Submitted by Hasan (Prospective Ph.D. student)

Previously, I submitted the truss hand calculation (FEM) for Reaction force, deflection, and stress in each element, **deformed and undeformed plot**. This is the corresponding code for the Hand calculation. (direct stiffness method).

Four Matlab files complete the whole function:

1. local_stiffness_matrix (local nodal stiffness matrix)

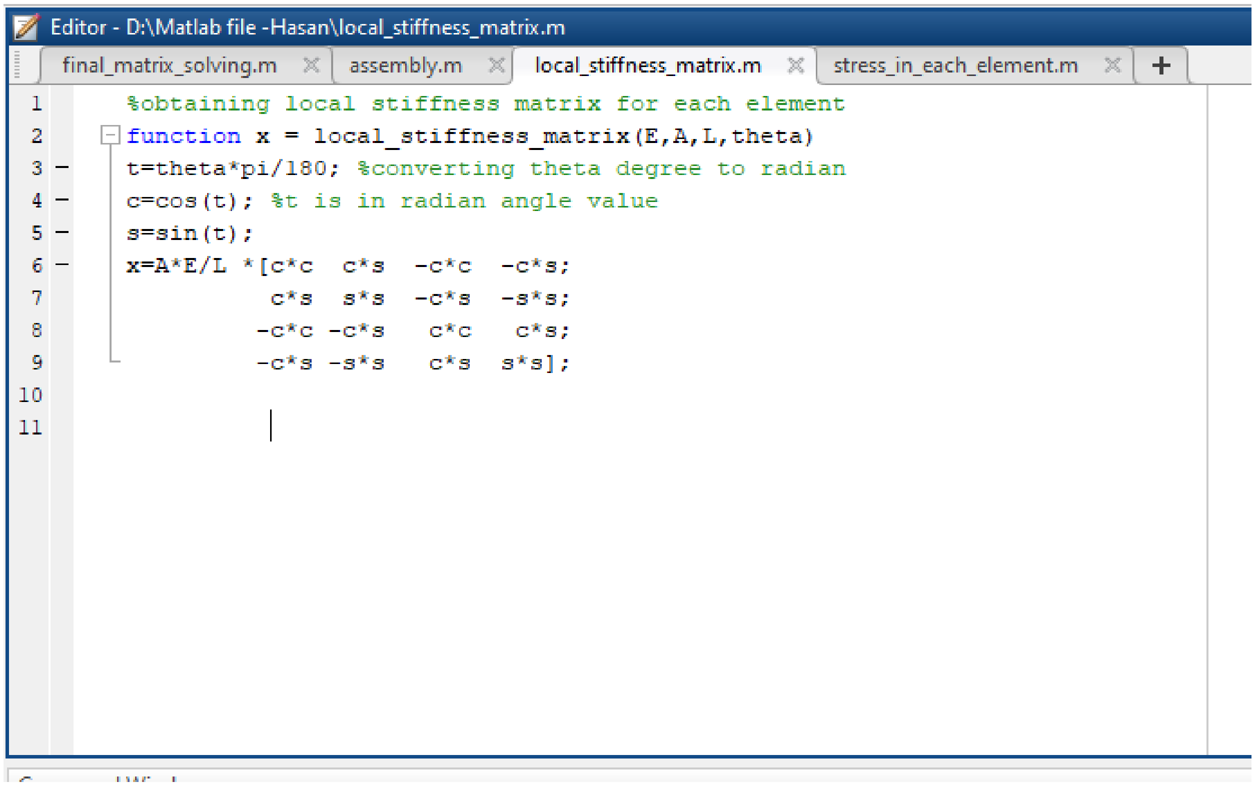
2. assembly (global stiffness matrix from local stiffness matrix)

3. stress_in_each_element (calculation of the stress in each element)

4. final_matrix_solving (getting nodal deflection and reaction force, deformed vs undeformed plot)

1. local_stiffness_matrix

```
%obtaining local stiffness matrix for each element
function x = local_stiffness_matrix(E,A,L,theta)
t=theta*pi/180; %converting theta degree to radian
c=cos(t); %t is in radian angle value
s=sin(t);
x=A*E/L *[c*c  c*s  -c*c  -c*s;
          c*s  s*s  -c*s  -s*s;
         -c*c -c*s   c*c   c*s;
         -c*s -s*s   c*s   s*s];
```



```
%obtaining local stiffness matrix for each element
function x = local_stiffness_matrix(E,A,L,theta)
t=theta*pi/180; %converting theta degree to radian
c=cos(t); %t is in radian angle value
s=sin(t);
x=A*E/L *[c*c  c*s  -c*c  -c*s;
          c*s  s*s  -c*s  -s*s;
         -c*c -c*s   c*c   c*s;
         -c*s -s*s   c*s  s*s];
```

Figure 1: Local_stiffness_matrix code

2. assembly

%assembling global stiffness matrix

function y = assembly(K,k,i,j)

%i and j are the connceting node number of the element

%Here capital K value is 10*10 golbal stifness matix, small k(row,column)is the local stiffness corresponding position value

