International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

Finite Element Analysis of a 2-D Linear Static Structure (Flat Plate) Using Matlab

Ranbir Mukhya[1], Areeb Siraj[2], Gaurav Arya[3]

[1]Guide, Faculties of Mechanical Engineering Department

[2, 3]Department of Mechanical Engineering, University Institute of Engineering and Technology, C.S.J.M University, Kanpur – 208024

Abstract: ***A finite Element method approach has been developed for structural analysis of various industrial products. It helps to analyze and solve the problem of structural displacement due to various internal and external forces. And it also helps in the calculation stresses and strains. This method can be applied to typical MEMS structures such as beam, plate and other complex structures. The theoretical analysis of the finite element method is well established in the case of the triangular or tetrahedral meshes. In this case we use 4-node rectangular mesh, for the analysis of a flat plate of uniform thickness with the help of Matlab. Our theory supported by the some numerical experiments, which are taken from various engineering application, ranging from elasticity and in other forces and stress analysis.***

1. Introduction

Finite Element Analysis (FEA) is a numerical method for solving problems of engineering and mathematical physics. Typical problem areas of interest include structural analysis, heat transfer analysis, fluid flow analysis etc. The finite element method formulation of the problem results in a system of algebraic equations. The method yields approximate values of the unknowns at discrete number of points over the domain. To solve the problem, it subdivides a large problem into smaller, simpler parts that are called finite elements. The simple equations that model these finite elements are then assembled into a larger system of equations that models the entire problem.

In the first step above, the element equations are simple equations that locally approximate the original complex equations to be studied, where the original equations are often partial differential equations (PDE). To explain the approximation in this process, FEM is commonly introduced as a special case of Galerkin method. The process, in mathematical language, is to construct an integral of the inner product of the residual and the weight functions and set the integral to zero. In simple terms, it is a procedure that minimizes the error of approximation by fitting trial functions into the PDE. The residual is the error caused by the trial functions, and the weight functions are polynomial approximation functions that project the residual.

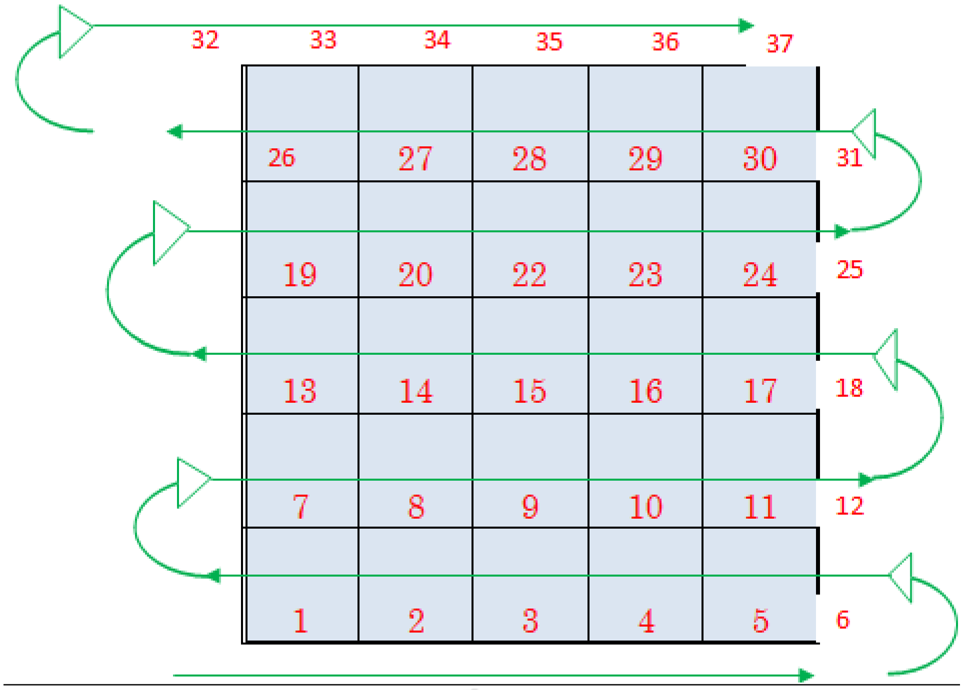
FEA as applied in engineering is a computational tool for performing engineering analysis. It includes the use of mesh generation techniques for dividing a complex problem into small elements, as well as the use of software program (in our case MATLAB) coded with FEM algorithm.

