ANSYS MACROS FOR ILLUSTRATING CONCEPTS IN MECHANICAL ENGINEERING COURSES

John R. Baker
Department of Mechanical Engineering
University of Kentucky

Abstract

Finite element analysis (FEA) is a widely used tool in structural analysis. Because of advances in graphics and animation capabilities, commercial FEA programs can be of great benefit in aiding students in understanding concepts in many undergraduate engineering courses. However, although software vendors have made great strides in enhancing the user-friendliness of commercial codes, proper use of complex FEA software still requires significant training and experience. There is typically not time to instruct students in use of FEA software, or in FEA theory, in courses which are not specifically FEA courses. This work builds on previously reported work and describes a number of macros that have been developed which automate modeling procedures using the FEA code, ANSYS, to easily produce results, and high-level graphical representations of results, related to a range of topics in structural mechanics. Macros have been developed that illustrate concepts in courses such as statics, dynamics, solid mechanics, machine design, and vibrations, with minimal input from the user. No FEA theoretical background is required to use the macros. They can be used by students in stand-alone assignments or by instructors in in-class demos. Some examples of results that can be generated and graphically depicted are stress contours due to beam bending, stress contours illustrating the stress concentration in the fillet region in a loaded bar with a step diameter change, animated structural vibration mode shapes, and plots of displacement and velocity of a projectile versus time. Inexperienced users can also easily display numerical results in tables generated by the software. Nonlinear solutions can be generated, for instance, in large deflection beam bending problems to illustrate the fact that standard textbook solutions are only valid within some operating range.

Introduction

ANSYS is a widely-used software tool for structural, thermal, and fluid flow analysis. Many universities include use of ANSYS[1-6] or a similar finite element package to some extent in their mechanical engineering curricula. However, a significant amount of time is needed to learn to use a complex finite element package such as ANSYS. Therefore, it is not practical to expect students to use ANSYS in a course, unless a course in which they learn to use ANSYS is a prerequisite, or else specific ANSYS tutorials for a certain type of analysis are provided. In previous work[6], incorporation of ANSYS in a range of courses, such as a fluid mechanics course, a heat transfer course, and a vibrations course, was described. The student assignments described in this previous work were tailored toward specific analysis for the students. An outgrowth from that experience is the effort described in this paper. It seems with the ANSYS macro capability, based on the versatile ANSYS Parametric Design Language (APDL), there could likely be some benefit of developing a wide range of flexible "canned" macros that instructors could use in many mechanical engineering classes; this would require extremely small time and effort on the part of the students and instructors in the classes. The results, including high level graphics generated within the software, would supplement standard course materials.

The following discussion varies from the discussion in the earlier work[6] in that the emphasis below is on clearly demonstrating the approach of using APDL for developing flexible

macros that could be used in courses in which the instructor does not have time to include any significant discussion of the basics of FEA, and there is not time for students to spend on lengthy ANSYS tutorials. Some brief class discussion should probably be provided to explain to the students that FEA is an approximate analysis technique and that results obtained will not necessarily agree exactly with closed-form textbook solutions in some cases. Also, it may be ill advised to use macros, such as those described below, in a curriculum in which no finite element instruction is provided before graduation, as students should understand the approximations and limitations associated with finite element analysis before they become practicing engineers. However, the advantage of using canned routines, based on a tool like ANSYS, in many courses would be that students can take advantage of the computing power and high-level graphics of a tool like ANSYS in courses they may take prior to a required FEA course, depending on how the curriculum of a particular university is set up chronologically.

It seems, as outlined below, that the software can be used to great advantage in illustrating concepts in many courses through use of customized macros by students who may not yet have had a FEA course. The software has reached a level of sophistication in terms of plotting capabilities and solution speed such that solutions to complex problems can be obtained by students in classes without a great deal of time expended by the students.

In the following section, three different types of analysis are described and full listings of the associated macros that have been developed are also provided. One set of examples relates to static stress analysis in beams, another example relates to plate vibration, and a final example relates to projectile motion. From the discussion and review of the listings of the macros, it should be clear that the macros could be easily altered to optimize their use in a specific setting or for a specific purpose. These examples are intended to simply provide an overview of the approach. Similar macros have been used to illustrate concepts in both a vibrations course and a control theory course at the University of Kentucky.

Examples

Static Beam Analysis Macros

Euler beam theory is a primary topic covered in typical solid mechanics classes. Here, simple example macros are discussed which perform static stress analysis on cantilevered beams with a lateral shear force and/or a pure bending moment at the unclamped end. Certainly, more options can be added related to loading and boundary conditions. Currently, there are two macros for generating models with two different cross-section shapes. One macro, named *rbeam.mac,* creates a 3D model of a rectangular cross-section beam, and another similar macro, *ibeam.mac*, creates a 3D model of an I-Beam. The full listing of the macro, *rbeam.mac*, written in APDL, is provided in Appendix A. These macros utilize ANSYS Solid45 3D solid finite elements.

The user (student) can use the macros, prior to receiving any FEA instruction, to generate a 3-D solid finite element model of a beam and calculate the deflections and stresses due to the defined loading by entering a single command at the ANSYS command prompt. The macros must be in the user's ANSYS working directory. To generate a 3D model of a rectangular cross-section beam, the format of the command, which must be entered at the ANSYS command line, is simply

$$rbeam,B,H,L,F,M,E,v$$

where the dimension parameters B,H, and L, and the force, F, and moment, M, are defined in Figure 1. E is Young's Modulus and v is Poisson's ratio.

Therefore, to analyze a steel beam with a rectangular cross-section of 1" x 2", a length of 40 inches, a lateral end force of F=200 pounds,

the student could input the following command at the ANSYS command prompt:

rbeam,1,2,40,200,0,30E6,0.30

The model is automatically created, and a plot with two separate views appears showing the deflection pattern, with color contours indicating the normal stress in the axial direction. The plot that appears using the input above is shown in Figure 2.

For additional postprocessing, a macro named *beampost.mac* is supplied. This macro is listed in Appendix B. The format of the command to execute this macro is:

beampost,option,zloc

If, option=1, a stress contour plot for normal stresses in the axial direction is produced for a segment of the beam about the axial location entered as "zloc". If zloc is zero, the stress contour plot includes the entire length of the beam. For option=2, a stress contour plot for shear stress in the beam is produced for a segment of the beam about the axial location entered as "zloc". For option=3, a plot of normal stress vs. vertical location in the beam section at the axial location "zloc" is produced. For option=4, a plot of shear stress vs. vertical location in the beam section at the axial location "zloc" is produced.

