



Stony Brook University
Mechanical Engineering

MEC 529 – Introduction to Finite Element Methods

Project Report:

**Structural Analysis of a Truss Tower and Stress Analysis of a 2D plate
with a hole.**

Submitted by: Sravana Sumanth Koneru

ID: 112526331

December 11, 2020

PART 1 – Structural Analysis on a Truss Tower

Objective: The aim of this project is to perform a **structural analysis on a Tower truss**. Truss is one of the simplest and most used structural elements. The main feature of the truss is that the forces can only be transmitted along the axial direction, leading to a deformation only along axial direction. We perform a structural analysis on a tower which is made totally of trusses to calculate the deformations of the nodes and tensile and compressive stresses in the elements. **MATLAB** is used for the calculations and plotting in the analysis.

Problem Description and analysis: As stated above, a truss tower of 160m height and 50m width is used. The material properties of the truss are then defined. The truss is assumed to be made up of Steel with Modulus of elasticity **210GPa** and the area of each element is assumed to be **200mm²** i.e., $2.1 \times 10^{11} \text{ N/mm}^2$. The truss had **34 nodes** and has **65 elements**. The defined truss is shown in **Figure 1** (Black dotted structure).

For the analysis, all the nodes are first defined and numbered in a cyclic manner. Then they are assigned respective coordinates. The elements are then defined by connecting the required nodes. The fixed nodes in the truss are then defined. The forces are then defined on the required nodes. Then using MATLAB coding, the stiffness matrix of each element is determined and is assembled as a global stiffness matrix. Using the global stiffness matrix, known displacements and known forces, we determine the unknown forces and displacements of all the elements in the truss. The obtained data is then used to plot the deformed structure (**Figure 1**).

The following forces and boundary conditions are applied on the truss.

Boundary conditions and Forces used:

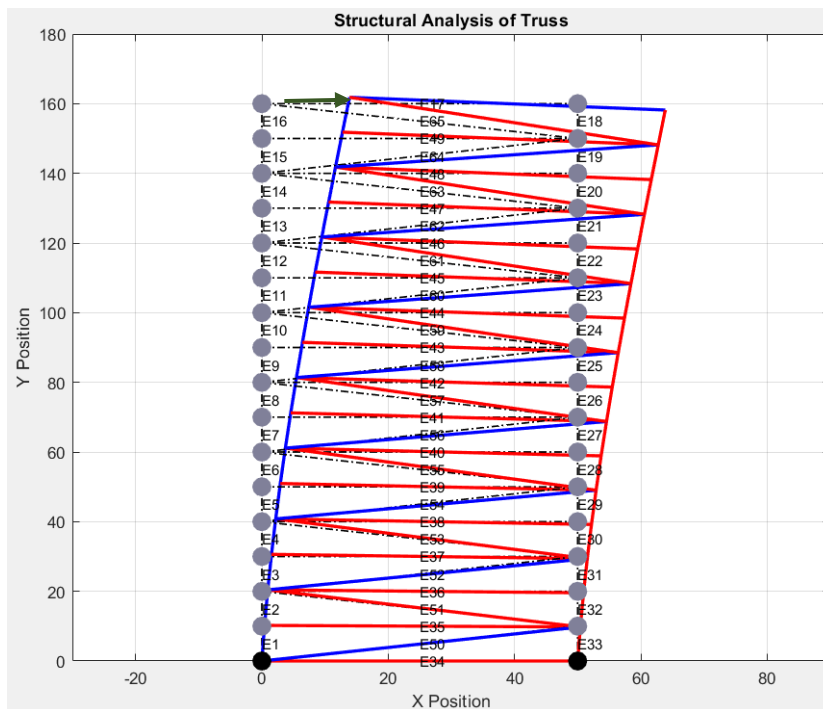
Total number of nodes taken is **34**. Total number of elements taken is **65**.

A force of **10KN** is applied on **Node 17** (0,160) in positive x direction. The truss is taken to be fixed at bottom nodes i.e. **Node 1 and Node 34**. Node 1 Position: (0,0), Node 34 position: (50,0).

Results:

The MATLAB code used is shown in **Appendix B** and the node numbers, coordinates are shown in **Appendix A**. The following **figure 1** shows the geometry of the truss, positions of the applied forces undeformed and deformed structure.

Figure 2 and **Figure 3** shows the value of stress applied on each node. Here, the negative value depicts the compression and positive value is for tension. Figure 2 depicts the change in stresses in the elements on either side of the truss and Figure 3 depicts the changes in stresses of the elements in the middle.