FEA is a good choice for analyzing problems over complicated domains like cars and oil pipelines, machinery parts, objects under stress etc. FEA simulations provide a valuable resource as they remove multiple instances of creation and testing of hard prototypes for various high fidelity situations. For instance, in a frontal crash simulation it is possible to increase prediction accuracy in "important" areas like the front of the car and reduce it in its rear thus reducing cost of the simulation.

The purpose of finite element analysis (FEA) is to reduce the number of prototypes and experiments that have to be run when designing, optimizing, or controlling a device or process.

2. Terminologies

1) *Discretization* - It is a method of subdividing a complex structure (structural problem) into a convenient no. of smaller components.
2) *Nodes* –A node is a specific point in the finite element at which the value of the field variable is to be explicitly calculated.
3) *Stiffness Matrix* – The stiffness matrix contains the geometric and material behavior information that indicates the resistance of the element to deformation when subjected to loading. Such deformation may include axial, bending, shear, and torsional effects. For finite elements used in non-structural analyses, such as fluid flow and heat transfer, the term stiffness matrix is also used, since the matrix represents the resistance of the element to change when subjected to external influences.
4) *Boundary Conditions* - Boundary conditions are a set of loading, constraints and contact conditions that define the status of your simulation in one possible design configuration.
5) *Node Numbering:*



Basic steps for solving structural problems using FEA

1. Domain discretization
2. Select element type

a) 4-Noded triangle
b) 8-Noded triangle
c) 4-Noded quadrilateral
d) 8-Noded quadrilateral

