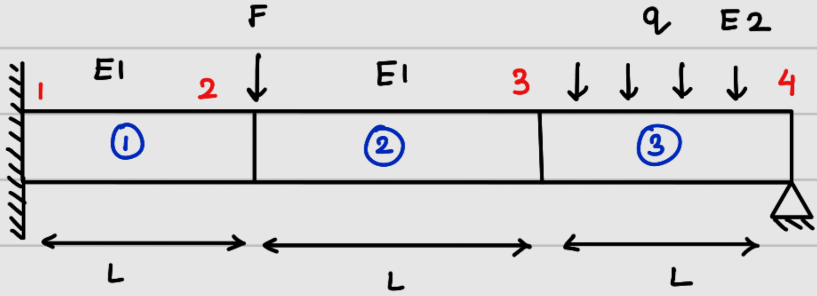
CH20B012

Anish Anand Pophale

AM5450 Assignment 5

Q1) Consider the beam divided into 3 elements (4 nodes) as shown below,



The interface is at node 3

$\therefore$ we need to find v_3 and θ_3

From the boundary conditions, $v_1 = \theta_1 = v_4 = 0$

All further calculations are done using the MATLAB code given below.

Q1)

MATLAB Code:

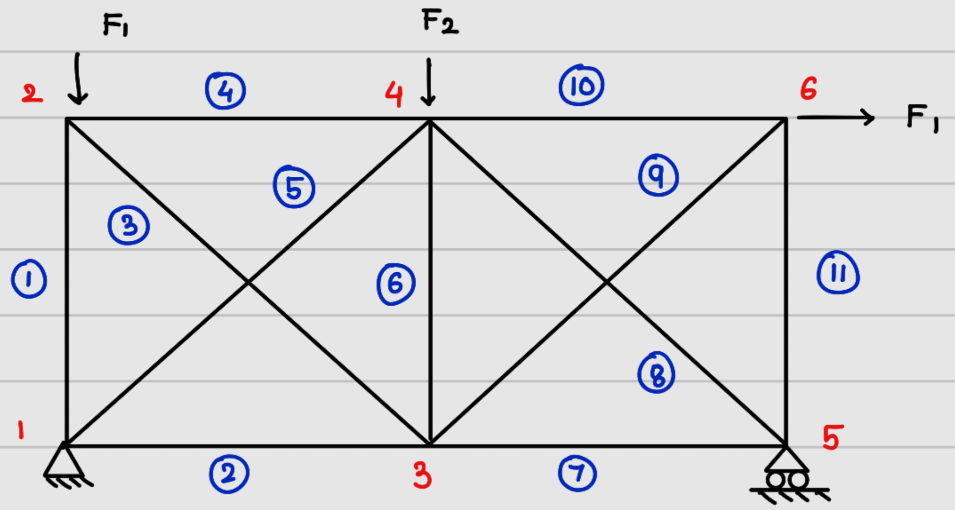
```
clc; clear all; close all; format compact; format shortg;

%Constants
E1 = 200*10^9;
E2 = 70*10^9;
L = 2;
q = -10*10^3; %q is in downward direction (-ve Y direction)
F = -18*10^3; %F is in downward direction (-ve Y direction)
I = 4*10^-4;

%Stiffness matrices for each element
k1 = stiffness(E1,I,L);
k2 = stiffness(E1,I,L);
k3 = stiffness(E2,I,L);

%Global Stiffness Matrix
Kglobal = zeros(8,8);
Kglobal(1:4,1:4) = Kglobal(1:4,1:4) + k1;
Kglobal(3:6,3:6) = Kglobal(3:6,3:6) + k2;
Kglobal(5:8,5:8) = Kglobal(5:8,5:8) + k3;

%Global point and distributed load vectors
rF = zeros(8,1);
rF(3) = F;

rq = zeros(8,1);
rq(5:8) = [q*L/2; q*L^2/12;q*L/2; -q*L^2/12];

%Removing rows and columns with EBC (u and theta are known)
Kglobal_r = Kglobal;
Kglobal_r([1,2,7],:) = [];
Kglobal_r(:,[1,2,7]) = [];
rq_r = rq;
rq_r([1,2,7]) = [];
rF_r = rF;
rF_r([1,2,7]) = [];

%Solving for d vector
d_r = Kglobal_r\(rq_r + rF_r);
d = zeros(8,1);
d(3:6) = d_r(1:4);
d(end) = d_r(end);

fprintf("The deflection at the interface is %f m\n",d(5))
fprintf("The slope of the beam at the inteface is %f\n",d(6))

%Functon to calculate stiffness matrix of beam element
function k = stiffness(E,I,L)
    k = ((E*I)/(L^3))*[12, 6*L, -12, 6*L; 6*L, 4*L^2, -6*L, 2*L^2; -12, -6*L, 12,
-6*L; 6*L, 2*L^2, -6*L, 4*L^2];
end
```