% every iteration, the global sitffness matrix will be updated

K(2*i-1, 2*i-1) = K(2*i-1, 2*i-1) + k(1,1);%for k1:i=3,j=1,K(2*3-1=5, 2*3-1=5) Ux3,Ux3

K(2*i-1, 2*i) = K(2*i-1, 2*i) + k(1,2);%for k1:i=3,j=1,K(2*3-1=5, 2*3 =6) Uy3,Ux3

K(2*i-1, 2*j-1) = K(2*i-1, 2*j-1) + k(1,3);%for k1:i=3,j=1,K(2*3-1=5, 2*1-1=1) Ux1,Ux3

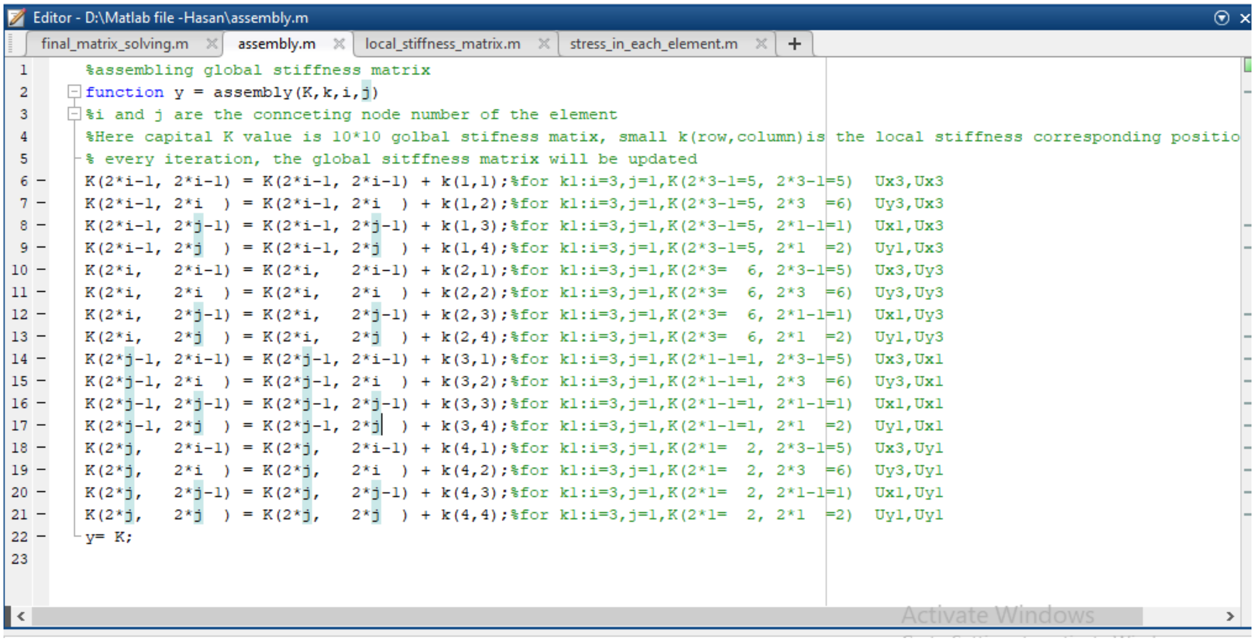
K(2*i-1, 2*j) = K(2*i-1, 2*j) + k(1,4);%for k1:i=3,j=1,K(2*3-1=5, 2*1 =2) Uy1,Ux3

K(2*i,   2*i-1) = K(2*i,   2*i-1) + k(2,1);%for k1:i=3,j=1,K(2*3= 6, 2*3-1=5) Ux3,Uy3

K(2*i,   2*i) = K(2*i,   2*i) + k(2,2);%for k1:i=3,j=1,K(2*3= 6, 2*3 =6) Uy3,Uy3

K(2*i,   2*j-1) = K(2*i,   2*j-1) + k(2,3);%for k1:i=3,j=1,K(2*3= 6, 2*1-1=1) Ux1,Uy3

```
K(2*i,   2*j  ) = K(2*i,   2*j  ) + k(2,4);%for k1:i=3,j=1,K(2*3=  6, 2*1  =2)  Uy1,Uy3

K(2*j-1, 2*i-1) = K(2*j-1, 2*i-1) + k(3,1);%for k1:i=3,j=1,K(2*1-1=1, 2*3-1=5)  Ux3,Ux1

K(2*j-1, 2*i  ) = K(2*j-1, 2*i  ) + k(3,2);%for k1:i=3,j=1,K(2*1-1=1, 2*3  =6)  Uy3,Ux1

K(2*j-1, 2*j-1) = K(2*j-1, 2*j-1) + k(3,3);%for k1:i=3,j=1,K(2*1-1=1, 2*1-1=1)  Ux1,Ux1

K(2*j-1, 2*j  ) = K(2*j-1, 2*j  ) + k(3,4);%for k1:i=3,j=1,K(2*1-1=1, 2*1  =2)  Uy1,Ux1

K(2*j,   2*i-1) = K(2*j,   2*i-1) + k(4,1);%for k1:i=3,j=1,K(2*1=  2, 2*3-1=5)  Ux3,Uy1

K(2*j,   2*i  ) = K(2*j,   2*i  ) + k(4,2);%for k1:i=3,j=1,K(2*1=  2, 2*3  =6)  Uy3,Uy1

K(2*j,   2*j-1) = K(2*j,   2*j-1) + k(4,3);%for k1:i=3,j=1,K(2*1=  2, 2*1-1=1)  Ux1,Uy1

K(2*j,   2*j  ) = K(2*j,   2*j  ) + k(4,4);%for k1:i=3,j=1,K(2*1=  2, 2*1  =2)  Uy1,Uy1

y= K;
```

