

ANSYS Short Course

Tim Langlais
langlais@me.umn.edu

August 16, 1999

Contents

1	Introduction	1
1.1	Starting ANSYS	2
1.2	Getting Help	2
2	Examples	2
2.1	2D Structural Example	3
2.2	2D Thermal Example	5
3	Batch Processing	7
3.1	Using Parameters	7
3.2	Example Batch File	8
3.3	Checking Line/Area/Volume/Node Numbers	9
3.4	ANSYS Batch Language	9
3.5	Useful Shell Script	10
4	Modeling	10
4.1	3D Exercise	10
4.2	Importing IGES Files	11
5	Meshing	11
5.1	Free Meshing	11
5.2	Mapped Meshing	12
6	Solution	15
6.1	Solvers	15
7	Post Processing	15
7.1	Element Table	15
7.2	X-Y Plots	16
7.3	Printing	16
8	Tips, Tricks, and Other Random Comments	16
8.1	ANSYS Files	16
8.2	Memory Allocation	17
8.3	Disk Space and Network Traffic	17
8.4	Running ANSYS without Wasting Resources	17
8.5	Optimization	17

1 Introduction

ANSYS is a commercial finite-element analysis software with the capability to analyze a wide range of different problems. ANSYS runs under a variety of environments, including IRIX, Solaris, and Windows NT. Like any finite-element software, ANSYS solves governing differential equations by breaking the problem into small elements. The governing equations of elasticity, fluid flow, heat transfer, and electro-magnetism can all be solved by the finite-element method in ANSYS. ANSYS can solve transient problems as well as nonlinear problems. This document will focus on the basics of ANSYS using primarily structural examples.

ANSYS is available on all MEnet Sun and SGI machines. It is available on the Linux machines by remote-login only. On the bright side, rumor has it that ANSYS is looking into a Linux port. Currently, MEnet uses the Research/Faculty version of ANSYS 5.4. The Research/Faculty license level permits larger, more complex models than does the current level running on the IT Labs machines.

This document is meant to be a starting point. The material covered here is by no means comprehensive. In fact, we will only scratch the surface of ANSYS's capabilities. Given that, I will try to cover most of what I know about ANSYS and some tricks I have learned while using it. The document will begin with two simple examples, taking the user through all of the steps of creating a model, meshing, adding boundary conditions, solving, and, finally, looking at the results. The remainder of this document will offer tips and tricks for each of the steps.

You should use this document in conjunction with "ansys-course.tar.gz," an archive file that contains this document and all of the examples used in this document in batch file format. From MEnet machines,

```
unix% cp ~langlais/ansys-course.tar.gz .
unix% gunzip ansys-course.tar.gz
unix% tar -xvf ansys-course.tar
```

This will create a directory called "ansys" with several sub-directories.

1.1 Starting ANSYS

To start interactive ANSYS

```
unix% module add ansys
unix% ansys54 -g -p ansysrfr
```

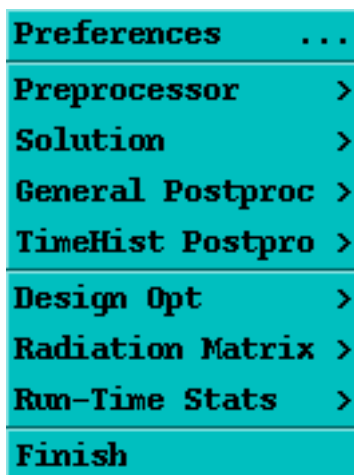
and away you go... Alternately, you can use the ANSYS launcher

```
unix% module add ansys
unix% xansys54
```

Click on **Interactive ...**, which will bring up a menu of startup options. Click on **Run** and away you go...

Note that **-g** is an option telling ANSYS to start the GUI (Graphical User Interface) and **-p ansysrfr** tells ANSYS which license code to use.

Once you have hit **Enter**, you should see 6 new windows on your screen: a **Utility Menu** at the top, an **Input** and **Toolbar** menu below that, and finally a **Main Menu** and **Graphics** window. The majority of commands can be gotten to via the **Utility Menu** or the **Main Menu**. I will focus on the **Main Menu** here.



All commands are assumed to start from the **Main Menu** unless otherwise specified.

1.2 Getting Help

ANSYS has excellent on-line help available through the **Utility Menu** under **Help**. There are six basic ANSYS manuals: 1) the **Theory Manual**, 2) the **Analysis Guides**, 3) the **Commands Manual**, 4) the **Elements Manual**, 5) the **Operations Guide**, and 6) the **Workbook**.

As suggested by the name, the **Theory Manual** discusses the underlying theory of finite-elements. The manual also covers the underlying equations being solved by ANSYS for each type of problem. The **Theory Manual** is a good place to begin to

familiarize yourself with the mathematics of what ANSYS does for each class problems.

The **Analysis Guides** are perhaps the most useful of the manuals. These guides explain how to use ANSYS to model problems and cover all major aspects of using ANSYS.

The **Commands Manual** is an exhaustive reference of all ANSYS commands. The commands are referenced according to batch commands, *not* GUI commands (one good reason to be familiar with batch processing).

The **Elements Manual** covers the details of all the elements available in ANSYS—the nodes, the variables, any constants, etc.

The **Operations Guide** offers basic information about how to run ANSYS. This guide would be a good starting point for a new user.

The **Workbook** contains several examples with step by step instructions. The examples include sample structural, dynamic, and thermal problems. These examples are the same as those contained in the ANSYS workbook available at the bookstore for more than \$50.

Within the **ANSYS Help** you can use the search utility to find instances of certain keywords. You can use the table of contents to page through or you can use the indices to find certain sections. You are encouraged to make full use of the help. That is how this author learned to use ANSYS.

Besides the **ANSYS Help**, there are other online resources. ANSYS, of course, maintains a site: <http://www.ansys.com/>. Texas A&M has converted the 5.5 manuals to HTML and posted them on the web. They are available at <http://terminator.tamu.edu/softwareDocs/ansys/realtoc.html>. While no newsgroup is devoted entirely to ANSYS, ANSYS related discussions often appear on the `sci.engr.mech` and `sci.engr.analysis` newsgroups. Finally, Karl Geisler has written a tutorial for ME5345, Heat Transfer in Electronic Equipment: <http://www.me.umn.edu/courses/me5345/ansys.html>.

2 Examples

Regardless of the type of problem involved, an ANSYS analysis consists of the same steps: modeling, meshing, solution, and post processing.

The modeling phase entails geometry definition. This is where you draw a 2D or 3D representation of the problem.

During the meshing phase you will define material properties and choose a finite element suitable for the problem. The last step of the meshing phase is to discretize the model—i.e. create the mesh.

In the solution phase, boundary conditions and loads need to be defined. The types of loads and boundary conditions you select depend on the simplifications being made. ANSYS will

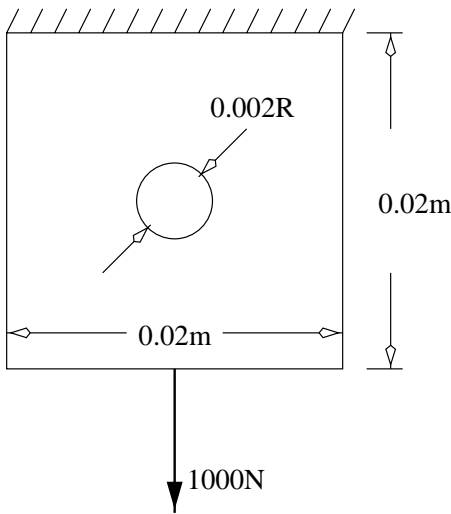
then attempt to solve the system of equations defined by the mesh and boundary conditions.