As examples of results generated using *beampost.mac*, for the case outlined above, with a lateral end forced of 200 pounds, and no additional bending moment applied, if the following command is executed, the normal stress contour plot in Figure 3 is produced:

beampost,1,20

The macros include plotting commands that automatically generate some annotations on the plots, so students can verify the information they are viewing. Note that, in the input as entered above, the option "1" was selected, to request a normal stress contour plot about the

Z=20" axial location (the beam is 40 inches long in this case). The annotations in the plot in Figure 3 indicate that the beam segment plotted spans from 19.733 inches to 20.267 inches, and this distance is measured from the clamped end. If the command is entered as "beampost,2,20", the shear stress contour plot in Figure 4 is produced. If the command "beampost,3,20" is executed, the plot of normal stress as a function of vertical location in cross-section, shown in Figure 5, is produced. If the command, "beampost,4,20", is entered, the plot of shear stress as a function of vertical location in the cross-section, shown in Figure 6, is produced.

The results shown are in very good agreement with that expected from Euler beam theory. It should be noted that ANSYS does include beam finite elements, which are simple 2-node line elements, in its element library. These could be used to obtain excellent numerical results as compared to beam theory, but a primary point of these particular macros is to allow for better visualization of results through stress contours in a 3D representation of a structure.

A second example was run using *rbeam.mac*, but in this case, a pure moment was applied and no lateral shear force. Normal stress contours for the case with a shear force are compared to those with a pure bending moment, with stress contours for both cases shown in Figure 7. These plots could be helpful to students trying to understand the effect of various loading assumptions on the state of stress in a beam. For the case with a pure moment (results shown at right in Figure 7), the normal stress through the cross-section does not vary with axial location along the beam. This result is in agreement with beam theory and is a different result from the case shown at left in Figure 7, which corresponds to an applied shear force, which produces a varying moment and, thus, a normal stress distribution that varies along the beam's axial direction.

As a further example, the more complicated I-beam shape can be modeled using the macro *ibeam.mac*. The usage of both *rbeam.mac* and

ibeam.mac is similar. Parameters needed for input to *ibeam.mac* are shown in Figure 8. The same postprocessing macro, *beampost.mac*, can be used for either a rectangular cross-section beam or an I-beam. Results for the I-Beam were generated assuming a lateral end shear force but no additional bending moment.

Figure 9 shows the normal stress distribution in a segment of the I-beam, and Figure 10 shows the shear stress distribution in the same segment. Both plots were generated using *beampost.mac,* in the manner described above. The plots of normal stress and shear stress along a vertical path on the cross-section, generated using *beampost.mac*, are shown in Figure 11 and illustrate the fact that the shear stress distribution for the I-beam has a different variation than that for a rectangular beam (this is seen by comparing Figure 6 and the right side plot of Figure 11); but for both the rectangular beam and the I-beam, the normal stress distribution varies linearly along a vertical path on the cross-section.

The macro *beampost.mac* also outputs the shear stress and normal stress, as a function of distance along the cross-section, to a file named "beamstress.txt" whenever option 3 or 4 is chosen.

Plate Vibration Analysis Macro

As an example of a second type of system analysis using a "canned" macro, a macro for performing modal analysis on a plate with ANSYS is provided. This macro utilizes ANSYS Shell63 shell elements. In this example, the macro is named *platevib.mac*, and the format for executing the macro at the ANSYS command line is

platevib,W,L,T,E,ρ,ν,BC

where W is the plate width along the global X-direction, L is the plate length along the global Y-direction, T is the plate thickness, E is Young's Modulus, ρ is the mass density, ν is Poisson's ratio, and BC is a boundary condition

option. As currently written, the macro allows for two boundary condition options, but more could be easily added. For BC=1, the ends at Y=0 and Y=L are clamped, and for BC=0, free-free boundary conditions are assumed. The macro plots the mode shape of the first bending mode in a color contour plot on the screen and writes the first two bending mode natural frequencies and corresponding mode shapes (excluding rigid body modes in the free-free case) to a file named "plateout.txt". Of course, additional post-processing options and boundary condition options could easily be added.

For a case of a steel plate with W=L=20", T=0.1", assuming fixed end boundary conditions, the macro would be executed using the command

platevib,20,20,0.1,30E6,0.282/386.4,0.30,1

The resulting mode shape plot is shown in Figure 12, and the natural frequency listing is shown in Figure 13. The full macro listing is provided in Appendix C.

Projectile Motion Analysis

As a final simple example, a single node finite element model with a lumped mass is created, and the motion of this point mass in a gravitational field is calculated as a function of time. The short macro is listed in Figure 14. The command format to execute it is

project,ivel,theta,m,g,t

where ivel is the initial projectile velocity, theta is the angle at which it is fired with respect to horizontal, m is the mass of the projectile, g is acceleration due to gravity, and t is total analysis time.

To analyze for 3 seconds the motion of a 10 pound projectile with an initial velocity of 1000 in/sec fired at 30° with respect to horizontal, the following command would be entered

project,1000,30,10/386.4,386.4,3

The macro plots the projectile trajectory in the X-Y plane (where X is horizontal), as shown in Figure 15. It also writes out X and Y projectile location as a function of time to a text file named "trajectory.txt". As mentioned with regard to the other examples above, more options could easily be added to this analysis macro.

Summary and Conclusions

Canned, flexible macros for use with the ANSYS finite element software package have been overviewed. It seems that their use could easily provide good supplementary information to reinforce concepts in a range of undergraduate engineering courses. No FEA background is required on the part of the user or instructor. The macros can be executed by a student with no ANSYS background by entering a single command at the ANSYS command line. Students can vary system parameters and observe the effect on calculated system responses through high level graphical representations. The postprocessing commands required to generate the results plots are also programmed into macros.

References

1. Wosu, S., Lovell, M., and Goldbach, R., "Pitt Engineering Career Access Program: Building a Pipeline for Success through Project CARE", Proceedings of the *2004 ASEE Annual Conference & Exposition,* American Society for Engineering Education, (2004).

2. Leake, J.M., "Development of an Advanced Course in Computer-Aided Design, Analysis, and Prototyping", Proceedings of the *2004 ASEE Annual Conference & Exposition,* American Society for Engineering Education, (2004).

3. Chattopadhyay, S., "Teaching of Design in Various Academic Settings", Proceedings of the *2004 ASEE Annual Conference & Exposition,* American Society for Engineering Education, (2004).

4. Hagigat, C., "Using Finite Element Software for Enhancing a Vibration Analysis Curriculum", Proceedings of the *2004 ASEE Annual Conference & Exposition,* American Society for Engineering Education, (2004).

