Gökay KART
21832009

23.12.2022

HACETTEPE UNIVERSITY

Department of Mechanical Engineering

MMÜ420 Finite Element Analysis

Term Project

Gökay KART

21832009

Gökay KART
21832009

23.12.2022

2-D Truss System

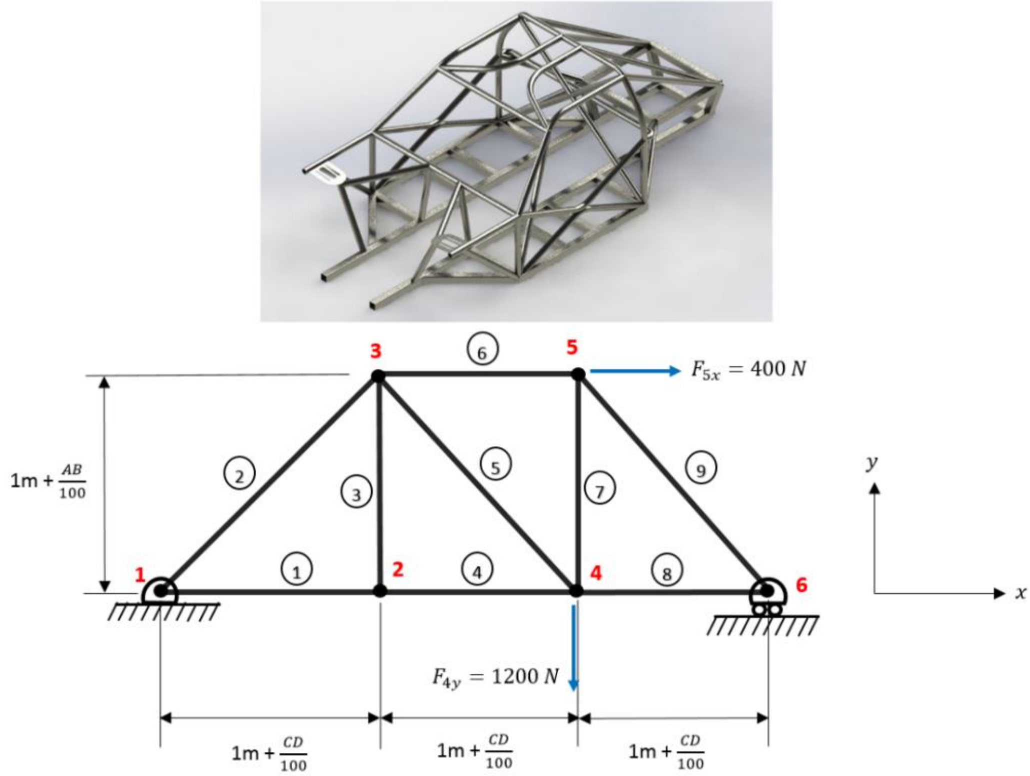

Fig. 1. The space frame type car chassis and its some portion projected onto xy-plane.

Data given for the question

- Elastic modulus, E = 200 GPa
- Poisson's ratio, v = 0.30
- Diameter, d = 0.0564 m
- Cross-sectional area, A =(π*d^2)/4 = 0.0025 m^2
- AB = 21 → 1 + 21/100 = 1.21 m
- CD = 09 → 1 +   9/100 = 1.09 m

A-) The MATLAB code for the Truss System in **Fig.1** was created as shown in *APPENDIX-1*. While creating the MATLAB code, the 3-element Truss System example given in the lesson was used as a source. Analysis is done in MATLAB application, nodal displacement values are determined. The displacement magnitude is mm.

Gökay KART
21832009
23.12.2022

```
Displacement_RESULT =

    1.0000    0.0000
    2.0000   -0.0000
    3.0000    0.0014
    4.0000   -0.0056
    5.0000    0.0046
    6.0000   -0.0056
    7.0000    0.0027
    8.0000   -0.0088
    9.0000    0.0036
   10.0000   -0.0065
   11.0000    0.0046
   12.0000         0
```

B-) The MATLAB code for the Truss System in **Fig.1** was created as shown in *APPENDIX-1*. Analysis is done in MATLAB application, element stress values are determined. The stress magnitude is MPa.

```
Stress_RESULT =

    1.0000    0.2508
    2.0000   -0.1357
    3.0000         0
    4.0000    0.2508
    5.0000    0.1357
    6.0000   -0.1816
    7.0000    0.3792
    8.0000    0.3416
    9.0000   -0.5104
```

C-) The MATLAB code for the Truss System in **Fig.1** was created as shown in *APPENDIX-1*. Analysis is done in MATLAB application. The undeformed and deformed shapes of the lattice system are drawn as seen in **Fig.2** and **Fig.3**. The real forces given in the question in **Fig.2** have been used, but the forces have been increased *1*10^4* times in **Fig.3** to make the deformations in the figure clearer.

