

**FEA**

Computational Assignment No. 01

**Instructor:**

Mr. Malik Hassan

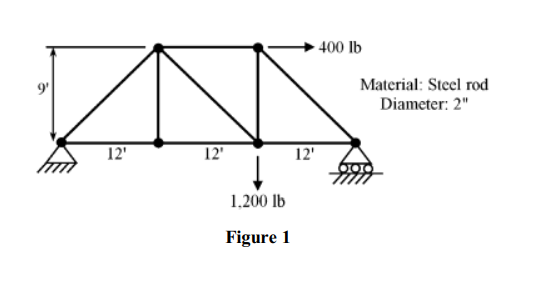
**Date:** 18 April 2022

**Submitted by:**

**Waqar Ahmad**

**2018506**

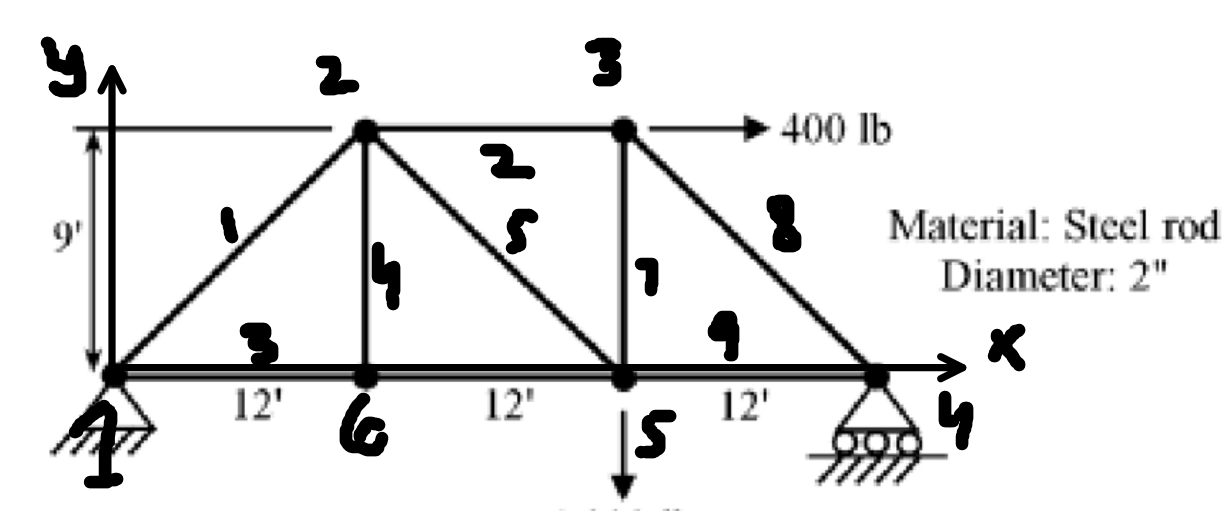
Question: You are required to evaluate the stresses (tension and  
compression) in nine bar truss depicted in Figure 1 through MATLAB  
code. Also, determine the stresses analytically and comment on the error  
if any. Moreover, show the deformed and un-deformed geometry of the  
truss and justify that truss deformation is with in elastic limit.  
While developing a code you may follow following steps:  
•Code accept appropriate data, orientation of element, local and  
global code number and boundary condition details  
•Generate element matrices and transform to global reference frame  
•Perform assembly incorporating boundary conditions  
•Solve for stresses



**Code In Matlab:**

% co-ordinates of nodes in x and y(unit inches)

x=12\*[0 12 24 36 24 12]';



y=12\*[0 9 9 0 0 0]';

% There are six nodes

Total\_node=6;

% Element and it node combination

Elements=[1 2;2 3;1 6;6 2;2 5;6 5;5 3;3 4;5 4];

E\_num=9 %Total number of element

% Area of elements

diameter=2; % diameter

A=(pi/4)\*(diameter^2)\*ones(E\_num,1);

% Young Modulus of element, steel E = 29e6 psi and make it matrice

E=29e6\*ones(E\_num,1);

**Algorithm for stiffness matrix:**

for i=1:E\_num

% Check for element nodes

j=Elements(i,1);

k=Elements(i,2);

%Pythagorean theorem

L(i)=sqrt((x(k)-x(j))^2+(y(k)-y(j))^2);

B{i}=[1/L(i),0,-1/L(i),0];

