B. TECH. PROJECT REPORT

On

A Finite Element Study on Bending of Isotropic Plates

BY

Shubham Kumar Chayla
(160004035)

DISCIPLINE OF CIVIL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE
November 2019

A Finite Element Study on Bending of Isotropic Plates

A PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of the degrees

of
BACHELOR OF TECHNOLOGY
in

CIVIL ENGINEERING

Submitted by:
Shubham Kumar Chayla
(1600040035)

Guided by:
Dr. Kaustav Bakshi

INDIAN INSTITUTE OF TECHNOLOGY INDORE

November 2019

CANDIDATE'S DECLARATION

I hereby declare that the project entitled "**A Finite Element Study on Bending of Isotropic Plates"** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in 'Civil Engineering' completed under the supervision of **Dr. Kaustav Bakshi, Assistant Professor, Civil Engineering, IIT Indore** is an authentic work.

Further, I declare that I have not submitted this work for the award of any other degree elsewhere.

Shubham Kumar Chayla

CERTIFICATE by BTP Guide

It is certified that the above statement made by the student is correct to the best of my knowledge.

Dr, kaustav Bakshi

Assistant Professor

Department of Civil Engineering

IIT Indore

<u>Preface</u>

This report on "A Finite Element Study on Bending of Isotropic Plates" is prepared under the guidance of Dr. Kaustav Bakshi.

Through this report I have tried to give a detailed analysis of deflections and stress of an 8-noded element plate. I have tried to the best of our abilities and knowledge to explain the content in a lucid manner. We have also added MATLAB code, tables and figures to make it more illustrative.

Shubham Kumar Chayla
B.Tech. IV Year
Discipline of Civil Engineering
IIT Indore

Acknowledgements

I wish to thank Dr. Kaustav Bakshi for his kind support and valuable guidance and giving me an opportunity to work for the B. Tech project under his supervision. I owe profound gratitude to him who took a keen interest in our project and I am extremely fortunate to have his guidance.

I respect and thank Mrs. Neelima Satyam and faculty members of the civil engineering department for their constant encouragement and more over for their timely support and guidance till the completion of the project. I would like to acknowledge the library of IIT Indore for their support for providing all the necessary information for developing a good system.

It is their help and support, due to which we became able to complete the design and technical report.

I am thankful to get constant encouragement, support and guidance from B. Tech committee. Without their support this report would not have been possible.

Shubham Kumar Chayla
B.Tech. IV Year
Discipline of Civil Engineering
IIT Indore

Abstract

This research work aims to study the static behavior of isotropic plates for varying boundary conditions and cut-outs. The finite element method is adopted for isotropic plate by considering an eight noded quadrilateral element. The isoparametric finite element code is developed in MATLAB and the results obtained are showing good agreement when compared with the results available in literature. Once the present study confirms the accuracy of the proposed code it further concentrates on bending behavior of plates with complicated support conditions made of clamped, simply supported and free boundary conditions. The deformations and stress resultants are studied for solid plates and plates with cut-outs. The cut-outs are applied in plates for better ventilation, conduits for passage of air-conditioning ducts, electrical and telecommunication cables. The study furnishes the results to the practicing civil engineers so that an optimum plate configuration with proper location and size of cut-out can be selected for a given quantity of material consumption.

Table of Contents

List of Figures

27. Bending moment acting on face normal to y-direction Simply Supported for x=0, x=l and Clamped for y=0, y=b
28. Out of plane Shear Force along x direction Simply Supported for x=0, x=l and Clamped for y=0, y=b
29. Out of plane Shear Force along x direction Simply Supported for x=0, x=l and Clamped for y=0, y=b

List of Tables

Introduction

Finite element Method (FEM) is a powerful computational technique used for solving engineering problems that are subjected to general boundary conditions because it can reduce a problem with infinite no of degrees to a finite degree problem with the help of discretization which is done according to the problem. The system is discretized into a finite number of parts known as elements. For a beam or rod the discretization procedure divides the whole rod/beam into small linear elements thus helping to apply the basic governing equations on each and every element and since all the elements being the part of the complete rod/beam all are related with the help of global stiffness matrices and the boundary conditions are applied in order to solve the whole matrix of equations and get the values of the unknown values at each node. In the case with the 2 dimensional plates here the plate is discretized into rectangular elements and the boundary conditions are analysed to get the unknown values at the discretized nodes but the disadvantage with this is it is only a numerical method it can only come close to the analytical value but cannot be equal to it on the other hand the great advantage which comes with FEM is it can easily solve the complex governing equations which are very difficult to solve analytically and takes very long time in getting solved ,thus saving from huge losses to modern industries. All these favourable advantages come at the low cost of little inaccuracy since it's a numerical method.

