FEA Homework 4

March 14, 2022

Gabe Morris

[1]:

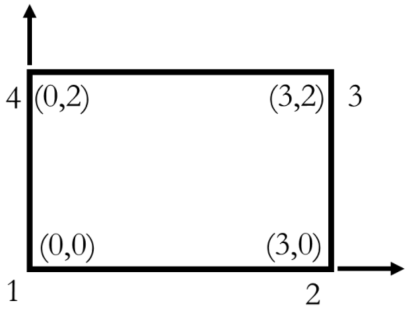
```python
# Notebook Preamble
%config InteractiveShell.ast_node_interactivity='all'
import matplotlib.pyplot as plt
import numpy as np
import sympy as sp
import pandas as pd
from IPython.display import display, Latex
import warnings

warnings.filterwarnings('ignore', category=FutureWarning)

plt.style.use('maroon_ipynb.mplstyle')
```

Contents

ME 6233 Homework 4 Gabe Morris
gnm54

1 Problem 1

1.1 Given



For a rectangular element, the displacements at four nodes are given by

$$u_1 = 0$$

$$v_1 = 0$$

$$u_2 = -0.5$$

$$v_2 = -0.5$$

$$u_3 = 0.75$$

$$v_3 = 1.25$$

$$u_4 = 0.5$$

$$v_4 = 1$$

1.2 Find

a. Calculate the displacement (u, v) at point $(x, y) = (0.7, 1.3)$.
b. Calculate the strain ϵ_{xx} at point $(x, y) = (0.7, 1.3)$

1.3 Solution

From Fig. 3.4-1 in the text, $a = 1.5$ and $b = 1$. These are the center to edge dimensions of the rectangle and are used in the shape functions/strain displacement matrix.

The analysis of the bilinear quadrilateral depends on the origin being at the center of the rectangle. Therefore, the point $(0.7, 1.3)$ relative to the origin at the center is $(0.7, 1.3) - (a, b) = (-0.8, 0.3)$.

1.3.1 Part A

The displacements u, v can be defined as the dot product between the shape function and the displacement vectors

$$u(x, y) = \vec{N} \cdot \vec{u}$$
$$v(x, y) = \vec{N} \cdot \vec{v}$$

where the shape function in the vector form is,

$$N = \begin{bmatrix} \frac{(a-x)(b-y)}{4ab} \\ \frac{(a+x)(b-y)}{4ab} \\ \frac{(a+x)(b+y)}{4ab} \\ \frac{(a-x)(b+y)}{4ab} \end{bmatrix}$$

and $\vec{u}$ and $\vec{v}$ are $\langle u_1, u_2, u_3, u_4 \rangle$ and $\langle v_1, v_2, v_3, v_4 \rangle$.

[2]:
```
# Define known parameters
# The underscore denotes a numerical value, while no underscore denotes a␣
↪symbol.
a_, b_ = 1.5, 1
x_, y_ = -0.8, 0.3
u_, v_ = [0, -0.5, 0.75, 0.5], [0, -0.5, 1.25, 1]
d_ = np.array(list(zip(u_, v_))).flatten()

# Define symbols
a, b, x, y = sp.symbols('a b x y')
u_vec, v_vec = sp.Matrix(u_), sp.Matrix(v_)
d_vec = sp.Matrix(d_)

# Shape function
N = 1/(4*a*b)*sp.Matrix([(a - x)*(b - y), (a + x)*(b - y), (a + x)*(b + y), (a␣
↪- x)*(b + y)])

sub = [(a, a_), (b, b_), (x, x_), (y, y_)]
N_ = N.subs(sub)

u = sp.DotProduct(N, u_vec)
v = sp.DotProduct(N, v_vec)
sp.Eq(u, N_.dot(u_vec), evaluate=False)
sp.Eq(v, N_.dot(v_vec), evaluate=False)
```

[2]:

$$\mathrm{DotProduct}\left(\begin{bmatrix} \frac{(a-x)(b-y)}{4ab} \\ \frac{(a+x)(b-y)}{4ab} \\ \frac{(a+x)(b+y)}{4ab} \\ \frac{(a-x)(b+y)}{4ab} \end{bmatrix}, \begin{bmatrix} 0 \\ -0.5 \\ 0.75 \\ 0.5 \end{bmatrix}\right) = 0.322083333333333$$

[2]:

$$\operatorname{DotProduct}\left(\left[\begin{matrix}\frac{(a-x)(b-y)}{4ab}\\ \frac{(a+x)(b-y)}{4ab}\\ \frac{(a+x)(b+y)}{4ab}\\ \frac{(a-x)(b+y)}{4ab}\end{matrix}\right], \left[\begin{matrix}0\\ -0.5\\ 1.25\\ 1\end{matrix}\right]\right) = 0.647083333333333$$

Thus, the displacements u, v at $(-0.8, 0.3)$ are $u = 0.322$ and $v = 0.647$.

