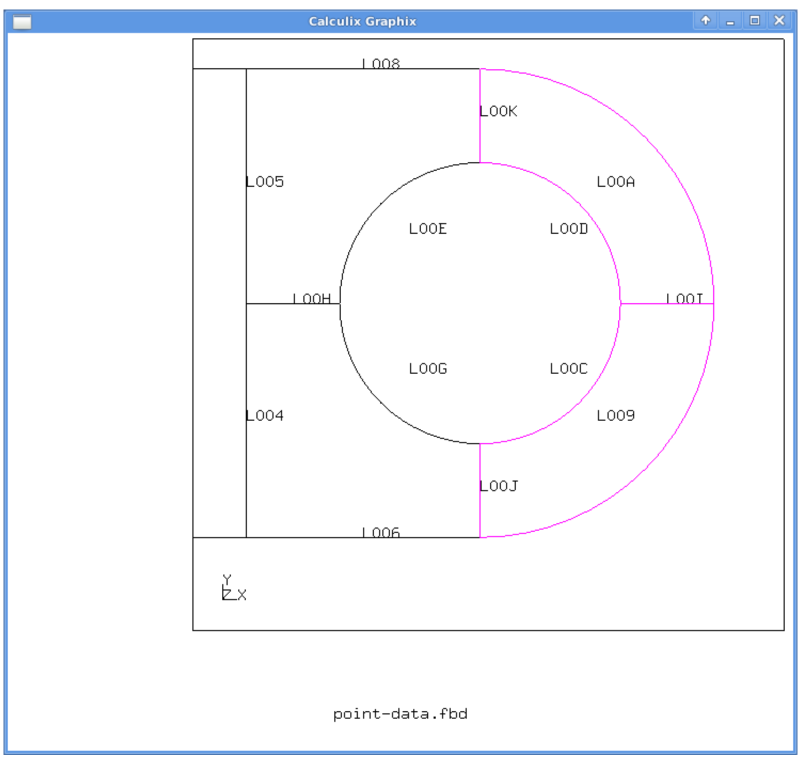

Figure 2.7:

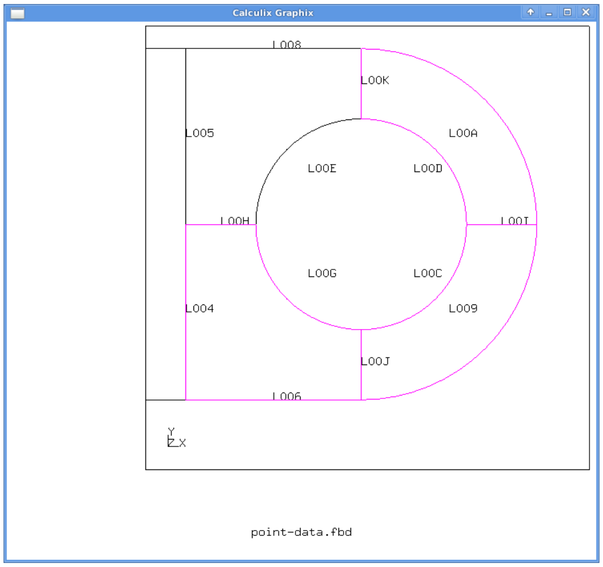

Figure 2.8:

to add surface to the graphics window. Zoom in to make selection easy. See Figure 2.7

Now let us define the surfaces with five segments each. Type:

1. qsur
2. focus mouse on L006 and type 1
3. focus mouse on L00J and type 2
4. focus mouse on L00G and type 3
5. focus mouse on L00H and type 4
6. focus mouse on L004 and type 5
7. type g to generate the surface

You should have something similar to the Figure 2.8.

Repeat for the other surface using sequence L008, L005, L00H, L00E, L00K.

To zoom out and see the complete model, type:

```
frame
```

Finally define the last surface, type:

1. qsur
2. focus mouse on L003 and type 1
3. focus mouse on L001 and type 2
4. focus mouse on L004 and type 3
5. focus mouse on L005 and type 3
6. focus mouse on L002 and type 4
7. type g to generate the surface [3]

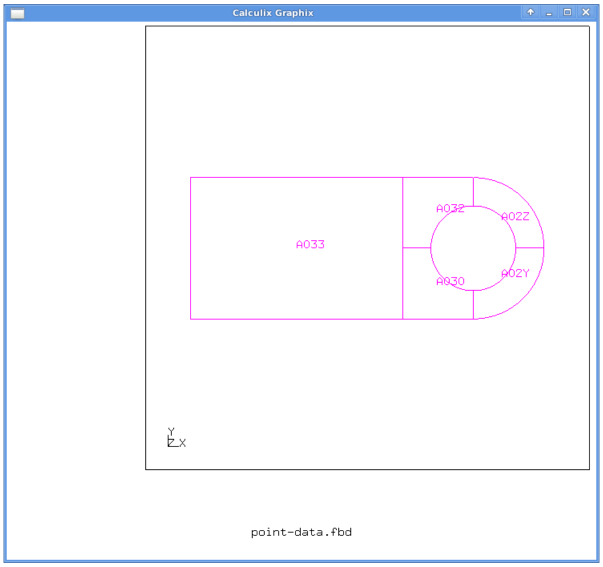

Figure 2.9:

So far all we have is a 2D profile of the lug as defined by surfaces[4]. See Figure 2.9

Next we will create a set list [5] consisting of the five surfaces and perform a simple "extrude" to generate a 3D body for meshing. Type:

```
prnt se
```

To list defined sets.

You should see a set named "all" with 13 points, 17 lines and five surface (p:13 l:17 s:5). Type:

```
qadd se1
```

Turn on window select and make sure the window is large enough to capture all the surfaces. See Figure 2.10

```
type s
prnt se
```

[6]

you should see a new set called "se1" with five surfaces. if not ...

```
zap se1
```

to delete set "se1" and start over. When you are done adding all surfaces to set se1, type:

```
swep se1 se2 tra 0 0 .02
```

this will translate the profile in the 3-direction (z-direction) to create a solid body representing the lug.

rotate the model using the mouse in graphics window to see the swept profile, as shown in Figure 2.11

```
plot ldiv all
```

[1]surface will appear on the screen as they are drawn, to avoid confusion.

[2]use: plot sa all to verify

[3]Note the use of five segments to define only four sides of a surface, this is completely valid.

[4]The names of entities are not important and yours may vary depending on the auto numbering that **cgx** performed or how many attempts it took to get it right.

[5]entity groups

[6]selects all surface in the capture window

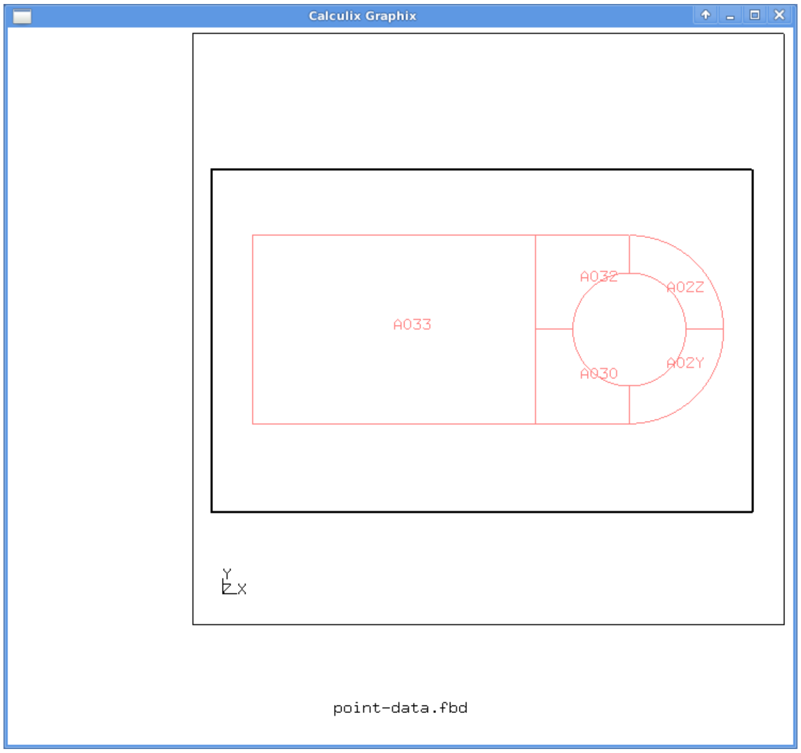

Figure 2.10:

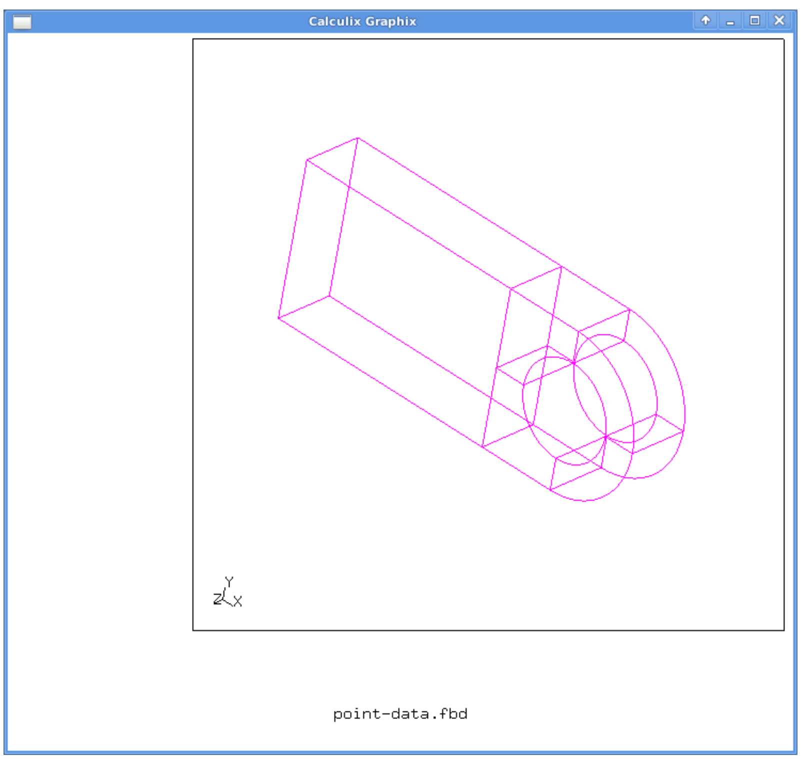

Figure 2.11:

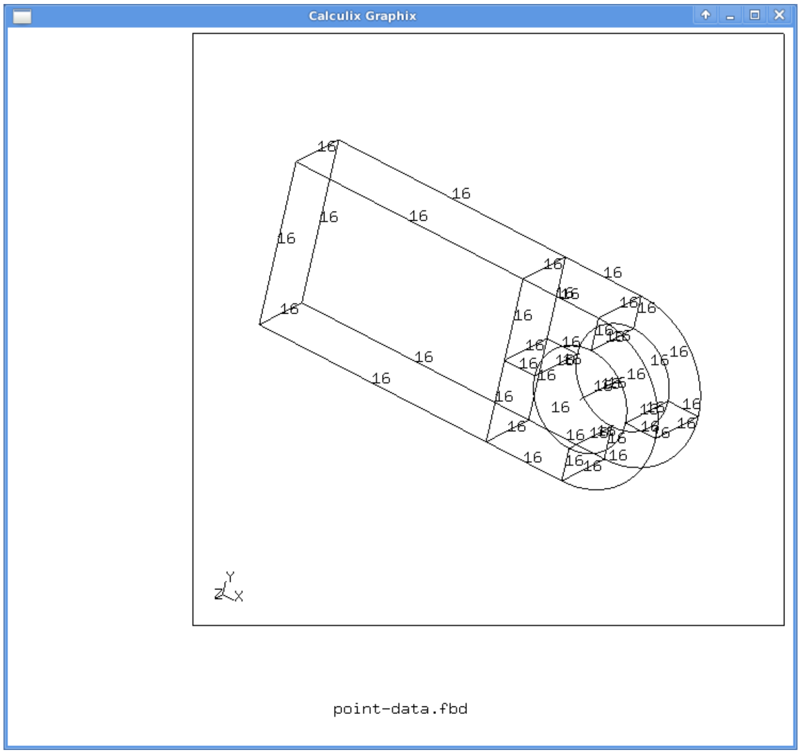

Figure 2.12:

This shows that all the line segments have four divisions, this controls the mesh density, we will change it to make the mesh suit our needs. Type the following:

```
div all auto
plot ldiv all
```

You notice that the lines now have 98 segments each, this is too large a value to compute a mesh for such a simple part.

```
div all 16
```

Each line is divided into 16 segments as shown in Figure 2.12

Let us quickly delete the line that was generated when the center point of the hole was swept in the z-direction, type:

```
plot la all
```

zoom in to get a closer view, type:

```
qdel
```

and enable window selection

type l [7] to select the line and q when done to delete the line. See Figure 2.13

```
frame
```

to zoom out again, and type:

```
plot ldiv all
```

[7] as in L not the number one

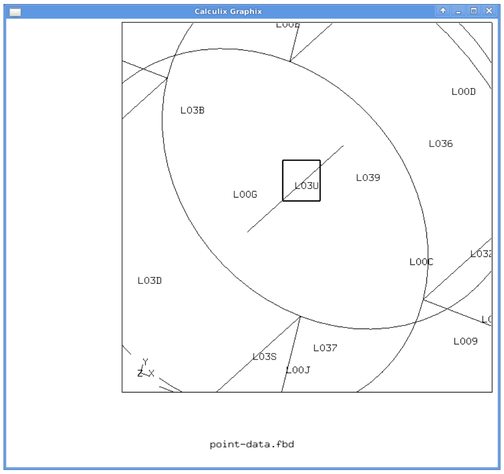

Figure 2.13:

to switch back to line division view.

Now we will define the element type to use for meshing. [8]

```
elty all he20
mesh all
```

Wait for the ready prompt in the command window, the warnings can be ignored. If you do not see the ready prompt then the mesh was not computed successfully. Type:

```
plot e all
```

to display the mesh. See Figure 2.14.

From the graphics menu in the **cgx** window:

enable "Toggle Element Edges" and rotate the model to see the mesh density and element edges.

If you are not satisfied with the mesh or would like to perform studies involving various mesh densities [9] you can delete the mesh and start over.

```
del mesh all
```

change the line divisions

```
div all XX
```
[10]

and recreate the mesh.

to view the listing of generated nodes and elements

```
type prnt se all
prnt se
1 all stat:o n:22761 e:4864 f:2176 p:26 l:46 s:28 b:5 L:0 S:0 se:0 2
se1 stat:c n:1645 e:0 f:438 p:0 l:0 s:5 b:0 L:0 S:0 se:0 3
-NJBY stat:c n:0 e:0 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0 4
se2 stat:c n:1912 e:0 f:608 p:13 l:17 s:5 b:0 L:0 S:0 se:0
```

Note that there are 22761 nodes and 4864 elements in the model. Lets change this and make a course mesh. Figure 2.15.

[8]he20: 20 Node brick element, type 4
[9]Remember: ldiv controls mesh density
[10]XX being a numerical value

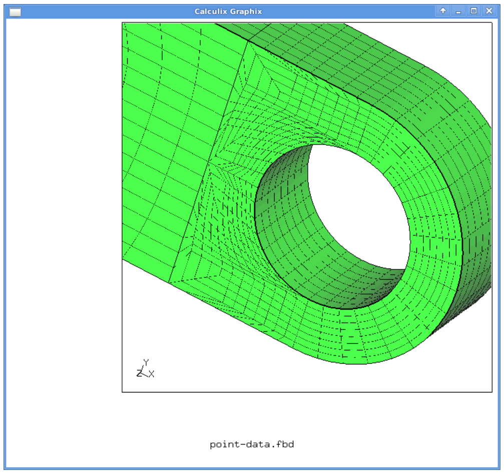

Figure 2.14:

```
del mesh all
div all 8
plot ldiv all
mesh all
plot e all
prnt se
1 all stat:o n:3269 e:608 f:544 p:26 l:46 s:28 b:5 L:0 S:0 se:0 2
se1 stat:c n:369 e:0 f:74 p:0 l:0 s:5 b:0 L:0 S:0 se:0 3
-NJBY stat:c n:0 e:0 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0 4
se2 stat:c n:500 e:0 f:152 p:13 l:17 s:5 b:0 L:0 S:0 se:0
```

Now we have 3269 nodes and 608 elements

From the **cgx** graphics window menu select :

```
Viewing ▷ Show Bad Elements
```

You should see a message in the terminal window as follows:

```
No bad elements in set:all
prnt se
```

The set -NJBY should have zero elements if there are no errors in the model.

```
3 -NJBY stat:c n:0 e:0 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
```

Let us save the model so we can prepare the output for analysis. Type:

```
send all abq
```

This will write the output of set all to a file called *all.msh*, this is essentially the node and element data that can be pasted into the templates to prepare the analysis file. You might want to open it for viewing in a text editor.

To read is back into **cgx** at a later time, type:

```
cgx -c all.msh
```

Next type:

```
send all fbd
```

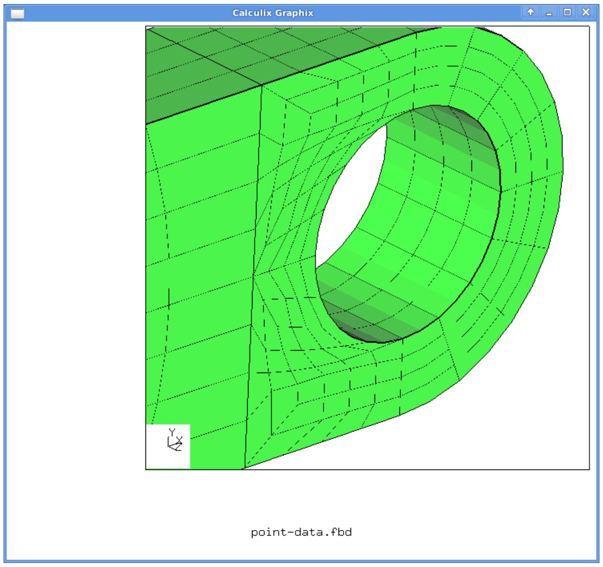

Figure 2.15:

Will write the geometry to a file called *all.fbd* in case you want to save or edit the model [11] or to work on it a later point.

```
send all frd
```

Creates a file called *all.frd* that can be opened in another session, however it is not used to submit a job directly to the solver **ccx**.

Now go ahead and save the files, you may quit the **cgx** session.

I will not cover the preparation of the analysis file using this model, instead I will be using the mesh generated by **NETGEN**. If you want to learn how to use this model for analysis please skip to Chapter 3. The next section demonstrates how to mesh using **NETGEN**. The complete analysis file using this mesh is included in the *run-samples* directory as *cgx-model.inp*. Results of the model in this section are discussed in the Appendix.

2.2 Using NETGEN to create the mesh

Create a new directory and copy the file *lug.iges* from the *templates* folder into this new directory.

To launch **NETGEN** with the model loaded, from the UNIX prompt type:

```
netgen lug.igs
```
[12]

If you are on Windows substitute with the appropriate procedure to launch the application and load the required file.

You should see the lug model as shown in Figure 1.1 Section 1.3.

From the **NETGEN** Menu:

Geometry ▷IGES/STEP Topology Explorer

On the Healing Tolerance selector click the down arrow to change the value to 1 e-9 [13]

Click the Heal geometry button. Figure 2.16

In the UNIX terminal window you'll see some messages ...

[11] as a text file

[12] STEP and STL files are also included for experimentation, the STL file has coordinates data in meters so substitute the units if you wish to use that for meshing and analysis, email me if you meed help using STL files

[13] although the default value of 0.001 should work just fine

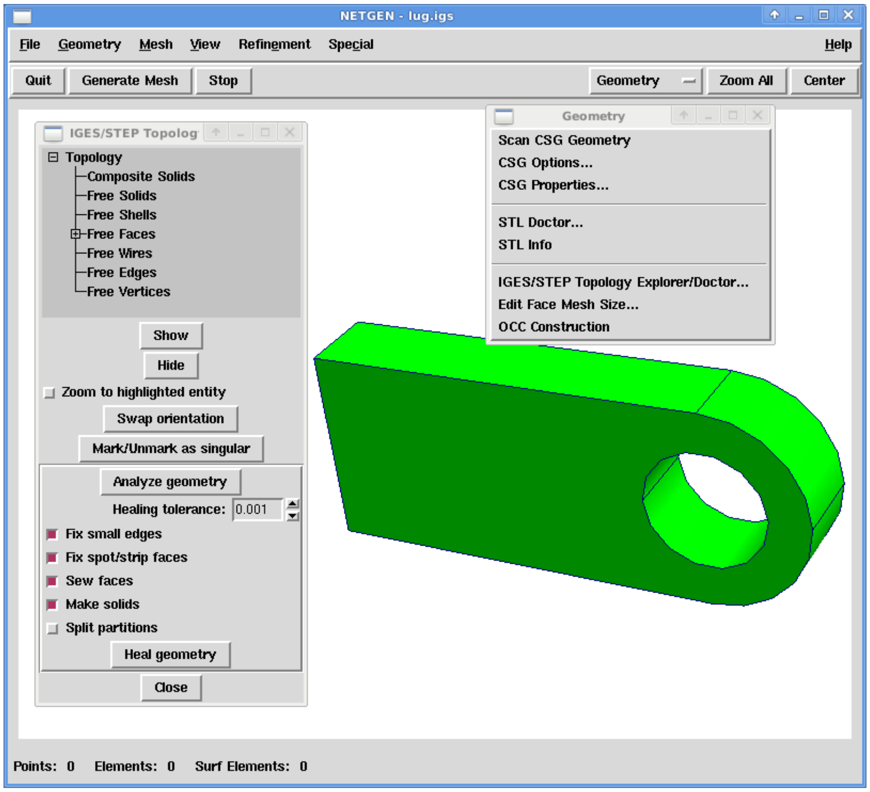

Figure 2.16:

```
Starting geometry healing procedure (tolerance: 1e-09)
-----------------------------------
- repairing faces
- fixing small edges
- fixing spot and strip faces
- sewing faces
- making solids -----------------------------------
Compounds : 0 (1)
Composite solids: 0 (0)
Solids : 1 (0)
Shells : 1 (0)
Wires : 11 (11)
Faces : 9 (9)
Edges : 21 (42)
Vertices : 14 (42)
Total surface area : 19005.3 (19005.3)
Preparing visualization (deflection = 0.01) ... done
Building topology tree ... done
```

Click the Close button to exit the dialog box
From the Main Menu:
Mesh ▷ Meshing Options
On the Meshing Options panel, under the General tab.
Select the Second order elements toggle
For Mesh granularity select very fine from the drop down list.
Click the Apply button and then the Done button to close the panel. Figure 2.17
In the **NETGEN** Main window, click the Generate Mesh button below the Main menu.
Check the UNIX terminal window for the Meshing done message, you may or may not see

