FEA Homework 2

February 14, 2022

Gabe Morris

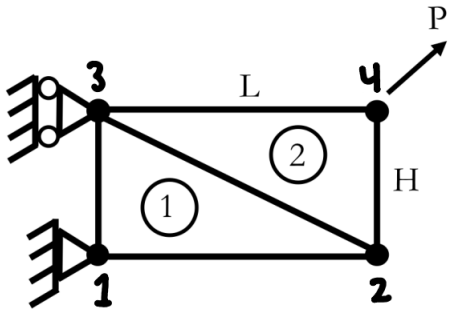
[1]:
```
import sympy as sp
import matplotlib.pyplot as plt
from IPython.display import display, Latex

plt.style.use('maroon.mplstyle')

display_latex = lambda text: display(Latex(text))

def round_expr(expr, num_digits):
    return expr.xreplace({n : round(n, num_digits) for n in expr.atoms(sp.
↪Number)})
```

Contents

ME 6233 Homework 2 Gabe Morris
gnm54

1 Problem 1

1.1 Given



$P = 150\,lb$, $L = 5\,in$, $H = 2\,in$, $t = 0.5\,in$, $E = 30 \cdot 10^6\,psi$, and $\nu = 0.30$

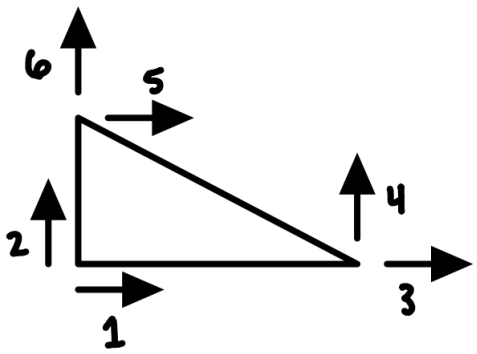
Notice that the global nodes have been rearranged. This was done to make the mapping easier.

1.2 Find

a. The global stiffness matrix
b. The displacements at each node
c. The stresses within each element
d. Plot the undeformed and deformed shape

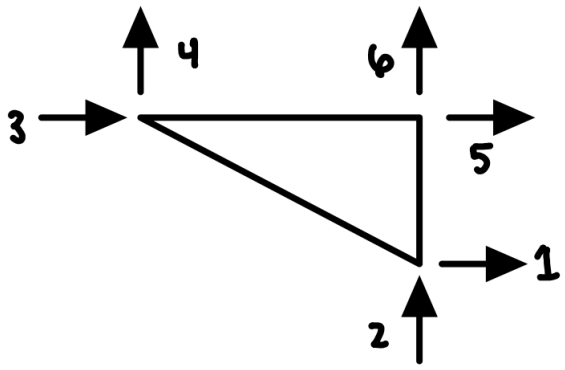
1.3 Solution

For the first element,



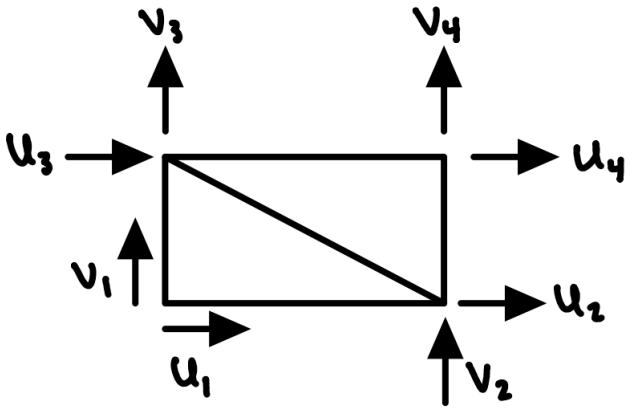
$\delta_1 = \delta_2 = \delta_5 = 0$
$\delta_3 = u_2$
$\delta_4 = v_2$
$\delta_6 = v_3$

For the second element,



$\delta_3 = 0$
$\delta_1 = u_2$
$\delta_2 = v_2$
$\delta_4 = v_3$
$\delta_5 = u_4$
$\delta_6 = v_4$