Finally, when the solution is complete, you will need to review the results using the post processor. These results may be color contour plots, line plots, or simply a list of DOF results for each node.

2.1 2D Structural Example

The easiest way to learn ANSYS is through a simple example. This section will cover the steps needed to analyze a plate with a hole. You are encouraged to test different menu items and to use the help often.

Say we would like to analyze a plate with a hole that is being loaded axially in the plane.



We will assume that the plate is thin-enough to be in a plane stress state, meaning that we can model the plate in 2D.

ANSYS has a very powerful modeler built into the pre-processor. The modeler allows the user to construct surfaces and solids to model a variety of geometries. For any given geometry, there are often several different ways to create the model.

Start by assigning a file name to your work. From the Utility Menu,

```
File
Change Jobname ...
[/FILENAM] Enter new jobname platestr
```

Click OK to accept. The jobname must be 8 characters or fewer.

Start by entering the pre-processor,

```
Preprocessor >
```

To make our drawing easier, we will use the ANSYS workplane, which is simply a 2D grid for drawing using the mouse. From the Utility Menu,

```
WorkPlane
WP Settings
Grid and Triad
Snap Incr 0.0005
Spacing 0.001
Minimum -0.015
Maximum 0.015
Tolerance 0.00003
```

To display the workplane

```
WorkPlane
Display Working Plane
```

If your workplane is too small, you will need to zoom in (from the Utility Menu)

```
PlotCtrls
Pan, Zoom, Rotate ...
Box Zoom
```

Use the mouse to click two corners of a box around your workplane grid. Leave the Pan-Zoom-Rotate window off to the side since you will likely need it later on. You can position the workplane using the Δ , ∇ , \triangleleft , and \triangleright buttons. You will find this menu quite useful in navigating models and results.

Now draw the rectangular area centered around (0,0)

```
Preprocessor >
-Modeling-
Create >
-Areas-
Rectangle >
By 2 Corners +
```

Click and hold down on the left mouse button. The workplane coordinates will appear in the popup menu. Position the mouse over $X = -0.01$ and $Y = 0.01$ and let go of the left mouse button. Now position the mouse over $(0.01, -0.01)$ and click again. You should have a rectangular area. Click on OK to complete.

Note that you can return to any of the popup menus spawned by the Main Menu at any time. So if you make a mistake, you can always take one or two steps back. Start anywhere in this sequence to draw the circle,

```
-Modeling-
Create >
-Areas-
Circle >
Solid Circle +
```

Notice that the middle line of the Input window instructs you to pick two workplane locations—a center and a radius. Click on (0,0), then (0.002,0). Click on OK to complete.

Now we'd like to subtract the circular area from the square area,

```
-Modeling-
Operate >
```

```
-Booleans-
  Subtract >
    Areas +
```

Click on the box. This will spawn an error message to let you know that there are two areas. If the square is highlighted, click OK, otherwise, choose **Next** until the square is highlighted then OK. Click OK in the **Subtract** popup menu. Select the center circle (again, this will raise a warning; make certain you have selected the circle). Click on OK in the **Subtract** popup menu and you have a 2D plate with a hole.

Note that each entry in all of the menus spawned from the **Main Menu** are coded: entries with **-Text-** are not selectable, entries with **Text >** will spawn another menu (with additional selections required before an action) and those with **Text ...** or **Text +** will spawn a popup menu.

This would be a good point to save your work. Use the **Toolbar** menu, **SAVE.DB**. This will save all of the pertinent information in an ANSYS file called **platestr.db**.

We need to assign material properties to the model. Only structural properties are needed. From the **Preprocessor** menu,

```
Material Props >
  -Constant-
  Isotropic ...
  Young's Modulus EX 200e9
  Poisson's ratio (minor) NUXY 0.3
```

Now that we have a model, we need to mesh the model. But first we'll need to choose an element type with which to mesh. We will select a planar 8-noded quadrilateral element used for structural analysis.

```
Preprocessor >
  Element Type >
  Add/Edit/Delete ...
  Add ...
  Structural Solid
  Quad 8node 82
```

You will notice in the left window a list of general categories, **Structural Mass**, **Structural Link**, **Structural Solid**, etc. A number of different specific elements will appear in the right window for each general category. Each element has its own set of DOFs, which are the degrees of freedom for which ANSYS will find a solution. See the ANSYS online help for more information on specific elements. Click on OK. The **Element Types Menu** should now show **PLANE82** as element type 1. This element can be used for plane stress, plane strain, and axisymmetric problems. From the **Element Types Menu**

```
Options ...
  Element behavior K3: Plane Stress
  Help
```

Clicking on the **Help** brings up the ANSYS file on the **PLANE82** element, which explains several of the available options. Leave

the **Help** window up for later reference if you like. Click **OK** in the **PLANE82** element type options menu to close the window. Click **Close** to accept the changes you have made.

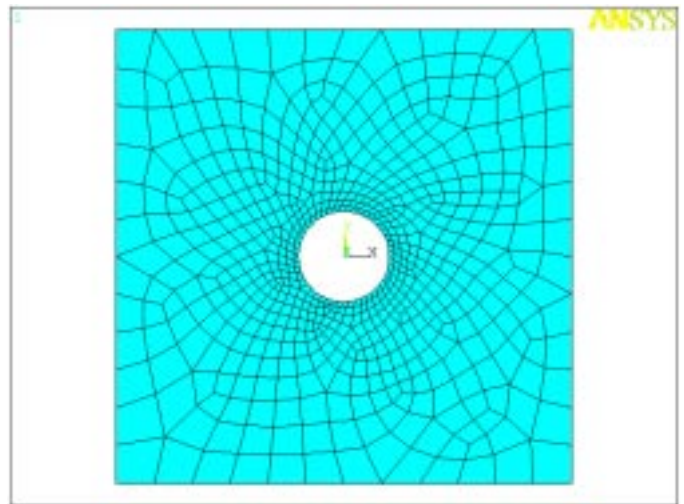
Now we can mesh the model. There are any number of ways to mesh a model, some good, some bad. For now we will use a simple approach:

```
-Meshing-
  Size Cntrl >
  -ManualSize-
  -Global-
  Size ...
  NDIV No. of element divisions 12
```

This specifies the number of element divisions for each line that forms the model. To mesh the model,

```
-Meshing-
  Mesh >
  -Areas-
  Free +
```

Select the plate area and click OK. ANSYS will mesh the model and plot the elements in the **Graphics** window. Your mesh ought to look something like,



To complete the model, we need to add boundary conditions. Return to the **Main Menu**,

```
Solution >
  -Loads-
  Apply >
  -Structural-
  Displacement >
  On Nodes +
```

Select **Box** in the selection window and draw a box around the line defining the “top” of the plate. This should select all of the nodes along that line. The **Apply U,ROT on Nodes** menu will pop up.

```
Lab2 DOFs to be constrained All DOF
```

Return to the Solution menu to apply a load

```
-Loads-
Apply >
-Structural-
Force/Moment >
On Nodes +
```

Again, select Box and select all of the nodes on the “bottom” line of the plate. The number of nodes you selected ought to be listed in the Apply F/M on Nodes menu under Count. Remember that number (I selected 25). Click OK, spawning another menu

```
[F] Apply Force/Moment on Nodes
Lab Direction of force/mom FY
VALUE Force/moment value -1000/25
```

Click OK. This will apply a total load of 1000N to the “bottom” edge of the plate (or 1000/25 per node for 25 nodes). The model is now complete.

Now tell ANSYS to find the solution. From the Solution menu,

```
-Solve-
Current LS
```

This will spawn two new windows. Click OK in the Solve Current Load Step window. This will begin the solution process. ANSYS will alert the user when the solution is done.