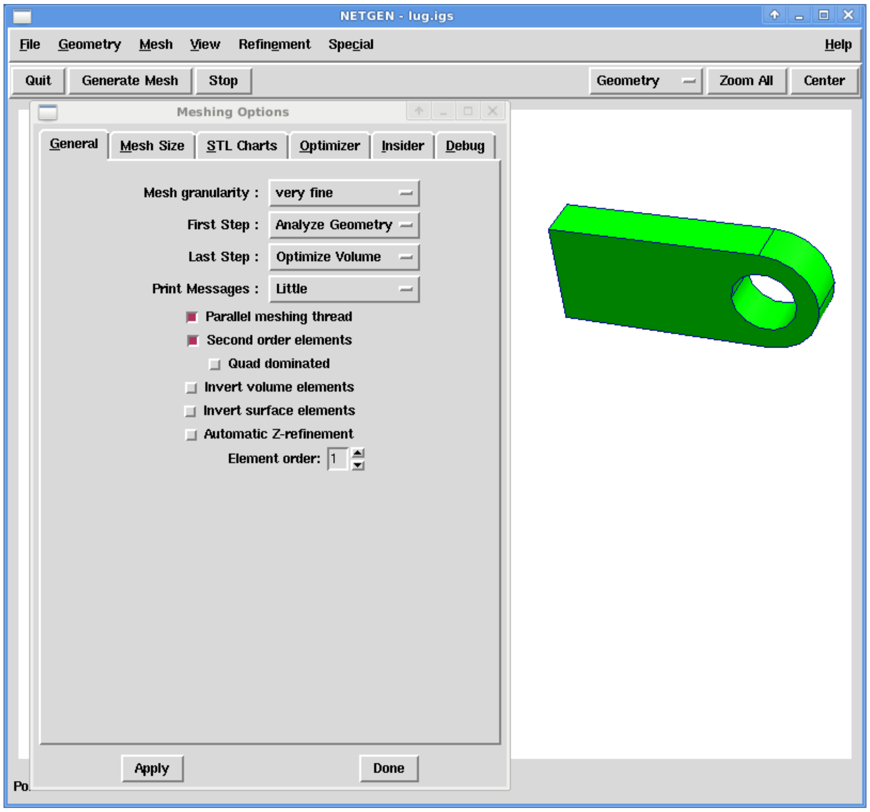

Figure 2.17:

anything in the **NETGEN** graphics window at this point.

```
Meshing subdomain 1 of 1
Delaunay meshing
start tetmeshing
Success !
1392 points, 6249 elements
Remove Illegal Elements
Volume Optimization
Meshing done, time = 1.98 sec
```

Just move the mouse in the graphics window while clicking the left mouse button (Rotates model) and the mesh will appear. Figure 2.18

ALL Done ... Just have to save the model. See Figure 2.19

From the Main Menu select:

File ▷ Export Filetype ▷ Abaqus Format (toggle on)

File ▷ Export Mesh

Name the file and specify the extension e.g. *lug.inp* [14]

You now have a plain text file called *lug.inp* in the directory where the IGES file is located and it can be viewed with **SciTE**, it contains a listing of the nodes and elements.

If you compare it to the one created in **cgx** (*all.msh*) you'll notice that the nodes defined here are indeed in units of mm, whereas those in *all.msh* are in meters.

Exit **NETGEN**, but do not close the UNIX terminal window, we will work in this directory for the next section. You can delete the IGES, STEP and STL files for now, we won't be needing them and there are copies in the *templates* directory anyway.

[14] I have had **NETGEN** crash sometimes when the extension was not specified, but that was on an older version running on Windows - the risk is not worth it if you have a large model and the mesh has been computing for hours and then it crashes right when you are done and ready to save your work.

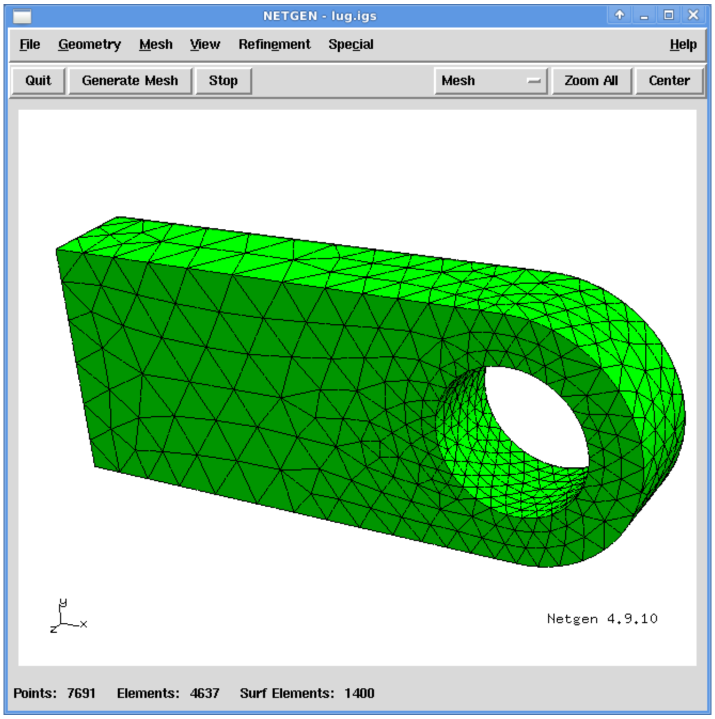

Figure 2.18:

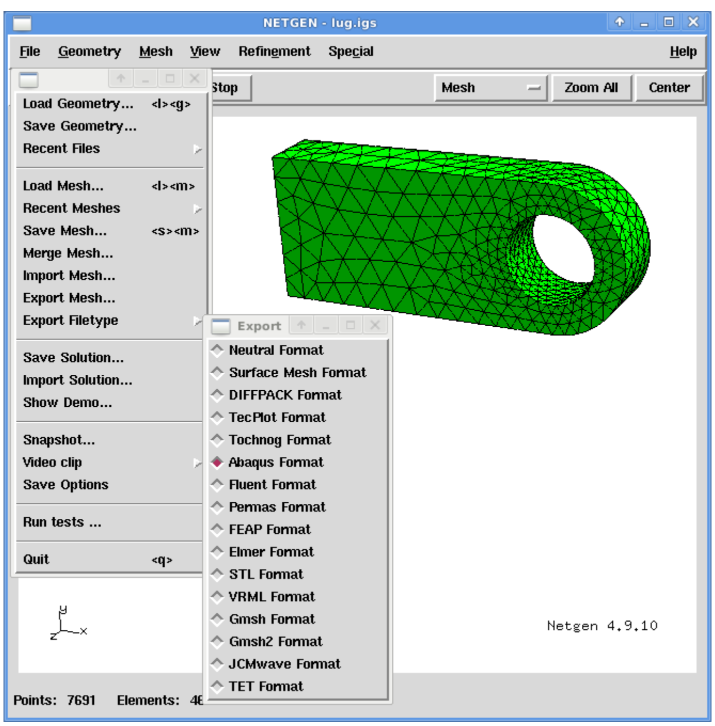

Figure 2.19:

Chapter 3

Applying Loads and Boundary Conditions

In order to apply loads and boundary conditions we need to define sets of nodes, elements, faces etc. For the purpose of this tutorial we will create four sets, these are common among all five jobs so we only define them once as long we use the same mesh for all cases. These are as follows:

1. The rear of the lug where it will be fixed in the x,y,z directions [1]
2. A set consisting of a single node on the rear corner to examine various results (e.g. max stress at the welded edge)
3. The faces on the lower half of the hole where the pressure loads will be applied
4. A set consisting of a single reference node inside the hole to do Time-History plots for dynamic analysis

Let us begin by loading the model into **cgx** ... at the UNIX shell prompt type:

`cgx -c lug.inp` [2]

The shell window will display the number and type of element in your model and the **cgx** graphics window will appear:

```
Elements: 4637 Nodes:7691 Datasets:0 MinElemNr: 1 MaxElemNr: 4637 MinNodNr:1 MaxNodNr:7691
read in 0.060000 sec
found elements of type 1: 0
found elements of type 2: 0
found elements of type 3: 0
found elements of type 4: 0
found elements of type 5: 0
found elements of type 6: 4637
found elements of type 7: 0
found elements of type 8: 0
found elements of type 9: 0 f
found elements of type 10: 0
found elements of type 11: 0
found elements of type 12: 0
```

[1] 1,2,3 direction in **CalculiX** and Abaqus terminology

[2] or whatever you called the **NETGEN** output during the save

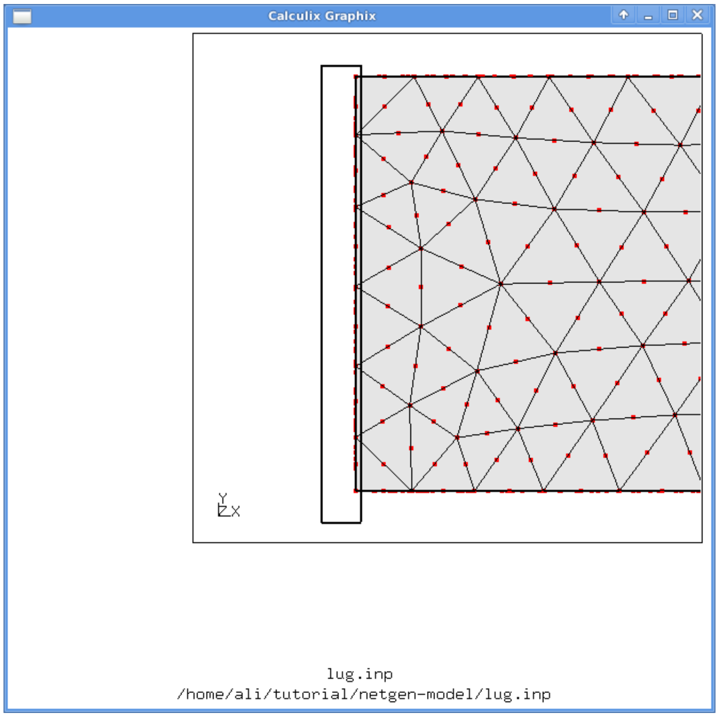

Figure 3.1:

3.1 Rear Fixed End (Set=fix1)

In the graphics window select "Toggle Element Edges" to see the mesh. Then select "Show Bad Element" and verify the you see the message "No bad elements in set:all " in the terminal window.

```
rot -z
```

will orient the lug in the X-Y plane

```
plus n all
```

will plot all the nodes along with the elements already displayed
zoom in to the view rear end of the lug
We will create a set name "fix1" with all the nodes on the rear boundary. type:

```
qadd fix1
```

activate window select [3] and create a long narrow window, Figure 3.1 to include all the nodes on the rear edge and type:
n to select nodes, followed by **q** to quit when done.
you may do **prnt se** to see what your window caught:

```
5 fix1 stat:c n:83 e:0 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
```

3.2 Rear Corner Node (Set=N2)

Rotate the model so you can see any one of the rear corner, does not matter which side, type:

```
qenq
```

[3] you should be an expert on this by now

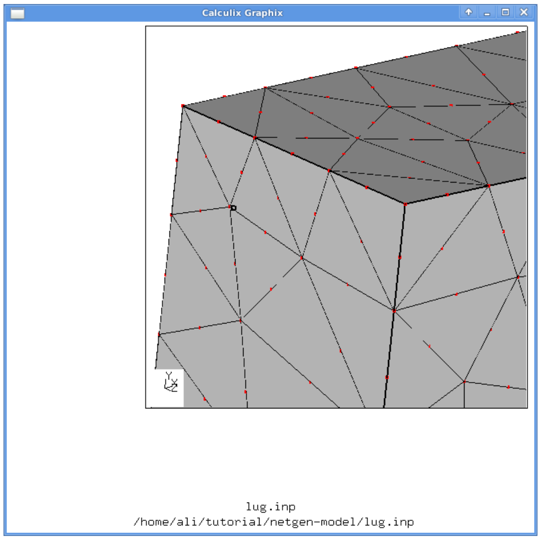

Figure 3.2:

and hover the mouse on any one corner node and type **n** to list the designated node.

You may do this repeatedly to identify any number of nodes.

The output is printed in the shell window listed below:

```
qenq
5 v= 0.000000e+00 xyz= 0.000000 50.000000 20.000000 axyz= 21.801409 0.000000 90.000000
rxyz= 53.851648 20.000000 50.000000     in set=fix1(5),
9 v= 0.000000e+00 xyz= 0.000000 0.000000 0.000000 axyz= nan nan nan
rxyz= 0.000000 0.000000 0.000000     in set=fix1(5),
```

Note the number of the node, in this case it will be 5 (or 9) this is what makes up the second set. See Figure 3.2.