Output:

```
Command Window
The deflection at the interface is -0.000854 m
The slope of the beam at the inteface is -0.000030
fx >>
```

Therefore, the deflection of the beam at the interface is calculated as **-0.000854 m**.

Q2) The nodes and elements in the given truss structure are numbered as follows -



∴ There are 6 nodes and 11 elements.

DOF / node = 2, ∴ Total DOF = $6 \times 2 = 12$

Using EBC, $u_1 = v_1 = v_5 = 0$

All further calculations are done using the MATLAB code given below.

Q2)

MATLAB Code:

```
clc; clear all; close all; format compact; format shortg;

%Constants
L = 0.5;
E = 200*10^9;
d = 20*10^-3; A = (pi*d^2)/4;
F1 = 10*10^3; F2 = 12*10^3;

nodes = [0,0; 0,L; L,0; L,L; 2*L,0; 2*L,L]; %Node coordinates
conn = [1,2; 1,3; 2,3; 2,4; 1,4; 3,4; 3,5; 4,5; 3,6; 4,6; 5,6]; %Connectivity
N = size(conn,1); %Number of elements;
n = size(nodes,1); %Number of nodes;

Kglobal = zeros(2*n,2*n);
%Constructing the global stiffness matrix
for i = 1:N
    node1 = conn(i,1);
    node2 = conn(i,2);

    x1 = nodes(node1,1); y1 = nodes(node1,2);
    x2 = nodes(node2,1); y2 = nodes(node2,2);

    k = stiffness(x1,y1,x2,y2,E,A);

    Kglobal(2*node1-1:2*node1,2*node1-1:2*node1) = Kglobal(2*node1-
1:2*node1,2*node1-1:2*node1) + k(1:2,1:2);
    Kglobal(2*node1-1:2*node1,2*node2-1:2*node2) = Kglobal(2*node1-
1:2*node1,2*node2-1:2*node2) + k(1:2,3:4);
    Kglobal(2*node2-1:2*node2,2*node1-1:2*node1) = Kglobal(2*node2-
1:2*node2,2*node1-1:2*node1) + k(3:4,1:2);
    Kglobal(2*node2-1:2*node2,2*node2-1:2*node2) = Kglobal(2*node2-
1:2*node2,2*node2-1:2*node2) + k(3:4,3:4);
end

%Constructing the global force vector
Fglobal = zeros(2*n,1);
Fglobal(4) = -F1;
Fglobal(8) = -F2;
Fglobal(11) = F1;

%Removing part of the matrices which are not solved for (EBC)
Kglobal_r = Kglobal; Fglobal_r = Fglobal;
Kglobal_r([1,2,10],:) = [];
Kglobal_r(:,[1,2,10]) = [];
Fglobal_r([1,2,10]) = [];

%Solving for the displacements
d_r = Kglobal_r\Fglobal_r;
%Adding the known u values from EBC to the solution vector
d = zeros(n,1);
d(3:9) = d_r(1:7);
d(11:12) = d_r(8:9);
```

```
u = zeros(n,1); v = zeros(n,1);
for i = 1:n
    u(i) = d(2*i-1);
    v(i) = d(2*i);
end

fprintf("Nodal Displacements:\n")
T1 = table((1:1:n)',u,v,VariableNames=["Node","u (m)","v (m)"]);
disp(T1);

%Calculating the reactions at supports
R1x = Kglobal(1,:)*d; %Reaction at node 1 in X direction
R1y = Kglobal(2,:)*d; %Reaction at node 1 in Y direction
R5y = Kglobal(10,:)*d; %Reaction at node 5 in Y direction

fprintf("\nReaction Forces:\n")
fprintf("Reaction force in X direction at node 1 = %f N\n",R1x)
fprintf("Reaction force in Y direction at node 1 = %f N \n",R1y)
fprintf("Reaction force in Y direction at node 5 = %f N \n\n",R5y)

%New node positions
nodes_new = zeros(n,2);
for i = 1:n
    nodes_new(i,1) = nodes(i,1) + d(2*i-1);
    nodes_new(i,2) = nodes(i,2) + d(2*i);
end

%Calculating stress and force in each element
stress_vec = zeros(N,1);
force_vec = zeros(N,1);
for i = 1:N
    node1 = conn(i,1);
    node2 = conn(i,2);

    x1 = nodes(node1,1); y1 = nodes(node1,2);
    x2 = nodes(node2,1); y2 = nodes(node2,2);

    u1 = d(2*node1-1); v1 = d(2*node1);
    u2 = d(2*node2-1); v2 = d(2*node2);

    stress_vec(i) = element_stress(x1,y1,x2,y2,u1,v1,u2,v2,E);
    force_vec(i) = A*stress_vec(i);
end

fprintf("Elemental Stress and Forces:\n")
T2 = table((1:1:N)',stress_vec,force_vec,VariableNames=["Element","Stress
(Pa)","Force (N)"]);
disp(T2)

%Plotting the old and new truss structure
figure;
hold on;
for i = 1:N
    node1 = conn(i,1);
    node2 = conn(i,2);

    x1 = nodes(node1,1); y1 = nodes(node1,2);
    x2 = nodes(node2,1); y2 = nodes(node2,2);
```

```
    plot([x1,x2],[y1,y2],'-.r',LineWidth=1.5)

    x1_new = nodes_new(node1,1); y1_new = nodes_new(node1,2);
    x2_new = nodes_new(node2,1); y2_new = nodes_new(node2,2);

    plot([x1_new,x2_new],[y1_new,y2_new],'--.b',LineWidth=1.5)
end
title("Undeformed (Red) and Deformed (Blue) Truss Structure")
xlabel("x (m)")
ylabel("y (m)")

%function to calculate the stiffness matrix in GCS
function k = stiffness(x1,y1,x2,y2,E,A)
    l = sqrt((x1-x2)^2 + (y1-y2)^2);
    c = (x2-x1)/l;
    s = (y2-y1)/l;

    T = [c,s,0,0; 0,0,c,s]; %Transformation matrix
    kl = (E*A/l)*[1,-1; -1,1];
    k = T'*kl*T;
end

%function to calculate stress and strain in each element
function s = element_stress(x1,y1,x2,y2,u1,v1,u2,v2,E)
    l = sqrt((x1-x2)^2 + (y1-y2)^2);
    c = (x2-x1)/l;
    s = (y2-y1)/l;

    T = [c,s,0,0; 0,0,c,s]; %Transformation matrix
    d = [u1;v1;u2;v2];
    dl = T*d;
    s = E*(dl(2)-dl(1))/l;
end
```