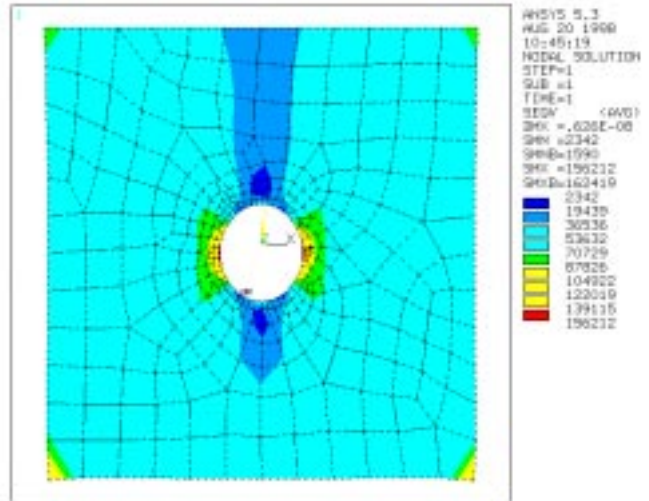
Note that a batch file copy of the above example is located in `ansys/batch/platestr`.

Finally, let’s view the results using the postprocessor,

```
General Postproc >
Plot Results >
-Contour Plot-
Nodal Solu ...
```

In the Contour Nodal Solution Data menu select

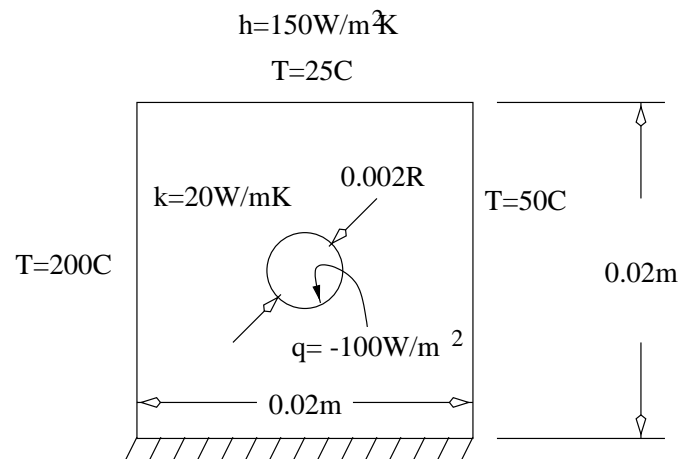
```
Stress
von Mises SEQV
OK
```



You will notice high stress regions on the bottom corners of the plate in the SEQV plot. Since we applied loads directly to the nodes, those loads are considered point loads at each node. This may not reflect reality, especially if the load is distributed evenly over the edge in the real world. Consequently, results close to the point loads are likely to be in error.

2.2 2D Thermal Example

Now let’s try a thermal analysis of the following problem,



First, we need to clear the old structural analysis (you can save at this point if you wish). From the Utility Menu,

```
File
Clear and Start New ...
OK
Yes
```

Don’t forget to rename the job. From the Utility Menu,

```
File
Change Jobname ...
[/FILENAME] Enter new jobname platethr
```

Click OK to accept.

Rather than redraw the plate, input a batch file that does that for you by typing in the Input window,

```
/input,plate
```

You should see a picture of the usual plate with hole. Zoom in if you need to.

We need to assign thermal properties to the model.

```
Preprocessor >
Material Props >
-Constant-
Isotropic ...
Thermal conductivity KXX 20
```

Now we need to pick a thermal element for analysis,

```
Preprocessor >
Element Type >
Add/Edit/Delete ...
Add ...
Thermal Solid
Quad 8node 77
```

Click on OK. Click on Options ... if you would like to see the different options available for this element. We will use the defaults. Close the Element Types window.

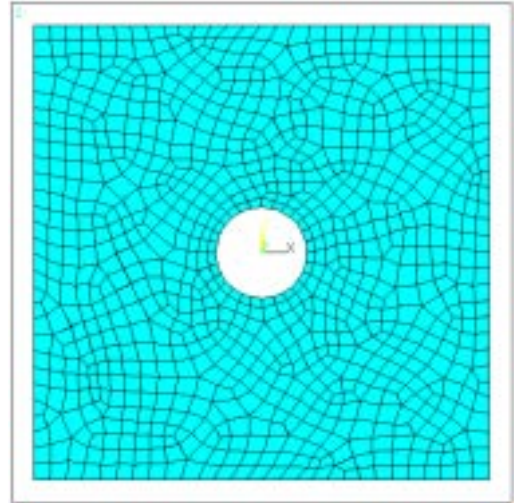
Like the structural analysis, we need to mesh the model. We will use ANSYS's built in Smart mesher. Be cautious using this tool. Since ANSYS does not know what you are solving for or what the boundary conditions will be, it cannot know what the best mesh is.

```
-Meshing-
Size Cntrls >
-SmartSize-
Basic ...
LVL Size Level 3
OK
```

Now mesh the area

```
-Meshing-
Mesh >
-Areas-
Free +
```

The result should be a fairly uniform fine mesh.



Finally, let's add the boundary conditions. Return to the main menu,

```
Solution >
-Loads-
Apply >
-Thermal-
Temperature >
On Nodes +
```

Box the area around the nodes on the farmost right of the plate.

```
VALUE Temperature value 50
OK
```

Repeat the procedure for the left side nodes and enter a temperature of 200. Now apply the convection boundary condition,

```
-Loads-
Apply >
-Thermal-
Convection >
On Nodes +
```

Pick the nodes on the top line, click OK and enter the following,

```
VALI Film Coefficient 150
VAL2I Bulk Temperature 25
OK
```

By default the bottom boundary condition is adiabatic. Finally, enter the boundary condition for the center hole,

```
-Loads-
Apply >
-Thermal-
Heat Flux >
On Lines +
```

Zoom in on the center hole to pick the lines that define the inner edge of the hole.

```
VALI Heat flux value -100
OK
```

Lines and areas are solid model features. You must transfer boundary conditions imposed on these features to the nodes along those features,

```
-Loads-
Operate >
  Surface Loads ...
  OK
```

Now tell ANSYS to find the solution. From the **Solution** menu,

```
-Solve-
Current LS
```

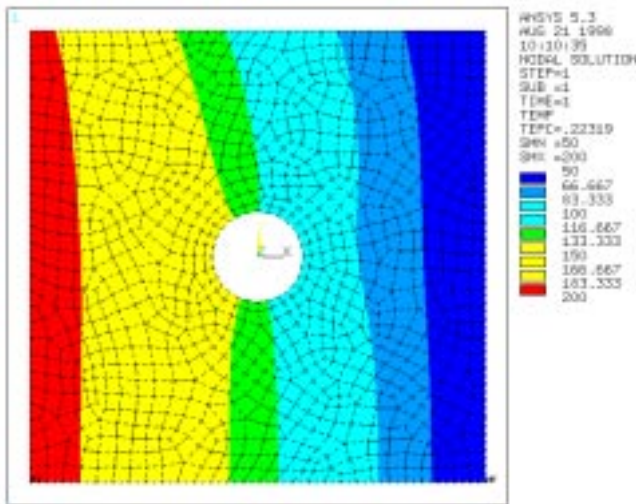
Note that a batch file copy of the above example is located in `ansys/batch/platethr`.

When the solution has finished, you can view the temperature profile from the postprocessor,

```
General Postproc >
Plot Results >
  -Contour Plot-
  Nodal Solu ...
```

In the **Contour Nodal Solution Data** menu select

```
DOF
Temperature
OK
```



3 Batch Processing

There two primary ways to use ANSYS—interactively through the graphical user interface and through the use of batch files and ANSYS commands. Up to this point, we have used the GUI exclusively. It is easiest to learn ANSYS interactively, especially when compared to the daunting task of learning

all of the relevant ANSYS commands. But do not be fooled! Easier does not mean better or faster.