3.3 Faces for Pressure Loading (Set=load)

```
plot f all
```

This will clear the display and plot just the faces (default color yellow)

```
rot -z
```

zoom in to focus on just the hole

Let us define a set of faces (for pressure loading)

```
qadd load
```

activate window select and type **f** to select faces, its OK to select more than what is required for now.

make sure the window is large enough to includes more than half of the lower portion of the hole as shown in Figure 3.3

Now we will remove the extra face in our set load

```
prnt se
6     load stat:c n:0 e:0 f:332 p:0 l:0 s:0 b:0 L:0 S:0 se:0
```

```
plus f load g
```

plots the selected faces in green for easy visual clues. Figure 3.4

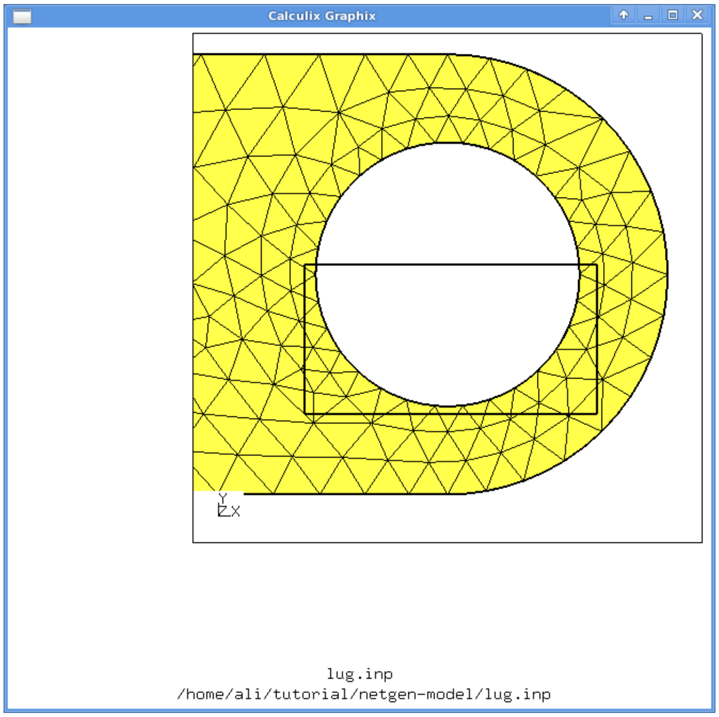

Figure 3.3:

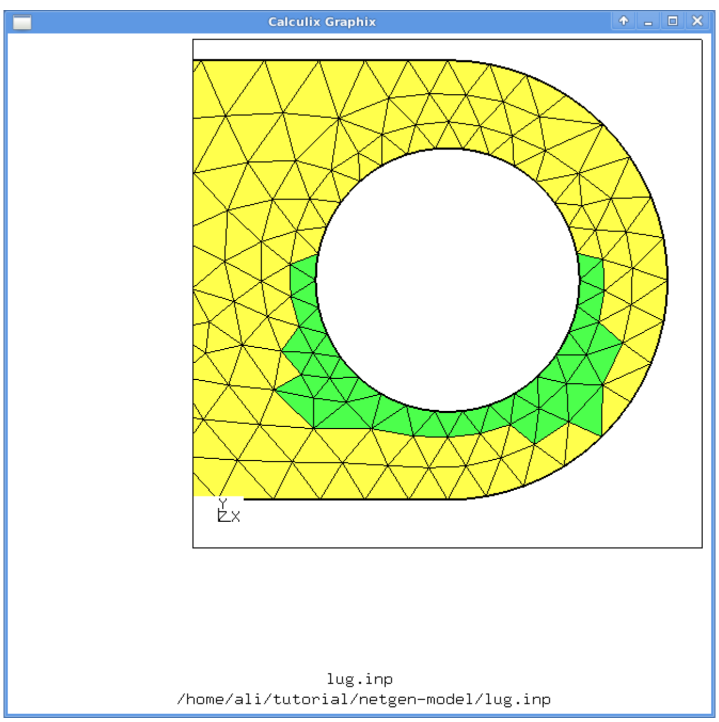

Figure 3.4:

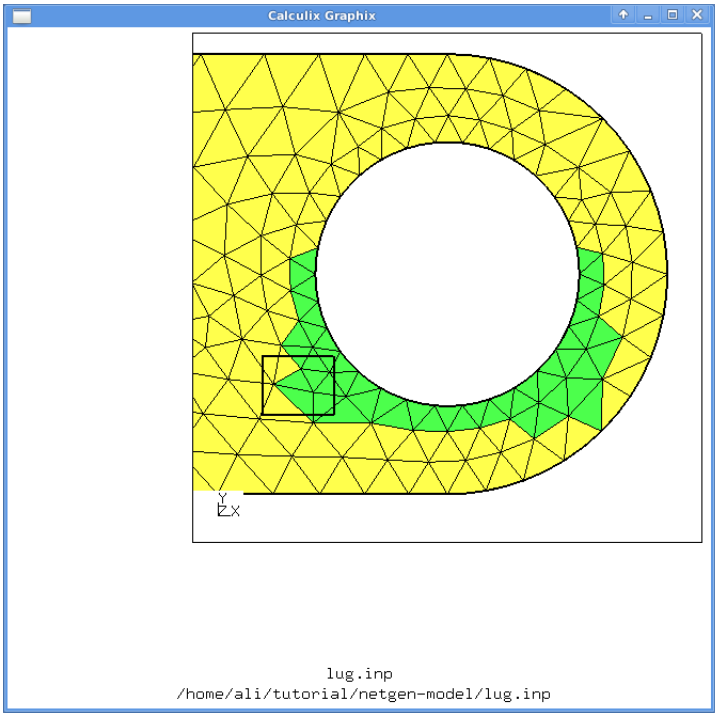

Figure 3.5:

```
qrem load
```

create a small select window to catch the extra faces, its not necessary to catch them all in one attempt, see Figure 3.5

Catch a few of the faces and type **f** to remove them from the set, keep moving the window around and hitting **f** till only the inner faces on the cylinder shape are left, you won't can see them till you rotate the model later, but you get the idea by looking at Figure 3.6. Zoom, Rotate and Pan are enabled during the removal process to get closer to the faces in question if necessary.

Once the outer faces have been removed the model needs to be cleaned up a bit more. There are two horizontal and two vertical lines dividing the hole into four quadrants, we need to un-select all faces above the two horizontal lines.

```
qrem load
```

No need to create a select window, hover over the face in question and type **f** to remove, again Pan, Zoom and Rotate are allowed so clean up both sides in a single command. The final result should look like Figure 3.6

Make sure both side of the hole are cleaned up and there are no extra faces in the set.

```
rot -z
```

to see one side

```
rot z
```

to see the other side

3.4 Lower Center of Hole (Set=N1)

zoom in to the center of the lower end of the hole

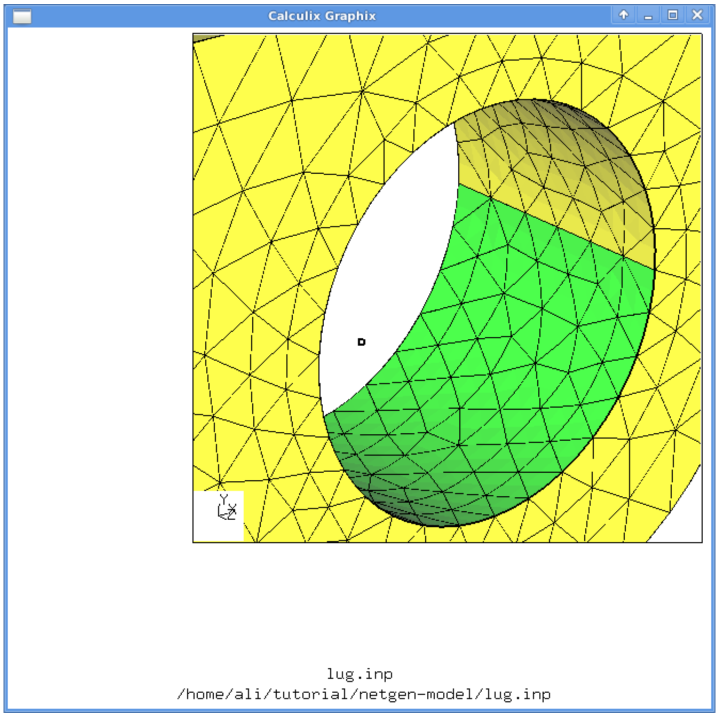

Figure 3.6:

```
plus n all
qenq
```

Hover over a node that looks like its right in the center and type **n** to identify it, if you have trouble selecting a node turn on window select but keep the window small enough to include only one node at a time. In this case it happens to be node 491

```
491 v= 0.000000e+00 xyz= 100.170840 9.996778 9.152117 axyz= 42.474311 84.779660 5.699093
rxyz= 13.553480 100.588063 100.668430
```

Note the identifier, this is the node for the fourth set. Type:

```
prnt se
prnt se
1      all stat:o n:7691 e:4637 f:1400 p:0 l:0 s:0 b:0 L:0 S:0 se:0
2      PART1 stat:c n:0 e:4637 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
3      +C3D10 stat:c n:0 e:4637 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
4      -NJBY stat:c n:0 e:0 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
5      fix1 stat:c n:83 e:0 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
6      load stat:c n:0 e:0 f:206 p:0 l:0 s:0 b:0 L:0 S:0 se:0
```

Note that set **fix1** only has nodes (n:83) and set **load** only has faces (f:206). If we wanted to include the nodes associated with those faces, say to apply concentrated loads (*CLOAD) instead of pressure loads (*DLOAD) all we have to do is
type: comp load do[5] [4]

```
send fix1 abq nam
```

[4] do is short for down and that would include the nodes, we don't need this data but I will demonstrate the use, after we save sets to output files.

Wait for the ready prompt, this will write an ABAQUS format file called *fix1.nam* with just the names of the nodes in set "fix1", if you leave out the *nam* option you get a file with names and coordinates, we don't want that.

```
send load abq press 50.0
```

Wait for the read prompt, this will write an ABAQUS format file called *load.dlo* with the faces in set "load" and a pressure of magnitude 50.0.

Before you exit the **cgx** session, open another terminal window and verify that these files were created.

You should now have a total of three files:

lug.inp contains the complete model definition in terms of nodes and elements

fix1.nam contains the nodes that can be fixed in whatever direction we wish

load.dlo contains the faces and the pressure magnitude [5]

We are now ready to build our complete analysis files for **ccx** to crunch.

One last thing, remember the comp <set> do command, here is the output before and after:

```
6 load stat:c n:0 e:0 f:206 p:0 l:0 s:0 b:0 L:0 S:0 se:0

comp set do ...

6 load stat:c n:455 e:0 f:206 p:0 l:0 s:0 b:0 L:0 S:0 se:0
```

Notice the 455 nodes added to set number six, we can write these out and apply *CLOADS if we wanted to.

[5] we can always modify the pressure magnitude by doing a search and replace in our text editor

Chapter 4

Preparing for Analysis with CCX

In this section we will use the files created previously and build an complete analysis job for each of our five test cases. I will only provide detailed instructions for the first case, the remaining cases are very similar and only the highlights will be discussed where necessary.

4.1 Linear Static Analysis

Begin by creating a new directory, lets call it *case1*. Copy the three files *lug.inp*, *fix1.nam*, *load.dlo* into this new directory. From the *templates* folder copy the template *case1-template.inp*

From a shell prompt in this new directory, type:

```
scite *
```
[1]

This will launch **SciTE** with all four files open so you can cut and paste between them. Figure 4.1

The template file *case1-template.inp* will become the actual **ccx** job for static analysis, the remaining files can be deleted once their contents have been pasted into this file in the appropriate sections. **SciTE** uses a TABBED interface [2]

There is no set process to this, but I will start with the smaller files first. Let us paste in the applied pressure loads. Locate the section in the template file with the lines:

```
*DLOAD
** Pressure based on load (50 pa)
```
[3]

Now switch over to the file *load.dlo* by clicking on its TAB

delete the first line which is a comment

Type Ctrl-A to select all, Ctrl-X to cut [4]

Switch back to the file *case1-template.inp* and paste the contents directly below the comment line.