E1=29e6;

% Transfer function

l(i)=(x(k)-x(j))/L(i);

m(i)=(y(k)-y(j))/L(i);

M=[l(i),m(i);-m(i),l(i)];

Z=[0,0;0,0];

T{i}=[M,Z;Z,M];

% local stiffness matrice 

Local\_K{i}=A(i)\*T{i}'\*B{i}'\*E1\*B{i}\*T{i}\*L(i);

end

**Algorithm Global Stiffness Matrix:**

Global\_K=0;

for i=1:E\_num

%Making the matrice of 12 by 12

% and making zero for each interval

tem\_K{i}=zeros(2\*Total\_node,2\*Total\_node);

j=Elements(i,1);

k=Elements(i,2);

% Replace the coordinate of local matrice in global

tem\_K{i}(2\*j-1:2\*j,2\*j-1:2\*j)=local\_K{i}(1:2,1:2);

tem\_K{i}(2\*j-1:2\*j,2\*k-1:2\*k)=local\_K{i}(1:2,3:4);

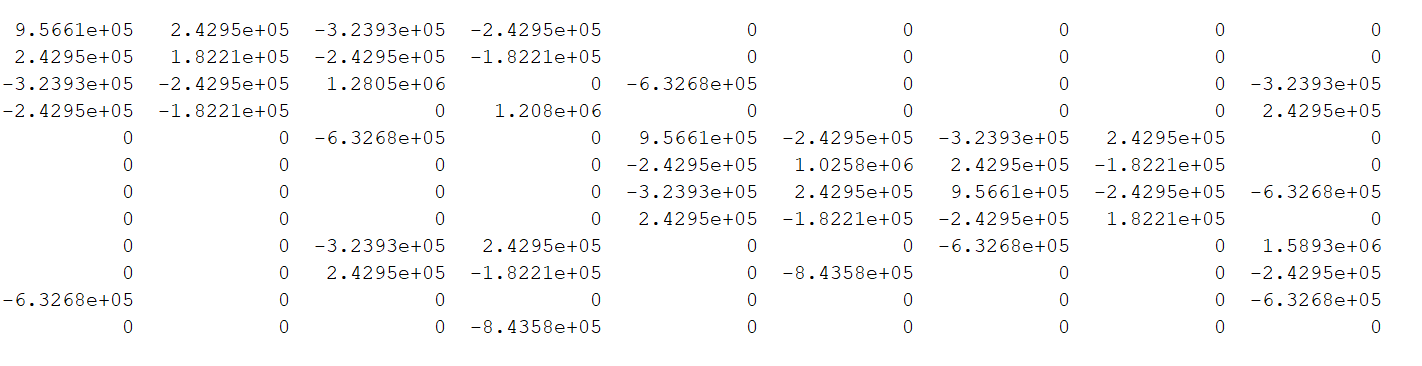
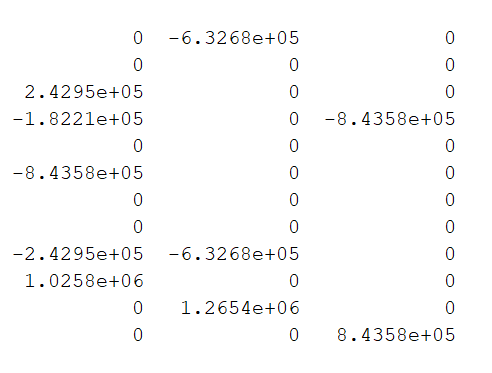
tem\_K{i}(2\*k-1:2\*k,2\*k-1:2\*k)=local\_K{i}(3:4,3:4);

tem\_K{i}(2\*k-1:2\*k,2\*j-1:2\*j)=local\_K{i}(3:4,1:2);

%add all matric

Global\_K=Global\_K+tem\_K{i};

end

**Force Matrice:**

F=[0 0 0 0 400 0 0 0 0 1500 0 0]';