1.3.2 Part B

The strain displacement matrix is

[3]:
```
# Getting B
B = 1/(4*a*b)*sp.Matrix([
    [-(b - y), 0, b - y, 0, b + y, 0, -(b + y), 0],
    [0, -(a - x), 0, -(a + x), 0, (a + x), 0, (a - x)],
    [-(a - x), -(b - y), -(a + x), b - y, a + x, b + y, a - x, -(b + y)]
])
sp.Eq(sp.Symbol('B'), B, evaluate=False)
```

[3]:

$$B = \left[\begin{matrix}\frac{-b+y}{4ab} & 0 & \frac{b-y}{4ab} & 0 & \frac{b+y}{4ab} & 0 & \frac{-b-y}{4ab} & 0\\ 0 & \frac{-a+x}{4ab} & 0 & \frac{-a-x}{4ab} & 0 & \frac{a+x}{4ab} & 0 & \frac{a-x}{4ab}\\ \frac{-a+x}{4ab} & \frac{-b+y}{4ab} & \frac{-a-x}{4ab} & \frac{b-y}{4ab} & \frac{a+x}{4ab} & \frac{b+y}{4ab} & \frac{a-x}{4ab} & \frac{-b-y}{4ab}\end{matrix}\right]$$

The strain may be calculated using $\epsilon = Bd$.

[4]:
```
Bd = sp.MatMul(B, d_vec)
Bd_doit = Bd.doit().subs(sub)
sp.Eq(Bd, Bd_doit, evaluate=False)
```

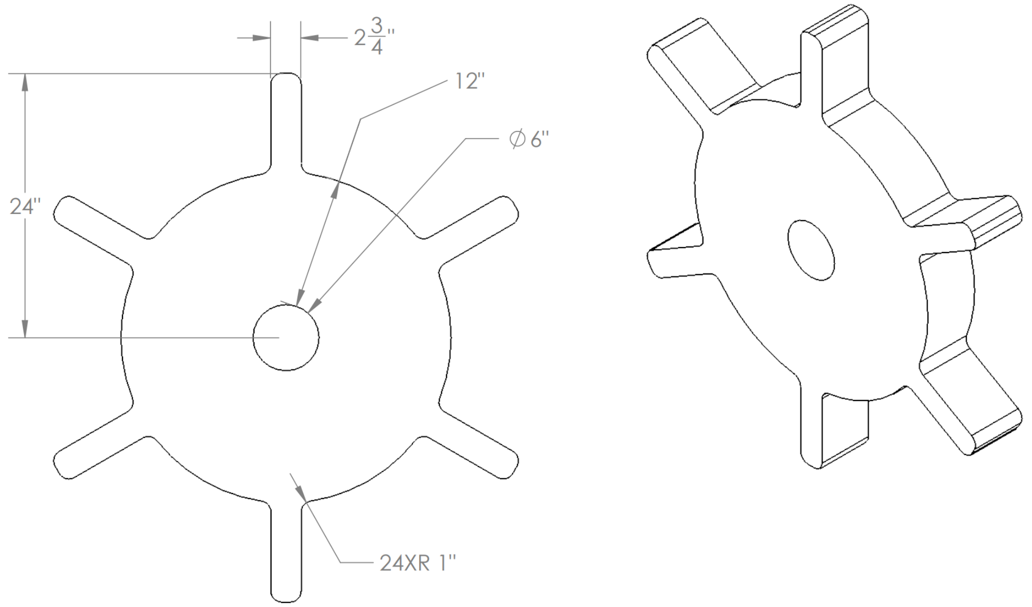
[4]:

$$\left[\begin{matrix}\frac{-b+y}{4ab} & 0 & \frac{b-y}{4ab} & 0 & \frac{b+y}{4ab} & 0 & \frac{-b-y}{4ab} & 0\\ 0 & \frac{-a+x}{4ab} & 0 & \frac{-a-x}{4ab} & 0 & \frac{a+x}{4ab} & 0 & \frac{a-x}{4ab}\\ \frac{-a+x}{4ab} & \frac{-b+y}{4ab} & \frac{-a-x}{4ab} & \frac{b-y}{4ab} & \frac{a+x}{4ab} & \frac{b+y}{4ab} & \frac{a-x}{4ab} & \frac{-b-y}{4ab}\end{matrix}\right]\left[\begin{matrix}0\\ 0\\ -0.5\\ -0.5\\ 0.75\\ 1.25\\ 0.5\\ 1.0\end{matrix}\right] = \left[\begin{matrix}-0.00416666666666666\\ 0.5875\\ 0.333333333333333\end{matrix}\right]$$

Thus, $\epsilon_{xx} = -0.00417$.

2 Problem 2

2.1 Given



A steel paddle wheel with a thickness of 6.0 inches is placed in a scenario in which the paddles are loaded normally in a counterclockwise pattern at a total force of 130 lbf on the bottommost paddle. The dimensions of the paddle wheel are given in the schematic. The properties for steel are ($E = 30 \cdot 10^6\,psi$; $\nu = 0.3$).