It turns out that solely using interactive ANSYS has several disadvantages:

- Interactive use requires the user to save the model geometry, mesh, and results in a *.db file. The *.db files can get as large as 50MB or more. We all have limited quotas. You will get no sympathy from the systems staff (they have quotas too) so you need to learn to conserve space.
- Interactive use is slow if you need to repeat operations. Mouse clicks are great until you have to do them over and over and over again.
- For long jobs, interactive use ties up a console. If you aren't using the machine for something else while ANSYS is solving, you are wasting resources. Besides, it is against MEnet policy to start a job on the console and leave for more than 15 minutes.

The main disadvantage of batch processing is the steep learning curve. The advantages of batch processing are many:

- An entire model, mesh, and solution description can be contained in a file of 10-100K.
- You can run niced background batch jobs any time. Submit a job, go grab a bite, come back and look at the results.
- Batch processing is highly modular. If you spend time creating batch files, changing dimensions and mesh densities is a snap.
- You can optimize or make several ANSYS runs without having to do everything (changing parameters, dimensions, etc.) by hand.

In short, batch processing saves time!

Batch processing involves interacting with ANSYS through its command structure rather than through the GUI. (Actually, the GUI commands can all be linked to ANSYS batch commands.) It involves learning another computer language.

3.1 Using Parameters

ANSYS has the ability to use and store scalar, vector, and matrix parameters. Scalar parameters come in handy when you are drawing a complex geometry, it being far easier to remember names like **WIDTH** and **LENGTH** rather than 0.10954 and 1.7628. These scalar parameters are also a powerful way to build modular geometries. Let's draw the plate using parameters. (Clear the previous analysis as in the thermal example).

From the **Utility Menu**,

Parameters
Scalar Parameters ...

In the Selection box of the Scalar Parameters enter the parameters

```
LENGTH=0.02  
WIDTH=0.02  
RAD=0.002
```

Now let's draw the same plate with a hole using parameters.

```
Preprocessor >  
-Modeling-  
Create >  
-Areas-  
Rectangle >  
By 2 Corners +
```

Now enter the following into the popup

```
WP X -WIDTH/2  
WP Y -LENGTH/2  
Width WIDTH  
Height LENGTH
```

Click OK to complete. Note how we can also include mathematical operations like $-WIDTH/2$ in any of the fields where parameters are accepted. The same procedure applies for the circular area.

```
-Modeling-  
Create >  
-Areas-  
Circle >  
Solid Circle +
```

```
WP X 0  
WP Y 0  
Radius RAD
```

Click on OK to complete. Finally, subtract the circular area from the square area as before,

```
-Modeling-  
Operate >  
-Booleans-  
Subtract >  
Areas +
```

Note that parameter names are limited to 8 characters. Names beginning with numbers are not allowed, nor are special characters that could otherwise be construed as operators.

Unlike Pro/E and other CAD packages, changes in parameters are not automatically reflected in the geometry. Thus, using parameters from the GUI is not as useful as it could be. The real power of parameters is seen when they are used to define the geometry within a batch file.

3.2 Example Batch File

Here is a batch file that draws the same plate with a hole using a different method. The plate is split into two areas to achieve a certain mapped mesh. You can find a copy of the file in `ansys/batch/platebth`.

```
! define some parameters  
WIDTH=0.02           ! width of the plate  
HEIGHT=0.02         ! height of the plate  
WID_BY2=WIDTH/2.0  
HGHT_BY2=HEIGHT/2.0  
RADIUS=0.002        ! radius of the hole  
  
!  
! This file draws a 2D model of a plate  
! with a hole using keypoints.  
! Lines and areas are created using the  
! keypoints.  
  
! enter the pre-processor  
/prep7  
  
! now create the corners of the plate  
k,1,-0.01,-0.01  
k,2,0,-0.01  
k,3,0.01,-0.01  
k,4,0.01,0.01  
k,5,0,0.01  
k,6,-0.01,0.01  
  
! create the lines which define  
! the plate edges  
l,1,2 ! line #1  
l,2,3 ! line #2  
l,3,4 ! line #3  
l,4,5 ! line #4  
l,5,6 ! line #5  
l,6,1 ! line #6  
  
! create the keypoint for the center of  
! the hole and hole radius  
k,10,0,0  
k,11,0,-0.002  
k,12,0.002,0  
k,13,0,0.002  
k,14,-0.002,0  
  
! create the arcs that define the circle  
larc,11,12,10,RADIUS ! line #7  
larc,12,13,10,RADIUS ! line #8  
larc,13,14,10,RADIUS ! line #9  
larc,14,11,10,RADIUS ! line #10  
  
! draw connecting lines from the  
! circle to the box
```

```

1,11,2 ! line #11
1,13,5 ! line #12

! now create the areas
al,2,3,4,12,8,7,11 ! area #1
al,1,11,10,9,12,5,6 ! area #2

! concatenate some lines before meshing
! for the rhs box
lssel,all,all ! select all lines
lssel,s,,1
lssel,a,,6
lssel,a,,5
lccat,all

! for the lhs box
lssel,all,all
lssel,s,,2,4,1
lccat,all

! for the rhs hole
lssel,all,all
lssel,s,,7,8,1
lccat,all

! for the lhs hole
lssel,all,all
lssel,s,,9,10,1
lccat,all

allssel,all,all

! now select the element type;
! 8-noded structural solid
! assign it as element #1
et,1,plane82

! select mapped (quadrilateral ONLY)
! meshing
eshape,2

! select the number of element
! divisions per line
esize,0.001

! mesh the areas
amesh,1,2

/eof

```

All lines beginning with ! are comment lines; everything after the ! is ignored for that line. The /eof command signals the end of input. If you would like to test just a portion of your batch file, you can do so by placing an /eof anywhere in your

batch file. To test a batch file from the GUI, simply type /input,file in the Input window. Note that ANSYS is very finicky about the filenames you choose—filenames must be fewer than 9 letters. Furthermore, the files must reside in the present working directory.

One quick way to learn ANSYS batch commands is to check the *.log files. Whenever you start a session, ANSYS logs all of the commands issued through the GUI or the Input window to that file. Consequently, if you know how to do something through the GUI, after performing the operation you can check the *.log file to find the command name and learn more about it in the **Commands Manual**. But beware of cutting and pasting directly from the *.log file into your batch file! The ANSYS commands generated by the GUI generally have special arguments to denote graphical picking with the mouse, arguments that are not available during batch processing.

3.3 Checking Line/Area/Volume/Node Numbers

When building a batch file, it is often useful to know how ANSYS numbers the lines, areas, and volumes. To turn numbering on (from the Utility Menu,

```

PlotCtrls
Numbering ...
...
OK

```

You will notice that the numbers are annoyingly small and difficult to read. Zooming in does not increase the font size of the numbers. In the Input window,

```
/dev,text,2,150
```

The last number, 150 in this case, is the percentage increase in the font size.

3.4 ANSYS Batch Language

The ANSYS batch language has many features of the FORTRAN programming language. If statements and do loops can all be included in ANSYS batch files. In addition ANSYS has several built-in functions for further manipulation of ANSYS results or geometry parameters.

