University of Utah
ME EN 6510/5510
Introduction to Finite Elements
Fall 2005

Running ANSYS on the CADE lab workstations

CADE Lab Accounts
You need to get account information in EMCB 224 from the operators at the back. If you already have an account, you can use your existing account equally well.

ANSYS is running on the workstations in EMCB 224. You can either work directly in the CADE lab or access the CADE lab machines over the network through machines lab3-1.eng.utah.edu through lab3- 24.eng.utah.edu and other CADE lab machines. Please contact the CADE lab website for more information (http://www.cade.utah.edu and look under FAQ). The CADE lab operators can help if you cannot find the information on CADE website.

If you are having trouble logging on to a UNIX machine in the CADE lab, the first thing to do is to try another machine. If you are still having trouble, it is possible that your account is not set up correctly. See the operators in the back of EMCB 224 to ensure that you have access and that your dot files are set up properly.

UNIX Operating System
The operating system used on the CADE machines is UNIX. At start up, you will be asked if you want to start windows. Confirm that you do want to start windows. It may be helpful to learn a few UNIX commands to manipulate files and perform other routine tasks. A basic UNIX tutorial can also be found on the web at:

   www.math.utah.edu/lab/unix/unix-tutorial.html

Common UNIX commands are:

  ls list your files
  pwd print the current directory name or lists path
  mkdir [*directory_name*] add a new directory (directories hold files)
  cd [*directory_name*] change to another directory
  cd.. back up a directory level
  cp [*filename*] [*newfilename*] copy a file to another file
  rm [*filename*] remove a file
  cat [*filename*] list the contents of a text file
  tail [*filename*] display the end of a text file
  man [*unix_command*] **display information on a specific command**
  quota show amount of disk space available to you

Note: If information is ever needed on how to use a specific UNIX command, it is a good idea to look up the manual page for that command. For example, at the UNIX prompt, type: ***man cp***
This will supply information on how to use the ***cp*** command. Type Q to quit.

Introduction to ANSYS
ANSYS is a large and complex program with many different capabilities. Learning the program and documentation of the various features can be a significant challenge for the user. This handout is intended to assist the new ANSYS user get started.

Starting ANSYS
We are currently using version 8.0 of ANSYS. The ANSYS program can be started two ways in the CADE lab. The first is to use the menu provided by the window manager in X-Windows. A more convenient way is to go to a directory of your choice and then type in at the UNIX prompt in the **xterm** window:

xansys &

This command opens the most recent version of ANSYS. If this does not work, see the operators in the CADE lab to make sure that your path is set properly.

Five sub-windows (in one big window) will then come up on the screen. At the top of your screen is the general ANSYS utility bar with several pull down menus, including the interactive help menu. These commands are available at all times during ANSYS operation. Below this is the command input window. Here ANSYS commands can be input directly, if the user has learned the command direct input form (as opposed to graphically). Below and to the left of the command input window you will find the main menu window. Here you have access to each of the three modules for solving a finite element model.

- The **Preprocessor** module is used to build the model
- The **Solution** module is used to solve the model and generate the results
- The **General Postprocessor** module is used to review the analysis results.

Activating each of these modules will bring up
a window of available commands unique to each module. The large window to the right of the main menu is the graphics display window. Model features such as geometry, nodes, elements, boundary conditions and results can be displayed in this window. The fifth window, to the right of the command input window, is a user customized toolbar. This window allows the user to specify their own buttons to perform ANSYS tasks. It is not likely that there will be much use for this window in this class.

How to Work with ANSYS
ANSYS can be run through the graphic user interface and commands input window, or by creating an ASCII text file for input, or both. While the user could enlist one method or the other exclusively, it is often more desirable to use a combination of both graphical methods and input files. An input file is often useful for defining the initial model geometry, characteristics, and constraints. This is because the input file can be easily changed to correct the errors that are inevitable when setting up a new ANSYS model. Once the input is correct and the model geometry and material characteristics are read into ANSYS, it is then desirable to examine the results using either direct command input with the command input window, or the graphical menu selections. This technique of solving ANSYS problems will become clearer as you read on through this tutorial and begin to work some examples. First let us take a quick look at how ANSYS works with files.