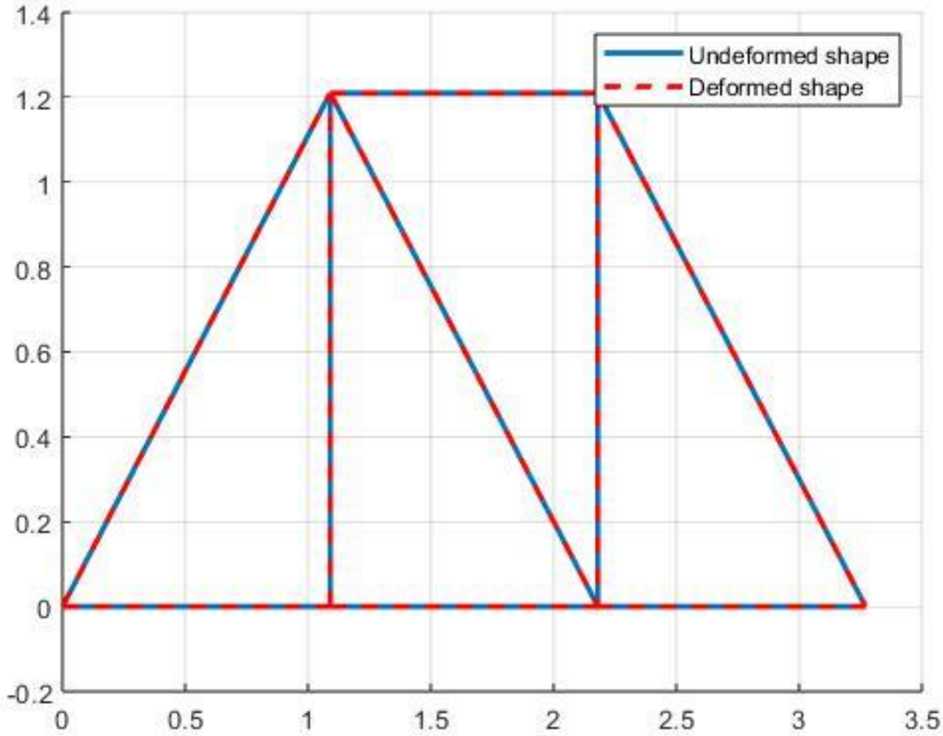

Fig. 2. The Undeformed and Deformed Shape in MATLAP. (with Real Force)

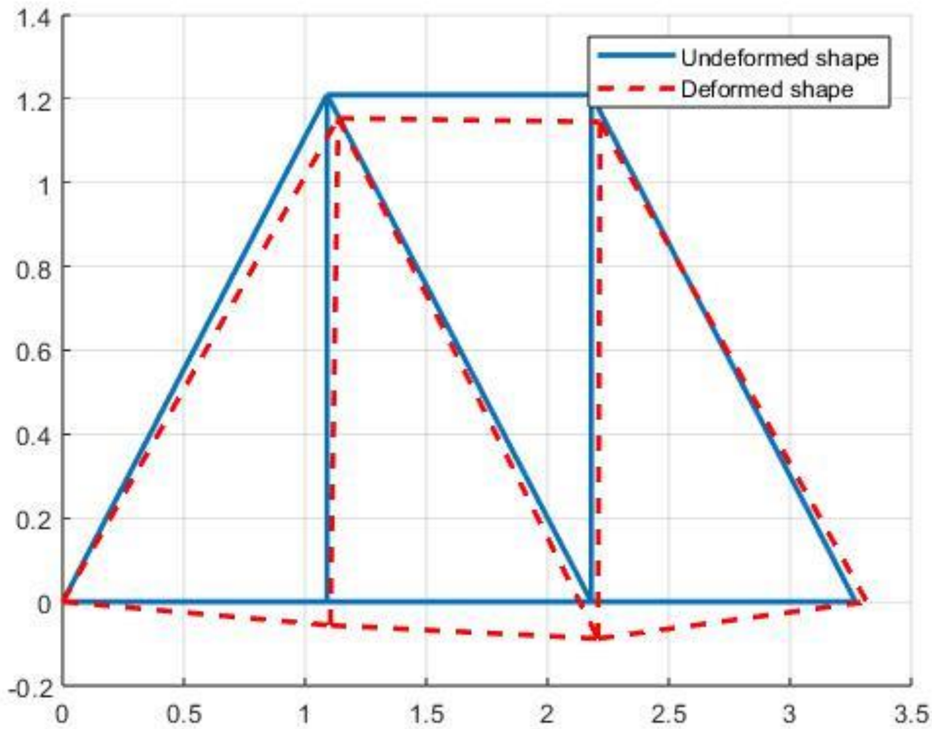

Fig. 3. The Undeformed and Deformed Shape in MATLAP. (with Increased Force)

Gökay KART
21832009
23.12.2022

D-) The given APDL code is customized as shown in *APPENDIX-2*. Then the Mechanical APDL program is run. The code is run in the application. Then, the analysis is done with the steps given below.

