

# 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 \*, but are encouraged to attempt them anyhow.

## PROBLEM 1

Consider the frame shown below. For each node  $z = 1, 2, 3$  we have associated coordinates  $\mathbf{q}_z$  and associated global degrees of freedom  $\mathbf{u}_z$  and  $\theta_z$ , where both  $\mathbf{q}$  and  $\mathbf{u}$  are vectors while  $\theta_z$  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  $\mathbf{m}_s$ .

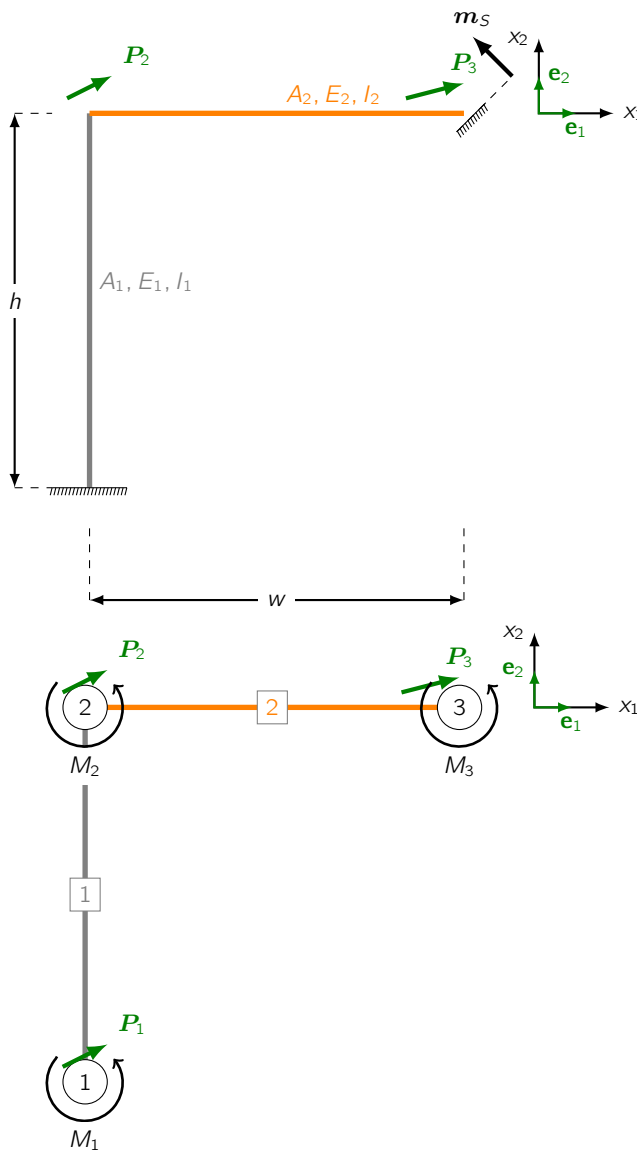


Figure 1: The system of uniaxial rods

1. For each node write the equilibrium equations in terms of the external forces  $\mathbf{P}_k$ ,  $k = 1, 2, 3$  and moments  $M_k$ ,  $k = 1, 2, 3$ , and the internal forces  $\mathbf{f}_{ij}^e$  and moments  $m_{ij}^e$ .
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*.
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.
4. Using  $\mathbf{K}_{fw}^e, \mathbf{k}_{f\theta}^e, \dots$  (cf. lecture notes), write down the equilibrium equations in matrix form.
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  $\mathbf{m}_S$  as well as to rotate freely. Apply the aforementioned conditions to the matrix form of the previous step.

## 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  $\mathbf{P}_2 = 10\mathbf{e}_1 + 5\mathbf{e}_2$  and  $M_2 = 3$ , as well as  $\mathbf{P}_3 = 2\mathbf{e}_1 + 5\mathbf{e}_2$  (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) .`
2. Assemble the global stiffness matrix and load vector.
3. Knowing that  $\mathbf{w}_1 = \mathbf{0}$  and  $\theta_1 = 0$  as well as  $\mathbf{m}_S = -\cos(\pi/4)\mathbf{e}_1 + \sin(\pi/4)\mathbf{e}_2$ , apply the appropriate boundary conditions.
4. Solve for the displacements and rotations.
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  $u : \Gamma \rightarrow \mathbb{R}$ , with  $\Gamma = [0, \ell]$  such that

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

and

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

where  $t$ ,  $m$  are given quantities.

1. Give a physical interpretation of the boundary value problem given above.
2. Identify the Dirichlet and Neumann boundary conditions.
3. What is the set of trial functions  $\mathcal{S}$ ?
4. What is the set of test functions  $\mathcal{V}$ ?
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.
6. ★ Let the potential functional be given by

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

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.