2018

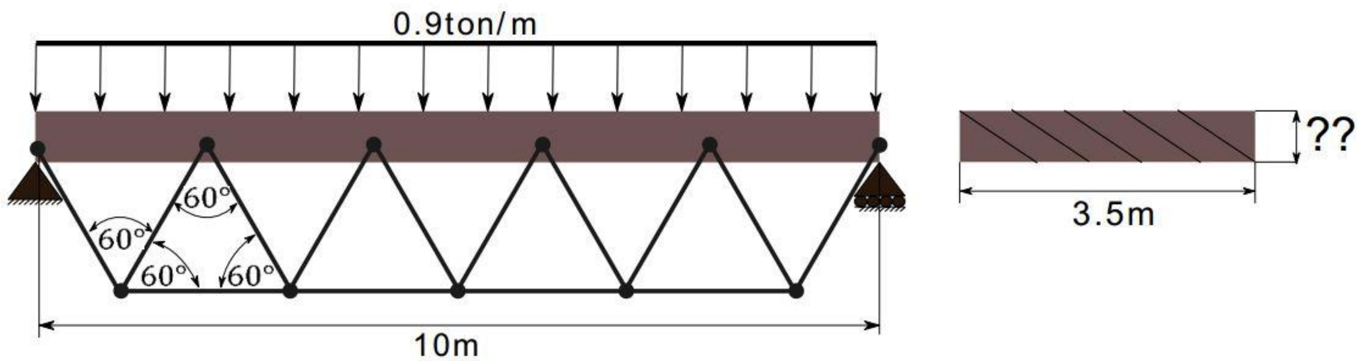
FEA Structure Analysis

SPC 402 – FEA - PROJECT REPORT

TABLE OF CONTENTS

PROBLEM STATEMENT

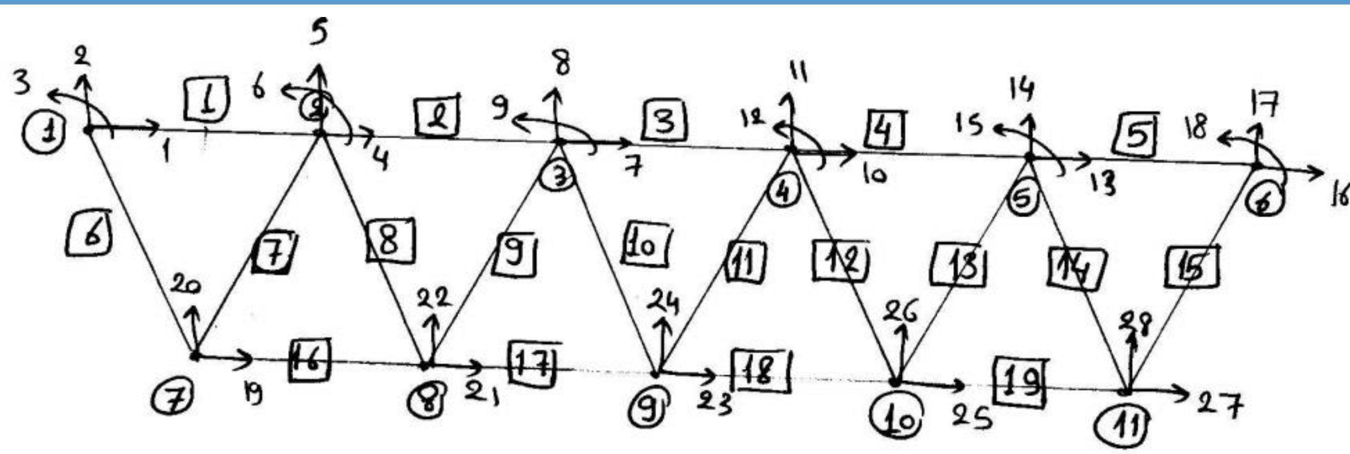
The bridge deck is modeled as a combination of bar and beam bending elements. The support is modeled as a standard truss.



