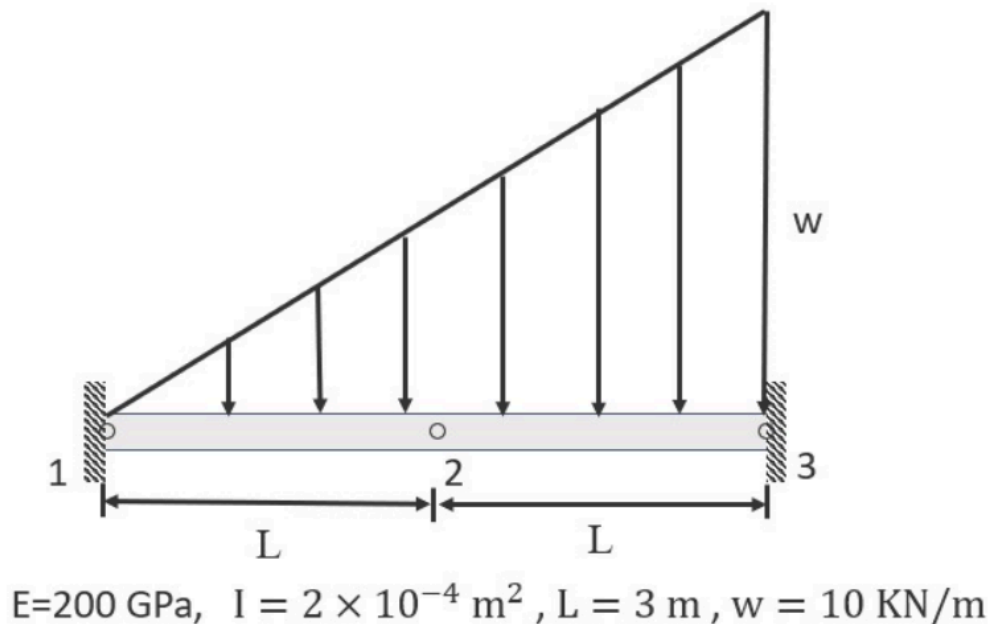


✓ Problem 1

1 solution submitted (max: 3) | [View my solutions](#)



Consider the problem shown in the figure. Note that there are 3 nodes and 2 elements. The material properties are $E = 200 \text{ GPa}$, $I = 2 \times 10^{-4} \text{ m}^2$ and $L = 3 \text{ m}$. The triangular load applied has $w = 10 \text{ kN/m}$. The element stiffness matrix is

$$\frac{EI}{L} \begin{pmatrix} 12/L^2 & 6/L & -12/L^2 & 6/L \\ & 4 & -6/L & 2 \\ & & 12/L^2 & -6/L \\ & & & 4 \end{pmatrix}$$

You need to find the nodal equivalent loads for the two elements separately. The first element is subjected to a triangular load of maximum intensity $w/2$, while the second element has a uniformly distributed load of intensity $w/2$ plus a triangular load of maximum intensity $w/2$. Equivalent loads from neighbouring elements should be added at a node as we have discussed in the class. The global equivalent loads coming from the two elements have to be stored in F , while the equivalent loads from the individual elements must be stored in F_{local} . The stiffness matrix is stored in K_{global} before applying boundary conditions and in K_{prime} after applying them.

After solving for the nodal dof's (use Matlab's `inv` function to invert the global stiffness K_{prime}), you need to find the reaction forces and moments at all the nodes.

Also, use the displacement solution to find the reaction forces F_{reac} .

```
% problem 4a: problem with beam elements
% built in beams with combination of distributed and concentrated loads
coor=[0 3 6]; %nodal coordinates
conn=[1 2;2 3];
dest=[1 2 3 4;3 4 5 6];
nelem= size(conn,1); %number of elements
nnode = size(coor,1); %number of nodes
% material properties
E = 210e09;
```

```

I = 2e-04;
EI=E*I;
% loads
w=10e03;
% form stiffness matrix and assemble

nnode = length(coor);
ndof=2*nnode;    %no of dof

Kglobal=zeros(ndof,ndof);
F=zeros(ndof,1);

for ielem=1:nelem
    n1=conn(ielem,1);    n2=conn(ielem,2);

    x1=coor(n1); x2=coor(n2);    %global coordinates of each element
    l=x2-x1;
    L=l;

    % local distributed force equivalent at nodes
    Klocal=EI/l*[ 12/l^2    6/l    -12/l^2    6/l
                  6/l      4      -6/l      2
                 -12/l^2   -6/l    12/l^2   -6/l
                  6/l      2      -6/l      4    ];
    Kloc=Klocal;

    % local distributed force equivalent at nodes

    if ielem == 1
        q1 = 0;          q2 = -w/2;          % element 1: triangular up to w/2
    end
    if ielem == 2
        q1 = -w/2;       q2 = -w;           % element 2: UDL w/2 + triangular
w/2
    end
    Flocal = [ L*(7*q1+3*q2)/20;
               L^2*(3*q1+2*q2)/60;
               L*(3*q1+7*q2)/20;
               -L^2*(2*q1+3*q2)/60 ];
    Floc=Flocal;

```

```

    d = [2*n1-1 2*n1 2*n2-1 2*n2];
    Kglobal(d,d) = Kglobal(d,d) + Kloc;
    F(d)          = F(d) + Floc;
end

% apply boundary conditions
% apply boundary conditions (built-in at nodes 1 and 3)
bc_dofs = [1 2 5 6];
u_presc = zeros(ndof,1);

Fprime = F;
Kgprime = Kglobal;
for d = bc_dofs
    Kgprime(d,:) = 0; Kgprime(:,d) = 0; Kgprime(d,d) = 1;
    Fprime(d) = u_presc(d);
end
% solve
U = inv(Kgprime)*Fprime;
% determine reaction forces
    % U=[v1, th1, v2, th2, v3, th3]^T

Freac = Kglobal*U - F

```