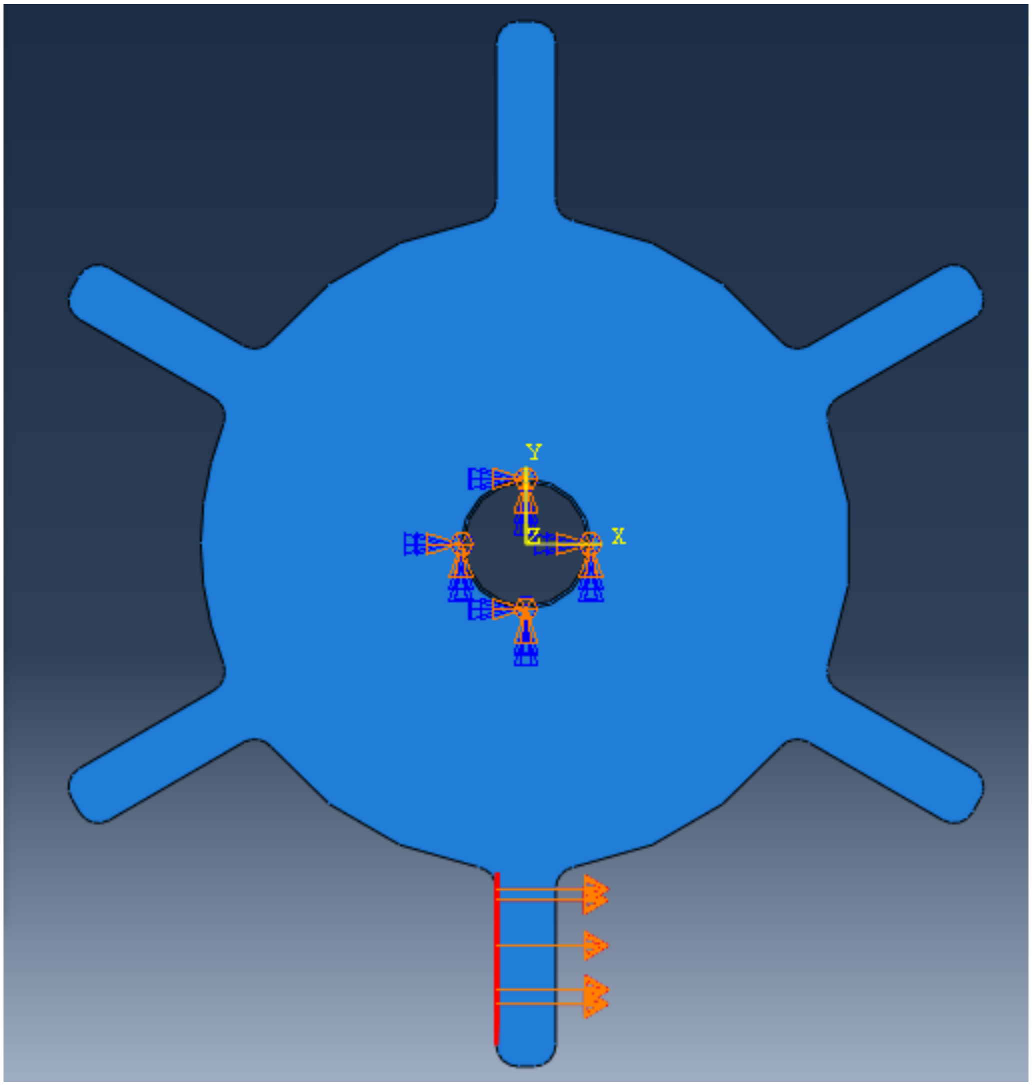
2.2 Find

Perform a mesh convergence analysis on the paddle wheel using elements of your choosing.

- Include a minimum of 4 mesh densities.
- Use the probe tool in Abaqus to determine the stress at the same location in each simulation rather than taking the maximum stress.
- Consider partitioning the geometry, so the mesh puts a node in the same location each time.

