HOMEWORK 4

CEE 361-513: Introduction to Finite Element Methods

Due: Friday Oct. 20 @ Midnight

NB: Students taking CEE 513 must complete all problems. All other students will not be graded for problems marked with $\star$, but are encourage to attempt them anyhow.

PROBLEM 1

Consider the frame shown below. Foreach node $z = 1, 2, 3$ we have associated coordinates $\boldsymbol{q_z}$ and associated global degrees of freedom $\boldsymbol{u}_z$ and θ_z, where both $\boldsymbol{q}$ and $\boldsymbol{u}$ are vectors while θ_z are scalar rotations. At node 3 the frame is free to rotate but is constrained to move along a plane whose normal is given by $\boldsymbol{m_s}$.

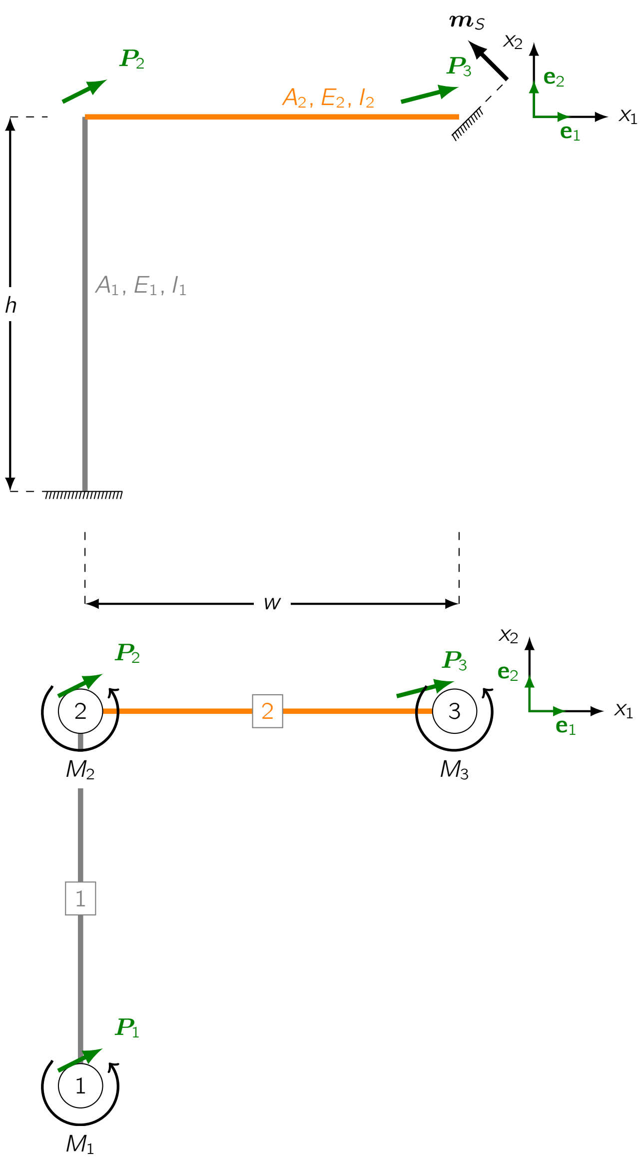


Figure 1: The system of uniaxial rods

1. For each node write the equilibrium equations in terms of the external forces $\boldsymbol{P}_k$, $k = 1, 2, 3$ and moments M_k, $k = 1, 2, 3$, and the internal forces $\boldsymbol{f}^e_{i,j}$ and moments $m^e_{i,j}$.

```
Solution :
```

$$
\begin{aligned}
\boldsymbol{V}_1 &= -\boldsymbol{f}_i^{n1} + \boldsymbol{f}_i^{s1} \\
M_1 &= -m_i^1 \\
\boldsymbol{V}_2 &= \boldsymbol{f}_j^{n1} - \boldsymbol{f}_j^{s1} - \boldsymbol{f}_i^{n2} + \boldsymbol{f}_i^{s2} \\
M_2 &= -m_i^2 + m_j^1 \\
\boldsymbol{V}_3 &= \boldsymbol{f}_j^{n2} - \boldsymbol{f}_i^{s2} \\
M_3 &= m_j^2
\end{aligned}
$$