Here is a simple example of an if structure. It is quite common for a problem to have several different scenarios. In this case, there are two different loadings denoted by parameters AXIAL and TORQ

```

! If axial loading...
*if,AXIAL,EQ,1,then
! apply the axial force
allsel
nset,r,loc,x,TOTAL_L-SMALLE,TOTAL_L+SMALL_E
*get,NODECNT,node,,count
f,all,fx,FAXIAL/NODECNT

```

```

*endif

! If torque loading
*if,TORQ,EQ,1,then
  ! apply the torsion force
  allsel
  nsel,r,loc,x,TOTAL_L-SMALLE,TOTAL_L+SMALL_E
  nsel,r,loc,y,NOM_R-SMALLE,NOM_R+SMALLE
  f,all,fz,-FTORQ/4

  allsel
  nsel,r,loc,x,TOTAL_L-SMALLE,TOTAL_L+SMALL_E
  nsel,r,loc,y,-NOM_R+SMALLE,-NOM_R-SMALLE
  f,all,fz,FTORQ/4

  allsel
  nsel,r,loc,x,TOTAL_L-SMALLE,TOTAL_L+SMALL_E
  nsel,r,loc,z,NOM_R-SMALLE,NOM_R+SMALLE
  f,all,fy,FTORQ/4

  allsel
  nsel,r,loc,x,TOTAL_L-SMALLE,TOTAL_L+SMALL_E
  nsel,r,loc,z,-NOM_R+SMALLE,-NOM_R-SMALLE
  f,all,fy,-FTORQ/4
*endif

```

The values of TORQ and AXIAL can be set at the ANSYS command line,

```
ansys54 -AXIAL 0 -TORQ 1 ...
```

which sets AXIAL=0 and TORQ=1. Do loops (using the *do command) can be used to map the effect of changing parameters on the results.

3.5 Useful Shell Script

Here is an example of a useful shell script that runs analyses for two load cases, AXIAL and TORQ, packages the model and results (located in the *.db file), and copies an archived version to a home directory. You should read up on shell scripts before attempting to modify and/or use this script.

```

#!/usr/local/bin/tcsh

# set ANSYS paths, etc
module add ansys

# run AXIAL loading
ansys54 -AXIAL 1 -TORQ 0 -p ansysrf -m 128 < \
  tshaft > tshaft-AXIAL.out

# package up results and copy
tar -cvf - tshaft tshaft.db | \
  gzip -9 > tshaft-AXIAL.tar.gz
cp tshaft-AXIAL.tar.gz \

```

```

~langlais/John_Deere/ANSYS/notched-shaft/

# removes everything
\rm -f tshaft.*

# run torque
ansys54 -AXIAL 0 -TORQ 1 -p ansysrf -m 128 < \
  tshaft > tshaft-TORQ.out

# package up results and copy
tar -cvf - tshaft tshaft.db | \
  gzip -9 > tshaft-TORQ.tar.gz
cp tshaft-TORQ.tar.gz \
  ~langlais/John_Deere/ANSYS/notched-shaft/

# removes everything
\rm -f tshaft.*

exit

```

4 Modeling

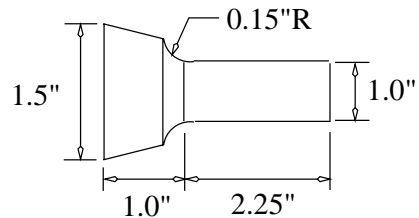
ANSYS uses a very powerful modeler to help the user create 2D and 3D models. You will find that there are any number of methods you can use to create a model.

All of the previous examples have focused on 2D models. The first example used areas and boolean operations to achieve a hole-in-plate model. By contrast, the hole-in-plate created in the section on batch files uses keypoints and lines to define areas.

These two contrasting methods apply to 3D modeling as well where volumes can be made from volumetric primitives like cylinders, spheres, and cones or can be made from extruded and swept lines and areas.

4.1 3D Exercise

As an exercise in 3D modeling, try creating this shaft, shown here in a side view.



The cross-section is circular so you should start by creating a cylinder or revolving an area. Note that the notch root is intended to be a 90° arc, though it isn't shown explicitly on the figure.

4.2 Importing IGES Files

While the ANSYS solid modeler is very powerful, there are other packages for solid modeling. Pro/ENGINEER has several FEA-related modules, including its own FEA solver. Note that Pro/Mechanica uses *different* elements, called p-elements, from the default ANSYS elements. p-elements achieve their accuracy by using higher order interpolation functions while the more popular h-elements achieve higher accuracy by using more interpolation points (i.e. a finer mesh).

Should you want to interface ANSYS with Pro/E there are two ways to do it. You can mesh objects in Pro/E, although I would recommend against that since 1) you have to open Pro/E every time you want to refine the mesh and 2) ANSYS has far superior meshing capabilities. The other option is to export an IGES file from Pro/E and read the IGES file in ANSYS.

This is the procedure for exporting an IGES file from the Pro/E Part menu,

```
Interface
  Export
    IGES
      WfrmSurfs
```

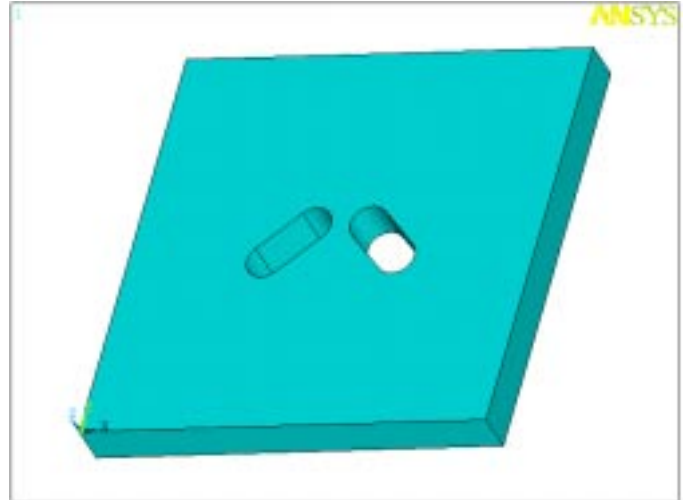
You can select **Wireframe**, **Surfaces**, or **WfrmSurfs**. Whether you want wireframe, surfaces, or both will depend on the nature of the model. Realize that in any case you will likely have to do some work on the model once it has been imported into ANSYS. You will likely need to merge lines and surfaces. You may even need to split up volumes or areas and segment lines for meshing.

From the ANSYS Utility Menu,

```
File
  Import >
    IGES ...
```

No doubt you will notice the Pro/E option of **Import**. Unfortunately, that utility is not available in our version of ANSYS, though we are looking into getting it. The default options for importing IGES should be sufficient. Find the file `twoslot.iges` that has been included in the `batch` subdirectory and click OK. To plot the volume, from the **Utility Menu**,

```
Plot
  Volumes
```



Another IGES example is located in the ANSYS Workbook. You can access this from the ANSYS Utility Menu

```
Help
  Table of Contents ▾
  ANSYS Workbook
```

Select Chapter 3, “Import of Solid Models.” Note that the IGES files are located in the directory `/stage/ansys54/data/workbook/iges/` here on the MEnet systems.