It is required that we write a code to analyze the structure, and to design the height of the deck and the cross-sectional areas of the truss to make the weight of the bridge as small as possible.

	Beam	**truss**
Material	Concrete	Steel
E	$41Gpa$	$207Gpa$
Allowable stress (σ_{all})	$130Mpa$	$220Mpa$

SOLVING THE PROBLEM



We first assumed values for height of the frame and areas of the truss bars (assuming all of them are the same) and observed the stresses on all members. From this step we got an insight about how to tackle the problem, and we also noticed some important points:

- Changing the h and A values was independent from each other.

- The truss is symmetric, so we were able to assume that the members holding equal stresses have equal areas: elements 6,7,14,15,16,19 & elements 8,9,12,13 & elements 17,18. We also noticed that elements 10,11 are zero members. By knowing this we were able to assume that we can solve for only three different areas.

After that, we assumed a fixed A value for all members and optimized the h value. From this step we got the critical h value by making the ratio of one member's calculated stress to the allowable stress range from 0.999 to 1.

Then we used this h value as a constant and solved for the 3 varying areas again by making the ratio of the calculated stress to the allowable stress range from 0.999 to 1.

CODE WALKTHROUGH:

The code mainly consists of 3 functions.

The first function used is "analyze_structure". It analyzes the structure given, calculates the stresses in all the members and gets the ratio between those stresses and the allowable stress.

The second function is "optimize_h" which uses a range of h values and tries different solutions for them using the "analyze_structure" function. The function then produces the critical h, the ratio, and the stresses in all frame members in that case.

The third function "optimize_A" uses a range of A values and tries different solutions for them using "analyze_structure" function. The function then produces the critical A values, the ratio, and the stresses in all truss members in that cases.

The outputs of the run are the height of the frame, the areas of all the truss bars, and the stress in all the structure elements.

```
clear all
%optimizing h
hrange=linspace(0.0001,0.01,2000);
Ab=0.5e-4; %guess
[h,ratiof,stressf]=optimize_h(hrange,Ab);
%optimizing A
Arange=linspace(0.5e-4,4e-4,2000);
[About,ratiobout,stressbout,Abint,ratiobint,stressbint,Abvert,ratiobvert,stressbvert, stressb]=optimize_A(h,Arange);
%Arranging stress vector
stress=[];
stress([1,2,3,4,5],1)=stressf;
stress([6,7,14,15,16,19],1)=stressbout;
stress([8,9,12,13],1)=stressbint;
stress([17,18],1)=stressbvert;
stress(10,1)=stressb(10-5,1);
stress(11,1)=stressb(11-5,1);
%FINAL OUTPUTS
fprintf('Optimal value for the height of the frame h is %f\n',h);
fprintf('Optimal value for the Area of bar elements number (6,7,14,15,16,19) is %f\n',About);
fprintf('Optimal value for the Area of bar elements number (8,9,12,13) is %f\n',Abint);
fprintf('Optimal value for the Area of bar elements number (17,18) is %f\n',Abvert);
elemn={'1','2','3','4','5','6','7','8','9','10','11','12','13','14','15','16','17','18','19'};
output=table(stress,'rowNames',elemn);
disp(output)
```

ANALYZING THE GIVEN STRUCTURE

```
function [ratiof,stressf,ratiob,stressb] = analyze_structure(h,Ab)
```

1- At first we number all elements and nodes, and make a matrix that specifies each element, the nodes connected to it, and the angle each member makes with the positive x-axis.

```
%take input as [element no, node no1, node no2, theta with positive x]
enn=[1 1 2 0;
    2 2 3 0;
    3 3 4 0;
    4 4 5 0;
    5 5 6 0;
    6 7 1 120;
    7 7 2 60;
    8 8 2 120;
    9 8 3 60;
    10 9 3 120;
    11 9 4 60;
    12 10 4 120;
    13 10 5 60;
    14 11 5 120;
    15 11 6 60;
    16 7 8 0;
    17 8 9 0;
    18 9 10 0;
    19 10 11 0];
ennframe=enn(1:5,:);
ennbar=enn(6:19,:);
```