Output:

```
Command Window
Nodal Displacements:
    Node        u (m)          v (m)
    ____    __________    __________

     1               0             0
     2      0.00011511    -7.8271e-05
     3      8.8841e-05    -0.00010085
     4      0.00011642    -0.00014262
     5       0.0001333             0
     6      0.00015292    -4.3079e-05

Reaction Forces:
Reaction force in X direction at node 1 = -10000.000000 N
Reaction force in Y direction at node 1 = 11000.000000 N
Reaction force in Y direction at node 5 = 11000.000000 N

Elemental Stress and Forces:
    Element    Stress (Pa)    Force (N)
    _______    ___________    _________

       1       -3.1309e+07     -9835.9
       2        3.5537e+07       11164
       3       -7.3886e+05     -232.12
       4        5.2245e+05      164.13
       5       -5.2404e+06     -1646.3
       6       -1.6709e+07     -5249.3
       7        1.7783e+07      5586.6
       8       -2.5148e+07     -7900.6
       9        2.4369e+07      7655.8
      10        1.4599e+07      4586.6
      11       -1.7232e+07     -5413.4
fx >>
```

Plots:

- The nodal displacements are very small, so the deformation in the truss structure is not clearly visible in Fig1.
- So, in Fig2, the value of E is reduced to 2 GPa so that the deformation increases and is clearly visible