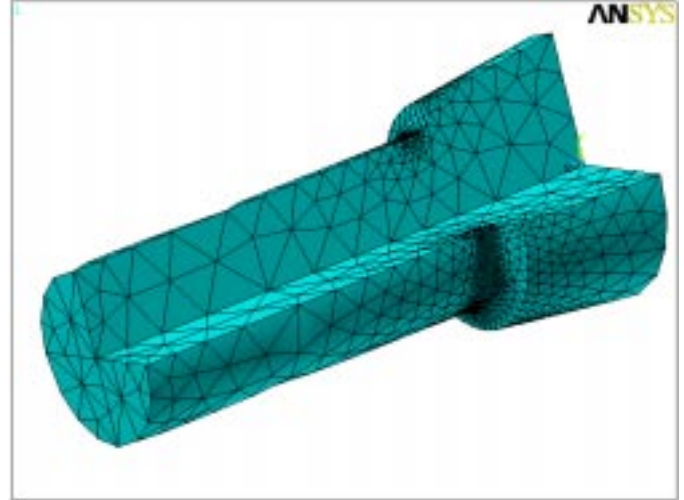
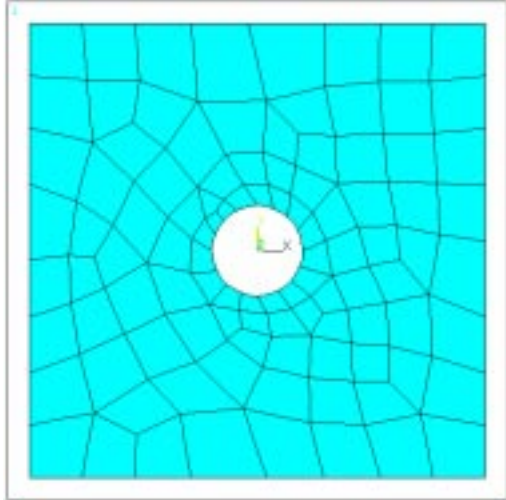
5 Meshing

Meshing a model can be the most difficult part of using any finite element package. While ANSYS gives the user a variety of automatic options so far as meshing is concerned, you are urged to use caution when using these tools. It is usually best to think about how you would like to mesh your model before you even go about making a model and creating areas. You will find that the time you spend thinking about how to split up areas and volumes will be time well spent.

In general, ANSYS has two methods of meshing: free meshing and mapped meshing.

5.1 Free Meshing

The figure below shows an example free mesh



The free mesh has no recognizable pattern and no regularity in the element shapes. Free meshing is easy but for complex geometries can often lead to distorted elements that undermine accuracy. Too often users free mesh a model—because it is easy—without bothering to worry about the resulting mesh.

Free meshing is available for 2D quadrilateral and triangular element shapes. Free meshing can only produce 3D tetrahedral elements for solid models, however.

Load in a copy of a shaft that is similar to the one discussed in the modeling section, `/input,tshaft`. We will use this as an example for the free meshing of 3D models. The elements should already be defined as quadratic structural solids. We'll use ANSYS's handy mesh tool for meshing,

```
Preprocessor >
MeshTool ...
```

This will pop up the MeshTool window. Select **Smart Size** and set the level to 6. ANSYS will attempt to determine the proper mesh. Now click **MESH**. You should get a mesh with more elements concentrated around the area of the notch root. You can refine the mesh in certain areas. Try

```
Refine: at Lines
```

Click **Refine**. Now select the lines around the circumference of the shaft near the notch root.

```
[LREF] Refine mesh at lines
LEVEL Level of refinement 2
OK
```

ANSYS will work for a while to refine the mesh around the notch root.

Note that you can unselect elements to see how the mesh looks inside. From the **Utility Menu**

```
Select
Entities ...
```

In the **Select Entities Menu**,

```
Elements
By Num/Pick
Unselect
OK
```

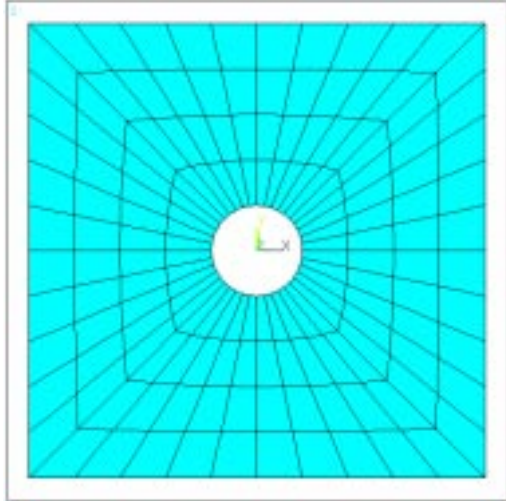
Use the **Pan-Zoom-Rotate** window to position the shaft. Then select the elements you wish to unselect using the box select. Finally, from the **Utility Menu**,

```
Plot
Elements
```

You can clear the mesh at any time by selecting **CLEAR** from the **MeshTool** window.

5.2 Mapped Meshing

Mapped meshes are easier to control and are oftentimes more accurate. Mapped meshes allow the user to more carefully specify the size and shape of the mesh in local regions. Mapped meshing is available for 2D and 3D elements. The figure below shows an example



Note the regularity in the mesh that virtually eliminates the possibility of varying results due to varying mesh sizes around an area of interest. There are restrictions to the use of mapped meshing,

- 2D** Each area must be four-sided—i.e. be made up of four lines. If the area is made up of more lines, you will need to split up the area to create sub-areas with four sides or you must concatenate lines so that four lines define the area.
- 3D** Each volume must have 6 faces (6 bounding areas). You will need to split volumes or concatenate lines and areas to create 6-faced volumes.

Mapped meshes are controlled by specifying element divisions on boundaries and by splitting areas and volumes in certain ways. Once you have split the areas and/or volumes in accordance with the above rules, use `lsel` to select the lines and `lesize` to set the number of element divisions along that line. Here is an example of how one might develop a mapped mesh for the plate with a hole. This batch file is located in `ansys/batch/platemsh`.

```
! name the file
/filename, platemsh

/prep7
```

```
! Define the box outer section (WIDTH,LENGTH)
! and the round inner section (RAD) diameters
! All dimensions in meters
PI=3.14159265359
WIDTH=0.02           ! overall width
LENGTH=0.02          ! overall length
HALF_WID=WIDTH/2.0
HALF_LEN=LENGTH/2.0
RAD=0.002            ! radius of thru-hole
```

```
COSSIN=cos( 45.0*PI/180.0 )

!# of element div's on each 45
! degree arc in the hole
DIV_HOLE=7

!# of element div's extending
! radially from the hole
DIV_HL_R=12

! Define keypoints for the box
k,1,HALF_WID,0
k,2,HALF_WID,HALF_LEN
k,3,0,HALF_LEN
k,4,-HALF_WID,HALF_LEN
k,5,-HALF_WID,0
k,6,-HALF_WID,-HALF_LEN
k,7,0,-HALF_LEN
k,8,HALF_WID,-HALF_LEN

! Define the keypoints for the circle
k,10,0,0
k,11,RAD,0
k,12,COSSIN*RAD,COSSIN*RAD
k,13,0,RAD
k,14,-COSSIN*RAD,COSSIN*RAD
k,15,-RAD,0
k,16,-COSSIN*RAD,-COSSIN*RAD
k,17,0,-RAD
k,18,COSSIN*RAD,-COSSIN*RAD

! Define the lines for the outer box
l,1,2 !1
l,2,3 !2
l,3,4 !3
l,4,5 !4
l,5,6 !5
l,6,7 !6
l,7,8 !7
l,8,1 !8

! Define the arcs for the inner circle
larc,11,12,10,RAD, !9
larc,12,13,10,RAD, !10
larc,13,14,10,RAD, !11
larc,14,15,10,RAD, !12
larc,15,16,10,RAD, !13
larc,16,17,10,RAD, !14
larc,17,18,10,RAD, !15
larc,18,11,10,RAD, !16

! Define the lines between the box and
! the inner circle
l,1,11 !17
l,2,12 !18
```

```

1,3,13 !19
1,4,14 !20
1,5,15 !21
1,6,16 !22
1,7,17 !23
1,8,18 !24

! Define the areas
al,1,18,9,17
al,2,19,10,18
al,3,20,11,19
al,4,21,12,20
al,5,22,13,21
al,6,23,14,22
al,7,24,15,23
al,8,17,16,24

! Now segment the lines before meshing
! segment the lines on the outer boundary
lssel,s,,1,8,1
lesize,all, , ,DIV_HOLE,1,

! segment the lines that define the hole
allsel
lssel,s,,9,16,1,
lesize,all, , ,DIV_HOLE,1,

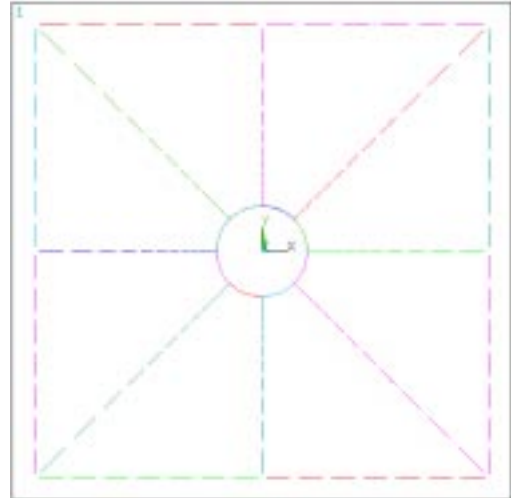
! segment the lines extending radially
! from the hole
lssel,all,all
lssel,s, , ,17,24,1
lesize,all, , ,DIV_HL_R,0.15,

! select the element and shape
et,1,plane82
type,1
eshape,3

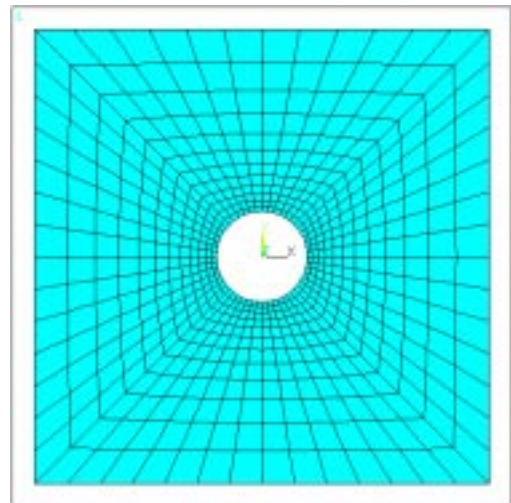
! select everything and mesh the areas
allsel
amesh,all

/eof