2- We number the degrees of freedom and specify which degrees of freedom are connect to which nodes.

```
%indexing nodes
nodesframe=1:6;
nodesbar=7:11;

%indexing dof
dofframe=1:1:length(nodesframe)*3;
dofframe=reshape(dofframe,3,[])';
dofbar=1+length(nodesframe)*3:1:length(nodesframe)*3+length(nodesbar)*2;
dofbar=reshape(dofbar,2,[])';

%ndoftotal=[node no, u, v, theta]
n3dofframe=[nodesframe' dofframe];
n2dofbar=[nodesbar' dofbar];
n2dofallbar=[n2dofbar zeros(length(n2dofbar),1)];
ndoftotal=[n3dofframe; n2dofallbar];
```

3- We start constructing the connectivity matrices containing the element number and its 2 nodes degrees of freedoms. We construct one for the frame and one for the truss bar separately.

```
%connectframe=[element number, node1 dof, node2 dof];
connectframe=[];
for m=1:length(ennframe)
    connectframe=[connectframe; m ndoftotal(enn(m,2),2:4) ndoftotal(enn(m,3),2:4)];
end
connectbar=[];
for m=ennbar(1,1):ennbar(length(ennbar),1)
    connectbar=[connectbar; m ndoftotal(enn(m,2),2:3) ndoftotal(enn(m,3),2:3)];
end
```

4- We calculate the values of the moment of inertia and the area of the frame from the givens.

```
%givens
L=2;
b=3.5;
Ef=41e09;
Eb=207e09;
wd=900*9.8;
%calculate
I=(1/12)*b*(h^3);
Af=b*h;
```

5- In this step, from those formulae, we calculate the stiffness matrices (local &global) for each of both the frame and truss members. We construct a 3-D array for each type (frame-truss) containing all its members stiffness matrices in its pages. Note that: the frame matrix local matrix is the same as its global one, while the truss members will need to be transformed.

Bar (Truss):

$$k = \frac{EA}{L}\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Transformation Matrices T

To transform the local stiffness matrix from the local axes to the global axes.

$$\left[\mathbf{K}^e\right] = [\mathbf{T}]^T[\mathbf{k}][\mathbf{T}]$$

Bar (Truss):

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_P & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_P \end{bmatrix} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 & 0 \\ -\sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & \cos\alpha & \sin\alpha \\ 0 & 0 & -\sin\alpha & \cos\alpha \end{bmatrix}$$

Plane Frame:

Plane frame= Beam + Bar (it doesn't neglect the axial deformation)

$$k = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}$$

```
%Building k local & global frame & truss for all elements
for i=1:5
    klf=[(Ef*Af)/L 0 0 -(Ef*Af)/L 0 0;
        0 (12*Ef*I)/L^3 (6*Ef*I)/L^2 0 -(12*Ef*I)/L^3 (6*Ef*I)/L^2;
        0 (6*Ef*I)/L^2 (4*Ef*I)/L 0 -(6*Ef*I)/L^2 (2*Ef*I)/L;
        -(Ef*Af)/L 0 0 (Ef*Af)/L 0 0;
        0 -(12*Ef*I)/L^3 -(6*Ef*I)/L^2 0 (12*Ef*I)/L^3 -(6*Ef*I)/L^2;
        0 (6*Ef*I)/L^2 (2*Ef*I)/L 0 -(6*Ef*I)/L^2 (4*Ef*I)/L];
    kgf=klf;
    kgftotal(:,:,i)=kgf;
end

for i=6:19
    klb=((Eb*Ab)/L).*[1 0 -1 0;
        0 0 0 0;
        -1 0 1 0;
        0 0 0 0];
    R=[cosd(enn(i,4)) sind(enn(i,4)); -sind(enn(i,4)) cosd(enn(i,4))];
    Tb=[R zeros(2,2); zeros(2,2) R];
    kgb=transpose(Tb)*klb*Tb;
    kgbtotal(:,:,i)=kgb;
end
```