%consider the reaction force zero which will be resolve latter

**Remove the 1st and 2nd Row and Column:**

node=1

u(2\*node-1:2\*node)=0

Global\_K(2\*node-1:2\*node,:)=0

Global\_K(:,2\*node-1:2\*node)=0

Global\_K(2\*node-1,2\*node-1)=1

Global\_K(2\*node,2\*node)=1

F(2\*node-1:2\*node)=0

**Remove the lastst Row and Column:**

node=12

u(2\*node)=0

Global\_K(2\*node,:)=0

Global\_K(:,2\*node)=0

Global\_K(2\*node,2\*node)=1

F(2\*node)=0

**Algorithm for Reduced row echelon form:**

Aug=[Global\_K F];

for i=1:1:2\*Total\_node

for j=i+1:2\*Total\_node

Aug(j,:)=Aug(j,:)-(Aug(j,i)/Aug(i,i))\*Aug(i,:);

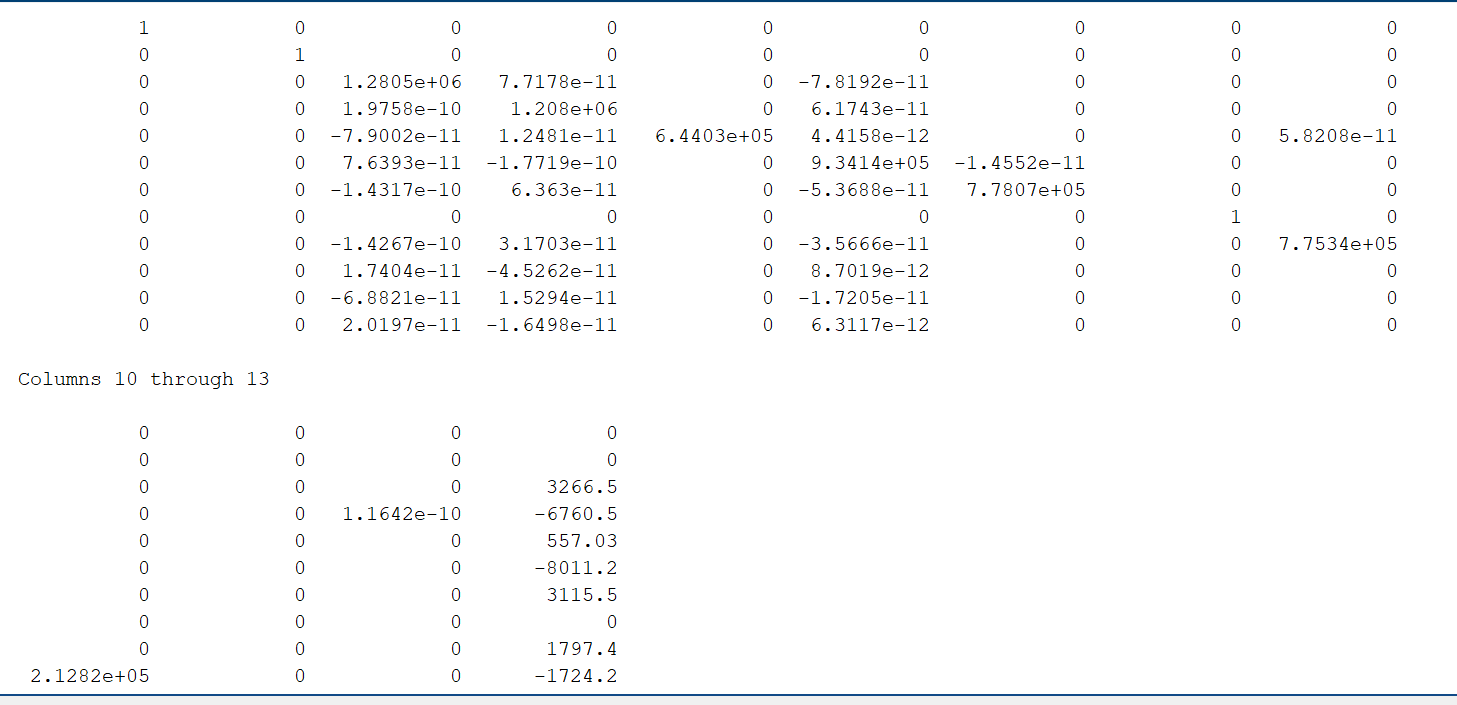
end

for j=i-1:-1:1

Aug(j,:)=Aug(j,:)-(Aug(j,i)/Aug(i,i))\*Aug(i,:);