LEGEND:

- Dotted lines show undeformed structure.
- Blue solid lines show the truss elements under tension.
- Red solid lines show the truss elements under compression.
- Grey dots show free nodes.
- Black dots show fixed nodes.
- Forces are shown by the arrow mark.

Figure 1 Undeformed and Deformed Structure of the truss.

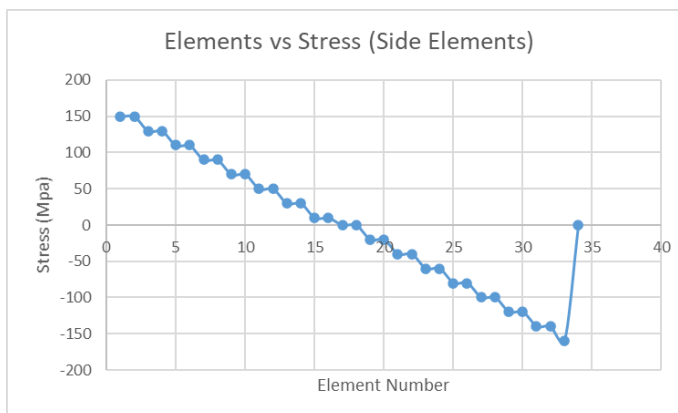


Figure 2 Elements vs Stresses in elements on the side

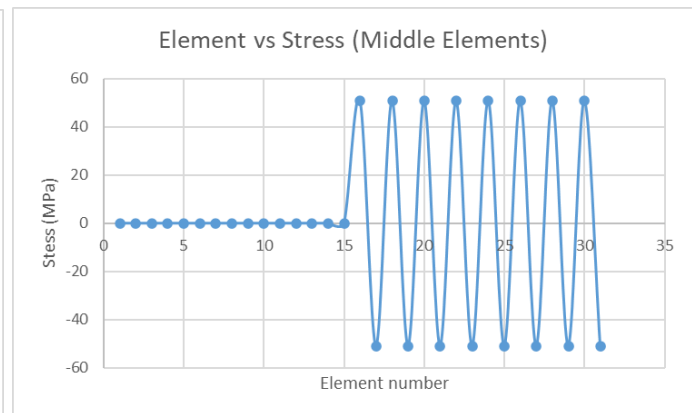


Figure 3. Elements vs Stresses in elements in the Middle elements

Conclusion:

From the structural analysis done, the deformed plot of the truss is plotted. From the stress plot, we can observe the changes in stresses in elements along the truss. A maximum tensile stress of **150 MPa** is observed in **element 1** and a maximum compressive stress of **150 MPa** is observed in **element 33**. As expected, tension is observed on the elements on the left, i.e., where the load is applied, and a compressive stress is observed for the elements under tension.

PART 2 – Stress Analysis in a 2D Plate with a hole

Objective:

The aim of this part of the project is to perform a stress analysis in a 2D Plate with a hole. For this a rectangular plate with a tilted square hole is used and taken as a plane stress problem (negligible thickness). **Constant Stress Triangles** is used for the analysis. The main goal is to perform analysis on this plate and calculate stresses and deformations in the plate. **MATLAB** is used to perform this analysis.

Problem Description and analysis done:

To achieve the goal of this project, first the part is designed with relevant dimensions. The material used for the plate is Steel and the modulus of elasticity is **210 GPa** and Poissons ratio is **0.3** and the thickness is **1 mm**.

For the analysis, the rectangular plate is divided into **60** three-noded triangular elements. The nodes are then numbered in a cyclic order. Then an element number is assigned to each element. All the nodes are given the respective coordinates and are input into the **MATLAB** to generate the geometry of the plate and the hole. The fixed nodes are then defined. The forces are then defined on the required nodes. We use a **constant stress triangle** to obtain the results. Then using MATLAB, the stiffness matrix of each element is calculated and assembled to form global stiffness matrix which we use in turn to find the displacements and stresses in each triangular element. The obtained data is then used to plot the Stress plot of the plate (**Figure 4**).

The following forces and boundary conditions are imposed on the plate.

Boundary conditions and Forces used:

Total number of nodes: **44**.

Total number of elements: **60**.

A force of **2.5 KN** is applied on **Nodes 9** and **44** in the positive x direction. The plate is fixed at Nodes 1, 18, 19, 35 and 36.