The global displacements are,



1.3.1 Part A

[2]:
```
# Define numerical inputs here
P_ = 150
L_, H_ = 5, 2
t_ = 0.5
E_, nu_ = 30e6, 0.3

# Define local coordinates for each element
x1 = [0, L_, 0]
```

```
y1 = [0, 0, H_]

x2 = [L_, 0, L_]
y2 = [0, H_, H_]

# The area
A_1 = ((x1[1] - x1[0])*(y1[2] - y1[0]) - (x1[2] - x1[0])*(y1[1] - y1[0]))/2
A_2 = -1*((x2[1] - x2[0])*(y2[2] - y2[0]) - (x2[2] - x2[0])*(y2[1] - y2[0]))/2
A_1, A_2
```

[2]:
```
(5.0, 5.0)
```

[3]:
```
# Define a symbolic B
A, y_23, y_31, y_12, x_32, x_13, x_21 = sp.symbols('A y_{23} y_{31} y_{12}␣
↪x_{32} x_{13} x_{21}')
B = 1/(2*A)*sp.Matrix([
    [y_23, 0, y_31, 0, y_12, 0],
    [0, x_32, 0, x_13, 0, x_21],
    [x_32, y_23, x_13, y_31, x_21, y_12]
])
B
```

[3]:
$$\begin{bmatrix} \frac{y_{23}}{2A} & 0 & \frac{y_{31}}{2A} & 0 & \frac{y_{12}}{2A} & 0 \\ 0 & \frac{x_{32}}{2A} & 0 & \frac{x_{13}}{2A} & 0 & \frac{x_{21}}{2A} \\ \frac{x_{32}}{2A} & \frac{y_{23}}{2A} & \frac{x_{13}}{2A} & \frac{y_{31}}{2A} & \frac{x_{21}}{2A} & \frac{y_{12}}{2A} \end{bmatrix}$$

[4]:
```
# Numeric B
# Remember that indices start at 0
B1 = B.subs([
    (A, A_1),
    (y_23, y1[1] - y1[2]),
    (y_31, y1[2] - y1[0]),
    (y_12, y1[0] - y1[1]),
    (x_32, x1[2] - x1[1]),
    (x_13, x1[0] - x1[2]),
    (x_21, x1[1] - x1[0])
])
B1
```

[4]:
$$\begin{bmatrix} -0.2 & 0 & 0.2 & 0 & 0 & 0 \\ 0 & -0.5 & 0 & 0 & 0 & 0.5 \\ -0.5 & -0.2 & 0 & 0.2 & 0.5 & 0 \end{bmatrix}$$

[5]:
```
B2 = B.subs([
    (A, A_2),
    (y_23, y2[1] - y2[2]),
    (y_31, y2[2] - y2[0]),
    (y_12, y2[0] - y2[1]),
```

```
    (x_32, x2[2] - x2[1]),
    (x_13, x2[0] - x2[2]),
    (x_21, x2[1] - x2[0])
])
B2
```

[5]:
$$\begin{bmatrix} 0 & 0 & 0.2 & 0 & -0.2 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & -0.5 \\ 0.5 & 0 & 0 & 0.2 & -0.5 & -0.2 \end{bmatrix}$$

[6]:
```
E = E_/(1 - nu_**2)*sp.Matrix([
    [1, nu_, 0],
    [nu_, 1, 0],
    [0, 0, (1 - nu_)/2]
])
round_expr(E, 3)
```

[6]:
$$\begin{bmatrix} 32967032.967 & 9890109.89 & 0 \\ 9890109.89 & 32967032.967 & 0 \\ 0 & 0 & 11538461.538 \end{bmatrix}$$

[7]:
```
k1_local = t_*A_1*sp.transpose(B1)*E*B1
round_expr(k1_local, 3)
```

[7]:
$$\begin{bmatrix} 10508241.758 & 5357142.857 & -3296703.297 & -2884615.385 & -7211538.462 & -2472527.473 \\ 5357142.857 & 21758241.758 & -2472527.473 & -1153846.154 & -2884615.385 & -20604395.604 \\ -3296703.297 & -2472527.473 & 3296703.297 & 0 & 0 & 2472527.473 \\ -2884615.385 & -1153846.154 & 0 & 1153846.154 & 2884615.385 & 0 \\ -7211538.462 & -2884615.385 & 0 & 2884615.385 & 7211538.462 & 0 \\ -2472527.473 & -20604395.604 & 2472527.473 & 0 & 0 & 20604395.604 \end{bmatrix}$$