6- Using the connectivity matrices and sparse function, we put the global matrices of each element of the frame and truss bars in one global matrix that preserves the contribution of all the elements.

```
%%%preparing for kglobal
%frame
rowf=[];
colf=[];
kgfelements=[];
for m=1:5
    for i=2:7
        for j=2:7
            rowf=[rowf; connectframe(m,i)];
            colf=[colf; connectframe(m,j)];
            kgfelements= [kgfelements; kgftotal(i-1,j-1,m)];
        end
    end
end
%bar
rowb=[];
colb=[];
kgbelements=[];
for m=6:19
    for i=2:5
        for j=2:5
            rowb=[rowb; connectbar(m-5,i)];
            colb=[colb; connectbar(m-5,j)];
            kgbelements= [kgbelements; kgbtotal(i-1,j-1,m)];
        end
    end
end
```

```
%sum & sparse
row=[rowf; rowb];
col=[colf; colb];
kgelements=[kgfelements; kgbelements];
kgtotal=sparse(row,col,kgelements);
kg=full(kgtotal);
kg28=kg;
```

7- We then prepared the local and global load matrices for each element of the structure, and set the connectivity matrix of the members. Then again using the connectivity matrix and sparse functions, we were able to reach the total load vector.

$$\{L\} = -\begin{Bmatrix} \dfrac{w_o L}{2} \\ \dfrac{w_o L^2}{12} \\ \dfrac{w_o L}{2} \\ -\dfrac{w_o L^2}{12} \end{Bmatrix}$$

```
%%LOAD
%local global load
for i=1:5
    Lgf=[-wd*L/2; -(wd*L^2)/12; -wd*L/2; (wd*L^2)/12];
    Lftotal(:,:,i)=Lgf;
end

connectLf=[connectframe(:,1), connectframe(:,3), connectframe(:,4), connectframe(:,6), connectframe(:,7)];
```

```
%load sparse
%frame
rowLf=[];
Lfelements=[];
for m=1:5
    for i=2:5
        rowLf=[rowLf; connectLf(m,i)];
        Lfelements= [Lfelements; Lftotal(i-1,1,m)];
    end
end
%set bars to zero
rowLb=[];
Lbelements=[];
for m=6:19
    for i=2:5
        rowLb=[rowLb; connectbar(m-5,i)];
        Lbelements= [Lbelements; 0];
    end
end
```

```
%sum & sparse
rowL=[rowLf; rowLb];
Lgelements=[Lfelements; Lbelements];
Lgtotal=sparse(rowL,ones(length(rowL),1),Lgelements);
Lg=full(Lgtotal);
Lg28=Lg;
```

8- From the structure we know that certain deltas are restrained. By setting these values to zeros we can omit the corresponding rows and columns in the global k matrix, and the corresponding rows in the load matrix. We can then solve for deltas (degrees of freedom).

$$[k_g] + [\Delta] = \{R\} + \{L_g\}$$

$$[\Delta] = [k_g]^{-1}\{L_g\}$$

```
%%%Deleting rows of delta corresponding supports
kg(1,:)=[]; kg(:,1)=[];
kg((2-1),:)=[]; kg(:,(2-1))=[];
kg((17-2),:)=[]; kg(:,(17-2))=[];

Lg(1,:)=[];
Lg((2-1),:)=[];
Lg((17-2),:)=[];

%calc deltas
delta25=kg\Lg;
delta28=[0;0;delta25(1:17-2-1,:);0;delta25(17-2:25)];
```

9- Then we calculate the internal forces from each member, and transform it to the global axes. Using the forces, we can calculate the stresses in all the members. We then calculate the ratio between the stresses calculated and the allowable stress.