Results:

The MATLAB code used for the analysis is shown in **Appendix B** and the node numbers, coordinates are shown in **Appendix A**. The following **figure 4** shows the X stress plot, geometry, nodes and elements in the plate.

Figure 5 shows the change in stress in each element. Here, the negative value depicts the compression and positive value is for tension. **Figure 6** depicts the X and Y displacement in each node throughout the structure.

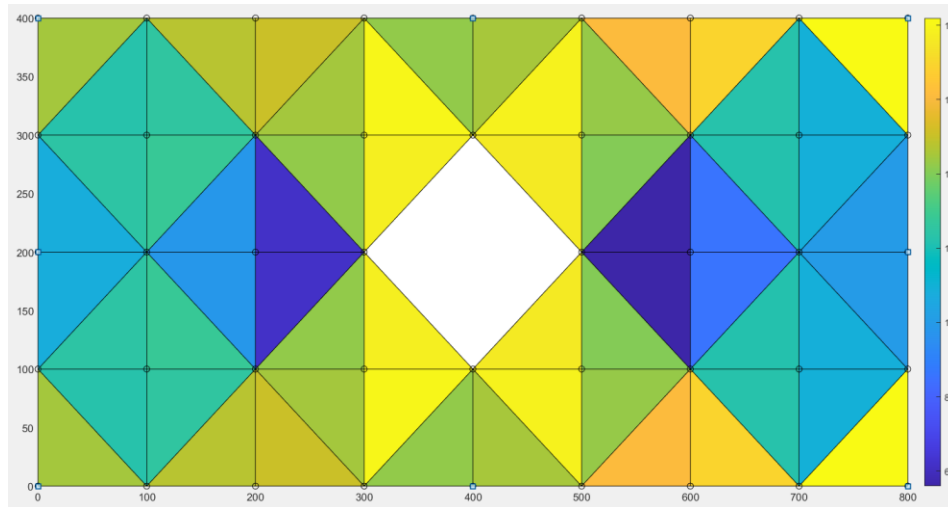


Figure 4. X stresses on Undeformed plate with a hole

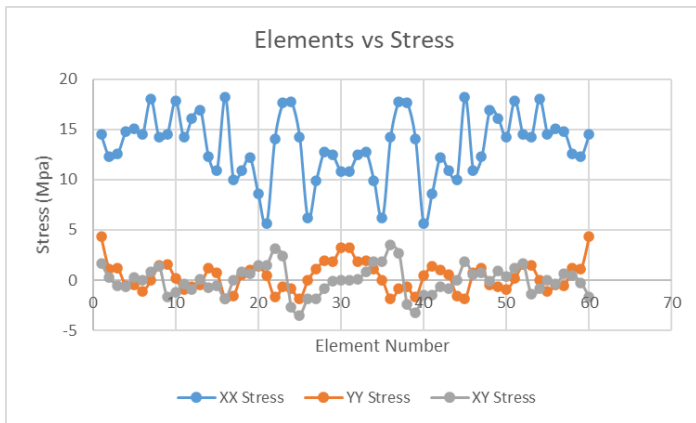


Figure 5 Elements vs Stresses

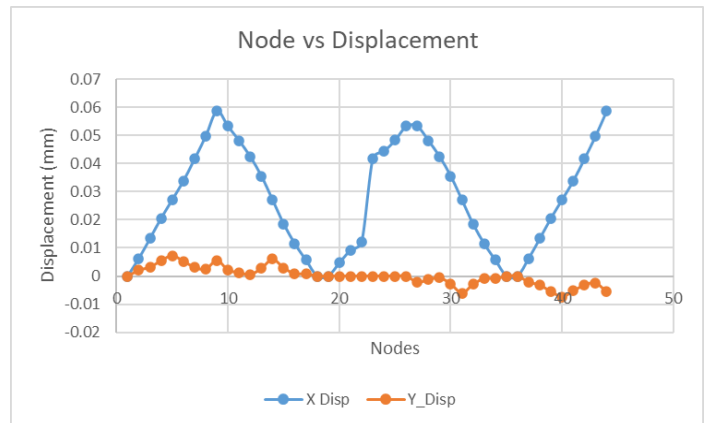


Figure 6. Nodes vs displacement

Conclusion:

From the structural analysis done, a solid simulation on a 2D plate with a hole is performed. The **XX - stress plot on the undeformed structure** is plotted. As expected, we see that the maximum stresses are found at the hole. For a force of **5000 KN** applied, a maximum stress of around **20Mpa** is observed. It can also be observed that the stresses in XX plane are higher compared to the other two stresses. The displacement can mostly be found in X direction which is expected as the load is applied only in X direction.