- To determine the Nodal displacement;

Main Menu >> General Postproc >> List Result >> Nodal Solution >> DOF Solution >> Displacement vector sum >> Apply.

- To determine the Element stresses;

Main Menu >> General Postproc >> List Result >> Element Solution >> Stress >> von Mises stress >> Apply.

- To determine the Undeformed and Deformed shapes;

Main Menu >> General Postproc >> Plot Result >> Deformed Shape >> Def +undeformed >> Apply.

- Node displacement results using Link 180 elements

```
PRINT U    NODAL SOLUTION PER NODE

 ***** POST1 NODAL DEGREE OF FREEDOM LISTING *****

 LOAD STEP=     0  SUBSTEP=     1
  TIME=    1.0000      LOAD CASE=   0

 THE FOLLOWING DEGREE OF FREEDOM RESULTS ARE IN THE GLOBAL COORDINATE SYSTEM

   NODE       UX
      1   0.0000
      2  0.13669E-005
      3  0.45913E-005
      4  0.27337E-005
      5  0.36016E-005
      6  0.45954E-005

MAXIMUM ABSOLUTE VALUES
NODE          6
VALUE   0.45954E-005
```

```
PRINT U    NODAL SOLUTION PER NODE

 ***** POST1 NODAL DEGREE OF FREEDOM LISTING *****

 LOAD STEP=     0  SUBSTEP=     1
  TIME=    1.0000      LOAD CASE=   0

 THE FOLLOWING DEGREE OF FREEDOM RESULTS ARE IN THE GLOBAL COORDINATE SYSTEM

   NODE       UY
      1   0.0000
      2 -0.56228E-005
      3 -0.56228E-005
      4 -0.87829E-005
      5 -0.64887E-005
      6   0.0000

MAXIMUM ABSOLUTE VALUES
NODE          4
VALUE  -0.87829E-005
```

Gökay KART
21832009

23.12.2022

- Element Stress (von Mises) results using Link 180 elements

```
PRINT S    PRIN ELEMENT SOLUTION PER ELEMENT

***** POST1 ELEMENT NODAL STRESS LISTING *****

 LOAD STEP=     0  SUBSTEP=     1
  TIME=    1.0000      LOAD CASE=   0

 THE FOLLOWING X,Y,Z VALUES ARE IN GLOBAL COORDINATES


 ELEMENT=       1        LINK180
   NODE    S1            S2           S3            SINT        SEQV
     1  0.25080E+006  0.0000       0.0000       0.25080E+006 0.25080E+006
     2  0.25080E+006  0.0000       0.0000       0.25080E+006 0.25080E+006

 ELEMENT=       2        LINK180
   NODE    S1            S2           S3            SINT        SEQV
     1   0.0000       0.0000      -0.13566E+006 0.13566E+006 0.13566E+006
     3   0.0000       0.0000      -0.13566E+006 0.13566E+006 0.13566E+006

 ELEMENT=       3        LINK180
   NODE    S1            S2           S3            SINT        SEQV
     2   0.0000       0.0000       0.0000        0.0000       0.0000
     3   0.0000       0.0000       0.0000        0.0000       0.0000

 ELEMENT=       4        LINK180
   NODE    S1            S2           S3            SINT        SEQV
     2  0.25080E+006  0.0000       0.0000       0.25080E+006 0.25080E+006
     4  0.25080E+006  0.0000       0.0000       0.25080E+006 0.25080E+006

 ELEMENT=       5        LINK180
   NODE    S1            S2           S3            SINT        SEQV
     3  0.13566E+006  0.0000       0.0000       0.13566E+006 0.13566E+006
     4  0.13566E+006  0.0000       0.0000       0.13566E+006 0.13566E+006

 ELEMENT=       6        LINK180
   NODE    S1            S2           S3            SINT        SEQV
     3   0.0000       0.0000      -0.18160E+006 0.18160E+006 0.18160E+006
     5   0.0000       0.0000      -0.18160E+006 0.18160E+006 0.18160E+006

 ELEMENT=       7        LINK180
   NODE    S1            S2           S3            SINT        SEQV
     4  0.37920E+006  0.0000       0.0000       0.37920E+006 0.37920E+006
     5  0.37920E+006  0.0000       0.0000       0.37920E+006 0.37920E+006

***** POST1 ELEMENT NODAL STRESS LISTING *****

 LOAD STEP=     0  SUBSTEP=     1
  TIME=    1.0000      LOAD CASE=   0

 THE FOLLOWING X,Y,Z VALUES ARE IN GLOBAL COORDINATES


 ELEMENT=       8        LINK180
   NODE    S1            S2           S3            SINT        SEQV
     4  0.34160E+006  0.0000       0.0000       0.34160E+006 0.34160E+006
     6  0.34160E+006  0.0000       0.0000       0.34160E+006 0.34160E+006

 ELEMENT=       9        LINK180
   NODE    S1            S2           S3            SINT        SEQV
     5   0.0000       0.0000      -0.51038E+006 0.51038E+006 0.51038E+006
     6   0.0000       0.0000      -0.51038E+006 0.51038E+006 0.51038E+006
```