[8]:
```
k2_local = t_*A_2*sp.transpose(B2)*E*B2
round_expr(k2_local, 3)
```

[8]:
$$\begin{bmatrix} 7211538.462 & 0 & 0 & 2884615.385 & -7211538.462 & -2884615.385 \\ 0 & 20604395.604 & 2472527.473 & 0 & -2472527.473 & -20604395.604 \\ 0 & 2472527.473 & 3296703.297 & 0 & -3296703.297 & -2472527.473 \\ 2884615.385 & 0 & 0 & 1153846.154 & -2884615.385 & -1153846.154 \\ -7211538.462 & -2472527.473 & -3296703.297 & -2884615.385 & 10508241.758 & 5357142.857 \\ -2884615.385 & -20604395.604 & -2472527.473 & -1153846.154 & 5357142.857 & 21758241.758 \end{bmatrix}$$

[9]:
```
k1_global = k1_local.col_insert(6, sp.zeros(rows=6, cols=2)).row_insert(6, sp.
↪zeros(rows=2, cols=8))
round_expr(k1_global, 3)
```

[9]:

$$\begin{bmatrix} 10508241.758 & 5357142.857 & -3296703.297 & -2884615.385 & -7211538.462 & -2472527.473 & 0 & 0 \\ 5357142.857 & 21758241.758 & -2472527.473 & -1153846.154 & -2884615.385 & -20604395.604 & 0 & 0 \\ -3296703.297 & -2472527.473 & 3296703.297 & 0 & 0 & 2472527.473 & 0 & 0 \\ -2884615.385 & -1153846.154 & 0 & 1153846.154 & 2884615.385 & 0 & 0 & 0 \\ -7211538.462 & -2884615.385 & 0 & 2884615.385 & 7211538.462 & 0 & 0 & 0 \\ -2472527.473 & -20604395.604 & 2472527.473 & 0 & 0 & 20604395.604 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

[10]:
```
k2_global = k2_local.col_insert(0, sp.zeros(rows=6, cols=2)).row_insert(0, sp.
↪zeros(rows=2, cols=8))
round_expr(k2_global, 3)
```

[10]:
$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7211538.462 & 0 & 0 & 2884615.385 & -7211538.462 & -2884615.385 \\ 0 & 0 & 0 & 20604395.604 & 2472527.473 & 0 & -2472527.473 & -20604395.604 \\ 0 & 0 & 0 & 2472527.473 & 3296703.297 & 0 & -3296703.297 & -2472527.473 \\ 0 & 0 & 2884615.385 & 0 & 0 & 1153846.154 & -2884615.385 & -1153846.154 \\ 0 & 0 & -7211538.462 & -2472527.473 & -3296703.297 & -2884615.385 & 10508241.758 & 5357142.857 \\ 0 & 0 & -2884615.385 & -20604395.604 & -2472527.473 & -1153846.154 & 5357142.857 & 21758241.758 \end{bmatrix}$$

[11]:
```
k_global = k1_global + k2_global
round_expr(k_global, 1)
```

[11]:
$$\begin{bmatrix} 10508241.8 & 5357142.9 & -3296703.3 & -2884615.4 & -7211538.5 & -2472527.5 & 0 & 0 \\ 5357142.9 & 21758241.8 & -2472527.5 & -1153846.2 & -2884615.4 & -20604395.6 & 0 & 0 \\ -3296703.3 & -2472527.5 & 10508241.8 & 0 & 0 & 5357142.9 & -7211538.5 & -2884615.4 \\ -2884615.4 & -1153846.2 & 0 & 21758241.8 & 5357142.9 & 0 & -2472527.5 & -20604395.6 \\ -7211538.5 & -2884615.4 & 0 & 5357142.9 & 10508241.8 & 0 & -3296703.3 & -2472527.5 \\ -2472527.5 & -20604395.6 & 5357142.9 & 0 & 0 & 21758241.8 & -2884615.4 & -1153846.2 \\ 0 & 0 & -7211538.5 & -2472527.5 & -3296703.3 & -2884615.4 & 10508241.8 & 5357142.9 \\ 0 & 0 & -2884615.4 & -20604395.6 & -2472527.5 & -1153846.2 & 5357142.9 & 21758241.8 \end{bmatrix}$$

