CE 425 – FALL 2014
HOMEWORK-4 SOLUTIONS

Q1)

```
clear all
clc

%INPUT PHASE
%CE425 Fall 2014 HW4-Q1
% (Prepared by Alper Aldemir for Fall 2013)

XY=[0 0; 0 3; 4 3; 4 0];
%This matrix containes the x and y coordinates of the nodes,
respectively.
%The total number of rows is the total number of nodes. (NumNode)
XY=[X Y] for each node
M=[0.02 0.08 200000*10^6; 0.01 0.01 200000*10^6];
%This is a material matrix. Each row is composed of the area,
%the moment of inertia and the modulus of elasticity of different
members.
%The total number of rows is dependent on the total number of
different material data sets. M=[A I E] for each material set
C=[1 2 1 1; 2 3 1 1; 3 4 1 1; 1 3 2 1];
%This is the connectivity matrix. It connects the elements to the
nodes, the materials and the element types.
%The total number of rows is the total number of elements.
(NumElem)This matrix is composed of starting node,
%ending node, the material set number and the element type (0 for
truss and 1 for frame) in each columns. C=[SN EN MS ET] for each
elements
S=[1 1 1 0; 4 0 1 0];
%This matrix shows the restrains. The total number of rows is the
total number of supports. (NumSupport)
%The columns refer node number, the x restrained, the y
restrained and the restrained about z axis.
%The ones show the structure is restrained and zeros show that
the ends are free. S=[Node X Y Z] for each support
L=[2 10000 -10000 0; 3 10000 -10000 0];
%This matrix is used to show the nodal loads. The total numbers
of rows are
%equal to the total number of nodal loads. (NumLoadJoint)The
columns are node number, the global x load, the global y load
%and the global z moment. L=[Node XL YL ZM] for each loading node
%
%MAIN PROGRAM
%
```

```
NumNode=size(XY,1);
NumElem=size(C,1);
NumSupport=size(S,1);
NumLoadJoint=size(L,1);

[E NumEq]=dof(NumNode,NumSupport,S);                         %This
calculates the active degrees of freedoms.
[Angle Le]=framelength(NumElem,C,XY);
%This calculates the length and orientation of the members.

for i=1:NumElem
    if C(i,4)==0
        klocal{i,1}=trussstiffness(i,C,M,Le);
    elseif C(i,4)==1
        klocal{i,1}=framestiffness(i,C,M,Le);
    end
end

for i=1:NumElem
    if C(i,4)==0
        [k{i,1}, R{i,1}]=trussglobalstiff(i,Angle,klocal);
    elseif C(i,4)==1
        [k{i,1}, R{i,1}]=frameglobalstiff(i,Angle,klocal);
    end
end

[Contribution]=dofcontribution(NumElem,NumNode,E,C);         %This
determines the degreee of freedom contributions of elements.

K=zeros(NumEq,NumEq);
%The global stiffness matrix is a square matrix having a size
equal to the number of degrees of freedom. (NumEq)

for i=1:NumElem
    if C(i,4)==0
        K=trussmapper(i,Contribution,k,K);
    elseif C(i,4)==1
        K=framemapper(i,Contribution,k,K);
    end
end

P=globalload(NumEq,NumLoadJoint,NumNode,L,E)

D=K\P                       %The structural displacements in
global coordinates.

flocal=localframememberforces(NumElem,C,Contribution,D,klocal,R)
```