5. Graham, L., and Khan, M., "Use of ANSYS and MATLAB in an Introductory Finite Element Course", *Computers in Education Journal*, Vol. XII, No. 1, pp. 32-37, (2002).

6. Baker, J., Capece, V., and Lee, R., "Integration of Finite Element Software in Undergraduate Engineering Courses", *Computers in Education Journal*, Vol. XII, No. 2, pp. 10-19, (2002).

Biographical Information

John R. Baker is an Assistant Professor of Mechanical Engineering at the University of Kentucky Extended Campus Program in Paducah, KY. He received his B.S., M.S., and Ph.D. in Mechanical Engineering from the University of Kentucky in Lexington, KY. After obtaining his B.S., he spent three years working in the Plastics Division of Eastman Chemical Products, Inc. He entered his current position in July 2000.
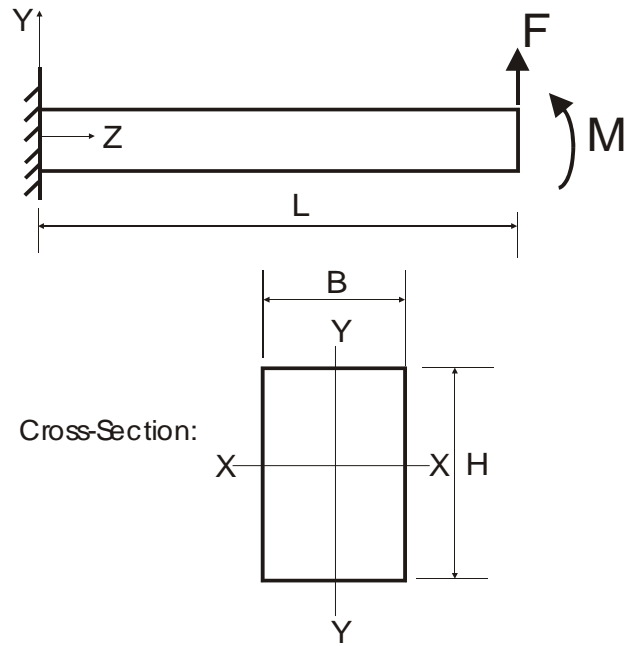
Figure 1: Parameters for the use with the macro
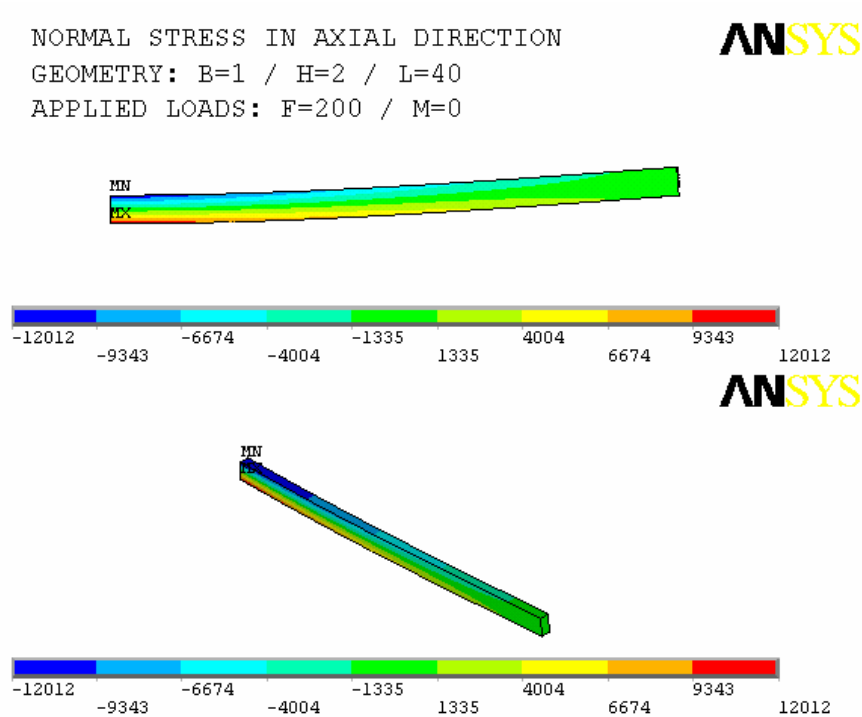*rbeam.mac*.

Figure 2: Stress contour plots resulting from executing the macro *rbeam.mac* with a lateral
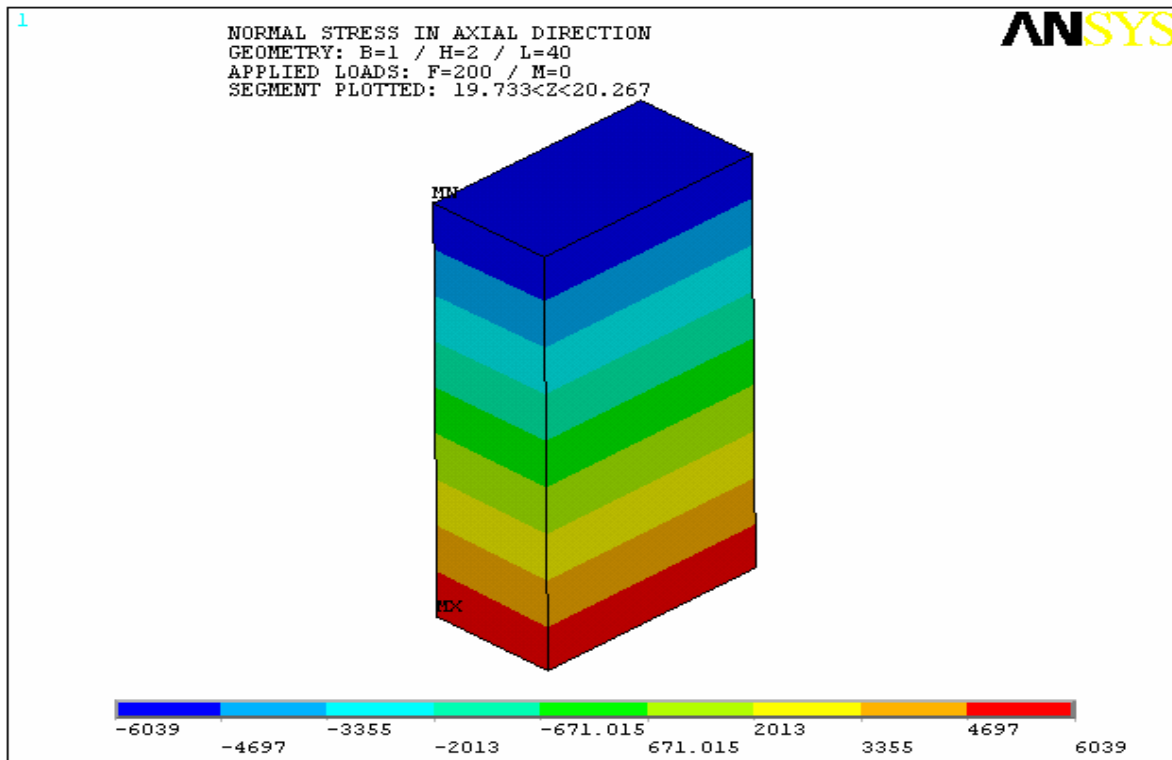end force, but no moment.