2. Write the general expression of internal forces (and moments) as the matrix vector operation of the *local element stiffness* and the *local degrees of freedom*.

```
Solution :
Internal forces as matrix vector operation of the local element stiffness and local
degrees of freedom.
```

$$
\begin{bmatrix} \boldsymbol{V}_i \\ M_i \\ \boldsymbol{V}_j \\ M_j \end{bmatrix} = \begin{bmatrix} [\boldsymbol{K_{fw}}] & [\boldsymbol{k}_{\boldsymbol{f}\theta}] & [-\boldsymbol{K_{fw}}] & [\boldsymbol{k}_{\boldsymbol{f}\theta}] \\ [\boldsymbol{k}_{m\boldsymbol{w}}]^T & [k_{m\theta}] & [-\boldsymbol{k}_{m\boldsymbol{w}}]^T & [\hat{k}_{m\theta}] \\ [-\boldsymbol{K_{fw}}] & [-\boldsymbol{k}_{\boldsymbol{f}\theta}] & [\boldsymbol{K_{fw}}] & [-\boldsymbol{k}_{\boldsymbol{f}\theta}] \\ [\boldsymbol{k}_{m\boldsymbol{w}}]^T & [\hat{k}_{m\theta}] & [-\boldsymbol{k}_{m\boldsymbol{w}}]^T & [k_{m\theta}] \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_i \\ \theta_i \\ \boldsymbol{w}_j \\ \theta_j \end{bmatrix}
$$

```
where:
```

$$
\begin{aligned}
\boldsymbol{K_{fw}} &= \frac{A_e E_e}{\ell_e} \boldsymbol{n}^e \otimes \boldsymbol{n}^e + \frac{12 E_e I_e}{\ell_e^3} \boldsymbol{s}^e \otimes \boldsymbol{s}^e \\
k_{m\theta} &= \frac{4 E_e I_e}{\ell_e} \\
\hat{k}_{m\theta} &= \frac{2 E_e I_e}{\ell_e} \\
\boldsymbol{k}_{m\boldsymbol{w}} = \boldsymbol{k}_{\boldsymbol{f}\theta} &= \frac{6 E_e I_e}{\ell_e^2} \boldsymbol{s}^e
\end{aligned}
$$

For each element e

3. For each element write the internal forces (and moments) as the matrix vector operation of the *local element stiffness* and the *GLOBAL degrees of freedom* using the connectivity array.

Solution :
For element 1

$$\begin{bmatrix} \boldsymbol{V}_1 \\ M_1 \\ \boldsymbol{V}_2 \\ M_2 \end{bmatrix} = \begin{bmatrix} [\boldsymbol{K}^1_{\boldsymbol{fw}}] & [\boldsymbol{k}^1_{\boldsymbol{f\theta}}] & [-\boldsymbol{K}^1_{\boldsymbol{fw}}] & [\boldsymbol{k}^1_{\boldsymbol{f\theta}}] \\ [\boldsymbol{k}^1_{\boldsymbol{mw}}]^T & [k^1_{m\theta}] & [-\boldsymbol{k}^1_{\boldsymbol{mw}}]^T & [\hat{k}^1_{m\theta}] \\ [-\boldsymbol{K}^1_{\boldsymbol{fw}}] & [-\boldsymbol{k}^1_{\boldsymbol{f\theta}}] & [\boldsymbol{K}^1_{\boldsymbol{fw}}] & [-\boldsymbol{k}^1_{\boldsymbol{f\theta}}] \\ [\boldsymbol{k}^1_{\boldsymbol{mw}}]^T & [\hat{k}^1_{m\theta}] & [-\boldsymbol{k}^1_{\boldsymbol{mw}}]^T & [k^1_{m\theta}] \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 \\ \theta_1 \\ \boldsymbol{w}_2 \\ \theta_2 \end{bmatrix}$$