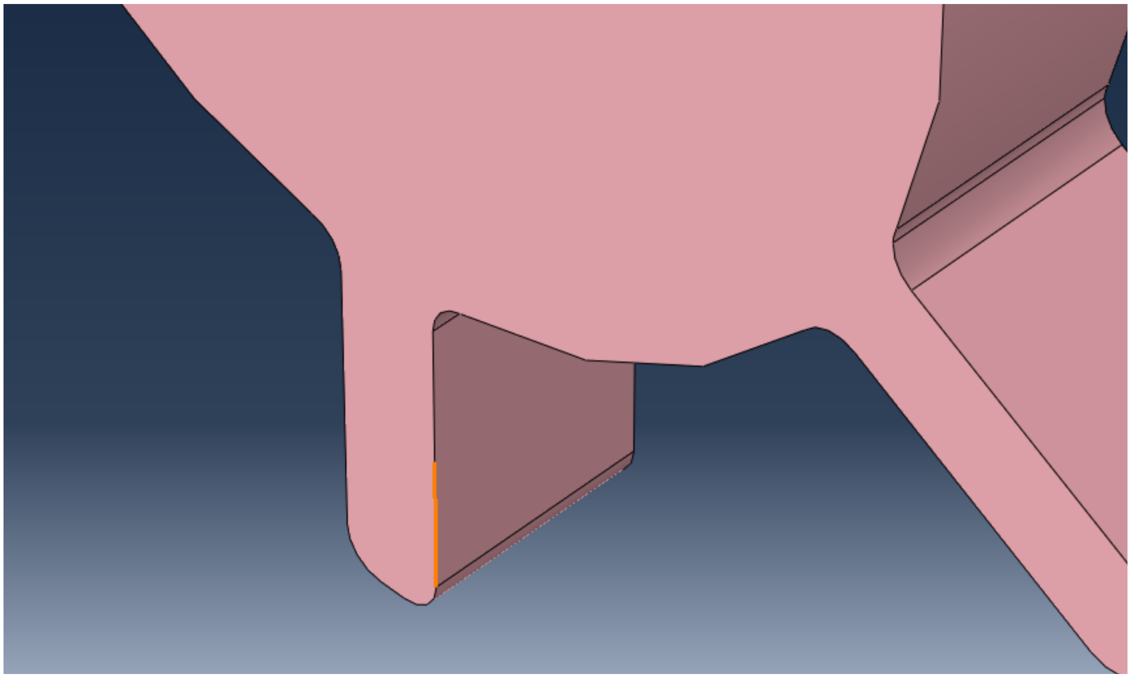
2.3 Solution

2.3.1 Boundary and Loading Conditions

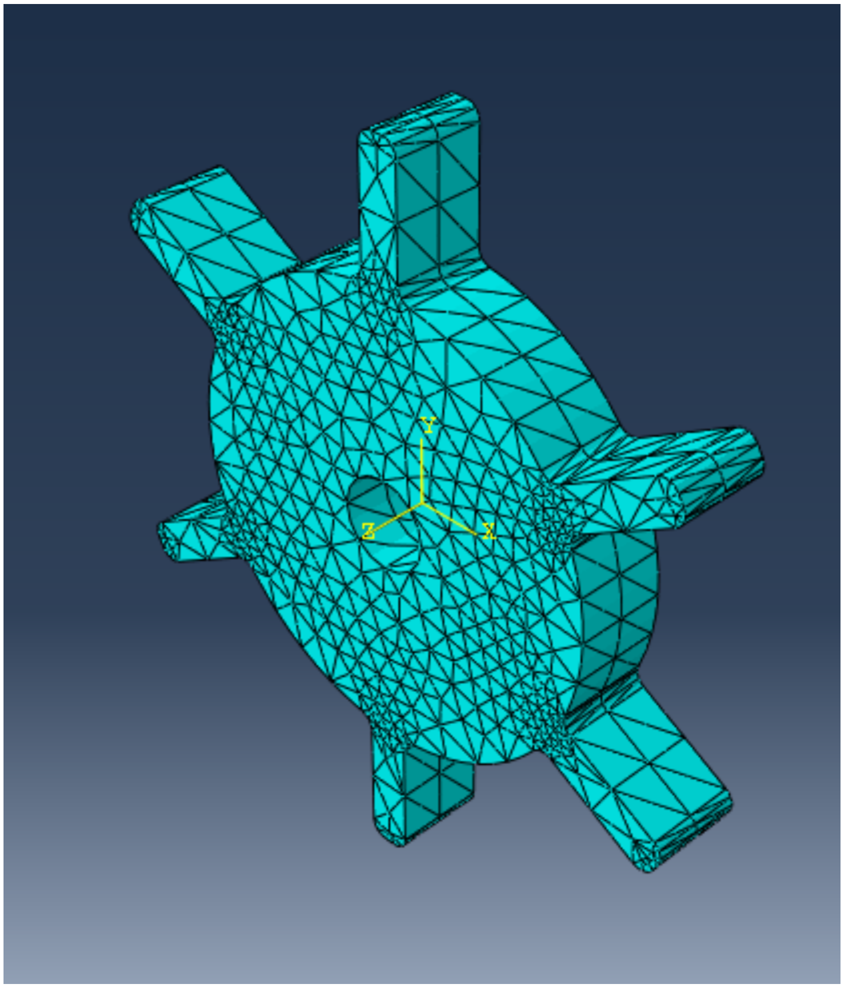


The above figure shows that there is 130 lbf acting on the vertical red face. The area of this surface is $43.0635\,in^2$, which means that a magnitude of $\frac{130\,lbf}{43.0635\,in^2} = 3.02\,psi$ was used for the surface traction. The boundary condition is the encastre option on the inner face of the wheel, which fixes all the degrees of freedom for the nodes on that surface.