3. Derive element equations by variational or energy methods

[K]{U} = {F}

[K] = Stiffness or Property Matrix **(Property)**

{U} = Nodal Displacement Vector **(Behavior)**

{F} = Nodal Force Vector **(Action)**

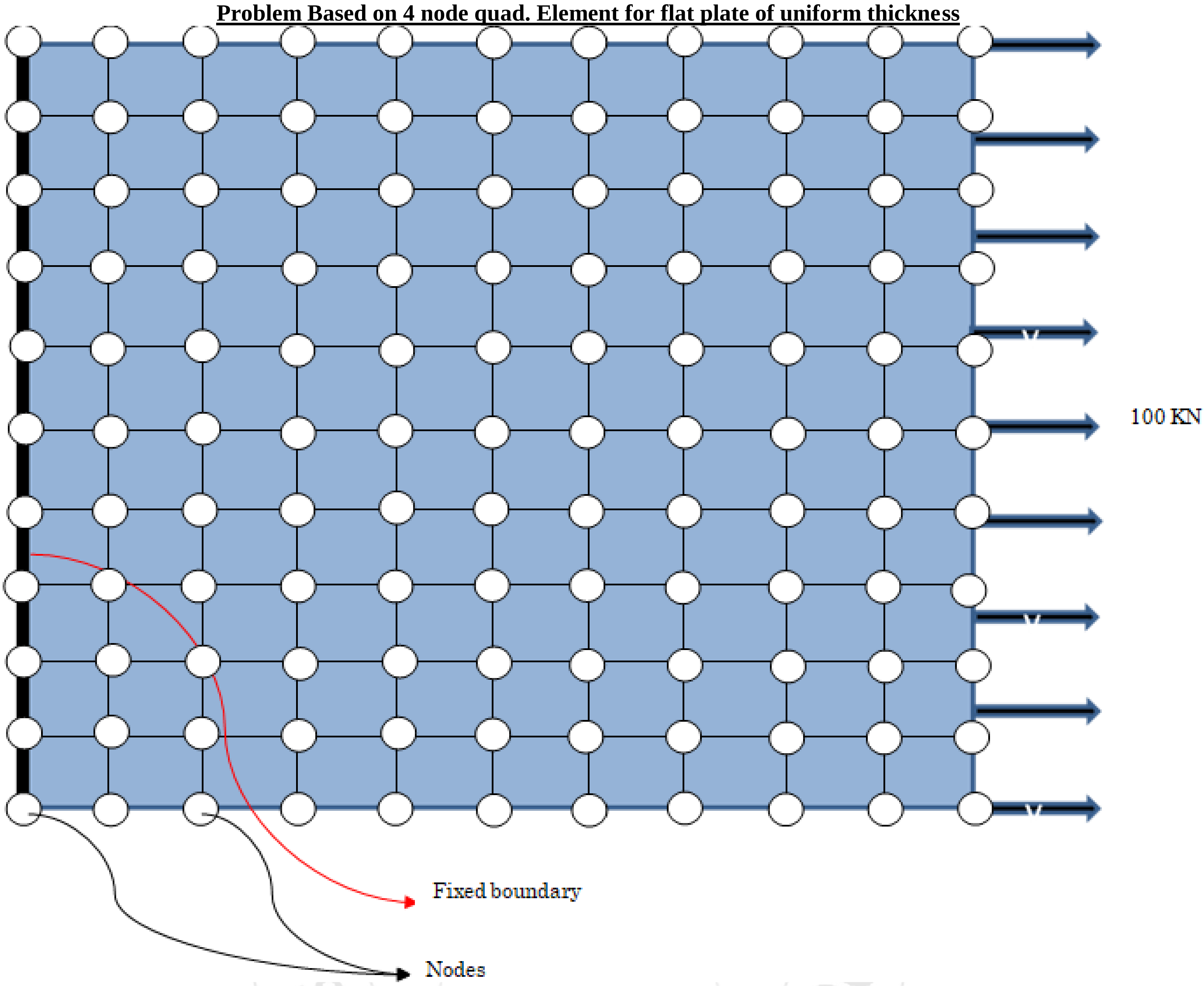
4. Assemble element equations to form global system
5. Apply initial and boundary conditions
6. Solve assembled system of equations for primary unknown variable (in our case displacement)
7. Solve assembled system of equations for secondary unknown variable

(in our case)

a) Deflection
b) Strain
c) Stress

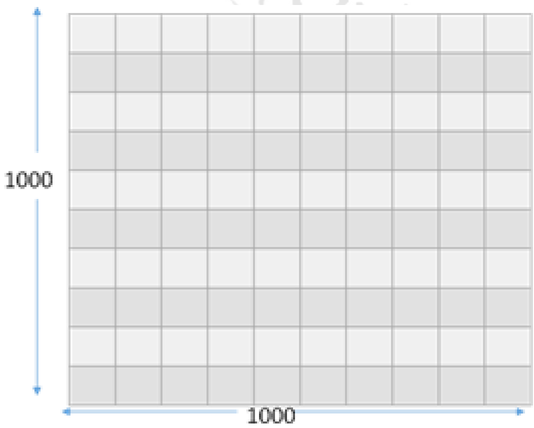
International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2018): 7.426

Problem Based on 4 node quad. Element for flat plate of uniform thickness

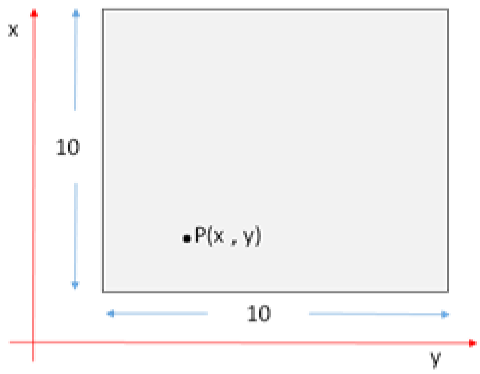


Ques. An aluminum sheet of uniform thickness 10mm fixed at one end, whereas a uniform uni-axial load of 100 KN is applied on the other end. Using FEA do structural analysis and find primary and secondary variables. Take Young's modulus = $2.15*10^9$ and Poisson's ratio = 0.4