Functions Used:

```
function [E NumEq]=dof(NumNode,NumSupport,S)
%This function calculates the active degree of freedom and
numbers them.

E=zeros(NumNode,3);

for i=1:NumNode                                         %
    for j=1:NumSupport                                  %
        if S(j,1)==i                                    %This part
deals with the inactive degrees of freedom.
            for k=1:3                                   %The inactive
degrees of freedom is tranferred with the help of the S matrix.
                E(i,k)=S(j,k+1);                        %Note that
instead of the zeros function, an else statement can be used.
            end                                         %
        end                                             %
    end                                                 %
end                                                     %

k=1;                                            %
for i=1:NumNode                                 %
    for j=1:3                                   %
        if E(i,j)~=0                            %
            E(i,j)=0;                           %This part is used
for numbering the active degrees of freedom.
        else                                    %
            E(i,j)=k;                           %
            k=k+1;                              %
        end                                     %
    end                                         %
end                                             %

NumEq=max(max(E));
```

```
function [Contribution]=dofcontribution(NumElem,NumNode,E,C)
%This function calculates the structural degree of freedom
contributions of the members.

Start_Contribution=[];
End_Contribution=[];
for n=1:NumElem
    for k=1:NumNode
        if C(n,1)==k
            Start_Contribution=E(k,:);
        end
        if C(n,2)==k
            End_Contribution=E(k,:);
        end
    end
    if C(n,4)==0
        Contribution{n,1}=[Start_Contribution(1,1:2)
End_Contribution(1,1:2)];
    elseif C(n,4)==1
        Contribution{n,1}=[Start_Contribution End_Contribution];
    end
end
```

```
function [k, R]=frameglobalstiff(i,Angle,klocal)
%This function calculates the global stiffness matrices for
members.

R=[cos(Angle(i,1))  sin(Angle(i,1)) 0 0 0 0;
  -sin(Angle(i,1)) cos(Angle(i,1)) 0 0 0 0;
   0 0 1 0 0 0;
   0 0 0 cos(Angle(i,1)) sin(Angle(i,1)) 0;
   0 0 0 -sin(Angle(i,1)) cos(Angle(i,1)) 0;
   0 0 0 0 0 1];

k=R'*klocal{i,1}*R;
```

```
function [Angle Le]=framelength(NumElem,C,XY)
%This function calculates the length and orientation of members.

for i=1:1:NumElem
    Coord(i,1)=XY(C(i,1),1);
    Coord(i,2)=XY(C(i,1),2);
    Coord(i,3)=XY(C(i,2),1);
    Coord(i,4)=XY(C(i,2),2);
end

for i=1:1:NumElem
    Le(i,1)=sqrt((Coord(i,3)-Coord(i,1))^2+(Coord(i,4)-
Coord(i,2))^2);
    Angle(i,1)=atan((Coord(i,4)-Coord(i,2))/(Coord(i,3)-
Coord(i,1)));
end



function K=framemapper(i,Contribution,k,K)

for j=1:6
    for m=1:6
        if Contribution{i,1}(1,j)~=0 && Contribution{i,1}(1,m)~=0

K(Contribution{i,1}(1,j),Contribution{i,1}(1,m))=K(Contribution{i
,1}(1,j),Contribution{i,1}(1,m))+k{i,1}(j,m);
        else
            K=K;
        end
    end
end
```

```
function klocal=framestiffness(i,C,M,Le)
%This function calculates the local stiffness matrix of an
element.

A=M(C(i,3),1);
E=M(C(i,3),3);
I=M(C(i,3),2);
L=Le(i,1);

c1=E*A/L;
c2=12*E*I/L^3;
c3=6*E*I/L^2;
c4=4*E*I/L;

klocal=[c1,0,0,-c1,0,0
        0,c2,c3,0,-c2,c3
        0,c3,c4,0,-c3,c4/2
        -c1,0,0,c1,0,0
        0,-c2,-c3,0,c2,-c3
        0,c3,c4/2,0,-c3,c4];
```

```
function P=globalload(NumEq,NumLoadJoint,NumNode,L,E)
%This function forms the global force vector.

P=zeros(NumEq,1);            %The loading vector should have NumEq
rows.
for m=1:NumLoadJoint                                    %
    for n=1:NumNode                                     %
        if L(m,1)==n                                    %
            for k=1:3                                   %
                if E(n,k)~=0                            %
                    P(E(n,k),1)=L(m,k+1);               %
                end                                     %The loading
vector is produced.
            end                                         %
        else                                            %
            P=P;                                        %
        end                                             %
    end                                                 %
end                                                     %
```

```
function
flocal=localframememberforces(NumElem,C,Contribution,D,klocal,R)
%This function calculates the frame member forces in local
coordinates.

for i=1:NumElem
    if C(i,4)==0
        for j=1:4
            if Contribution{i,1}(1,j)==0
                d{i,1}(1,j)=0;
            else
                d{i,1}(1,j)=D(Contribution{i,1}(1,j),1);
            end
        end
    elseif C(i,4)==1
        for j=1:6
            if Contribution{i,1}(1,j)==0
                d{i,1}(1,j)=0;
            else
                d{i,1}(1,j)=D(Contribution{i,1}(1,j),1);
            end
        end
    end
end

for i=1:NumElem
    dlocal{i,1}=R{i,1}*d{i,1}';
end

for i=1:NumElem
    flocal{i,1}=klocal{i,1}*dlocal{i,1};
end
```

```
function [k, R]=trussglobalstiff(i,Angle,klocal)
%This function calculates the global stiffness matrices for
members.

R=[cos(Angle(i,1))  sin(Angle(i,1)) 0 0;
   -sin(Angle(i,1)) cos(Angle(i,1)) 0 0;
   0 0 cos(Angle(i,1)) sin(Angle(i,1));
   0 0 -sin(Angle(i,1)) cos(Angle(i,1))];

k=R'*klocal{i,1}*R;
```

```
function K=trussmapper(i,Contribution,k,K)

for j=1:4
    for m=1:4
        if Contribution{i,1}(1,j)~=0 && Contribution{i,1}(1,m)~=0

K(Contribution{i,1}(1,j),Contribution{i,1}(1,m))=K(Contribution{i
,1}(1,j),Contribution{i,1}(1,m))+k{i,1}(j,m);
        else
            K=K;
        end
    end
end
```

```
function klocal=trussstiffness(i,C,M,Le)
%This function calculates the local stiffness matrix of an
element.

A=M(C(i,3),1);
E=M(C(i,3),3);
L=Le(i,1);

c1=E*A/L;

klocal=[c1,0,-c1,0;
        0,0,0,0;
       -c1,0,c1,0;
        0,0,0,0;];
```