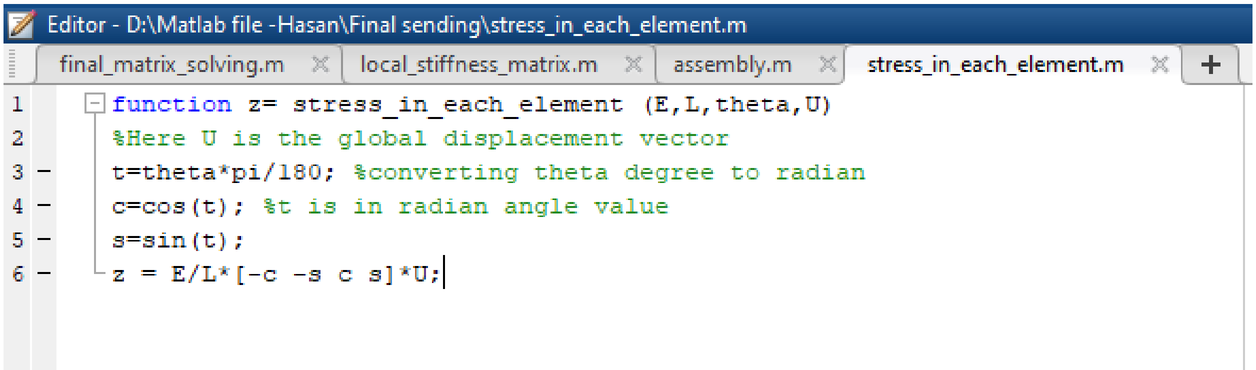


```
%assembling global stiffness matrix
function y = assembly(K,k,i,j)
%i and j are the connceting node number of the element
%Here capital K value is 10*10 golbal stifness matix, small k(row,column)is the local stiffness corresponding positio
% every iteration, the global sitffness matrix will be updated
K(2*i-1, 2*i-1) = K(2*i-1, 2*i-1) + k(1,1);%for k1:i=3,j=1,K(2*3-1=5, 2*3-1=5)  Ux3,Ux3
K(2*i-1, 2*i  ) = K(2*i-1, 2*i  ) + k(1,2);%for k1:i=3,j=1,K(2*3-1=5, 2*3  =6)  Uy3,Ux3
K(2*i-1, 2*j-1) = K(2*i-1, 2*j-1) + k(1,3);%for k1:i=3,j=1,K(2*3-1=5, 2*1-1=1)  Ux1,Ux3
K(2*i-1, 2*j  ) = K(2*i-1, 2*j  ) + k(1,4);%for k1:i=3,j=1,K(2*3-1=5, 2*1  =2)  Uy1,Ux3
K(2*i,   2*i-1) = K(2*i,   2*i-1) + k(2,1);%for k1:i=3,j=1,K(2*3=  6, 2*3-1=5)  Ux3,Uy3
K(2*i,   2*i  ) = K(2*i,   2*i  ) + k(2,2);%for k1:i=3,j=1,K(2*3=  6, 2*3  =6)  Uy3,Uy3
K(2*i,   2*j-1) = K(2*i,   2*j-1) + k(2,3);%for k1:i=3,j=1,K(2*3=  6, 2*1-1=1)  Ux1,Uy3
K(2*i,   2*j  ) = K(2*i,   2*j  ) + k(2,4);%for k1:i=3,j=1,K(2*3=  6, 2*1  =2)  Uy1,Uy3
K(2*j-1, 2*i-1) = K(2*j-1, 2*i-1) + k(3,1);%for k1:i=3,j=1,K(2*1-1=1, 2*3-1=5)  Ux3,Ux1
K(2*j-1, 2*i  ) = K(2*j-1, 2*i  ) + k(3,2);%for k1:i=3,j=1,K(2*1-1=1, 2*3  =6)  Uy3,Ux1
K(2*j-1, 2*j-1) = K(2*j-1, 2*j-1) + k(3,3);%for k1:i=3,j=1,K(2*1-1=1, 2*1-1=1)  Ux1,Ux1
K(2*j-1, 2*j  ) = K(2*j-1, 2*j  ) + k(3,4);%for k1:i=3,j=1,K(2*1-1=1, 2*1  =2)  Uy1,Ux1
K(2*j,   2*i-1) = K(2*j,   2*i-1) + k(4,1);%for k1:i=3,j=1,K(2*1=  2, 2*3-1=5)  Ux3,Uy1
K(2*j,   2*i  ) = K(2*j,   2*i  ) + k(4,2);%for k1:i=3,j=1,K(2*1=  2, 2*3  =6)  Uy3,Uy1
K(2*j,   2*j-1) = K(2*j,   2*j-1) + k(4,3);%for k1:i=3,j=1,K(2*1=  2, 2*1-1=1)  Ux1,Uy1
K(2*j,   2*j  ) = K(2*j,   2*j  ) + k(4,4);%for k1:i=3,j=1,K(2*1=  2, 2*1  =2)  Uy1,Uy1
y= K;
```