- The Undeformed and Deformed shapes results using Link 180 Elements

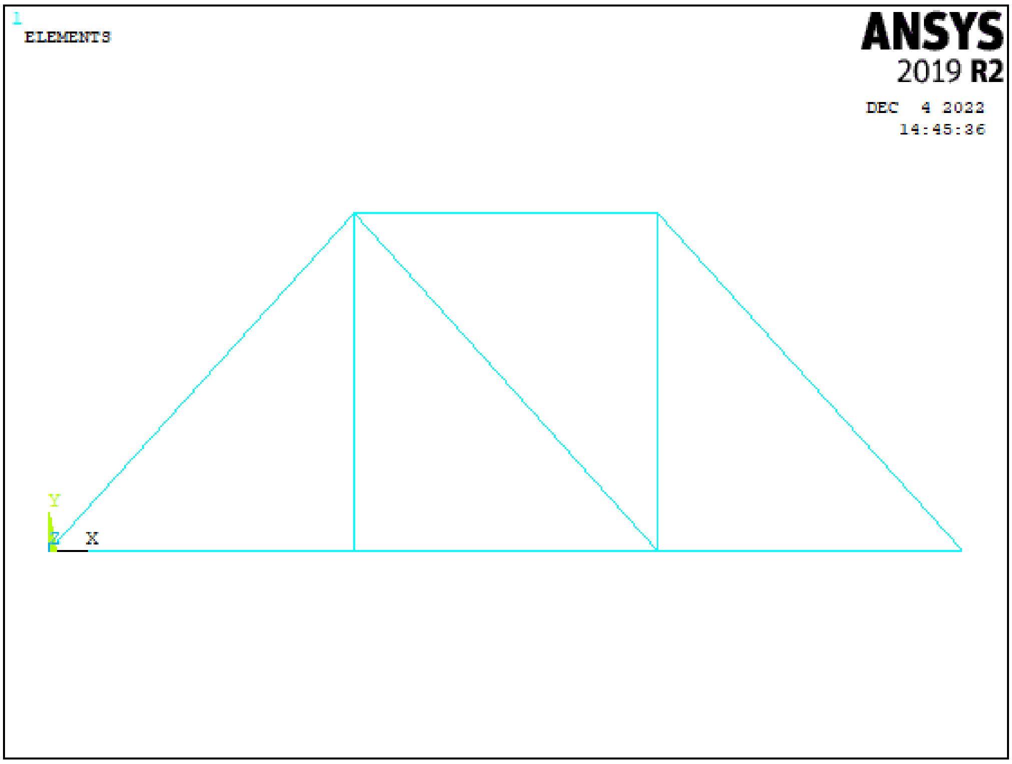

Fig.4. The Truss System in APDL notation.

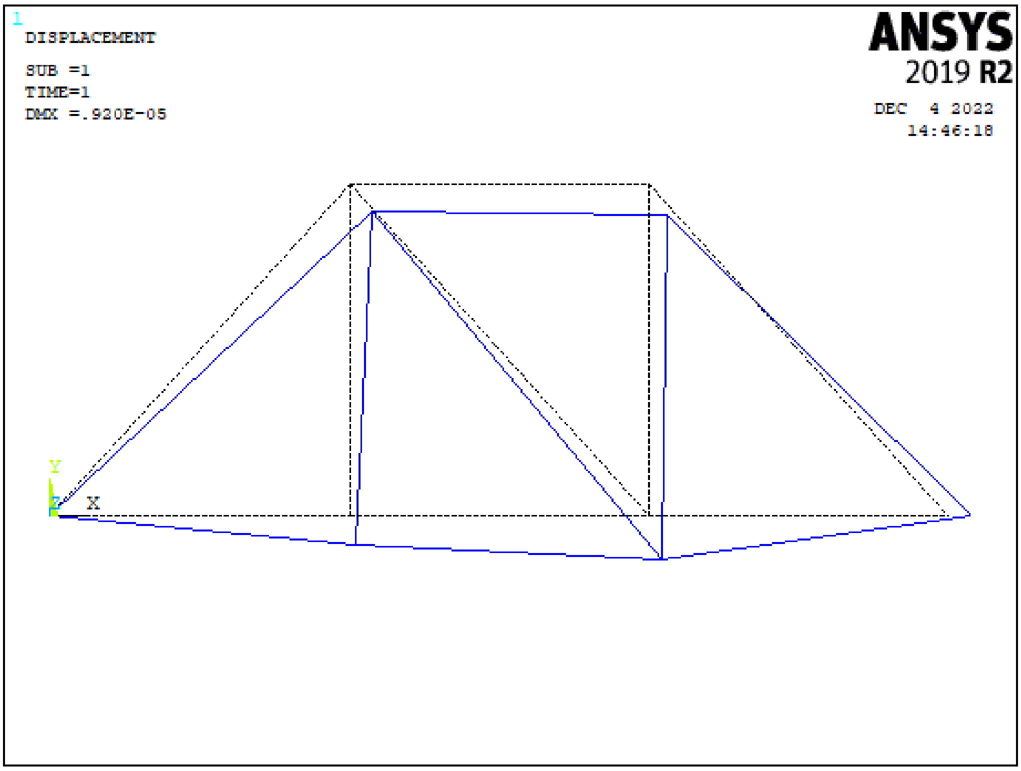

Fig.5. The Undeformed and Deformed Shape in ANSYS.