Formulation of Element Equations

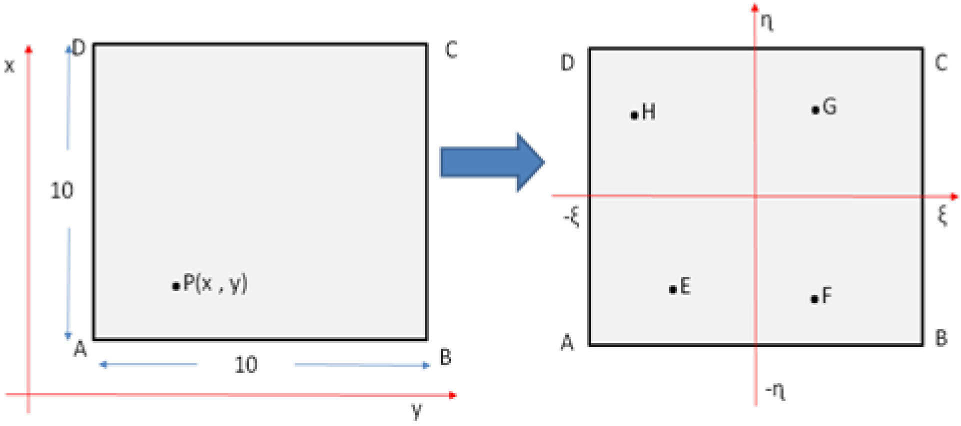


Discretization of a lamina of 1000 X 1000mm into sub element of 10X10mm



Sub element under consideration in x-y plane

(All Dimensions are in mm)



Transformation of x-y co-ordinate system into ξ-η co-ordinate system

Co-ordinates of different nodes

	X	Y	ξ	η
A	0	0	-1	-1
B	10	0	1	-1
C	10	10	1	1
D	0	10	-1	1

Co-ordinates of any point

$X = N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4$

$y = N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4$

Equations of Shape Function

- $N_1 = \frac{1}{4}[1-\xi][1-\eta]$ → (i)
- $N_1 = \frac{1}{4}[1+\xi][1-\eta]$ → (ii)
- $N_1 = \frac{1}{4}[1+\xi][1+\eta]$ → (iii)
- $N_1 = \frac{1}{4}[1-\xi][1+\eta]$ → (iv)

Displacement of any point

$U = N_1u_1 + N_2u_2 + N_3u_3 + N_4u_4$

$V = N_1v_1 + N_2v_2 + N_3v_3 + N_4v_4$

$[Ke]_{8x8}\,[U]_{8x1} = [F]_{8x1}$ → (a)

[Ke] = Stiffness Matrix

[U] = Displacement Matrix

[F] = Force Matrix

$[Ke] = {}_v\!\int B^TD^{-1}B\, dv$

B = Strain/Displacement Matrix

D = Material Matrix

dv = t dxdy

dxdy = |det[J]| dξdη

dv = t |det[J]| dξdη → (b)

J = Jacobian Matrix

From equation (a) and (b) , we get

$[Ke] = {}_v\!\int B^TD^{-1}B\, t\, |det[J]|\, d\xi d\eta$

(analytical form stiffness matrix)

$[Ke] = W_1W_2\, B^TD^{-1}B\, t\, |det[J]|\, d\xi d\eta$ → (c)

(numerical for stiffness matrix)

W_1W_2 = Weighted Functions

$$[J] = \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{pmatrix}_{2x2} \rightarrow (d)$$

Since,

$X = N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4$

$$\frac{\partial X}{\partial \xi} = \frac{\partial N_1X_1}{\partial \xi} + \frac{\partial N_2X_2}{\partial \xi} + \frac{\partial N_3X_3}{\partial \xi} + \frac{\partial N_4X_4}{\partial \xi} \rightarrow (e)$$

$$\frac{\partial X}{\partial \eta} = \frac{\partial N_1X_1}{\partial \eta} + \frac{\partial N_2X_2}{\partial \eta} + \frac{\partial N_3X_3}{\partial \eta} + \frac{\partial N_4X_4}{\partial \eta} \rightarrow (f)$$