Figure 2:assembly of global sitffness matrix code

3.Stress in each elements

```
function z= stress_in_each_element (E,L,theta,U)
%Here U is the global displacement vector
t=theta*pi/180; %converting theta degree to radian
c=cos(t); %t is in radian angle value
s=sin(t);
z = E/L*[-c -s c s]*U;
```



```
function z= stress_in_each_element (E,L,theta,U)
%Here U is the global displacement vector
t=theta*pi/180; %converting theta degree to radian
c=cos(t); %t is in radian angle value
s=sin(t);
z = E/L*[-c -s c s]*U;
```

Figure 3: Stress in each elements code

4. Final_matrix_solving

```
%solving 10 degree of freedon struss reaction force at the hinged point.
clear; clc;
%declering all data, all in SI unit
A=.02; %cross sectional area of each element
E=100000000000;  %Youngs Modulous for all element
L1=2;theta1=60;%lenth of the first element is 2 & angle is 60 degree
L2=2;theta2=120; %lenth of the second element is 2 & angle is 120 degree
L3=3.01;theta3=4.67;%lenth of the third element is 2 & angle is 4.67 degree
L4=2.83;theta4=45;%lenth of the fourth element is 2.83 & angle is 45 degree
L5=2;theta5=90; %lenth of the fifth element is 2 & angle is 90 degree


% local_stiffness_matrix(E,A,L,theta)from individual function.
k1= local_stiffness_matrix(E,A,L1,theta1);%stiffness matrix for elemetn 1; node 3 to 1
k2= local_stiffness_matrix(E,A,L2,theta2);%stiffness matrix for elemetn 2;node 4 to 1
k3= local_stiffness_matrix(E,A,L3,theta3);%stiffness matrix for elemetn 3;node 1 to 2
k4= local_stiffness_matrix(E,A,L4,theta4);%stiffness matrix for elemetn 4;node 4 to 2
k5= local_stiffness_matrix(E,A,L5,theta5);%stiffness matrix for elemetn 5;node 5 to 2
fprintf('Local stiffness matrix')
k1
k2
k3
k4
k5
%ending of local stiffness matrix part

%assembly process of global stiffness matrix K= assembly (K,k,i,j)
%Here uppercase K is global and lower case k is local stiffness matirix
K=zeros(10,10); %5nodes*2=10 degree of freedom. so 10*10 zeros matrix
K=assembly(K,k1,3,1);%connect the stiffeness between node i=3 to j=1 or element 1
K=assembly(K,k2,4,1);%connect the stiffeness between node i=4 to j=1 or element 2
K=assembly(K,k3,1,2);%connect the stiffeness between node i=1 to j=2 or element 3
```

```
K=assembly(K,k4,4,2);%connect the stiffeness between node i=4 to j=2 or element 4
K=assembly(K,k5,5,2);%connect the stiffeness between node i=5 to j=2 or element 5
fprintf('Golobal stiffness matrix K')
K
%fprintf('10*10 global stifness matrix is %.4f\n', K);
%elmenating 5,6,7,8,9,10 row and columns and getting
%reduced_global_stifness matrix(M)
M=[K(1:4);K(2,1:4);K(3,1:4);K(4,1:4)];
fprintf('Boundary Condition')
f=[0;0;1000;0]%boundary conditions
u=M\f; %backslash operator used for gauss elimination technique, here u is deflection value of
node 1 and 2 (Ux1,Uy1,Ux2,uy2)
%10 deflection value of the nodes
fprintf('deflection matrix')
U=[u(1:4); 0;0;0;0;0;0]
%now for reaction force (R)of each point in x and y
%direction,R1x=R1y=R2x=R2y=0; only three hinged point have 6 reactin force.
fprintf('Reaction force')
R=K*U
%stress in each element (sigma1,-----,sigma5)
u1=[U(5);U(6);U(1);U(2)];%U3x=5,U3y=6,U1x=1,U1y=2 are the connecting node deflection
u2=[U(7);U(8);U(1);U(2)];%U4x=7,U4y=8,U1x=1,U1y=2 are the connecting node deflection
u3=[U(1);U(2);U(3);U(4)];%U1x=,U1y=2,U2x=3,U2y=4 are the connecting node deflection
u4=[U(7);U(8);U(3);U(4)];%U4x=7,U4y=8,U2x=3,U2y=4 are the connecting node deflection
u5=[U(9);U(10);U(3);U(4)];%U5x=9,U5y=10,U2x=3,U2y=4 are the connecting node deflection
fprintf('Stress in element 1')
sigma1=stress_in_each_element(E,L1,theta1,u1)
fprintf('Stress in element 2')
sigma2=stress_in_each_element(E,L2,theta2,u2)
fprintf('Stress in element 3')
sigma3=stress_in_each_element(E,L3,theta3,u3)
fprintf('Stress in element 4')
sigma4=stress_in_each_element(E,L4,theta4,u4)
fprintf('Stress in element 5')
sigma5=stress_in_each_element(E,L5,theta5,u5)
%--------------------------- Plot drawing-------------------------
%Plot structure
f1=figure();
Co_ordinate =[0 0; 3 0.268; -1 -1.732; 1 -1.732; 3 -1.732]; %node coordinate. node 1 is in center
point
Connection_point = [3 1; 4 1; 1 2; 4 2; 5 2]; %element conncting node
df = [1 2; 3 4; 5 6; 7 8; 9 10];%each node deflection
serial.1=Ux1;2=Uy1;3=Ux2;4=Uy2;5=Ux3;6=Uy3;7=Ux4;8=Uy4;9=Ux5;10=Uy5
NN = size(Co_ordinate,1); %no of nodes
NE = size(Connection_point,1);%no of elements
NCOORD = zeros(size(Co_ordinate));%deformed co-ordinate generation through zero matrix
scale = 100;
for n =1:NN
    NCOORD(n,1) = Co_ordinate(n,1) +scale*U(df(n,1));
    NCOORD(n,2) = Co_ordinate(n,2) +scale*U(df(n,2));
end

for k =1:NE
```

```
    i=Connection_point(k,1);%first starting point of local node
    j=Connection_point(k,2);%ending point of local node
    x=[Co_ordinate(i,1) Co_ordinate(j,1)];
    y=[Co_ordinate(i,2) Co_ordinate(j,2)];
    xlim([-1.5 4]);
    ylim([-2 1]);
    plot(x,y,'k-');
    hold on
    ux=[NCOORD(i,1) NCOORD(j,1)];
    uy=[NCOORD(i,2) NCOORD(j,2)];
    xlim([-1.5 4]);
    ylim([-2 1]);
    plot(ux,uy,'r--');
    hold on
end
```