For element 2

$$\begin{bmatrix} \boldsymbol{V}_2 \\ M_2 \\ \boldsymbol{V}_2 \\ M_2 \end{bmatrix} = \begin{bmatrix} [\boldsymbol{K}^2_{\boldsymbol{fw}}] & [\boldsymbol{k}^2_{\boldsymbol{f\theta}}] & [-\boldsymbol{K}^2_{\boldsymbol{fw}}] & [\boldsymbol{k}^2_{\boldsymbol{f\theta}}] \\ [\boldsymbol{k}^2_{\boldsymbol{mw}}]^T & [k^2_{m\theta}] & [-\boldsymbol{k}^2_{\boldsymbol{mw}}]^T & [\hat{k}^2_{m\theta}] \\ [-\boldsymbol{K}^2_{\boldsymbol{fw}}] & [-\boldsymbol{k}^2_{\boldsymbol{f\theta}}] & [\boldsymbol{K}^2_{\boldsymbol{fw}}] & [-\boldsymbol{k}^2_{\boldsymbol{f\theta}}] \\ [\boldsymbol{k}^2_{\boldsymbol{mw}}]^T & [\hat{k}^2_{m\theta}] & [-\boldsymbol{k}^2_{\boldsymbol{mw}}]^T & [k^2_{m\theta}] \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_2 \\ \theta_2 \\ \boldsymbol{w}_3 \\ \theta_3 \end{bmatrix}$$

4. Using $\boldsymbol{K}^e_{\boldsymbol{fw}}, \boldsymbol{k}^e_{\boldsymbol{f\theta}}, \ldots$ (cf. lecture notes), write down the equilibrium equations in matrix form.

Solution :
Equilibrium equation in Matrix Form:

$$\begin{bmatrix} \boldsymbol{V}_1 \\ M_1 \\ \boldsymbol{V}_2 \\ M_2 \\ \boldsymbol{V}_3 \\ M_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}^1_{\boldsymbol{fw}} & \boldsymbol{k}^1_{\boldsymbol{f\theta}} & -\boldsymbol{K}^1_{\boldsymbol{fw}} & \boldsymbol{k}^1_{\boldsymbol{f\theta}} & \boldsymbol{O} & \boldsymbol{0} \\ [\boldsymbol{k}^1_{\boldsymbol{mw}}]^T & k^1_{m\theta} & -[\boldsymbol{k}^1_{\boldsymbol{mw}}]^T & \hat{k}^1{}_{m\theta} & \boldsymbol{0}^T & 0 \\ -\boldsymbol{K}^1_{\boldsymbol{fw}} & -\boldsymbol{k}^1_{\boldsymbol{f\theta}} & \boldsymbol{K}^1_{\boldsymbol{fw}} + \boldsymbol{K}^2_{\boldsymbol{fw}} & -\boldsymbol{k}^1_{\boldsymbol{f\theta}} + \boldsymbol{k}^2_{\boldsymbol{f\theta}} & -\boldsymbol{K}^2_{\boldsymbol{fw}} & \boldsymbol{k}^2_{\boldsymbol{f\theta}} \\ [\boldsymbol{k}^1_{\boldsymbol{mw}}]^T & \hat{k}^1{}_{m\theta} & [-\boldsymbol{k}^1_{\boldsymbol{mw}}]^T + [\boldsymbol{k}^2_{\boldsymbol{mw}}]^T & k^1_{m\theta} + k^2_{m\theta} & -[\boldsymbol{k}^2_{\boldsymbol{mw}}]^T & \hat{k}^2{}_{m\theta} \\ \boldsymbol{O} & \boldsymbol{0} & -\boldsymbol{K}^2_{\boldsymbol{fw}} & -\boldsymbol{k}^2_{\boldsymbol{f\theta}} & \boldsymbol{K}^2_{\boldsymbol{fw}} & -\boldsymbol{k}^2_{\boldsymbol{f\theta}} \\ \boldsymbol{0}^T & 0 & [\boldsymbol{k}^2_{\boldsymbol{mw}}]^T & \hat{k}^2{}_{m\theta} & [-\boldsymbol{k}^2_{\boldsymbol{mw}}]^T & k^2_{m\theta} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 \\ \theta_1 \\ \boldsymbol{w}_2 \\ \theta_2 \\ \boldsymbol{w}_3 \\ \theta_3 \end{bmatrix}$$