Gökay KART
21832009

23.12.2022

E-) Truss System Analysis was solved with two different programs. As seen in **Table 1** and **Table 2**, if we first compare the numerical values in the analyzes, the results have less than one stub of a thousand. This similarity shows that there is no error in numerical operations. The reason for the difference is the rounding error criteria in two different programs. The Ansys program gives the user the option to scale the real value while presenting it graphically. The deformation in **Fig.5** is the enlarged form of the real values. In MATLAB application, changes must be made on the code in order to scale. **Fig.2** shows the deformation to which the actual force values are applied. Since this amount of deformation is quite small, it is not clearly visible on the figure. In the scenario where we increase the force values ten thousand times, the deformation in **Fig.3** is visible.

Table 1. Nodal displacements obtained by FE code and ANSYS Mechanical.

FE Code in MATLAB Displacement (mm)				FE Code in ANSYS Displacement (mm)			
Node 1 Ux	0.0000	Node 1 Uy	-0.0000	Node 1 Ux	0.0000	Node 1 Uy	-0.000000
Node 2 Ux	0.0014	Node 2 Uy	-0.0056	Node 2 Ux	0.001367	Node 2 Uy	-0.005623
Node 3 Ux	0.0046	Node 3 Uy	-0.0056	Node 3 Ux	0.004591	Node 3 Uy	-0.005623
Node 4 Ux	0.0027	Node 4 Uy	-0.0088	Node 4 Ux	0.002734	Node 4 Uy	-0.008783
Node 5 Ux	0.0036	Node 5 Uy	-0.0065	Node 5 Ux	0.003602	Node 5 Uy	-0.006489
Node 6 Ux	0.0046	Node 6 Uy	-0.0000	Node 6 Ux	0.004594	Node 6 Uy	-0.000000

Table 2. Element stresses obtained by FE code and ANSYS Mechanical.

FE Code in MATLAB Stress (MPa)		FE Code in ANSYS Stress (Von Mises) (MPa)	
Element – 1	0.2508	Element – 1	0.2508
Element – 2	-0.1357	Element – 2	-0.13566
Element – 3	0.000	Element – 3	0.0000
Element – 4	0.2508	Element – 4	0.25080
Element – 5	0.1357	Element – 5	0.13566
Element – 6	-0.1816	Element – 6	-0.18160
Element – 7	0.3792	Element – 7	0.37920
Element – 8	0.3416	Element – 8	0.34160
Element – 9	-0.5104	Element – 9	-0.5138