Similarly,

$Y = N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4$

$$\frac{\partial Y}{\partial \xi} = \frac{\partial N_1y_1}{\partial \xi} + \frac{\partial N_2y_2}{\partial \xi} + \frac{\partial N_3y_3}{\partial \xi} + \frac{\partial N_4y_4}{\partial \xi} \rightarrow (g)$$

$$\frac{\partial X}{\partial \eta} = \frac{\partial N_1y_1}{\partial \eta} + \frac{\partial N_2y_2}{\partial \eta} + \frac{\partial N_3y_3}{\partial \eta} + \frac{\partial N_4y_4}{\partial \eta} \rightarrow (h)$$

From equations of shape function (i),(ii),(iii),(iv)

$\frac{\partial N_1}{\partial \xi} = \frac{-1}{4}[1-\eta]$ $\quad$ $\frac{\partial N_1}{\partial \eta} = \frac{-1}{4}[1-\xi]$

$\frac{\partial N_1}{\partial \xi} = \frac{1}{4}[1-\eta]$ $\quad$ $\frac{\partial N_1}{\partial \eta} = \frac{-1}{4}[1+\xi]$

→ (i)

$\frac{\partial N_1}{\partial \xi} = \frac{1}{4}[1+\eta]$ $\quad$ $\frac{\partial N_1}{\partial \eta} = \frac{1}{4}[1+\xi]$

$\frac{\partial N_1}{\partial \xi} = \frac{-1}{4}[1+\eta]$ $\quad$ $\frac{\partial N_1}{\partial \eta} = \frac{1}{4}[1-\xi]$

From equations (e), (f), (g), (h) and (i) , we get

$$\frac{\partial X}{\partial \xi} = \frac{1}{4}[-X_1(1-\eta) + X_2(1-\eta) + X_3(1+\eta) - X_4(1+\eta)] \longrightarrow (j)$$

$$\frac{\partial X}{\partial \eta} = \frac{1}{4}[-X_1(1-\xi) - X_2(1+\xi) + X_3(1+\xi) + X_4(1-\xi)] \longrightarrow (k)$$

$$\frac{\partial Y}{\partial \xi} = \frac{1}{4}[-y_1(1-\eta) + y_2(1-\eta) + y_3(1+\eta) - y_4(1+\eta)] \longrightarrow (l)$$

$$\frac{\partial Y}{\partial \xi} = \frac{1}{4}[-y_1(1-\xi) - y_2(1+\xi) + y_3(1+\xi) + y_4(1-\xi)] \longrightarrow (m)$$

Gauss Points

$E = \frac{-1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}$ $F = \frac{1}{\sqrt{3}}, \frac{-1}{\sqrt{3}}$ $G = \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}$ $H = \frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}$