APPENDIX – A (Figures Used)

For Part 1 (Truss Tower):

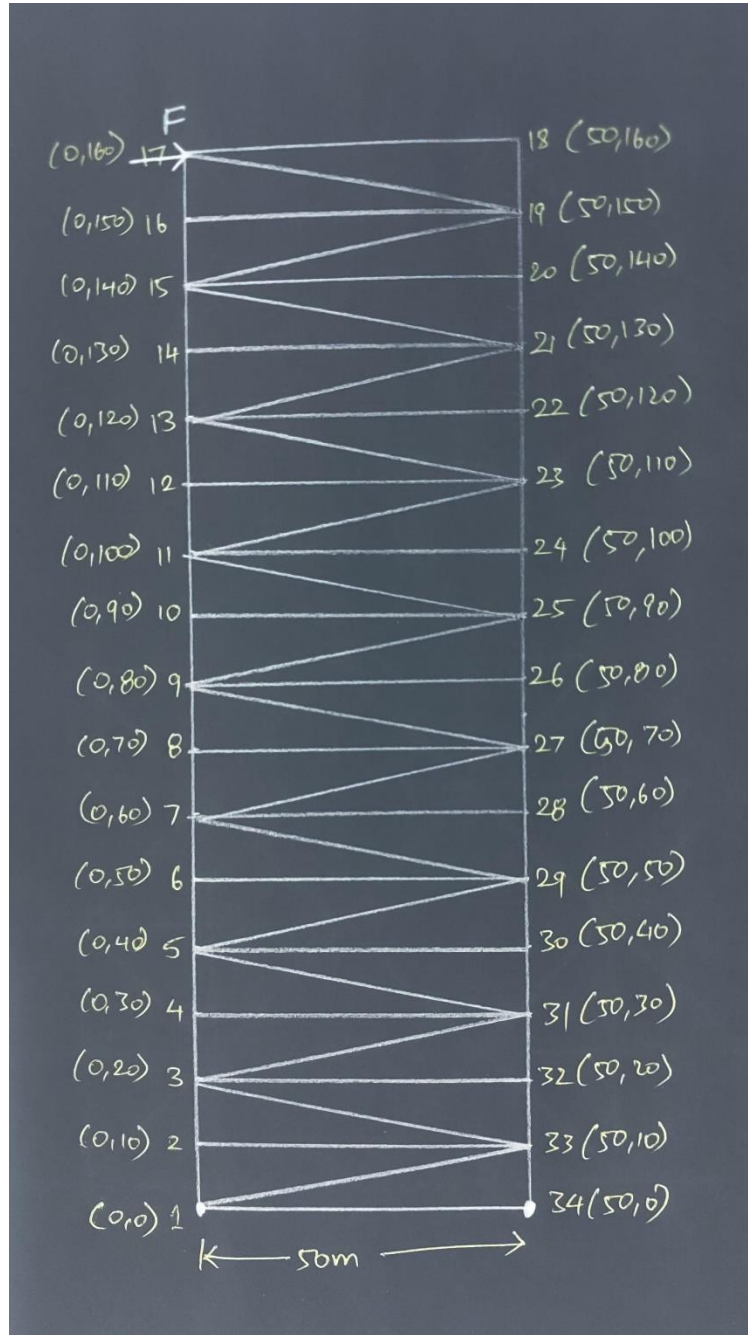


Figure 7. Coordinates and Node numbers used for truss in Part 1

For Part 2 (Stress Analysis in 2D Plate with a Hole):