5. At node 1 we prevent the frame from moving (i.e. all displacements and rotations are zero). At node 3 we allow the truss to move along a plane whose unit normal is $\boldsymbol{m}_S$ as well as to rotate freely. Apply the aforementioned conditions to the matrix form of the previous step.

Solution :
Applying constraints on the system:

$$\begin{bmatrix} \boldsymbol{0} \\ 0 \\ \boldsymbol{V}_2 \\ M_2 \\ \boldsymbol{V}_3 \\ M_3 \\ 0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{O} & \boldsymbol{0} & \boldsymbol{O} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0}^T & 1 & \boldsymbol{0}^T & 0 & \boldsymbol{0}^T & 0 & 0 \\ -\boldsymbol{K}^1_{\boldsymbol{fw}} & -\boldsymbol{k}^1_{\boldsymbol{f\theta}} & \boldsymbol{K}^1_{\boldsymbol{fw}} + \boldsymbol{K}^2_{\boldsymbol{fw}} & -\boldsymbol{k}^1_{\boldsymbol{f\theta}} + \boldsymbol{k}^2_{\boldsymbol{f\theta}} & -\boldsymbol{K}^2_{\boldsymbol{fw}} & \boldsymbol{k}^2_{\boldsymbol{f\theta}} & \boldsymbol{0} \\ [\boldsymbol{k}^1_{\boldsymbol{mw}}]^T & \hat{k}^1{}_{m\theta} & [-\boldsymbol{k}^1_{\boldsymbol{mw}}]^T + [\boldsymbol{k}^2_{\boldsymbol{mw}}]^T & k^1_{m\theta} + k^2_{m\theta} & -[\boldsymbol{k}^2_{\boldsymbol{mw}}]^T & \hat{k}^2{}_{m\theta} & 0 \\ \boldsymbol{O} & \boldsymbol{0} & -\boldsymbol{K}^2_{\boldsymbol{fw}} & -\boldsymbol{k}^2_{\boldsymbol{f\theta}} & \boldsymbol{K}^2_{\boldsymbol{fw}} & -\boldsymbol{k}^2_{\boldsymbol{f\theta}} & -\boldsymbol{m}_s \\ \boldsymbol{0}^T & 0 & [\boldsymbol{k}^2_{\boldsymbol{mw}}]^T & \hat{k}^2{}_{m\theta} & [-\boldsymbol{k}^2_{\boldsymbol{mw}}]^T & k^2_{m\theta} & 0 \\ \boldsymbol{0}^T & 0 & \boldsymbol{0}^T & 0 & \boldsymbol{m}_s^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 \\ \theta_1 \\ \boldsymbol{w}_2 \\ \theta_2 \\ \boldsymbol{w}_3 \\ \theta_3 \\ \lambda \end{bmatrix}$$

PROBLEM 2

Consider the frame of Problem 1. Let $w = 10$, $h = 10$, and $A_1 = E_1 = I_1 = 1$, $A_2 = E_2 = I_2 = 2$. Further let $\boldsymbol{P}_2 = 10\mathbf{e}_1 + 5\mathbf{e}_2$ and $M_2 = 3$, as well as $\boldsymbol{P}_3 = 2\mathbf{e}_1 + 5\mathbf{e}_2$ (all other external loads are zero). Using as reference the starter code below (cf. "fill me here" comments for missing items) do the following:

1. Write a function for the local stiffness matrix of a frame element (combined axial and bending), following the template below
`def local_stiffness_frame(elts,crds,e) .`

```
# Define the element properties
elements = {};
elements[0]={'A':1.0,'E':1.0, 'I':1.0, 0:0, 1:1}
elements[1]={'A':2.0,'E':2.0, 'I':2.0, 0:1, 1:2}
# Define the coordinates
coordinates = np.array([[0.0,0.0],[0.0,10.0],[10.0,10.0]])
# Define element degree of freedom
ele_dof = 2
# total number of elements
nel = 2

def local_stiffness_frame(elts, crds, e):
        A, E, I = (elts[e]['A'], elts[e]['E'], elts[e]['I'])
        n = np.array(crds[elts[e][1]])-np.array(crds[elts[e][0]])
        L = LA.norm(n)
        n /= L
        R = np.array([[0,-1],[1,0]])
        s = np.dot(R,n)
        Kfw = A*E/L*np.outer(n,n)+12*E*I/(L**3)*np.outer(s,s)
        kmt = 4*E*I/L
        khmt = 2*E*I/L
        kmw = 6*E*I/(L**2)*s
        kft = 6*E*I/(L**2)*s
        space_dim = n.size
        n_nodes = 2
        n_dof = space_dim*n_nodes+n_nodes
        Ke = np.zeros((n_dof, n_dof))
        # Diagonal terms
        Ke[0:2,0:2] = Ke[3:5,3:5] = Kfw
        Ke[2,2] = Ke[5,5] = kmt
        # Non-diagonal terms
        Ke[0:2,2] = Ke[0:2,5] =kft
        Ke[0:2,3:5] = -Kfw
        Ke[2,3:5] = -kmw
        Ke[2,5] = khmt
        Ke[3:5,5] = -kft
        lower_indices = np.tril_indices(n_dof,-1)
        Ke[lower_indices] = 0.
        Ke += np.triu(Ke,1).T
        return Ke
```

2. Assemble the global stiffness matrix and load vector.

```
nnodes = 3
space_dim = 2
local_dof = space_dim+(space_dim-1)
#each node has space_dim number of translation + (space_dim-1) number of rotation
num_dof = space_dim*nnodes+nnodes*(space_dim-1)
KG = np.zeros((num_dof,num_dof))

# Loop over all elements
for e in range(nel):
        # Obtain the element stiffness matrix
        KE = local_stiffness_frame(elements, coordinates, e)
        # Assemble the global stiffness matrix
        for p in range(ele_dof):
                global_p = elements[e][p]
                for q in range(ele_dof):
                        global_q = elements[e][q]
                        KG[global_p*local_dof:(global_p+1)*local_dof,global_q*local_dof\
                        :(global_q+1)*local_dof] \
                        += KE[p*local_dof:(p+1)*local_dof,q*local_dof:(q+1)*local_dof]
```

3. Knowing that $\boldsymbol{w}_1 = \mathbf{0}$ and $\theta_1 = 0$ as well as $\boldsymbol{m}_S = -cos(\pi/4)\mathbf{e}_1 + \sin(\pi/4)\mathbf{e}_2$, apply the appropriate boundary conditions.

```
ms = np.array([-np.cos(np.pi/4),np.sin(np.pi/4)])
row = np.append(np.zeros(6),ms)
row = np.append(row,0.)
row = np.resize(row,(1,num_dof))
col = np.append(np.zeros(6),-ms)
col = np.append(col,0.)
col = np.append(col,0.)
col = np.resize(col,(num_dof+1,1))

K_new = KG.copy()
K_new = np.vstack([K_new, row])
K_new = np.hstack([K_new, col])

# Nodes of known displacement
# Set one if known else 0
bc = np.zeros(num_dof+1)
bc = [1,1,1,0,0,0,0,0,0,0]

# Given load
P = np.zeros(len(bc))
P[3] = 10.
P[4] = 5.0
P[5] = 3.0
P[6] = 2.0
P[7] = 5.0

# Dirichlet Boundary conditions
g = np.zeros(len(bc))
# Updated force matrix
F = np.zeros(len(bc))
#F = np.array([[u1, P, u3, u4]])
# Initialize a new matrix with KG values
K = K_new.copy()
# Updated Stiffness matrix
for b in range(len(bc)):
        for num in range(len(bc)):
                if bc[b] == 1:
                        if b == num:
                                K[b,num] = 1.0
                        else:
                                K[b,num] = 0.0
# Updated Stiffness matrix
for b in range(len(bc)):
        if bc[b] == 1:
                F[b] = g[b]
        else:
                F[b] = P[b]
```