Remove any blank lines up to the next keyword (*NODE):

```
*NODE FILE
```

Save your work in *case1-template.inp* with a quick Ctrl-S

Switch to the file *fix1.nam* by clicking on its TAB

Delete the first two lines from the file, they are already in the template file.

Select everything else, Ctrl-A , then Ctrl-X

Switch back to *case1-template.inp*

click right below the lines:

[1] or you can type the names of all four files if you prefer

[2] like FireFox, you can switch between files by clicking on the TAB

[3] Comments begin with **, this is a comment

[4] or Ctrl-C to copy

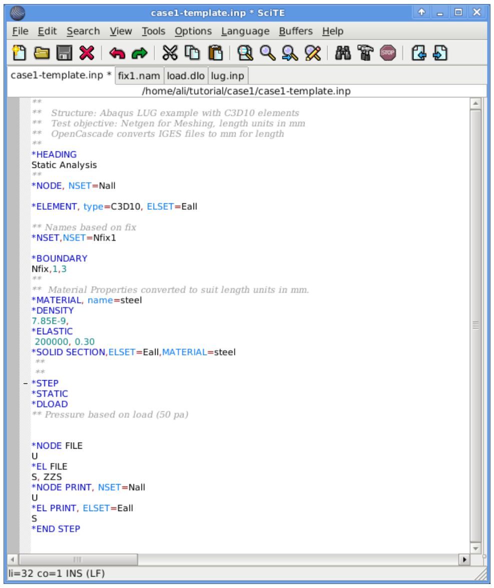

Figure 4.1:

```
** Names based on fix
*NSET,NSET=Nfix1
```

and type Ctrl-V to paste the set “fix1”, delete the blank lines up-to the lines :

```
** Location Bottom of hole
*NSET,NSET=N1
```

So far we have defined the nodes where the lug is fixed in all three directions [5] and the faces where the pressure loads are applied, and the pressure magnitude.

Next we'll define sets N1 and N2 that consist of a single node each. These sets are not needed in the Static Analysis portion but they will be in Dynamic Analysis so I will cover them here and not later. These are nodes we listed earlier with the **qenq** command. Make the following changes depending on what those node numbers were:

```
** Location Bottom of hole
*NSET,NSET=N1
INPUT NODE NUMBER HERE
** Location Attach point
*NSET,NSET=N2
INPUT NODE NUMBER HERE
```

Should read as below:

```
** Location Bottom of hole
*NSET,NSET=N1
491
** Location Attach point
*NSET,NSET=N2
5
```

Save your work in *case1-template.inp*, you may also save and close *fix1.nam* and *load.dlo*, they will be empty if you used Ctrl-X and can be deleted from the system. There are copies elsewhere.

Scroll to the very top of the file *case1-template.inp*, we will now paste in all the nodes and elements to define the complete model.

[5] see the *BOUNDARY card a few lines down in the file

Switch to the file *lug.inp* and remove the top three lines including the keyword *NODE
Select all lines until but not including the line with the keyword *ELEMENT.[6]
Cut and paste the selected node entries into *case1-template.inp* right below the keyword line

```
*NODE, NSET=Nall
```

All that is left in *lug.inp* now is the element definitions, remove the first line (*ELEMENT ...), its already in the template, select and cut everything and paste it *into case1-template.inp* right below the line with the keyword:

```
*ELEMENT, type=C3D10, ELSET=Eall
```

We now have a complete file to submit to **ccx** for a static analysis run. Before we submit it we can open it in **cgx** and make sure everything looks good.

Let us rename it so it is no longer called a template. I'll just call it *case1.inp*

```
cgx -c case1.inp
```

rotate the model so you can see the restrained edge and the bottom of the hole

```
prnt se
1 all stat:o n:7691 e:4637 f:1400 p:0 l:0 s:0 b:0 L:0 S:0 se:0
2 Nall stat:c n:7691 e:0 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
3 Eall stat:c n:0 e:4637 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
4 +C3D10 stat:c n:0 e:4637 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
5 Nfix1 stat:c n:83 e:0 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
6 N1 stat:c n:1 e:0 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
7 N2 stat:c n:1 e:0 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
```

You may reference sets by number instead of set names:

```
plus n 5 g
```

plots the restrained nodes in green

No need to save anything in **cgx**. Use the menu or command to quit.

4.1.1 Submitting the Analysis Job

Is as simple as typing:

```
ccx case1 [7]
```

If there are no errors in the input file you will see a Job Finished message:

```
STEP           1
Static analysis was selected
Decascading the MPC's
Determining the structure of the matrix:
number of equations
22824
number of nonzero matrix elements
874665
Factoring the system of equations using spooles
Using up to 1 cpu(s) for spooles.  Estimating the stress errors
Job finished
```

[6] The easiest way to do this in a large model (over 30,000 nodes) is to do Ctrl-F (find keyword ELEMENT), place the cursor at the end of the last node listing and then select everything to the first line of the file.

[7] do not type the .inp portion of the file name

4.1.2 Getting the most out of your CPUs - Multi Processor Support.

If you are using a copy of **CalculiX** that was complied with multi-processor support, like the Debian binary package[8], and would like to use multi-processor support you can set the variable **CCX_NPROC** to the number of CPUs (cores) you'd like to dedicate to the run.

For bash or sh type:

```
export CCX_NPROC=2
```

...in the shell window that is used to submit the job:

```
STEP 1
Static analysis was selected
Decascading the MPC's
Determining the structure of the matrix:
number of equations
22824
number of nonzero matrix elements
874665
Factoring the system of equations using spooles
Using up to 2 cpu(s) for spooles.
Estimating the stress errors
Job finished
```

The file *spooles.out* in the run directory will list how many threads were used for the job.

If you do not wish to complete the remaining Dynamic Analysis cases you can skip to Chapter 5 for Post-Processing with **cgx**.

4.2 Modal Dynamic Analysis

To determine the response of the lug to a dynamic load [8] we will first determine its mode shapes and natural frequencies. We will begin by computing the first 10 natural frequency values, see the *FREQUENCY input card[4], these natural frequencies will later be used to perform the dynamic analysis, *MODAL DYNAMIC input card[4].

If you look at the supplied template file you'll note that there are two *STEP cards, the first one computes the frequency response, we are requesting the first 10 values beginning with the lowest value of 0.01 cycles/time.

```
*STEP
*FREQUENCY,SOLVER=ARPACK,STORAGE=YES
10,0.01
...
```

Next we will use this data to perform the dynamic analysis, using the *MODAL DYNAMIC card.

```
*AMPLITUDE,NAME=A1
0.,1.,5.E-4,1.
...
...
...
*STEP
*MODAL DYNAMIC
1.E-5,1.E-3
*DLOAD,AMPLITUDE=A1
```

[8] Impact or Sudden Loading

Pressure loads [9] are applied as a function of time, *AMPLITUDE card[4] and the time period for the second *STEP card is 0.005 sec starting at an initial time increment of 1.E-5.

Begin by creating a new directory for this analysis, I'll call it *case2*. Copy the file *case1.inp* from the static analysis job into this directory. Also copy the file *case2-template.inp* into this new directory. Open both files in **SciTE**

As explained previously, the objective is to populate the template with the necessary Node and Element data. I will not go into the details, but you should be able to cut the necessary sections out of *case1.inp* into the template file if you have successfully completed Section 4.1 of this tutorial.

The only portions that differ are the two *STEP definitions.

Output requested is the Displacement value for the Node at the bottom of the hole, this will allow us to plot the Time- History Plots.

To avoid confusion [10] and for the sake of consistent book-keeping, delete the file *case1.inp* after its contents have been placed in *case2-template.inp* and rename it to *case2.inp*, since its no longer just a template..

We are now ready to run the job.

```
ccx case2
```

The analysis will run for some time [11] and you should see the status in the terminal window as the time increments starting at 1.E-5 and running up to 1. E-3. And finally the Message:

```
Job finished
```

4.3 Modal Dynamic Analysis - Steady State

Will will now evaluate the response of the lug using the STEADY STATE parameter[4], from the **ccx** documentation:

> The parameter STEADY STATE can be used to continue a modal dynamics calculation until steady state has been reached. In that case the total time period is set to 10.^10 and does not have to be specified by the user.

Note that the second line parameters are different when STEADY STATE is specified, in this case the second field is used to represent the Relative error for steady state conditions to be satisfied. In the template file this is specified as 0.1 %[4]

In addition we will introduce damping in the structure using the *MODAL DAMPING card. The template for this job, namely *case3-template.inp* is very similar to the previous case except for these two changes.

Copy the template and the file *case2.inp* from the previous job to a new directory and prepare the new input file. Run the job:

```
ccx case3
```

4.4 Dynamic Analysis - Explicit Procedure

Direct Integration Dynamic Analysis using the *DYNAMIC card[4]differs from the previous two cases in the sense that there is no need to compute the frequency response, hence only a single *STEP card is required.

Once again the template is supplied as *case4-template.inp* copy this to a new directory, *case4*, and prepare the input file and run the analysis.

[9] *DLOAD card
[10] we are dealing with five cases here, and several files
[11] depending on you CPU

4.5 Dynamic Analysis - Implicit Procedure

This identical to the previous case except that we will use the IMPLICIT procedure instead of EXPLICIT.

Use the template *case5-template.inp* to perform the analysis.

Note that there are 500 increments and the output file will be over 500 MB and requires a considerable amount of free memory to load in **cgx** for post-processing. [12]

[12] May not load in **cgx** on Windows

Chapter 5

Results, Time-History Plots and etc.

5.1 A Review of the ABAQUS Results.

As mentioned previously, this example is taken from the ABAQUS Documentation[6], Chapter 4. Lets quickly review the results presented there so we can compare our good work. In the ABAQUS example, using C3D20R elements and a fairly course mesh the results obtained are : Mises Stress 330 MPa at the rear welded end, and a maximum displacement of 0.3 mm at the bottom of the hole.

The table below shows the ABAQUS static analysis results as a function mesh density variation[1].

Mesh	No. of Elements	Displacement at Bottom of Hole	Stress at Bottom of Hole	Stress at Attachment
Course	14	3.07E-4	256 E6	312 E6
Normal	112	3.13E-4	311 E6	365 E6
Fine	448	3.14E-4	332 E6	426 E6
Very Fine	1792	3.15E-4	345 E6	496 E6

In ABAQUS only the results of the Explicit Dynamic Analysis are presented as Time Vs Energy Plots, which we do not have access to in **CalculiX** as far as I know. We are however able to plot Time Vs Displacement or Time Vs Stress Plots. An important aspect of performing Dynamic Analysis is that it shows how this particular type of loading causes the lug to oscillate. The history of oscillation will be plotted and we will note that its value compares to that obtained in ABAQUS, 0.6 mm maximum displacement at the bottom of the hole.

Note that the Finest Mesh used only has 1792 Elements wheres our **NETGEN** Mesh has 4637 Elements [2], the ABAQUS Mises Stress Vs Time Plots for Explicit Dynamic Analysis show that the peak Stress at the fixed end is off the order of 550 MPa.

5.2 Using STEPS and SCAL commands during Post-Processing

I have found that the easiest way to locate point of interest [3] in a large assembly while post-processing is to initially limit the number of colors using during a plot. This allows one to quickly locate the area of highest stress or max. displacement.

steps command controls the number of colors used in a plot. The default value is 21 colors

scal command allows one to control the deformed shape of the plot[4]

I will demonstrate the use of both commands shortly.