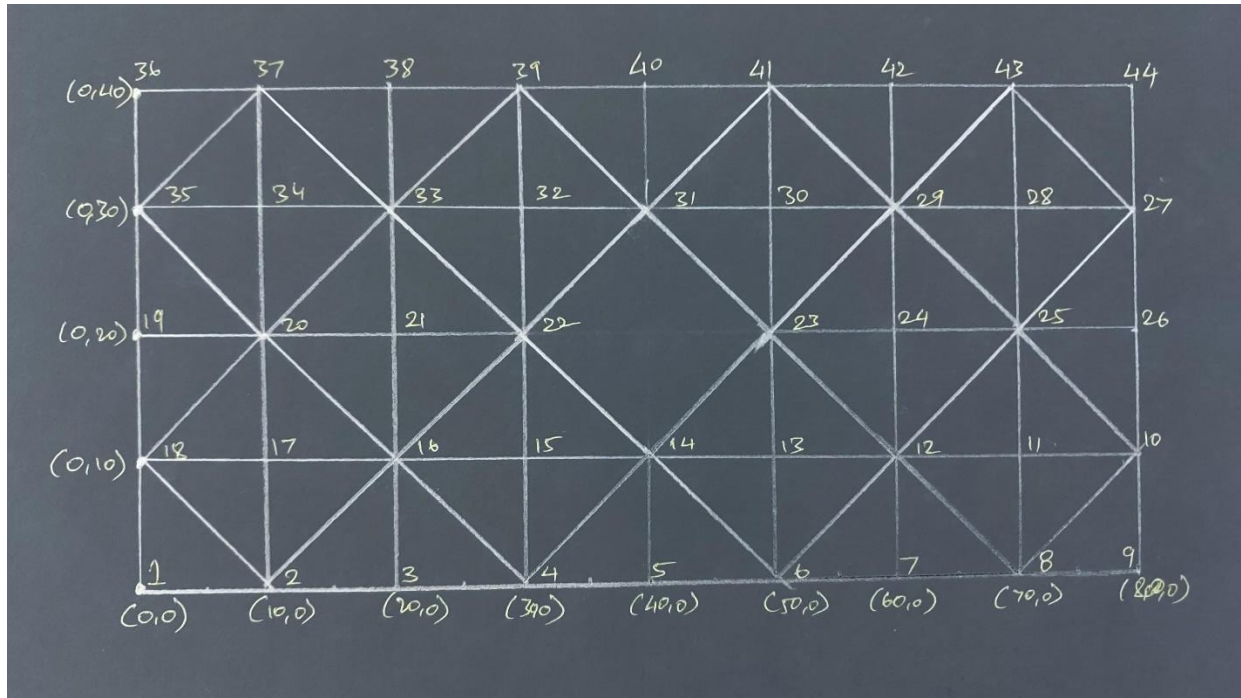


Figure 8. Coordinates and Node numbers used for truss in Part 2

APPENDIX - B (Codes Used)

MATLAB Code used for part 1:

```
% This is a project submission for MEC 539 - Introduction to Finite element
% Methods. This code does a structural analysis on a truss tower.
% Author: Sravana Sumanth Koneru
% ID : 112526331
% Stony Brook University

clc;
clear;

%% inputs
%area of element in sq.m.
Area = 0.0002* ones(1,70);

%Youngs Modulus in N/m2
E = 2.1e11 *ones(70);

% forces on nodes in N
%Defined by [Node_no. X_Force Y_Force]
Nodal_forces = [17 10000 0];

%Scaling Factor
Scaling = 30;

%position of nodes
%Defined by [Node_No X Y X_Fix Y_Fix]
%1 = Fixed, 0 = free;
Nodal_pos = [1 0 0 1 1;...
2 0 10 0 0;...
3 0 20 0 0;...
4 0 30 0 0;...
5 0 40 0 0;...
6 0 50 0 0;...
7 0 60 0 0;...
8 0 70 0 0;...
9 0 80 0 0;...
10 0 90 0 0;...
11 0 100 0 0;...
12 0 110 0 0;...
13 0 120 0 0;...
14 0 130 0 0;...
15 0 140 0 0;...
16 0 150 0 0;...
17 0 160 0 0;...
18 50 160 0 0;...
19 50 150 0 0;...
20 50 140 0 0;...
21 50 130 0 0;...
22 50 120 0 0;...
23 50 110 0 0;...
24 50 100 0 0;...
25 50 90 0 0;...
26 50 80 0 0;...
27 50 70 0 0;...
28 50 60 0 0;...
29 50 50 0 0;...
30 50 40 0 0;...
31 50 30 0 0;...
32 50 20 0 0;...
33 50 10 0 0;...
34 50 0 1 1];

%Definition of Elements

for i = 1:33
    Element_cat1(i,:) = [i , i+1];
```



```

end

for i1 = 1:16
    Element_cat2(i1,:) = [i1,35-i1];
end

Element_cat3 = [1 33;...
                33 3;...
                3 31;...
                31 5;...
                5 29;...
                29 7;...
                7 27;...
                27 9;...
                9 25;...
                25 11;...
                11 23;...
                23 13;...
                13 21;...
                21 15;...
                15 19;...
                19 17];

%Element Matrix
Elements = [Element_cat1;Element_cat2;Element_cat3];

%% Calculations

% calculation of Stiffness Matrix
k = Local_K_Elem(Nodal_pos,Elements,Area,E);
N_node = size(Nodal_pos,1);
N_element = size(Elements,1);
K_sum = zeros(2*N_node);
I_dn = zeros(N_element,4);
for i=1:N_element
    I_dn(i,:) = [(Elements(i,1)*2-1) Elements(i,1)*2 (Elements(i,2)*2-1) Elements(i,2)*2] ;
    K_sum(I_dn(i,:),I_dn(i,:)) = K_sum(I_dn(i,:),I_dn(i,:)) + k{i};
end

%Defining a global loading vector based on load P
Load_sum = zeros(2*N_node,1);
for i=1:size(Nodal_forces,1)
    Load_sum(Nodal_forces(i,1)*2-1)=Nodal_forces(i,2)*cos(Nodal_forces(i,3)*pi/180);
    Load_sum(Nodal_forces(i,1)*2)= Nodal_forces(i,2)*sin(Nodal_forces(i,3)*pi/180);
end
N_base =sum(~Nodal_pos,1);
IDf = zeros(N_base(4)+N_base(5),1);
for i = 1:N_node
    if Nodal_pos(i,4)==0
        for q = 1: N_base(4)+N_base(5)
            if IDf(q)==0
                IDf(q)=Nodal_pos(i,1)*2-1;
                break
            end
        end
    end
    if Nodal_pos(i,5)==0
        for y = 1:N_base(4)+N_base(5)
            if IDf(y)==0
                IDf(y)=Nodal_pos(i,1)*2;
                break
            end
        end
    end
end
k_displacement = K_sum(IDf,IDf);
p_disp = Load_sum(IDf);
u_disp = linsolve(k_displacement,p_disp);
u_sum = zeros(2*(N_node),1);
u_sum(IDf) = u_disp;

% Reactions