Figure 3: Normal stress plot on a segment of the rectangular beam automatically generated with the command: *beampost,1,20*.
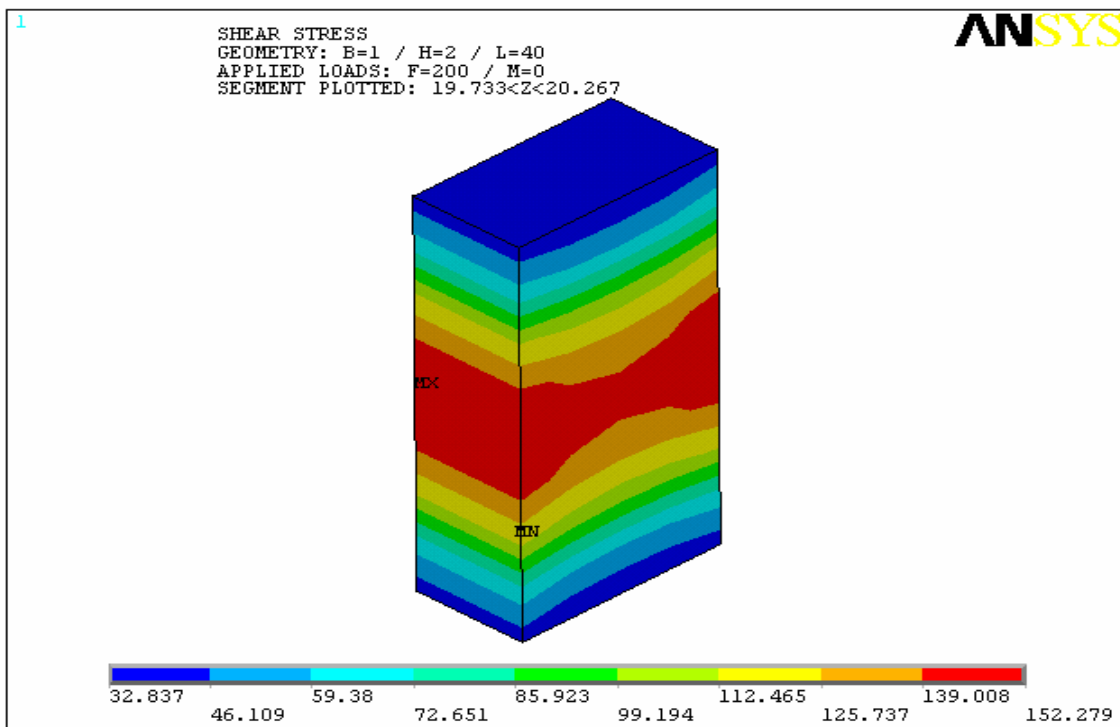

Figure 4: Shear stress plot on a segment of the beam automatically generated with the command: *beampost,2,20*.
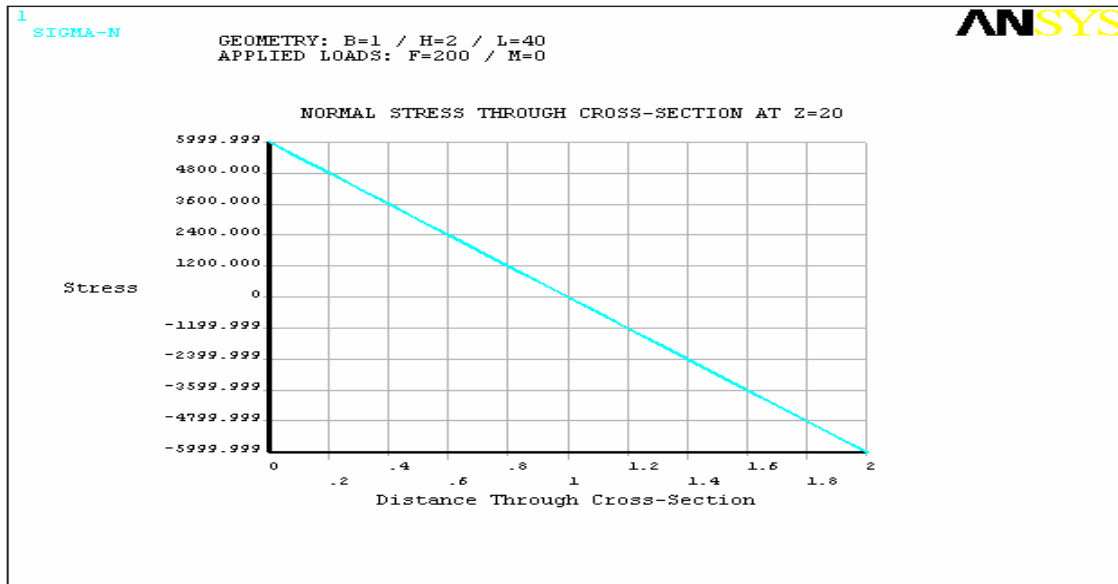
Figure 5: Plot of normal stress as a function of vertical location in cross-section at a specific axial location (z=20 inches) automatically generated with the command: *beampost,3,20*.
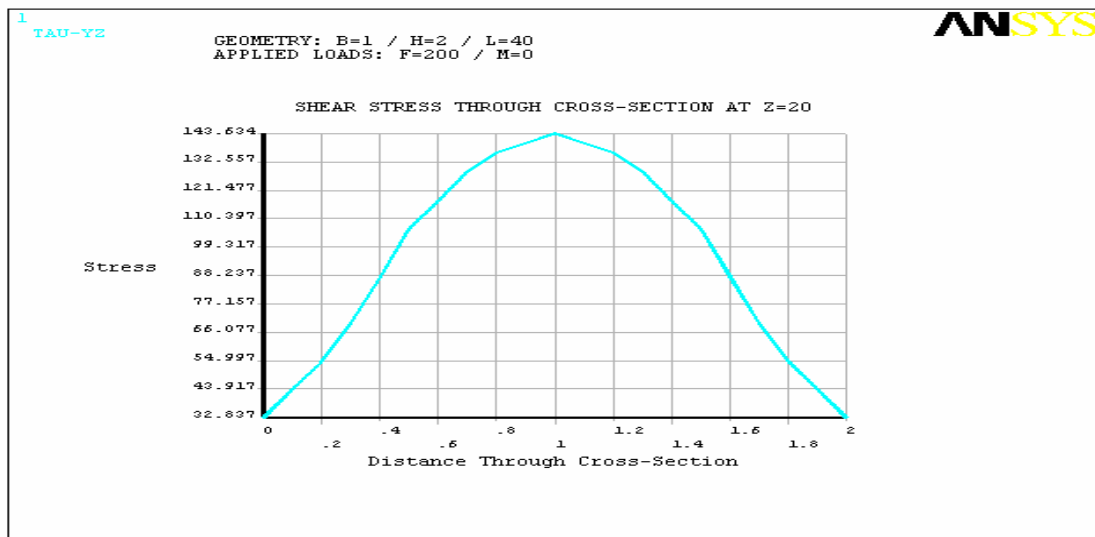

