

# Finite Element Method Course Assignments

Mingpei Cang, Student ID: 201775010

**Abstract.** This coursework solved a beam element problem and a 2D plane problem. For Assignment 1, I selected two kinds of beam element to find the solution: BEAM188 and BEAM189, corresponding to two-node beam element and three-node beam element, respectively.

As for the 2D plane stress problem, two elements types - PLANE182 and PLANE183 - were considered, which were different in result accuracy because the number of nodes in the latter element is twice that in the former. Moreover, two kinds of mesh methods, triangular and rectangular, were conducted in order to make comparisons. Therefore, 4 different FE models were discussed in this plane stress problem. What is more, the distribution of principal strains at integration points was also plotted in this section.

## 1 Beam Element Problem

In the beam, given the material = Steel, Length  $L = 1.0$  m, rectangular cross-section  $B = 4.0$  cm,  $H = 10.0$  cm, a displacement  $U = 30.0$  cm is applied in y-direction at the end of the beam, as shown in figure 1.

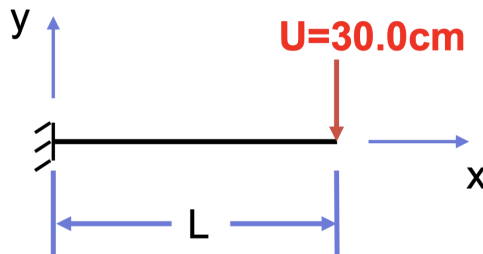


Figure 1. Beam Problem Description

### Tasks:

1. Use the displacement and strain at the point  $(\frac{L}{2}, 0)$  to perform a mesh convergence study, plot the curves of displacement/strain vs. element number.
2. Output the displacement, strain and stress distributions over the entire beam.

Because the displacement at the free end of the beam is too large compared to beam length, it is a non-linear geometry problem.

### 1.1 Modelling Process

In this subsection, a detailed modelling process was presented to help readers reconstruct the result.

#### 1.1.1 Preprocessing

I selected BEAM188 and BEAM 189 separately as the element type for this assignment. According to the ANSYS Help document, BEAM188 is a 2-node beam element and BEAM189 is a 3-node beam element. That is to say, BEAM189 has higher accuracy than the other one.

As for material, I chose the elastic modulus and Poisson's ratio of 25Mn, which is  $2.1 \times 10^{11}$  Pa and 0.28, respectively. Density is not a necessary material property in this problem.

As is shown at the start of this section, the beam section is a rectangular area, with width equals to 4 cm and height equals to 10cm. SECTYPE and SECDATA commands were used here.

Three keypoints were defined in this problem, the first two stand for the left and right ends of this beam, while the third keypoint helps ANSYS to determine the orientation of the beam cross-sections during beam meshing. The coordinates of these keypoints are (1, 0, 0), (0, 1, 0) and (0.5, 0.5, 0).

Then the LATT command was used here to associate predefined element type, material, cross-section with line 12.

```
LATT,1,,1,,3,1
```

I used a \*DO loop to output results of different mesh numbers from 2 to 60, with step length 2. The result will be output in a TXT file.

#### 1.1.2 Solve

Since 30cm is not a small displacement compared to the 1-meter length, a non-linear geometry problem should be considered. In other words, NLGEOM should be set to 'ON'.

Two constraints were added to keypoint 1 and keypoint 2 with DK command, which restricted all degree of freedom of keypoint 1 and  $Y = -0.3$  of keypoint 2.

### 1.1.3 Postprocessing

Firstly, 'Display Element' (/ESHAPE) was turned on and nodal displacement contour of Y-direction was plotted using PLNSOL.

Then, I used \*GET command to get midpoint's Y-direction displacement:

```
*GET,UY,NODE,NMESH+2,U,Y
```

Where NMESH is the node number of middle nodes. The rule of numbering can be described as: The beams' left node and right node numbers are 1 and 2 respectively, then node number increases along the x-axis from the second left node, which is numbered node 3.