```

```

Reac_sum = K_sum*u_sum-Load_sum;
XY_def = [Nodal_pos(:,2) Nodal_pos(:,3)];
for i=1:2*N_node
    if u_sum(i)~=0 && mod(i,2)==1
        XY_def((i/2)+.5,1) = Nodal_pos((i/2)+.5,2)+Scaling*u_sum(i);
    end
    if u_sum(i)~=0 && mod(i,2)==0
        XY_def((i/2),2) = Nodal_pos((i/2),3)+Scaling*u_sum(i);
    end
end
% Applied Force
[fe,stress] = LOCALforce(Nodal_pos,Elements,Area,E,u_sum);

%% Plots

for i=1:N_element
    Af = max(Area);
    plot([Nodal_pos(Elements(i,1),2) Nodal_pos(Elements(i,2),2)], [Nodal_pos(Elements(i,1),3)
    Nodal_pos(Elements(i,2),3)], 'k-', 'Linewidth',1)
    hold on
    if fe(i)>0
        plot([XY_def(Elements(i,1),1) XY_def(Elements(i,2),1)], [XY_def(Elements(i,1),2)
        XY_def(Elements(i,2),2)], 'b', 'Linewidth',2)
    else
        plot([XY_def(Elements(i,1),1) XY_def(Elements(i,2),1)], [XY_def(Elements(i,1),2)
        XY_def(Elements(i,2),2)], 'r', 'Linewidth',2)
    end
    title('Structural Analysis of Truss')
    xlabel('X Position')
    ylabel('Y Position')
    axis([-30 90 0 180])
    hold on
    grid on

    text((Nodal_pos(Elements(i,1),2)+Nodal_pos(Elements(i,2),2))/2, (Nodal_pos(Elements(i,1),3)+Nodal
    _pos(Elements(i,2),3))/2, ['E', num2str(i)], 'Color','k', 'FontSize',9);
end
for i=1:N_node
    if Nodal_pos(i,4)==0 && Nodal_pos(i,5)==0
        plot(Nodal_pos(i,2),Nodal_pos(i,3), 'o', 'Color',[.50 .50 .60], 'Linewidth',6)
    else
        plot(Nodal_pos(i,2),Nodal_pos(i,3), 'ko', 'Linewidth',6)
    end
end

%% Functions Used

function kK = Local_K_Elem(Nodal_pos,ID,A,E)
for i=1:size(ID,1)
    dx(i) = Nodal_pos(ID(i,2),2)-Nodal_pos(ID(i,1),2);
    dy(i) = Nodal_pos(ID(i,2),3)-Nodal_pos(ID(i,1),3);
    L(i) = (dx(i)^2+dy(i)^2)^.5;
    c(i) = dx(i)/L(i);
    s(i) = dy(i)/L(i);
    h = [-c(i);-s(i);c(i);s(i)];
    kK{i} = (A(i)*E(i)/L(i))*h*h';
end
end
function [fe,stress_OUT] = LOCALforce(N_XY,ID,A,E,u_sum) %Force
for i=1:size(ID,1)
    dXx(i) = N_XY(ID(i,2),2)-N_XY(ID(i,1),2);
    dYy(i) = N_XY(ID(i,2),3)-N_XY(ID(i,1),3);
    L(i) = (dXx(i)^2+dYy(i)^2)^.5;
    c(i) = dXx(i)/L(i);
    s(i) = dYy(i)/L(i);
    h = [-c(i);-s(i);c(i);s(i)];
    fe(i) = (A(i)*E(i)/L(i))*h*u_sum([ID(i,1)*2-1 ID(i,1)*2 ID(i,2)*2-1 ID(i,2)*2]);
    stress_OUT(i) = fe(i)/A(i); %stress values
end
end