Defu and sheikh (2005) have presented the mathematical approach for large deflection of rectangular plates. Their analysis, based on the two fourth order and second-degree partial differential von Karman equations, found lateral deflection to applied load. This solution can be used to direct practical analysis of plates with different boundary conditions. Bakker et al. (2008) have studied the approximate analysis method for large deflection of rectangular thin plate with simply supported boundary condition under the action of transverse loads. This approach gives the shape of initial and total deflection of plates. Jain (2009) recently analysed the effect of D/A ratio (where D is hole diameter and A is plate width) upon stress concentration factor and deflection in isotropic and orthotropic plates under transverse static loading with central circular hole under transverse static loading. He considered three types of elements to solve square plate problems with various boundary conditions and loadings.

The present work deals with the analysis of an isotropic rectangular element being considered as a plane stress condition. This paper deals with FEA of isotropic rectangular plates under various boundary conditions. The aim of the present work is to study the bending stiffness of uniformly loaded plates for varying aspect ratio and boundary conditions, to predict the deflections and stress resultants of plates, to verify varying maximum deflection for different position of rectangular hole and load in plate. Throughout the analysis, the element adopted is eight noded quadrilateral elements.

Formulation

For the present development, we assume an eight noded isotropic element with eight nodes with five degree of freedom i.e., *u*, *v*, *w* which denotes displacement along *x*, *y*, *z* axes and *α*, β denotes rotations about *x* and *y* axes at node at origin of the plate element. We assume that the stress conditions are those of two-dimensional plane elasticity; hence, the stress-strain relationship is given by

$$\{\sigma\} = D\{\mathcal{E}\}$$

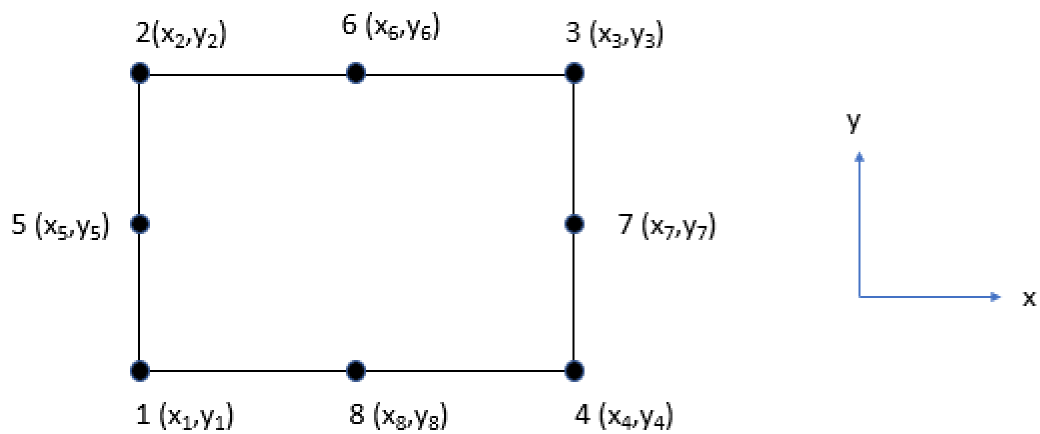

Figure 1

where $$\{\sigma\} = \{N_x \quad N_y \quad N_{xy} \quad M_x \quad M_y \quad M_{xy} \quad Q_x \quad Q_y\}^T$$

$$\{\mathcal{E}\} = \{\mathcal{E}_x \quad \mathcal{E}_y \quad \Upsilon_{xy} \quad \chi_x \quad \chi_y \quad \chi_{xy} \quad \Upsilon_{xz} \quad \Upsilon_{yz}\}^T$$