[1] The data is taken from the ABAQUS Documentation, pages 4-50 and 4-52

[2] C3D10 as opposed to C3D20R

[3] Max Stress or Displacement

[4] e.g scal d 10 will scale the deformed shape by 10X (note its compounded, so you should reset it to 1 before using higher values)

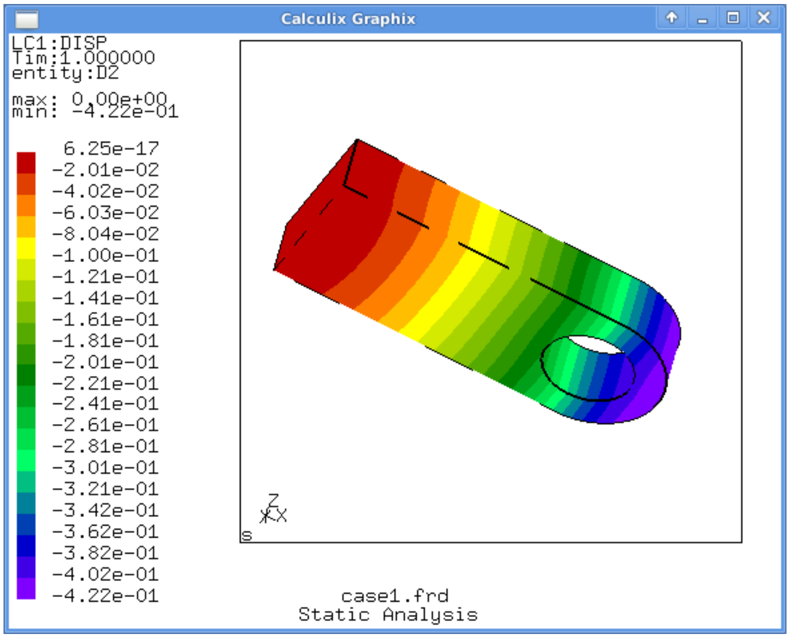

Figure 5.1:

5.3 Results: Linear Static Analysis

Note that the ABAQUS results are in units (m,kg,s,K) whereas our results are in units (mm,N,s,K). If the job competed successfully there will be five files files in the directory: [5]

case1.dat results file in plain text

case1.frd output file for post-processing in **cgx**

case1.inp the input file

case1.sta job summary

spooles.out usually empty

type:

```
cgx -v case1.frd case1.inp
```

This will load the results into **cgx** along with the input file which defines the sets. Loading the input file is not necessary in this case but is required when viewing results from the Dynamic Analysis runs. [6]

Let us view the displacement plot in the 2-direction[7]. Figure 5.1

Note that the maximum displacement in the color chart is -4.22E-1mm, however the value in the vicinity of the hole is actually in the range 0.301 mm to 0.342 mm.

Maximum and Minimum values are always printed in the terminal window [8], and can be copied into reports if necessary. Type:

```
steps 5
```

Note that center of the Hole is now represented by a single band of color with a max value of 3.38E-1 mm. As shown in Figure 5.2

To go back to the default 21 color plot, type:

[5]If you did you book-keeping and deleted every thing unnecessary
[6]You will need to reference your sets
[7]Y-direction
[8]make sure its not behind some other window

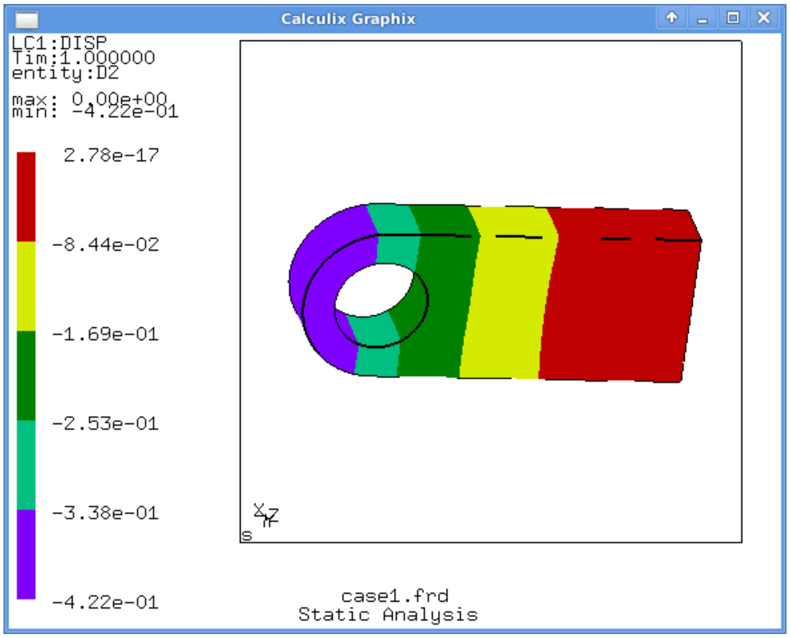

Figure 5.2:

```
steps 21
```

Now let us plot the deformed shape. From the **cgx** menu:
Viewing ▷ Toggle Add-Displacement
Orient the view:

```
rot z
scal d 50
```

This will scale the deformed shape of the plot by 50X, see Figure 5.3. Type:

```
scal d
```

To reset the deformation scale
Plot the Mises Stress and note the values in the terminal window. Figure 5.4

```
Dataset:2 name= STRESS entity:Mises
maxvalue:4.331894e+02 at node:8
minvalue:4.647982e+00 at node:1367
```

The max value is 433 $\frac{N}{mm^2}$ and not something like 4.33 E6 $\frac{N}{m^2}$ as shown in the table in Section 5.1 this is because we switched units, however we can still make comparisons to the ABAQUS results.

```
steps 6
```

Notice the stress plot now only uses 6 colors. Rotate the model to locate the yellow band as show in Figure 5.5. The yellow color represents the area including and around the faces where the pressure was applied. Notice how the yellow band corresponds to a range of 290 to 362, now compare that to results in the table at the beginning of this chapter.

```
prnt se
```

Notice that sets N1 and N2 are listed [9]

[9] As numbers 6 and 7, we can use names or numbers

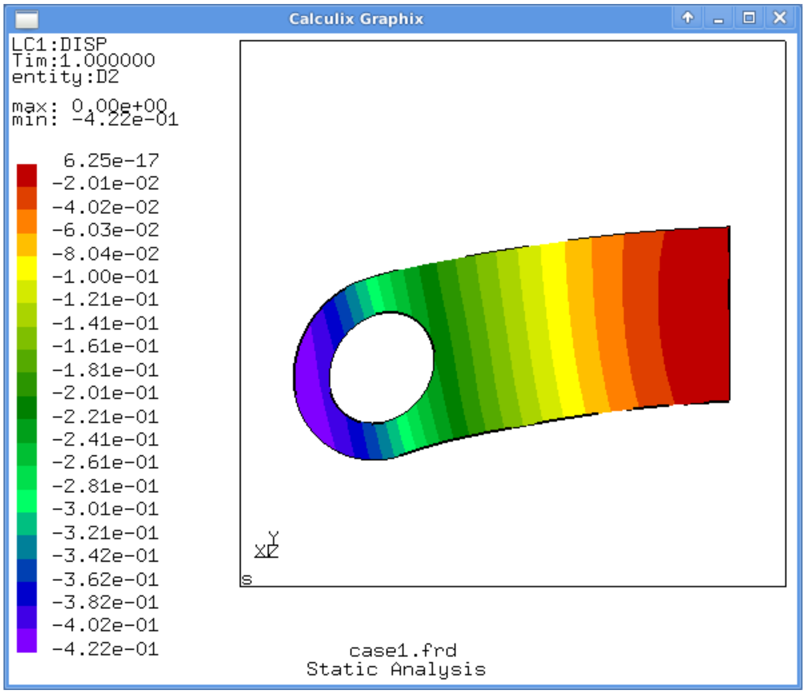

Figure 5.3:

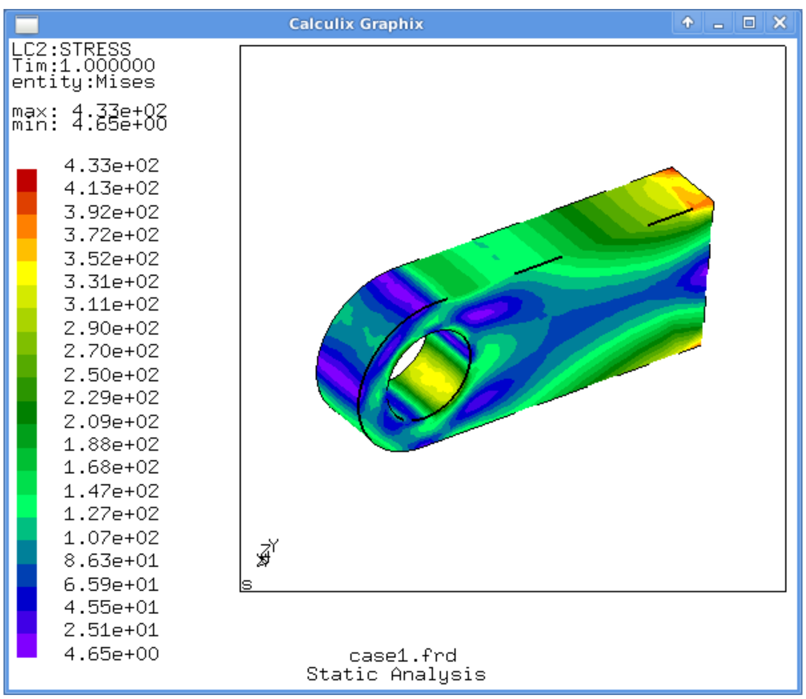

Figure 5.4:

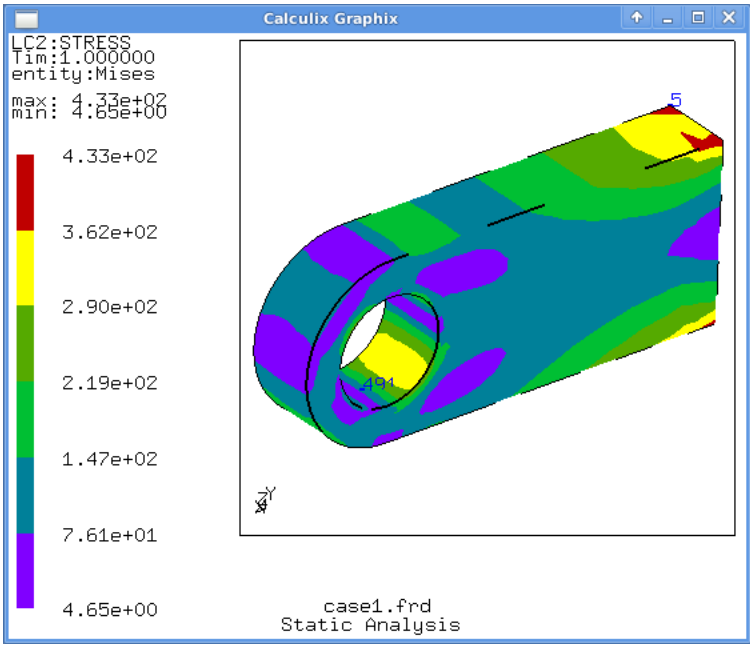

Figure 5.5:

```
6 N1 stat:c n:1 e:0 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
7 N2 stat:c n:1 e:0 f:0 p:0 l:0 s:0 b:0 L:0 S:0 se:0
```

Type:

```
plus na 6 b
```

followed by ...

```
plus na 7 b
```

you see the nodes and there number appear on the screen in blue [10]

Let us query the Stress magnitude at the welded point as shown in Figure 5.6, zoom in to the corner where Node 5 is and type: [11]

```
qenq
qenq
5 v= 4.227494e+02 xyz= 0.000000 50.000000 20.000000 axyz= 21.801409 0.000000 90.000000
rxyz= 53.851648 20.000000 50.000000
in set=Nall(2),Nfix1(5),N2(7),+bou(8),+bou1(9),+bou2(10),+bou3(11),
```

```
steps 21
frame
```

this will reset the plot to 21 colors and zoom to show the full model.