MATLAB Code

```
n = [2 2];
L = [.1 .1];
nu = .4;
E = 2.15e9;             %YOUNG'S MODULUS
nx = n(1)+1;             %No. of nodes by X
ny = n(2)+1;             %No. of nodes by y
nNodes = nx*ny;           %total no. of nodes
ne = n(1)*n(2);          %total no. of elements
bnd = [];              %boundary condition
dx = L(1)/n(1);          %space step by X
dy = L(2)/n(2);          %space step by y

%Gaussian points
Zeta = [-(1/3)^0.5; -(1/3)^0.5; (1/3)^0.5; (1/3)^0.5 ];
%local zeta
Eta = [-(1/3)^0.5; (1/3)^0.5; (1/3)^0.5; -(1/3)^0.5 ];    %local
eta
W = 1;
U = zeros(2*nNodes,1);            %displacement matrix
K = zeros (2*nNodes,2*nNodes);     %stiffness matrix
res.F = zeros(2*nNodes,1);          %force vector
eID = 1;

%Loop over all elements
for j=1:n(2)
    for i=1:n(1)
        % Node IDs
        e(eID).n(1) = (j-1)*nx+i;
        e(eID).n(2) = j*nx+i;
        e(eID).n(3) = j*nx+i+1;
        e(eID).n(4) = (j-1)*nx+i+1;
        % Global node position
        e(eID).x(1,1) = (i-1)*dx;
        e(eID).y(1,1) = (j-1)*dy;
        e(eID).x(2,1) = (i-1)*dx;
        e(eID).y(2,1) = j*dy;
        e(eID).x(3,1) = i*dx;
        e(eID).y(3,1) = j*dy;
        e(eID).x(4,1) = i*dx;
        e(eID).y(4,1) = (j-1)*dy;
        e(eID).U = zeros(8,1);           % Modal displacement
        e(eID).Umagnitude = zeros(4,1);   % Magnitude of
displcement
        e(eID).E = E;                  % Young's modulus
        e(eID).nu = nu;                 % Nu
        e(eID).D = zeros(3,3);
        % Calculate 'D' matrix
        e(eID).D(1,1) = e(eID).E/(1-e(eID).nu^2);
        e(eID).D(1,2) = e(eID).E/(1-e(eID).nu^2)*e(eID).nu;
        e(eID).D(2,1) = e(eID).E/(1-e(eID).nu^2)*e(eID).nu;
        e(eID).D(2,2) = e(eID).E/(1-e(eID).nu^2);
        e(eID).D(3,3) = e(eID).E/(1-
e(eID).nu^2)*0.5*e(eID).nu;
        eID = eID+1;
    end
end


for eID= 1:ne
    Klocal = zeros(8,8);            % Local stiffness matrix
    for gp =1:4                 % Loop over all gaussian
points
        zeta = Zeta(gp,1);
        eta = Eta(gp,1);
        dNdzeta(1) = -0.25*(1-eta);
        dNdzeta(2) = -0.25*(1+eta);
        dNdzeta(3) = 0.25*(1+eta);
        dNdzeta(4) = 0.25*(1-eta);
        dNdeta(1) = 0.25*(1-zeta);
        dNdeta(2) = -0.25*(1-zeta);
        dNdeta(3) = -0.25*(1+zeta);
        dNdeta(4) = 0.25*(1+zeta);
        dNdlocal = [dNdzeta; dNdeta];
        % Jacobian Matrix
        J(1,1) = dNdzeta*e(eID).x;
        J(1,2) = dNdzeta*e(eID).y;
        J(2,1) = dNdeta*e(eID).x;
        J(2,2) = dNdeta*e(eID).y;
        dNdglobal = J\dNdlocal;

        B(1,1)= dNdglobal(1,1);
        B(1,2)= 0;
        B(1,3)= dNdglobal(1,2);
        B(1,4)= 0;
        B(1,5)= dNdglobal(1,3);
        B(1,6)= 0;
        B(1,7)= dNdglobal(1,4);
        B(1,8)= 0;
        B(2,1)= 0;
        B(2,2)= dNdglobal(2,1);
        B(2,3)= 0;
        B(2,4)= dNdglobal(2,2);
        B(2,5)= 0;
```

```
        B(2,6)= dNdglobal(2,3);
        B(2,7)= 0;
        B(2,8)= dNdglobal(2,4);
        B(3,1)= dNdglobal(2,1);
        B(3,2)= dNdglobal(1,1);
        B(3,3)= dNdglobal(2,2);
        B(3,4)= dNdglobal(1,2);
        B(3,5)= dNdglobal(2,3);
        B(3,6)= dNdglobal(1,3);
        B(3,7)= dNdglobal(2,4);
        B(3,8)= dNdglobal(1,4);
        Klocal = Klocal+W*det(J)*B.'*e(eID).D*B*1e-8;    %
Final local stiffness matrix
    end
    % Global stiffness matrix
    for i=1:4
    for j=1:4
        K(e(eID).n(i)*2-1:e(eID).n(i)*2,e(eID).n(j)*2-
1:e(eID).n(j)*2)=...
            K(e(eID).n(i)*2-1:e(eID).n(i)*2,e(eID).n(j)*2-
1:e(eID).n(j)*2)...
            + Klocal(i*2-1:i*2,j*2-1:j*2);
    end
    end
end
F = zeros(2*nNodes,1);
U = zeros(2*nNodes,1);
Del = zeros(2*nNodes,1);
Strain = zeros(2*nNodes,1);
Stress = zeros(2*nNodes,1);
for i = 1:6:2*nNodes
    F(i+4,1) = 100;
    F = F+F(i,1);
end
RCOND = 1.908653e-19;
U = (K\F)*RCOND;
%{
Due to the warning given below U is multiplied by RCOND
= 1.908653e-19

Warning: Matrix is close to singular or badly scaled
Results may be inaccurate. RCOND =  1.908653e-19
%}
Del = diff(U);
Strain = Del/L(1);
Sress = Strain*E;
```