1.3.2 Part B

[12]:
```
F_1, F_2, F_5 = sp.symbols('F_1 F_2 F_5')
F = sp.Matrix([
    [F_1],
    [F_2],
    [0],
    [0],
    [F_5],
    [0],
    [P_*sp.sqrt(2)/2],
    [F_*sp.sqrt(2)/2]
]).n()
F
```

[12]:

$$\begin{bmatrix} F_1 \\ F_2 \\ 0 \\ 0 \\ F_5 \\ 0 \\ 106.066017177982 \\ 106.066017177982 \end{bmatrix}$$

[13]:
```
u2, v2, v3, u4, v4 = sp.symbols('u_2 v_2 v_3 u_4 v_4')
d = sp.Matrix([
    [0],
    [0],
    [u2],
    [v2],
    [0],
    [v3],
    [u4],
    [v4]
])
d
```

[13]:
$$\begin{bmatrix} 0 \\ 0 \\ u_2 \\ v_2 \\ 0 \\ v_3 \\ u_4 \\ v_4 \end{bmatrix}$$

[14]:
```
system = sp.Eq(F, k_global*d)
round_expr(system, 3)
```

[14]:
$$\begin{bmatrix} F_1 \\ F_2 \\ 0 \\ 0 \\ F_5 \\ 0 \\ 106.066 \\ 106.066 \end{bmatrix} = \begin{bmatrix} -3296703.297u_2 - 2884615.385v_2 - 2472527.473v_3 \\ -2472527.473u_2 - 1153846.154v_2 - 20604395.604v_3 \\ 10508241.758u_2 - 7211538.462u_4 + 5357142.857v_3 - 2884615.385v_4 \\ -2472527.473u_4 + 21758241.758v_2 - 20604395.604v_4 \\ -3296703.297u_4 + 5357142.857v_2 - 2472527.473v_4 \\ 5357142.857u_2 - 2884615.385u_4 + 21758241.758v_3 - 1153846.154v_4 \\ -7211538.462u_2 + 10508241.758u_4 - 2472527.473v_2 - 2884615.385v_3 + 5357142.857v_4 \\ -2884615.385u_2 + 5357142.857u_4 - 20604395.604v_2 - 1153846.154v_3 + 21758241.758v_4 \end{bmatrix}$$

[15]:
```
solved = sp.solve(system)
for key, value in solved.items():
    display_latex(f'${sp.latex(key)}={sp.latex(value)}$')
```

$F_1 = -265.165042944938$

$F_2 = -106.066017177975$

$F_5 = 159.099025766958$

$u_2 = 2.42131523003677 \cdot 10^{-5}$

$u_4 = 6.89954607183987 \cdot 10^{-6}$

$v_2 = 6.54725387051039 \cdot 10^{-5}$

$v_3 = -1.4243030764922 \cdot 10^{-6}$

$v_4 = 6.83110553439691 \cdot 10^{-5}$

1.3.3 Part C

[16]:
```
strain1 = B1*sp.Matrix([
    [0],
    [0],
    [solved[u2]],
    [solved[v2]],
    [0],
    [solved[v3]]
])
strain1
```

[16]:
$$\begin{bmatrix} 4.84263046007354 \cdot 10^{-6} \\ -7.12151538246102 \cdot 10^{-7} \\ 1.30945077410208 \cdot 10^{-5} \end{bmatrix}$$

[17]:
```
stress1 = E*strain1
stress1
```

[17]:
$$\begin{bmatrix} 152.603901052738 \\ 24.4166241684383 \\ 151.090473934855 \end{bmatrix}$$

[18]:
```
strain2 = B2*sp.Matrix([
    [solved[u2]],
    [solved[v2]],
    [0],
    [solved[v3]],
    [solved[u4]],
    [solved[v4]]
])
strain2
```