You may also want to plot the **Zienkiewicz-Zhu improved stress (key=ZZS)** and compare the results to the ones in 5.6.[12].

5.4 Results: Modal Dynamic Analysis

If you completed the job and it ran successfully there will be six files in the directory:

[10]You might have to rotate the model to make them visible
[11]Remember to select the node and type n, and q to quit when done
[12][4]Read the CCX Docs on **Zienkiewicz-Zhu error estimator**

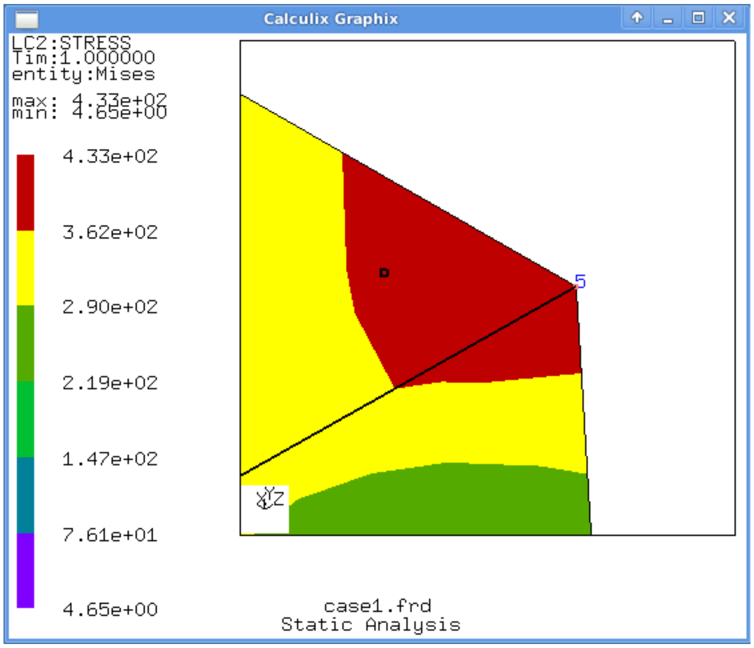

Figure 5.6:

case2.dat Contains the requested outputs in plain text

case2.eig Results file for the first *STEP, not human readable

case2.frd Results data for post-processing

case2.inp The input file

case2.sta The run summary, you can scroll to the end and see that there are 100 solution steps

spooles.out Solver details

Begin by opening the *case2.dat* file in **SciTE**.

We see that our Mode and Frequency values are printed at the top, along with the requested displacement output for set N1 in x,y,z coordinates.

Let us have some fun with post-processing:

```
cgx -v case2.frd case2.inp
```

You can ignore the warnings in the terminal window, we are only interested in reading the set definitions from the file *case2.inp*.

If you open the **cgx** menu and scroll over to "Datasets" and the last "More" entry you'll note that there are 220 result sets, far too many to make selections using the mouse and menu approach.

Since our first *STEP card computes 10 natural frequencies and we requested Node and Element Data (*NODE FILE and *EL FILE) we will skip the first 20 result sets, they are not actually displacement and stress values in the exact sense. To plot the displacement for the first *MODAL DYNAMIC results in the 2-direction (y-dir) we need to plot from dataset 21 type:

```
lc 21 e 2
```

Here lc 21 is the dataset (shown as 21 DISP on the menu) and e 2 is the value in the Y-Dir (shown as 2 D2 on the Entity sub menu). You should see something like Figure 5.7.

```
plot na 6
```

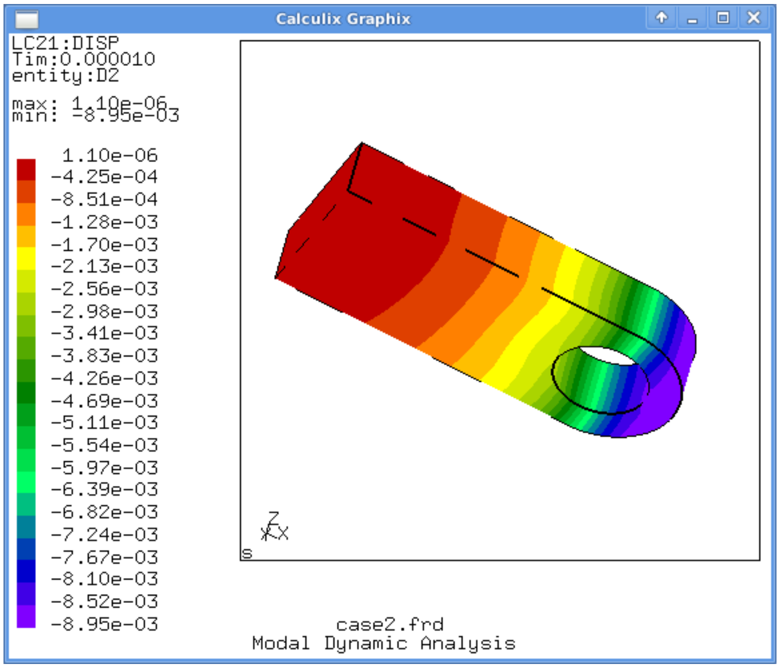

Figure 5.7:

6 being the set number for set N1 [13]

Use **qenq** and select the node to query its displacement in the 2-direction:

```
qenq
491 v= -5.685910e-03 xyz= 100.170840 9.996778 9.152117 axyz= 42.474311 84.779660 5.699093
rxyz= 13.553480 100.588063 100.668430 in set=Nall(2),N1(6),
```

You may also open the file *case2.dat* and verify that the value listed there matches what was printed in the terminal window. Compare this to the table at the beginning of this chapter.

Similarly we can plot the Mises Stress (entity 7) from the last (220) data set as shown in Figure 5.8. Type:

```
lc 220 e 7
```

This is all nice but we don't really have a clear idea as to what happens to the lug when subjected to the sudden loading. We can plot all 220 data sets sequentially and still not understand what really happens. This is where Time-History Plots come into the picture. Let us now observe the response of the bottom of the hole (Node set N1), as a function of Time, when subjected to dynamic loading. [14]

We know the last displacement set is numbered 219 by looking at the **cgx** menu, and the numbering scheme uses odd numbers for displacement sets and even for stress so we will start at set 21[15] and pick alternate sets. We want to plot the response in the 2-Direction (y-dir)

From **cgx** menu select:

Animate ▷ Toggle Dataset Sequence

```
lc 21 23 219 e 2
graph N1 t
```

[13] Hint: prnt se

[14] Can be any point really ! just need a reference

[15] Recall that the first 20 are for frequency data

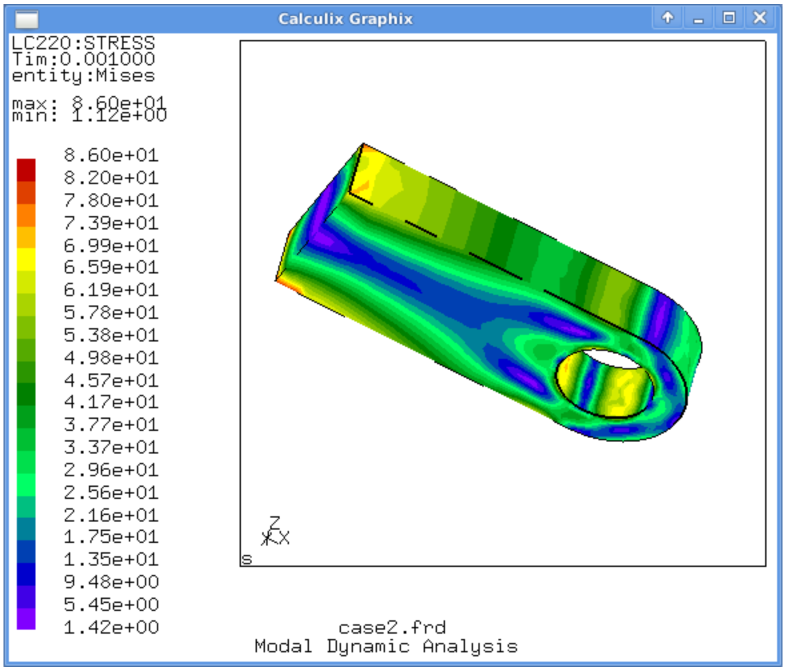

Figure 5.8:

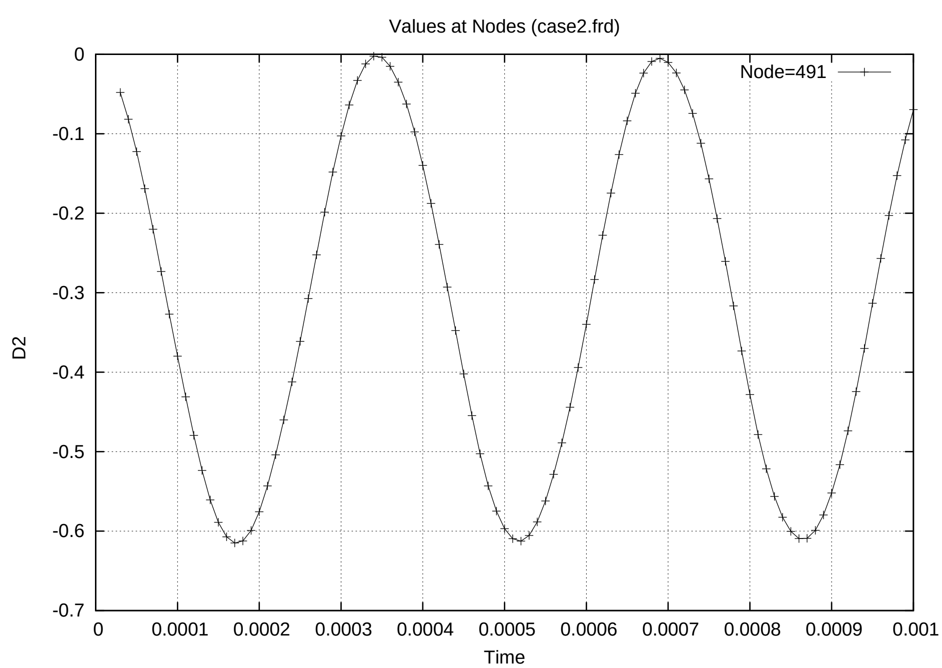

Figure 5.9:

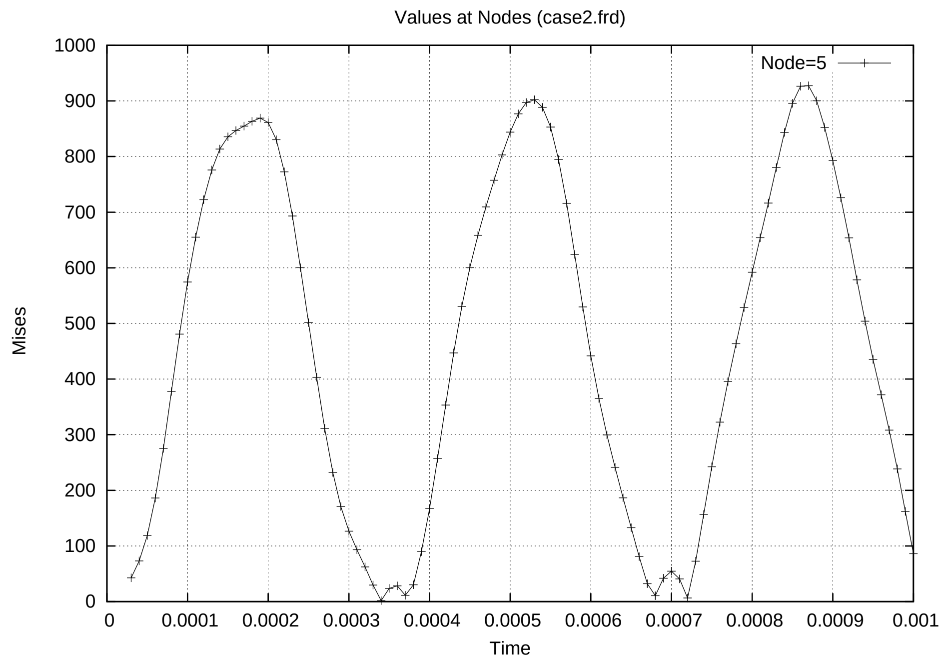

Figure 5.10:

Figure 5.9 shows the Displacement Vs Time plot for set N1. Note that the peak displacement is 0.6 mm which compares well to the ABAQUS results.