Figure 6: Plot of shear stress as a function of vertical location in cross-section at a specific axial location (z=20 inches) automatically generated with the command: *beampost,4,20*.
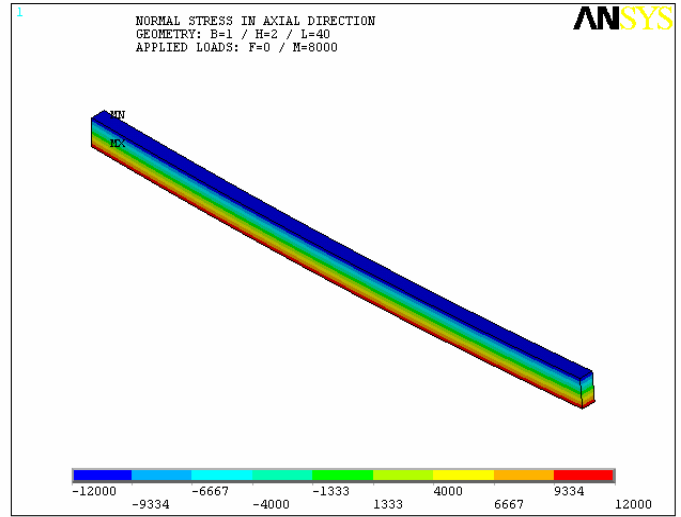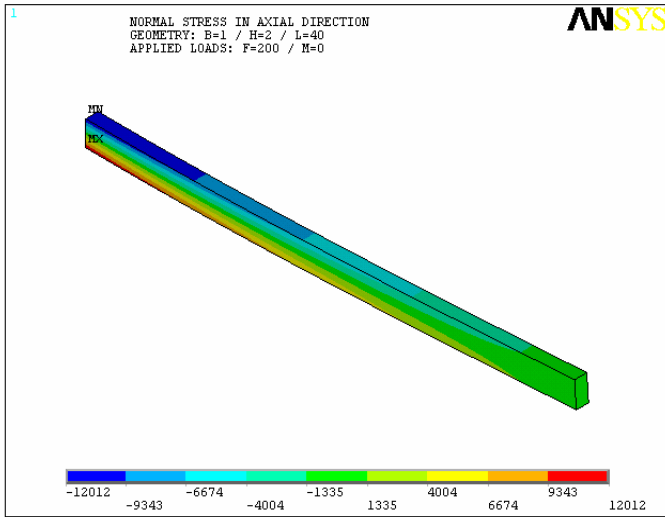
Figure 7: Contour plots of normal stress in axial direction automatically generated with the command: *beampost,1*. The case on the left is for a lateral end force and no additional applied moment, and the case on the right is for a pure bending moment, and no additional lateral end force.
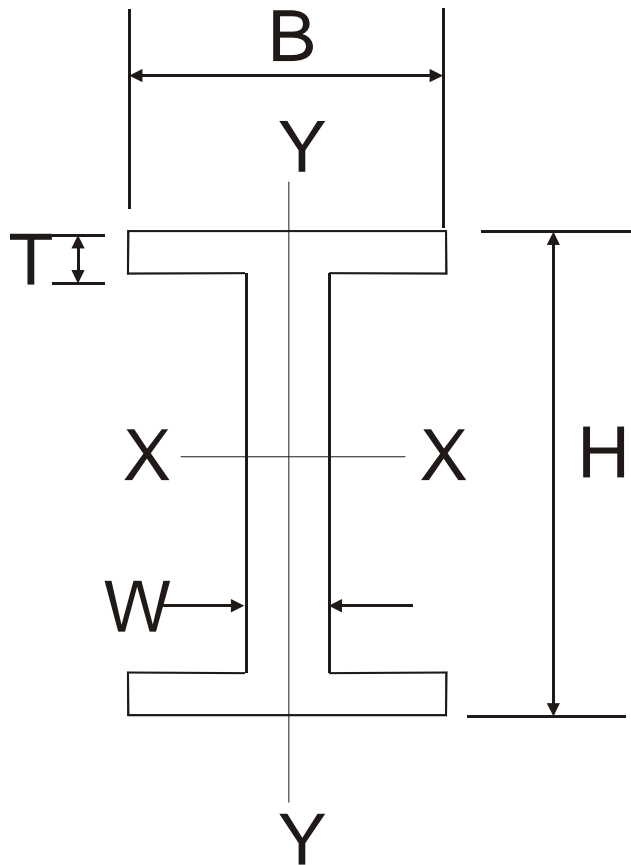