Gökay KART
21832009

23.12.2022

APPENDIX - 1

MATLAP CODE

```
clc,clear, clear all;


% PRE-PROCESSING STEP

nel=9;                              % Number of elements
nnel=2;                             % Number of nodes per element
ndof=2;                             % Number of dofs per node
nnode=6;                            % Total number of nodes in system
sdof=nnode*ndof;                    % Total system dofs

% Nodal coordinates of the structure
% AB=21 , CD=09
gcoord(1,1)=0.0;  gcoord(1,2)=0.0;    % x,y coordinate of node 1
gcoord(2,1)=1.09;  gcoord(2,2)=0.0;   % x,y coordinate of node 2
gcoord(3,1)=1.09;  gcoord(3,2)=1.21;  % x,y coordinate of node 3
gcoord(4,1)=2.18;  gcoord(4,2)=0.0;   % x,y coordinate of node 4
gcoord(5,1)=2.18;  gcoord(5,2)=1.21;  % x,y coordinate of node 5
gcoord(6,1)=3.27;  gcoord(6,2)=0.0;   % x,y coordinate of node 6

% Material and geometric properties
% All elements
elprop(1)=200e9;                   % Elastic modulus
elprop(2)=0.0025;                  % Cross-sectional area [m^2]

% Nodal connectivity matrix

nodes(1,1)=1;  nodes(1,2)=2;         % Nodes associated with Element-1
nodes(2,1)=1;  nodes(2,2)=3;         % Nodes associated with Element-2
nodes(3,1)=2;  nodes(3,2)=3;         % Nodes associated with Element-3
nodes(4,1)=2;  nodes(4,2)=4;         % Nodes associated with Element-4
nodes(5,1)=3;  nodes(5,2)=4;         % Nodes associated with Element-5
nodes(6,1)=3;  nodes(6,2)=5;         % Nodes associated with Element-6
nodes(7,1)=4;  nodes(7,2)=5;         % Nodes associated with Element-7
nodes(8,1)=4;  nodes(8,2)=6;         % Nodes associated with Element-8
nodes(9,1)=5;  nodes(9,2)=6;         % Nodes associated with Element-9

% Applied constraints

bcdof(1)=1;       % 1st dof (horizontal displ) is constrained.
bcval(1)=0;       % whose described value is 0.
bcdof(2)=2;       % 2nd dof (vertical displ) is constrained.
bcval(2)=0;       % whose described value is 0.
bcdof(3)=12;      % 12th dof (vertical displ) is constrained.
bcval(3)=0;       % whose described value is 0.

% Initialization with zero.
ff=zeros(sdof,1);                   % System force vector.
% ff=(Node-1 x direction;
%     Node-1 y direction;
%     Node-2 x direction;
%     Node-2 y direction;
%         ...
```

```
%          ...
%          ...
%     Node-6 x direction;
%     Node-6 y direction;
%     );
K=zeros(sdof,sdof);                  % Global system stiffness matrix.
index=zeros(nnel*ndof,1);            % index vector.
elforce=zeros(nnel*ndof,1);          % Element force vector
eldisp=zeros(nnel*ndof,1);           % Element nodal displacement vector.
ke=zeros(nnel*ndof,nnel*ndof);       % Element stiffness matrix.
stress=zeros(nel,1);                 % Stress vector for every element.

% Applied nodal force

ff(8)=-1200;      % Node 4 has a 1200[N] load in downwards.
ff(9)=400;        % Node 5 has a 400[N] load in +x directions.

% Loop for elements

for i=1:nel                                    % Loop for the total number of
elements
    nd(1)=nodes(i,1);                          % 1st connected node for the
(i)th element
    nd(2)=nodes(i,2);                          % 2nd connected node for the
(i)th element

    x1=gcoord(nd(1),1); y1=gcoord(nd(1),2);   % Coordinate of the 1st node
    x2=gcoord(nd(2),1); y2=gcoord(nd(2),2);   % Coordinate of the 2nd node

    leng=sqrt((x2-x1)^2+(y2-y1)^2);             % Element length

    if(x2-x1)==0
        beta=2*atan(1);                         % Angle between local and
gobal axes
    else
        beta=atan((y2-y1)/(x2-x1));             % Angle between local and
gobal axes
    end

    el=elprop(1);                               % Extract elastic modulus
    area=elprop(2);                             % Extract cross-sectional
area
    index=eldof(nd,nnel,ndof);                  % Extract system dofs for the
element
    ke=fetruss2(el,leng,area,0,beta,1);         % Compute Element Stiffness
Matrix-[ke]
    K=assembly(K,ke,index);                     % Assemble to Global
Stiffness Matrix

end

% Apply constraints and solve the matrix
[K,ff]=applyBC(K,ff,bcdof,bcval);               % Apply Boundary Conditions
(BCs)

% SOLUTION STEP
```

Gökay KART
21832009

23.12.2022

```
disp=K\ff;                                 % Solve matrix equation for
nodal displacements

% POST-PROCESSING STEP
% Post-computation for stress calculation

for i=1:nel                                % Loop for the total number
of elements
    nd(1)=nodes(i,1);                      % 1st connected node for the
(i)th element
    nd(2)=nodes(i,2);                      % 2nd connected node for the
(i)th element
    x1=gcoord(nd(1),1); y1=gcoord(nd(1),2);   % Coordinate of the 1st node
    x2=gcoord(nd(2),1); y2=gcoord(nd(2),2);   % Coordinate of the 2nd node

    leng=sqrt((x2-x1)^2+(y2-y1)^2);        % Element length

    if(x2-x1)==0
        beta=2*atan(1);                    % Angle between local and
gobal axes
    else
        beta=atan((y2-y1)/(x2-x1));        % Angle between local and
gobal axes
    end

    el=elprop(1);                          % Extract elastic modulus
    area=elprop(2);                        % Extract cross-sectional
area
    index=eldof(nd,nnel,ndof);             % Extract system dofs for the
element
    ke=fetruss2(el,leng,area,0,beta,1);    % Compute Element Stiffness
Matrix [ke]

    for j=1:(nnel*ndof)                    % Extract displacement
associated with (i)th element.
        eldisp(j)=disp(index(j));          % (i)th element
    end

    elforce=ke*eldisp;                     % Element force vector
    stress(i)=sqrt(elforce(1)^2+elforce(2)^2)/area; % Stress.

    if((x2-x1)*elforce(3))<0               % Check if tension or
commpression
        stress(i)=-stress(i);
    end

end

% Print FEM Solutions

disp_mm=disp*1000;                         % Displacement in [mm]s.
num1=1:1:sdof;
displacement_RESULT=[num1' disp_mm]        % Print displacments.
num2=1:1:nel;
stress_RESULT=[num2' stress/(10^6)]        % Print stresses in [MPa]s.

% PLOTTING OF DEFORMED SHAPE WITH UNDEFORMED ONE
```