2.3.2 Partition and Mesh

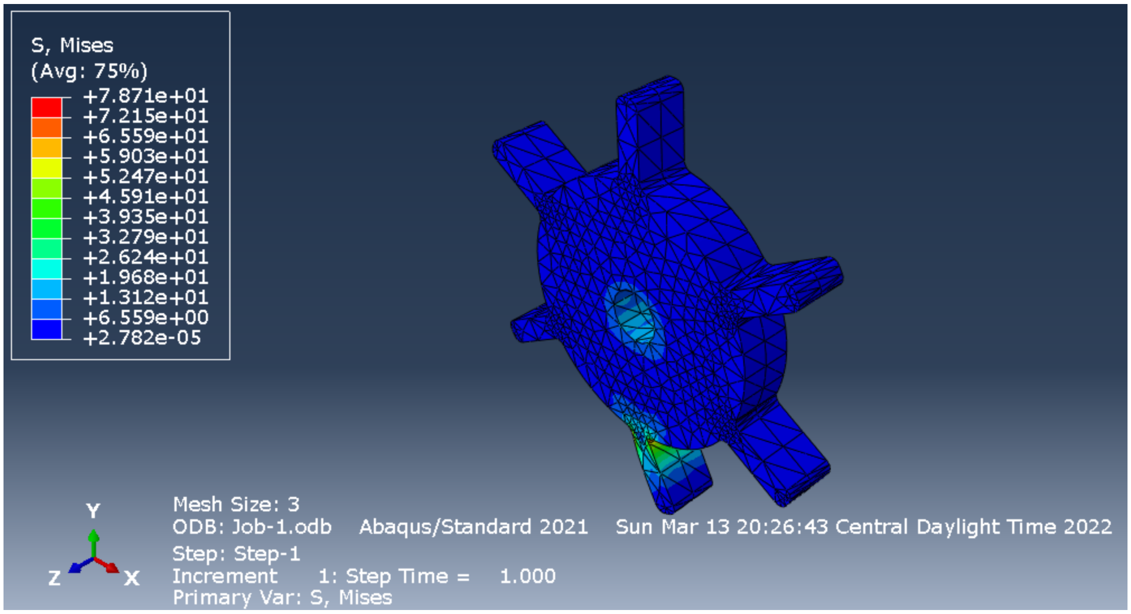


The line above contains a partition about the midpoint of the edge shown. The same element attached to the midpoint node will be considered for this analysis.



The mesh element chosen was the Tet C3D10 element type. This element was chosen due to the many circular surfaces of the geometry (mesh size of 3 inches shown above).

2.3.3 Results



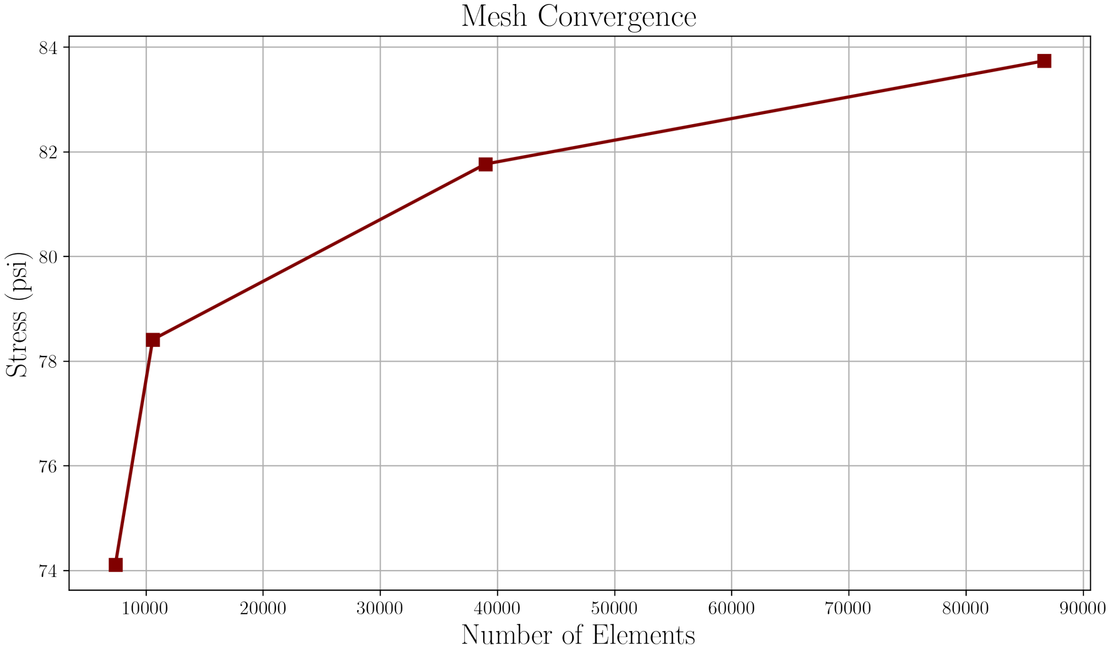
Above shows the results for the mesh size of 3 inches. The study above was repeated for different mesh sizes shown in the table below.

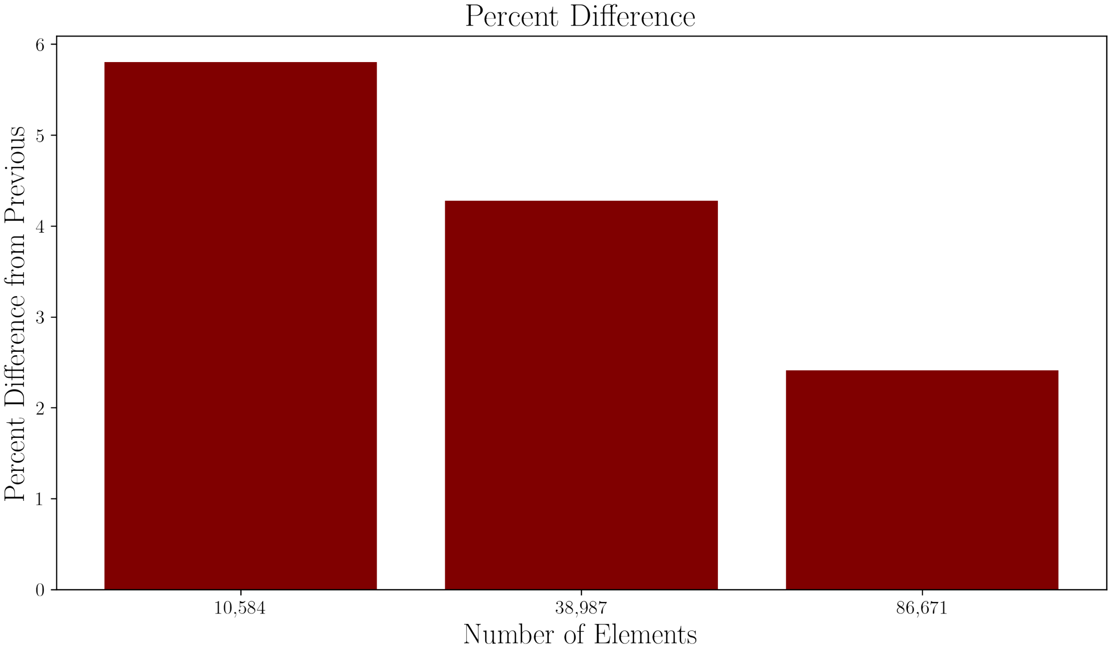
Mesh Size (in)	Total Elements	Stress Value (psi)
3	7,400	74.110
2	10,584	78.411
1	38,987	81.765
0.75	86,671	83.736