For an isotropic material, the stress/strain D matrix is

$$D = \begin{bmatrix} E1 & E2 & 0 & 0 & 0 & 0 & 0 & 0 \\ E2 & E1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & G1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & E3 & E4 & 0 & 0 & 0 \\ 0 & 0 & 0 & E4 & E3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & G2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & G3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & G4 \end{bmatrix}$$

where, E1= $\frac{E}{1-\nu^2}$, E2= $\frac{Et\nu}{1-\nu^2}$, E3= $\frac{Et^3\nu}{12(1-\nu^2)}$, E4= $\frac{Et^3\nu}{12(1-\nu^2)}$

G1= tg_{xy} , G2= $\frac{t^3 g_{xy}}{12}$, G3= tg_{xz} , G4= tg_{yz}

$g_{xy} = \frac{E}{2(1+\nu)}$, $g_{xz} = \frac{g_{xy}}{1.2}$, $g_{yz} = \frac{g_{xy}}{1.2}$

with E and ν denotes Young's modulus and Poisson's ratio, respectively.

The element stiffness relationship is given by

$$[k]\{d\} = \{f\}$$

where the stiffness matrix is given by

$$[k] = \int_{-1}^{1}\int_{-1}^{1}[B]^{T}[D][B]|J|drds$$

And the strain-displacement matrix B is given by

$$[B]_{(8\times40)} = [\ [B1]\ |\ [B2]\ |\ [B3]\ |\ [B4]\ |\ [B5]\ |\ [B6]\ |\ [B7]\ |\ [B8]]$$

Where

$$[Bi]_{(8\times5)} = \begin{bmatrix} \partial Ni/\partial x & 0 & 0 & 0 & 0 \\ 0 & \partial Ni/\partial y & 0 & 0 & 0 \\ \partial Ni/\partial y & \partial Ni/\partial x & 0 & 0 & 0 \\ 0 & 0 & 0 & \partial Ni/\partial x & 0 \\ 0 & 0 & 0 & 0 & \partial Ni/\partial y \\ 0 & 0 & 0 & \partial Ni/\partial y & \partial Ni/\partial x \\ 0 & 0 & \partial Ni/\partial x & N_i & 0 \\ 0 & 0 & \partial Ni/\partial y & 0 & N_i \end{bmatrix}$$

It may be noted that this expression for B assumes that displacements are numbered alternatively, thus, the nodal displacement and force vectors are given by

$$\{d\} = \{u_1\ v_1\ w_1\ \alpha_1\ \beta_1 \ldots\ldots\ldots\ u_2\ v_2\ w_2\ \alpha_2\ \beta_2 \ldots\ldots\ldots\ u_8\ v_8\ w_8\ \alpha_8\ \beta_8\}^{T}$$

And also, the strain can be calculated by the relationship given by

$$\{\mathcal{E}\} = [B]\{d\}$$

The N's are the shape functions in local co-ordinates, which for this eight noded element are given as

$$N_1 = \frac{1}{4}(1-\xi)\,(1-\eta)\,(-\xi-\eta-1)$$

$$N_2 = \frac{1}{4}(1-\xi)\,(1+\eta)\,(-\xi+\eta-1)$$

$$N_3 = \frac{1}{4}(1+\xi)\,(1+\eta)\,(\xi+\eta-1)$$

$$N_4 = \frac{1}{4}(1+\xi)\,(1-\eta)\,(\xi-\eta-1)$$

$$N_5 = \frac{1}{2}(1-\xi)\,(1+\eta)\,(1-\eta)$$

$$N_6 = \frac{1}{2}(1+\xi)\,(1-\xi)\,(1+\eta)$$

$$N_7 = \frac{1}{2}(1+\xi)\,(1+\eta)\,(1-\eta)$$

$$N_8 = \frac{1}{2}(1+\xi)\,(1-\xi)\,(1-\eta)$$

As the element is iso-parametric, the relationship between local and global co-ordinate system is given by

$$x = N_1 x_1 + N_2 x_2 + N_3 x_3 + N_4 x_4 + N_5 x_5 + N_6 x_6 + N_7 x_7 + N_8 x_8$$

$$y = N_1 y_1 + N_2 y_2 + N_3 y_3 + N_4 y_4 + N_5 y_5 + N_6 y_6 + N_7 y_7 + N_8 y_8$$

The numerically integrated element stiffness matrix can be expressed as

$$[k] = \sum_{i=1}^{N} \sum_{j=1}^{N} \omega_{ij}\ (\det J)_{ij}\ [B]^{T}[D][B]_{ij}$$

Where the subscripts i and j index the integrating points.