Everything that is done within ANSYS during a given session is recorded in a text file. This text is called your <filename>.log and is written to your current directory (default setting is file.log).

These commands could have been entered graphically using menus and selections, or they could have been entered directly with the command input window. The result is a file containing the history of all your work up to that point. When you are using GUI you can create a logfile by (Mainmenu<File<Write DB log file option). This file is not like the "emacs" file you create. All unnecessary operations like zooming in etc. will show up as commands. This file can be cleaned up and used for further reference. This will save time in having to retrace many steps in the process of model creation if a mistake was made or if a minor change needs to be implemented.

Here is how you would go about doing this:

At a UNIX prompt, copy the your_filename.log to a new filename:

cp filename.log newfile.inp

Edit the new file with any standard text editor. The standard UNIX system editor such as vi, emacs, and emacs19 are available for preparing files. Emacs19 is by far the easiest to use, and if you are in Xwindows it permits mouse support for editing and pull down menus for ease of use.

It is just about self-teaching, and a tutorial is available on the help menu. To invoke emacs19, just type
emacs
OR
emacs <filename.xxx>

where, filename is either a new file or an existing file that you want to edit. Typing "**emacs &**" from the UNIX prompt will open this editor in a new window, allowing ANSYS to be run in the current window. The following shortcut keys may also be used with emacs19, and may be useful

ctrl x ctrl f	Prompts you for the name of a file to be edited
ctrl x ctrl s	Saves the current file and prompts for a filename if unnamed
ctrl x ctrl c	Closes the editor
ctrl @	Sets a mark (also ctrl spacebar)
ctrl w	Deletes text between the mark and the cursor position
ctrl y	Inserts deleted text with ctrl w at the current cursor position

but remember that you can just use the "PC style" interface provided and skip most of the UNIX.

Enter ANSYS and in the direct command input window, enter **/input,newfile,inp** (this assumes your input file is called filename.inp). Note the use of commas in place of periods.

ANSYS commands entered within the command window use commas to separate fields. In this case, the extension on the filename (inp) is considered a separate field. Also, the ANSYS input format allows only a maximum of 8 characters in each field. Consider this when naming the files in the UNIX environment, which will accept much larger file names.

Printing From ANSYS
After you have everything running OK on a problem, you will want to print out hard
copy results. The output will be either in text or graphics form. In either case, be sure to display
the output on the screen first before printing, to be sure that you have what you want.
There are two methods to get a hard copy of text information. The easiest is to use the LIST pull
down menu (or type the desired list command in the direct input window) to list the desired data.
The data will be listed in a new window. Selecting the File option in the new window, one will
see the options either to save the data to a file or to print the data. If you select to print the data,
you will be prompted for the printer (cadeps or cadeps1). If you select to save the data, an ASCII
text file will be created in the current directory.

This file can then be printed with the "enscript" UNIX command.

To get a paper copy of plots, display the plot in the graphics window, then go to the utility menu,
select
>plot controls>Hardcopy, then highlight the graphics window (with the mouse), highlight print
to, and enter the name cadeps or cadeps1.

Managing Files
ANSYS writes a number of files to your directory for each run. The prefix of these files
will be the filename you enter at startup. These are used primarily for record keeping, and
resuming a run that has been stopped. For the projects in this class it is not necessary or desirable
to keep these files after you have finished the assignment. Use the UNIX command to list the
names of the files, and delete the files from your directory (**ls** to list, **rm** to remove). Two ascii
files of interest are the filename.log and the filename.err. These are ascii files that can be read.

The .log file has been discussed previously. The .err file list errors in command input structure or
model solution setup that occurred during the life history of the project (both files are appended
each time the model is restarted). Some other files of interest are the filename.db (database file),
filename.rst (structural analysis results file), and filename.rth (thermal analysis results file).
These are binary files and are only machine readable (you can not edit them). The database file
stores the model definition. The other two store analysis results. Both the .db and the appropriate
.r** file must be available to postprocess analysis results.