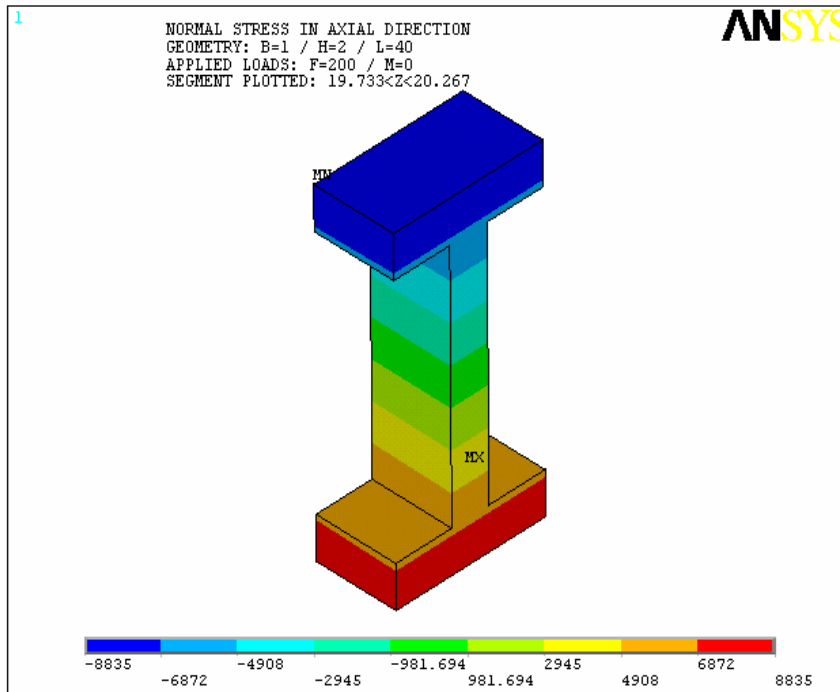Figure 8: Parameters for the use with the macro *ibeam.mac*.

Figure 9: Normal stress plot on a segment of the I-beam automatically generated with the command: *beampost,1,20*. The ibeam model was generated, and the solution was calculated, using the macro *ibeam.mac*.
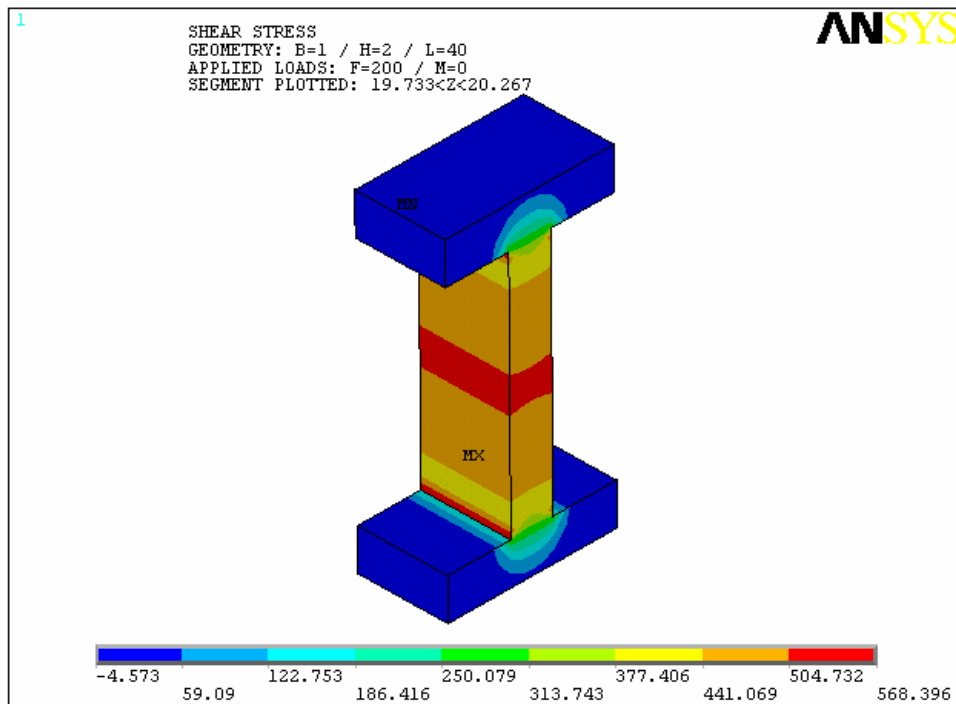

Figure 10: Shear stress plot on a segment of the I-beam automatically generated with the command: *beampost,2,20*. The ibeam model was generated, and the solution was calculated, using the macro *ibeam.mac*.
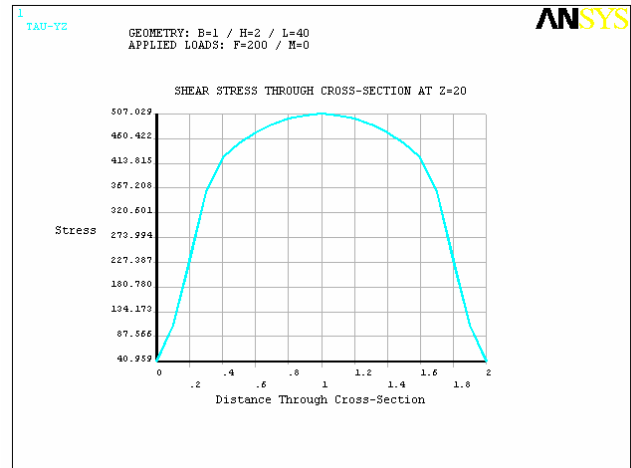
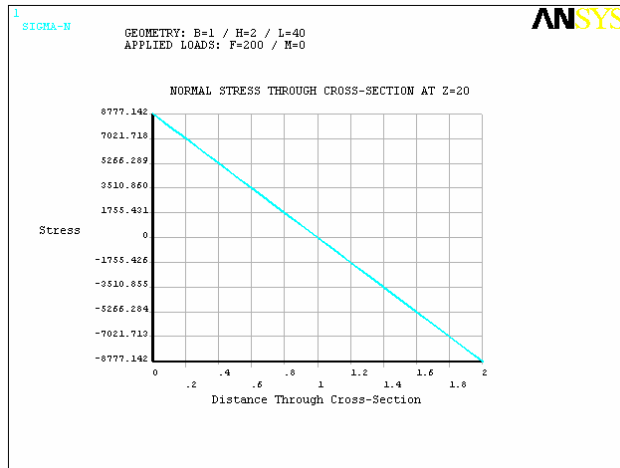Figure 11: At left, a plot of normal stress as a function of vertical location in cross-section of the I-beam, and at right, a plot of shear stress as a function of vertical location in the cross-section of the I-beam at the same axial location. These plots were automatically generated with the macro *beampost.mac*.
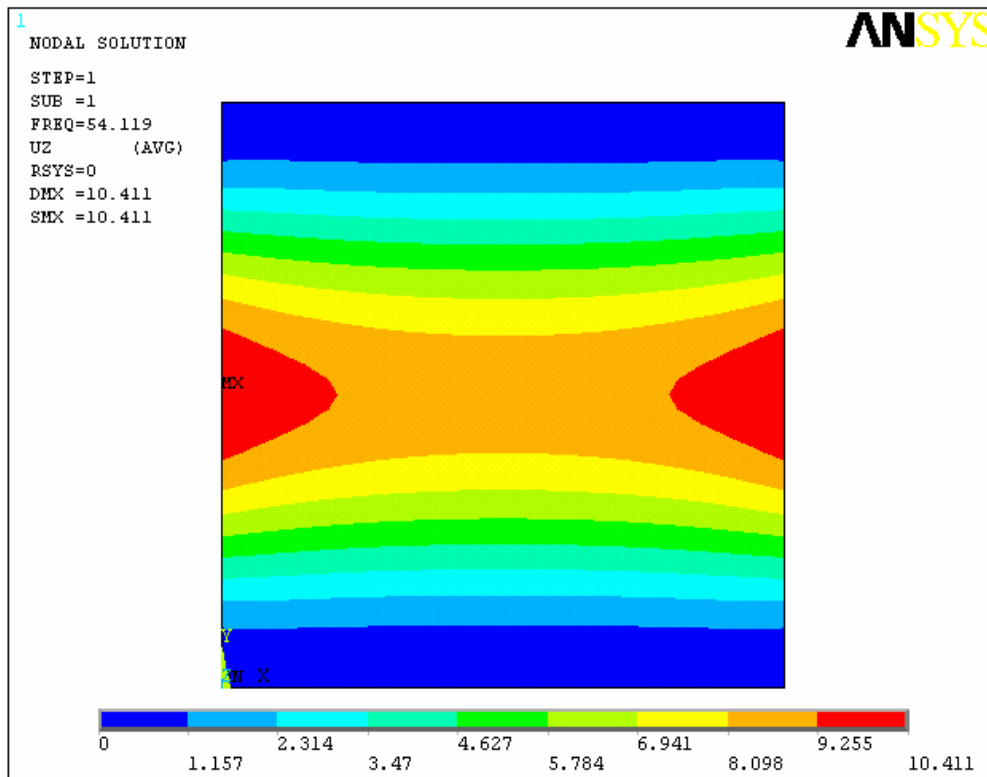