For the two-point formula considered here, N=2, the weighting coefficients ω_{ij} all equal unity, and the integrating points are located at $\pm 1/\sqrt{3}$ in local co-ordinates (ξ, η). There are four Gauss points located on plate as shown in the figure below.

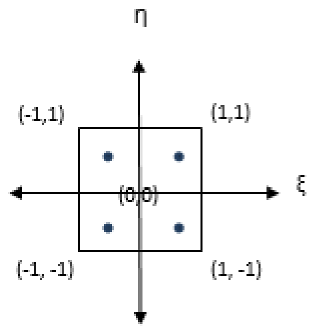

Figure 2

The scalar ($\det J$) is the determinant of the Jacobian matrix, where

$$J = \begin{bmatrix} \partial x/\partial \xi & \partial y/\partial \xi \\ \partial x/\partial \eta & \partial y/\partial \eta \end{bmatrix}$$

and this, together with the matrix $[B]^T[D][B]$ is evaluated at each Gauss point in turn.

2.1 Assembly of Global stiffness matrix

After computation of the elemental stiffness matrices of the members, the next step is the assembly of the global stiffness matrix of the entire plate elements. The global stiffness [K] is calculated by assembly elemental stiffness matrices with the help of direct stiffness method.

In the direct stiffness method, we get reduced stiffness matrix due to the applied boundary to the plate for the applied loading. Similarly assembling elemental force matrix, we get {F} the global force vector. The relationship for calculating deflections is given by

$$[K]\{d\} = \{F\}$$

$$\{d\} = [K]^{-1} [F]$$

2.2 Boundary Conditions

In this present study we have taken a plate of length l, breadth b and different boundary conditions for the further results as given below.