[5]:
```
%config InteractiveShell.ast_node_interactivity='last_expr'
```

[6]:
```
# Getting a plot
elements = np.array([7400, 10_584, 38_987, 86_671])
stresses = np.array([74.11, 78.411, 81.765, 83.736])

percents = [(stresses[i] - stresses[i - 1])/stresses[i - 1]*100 for i in␣
 ↪range(1, stresses.size)]

# Underscores used to supress unwanted output
plt.plot(elements, stresses, marker='s')
plt.xlabel('Number of Elements')
plt.ylabel('Stress (psi)')
plt.title('Mesh Convergence')
plt.show()

# Creating bar chart
plt.bar([f'{element:,}' for element in elements[1:]], percents)
plt.xlabel('Number of Elements')
```

```python
plt.ylabel('Percent Difference from Previous')
plt.title('Percent Difference')
plt.grid(visible=False)
plt.show()
```

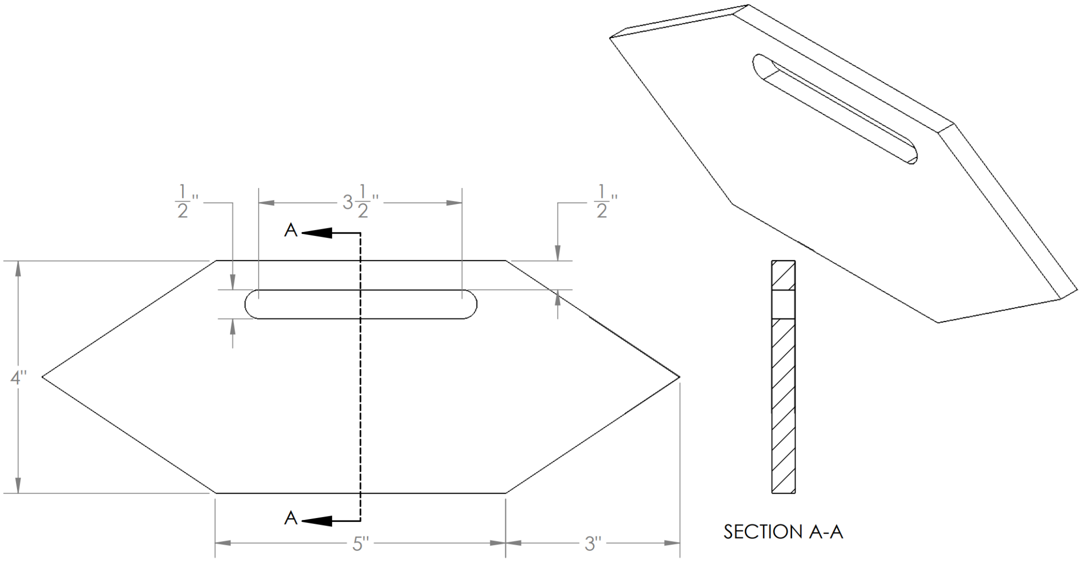




The results show that the increase in the number of elements will converge to the exact solution (around 84 psi). The bar chart shows the percent difference from the previous iteration and shows that the percent difference decreases with each iteration, which indicates that the derivative of the stress with respect to the number of elements is converging at zero.

3 Problem 3

3.1 Given



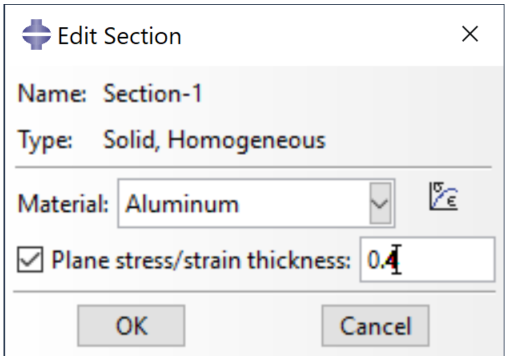
Given the steel machine element, an applied load of $F = 7,000\,lbf$ and a thickness of 0.4 inches is applied to the element shown above.

3.2 Find

a. Solve for the stress distribution along section A-A using a plane stress finite element analysis.
b. Compare the FEA results with a stress distribution calculated using mechanics of materials principles.