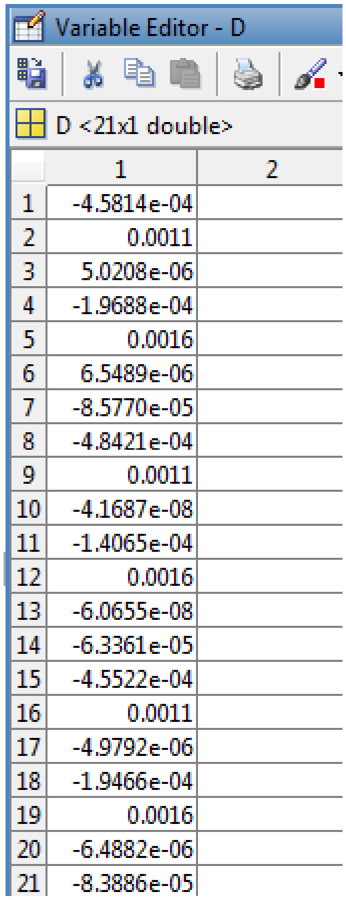
Q2)

a)

```
clear all
clc

%INPUT PHASE
%CE425 Fall 2014 HW4-Q2-a
% (Prepared by Alper Aldemir for Fall 2013)

XY=[0 0; 0 3; 0 6; 6 0; 6 3; 6 6; 12 0; 12 3; 12 6];
%This matrix containes the x and y coordinates of the nodes,
respectively.
%The total number of rows is the total number of nodes. (NumNode)
XY=[X Y] for each node
M=[0.25 0.0052 30000*10^6; 0.25 0.0052 30000*10^6];
%This is a material matrix. Each row is composed of the area,
%the moment of inertia and the modulus of elasticity of different
members.
%The total number of rows is dependent on the total number of
different material data sets. M=[A I E] for each material set
C=[1 2 1 1; 2 3 1 1; 2 5 1 1; 3 6 1 1; 4 5 1 1; 5 6 1 1; 5 8 1 1;
6 9 1 1; 7 8 1 1; 8 9 1 1];
%This is the connectivity matrix. It connects the elements to the
nodes, the materials and the element types.
%The total number of rows is the total number of elements.
(NumElem)This matrix is composed of starting node,
%ending node, the material set number and the element type (0 for
truss and 1 for frame) in each columns. C=[SN EN MS ET] for each
elements
S=[1 1 1 0; 4 1 1 0; 7 1 1 0];
%This matrix shows the restrains. The total number of rows is the
total number of supports. (NumSupport)
%The columns refer node number, the x restrained, the y
restrained and the restrained about z axis.
%The ones show the structure is restrained and zeros show that
the ends are free. S=[Node X Y Z] for each support
L=[2 10000 0 0; 3 20000 0 0];
%This matrix is used to show the nodal loads. The total numbers
of rows are
%equal to the total number of nodal loads. (NumLoadJoint)The
columns are node number, the global x load, the global y load