CASE 1 - Plate clamped on all sides

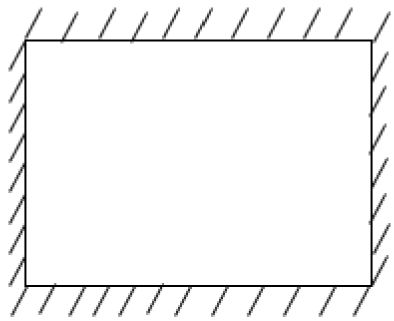

Figure 3

CASE 2 - Plate clamped on x=0, x=l

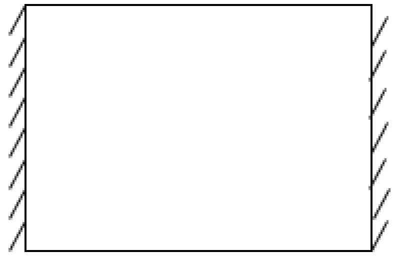

Figure 4

CASE 3 - Plate clamped on y=0, y=b

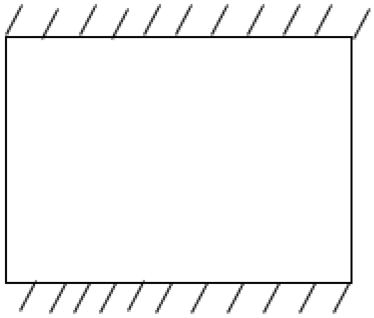

Figure 5

CASE 4 - Plate clamped on x=0, y=0

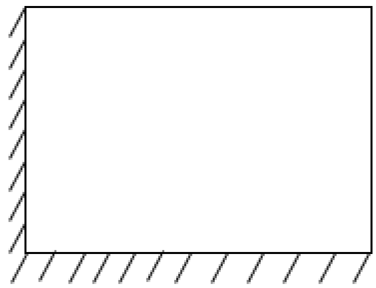

Figure 6

CASE 5 - Simply supported plate on all sides

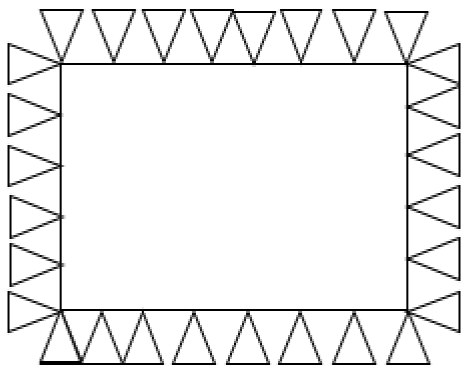

Figure 7

CASE 6 - Plate clamped on x=0, y=b and Simply supported plate on x=l, y=0

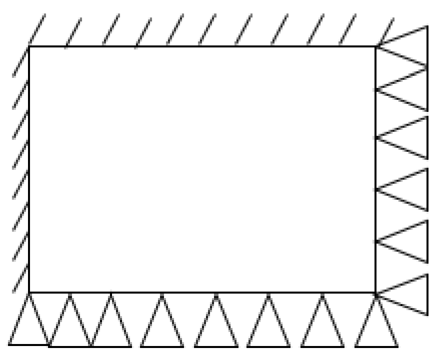

Figure 8

CASE 7 -   Plate clamped on x=l, y=0 and Simply supported plate on x=0, y=b

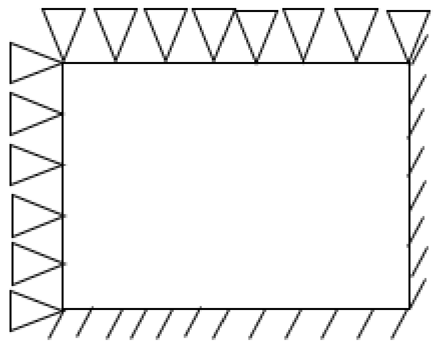

Figure 9

CASE 8 -   Plate clamped on x=0, x=l and Simply supported plate on y=0, y=b

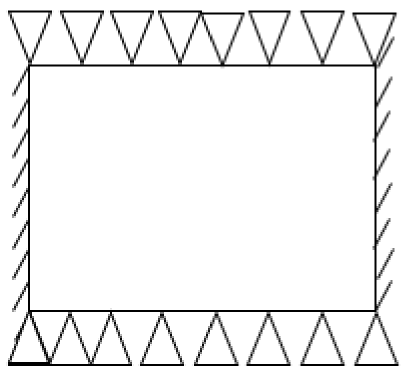

Figure 10

CASE 9 -   Plate clamped on y=0, y=b and Simply supported plate on x=0, x=l

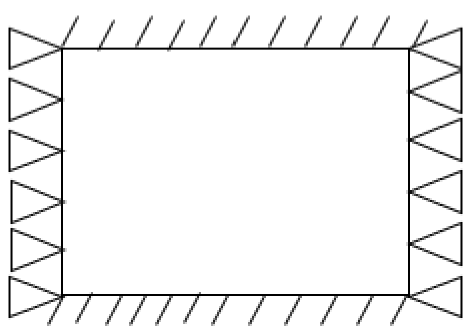

Figure 11

Results

3.1 Maximum displacement in z-direction

For a plate of length l, and breadth b the deflections at each node is calculated and node at which maximum displacement in z direction is plotted below. In the following table the maximum displacement is plotted with respect to aspect ratio of plate. The coordinates below is position at which load of 500 N is applied to the plate. Plate material is steel so Young's modulus is $2x10^5$ N/m^2, Poisson's ratio of 0.3 and plate thickness is 0.001 m, No. of plate elements is taken as 100.