Gökay KART
21832009

23.12.2022

```
% Undeformed shape

% Element-1
% x11: Element-1, Node-1 y11: Element-1, Node-1
% x12: Element-1, Node-2 y12: Element-1, Node-2
x11=gcoord(1,1);  y11=gcoord(1,2);  % x,y coordinate of node 1
x12=gcoord(2,1);  y12=gcoord(2,2);  % x,y coordinate of node 2

% Element-2
% x21: Element-2, Node-2 y21: Element-2, Node-1
% x22: Element-2, Node-3 y22: Element-2, Node-3
x21=gcoord(1,1);  y21=gcoord(1,2);  % x,y coordinate of node 1
x22=gcoord(3,1);  y22=gcoord(3,2);  % x,y coordinate of node 3

% Element-3 deformed coordinates
% x31: Element-2, Node-2 y21: Element-2, Node-2
% x32: Element-2, Node-3 y22: Element-2, Node-3
x31=gcoord(2,1);  y31=gcoord(2,2);  % x,y coordinate of node 2
x32=gcoord(3,1);  y32=gcoord(3,2);  % x,y coordinate of node 3

% Element-4 deformed coordinates
% x41: Element-2, Node-2 y41: Element-2, Node-2
% x42: Element-2, Node-4 y42: Element-2, Node-4
x41=gcoord(2,1);  y41=gcoord(2,2);  % x,y coordinate of node 2
x42=gcoord(4,1);  y42=gcoord(4,2);  % x,y coordinate of node 4

% Element-5 deformed coordinates
% x51: Element-2, Node-1 y51: Element-2, Node-3
% x52: Element-2, Node-3 y52: Element-2, Node-4
x51=gcoord(3,1);  y51=gcoord(3,2);  % x,y coordinate of node 3
x52=gcoord(4,1);  y52=gcoord(4,2);  % x,y coordinate of node 4

% Element-6 deformed coordinates
% x61: Element-2, Node-1 y21: Element-2, Node-3
% x62: Element-2, Node-3 y22: Element-2, Node-5
x61=gcoord(3,1);  y61=gcoord(3,2);  % x,y coordinate of node 3
x62=gcoord(5,1);  y62=gcoord(5,2);  % x,y coordinate of node 5

% Element-7 deformed coordinates
% x71: Element-2, Node-1 y21: Element-2, Node-5
% x72: Element-2, Node-3 y22: Element-2, Node-4
x71=gcoord(5,1);  y71=gcoord(5,2);  % x,y coordinate of node 5
x72=gcoord(4,1);  y72=gcoord(4,2);  % x,y coordinate of node 4

% Element-8 deformed coordinates
% x81: Element-2, Node-1 y21: Element-2, Node-4
% x82: Element-2, Node-3 y22: Element-2, Node-6
x81=gcoord(4,1);   y81=gcoord(4,2); % x,y coordinate of node 4
x82=gcoord(6,1);   y82=gcoord(6,2); % x,y coordinate of node 6

% Element-9 deformed coordinates
% x91: Element-2, Node-1 y21: Element-2, Node-5
% x92: Element-2, Node-3 y22: Element-2, Node-6
x91=gcoord(5,1);   y91=gcoord(5,2);  % x,y coordinate of node 5
x92=gcoord(6,1);   y92=gcoord(6,2);  % x,y coordinate of node 6

figure(1)
```

```
L(1)=line([x11 x12],[y11 y12],'linewidth', 2.0);
L(2)=line([x21 x22],[y21 y22],'linewidth', 2.0);
L(3)=line([x31 x32],[y31 y32],'linewidth', 2.0);
L(4)=line([x41 x42],[y41 y42],'linewidth', 2.0);
L(5)=line([x51 x52],[y51 y52],'linewidth', 2.0);
L(6)=line([x61 x62],[y61 y62],'linewidth', 2.0);
L(7)=line([x71 x72],[y71 y72],'linewidth', 2.0);
L(8)=line([x81 x82],[y81 y82],'linewidth', 2.0);
L(9)=line([x91 x92],[y91 y92],'linewidth', 2.0);
grid on;
hold on;

% % Deformed shape

% Element-1 deformed coordinates
% xd11: Element-1, Node-1 yd11: Element-1, Node-1
% xd12: Element-1, Node-2 yd12: Element-1, Node-2
 xd11=gcoord(1,1); yd11=gcoord(1,2);                          % x,y coordinate
of node 1
 xd12=gcoord(2,1)+disp(3); yd12=gcoord(2,2)+disp(4);          % x,y coordinate
of node 2

% Element-2 deformed coordinates
% xd21: Element-2, Node-1 yd21: Element-2, Node-1
% xd22: Element-2, Node-3 yd22: Element-2, Node-3
 xd21=gcoord(1,1);  yd21=gcoord(1,2);                        % x,y coordinate
of node 1
 xd22=gcoord(3,1)+disp(5);  yd22=gcoord(3,2)+disp(6);        % x,y coordinate