Results

```
Local stiffness matrix
k1 =

   1.0e+08 *

    2.5000    4.3301   -2.5000   -4.3301
    4.3301    7.5000   -4.3301   -7.5000
   -2.5000   -4.3301    2.5000    4.3301
   -4.3301   -7.5000    4.3301    7.5000


k2 =

   1.0e+08 *

    2.5000   -4.3301   -2.5000    4.3301
   -4.3301    7.5000    4.3301   -7.5000
   -2.5000    4.3301    2.5000   -4.3301
    4.3301   -7.5000   -4.3301    7.5000


k3 =

   1.0e+08 *

    6.6005    0.5392   -6.6005   -0.5392
    0.5392    0.0440   -0.5392   -0.0440
   -6.6005   -0.5392    6.6005    0.5392
   -0.5392   -0.0440    0.5392    0.0440


k4 =

   1.0e+08 *

    3.5336    3.5336   -3.5336   -3.5336
    3.5336    3.5336   -3.5336   -3.5336
   -3.5336   -3.5336    3.5336    3.5336
   -3.5336   -3.5336    3.5336    3.5336


k5 =

   1.0e+09 *

    0.0000    0.0000   -0.0000   -0.0000
    0.0000    1.0000   -0.0000   -1.0000
   -0.0000   -0.0000    0.0000    0.0000
   -0.0000   -1.0000    0.0000    1.0000

Golobal stiffness matrix K
K =

   1.0e+09 *

  Columns 1 through 7

    1.1600    0.0539   -0.6600   -0.0539   -0.2500   -0.4330   -0.2500
    0.0539    1.5044   -0.0539   -0.0044   -0.4330   -0.7500    0.4330
   -0.6600   -0.0539    1.0134    0.4073         0         0   -0.3534
   -0.0539   -0.0044    0.4073    1.3578         0         0   -0.3534
   -0.2500   -0.4330         0         0    0.2500    0.4330         0
   -0.4330   -0.7500         0         0    0.4330    0.7500         0
   -0.2500    0.4330   -0.3534   -0.3534         0         0    0.6034
    0.4330   -0.7500   -0.3534   -0.3534         0         0   -0.0797
         0         0   -0.0000   -0.0000         0         0         0
         0         0   -0.0000   -1.0000         0         0         0

  Columns 8 through 10

    0.4330         0         0
   -0.7500         0         0
   -0.3534   -0.0000   -0.0000
   -0.3534   -0.0000   -1.0000
         0         0         0
         0         0         0
   -0.0797         0         0
    1.1034         0         0
         0    0.0000    0.0000
         0    0.0000    1.0000

Boundary Condition
```

```
f =

           0
           0
        1000
           0

deflection matrix
U =

   1.0e-05 *

    0.1042
    0.0028
    0.1877
   -0.0521
         0
         0
         0
         0
         0
         0

Reaction force
R =

   1.0e+03 *

   -0.0000
   -0.0000
    1.0000 (This is not correct. This must be zero)
    0.0000
   -0.2728
   -0.4726
   -0.7272
   -0.0489
    0.0000
    0.5214

Stress in element 1
sigma1 =

   2.7285e+04

Stress in element 2
sigma2 =

  -2.4827e+04

Stress in element 3
sigma3 =

   2.6143e+04
```

```
Stress in element 4
sigma4 =

   3.3862e+04

Stress in element 5
sigma5 =

  -2.6072e+04
```