```

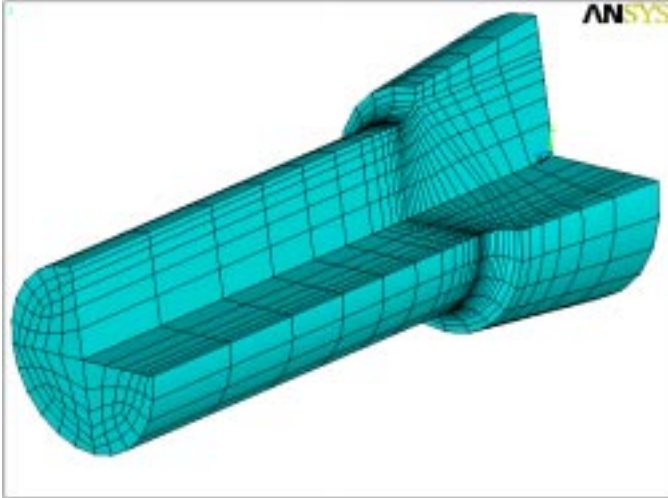


Note that the lines extending radially from the hole have larger element divisions towards the edges. This feature of mapped meshing allows the user to place smaller elements in the areas of high stress gradient (around the hole) while using larger elements where the gradient is not so steep (on the edge of the plate). The resulting mesh looks like



An example of a 3D mapped mesh is located in the file `qshaft`. Note how 4-sided areas are created so that when the area is revolved only 6-sided volumes result. The mesh looks significantly different from the free mesh of the same part shown in the previous subsection.

The resulting segmented lines look like



6 Solution

6.1 Solvers

The default direct frontal solver is fine for small linear problems. However, the size limitations become obvious when the user attempts to solve large 3D problems. Solving the FE problem is tantamount to solving a matrix equation with a very large matrix. Iterative methods are generally faster for bigger problems. ANSYS provides several different solver options, each of which may be more or less appropriate for a given problem. For structural analysis problems, I use the PCG or pre-conditioned conjugate gradient solver. From the Input window or in batch mode,

```
eqslv,pcg
```

For more info see the help for eqslv.

7 Post Processing

The ANSYS post processor provides a powerful tool for viewing results. This section will cover a few tips for using the post processor that you may not have discovered in the previous examples.

7.1 Element Table

The element table allows the user to make contour plots of any function. The function can be built from any variable that ANSYS calculates in its solution. First, clear any analyses using

```
File
Clear & Start New ...
OK
Yes
```

Type /input,platestr in the Input menu. This will create and solve the 2D structural example.

Say, for whatever reason, that we would like a contour plot of

$$s = \sigma_{xx}^2 + \sigma_{yy}^2$$

for this particular problem. First, let's visit the post processor and define items for the element table

```
General Postproc >
Element Table >
Define Table ...
Add...
```

Now you will "add" results to the element table, meaning that you will need to tell ANSYS which variables you would like to work with. Since we need σ_{xx} and σ_{yy} , add those

```
Lab User Label for Item s_xx
Stress
X-direction SX
Apply
Lab User Label for Item s_yy
Stress
Y-direction SY
OK
```

Click Close in the Element Table Data menu.

Now we need to operate on these variables.

```
Element Table >
Multiply ...
```

This will pop up another menu

```
LabR User label for result s_xx_sq
FACT1 1st Factor 1
Lab1 1st Element table item S_XX
FACT2 2nd Factor 1
Lab2 2nd Element table item S_XX
Apply
```

Now do the same for σ_{yy}

```
LabR User label for result s_yy_sq
FACT1 1st Factor 1
Lab1 1st Element table item S_YY
FACT2 2nd Factor 1
Lab2 2nd Element table item S_YY
Apply
```

Finally, we will sum these two

```
Element Table >
Add Items ...
```

which will spawn another menu

```
LabR User label for result s_sq
FACT1 1st Factor 1
Lab1 1st Element table item S_XX_SQ
FACT2 2nd Factor 1
Lab2 2nd Element table item S_YY_SQ
OK
```

To plot the results,

```
Element Table >
Plot Elem Table ...
```

From the Contour Plot of Element Table Data menu

```
[PLETAB] Contour Element Table Data
  Itlab Item to be plotted S_SQ
  Avglab Average...nodes? Yes - average
  OK
```

This produces a color contour plot of $s = \sigma_{xx}^2 + \sigma_{yy}^2$ over the surface of the model.

7.2 X-Y Plots

Sometimes, $X - Y$ plots are useful in interpreting results. They are especially useful when you need more visual accuracy than can be provided by filled contours.

In order to make an $X - Y$ plot, you first must define a path,

```
General Postprocessor >
  Path Operations >
    Define Path +
```

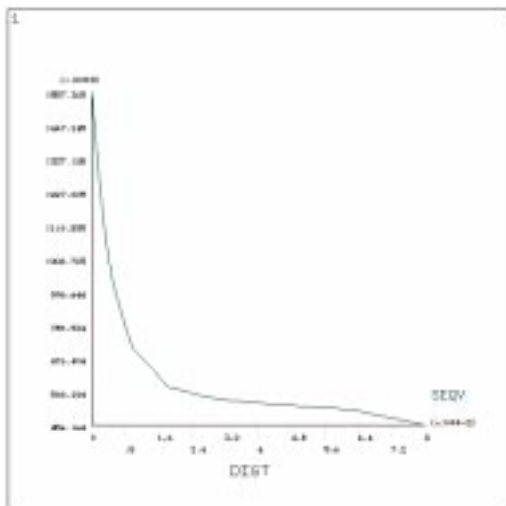
Now select two or more nodes that define the path (straight line or curve) along which you would like to plot a variable. Click OK when you are done. Whatever variable you wish to plot must be mapped to the path,

```
Path Operations >
  Map onto Path ...
```

Select the variable and Apply. Click OK when you are done. Finally, plot the variable along the path,

```
Path Operations >
  Plot Path Items ...
```

Select the variable and Apply. The equivalent stress from the hole edge to the plate edge in our structural example looks something like



Depending on the orientation of the model axes when you plot the path, you may need to reorient the $X - Y$ plot to see it.