Direct Command Input and Input Files
As an alternative to the graphical user interface described above, the ANSYS commands can be
input directly. This can be done either directly with the command input window or it can be done
by creating an input file using emacs19 or some other text editor and reading it into ANSYS.
Also, as mentioned above, these commands are recorded in the filename. log file. This file can be
edited to correct errors, and used to restart the run without repeating the operations that produced
the filename.log file.

Thus it is useful to be familiar with direct input commands. An overview of a general input file
format is as follows. The ANSYS commands are on the left, and the comments following the "!"
are for your information and are not entered. (note: ANSYS uses a ! for comments. If you have a
comment you would like to include within an input file, precede it with an
exclamation point. The comment will then extend to the end of the line).

/inter	! interactive mode
/show,x11	! tells ANSYS how to display graphics
/prep7	! The prep7 module (the preprocessor is for problem definition (just like clicking PREPROCESSOR in the menu)
/input,[filename]	! execute the file that you have created
/pbc, [option], [option]	! show boundary conditions on plots using options you supply
/psf, [option], [option]	! show loading, note options
eplot	! plot geometry, for you to check visually
nplot	! plot nodes
finish	! exit the prep7 preprocessor. If model is ok, proceed to the solution phase below, otherwise edit the input file to correct errors
/clear	! clear geometry from ANSYS
/solu	! enter the solution processor (just like clicking SOLUTION in the menu)
solve	! initiate the solution
finish	! exit the solution processor
/post1	! enter the post-processor, used to look at the results
/show,x11	! tells ANSYS how to display graphics
plnsol,[option], [option]	! a number of commands are used here to view the results
finish	! exits the post processor
/eof	! exits ANSYS – only use this if you are done working

This example is meant only to convey the form most of your input files will take. **It should not actually be entered, as it would not do anything useful (it does not supply ANSYS with any model geometry or constraints)**. An input file is always read from beginning to end, and the three stages of solving an ANSYS model must be adhered to at some point in the solution process. These are:

1) PREPROCESSOR (PREP7)
2) SOLUTION (SOLU)
3) POSTPROCESSOR (POST1)

As can be seen from the input file listing above, each module is entered, module specific commands are executed, and then each module is exited. This is important because each module only recognizes certain commands. For example, attempting to use the **plnsol** command while with the PREPROCESSOR will produce an error ("not a recognized PREP7 command") whereas the same command would produce no error when used within the appropriate POSTPROCESSOR module. Learning which commands are appropriate for which ANSYS module is a matter of experience. By working the examples below you will gain insight as to how to use these commands. Another thing to note about this general input file is that all three modes are entered, whereas in an actual input file this is not usually the case. You will often find it easiest to define only model geometry, characteristics and constraints with the input file (these are all PREPROCESSOR commands). Then run the solution and review the results manually using the command input window. The reason for this is that ANSYS will

attempt to continue with an input file even if errors are found. This can waste time if you realize that you made a mistake in the model definition. The best approach is to always make sure that you have the model properly defined and constrained before proceeding to the solution. Much of the ANSYS documentation can be obtained on-line. To get help on any command, type

help, <command>

Another option to get help is to select HELP on the menu bar at the top of the screen. Here you can read about the commands, theory, and element types used by ANSYS. If you still can not find what you need, written documentation is also available and can be checked out from the computer operators. Be sure to get the ANSYS version 5 manual, not older versions like the version 4.4 used previously. Some Common ANSYS commands
Below are some of the most widely used ANSYS commands. These and other commands will be used in the examples given below.

|	
k	! define a keypoint
l	! define a line
a	! define an area
n	! define a node
e	! define elements
nplot	! plot nodes
kplot	! plot keypoints
aplot	! plot areas
eplot	! plot elements
lplot	! plot lines
nlist	! list nodes
klist	! list keypoints
alist	! list areas
elist	! list elements
llist	! list lines
prdisp	! list displacements
pldisp	! plot displacements
prnsol	! list nodal solution
plnsol	! plot nodal solution
lmesh	! mesh lines (PREP7 command)
emesh	! mesh elements (PREP7 command)
amesh	! mesh areas (PREP7 command)
etable	! generate element table (POST1 command)
pretab	! print a list of etable results (POST1)
pletab	! plot etable results (POST1)
/pbc	! show boundary conditions
/psf	! show applied loads
/pnum	! apply nodal numbering
/show	! direct output to a device
/zoom	! zoom in or out