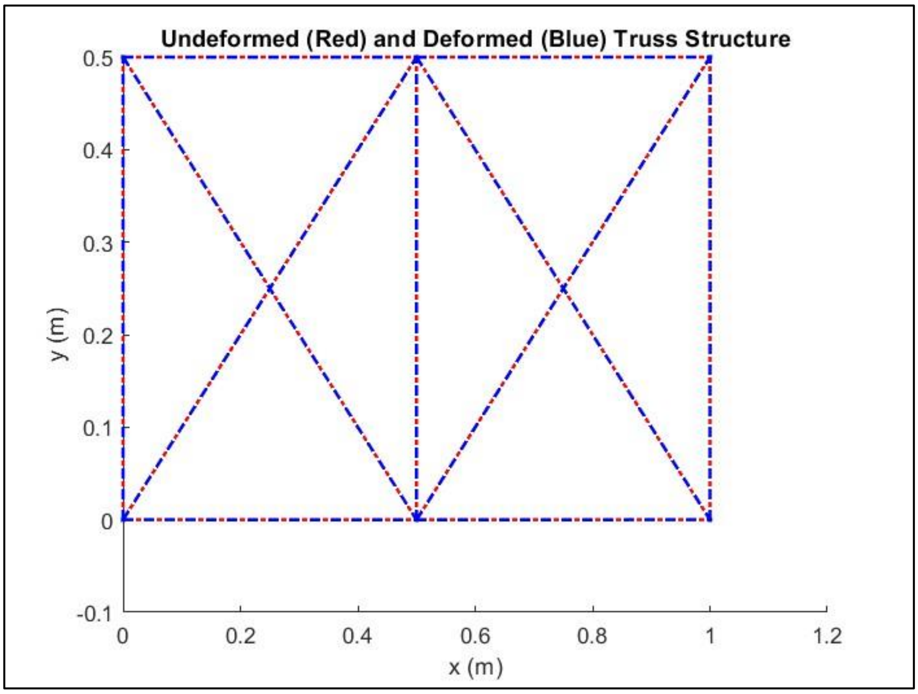


Fig 1: Deformed and undeformed truss structure

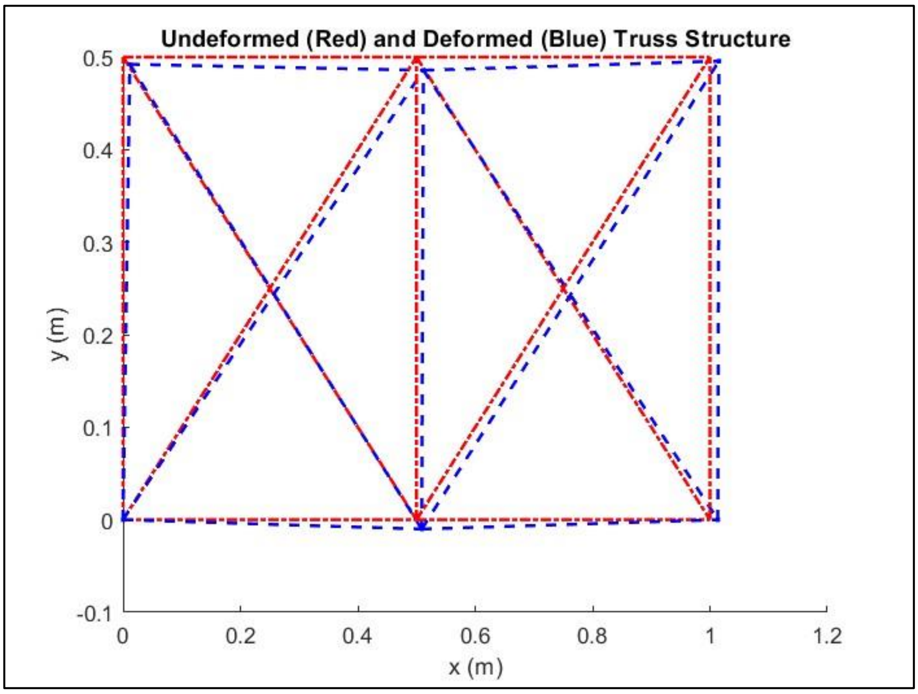


Fig 2: Deformed and undeformed truss structure using E = 2 GPa