The strain information cannot be reached simply using \*GET command because both BEAM188 and BEAM189 plot nodal values only at the corner, and ANSYS takes an average of the corner nodal values when the result is exported, while they are not stored since they are not explicitly calculated when I trying to get those values with APDL commands<sup>1</sup>.

After reviewing BEAM188 description in ANSYS Help, I found generalised strains via ETABLE or ESOL commands: generalised strains are stored in SMISC, with I and J corresponding to the left and right node number of the current element. Therefore, I extracted the right node strain of the midpoint's left element and the left strain of the midpoint's right element.

```
ETABLE, RightEx , SMISC, 20 ,AVG  
ETABLE, LeftEx , SMISC, 7 ,AVG
```

Then I can use the average of those two strains as the strain of the middle point.

## 1.2 Result

### 1.2.1 Convergence Study at Middle Point

The relationship between the number of elements and displacement/strain of the midpoint are shown as follows. Obviously, with the increase in the number of elements, the result is becoming more stable, that is to say, the solution is converged.

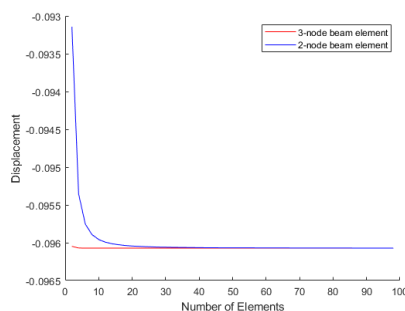


Figure 2. Displacement of the midpoint

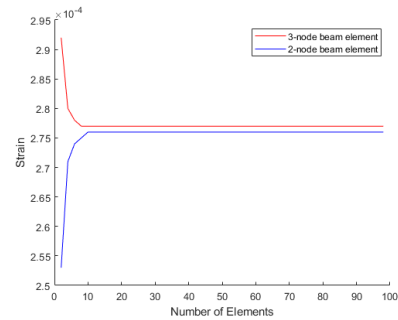


Figure 3. Strain of the midpoint

Interestingly, it is shown in Figure 3 that the strain is a decreasing function of the number of elements in 2-node element case, while it changes to an increasing function when 3-node beam element is applied to the model.

### 1.2.2 Contour Plot Results

The distributions of displacement, strain and stress over the beam are shown below.