%and the global z moment. L=[Node XL YL ZM] for each loading node
%
%MAIN PROGRAM
%
NumNode=size(XY,1);
```

```
NumElem=size(C,1);
NumSupport=size(S,1);
NumLoadJoint=size(L,1);

[E NumEq]=dof(NumNode,NumSupport,S);                              %This
calculates the active degrees of freedoms.
[Angle Le]=framelength(NumElem,C,XY);
%This calculates the length and orientation of the members.

for i=1:NumElem
    if C(i,4)==0
        klocal{i,1}=trussstiffness(i,C,M,Le);
    elseif C(i,4)==1
        klocal{i,1}=framestiffness(i,C,M,Le);
    end
end

for i=1:NumElem
    if C(i,4)==0
        [k{i,1}, R{i,1}]=trussglobalstiff(i,Angle,klocal);
    elseif C(i,4)==1
        [k{i,1}, R{i,1}]=frameglobalstiff(i,Angle,klocal);
    end
end

[Contribution]=dofcontribution(NumElem,NumNode,E,C);          %This
determines the degreee of freedom contributions of elements.

K=zeros(NumEq,NumEq);
%The global stiffness matrix is a square matrix having a size
equal to the number of degrees of freedom. (NumEq)

for i=1:NumElem
    if C(i,4)==0
        K=trussmapper(i,Contribution,k,K);
    elseif C(i,4)==1
        K=framemapper(i,Contribution,k,K);
    end
end

P=globalload(NumEq,NumLoadJoint,NumNode,L,E)

D=K\P                           %The structural displacements in
global coordinates.

flocal=localframememberforces(NumElem,C,Contribution,D,klocal,R)
```

Displacements (in m) ;

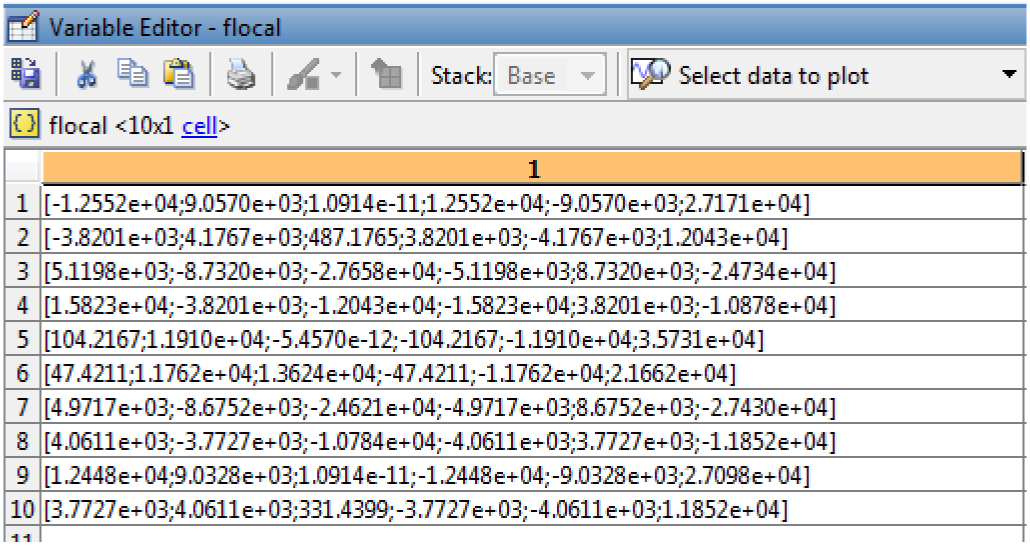


Variable Editor - D

D <21x1 double>

	1	2
1	-4.5814e-04	
2	0.0011	
3	5.0208e-06	
4	-1.9688e-04	
5	0.0016	
6	6.5489e-06	
7	-8.5770e-05	
8	-4.8421e-04	
9	0.0011	
10	-4.1687e-08	
11	-1.4065e-04	
12	0.0016	
13	-6.0655e-08	
14	-6.3361e-05	
15	-4.5522e-04	
16	0.0011	
17	-4.9792e-06	
18	-1.9466e-04	
19	0.0016	
20	-6.4882e-06	
21	-8.3886e-05	