```

MATLAB Code used for part 2:

```
% This is a project submission for MEC 539 - Introduction to Finite element
% Methods. This code does a Stress analysis on a 2D Plate.
% Author: Sravana Sumanth Koneru
% ID : 112526331
% Stony Brook University

global nnd nel nne nodof eldof n
global geom connec dee nf Nodal_loads
%
format short e
E = 210000; % Elastic modulus in MPa
nu = 0.3; % Poisson's ratio
thick = 1; % Beam thickness in mm
%
% Nodes coordinates x and y
%
geom = [0 0;...
        10 0;...
        20 0;...
        30 0;...
        40 0;...
        50 0;...
        60 0;...
        70 0;...
        80 0;...
        80 10;...
        70 10;...
        60 10;...
        50 10;...
        40 10;...
        30 10;...
        20 10;...
        10 10;...
        0 10;...
        0 20;...
        10 20;...
        20 20;...
        30 20;...
        50 20;...
        60 20;...
        70 20;...
        80 20;...
        80 30;...
        70 30;...
        60 30;...
        50 30;...
        40 30;...
        30 30;...
        20 30;...
        10 30;...
        0 30;...
        0 40;...
        10 40;...
        20 40;...
        30 40;...
        40 40;...
        50 40;...
        60 40;...
        70 40;...]
```

```

        80 40];
geom = geom*10;
% Element connectivity
%
conne = [1 2 18;...
        2 17 18;...
        2 16 17;...
        2 3 16;...
        3 4 16;...
        4 15 16;...
        4 14 15;...
        4 5 14;...
        5 6 14;...
        6 13 14;...
        6 12 13;...
        6 7 12;...
        7 8 12;...
        8 11 12;...
        8 10 11;...
        8 9 10;...

        10 26 25;...
        11 10 25;...
        12 11 25;...
        12 25 24;...
        12 24 23;...
        13 12 23;...
        14 13 23;...
        14 22 15;...
        16 15 22;...
        16 22 21;...
        16 21 20;...
        16 20 17;...
        18 17 20;...
        18 20 19;...

        19 20 35;...
        20 34 35;...
        20 33 34;...
        20 21 33;...
        21 22 33;...
        22 32 33;...
        22 31 32;...
        23 30 31;...
        23 29 30;...
        23 24 29;...
        25 29 24;...
        25 28 29;...
        25 27 28;...
        25 26 27;...

        27 44 43;...
        27 43 28;...
        29 28 43;...
        29 43 42;...
        29 42 41;...
        30 29 41;...
        31 30 41;...
        31 41 40;...
        31 40 39;...
        31 39 32;...
        33 32 39;...
        33 39 38;...
```

```

33 38 37;...
34 33 37;...
35 34 37;...
35 37 36];

[nnd, nodof] = size(geom);
[nel, nne] = size(connec);
eldof = nne*nodof;
% Material

dee = formdsig(E,vu);
%
%FIXED NODES*****
    nf = ones(nnd, nodof); % Initialize the matrix nf to 1
    nf(1,1) = 0; nf(1,2) = 0; % Prescribed nodal freedom of node 1
    nf(18,1) = 0; nf(18,2) = 0;
    nf(19,1) = 0; nf(19,2) = 0;
    nf(35,1) = 0; nf(35,2) = 0;
    nf(36,1) = 0; nf(36,2) = 0; % Prescribed nodal freedom of node 21
    % Counting of the free degrees of freedom

n=0;
for i=1:nnd
for j=1:nodof
if nf(i,j) ~= 0
n=n+1;
nf(i,j)=n;
end
end
end

% loading

Nodal_loads= zeros(nnd, 2);
%
Nodal_loads(44,1) = 1000.; Nodal_loads(44,2) = 0; % Node 2
Nodal_loads(27,1) = 1000.; Nodal_loads(27,2) = 0;
Nodal_loads(26,1) = 1000.; Nodal_loads(26,2) = 0;
Nodal_loads(10,1) = 1000.; Nodal_loads(10,2) = 0;
Nodal_loads(9,1) = 1000.; Nodal_loads(9,2) = 0;
%
%%%%%%%%%%%% End of input %%%%%%%%%%%%%

% The main program CST_PLANE_STRESS.m is listed next:
% This is a project submission for MEC 539 - Introduction to Finite element
% Methods. This code does a Stress analysis on a 2D Plate.
% Author: Sravana Sumanth Koneru
% ID : 112526331
% Stony Brook University
% THIS PROGRAM USES AN 3-NODE LINEAR TRIANGULAR ELEMENT FOR THE
% LINEAR ELASTIC STATIC ANALYSIS OF A TWO DIMENSIONAL PROBLEM
%
clear
clc
global nnd nel nne nodof eldof n
global geom connec dee nf Nodal_loads
%
format long g
fid =fopen('CST_COARSE_MESH_RESULTS.txt','w');

CST_COARSE_MESH_DATA;

% Assemble the global force vector
%
fg=zeros(n,1);