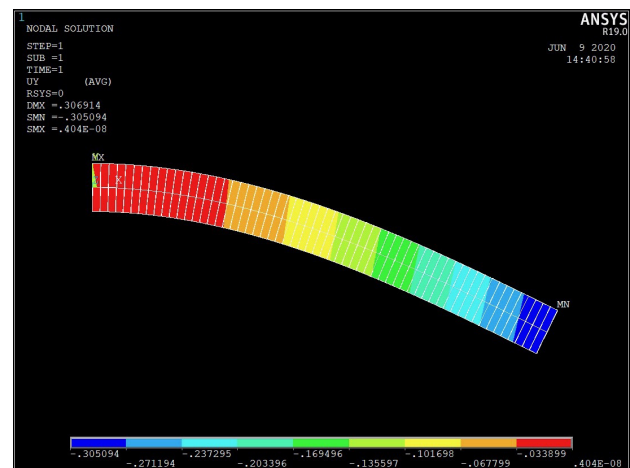


Figure 4. Distribution of Nodal Displacement

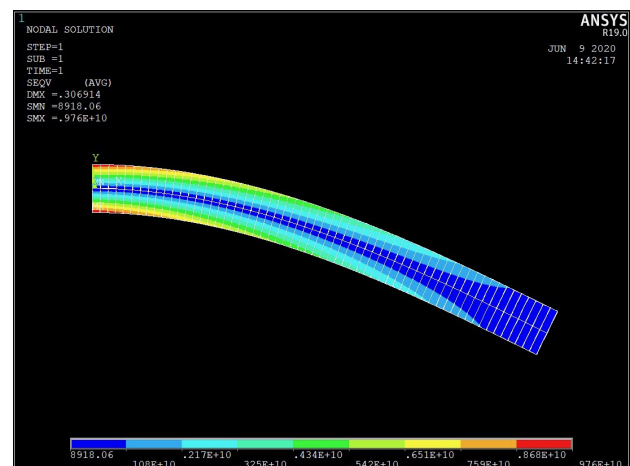
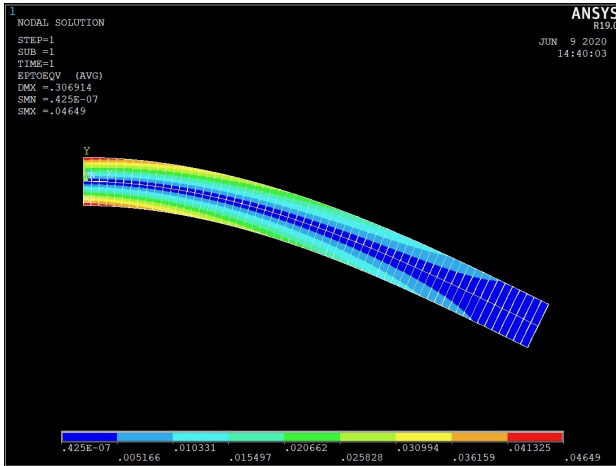


Figure 5. Distribution of Equivalent Stress

<sup>1</sup><https://studentcommunity.ansys.com/thread/nodal-data-not-stored/>



**Figure 6.** Distribution of Equivalent Strain

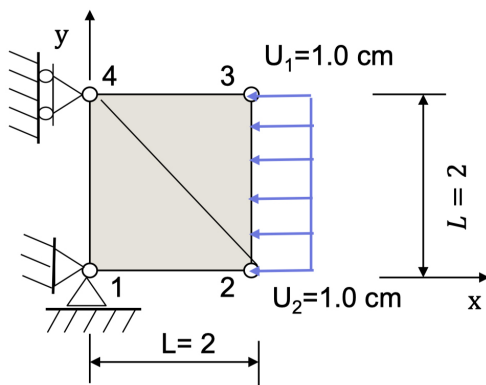
At the moment, the number of elements equals to 60, which means the result has converged according to 1.2.1, so it is believed that the contour plots above are relatively precise enough to present the result.

### 1.3 Short Conclusion

In this section, a thorough process of modelling procedure was presented in order to help readers reconstruct the model. After that, a comparison of two beam element types, BEAM188 and BEAM189, was conducted, and a mesh convergence study was performed. In the end, output the displacement, strain and stress distributions over the entire beam were shown.

## 2 Plane Problem

Given: Material = Steel, thickness  $t = 1.0$ , length  $L = 2.0$  m. A displacement loading was applied along edge 23, as shown in figure 7.



**Figure 7.** 2-D Plane Problem Description

### Tasks:

1. Use the displacement and strain at the point  $(\frac{L}{2}, 0)$  to perform a mesh convergence study, plot the curves of displacement/strain vs. element number.

2. Output the displacement, strain and stress distributions over the entire beam.
3. Output the principal strains at the elemental integration points.
4. Output the distributions of elemental strain and stress over the entire structure.

Because the displacement at the free end of the beam is too large compared to beam length, it is a non-linear geometry problem.

## 2.1 Modelling Process

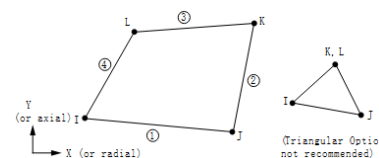
### 2.1.1 Modelling

The modelling process is almost same as Assignment 1 except it is a plane stress problem instead. Therefore, A command is used to define the area by connecting key-points.