3.3 Solution

3.3.1 Part A



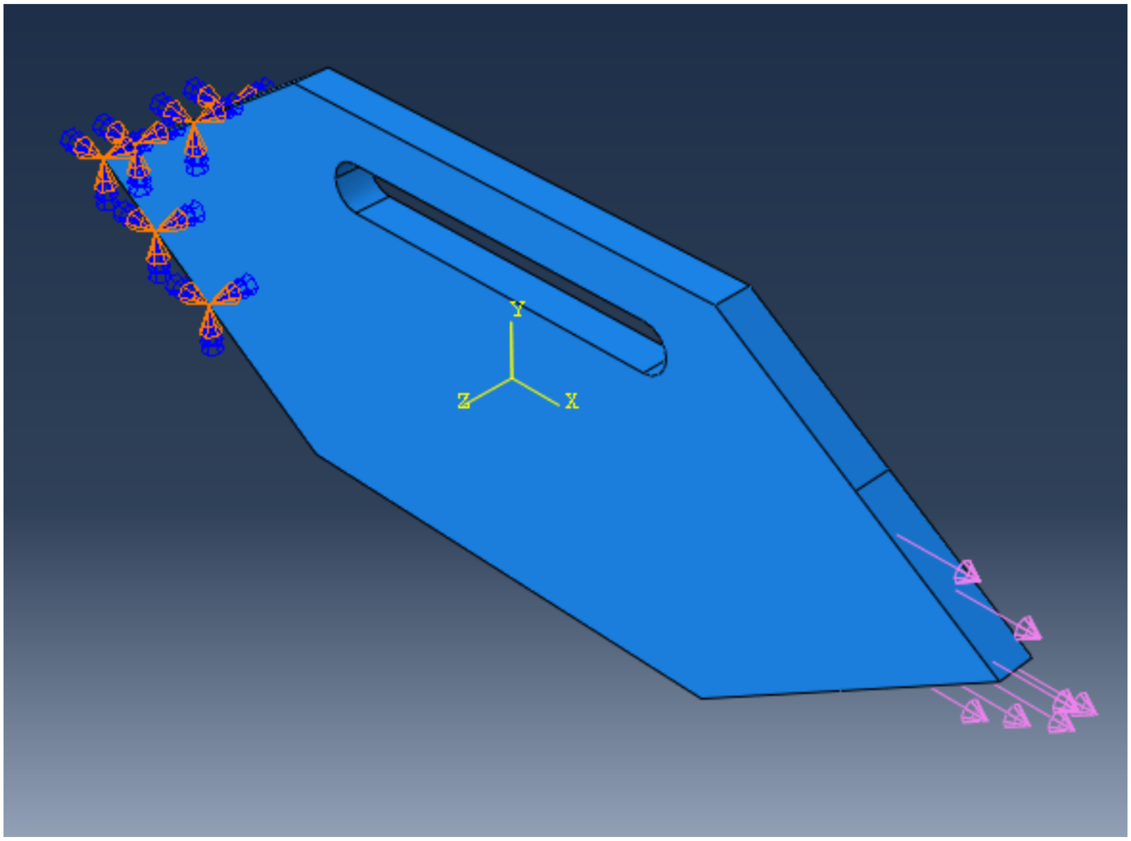
Edit Section

Name: Section-1
Type: Solid, Homogeneous

Material: Aluminum

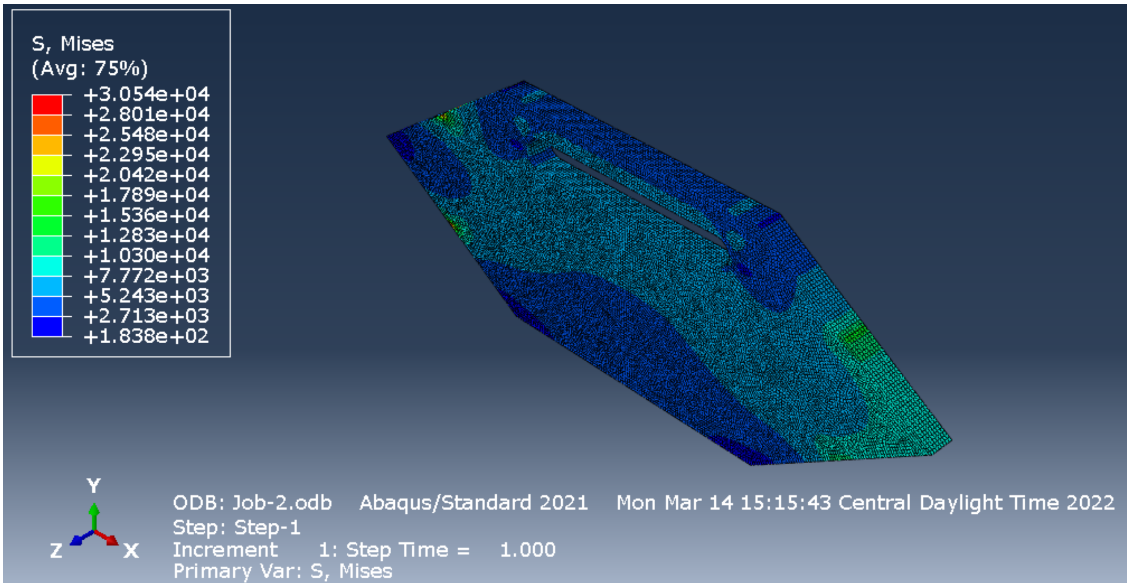
☑ Plane stress/strain thickness: 0.4

I am going to assume that this is aluminum with $E = 10\,ksi$ and $\nu = 0.35$. The plane stress option was chosen as shown above. A partition was created so that the force and boundary conditions are extended to the split faces shown below.