```

```

for i=1: nnd
if nf(i,1) ~= 0
fg(nf(i,1))= Nodal_loads(i,1);
end
if nf(i,2) ~= 0
fg(nf(i,2))= Nodal_loads(i,2);
end
end

% Assembly of the global stiffness matrix
% initialize the global stiffness matrix to zero
kk = zeros(n, n);

for i=1:nel
[bee,g,A] = elem_T3(i); % Form strain matrix, and steering vector
ke=thick*A*bee'*dee*bee; % Compute stiffness matrix
kk=form_kk(kk,ke, g); % assemble global stiffness matrix
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% End of assembly %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
delta = kk\fg ; % solve for unknown displacements
%
node_disp=zeros(nnd,2);
%
for i=1: nnd %
if nf(i,1) == 0 %
x_disp =0.; %
else
x_disp = delta(nf(i,1)); %
end
%
if nf(i,2) == 0 %
y_disp = 0.; %
else
y_disp = delta(nf(i,2)); %
end
node_disp(i,:) =[x_disp y_disp];
end
%
% Retrieve the x_coord and y_disp of the nodes located on the neutral axis
%
k = 0;
for i=1:nnd;
if geom(i,2)== 0.
k=k+1;
x_coord(k) = geom(i,1);
vertical_disp(k)=node_disp(i,2);
end
end

for i=1:nel
[bee,g,A] = elem_T3(i); % Form strain matrix, and steering vector
eld=zeros(eldof,1); % Initialize element displacement to zero
for m=1:eldof
if g(m)==0 eld(m)=0.;
else %
eld(m)=delta(g(m)); % Retrieve element displacement
end
end
%
eps=bee*eld; % Compute strains
EPS(i,:)=eps ; % Store strains for all elements
sigma=dee*eps; % Compute stresses
SIGMA(i,:)=sigma ; % Store strains for all elements
end

% Print results to file
print_CST_results;

% Plot the stresses in the x_direction
x_stress = SIGMA(:,1);

```

```
cmin = min(x_stress);
cmax = max(x_stress);
caxis([cmin cmax])

patch('Faces', connec, 'Vertices', geom, 'FaceVertexCData', x_stress, ...
'Facecolor', 'flat', 'Marker', 'o');
colorbar
%
plottools
```

REFERENCES:

1. Daryl L. Logan. 2000. A First Course in the Finite Element Method Using Algor (2nd. ed.). Brooks/Cole Publishing Co., USA.
2. Ferreira, António JM. MATLAB codes for finite element analysis. Springer Netherlands, 2009.