Figure 12: Mode shape plot generated with the macro *platevib.mac*, listed in Appendix C.

```
 *****   INDEX OF DATA SETS ON RESULTS FILE   *****

  SET    TIME/FREQ     LOAD STEP    SUBSTEP   CUMULATIVE
   1   54.119            1            1           1
   2   64.483            1            2           1
   3   106.43            1            3           1
   4   149.36            1            4           1
   5   164.05            1            5           1
   6   194.80            1            6           1
   7   213.87            1            7           1
   8   293.23            1            8           1
```

Figure 13: Natural frequency listing generated in the file "plateout.txt"
with the macro *platevib.mac*, which is listed in Appendix C.

```
ivel=arg1                          /post26
theta=arg2                         nsol,2,1,u,x
m=arg3                             nsol,3,1,u,y
g=arg4                             xvar,2
t=arg5                             /axlab,x,X
*afun,deg                          /axlab,y,Y
/prep7                             plvar,3
et,1,21                            lines,10000
r,1,m,m,m                          /output,trajectory,txt
n,1                                prvar,2,3
e,1                                /output
/solu
antype,trans
outres,all,all
acel,,g
ic,1,ux,,ivel*cos(theta)
ic,1,uy,,ivel*sin(theta)
time,t
deltim,t/100,t/100,t/100
solve
```
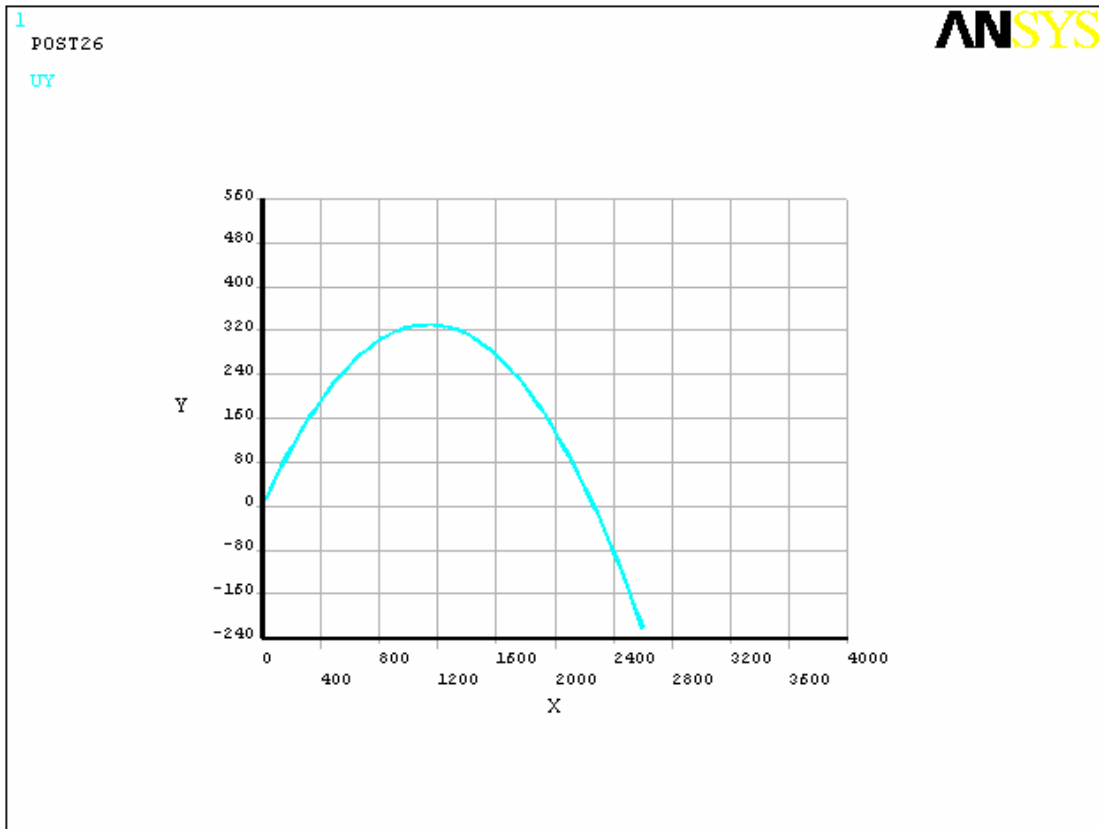
Figure 14: Contents of the macro *project.mac*.