Deformed vs undeformed plot:

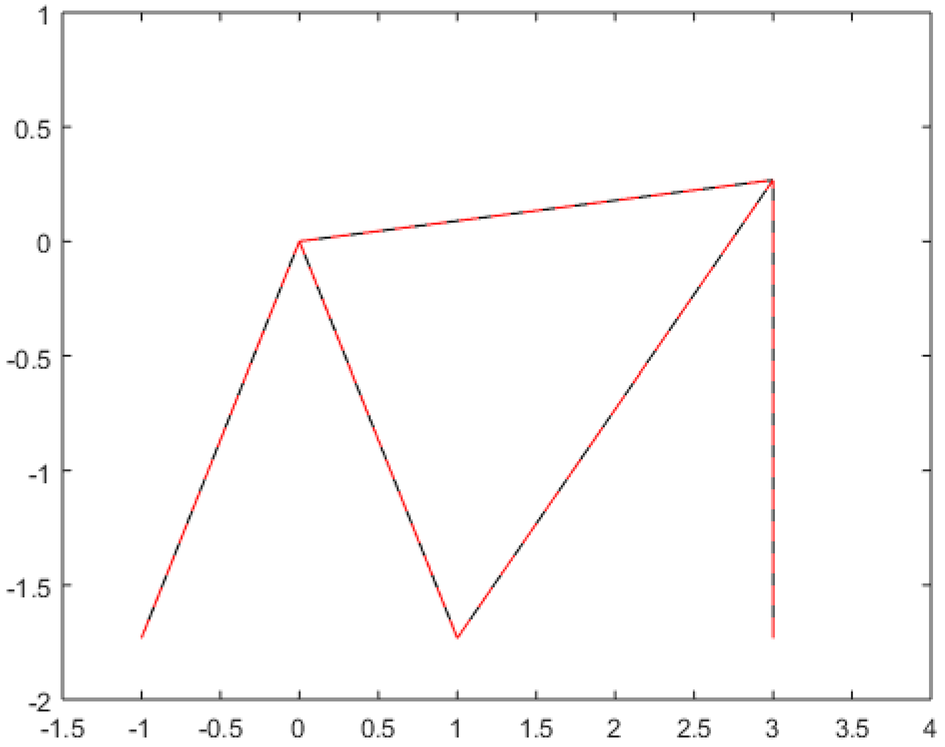


Figure 4: Main plot Published with MATLAB® R2015a

In all subsequent figures,

RED dotted line=deformed shape

Black straight line= undeformed shape