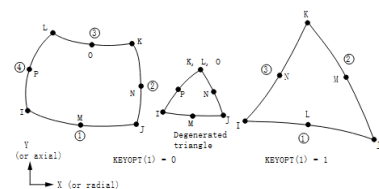
### 2.1.2 Meshing

I used 4 meshing methods to solve this model separately: PLANE182 with rectangular elements, PLANE183 with rectangular elements, PLANE182 with triangular elements and PLANE183 with triangular elements.

The difference between PLANE182 and PLANE183 is like the difference between BEAM188 and BEAM189 in the previous section. PLANE183 is a higher-order element type compared to PLANE182. Therefore, the solution is more accurate theoretically, so do rectangular elements compared to triangular elements.



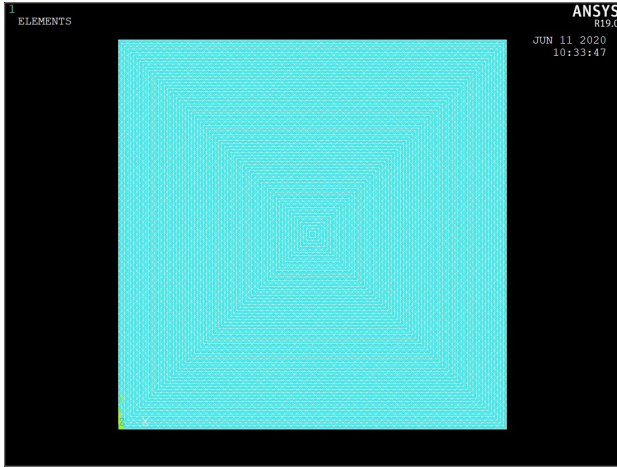
**Figure 8.** Element type PLANE182



**Figure 9.** Element type PLANE183

When a triangular element is needed, MSHAPE command should be set to 1. Moreover, in order to be sure that there was always a node in the centre of the plane, which should be at (1, 1), element size was set to 1 at first, resulting in a 4-element FE model. After that, EREFINE command was used to refine the mesh result. The more time I refined, the more triangular element I can obtain

with a node in the centre of the model, as is shown in the figure below.



**Figure 10.** 2-D plane problem with 16384 triangular elements

### 2.1.3 Solve

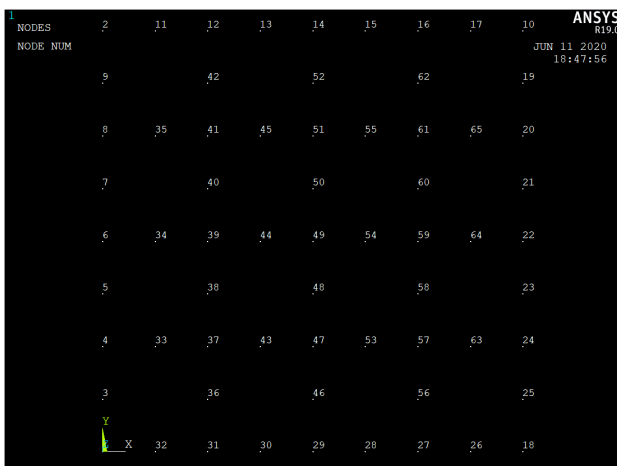
Non-linear geometry should be turned on to solve this problem.

### 2.1.4 Postprocessing

The midpoint's number could be easily obtained in triangular element cases because it remained unchanged when EREFINE command was used. However, in rectangular cases, midpoint's number changed as the number of elements increased. I successfully found the patterns of it by polynomial interpolation.

$$n_{mid} = \frac{3n^2}{2} + 6n + 1$$

Where  $n_{mid}$  is midpoint's number,  $n$  is the total number of elements. For example, midpoint's number is 49 when there are 16 PLANE183 elements in the FE model, as is shown in Figure 11.