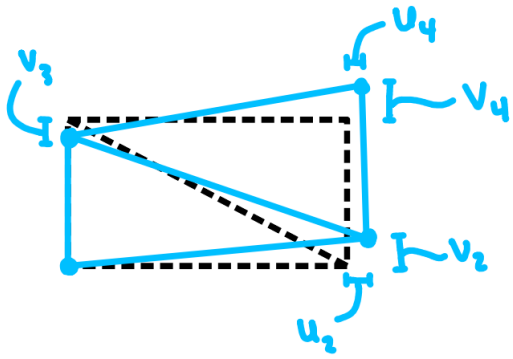
[18]:
$$\begin{bmatrix} -1.37990921436797 \cdot 10^{-6} \\ -1.41925831943259 \cdot 10^{-6} \\ -5.29026856982834 \cdot 10^{-6} \end{bmatrix}$$

[19]:
```
stress2 = E*strain2
stress2
```

[19]:

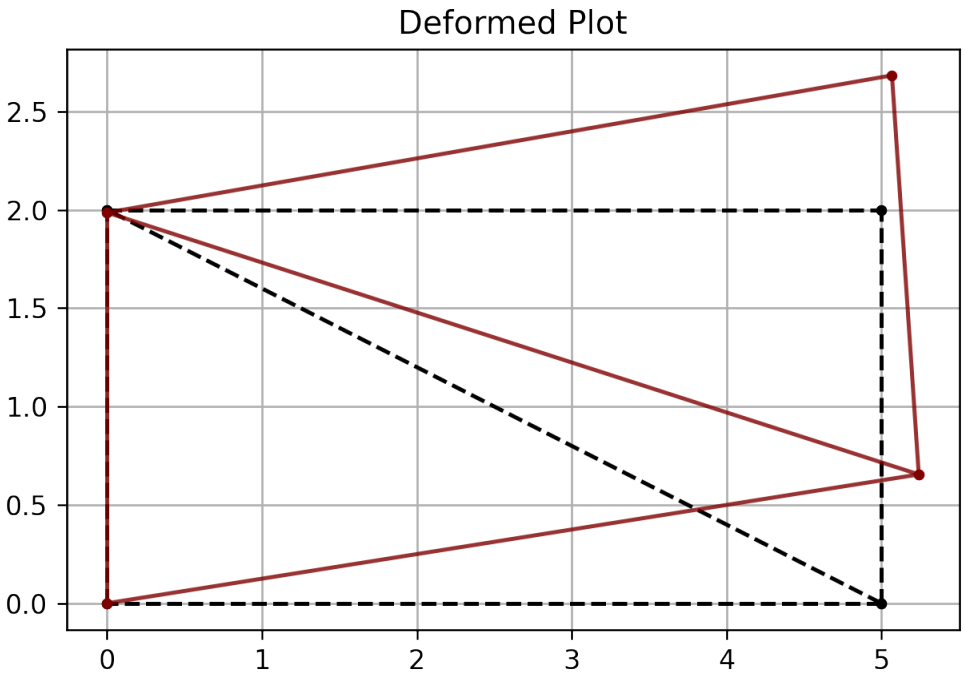
$$\begin{bmatrix} -59.5281333032225 \\ -60.4361895739445 \\ -61.0415604210962 \end{bmatrix}$$

1.3.4 Part D



Remember, I changed the definition of the nodes to make things easier. Here is a plot to scale:

[20]:
```
undeformed = {'color': 'black', 'ls': '--', 'label': 'undeformed', 'marker': '.
 ↪'}
deformed = {'color': 'maroon', 'label': 'deformed', 'marker': '.', 'alpha': 0.8}
s_ = 10_000  # How exaggerated the results are going to be

plt.plot([0, 0], [0, H_], [L_, 0], [0, 0], [L_, 0], [0, H_], [L_, L_], [0, H_],␣
 ↪[0, L_], [H_, H_], **undeformed)
plt.plot([0, 0], [0, H_ + s_*solved[v3]], [0, L_ + s_*solved[u2]], [0,␣
 ↪s_*solved[v2]], [0, L_ + s_*solved[u2]], [H_ + s_*solved[v3],␣
 ↪s_*solved[v2]], [L_ + s_*solved[u4], 0], [H_ + s_*solved[v4], H_ +␣
 ↪s_*solved[v3]], [L_ + s_*solved[u2], L_ + s_*solved[u4]], [s_*solved[v2], H_␣
 ↪+ s_*solved[v4]], **deformed)
plt.title('Deformed Plot')
plt.show()
```


Deformed Plot
2.5
2.0
1.5
1.0
0.5
0.0
0
1
2
3
4
5