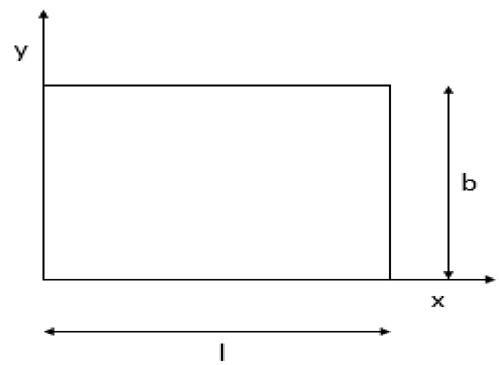

Figure 12

l/b	0.5(0.5/1)	1(1/1)	1.5(1.5/1)	2(2/1)	2.5(2.5/1)
BC1	-0.0496843 (0.25,0.5)	-0.1531200 (0.5,0.5)	-0.1916487 (0.75,0.5)	-0.1961131 (1,0.5)	-0.1953197 (1.25,0.5)
BC2	-0.0498251 (0.25,0.5)	-0.2088344 (0.5,0.5)	-0.5537987 (0.75,0.5)	-1.2132365 (1,0.5)	-2.3132071 (1.25,0.5)
BC3	-0.3041484 (0.25,0.5)	-0.2088344 (0.5,0.5)	-0.1987241 (0.75,0.5)	-0.1966976 (1,0.5)	-0.1953680 (1.25,0.5)
BC4	-0.2919412 (0.25,0.5)	-0.8469741 (0.5,0.5)	-1.0404669 (0.75,0.5)	-1.1647082 (1,0.5)	-1.2521249 (1.25,0.5)
BC5	-0.0496914 (0.25,0.5)	-0.1531568 (0.5,0.5)	-0.1917643 (0.75,0.5)	-0.1962522 (1,0.5)	-0.1955227 (1.25,0.5)
BC6	-0.0496878 (0.25,0.5)	-0.1531201 (0.5,0.5)	-0.1917065 (0.75,0.5)	-0.1961827 (1,0.5)	-0.1954211 (1.25,0.5)
BC7	-0.0496878 (0.25,0.5)	-0.1531201 (0.5,0.5)	-0.1917065 (0.75,0.5)	-0.1961827 (1,0.5)	-0.1954211 (1.25,0.5)
BC8	-0.0496848 (0.25,0.5)	-0.1531200 (0.5,0.5)	-0.1917195 (0.75,0.5)	-0.1962343 (1,0.5)	-0.1955134 (1.25,0.5)
BC9	-0.0496908 (0.25,0.5)	-0.1531200 (0.5,0.5)	-0.1916935 (0.75,0.5)	-0.1961311 (1,0.5)	-0.1953290 (1.25,0.5)

DICRETIZATION

The discretization is done according to the following figure and if the no of divisions gets increased it is done in the same manner. This discretization is done for 100 elements. The node and element number is shown accordingly