4. Solve for the displacements and rotations.

Solution :
u = LA.solve(K,F.T)

$$\boldsymbol{w}_1 = [822.3, 106.3]$$
$$\theta_1 = [25.99]$$
$$\boldsymbol{w}_2 = [825.7, 825.7]$$
$$\theta_2 = [95.41]$$

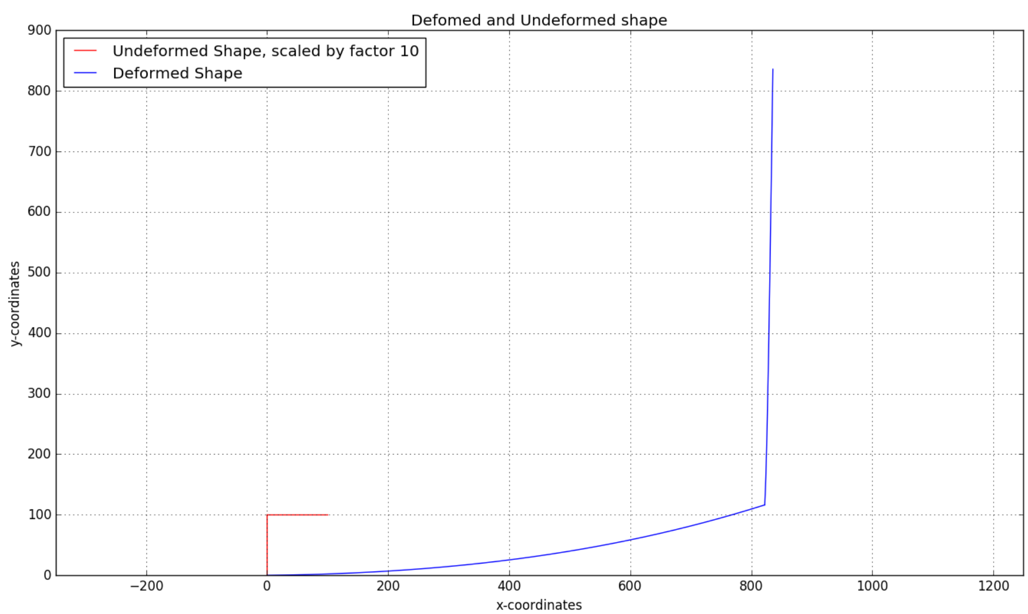
R = np.dot(K_{new},u)