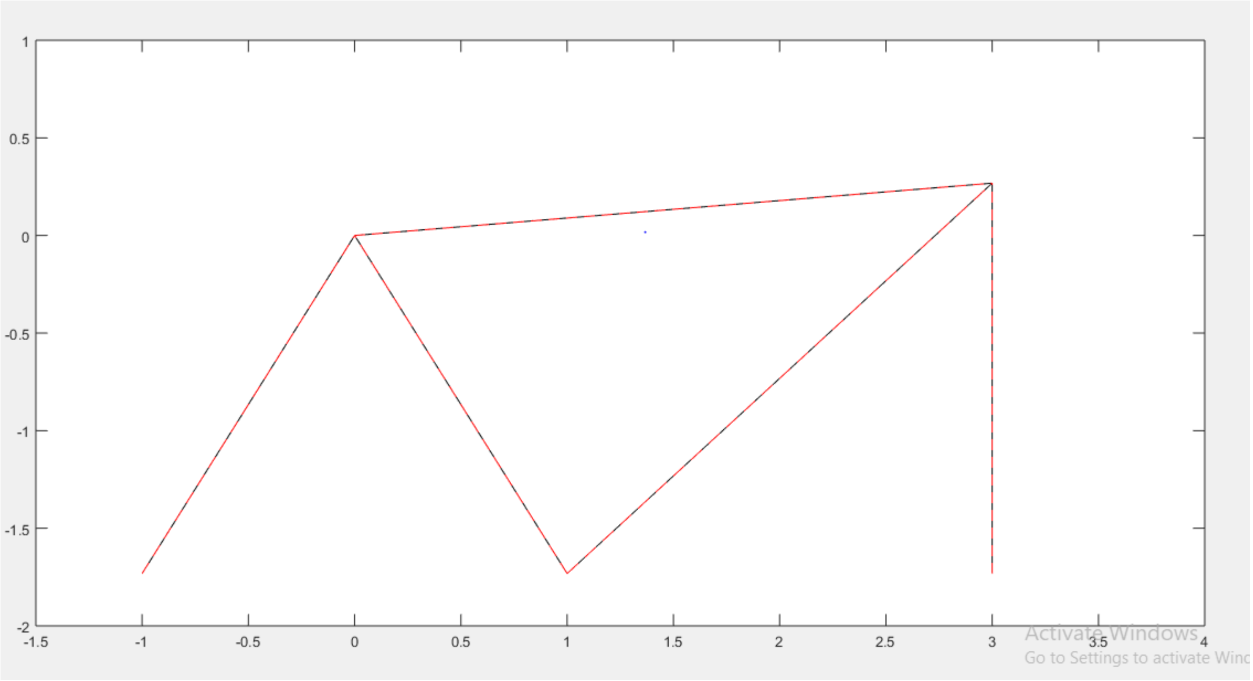


Figure 5: Deformed vs undeformed shape(1:1 ratio)

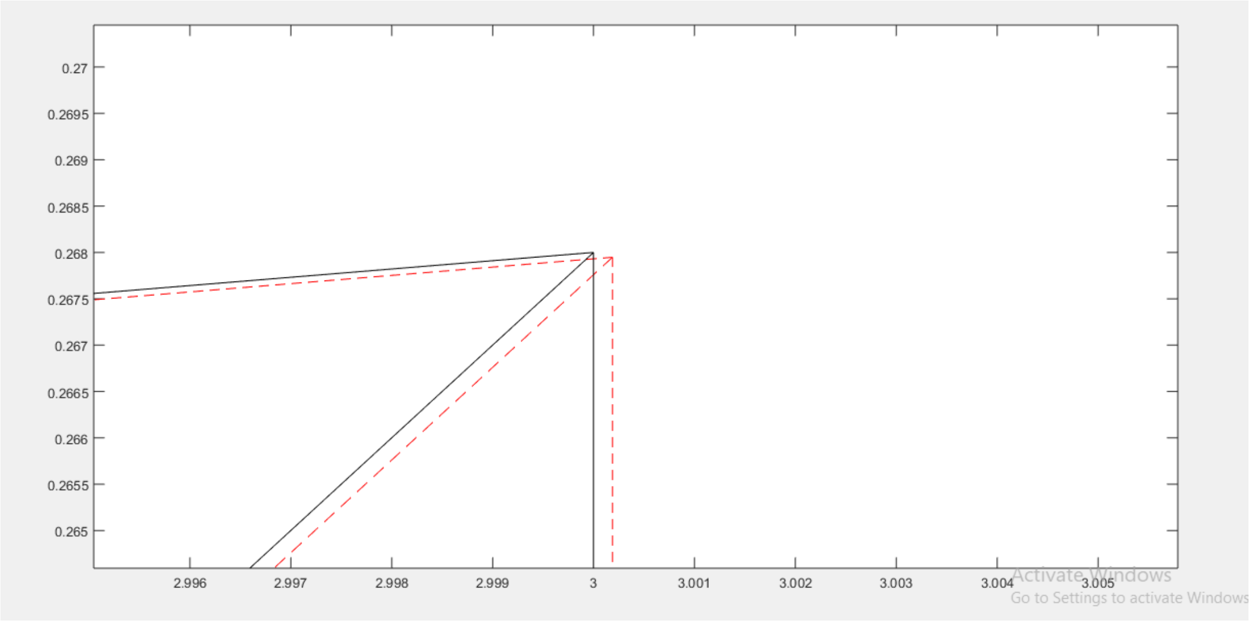


Figure 6: Node 2 deformed vs undeformed shape(external only horizontal load point)