$$\{f\}_i = [k]_i\{\Delta\}_i$$

$$\{f_{local\ axes}\} = [T]\,\{f_{global\ axes}\}$$

For truss bars: $\sigma = \frac{f_{axial}}{A}$

For frames: $\sigma = \frac{f_{axial}}{A} + \left(\frac{My}{I}\right)_{maximum}$, where $y = \frac{h}{2}$ and M is the maximum moment.

```
%%%internal forces
for i=1:5
    ff(:,:,i)=kgftotal(:,:,i)*[delta28(connectframe(i,2:7),1)];
    M(i,1)=max(abs(ff(3,:,i)),abs(ff(6,:,i)));
    stressf(i,1)=(abs(ff(1,:,i))/Af)+(M(i,1)*(h/2)/I);
end
for i=6:19
    fb(:,:,i)=kgbtotal(:,:,i)*[delta28(connectbar(i-5,2:5),1)];
    R=[cosd(enn(i,4)) sind(enn(i,4)); -sind(enn(i,4)) cosd(enn(i,4))];
    Tb=[R zeros(2,2); zeros(2,2) R];
    flb(:,:,i)=Tb*fb(:,:,i);
    stressb(i-5,1)=abs(flb(3,:,i))/Ab;
end
stressfallow=130e06;
stressballow=220e06;
ratiof=stressf./stressfallow;
ratiob=stressb./stressballow;
```

OPTIMIZING THE HEIGHT OF THE FRAME

```
function [h,ratiof,stressf] = optimize_h(hrange,Ab)
```

Using a fixed (assumed) value for the area of the truss, we can solve for the optimal height of the frame. We solve by making the ratio of one member's calculated stress to the allowable stress range from 0.999 to 1

```
for z=1:length(hrange)
h=hrange(z);
[ratiof, stressf, ratiob, stressb]=analyze_structure( h,Ab );
if (max(ratiof) >= 0.999) && (max(ratiof) <= 1)
     break
end

end

end
```

OPTIMIZING THE AREAS OF THE TRUSS BARS

```
function [About,ratiobout,stressbout,Abint,ratiobint,stressbint,Abvert,ratiobvert,stressbvert, stressb] = optimize_A(h,Arange)
```

Using the optimal h value we got, we can solve for the 3 varying areas by making the ratio of the calculated stress to the allowable stress range from 0.999 to 1.

```
for c=1:length(Arange)
Ab=Arange(c);
[ratiof, stressf, ratiob, stressb]=analyze_structure( h,Ab );
%elements 6 7 14 15 16 19
if (ratiob(6-5,1) >= 0.999) && (ratiob(6-5,1) <= 1)
    About=Ab;
    ratiobout=ratiob;
    stressbout=stressb(6-5,1);
end
%elements 8 9 12 13
if (ratiob(8-5,1) >= 0.99) && (ratiob(8-5,1) <= 1)
    Abint=Ab;
    ratiobint=ratiob;
    stressbint=stressb(8-5,1);
end
%elements 17 18
if (ratiob(17-5,1) >= 0.999) && (ratiob(17-5,1) <= 1)
    Abvert=Ab;
    ratiobvert=ratiob;
    stressbvert=stressb(17-5,1);
end

end
end
```

FINAL OUTPUT

```
Optimal value for the height of the frame h is 0.006251
Optimal value for the Area of bar elements number (6,7,14,15,16,19) is 0.000195
Optimal value for the Area of bar elements number (8,9,12,13) is 0.000091
Optimal value for the Area of bar elements number (17,18) is 0.000285
```

	stress
1	1.2996e+08
2	3.558e+07
3	1.0383e+07
4	3.558e+07
5	1.2996e+08
6	2.1992e+08
7	2.1992e+08
8	2.1802e+08
9	2.1802e+08
10	7.276e-08
11	3.638e-08
12	2.1802e+08
13	2.1802e+08
14	2.1992e+08
15	2.1992e+08
16	2.1992e+08
17	2.198e+08
18	2.198e+08
19	2.1992e+08