$$\boldsymbol{V}_1 = [-11.37, -10.63]$$
$$M_1 = [54.33]$$

λ obtained from the u vector

$$\lambda = 0.896$$

5. Plot the deformed shape of the frame.




```
elements = {}
elements[ ] = {'A': , 'E':  , 'I': , 0: , 1: } #fill me here

def local_stiffness_truss(elts,crds,e):
A, E = (elts[e]['A'],elts[e]['E'])
# Compute the director vector between the nodes
n = crds[elts[e][1]] - crds[elts[e][0]]
# Compute the lenght of the element
```

```
L = la.norm(n)
# Normalize the director vector
n /= L
# Compute the stiffness tensor
k = A*E/L*np.outer(n,n)
# The individual tensor entries of the local element stiff
ke = np.array([[k,-k],[-k,k]])
# Resize it
space_dim = n.size # the space dimensions
n_nodes = 2 # the number of nodes
n_dof = space_dim*n_nodes # the number of element degrees of freedom
ke = ke.reshape(n_dof, n_dof )
return ke


# Assemble the local stiffness matrix for a frame element
def local_stiffness_beam(elts,crds,e):
# Get element properties
A, E, I = (elts[e]['A'],elts[e]['E'],elts[e]['I'])
# Compute the director vector between the nodes
n = crds[elts[e][1]] - crds[elts[e][0]]
# Compute the lenght of the element
L = la.norm(n)
# Normalize the director vector
n /= L
# Define the rotation operation
R = np.array([[0,-1],[1,0]])
# Compute normal
s = # fill me here
# Compute the coefficients
Kfw =  # fill me here
kmt =  # fill me here
khmt =  # fill me here
kmw =  # fill me here
kft =  # fill me here
# The elt stiffness
space_dim = n.size # the space dimensions
n_nodes = 2 # the number of nodes
n_dof = space_dim*n_nodes + n_nodes # the number of element degrees of freedom
Ke = np.zeros((n_dof,n_dof))
# Add contributions
# The Diagonal blocks
Ke[0:2,0:2] = Ke[3:5,3:5] = Kfw
Ke[2,2] = Ke[5,5] = kmt
# The upper triangular portion
Ke[0:2,2] = Ke[0:2,5] =  kft
Ke[0:2,3:5] = Ke[0:2,3:5] = -Kfw
Ke[2,3:5] = -kmw
Ke[2,5] = khmt
Ke[3:5,5] = -kft
# Copy upper triangular to lower triangular (it's a symmetric matrix)
lower_indeces = np.tril_indices(n_dof,-1)
```

```
Ke[lower_indeces] = 0
Ke += np.triu(Ke,1).T
return Ke
```

PROBLEM 3

Consider the strong form of the bending of a beam to be given by: find $u : \Gamma \to \mathbb{R}$, with $\Gamma = [0, \ell]$ such that

$$EI\frac{d^4u}{dx^4} = f \quad \forall x \in \Gamma$$

and

$$u(0) = \frac{du}{dx}(0) = 0, \quad EI\frac{d^2u}{dx^2}(\ell) = m,\ EI\frac{d^3u}{dx^3}(\ell) = -t,$$

where t, m are given quantities.

1. Give a physical interpretation of the boundary value problem given above.

   Solution :
   The above boundary value problem could be interpreted as a beam clamped at $x = 0$; subject to an external load which produces moment m and shear t at $x = \ell$

2. Identify the Dirichlet and Neumann boundary conditions.

   Solution :
   Dirichlet BC :

   $$u(0) = 0$$
   $$\frac{du}{dx}(0) = 0$$

   Neumann BC:

   $$EI\frac{d^2u}{dx^2}(\ell) = m$$
   $$EI\frac{d^3u}{dx^3}(\ell) = -t$$

3. What is the set of trial functions $\mathcal{S}$?

   Solution :
   The set of trial functions $\mathcal{S}$:

   $$\mathcal{S} = \left\{ u | u \in H^2\left([0, \ell]\right), u(0) = 0, \frac{du}{dx}(0) = 0 \right\}$$

4. What is the set of test functions $\mathcal{V}$?

   Solution :
   The set of test functions $\mathcal{V}$:

   $$\mathcal{V} = \left\{ w | w \in H^2\left([0, \ell]\right), w(0) = 0, \frac{dw}{dx}(0) = 0 \right\}$$

5. Derive the weak form using the method of weighted residuals, and simplifying it to the point when the maximum number of derivatives on the test functions is equivalent to the one on the trial functions.