ANSYS Example #1

Solve the simple truss structure shown below

Truss Properties:
Cross Sectional Area = 5in^2
Modulus= 1 Msi = 1e6 psi
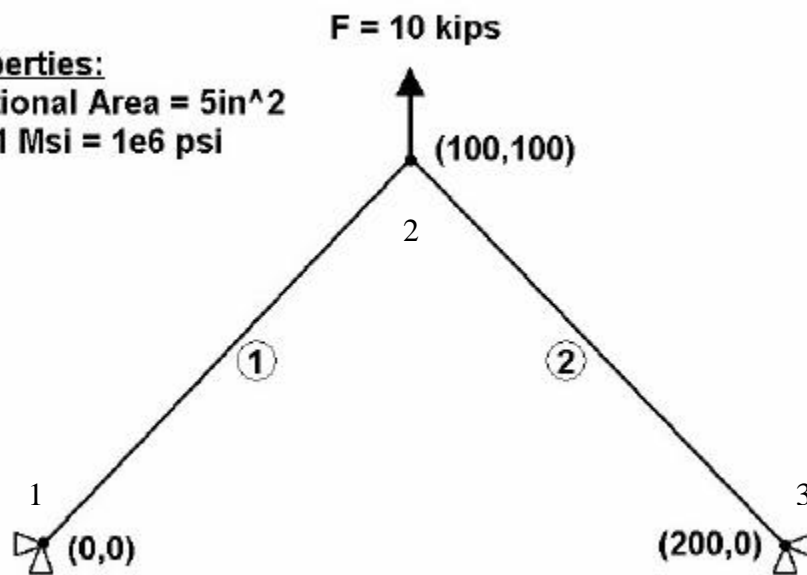
Use element type "link1" which is a 2-d truss element.

Method 1: GUI (Graphical User Interface)
The GUI is simply the window(s) you see when you launch ANSYS that contains the icons and menus used to create your model. This section of the lab shows how to solve the problem using the ANSYS GUI.

Step 1: Define Nodal Locations
Preprocessor→Modeling→Create→Nodes→In Active CS→(window) define node 1 with label and XYZ coordinates, hit *Apply* and repeat for nodes 2 and 3→when finished, select *OK* to exit window.

Note: You can check to make sure you have placed the nodes with the right coordinates by using the List menu (in uppermost menu bar). Choose *List→Nodes→*(options window) select *OK→*displays list of nodes with coordinates. (Another way to do this is simply typing *nlist* in the command input window.)

Step 2: Select Element Type
Preprocessor→Element Type→Add/Edit/Delete→(window) *Add...→*(window) highlight *Link, 2D-spar,→OK→CLOSE.*

Step 3: Define Material Properties
Preprocessor→Material Properties→Material Models→(window) double click *Structural/Linear/Elastic/Isotropic→*(window) input modulus and Poisson's ratio (not necessary for LINK elements [why?], but you can use ν=0.3)→*OK→*(close Material Model window).

Step 4: Define Physical Property Set 1 (same for elements 1 and 2)
Preprocessor→RealConstants→Add/Edit/Delete →(window) *Add...→*(window with element Type 1 {Link 1} highlighted) *OK→*(window) input Cross-sectional area of 5→*OK→CLOSE.*

Step 5: Build Elements Between the Nodes
First, look at the fields to the right of the bottom information window. You should see the current material type is equal to 1 (mat=1), the current element type is 1 (type=1), and the current physical property set is 1 (real=1). When you create elements, ANSYS builds the current element type using the current material type and current real constant set. Since all elements in our structure are of element type=1 (i.e. Link1), material 1 (i.e. E=1e6 and ν=0.3), and physical property set 1 (i.e A=5in^2), we can simply create all elements without having to worry about changing any of these settings.

Build Elements 1 and 2
Preprocessor→Modeling→Create→Elements→User Numbered→Thru Nodes→(window) assign element as No. 1, select *OK→*(selection window) pick nodes 1 and 2→*OK* (creates element 1).
Preprocessor→Modeling→Create→Elements→User Numbered→Thru Nodes→(window) assign element as No. 2, select *OK→*(selection window) pick nodes 2 and 3→*OK* (creates element 2).