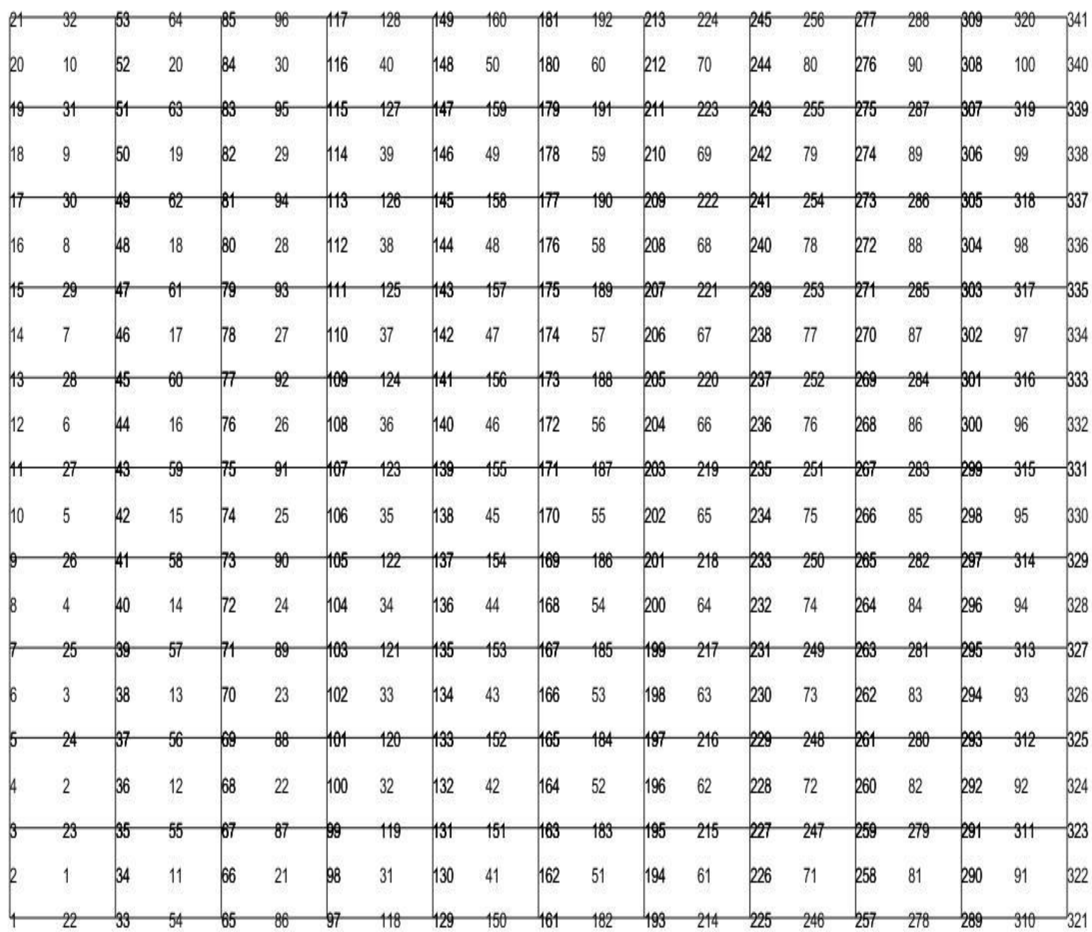

Figure 13 Mesh for 10 division in both x and y direction of plate

For a plate of length l=1m, and breadth b=1m, in the following table the maximum displacement is plotted with respect to cut off taken out with respect to change of cut off element in the above figure. Load of 500N is applied at the center of the plate i.e., (0.5,0.5). Plate material is steel so Young's modulus is $2x10^5$ N/m^2, Poisson's ratio of 0.3 and plate thickness is 0.001 m, No. of plate elements is taken as 100.

Element No.	1	12	23	34	45	56	67	78	89	100
BC1	-0.1513	-0.1537	-0.1541	-0.1559	-0.1833	-0.1833	-0.1559	-0.1541	-0.1537	-0.1531
BC2	-0.2104	-0.2100	-0.2096	-0.2125	-0.2428	-0.2428	-0.2125	-0.2096	-0.2100	-0.2104
BC3	-0.2104	-0.2100	-0.2096	-0.2125	-0.2428	-0.2428	-0.2125	-0.2096	-0.2100	-0.2104
BC4	-0.8469	-0.8489	-0.8543	-0.8550	-0.8348	-0.8311	-0.8511	-0.8548	-0.8548	-
BC5	-0.1531	-0.1537	-0.1541	-0.1559	-0.1834	-0.1834	-0.1559	-0.1541	-0.1537	-0.1531
BC6	-0.1531	-0.1537	-0.1541	-0.1559	-0.1833	-0.1833	-0.1559	-0.1541	-0.1537	-0.1531
BC7	-0.1531	-0.1537	-0.1541	-0.1559	-0.1833	-0.1833	-0.1559	-0.1541	-0.1537	-0.1531
BC8	-0.1531	-0.1537	-0.1541	-0.1559	-0.1833	-0.1833	-0.1559	-0.1541	-0.1537	-0.1531
BC9	-0.1531	-0.1537	-0.1541	-0.1559	-0.1833	-0.1833	-0.1559	-0.1541	-0.1537	-0.1531

For a plate of length l=1m, and breadth b=1m, in the following table the maximum displacement is plotted of a cutoff plate of 45th element with respect to change in position of load at node as in the above mesh. Load of 500N is applied at the center of the plate i.e., (0.5,0.5). Plate material is steel so Young's modulus is $2x10^5$ N/m^2, Poisson's ratio of 0.3 and plate thickness is 0.001 m, No. of plate elements is taken as 100.