end

end



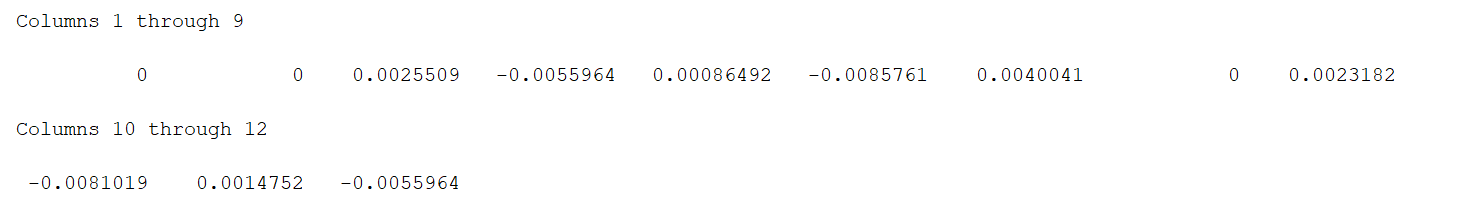
**Divide the force on stiffness which is already reduce:**

depormation=zeros(2\*Total\_node,1);

for i=1:2\*Total\_node

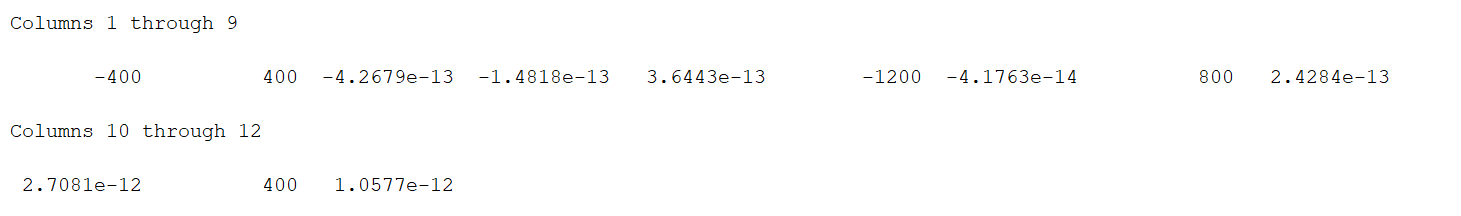
depormation(i,1)=Aug(i,2\*Total\_node+1)/Aug(i,i);

end



**Finding Support Reactions:**

RF=Global\_K\*depormation;



**Algorithm for Strain Calculations:**

strain=zeros(E\_num,1);

for i=1:E\_num

j=Elements(i,1);

k=Elements(i,2);

a=[l(i) m(i)];

b=[depormation(2\*k-1)-depormation(2\*j-1);depormation(2\*k)-depormation(2\*j)];

strain(i)=a\*b/L(i);

strain1(i,1)=i;

strain1(i,2)=strain(i);

end

**Algorithm for Stress Calculations:**

Stress=zeros(E\_num,1);

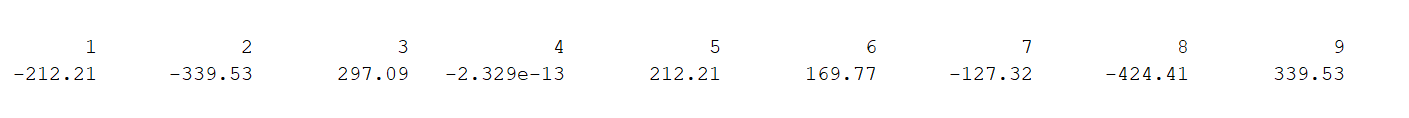
for i=1:E\_num

Stress(i)=E(i)\*strain(i);

stress1(i,1)=i;

stress1(i,2)=Stress(i);

end



**Deformed and Nn-Deformed geometry:**

for i=1:Elements

Afac = max(A);

plot([Elements(ID(i,1),2) Elements(ID(i,2),2)],[Elements(ID(i,1),3) Elements(ID(i,2),3) ],'Color',[.50 .50 .60])

hold on

if fe(i)>0

plot([Elements(ID(i,1),1) Elements(ID(i,2),1)],[Elements(ID(i,1),2) Elements(ID(i,2),2)],'b-.','Linewidth',3)

else

plot([Elements(ID(i,1),1) Elements(ID(i,2),1)],[Elements(ID(i,1),2) Elements(ID(i,2),2)],'b-.','Linewidth',3)

end

title('Deformed and Nn-Deformed geometry')

xlabel('X ')

ylabel('Y ')

hold on

grid on

end