of node 3

% Element-3 deformed coordinates
% xd31: Element-2, Node-2 yd21: Element-2, Node-2
% xd32: Element-2, Node-3 yd22: Element-2, Node-3
xd31=gcoord(2,1)+disp(3);  yd31=gcoord(2,2)+disp(4);         % x,y coordinate
of node 2
xd32=gcoord(3,1)+disp(5);  yd32=gcoord(3,2)+disp(6);         % x,y coordinate
of node 3

% Element-4 deformed coordinates
% xd41: Element-2, Node-2 yd41: Element-2, Node-2
% xd42: Element-2, Node-4 yd42: Element-2, Node-4
xd41=gcoord(2,1)+disp(3);  yd41=gcoord(2,2)+disp(4);         % x,y coordinate
of node 2
xd42=gcoord(4,1)+disp(7);  yd42=gcoord(4,2)+disp(8);         % x,y coordinate
of node 4

% Element-5 deformed coordinates
% xd51: Element-2, Node-1 yd51: Element-2, Node-3
% xd52: Element-2, Node-3 yd52: Element-2, Node-4
xd51=gcoord(3,1)+disp(5);  yd51=gcoord(3,2)+disp(6);         % x,y coordinate
of node 3
xd52=gcoord(4,1)+disp(7);  yd52=gcoord(4,2)+disp(8);         % x,y coordinate
of node 4

% Element-6 deformed coordinates
% xd61: Element-2, Node-1 yd21: Element-2, Node-3
% xd62: Element-2, Node-3 yd22: Element-2, Node-5
```

```
xd61=gcoord(3,1)+disp(5);  yd61=gcoord(3,2)+disp(6);     % x,y coordinate
of node 3
xd62=gcoord(5,1)+disp(9);  yd62=gcoord(5,2)+disp(10);    % x,y coordinate
of node 5

% Element-7 deformed coordinates
% xd71: Element-2, Node-1 yd21: Element-2, Node-5
% xd72: Element-2, Node-3 yd22: Element-2, Node-4
xd71=gcoord(5,1)+disp(9);  yd71=gcoord(5,2)+disp(10);    % x,y coordinate
of node 5
xd72=gcoord(4,1)+disp(7);  yd72=gcoord(4,2)+disp(8);     % x,y coordinate
of node 4

% Element-8 deformed coordinates
% xd81: Element-2, Node-1 yd21: Element-2, Node-4
% xd82: Element-2, Node-3 yd22: Element-2, Node-6
xd81=gcoord(4,1)+disp(7);  yd81=gcoord(4,2)+disp(8);     % x,y coordinate
of node 4
xd82=gcoord(6,1)+disp(11); yd82=gcoord(6,2)+disp(12);    % x,y coordinate
of node 6

% Element-9 deformed coordinates
% xd91: Element-2, Node-1 yd21: Element-2, Node-5
% xd92: Element-2, Node-3 yd22: Element-2, Node-6
xd91=gcoord(5,1)+disp(9);  yd91=gcoord(5,2)+disp(10);    % x,y coordinate
of node 5
xd92=gcoord(6,1)+disp(11); yd92=gcoord(6,2)+disp(12);    % x,y coordinate
of node 6


% % Plot deformed shape
L(10)=line([xd11 xd12],[yd11
yd12],'linewidth',2.0,'Color','red','LineStyle','--');
L(11)=line([xd21 xd22],[yd21
yd22],'linewidth',2.0,'Color','red','LineStyle','--');
L(12)=line([xd31 xd32],[yd31
yd32],'linewidth',2.0,'Color','red','LineStyle','--');
L(13)=line([xd41 xd42],[yd41
yd42],'linewidth',2.0,'Color','red','LineStyle','--');
L(14)=line([xd51 xd52],[yd51
yd52],'linewidth',2.0,'Color','red','LineStyle','--');
L(15)=line([xd61 xd62],[yd61
yd62],'linewidth',2.0,'Color','red','LineStyle','--');
L(16)=line([xd71 xd72],[yd71
yd72],'linewidth',2.0,'Color','red','LineStyle','--');
L(17)=line([xd81 xd82],[yd81
yd82],'linewidth',2.0,'Color','red','LineStyle','--');
L(18)=line([xd91 xd92],[yd91
yd92],'linewidth',2.0,'Color','red','LineStyle','--');

legend([L(1), L(10)],'Undeformed shape','Deformed shape');
% END
```

Gökay KART
21832009

23.12.2022

APPENDIX - 2

ANSYS APDL CODE

Only the section shown below has been updated in the APDL code.

Old version;

```
!*
K, 1, 0, 0, 0,
K, 2, 1, 0, 0,
K, 3, 1, 1, 0,
K, 4, 2, 0, 0,
K, 5, 2, 1, 0,
K, 6, 3, 0, 0,
!*
```

Updated version;

```
!*
K, 1, 0, 0, 0,
K, 2, 1.09, 0, 0,
K, 3, 1.09, 1.21, 0,
K, 4, 2.18, 0, 0,
K, 5, 2.18, 1.21, 0,
K, 6, 3.27, 0, 0,
!*
```