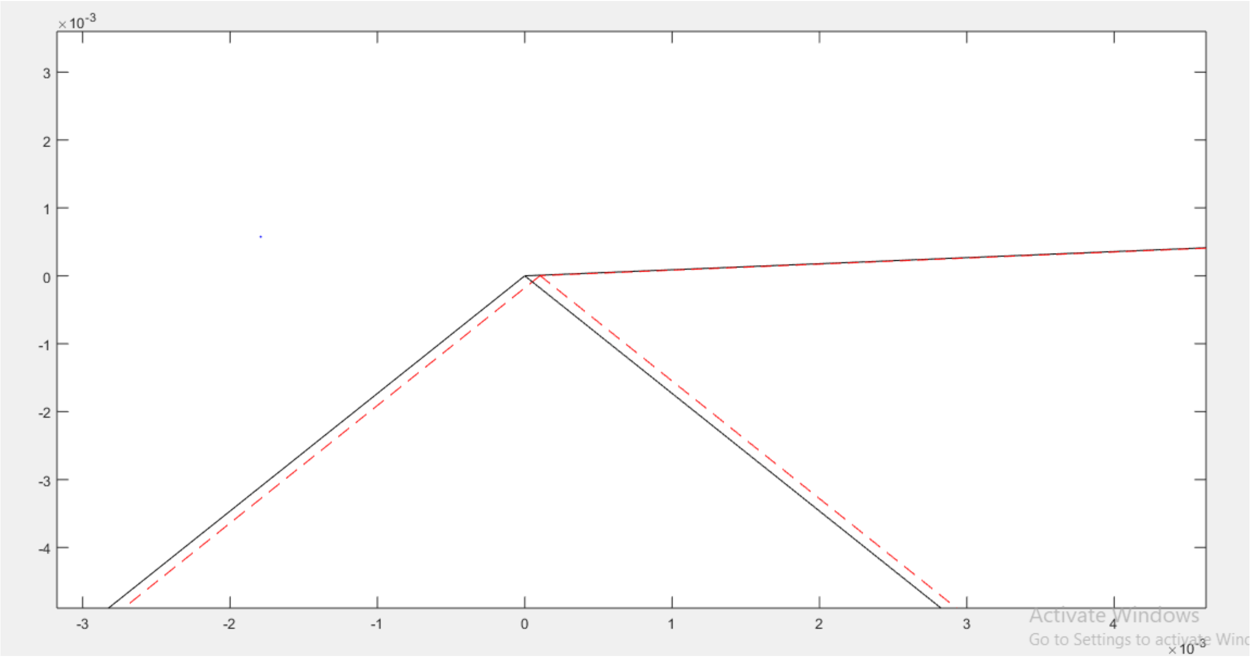


Figure 7: Node 1 deformed vs undeformed shape

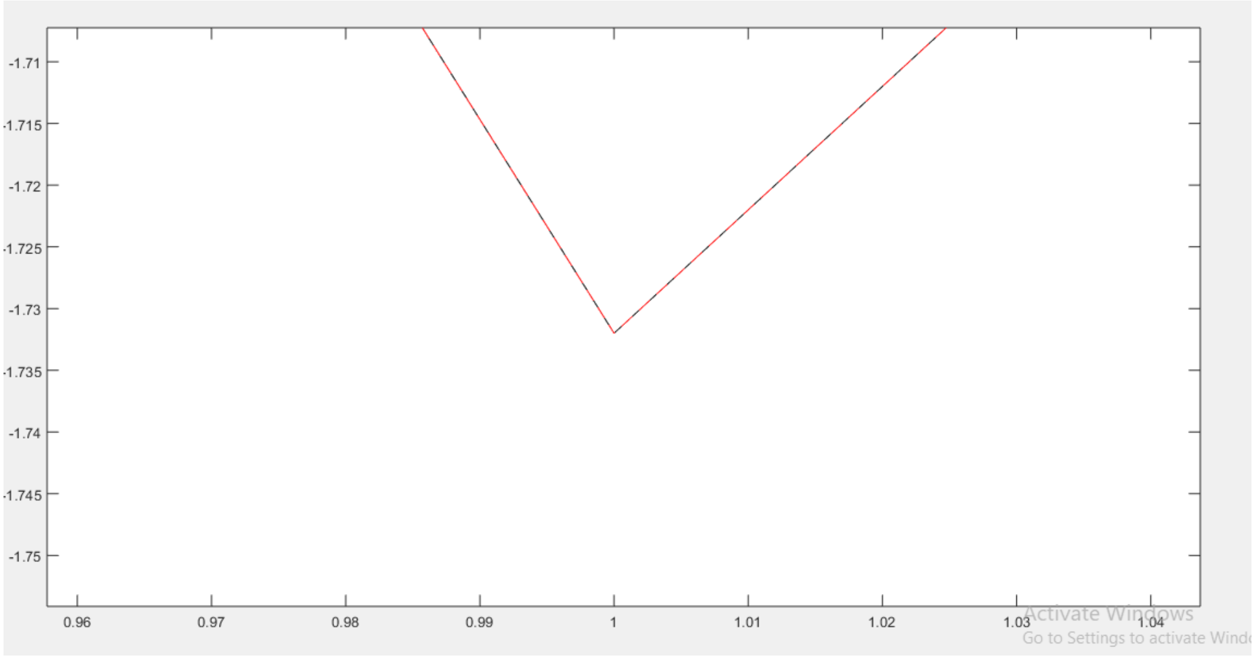


Figure 8: Node 4 deformed vs undeformed shape(hinged point)

Equilibrium Checking of the Entire System

- There is a by hand calculation error for reaction force which is also found in MATLAB code and output.
- Hand calculated global stiffness matrix first 4*4=16 value is almost same as code output.
- MATLAB by default took more values after the point value. So, most of the value is slightly different from hand calculation.
- Undeformed vs Undeformed plot follows the theoretical zero value in the hinged position.