Forces (in N and m);



	1
1	[-1.2552e+04;9.0570e+03;1.0914e-11;1.2552e+04;-9.0570e+03;2.7171e+04]
2	[-3.8201e+03;4.1767e+03;487.1765;3.8201e+03;-4.1767e+03;1.2043e+04]
3	[5.1198e+03;-8.7320e+03;-2.7658e+04;-5.1198e+03;8.7320e+03;-2.4734e+04]
4	[1.5823e+04;-3.8201e+03;-1.2043e+04;-1.5823e+04;3.8201e+03;-1.0878e+04]
5	[104.2167;1.1910e+04;-5.4570e-12;-104.2167;-1.1910e+04;3.5731e+04]
6	[47.4211;1.1762e+04;1.3624e+04;-47.4211;-1.1762e+04;2.1662e+04]
7	[4.9717e+03;-8.6752e+03;-2.4621e+04;-4.9717e+03;8.6752e+03;-2.7430e+04]
8	[4.0611e+03;-3.7727e+03;-1.0784e+04;-4.0611e+03;3.7727e+03;-1.1852e+04]
9	[1.2448e+04;9.0328e+03;1.0914e-11;-1.2448e+04;-9.0328e+03;2.7098e+04]
10	[3.7727e+03;4.0611e+03;331.4399;-3.7727e+03;-4.0611e+03;1.1852e+04]

Q2)

b)

```
clear all
clc

%INPUT PHASE
%CE425 Fall 2014 HW4-Q2-b
% (Prepared by Alper Aldemir for Fall 2013)

XY=[0 0; 0 3; 0 6; 6 0; 6 3; 6 6];
%This matrix containes the x and y coordinates of the nodes,
respectively.
%The total number of rows is the total number of nodes. (NumNode)
XY=[X Y] for each node
M=[0.25 0.0052 30000*10^6; 0.000007065 0.00000000000397
200000*10^6];
%This is a material matrix. Each row is composed of the area,
%the moment of inertia and the modulus of elasticity of different
members.
%The total number of rows is dependent on the total number of
different material data sets. M=[A I E] for each material set
C=[1 2 1 1; 2 3 1 1; 2 5 1 1; 3 6 1 1; 4 5 1 1; 5 6 1 1; 1 5 2 0;
2 4 2 0; 2 6 2 0; 3 5 2 0];
%This is the connectivity matrix. It connects the elements to the
nodes, the materials and the element types.
%The total number of rows is the total number of elements.
(NumElem)This matrix is composed of starting node,
%ending node, the material set number and the element type (0 for
truss and 1 for frame) in each columns. C=[SN EN MS ET] for each
elements
```

```
S=[1 1 1 0; 4 1 1 0];
%This matrix shows the restrains. The total number of rows is the
total number of supports. (NumSupport)
%The columns refer node number, the x restrained, the y
restrained and the restrained about z axis.
%The ones show the structure is restrained and zeros show that
the ends are free. S=[Node X Y Z] for each support
L=[2 10000 0 0; 3 20000 0 0];
%This matrix is used to show the nodal loads. The total numbers
of rows are
%equal to the total number of nodal loads. (NumLoadJoint)The
columns are node number, the global x load, the global y load
%and the global z moment. L=[Node XL YL ZM] for each loading node
%
%MAIN PROGRAM
%
NumNode=size(XY,1);
NumElem=size(C,1);
NumSupport=size(S,1);
NumLoadJoint=size(L,1);

[E NumEq]=dof(NumNode,NumSupport,S);                         %This
calculates the active degrees of freedoms.
[Angle Le]=framelength(NumElem,C,XY);
%This calculates the length and orientation of the members.

for i=1:NumElem
    if C(i,4)==0
        klocal{i,1}=trussstiffness(i,C,M,Le);
    elseif C(i,4)==1
        klocal{i,1}=framestiffness(i,C,M,Le);
    end
end

for i=1:NumElem
    if C(i,4)==0
        [k{i,1}, R{i,1}]=trussglobalstiff(i,Angle,klocal);
    elseif C(i,4)==1
        [k{i,1}, R{i,1}]=frameglobalstiff(i,Angle,klocal);
    end
end

[Contribution]=dofcontribution(NumElem,NumNode,E,C);         %This
determines the degreee of freedom contributions of elements.
```

```
K=zeros(NumEq,NumEq);
%The global stiffness matrix is a square matrix having a size
equal to the number of degrees of freedom. (NumEq)

for i=1:NumElem
    if C(i,4)==0
        K=trussmapper(i,Contribution,k,K);
    elseif C(i,4)==1
        K=framemapper(i,Contribution,k,K);
    end
end

P=globalload(NumEq,NumLoadJoint,NumNode,L,E)

D=K\P                         %The structural displacements in
global coordinates.

flocal=localframememberforces(NumElem,C,Contribution,D,klocal,R)
```