Solution :
The strong form of the equation is given by:

$$EI\frac{d^4u}{dx^4} = f \ \ \forall x \in \Gamma$$

Multiplying both the sides by the weight w and integrating:

$$\begin{aligned}
&\int_\Gamma EI\frac{d^4u}{dx^4} \ \ w \ \ dx = \int_\Gamma f \ \ w \ \ dx \\
&\left[EI\frac{d^3u}{dx^3}w\right]\Big|_0^\ell - \int_\Gamma EI\frac{d^3u}{dx^3}\frac{dw}{dx} \ \ dx = \int_\Gamma f \ \ w \ \ dx \\
&-tw(\ell) - \left[EI\frac{d^2u}{dx^2}\frac{dw}{dx}\right]\Big|_0^\ell + \int_\Gamma EI\frac{d^2u}{dx^2}\frac{d^2w}{dx^2} \ \ dx = \int_\Gamma f \ \ w \ \ dx \\
&-tw(\ell) - m\frac{dw}{dx}(\ell) + \int_\Gamma EI\frac{d^2u}{dx^2}\frac{d^2w}{dx^2} \ \ dx = \int_\Gamma f \ \ w \ \ dx
\end{aligned}$$

6. $\star$ Let the potential functional be given by

$$\Pi[u] = \int_\Gamma \frac{EI}{2}\left(\frac{d^2u}{dx^2}\right)^2 dx - \int_\Gamma f\, u dx - m\frac{du}{dx}(\ell) - tu(\ell).$$

Show that the statement

$$\langle \delta\Pi, \delta u\rangle = 0, \quad \forall \delta u \in \mathcal{V}$$

is equivalent to the weak form derived with the method of weighted residuals.

Solution :
The potential function is given by:

$$\Pi[u] = \int_\Gamma \frac{EI}{2}\left(\frac{d^2u}{dx^2}\right)^2 \ \ dx - \int_\Gamma f \ \ u \ \ dx - m\frac{du}{dx}(\ell) - tu(\ell)$$

$$\langle \delta\Pi, \delta u\rangle = \left.\frac{d\Pi[u^*]}{d\alpha}\right|_{\alpha=0} = 0$$

where $u^* = u + \alpha w$ where $w \in \mathcal{V}$ and $\alpha \in \mathbb{R}$

$$\langle \delta\Pi, \delta u\rangle = \left.\frac{d\Pi[u^*]}{d\alpha}\right|_{\alpha=0}$$

$$\begin{aligned}
0 &= \frac{d}{d\alpha}\left.\left(\int_\Gamma \frac{EI}{2}\left(\frac{d^2u^*}{dx^2}\right)^2 \ \ dx - \int_\Gamma f \ \ u^* \ \ dx - m\frac{du^*}{dx}(\ell) - tu^*(\ell)\right)\right|_{\alpha=0} \\
0 &= \frac{d}{d\alpha}\left.\left(\int_\Gamma \frac{EI}{2}\left(\frac{d^2(u+\alpha w)}{dx^2}\right)^2 \ \ dx - \int_\Gamma f \ \ (u+\alpha w) \ \ dx - m\frac{d(u+\alpha w)}{dx}(\ell) - t(u+\alpha w)(\ell)\right)\right|_{\alpha=0} \\
0 &= \left.\left(\int_\Gamma EI\frac{d^2(u+\alpha w)}{dx^2}\frac{d^2w}{dx^2} \ \ dx - \int_\Gamma f \ \ w \ \ dx - m\frac{dw}{dx}(\ell) - tw(\ell)\right)\right|_{\alpha=0} \\
0 &= \int_\Gamma EI\frac{d^2u}{dx^2}\frac{d^2w}{dx^2} \ \ dx - \int_\Gamma f \ \ w \ \ dx - m\frac{dw}{dx}(\ell) - tw(\ell)
\end{aligned}$$

which is equivalent to the weak form derived using the method of weighted residuals.