Figure 15: Projectile trajectory plot generated automatically in ANSYS using the macro project.mac, listed in Figure 14.

```
B=arg1
H=arg2
L=arg3
F=arg4
M=arg5
Modulus=arg6
Poisson=arg7
/prep7
et,1,Solid45
mp,ex,1,Modulus
mp,prxy,1,Poisson
block,-B/2,0,0,H/2,0,L
block,0,B/2,0,H/2,0,L
block,-B/2,0,H/2,H,0,L
block,0,B/2,H/2,H,0,L
nummrg,all
hdiv=nint(75*H/L)
bdiv=nint(75*B/L)
ksel,s,loc,x,-B/2
ksel,r,loc,y,0
lslk,s,1
lesize,all,,,150
ksel,s,loc,z,L
ksel,r,loc,y,0
lslk,s,1
lesize,all,,,bdiv
ksel,s,loc,z,L
ksel,r,loc,x,-B/2
lslk,s,1
lesize,all,,,hdiv
vmesh,all
nsel,s,loc,z,L
nsel,u,loc,x,-B/2
nsel,u,loc,x,B/2
nsel,u,loc,y,0
nsel,u,loc,y,H
*get,nn1,node,0,count
cm,set1,node
nsel,s,loc,z,L
cmsel,u,set1
*get,nn2,node,0,count
fvalue=F/(nn1+(0.5*(nn2-4))+1)
cmsel,s,set1
f,all,fy,fvalue
nsel,s,loc,z,L
cmsel,u,set1
```

```
f,all,fy,0.5*fvalue
nsel,s,loc,z,L
nsel,r,loc,x,-B/2
nsel,r,loc,y,0
f,all,fy,0.25*fvalue
nsel,s,loc,z,L
nsel,r,loc,x,B/2
nsel,r,loc,y,0
f,all,fy,0.25*fvalue
nsel,s,loc,z,L
nsel,r,loc,x,-B/2
nsel,r,loc,y,H
f,all,fy,0.25*fvalue
nsel,s,loc,z,L
nsel,r,loc,x,B/2
nsel,r,loc,y,H
f,all,fy,0.25*fvalue
allsel
nsel,s,loc,z,L
sfgrad,pres,,y,H/2,12*M/(B*(H**3))
sf,all,pres,0
nsel,s,loc,z,0
d,all,uz,0
nsel,r,loc,y,H/2
d,all,uy,0
nsel,r,loc,x,0
d,all,ux,0
allsel
/solu
solve
save,beam,db
/post1
/window,1,top
/window,2,bot
/view,1,-1
/view,2,-1,1,1
/plopts,leg1,off
/plopts,date,off
/plopts,frame,off
/triad,off
/tspec,,1.5,1.5
/tlabel,-.7,0.9,NORMAL    STRESS    IN
AXIAL DIRECTION
/tlabel,-.7,0.8,GEOMETRY:    B=%B%   /
H=%H% / L=%L%
/tlabel,-.7,0.7,APPLIED   LOADS:   F=%F%
/ M=%M%
plns,s,z
```

```
resume,beam,db
posttype=arg1
z=arg2
*if,z,gt,0,then
  zloc=nint(150*z/L)
  zlo=(zloc-1)*L/150
  zhi=(zloc+1)*L/150
*endif
*if,posttype,gt,2,then
  *if,z,gt,L,then
    *msg,error
    Axial location entered is greater than length of model.
  *endif
*endif
/post1
/plopts,leg1,off
/plopts,date,off
/triad,off
/tspec,,1,1
/tlabel,-.5,0.85,GEOMETRY: B=%B% / H=%H% / L=%L%
/tlabel,-.5,0.8,APPLIED LOADS: F=%F% / M=%M%
*if,posttype,eq,1,then
  /tlabel,-.5,0.9,NORMAL STRESS IN AXIAL DIRECTION
  *if,z,gt,0,then
    nsel,s,loc,z,zlo,zhi
    esln,s,1
    /tlabel,-.5,0.75,SEGMENT PLOTTED: %zlo%<Z<%zhi%
  *endif
  /view,1,-1,1,1
  plns,s,z
*elseif,posttype,eq,2,then
  /tlabel,-.5,0.9,SHEAR STRESS
  *if,z,gt,0,then
    nsel,s,loc,z,zlo,zhi
    esln,s,1
    /tlabel,-.5,0.75,SEGMENT PLOTTED: %zlo%<Z<%zhi%
  *endif
  /view,1,-1,1,1
  plns,s,yz
.
.
.
.
.
.
*else
  /axlab,x,Distance Through Cross-Section
  /axlab,y,Stress
  nsel,s,loc,x,0
  nsel,r,loc,y,0
  nsel,r,loc,z,zloc*L/150
  *get,n1,node,0,num,max
  nsel,s,loc,x,0
  nsel,r,loc,y,H
  nsel,r,loc,z,zloc*L/150
  *get,n2,node,0,num,max
  allsel
  *if,posttype,eq,3,then
    path,sigma-n,2
    ppath,1,n1
    ppath,2,n2
    pdef,sigma-n,s,z
    /tlabel,-.3,0.6,NORMAL STRESS THROUGH CROSS-SECTION AT Z=%z%
    plpath,sigma-n
  *else
```

```
            path,tau-yz,2
            ppath,1,n1
            ppath,2,n2
            pdef,tau-yz,s,yz
            /tlabel,-.3,0.6,SHEAR STRESS THROUGH CROSS-SECTION AT Z=%z%
            plpath,tau-yz
          *endif
          /output,beamstress,txt
          prpath,sigma-n,tau-yz
          /output
        *endif
```

<u>Appendix C</u>: Listing of the macro *platevib.mac*.

```
/filnam,platevib                    /solu
/prep7                              antype,modal
w=arg1                              modopt,lanb,8,0,100000
L=arg2                              *if,bc,eq,1,then
tk=arg3                               nsel,s,loc,y,0
modulus=arg4                          d,all,all
density=arg5                          nsel,s,loc,y,L
poissons=arg6                         d,all,all
bc=arg7                               allsel
et,1,63                             *endif
r,1,tk,tk,tk,tk                     solve
ex,1,modulus                        /post1
dens,1,density                      /output,plateout,txt
prxy,1,poissons                     set,list
k,1                                 *if,bc,eq,0,then
k,2,w                                 set,1,7
k,3,w,L                               prdi
k,4,0,L                               set,1,8
a,1,2,3,4                             prdi
ksel,s,kp,,1,2                        /output
lslk,s,1                              set,1,7
lesize,all,,,40                       plns,u,z
ksel,s,kp,,2,3                      *else
lslk,s,1                              set,1,1
lesize,all,,,40                       plns,u,z
mshkey,1                              prdi
amesh,all                            set,1,2
                                      prdi
                                      /output
                                      set,1,1
                                      plns,u,z
                                    *endif
```