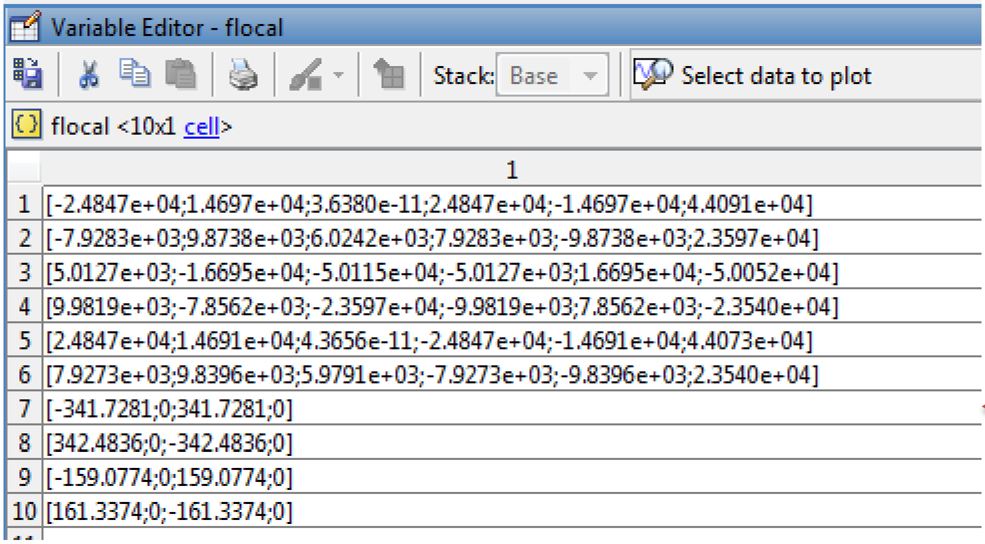
Displacements (in m) ;

Variable Editor - D

D <14x1 double>

	1	2
1	-7.4893e-04	
2	0.0018	
3	9.9389e-06	
4	-3.2497e-04	
5	0.0027	
6	1.3110e-05	
7	-1.5600e-04	
8	-7.4753e-04	
9	0.0018	
10	-9.9387e-06	
11	-3.2375e-04	
12	0.0027	
13	-1.3110e-05	
14	-1.5490e-04	

Forces (in N and m);



Variable Editor - flocal

Stack: Base | Select data to plot

flocal <10x1 cell>

	1
1	[-2.4847e+04;1.4697e+04;3.6380e-11;2.4847e+04;-1.4697e+04;4.4091e+04]
2	[-7.9283e+03;9.8738e+03;6.0242e+03;7.9283e+03;-9.8738e+03;2.3597e+04]
3	[5.0127e+03;-1.6695e+04;-5.0115e+04;-5.0127e+03;1.6695e+04;-5.0052e+04]
4	[9.9819e+03;-7.8562e+03;-2.3597e+04;-9.9819e+03;7.8562e+03;-2.3540e+04]
5	[2.4847e+04;1.4691e+04;4.3656e-11;-2.4847e+04;-1.4691e+04;4.4073e+04]
6	[7.9273e+03;9.8396e+03;5.9791e+03;-7.9273e+03;-9.8396e+03;2.3540e+04]
7	[-341.7281;0;341.7281;0]
8	[342.4836;0;-342.4836;0]
9	[-159.0774;0;159.0774;0]
10	[161.3374;0;-161.3374;0]