The Hex C3D8R element type was used due to the limitation of the number of nodes and because there are not many curved surfaces in the geometry. An approximate mesh size of 0.05 inches was used.

FEA Results



```
ODB: Job-2.odb    Abaqus/Standard 2021    Mon Mar 14 15:15:43 Central Daylight Time 2022
Step: Step-1
Increment      1: Step Time =     1.000
Primary Var: S, Mises
```

The stress values at the bottom of the plate, midway between the bottom slotted surface and the bottom of the plate, surfaces of the slotted hole, and top of the plate were evaluated in the table below. If the y value is starting from the bottom of the plate:

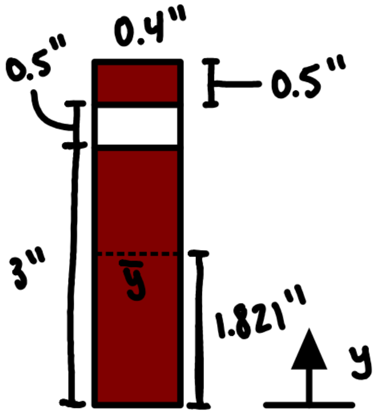
y (in)	Stress (psi)
0	3174
1.5	5200
3	7000
3.5	6162
4	4292

3.3.2 Part B

The mechanics of materials method would take into account the bending moment produced by the moment arm from the centroidal axis.

$$\sigma = \frac{F}{A} + \frac{My}{I}$$

This above method is not valid for cross-sections that are split into different areas, but it will get us into the ballpark of what the answer should be. Not utilizing this assumption would make the problem more difficult.



In the above image, the centroid in the vertical direction may be calculated from the datum at the bottom.

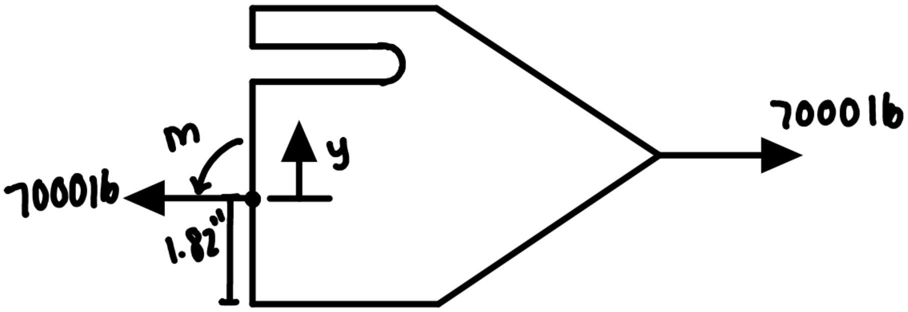
$$\bar{y} = \frac{1.5(3 \cdot 0.4) + 3.75(0.5 \cdot 0.4)}{3 \cdot 0.4 + 0.5 \cdot 0.4} = 1.821”$$

The moment of inertia about the centroidal axis may be determined using the parallel axis theorem.

[7]:
```
# Calculating the moment of inertia
y_bar_ = (1.5*3*0.4 + 3.75*0.5*0.4)/(3*0.4 + 0.5*0.4)
```

```
Iy_ = 1/12*0.4*3**3 + 3*0.4*(y_bar_ - 1.5)**2 + 1/12*0.4*0.5**3 + 0.5*0.4*(3 -␣
 ↪y_bar_ + 0.75)**2
Iy_  # in^4
```

```
[7]: 1.7720238095238097
```



The bending moment may now be calculating by summing the moments about the centroidal axis.

```
[8]: # Getting M
M_ = (2 - y_bar_)*7_000
M_
```

```
[8]: 1250.0000000000018
```

Now, the relationship of σ described above may now be used with the coordinate system starting at the centroidal axis.

```
[9]: # Getting the new coordinate points
# The following y values correspond to those surfaces described in the previous␣
 ↪section
y_values = np.array([-y_bar_, 0, 3 - y_bar_, 3.5 - y_bar_, 4 - y_bar_])

# Get stress values
stress_ = 7_000/1.4 + M_*y_values/Iy_
stress_
```

```
[9]: array([3715.14947934, 5000.        , 5831.37386631, 6184.0779308 ,
            6536.7819953 ])
```

3.3.3 Comparison

```
[10]: # Generating table
data = {'y': [0, 1.5, 3, 3.5, 4], r'$\sigma_{FEA}$': [3174, 5200, 7000, 6162,␣
 ↪4292], r'$\sigma_{Materials}$': stress_}
df = pd.DataFrame(data)
```

```python
display(Latex(df.to_latex(index=False, escape=False)))
```

y	σ_{FEA}	$\sigma_{Materials}$
0.0	3174	3715.149479
1.5	5200	5000.000000
3.0	7000	5831.373866
3.5	6162	6184.077931
4.0	4292	6536.781995

The values do differ due to 1) the mechanics of material method does not account for the hole in the cross-section and 2) the FEA model depends on the material properties because FEA requires the compatability equations in order to solve. Had I chosen a different material, the results could have been different, but nonetheless, the FEA results are preferred and more accurate because the loading conditions reflect the real world phenomena.