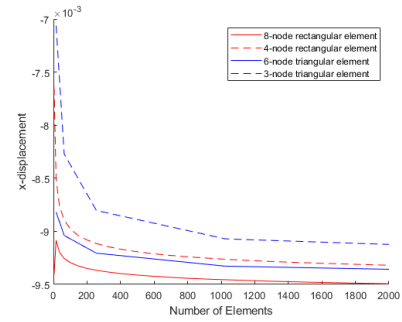
**Figure 11.** Node number when number of mesh is 4

## 2.2 Result

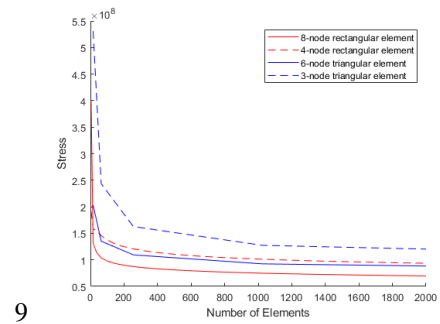
Nodal displacement and equivalent strain/stress of the midpoint were exported from 4 to 40000 elements.

### 2.2.1 Convergence Study at Middle Point

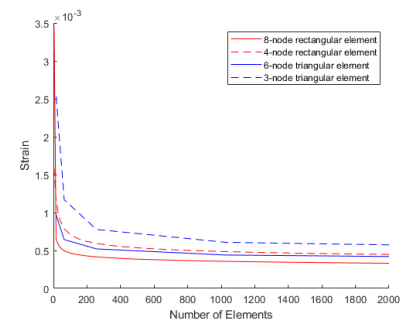
A convergence study was performed here to find the difference between PLANE182 and PLANE183, as well as triangular elements and rectangular elements.



**Figure 12.** Comparison of Nodal Displacement



**Figure 13.** Comparison of Equivalent Stress



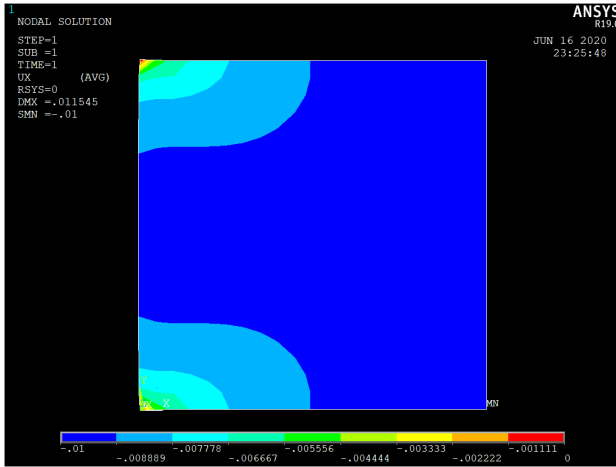
**Figure 14.** Comparison of Equivalent Strain

As is shown in Figure 12, Figure 13 and Figure 14, the most accurate result is the 8-node rectangular element case, because the displacement in x direction should be close to  $1 \times 10^{-2}$ m.

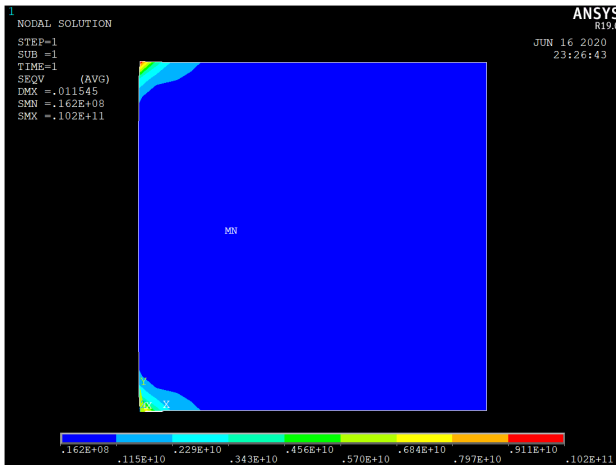
The result of 3-node triangular element case is most inaccurate and it converged slowly compared to the other three cases.