Similarly we plot the Mises Stress variation with time at the welded cornet (set N2) as shown in Figure 5.10.

```
lc 22 24 220 e 7
graph N2 t
```

Note that the peak stress is around 900 Pa and not 550 MPa (don't forget the UNITS we used are different) as computed by ABAQUS.

The files *graph_N1_DISP_D2.out* and *graph_N2_STRESS_Mises.out* will list the values used to generate the plots and can be reviewed or used in reports.[16]

5.5 Results: Modal Dynamic Analysis - Steady State

Begin by loading the files into **cgx** for post-processing:

```
cgx -v case2.frd case2.inp
```

Notice that only displacements for the second *STEP card were requested as output. Now let us plot the Displacement Vs Time plot as shown in Figure 5.11.

```
lc 1 2 94 e 2
graph N1 t
```

[16] If gnuplot and gv are not installed (or ghostview, okular, evince, etc.) you will not be able to view the plots automatically while still in **cgx**. The gnuplot files will be generated and the postscript conversion will have to be done manually.

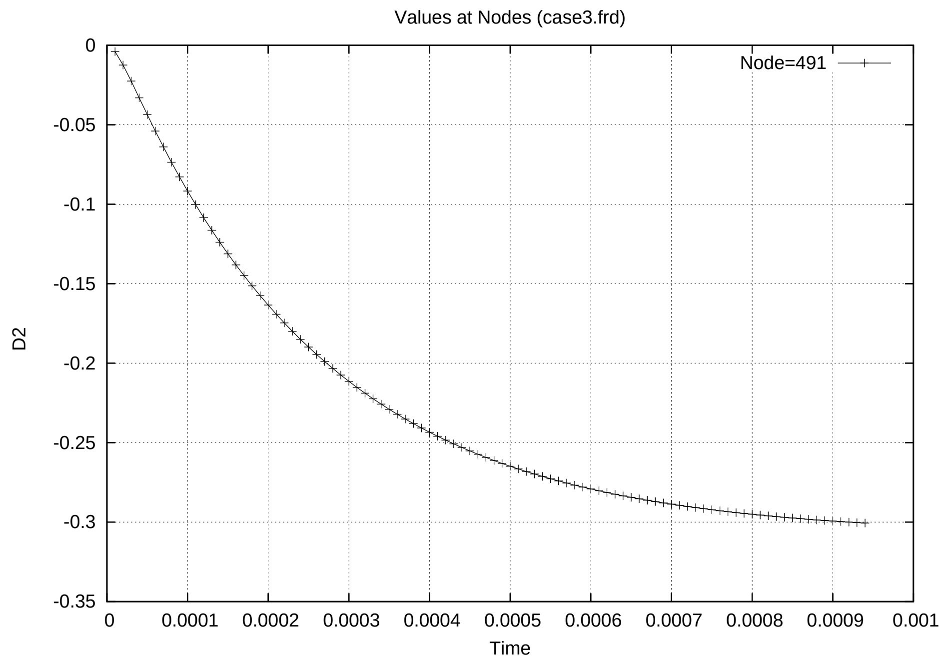

Figure 5.11:

The files *case3.dat* and *graph_N1_DISP_D2.out* list the computed values.

If you would want to plot the Stress as a function of Time you have to edit the input file and request that Stress output be written to files (*EL FILE and *EL PRINT cards).

You may also want to edit the input file, remove the *MODAL DAMPING card and re-run the analysis to see what results you get, note that you will need to increase the step increments as the default of 100 will not complete the job.

5.6 Results: Dynamic Analysis - Explicit Procedure

Not done

5.7 Results: Dynamic Analysis - Implicit Procedure

Load the results into **cgx**, this may take a while its a 500 MB file...

```
cgx -v case4.frd case4.inp
```

The first thing you'll note is that there are 1000 result sets.[17] Plot the the Displacement Vs Time plot as shown in Figure 5.12

lc 1 3 999 e 2
graph N1 t

Lets plot the Mises Stress Vs Time on the welded end as shown in Figure 5.13.

[17] And as the saying goes "A picture is worth a thousand words"

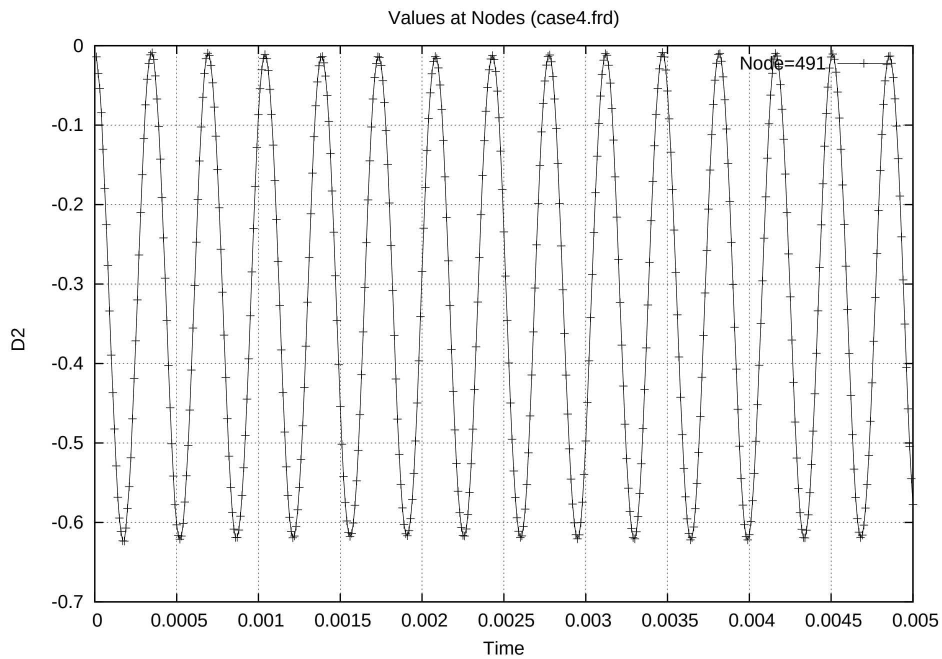

Figure 5.12:

lc 2 4 1000 e 7
graph N2 t

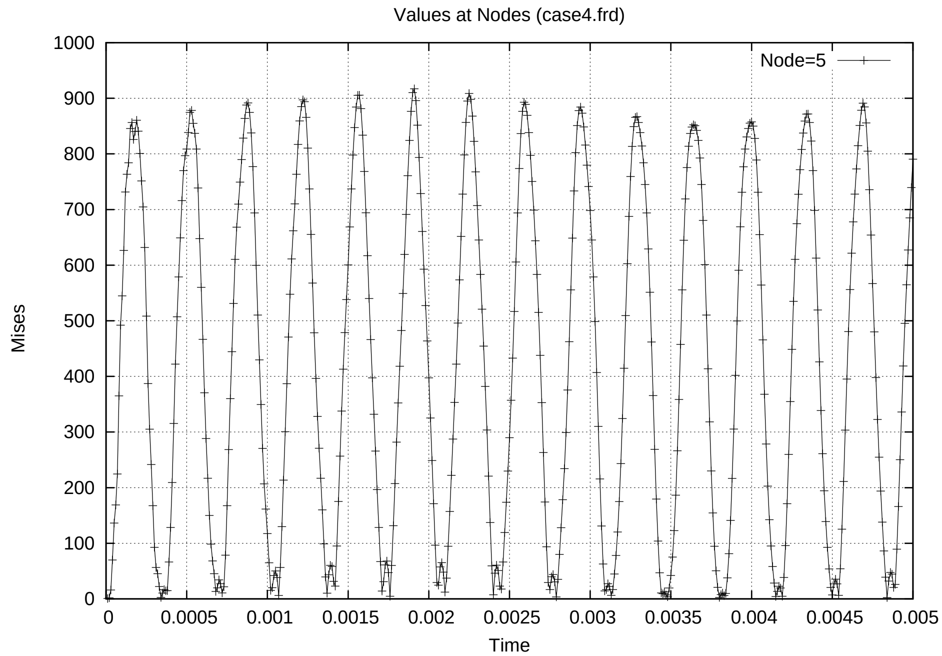

Figure 5.13:

5.8 Summary

I hope this tutorial was helpful as a hands-on guide to using **CalculiX** and **NETGEN** for doing simple Static and Dynamic Analysis and that it will serve as a refresher for those, who like myself, don't use the software on a regular basis and have to search through the official documentation to find answers to concepts that were already understood but quickly forgotten due to lack of exposure to the system.

Bibliography

[1] CalculiX Tutorial on calculix.de
`http://www.dhondt.de/tutorial.txt`

[2] Getting Started Guide on bconverged.com
`http://www.bconverged.com/calculix/index.php`

[3] CAE Linux
`http://caelinux.com/CMS/`

[4] ccx documentation

[5] cgx documentation

[6] Getting Started with ABAQUS Interactive Edition
Version 6.8

[7] Gnuplot
`http://www.gnuplot.info/`

[8] ccx and cgx and NETGEN packages for Debian systems
`http://roe10.physik.hu-berlin.de/CalculiXDebianPackages/binary`

[9] CalculiX Users Group
`http://groups.yahoo.com/group/calculix/`

[10] NETGEN
`http://www.hpfem.jku.at/netgen/`

[11] SciTE Source Code Editor
`http://www.scintilla.org/SciTE.html`

Appendix A

Additional Results:

The data provided here is a summary of results obtained while performing various mesh studies. Maximum Displacement is not actually the displacement at the bottom of the hole, it is simply the maximum value obtained from the color chart used while plotting the displacement plot for the Static Analysis Results.

The Stress at the rear corner was obtained by using the **qenq** command to query the results of the corner node, depending on what parameters were used to specify the output results, this may not exactly match the values in the generated plain text files.

Note that the corner exhibits the true condition of a Stress Singularity, and if more accurate results are desired then some sort of edge and corner fillet should be used while generating the solid model.

Element Type	No. of Elements	Maximum Displacement on Plots	Stress at Fixed Rear Corner
C3D20	352	.45 mm	379 MPa
C3D20	1774	.46 mm	405 MPa
C3D8	2816	.44 mm	363 MPa
C3D10	4931	.42 mm	518 MPa

The model created in Section 2.1 was also analyzed and results are provided here for the sake of completeness. Note that this model has 608 C3D20 Elements and **uses SI Units**:

```
Dataset:1
name= DISP entity:D2
maxvalue:0.000000e+00 at node:2614
minvalue:-4.505330e-04 at node:343
Dataset:2 name= STRESS   entity:Mises
maxvalue:5.222015e+08 at node:1043
minvalue:5.309704e+06 at node:3261
```

The Displacement at the bottom of the hole is represented by set N1 (node 136)

```
qenq
136 v= -3.333030e-04 xyz= 0.100000 0.010000 0.010000 axyz= 45.000000 84.289407 5.710593
rxyz= 0.014142 0.100499 0.100499 in set=Nall(2),N1(6),
```

Mises Stress at the rear corner node can similarly be obtained by querying set N2 (node 3240)

```
qenq
3240 v= 3.168858e+08 xyz= 0.000000 0.050000 0.000000 axyz= 0.000000 nan 90.000000
rxyz= 0.050000 0.000000 0.050000
```