Step 6: Apply Boundary Conditions
Preprocessor→Loads→Define Loads→Apply→Structural→Displacement→On Nodes→(window) pick nodes 1 and 3, then select *OK→*(window) highlight *ALL DOF*; make sure it shows *Apply As: Constant Value*; enter value as 0, select *OK*.

Step 7: Apply Loads
Preprocessor→Loads→Define Loads→Apply→Structural→Force/Moment→On Nodes→(window) pick node 2, then select *OK→*(window) choose Fy direction and enter 10,000→*Apply→OK*.

Step 8: Solve
Solution→Solve→Current LS→(asks you to review summary info) select *OK→*ANSYS will begin solving the problem and will post a message "Solution is done!" when it has finished. Close message windows and go to next step.

Step 9: View Results

Plot Deformation: *General Postproc→Plot Results→Contour Plot→Nodal Solution→*(window) highlight *DOF solution* and *Translation Ux* (or Uy—play with both); pick button *Def + undeformed*; select *OK*.

List Nodal Displacements: *General Postproc→List Results→ Nodal Solution→*(window) highlight *DOF solution* and *All DOFs*; select *OK*. (This can also be done by simply typing *prdisp* in the command input window.) You should get the following:

	Ux (in)	Uy (in)
Node1	0	0
Node2	?	? (you fill these in and compare with each other).
Node3	0	0

List Axial Stresses: *General Postproc→Element Table→Define Table* and Click **Add**. Enter *sigax* into the **label for item** input box. In **Item, Comp Result Data Item**, choose **By sequence num**, then select **LS,** and enter *LS,1* in the selection input box. Leave everything else as it is. You have just defined the element table containing axial stress for each element. To view the results, go to *General Postproc→Element Table→List Elem Table* and select **sigax** and click **Ok**.

You should get the following:

	Axial Stress (psi)
Element1	1414.2
Element2	1414.2

Method 2: Input file (write code directly for ANSYS solver)
A s an alternative, you can manually create an input file that contains the proper commands (code) to build and solve the model without ever having to use the GUI. (This input file is written for you when using the GUI—your menu selections, etc. are automatically written as code in the input file.) Shown below are the commands you can enter in a text file (or you can directly type the code into the command input window, which is highly recommended when you are first starting out) to solve this same problem:

```
/prep7              !initiates the input phase of ANSYS (preprocessor mode)
n,1,0,0,0           !creates node 1 at (0,0,0)
n,2,100,100,0       !creates node 2 at (100,100,0)
n,3,200,0,0         !creates node 3 at (200,0,0)
et,1,link1          !sets element type 1 to Link1
mp,ex,1,1e6         !sets modulus for material 1 to 1e6
mp,prxy,1,0.3       !sets Poisson's ratio for material 1 to 0.3
r,1,5               !defines property set 1 (i.e. A=5in2) for Link1 element type
e,1,2               !puts element of current type and properties (default real=1)
                    ! between nodes 1 and 2
e,2,3               !puts element of current type and properties between nodes 2 and 3
d,1,all,0           !constrains displacements at node 1 to zero
d,3,all,0           !constrains displacements at node 3 to zero
f,2,fy,10e3         !applies a force of 10,000 to node 2 in the y direction
finish              !exits the preprocessor mode
```

For solving the problem, you can type the following directly into the command input window:

```
/solu               !initiates the solution phase of ANSYS (solver mode)
solve               !runs the solution
finish              !after ANSYS has solved, this command is typed to exit the
                    !solution phase
```

For viewing results, you can type the following directly into the command input window:

```
/post1              !initiates the ANSYS postprocessor
pldisp,1            !plot displacements over original geometry
prdisp              !print (list) the nodal displacements
etable,sigax,ls,1   !load axial stresses (ls,1) into table under user defined name
                    !'sigax'
```

```
pretab,sigax          !print (list) axial stresses that are stored in the 'sigax' table
finish                !exits from the postprocessor
```

—