Warning: Matrix is close to singular or badly scaled. Results may be inaccurate. RCOND = 1.908653e-19.

Published with MATLAB® R2014a

3. Results

Displacement Matrix:

Variables - U | U | Del | Strain

18x1 double

	1	2
1	0.0405	
2	-0.0526	
3	0.0405	
4	-0.0246	
5	0.0405	
6	0.0035	
7	0.0124	
8	-0.0526	
9	0.0124	
10	-0.0246	
11	0.0124	
12	0.0035	
13	-0.0156	
14	-0.0526	
15	-0.0156	
16	-0.0246	
17	-0.0156	
18	0.0035	
19		
20		

Deflection Matrix:

Variables - Del | Del | U | Strain

17x1 double

	1	2
1	-0.0931	
2	0.0931	
3	-0.0651	
4	0.0651	
5	-0.0370	
6	0.0089	
7	-0.0651	
8	0.0651	
9	-0.0370	
10	0.0370	
11	-0.0089	
12	-0.0191	
13	-0.0370	
14	0.0370	
15	-0.0089	
16	0.0089	
17	0.0191	
18		
19		

Strain Matrix:

Variables - Strain | Strain | Del | U

17x1 double

	1	2
1	-0.9312	
2	0.9312	
3	-0.6506	
4	0.6506	
5	-0.3700	
6	0.0895	
7	-0.6506	
8	0.6506	
9	-0.3700	
10	0.3700	
11	-0.0895	
12	-0.1911	
13	-0.3700	
14	0.3700	
15	-0.0895	
16	0.0895	
17	0.1911	
18		
19		

Stress Matrix:

Variables - Stress | Strain | Del | U

17x1 double

	1	2
1	-2.0020e+09	
2	2.0020e+09	
3	-1.3988e+09	
4	1.3988e+09	
5	-7.9560e+08	
6	1.9239e+08	
7	-1.3988e+09	
8	1.3988e+09	
9	-7.9560e+08	
10	7.9560e+08	
11	-1.9239e+08	
12	-4.1081e+08	
13	-7.9560e+08	
14	7.9560e+08	
15	-1.9239e+08	
16	1.9239e+08	
17	4.1081e+08	
18		
19		

4. Conclusion

The conclusion that can be drawn from the above results is;

- Finite element analysis is an essential tool for helping us in determining the cause of the problems and recommending the solutions
- FEM analysis of structural failure should be adopted as standard tool in failure analysis
- With a trained engineer, FEM is quick and easy to deploy
- With the exponential increase in computing power FEM is economical to carry out.

References

Websites
[1] https://nptel.ac.in/reviewed_pdfs/105106051/lec36.pdf
[2] https://www.researchgate.net/.../268188312_A_Four-node_quadrilateral_element_for_fi...
[3] https://www.emeraldinsight.com/doi/full/10.1108/EC-04-2015-0102?fullSc=1
[4] web.mit.edu/.../The_finite_element_method_with_overlapping_elements_a_new_para...

Books
[5] Finite element method by J.N Reddy
[6] Computer aided design by R.K Srivastava
[7] Finite element analysis by P. Seshu

Programming books for Matlab
[8] Understanding MATLAB by S.N Alam and S.S Alam
[9] Introduction to MATLAB by Mathworks