7.3 Printing

ANSYS can print and/or save PostScript files. There are several options, depending on what your goal is. Here's an example from the Utility Menu,

```
PlotCtrls
  Hard Copy ...
    Graphics Window
    Gray Scale
    Landscape
    Save to: file.ps
    OK
```

You can control how your plots look from this menu, how many plots per window, etc. You are encouraged to experiment (and use the *help!*).

8 Tips, Tricks, and Other Random Comments

8.1 ANSYS Files

For a given `jobname`, ANSYS creates several files. Many of them are used by ANSYS in the solution of a problem and are of little use once the solution is complete. Here are some of the important extensions ANSYS uses to distinguish different files:

- *.err A log of all error and warning messages raised during this and all previous sessions.
- *.log A log of every ANSYS command issued by the user via GUI, Input menu, and batch files.
- *.db The ANSYS database file that stores all information about a model and a mesh.
- *.rst The file containing all the results of the previous solution.
- *.emat The element matrix needed by ANSYS for solution.
- *.esav The element saved data file.
- *.tri The triangularized matrix file used in the solution.
- *.erot, *.stat, *.PCS Temporary files used by ANSYS during solution.

Once you have saved a file using `SAVE_DB`, ANSYS will create a *.db file that contains all of the relevant model information. The filename is the same as the `jobname` that you assign to your model. You can access that file from the command-line

```
unix% ansys54 -g -p ansysrfr -j file
```

or from the menus. Remember to omit the `.db` extension from the filename. Using the menus, change the `jobname` (from the Utility Menu)

File

```
Change Jobname ...  
[/FILENAME] Enter jobname file  
OK
```

And from the toolbar

```
RESUME_DB
```

ANSYS writes the final results of an analysis to the *.db file and the *.rst file. Since the model and mesh are contained in the *.db file, that is the only file you need to keep once the solution is complete. You can remove the others without fear of losing vital information.

8.2 Memory Allocation

By default ANSYS allocates only a certain amount of computer memory (RAM) to store and solve models. You can request more memory at the command-line using the -m option. If you try to solve a large model and ANSYS runs out of memory (at which point it will ungracefully crash) you should request more memory. But if you request more memory than is available on the local machine, ANSYS can also crash. The default is 40 blocks. To request 64 blocks,

```
unix% ansys54 -g -p ansysrf -m 64 -j file
```

8.3 Disk Space and Network Traffic

If you plan to solve large models using ANSYS, you will need to think about efficient use of disk space. During the solution phase, ANSYS continually writes and reads large files from the directory where the solution has started. If you start the solution in your home directory, ANSYS writes and reads these files to/from the server over the network. This can overload the network and significantly slow down your solution. In addition, very few of us have large enough quotas to handle the amount of space ANSYS needs for large problems. You are encouraged to use the machine's local disk space for solution, located in /usr/tmp or /usr1/tmp depending on the machine. Some machines have large scratch spaces, others very small. Be aware and use

```
unix% df -k
```

to find out how much space is available on /. These temporary directories are cleaned often. *Do not use them for storage!*

Once the solution is done, you can archive and copy the important files to your home directory.

```
unix% tar -cvf file.tar file.db file  
unix% gzip -9 file.tar
```

To retrieve the info

```
unix% gunzip --stdout file.tar.gz | tar -xvf -
```

Archiving your files when not in use *saves valuable disk space*. You will get no sympathy from the MEnet staff if you ask for more disk space without first archiving files.

8.4 Running ANSYS without Wasting Resources

ANSYS is a resource hog. It uses large amounts of disk space, RAM, and CPU cycles. If you plan to run all but the simplest analyses, it is best to do them using batch files. Most importantly, you can run ANSYS without having to tie up a console—i.e., you can run your job in the background. Here is an example,

```
unix% nohup nice +20 ansys54 -p ansysrf <  
file > ans.out &
```

nohup: UNIX no hang-up command. So even if you log out, the ANSYS job will continue to run.

nice +20: UNIX command that “nices” the job by adding to its priority. This means that your background job will defer CPU cycles to the person logged in to the console. All background jobs *must* be niced.

< file: Pipes the batch file file to ANSYS.

> ans.out: Pipes any output to the file ans.out.

8.5 Optimization

Frequently, students use ANSYS for comparative analyses. ANSYS has optimization capabilities built right in. So if you want to see how changing a length or a diameter or a material property changes the stress at some critical location, ANSYS can do that *automatically*. You will need to use the optimization routines of ANSYS. To do so, you must draw your model using parameters (if you plan to change the geometry). The basic idea is to give ANSYS a list of parameters it can vary in the design, usually geometry-related. You must put bounds on each parameter. Then, the user applies constraints to the problem, e.g. the stress at point *X* cannot be greater than a certain value, the weight of the part must be less than *Y*, or the center of gravity must fall within a certain range. These are all considered optimization variables by ANSYS. Finally, the user must provide an objective function, a function that quantifies the “goodness” of the design. ANSYS will minimize this objective function subject to the constraints.

For instance, say we wished to optimize the design of our axially-loaded plate with hole to minimize the stress at the hole edge. We will assume that the radius of the hole can vary from 1mm to 9mm. We will start the optimization at a radius of 8mm to make things interesting. An excerpt from the batch file, located in ansys/batch/plateopt, appears below

```
! name the file  
/filename,plateopt
```

```
! enter the pre-processor  
/prep7
```

```
!
```

```

! This file draws a 2D model of a plate
! with a hole using areas.

WIDTH=0.02           ! width of the plate
HEIGHT=0.02          ! height of the plate
WID_BY2=WIDTH/2.0
HGHT_BY2=HEIGHT/2.0
RADIUS=0.008         ! radius of the hole
EPS=WIDTH*1.0e-4     ! small number
FAXIAL=1000          ! axial load in N

.
.
.

! select everything and solve
allsel
solve
finish

! enter the post-processor
/post1
set
allsel,all,all

! select the lines that define the hole
lsl,r,loc,x,-RADIUS-EPS,RADIUS+EPS
lsl,r,loc,y,-RADIUS-EPS,RADIUS+EPS

! select the nodes on the line;
! sort the equivalent stress to find
! the max and assign it to a variable
nsl,r,1
nsort,s,eqv
*get,eqvmax,sort,,max

! now do the optimization
/opt

! RADIUS is a design variable that
! we vary from 0.001 to 0.009
opvar,RADIUS,dv,0.001,0.009

! the stress is a state variable
! that we want to minimize (i.e. the
! objective function
opvar,eqvmax,obj

! assign an optimization
! technique
optype,first
opfrst,100,50,0.1
oploop,prep

```

```

opprnt,full

! execute
opexe
fini

/eof

```

ANSYS will output the value of each of the optimization variables and the objective function at the end of each iteration. Finally, ANSYS will return the optimal values of the design variables. For the plate example,

```

RADIUS (DV) 0.18750E-02
EQVMAX (OBJ) 0.15511E+06

```

As with any optimization, the results are not guaranteed to be the global optimum. Furthermore, the optimum found by the search may be different depending on the starting point (initial values of the optimization variables) that you choose.