### 2.2.2 Contour Plot Results

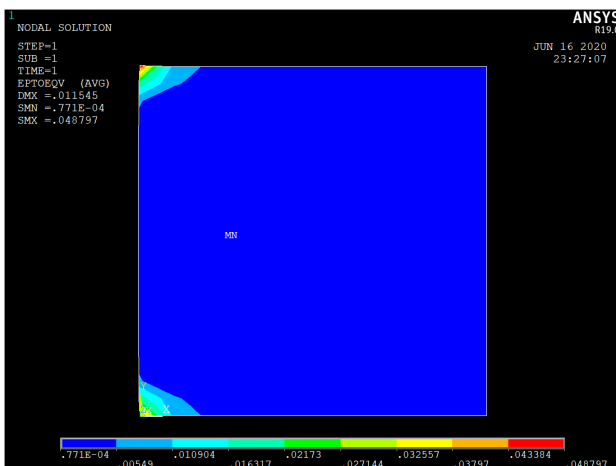
The distributions of displacement, strain and stress over the beam are shown in figures below.



**Figure 15.** Distribution of Nodal Displacement



**Figure 16.** Distribution of Equivalent Stress



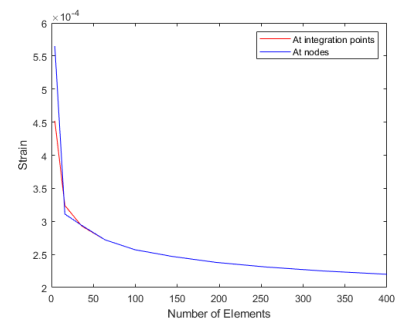
**Figure 17.** Distribution of Equivalent Strain

It can be seen that the strain and stress are almost uniformly distributed across the plane except for its upper and lower-left corner where constraints are added. This is consistent with the theoretical distribution.

### 2.3 Principal strains at integration points

In order to get values of integration points, we need to copy the integration point results to the nodes instead of extrapolating the linear portion of the integration point results to the nodes. In other words, ERESX command should be set to 'NO'.

The result is shown in table 1 with comparison to interpolated value.



**Figure 18.** Midpoint's strain at integration points

**Table 1.** Principal Strains at integration points

$N_{elem}$	At integration points	At nodes
4	0.000452	0.000565
16	0.000324	0.000311
36	0.000293	0.000294
64	0.000272	0.000272
100	0.000257	0.000257
144	0.000247	0.000247
196	0.000238	0.000238
256	0.000231	0.000231
324	0.000225	0.000225
400	0.000220	0.000220

As we can see, with the increase of element numbers, the difference between strain at nodes and integration points are becoming smaller. I suppose the reason is that the distance between nodes and integration points are smaller.

### 2.4 Short Conclusion

In this section, I mainly compared the difference between triangular element and rectangular element and the difference between PLANE182 and PLANE183. The midpoint's principal strains at elemental integration points are extracted to find the dissimilarity with nodal solutions acquired from interpolation.

### 3 Conclusion

I began studying and using CAE software like Abaqus in early 2019, and I successfully wrote some computational structural mechanics programs last semester, but I have to admit that it was until I had this FEM course that I somehow understand how it works behind GUI.

At first, I planned to use Abaqus to finish this coursework, because not only I have experiences in Abaqus, but also Python is my favourite programming language. What is more, APDL seems like some kind of assembly language, which should be difficult to get started for me. However, after searching for more information online about APDL, I found that it is really useful and this is a good chance for me to learn it. That is why I choose

APDL to finish this coursework. In the past two weeks, it proved that I was right. APDL macro is easy to code with the help of ANSYS user guide, and it is really concise and clean, compared to Abaqus Python macro.

In addition, this is the first time for me to completely use English to finish a coursework report. It is not like the writing parts in IELTS or CET exams. In the report, I have to try my best to make myself clear. Thankfully, it is not as difficult as I thought at the beginning.

In the end, I want to express my appreciation to my teacher, Mr Lyu Yongtao, and teaching assistant Yang Zhuoyue. Although this course was conducted online in this special semester, I still had a good learning experience.