Node No.	**1**	**35**	**69**	**103**	**137**	**171**	**205**	**239**	**273**	**307**	**341**
BC1	-	-0.010	-0.0395	-0.0894	-0.1616	-0.1833	-0.1361	-0.1362	-0.0394	-0.01035	-
BC2	-	-0.0639	-0.1287	-0.1646	-0.2297	-0.2428	-0.1999	-0.1619	-0.1285	-0.06382	-
BC3	-	-0.0639	-0.1287	-0.1646	-0.2297	-0.2428	-0.1999	-0.1619	-0.1285	-0.06382	-
BC4	-	-0.0103	-0.0400	-0.1076	-0.3443	-0.8348	-1.6323	-2.7351	-4.1748	-5.96955	-8.1456
BC5	-	-0.0104	-0.0400	-0.0895	-0.1617	-0.1834	-0.1362	-0.0872	-0.0399	-0.01041	-
BC6	-	-0.0103	-0.0398	-0.0894	-0.1616	-0.1833	-0.1362	-0.0872	-0.0396	-0.01038	-
BC7	-	-0.0103	-0.0398	-0.0894	-0.1616	-0.1833	-0.1362	-0.0872	-0.0396	-0.01038	-
BC8	-	-0.0103	-0.0398	-0.0894	-0.1616	-0.1833	-0.1362	-0.0872	-0.0396	-0.01038	-
BC9	--	-0.0103	-0.0398	-0.0894	-0.1616	-0.1833	-0.1362	-0.0872	-0.0396	-0.01038	-

3.2 Stress resultant using MATLAB

As we know we can calculate strain at every Gauss point of the element and by extrapolating the strain we can calculate strain at each and every node. Similarly, we can calculate stress at each node. Stress resultant M_x, M_y, Q_x, Q_y at different boundary conditions with load 500N at center and can be seen at every figure given below.

<u>Case 1-</u>

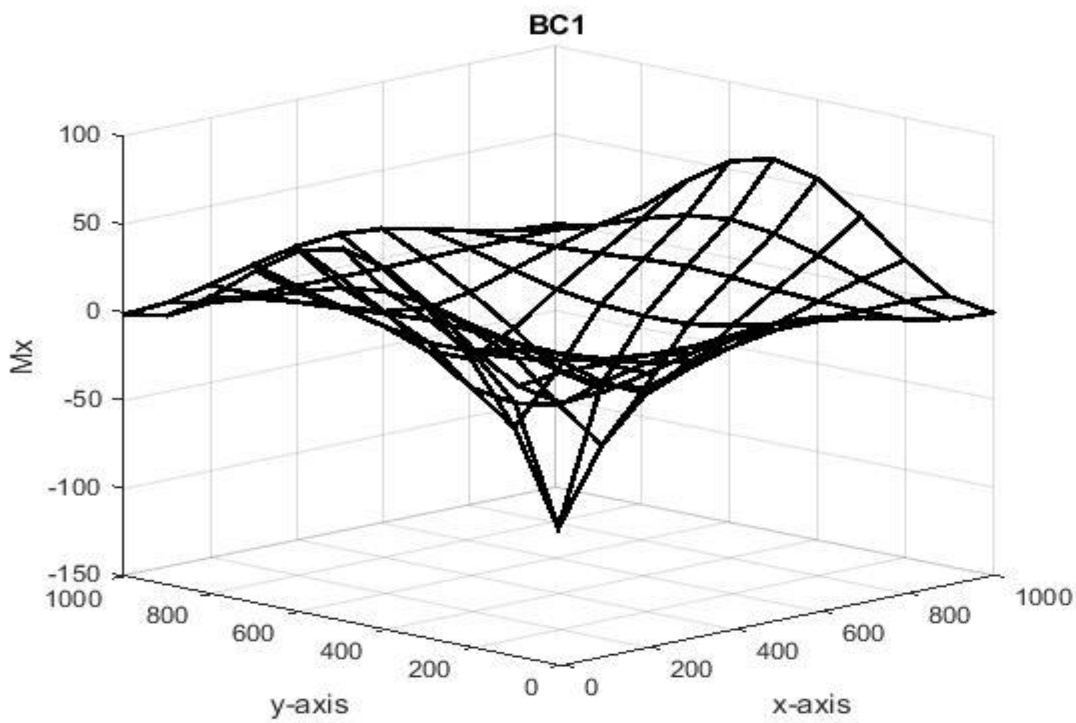

Figure 14 Bending moment acting on face normal to x-direction

Clamped on all sides

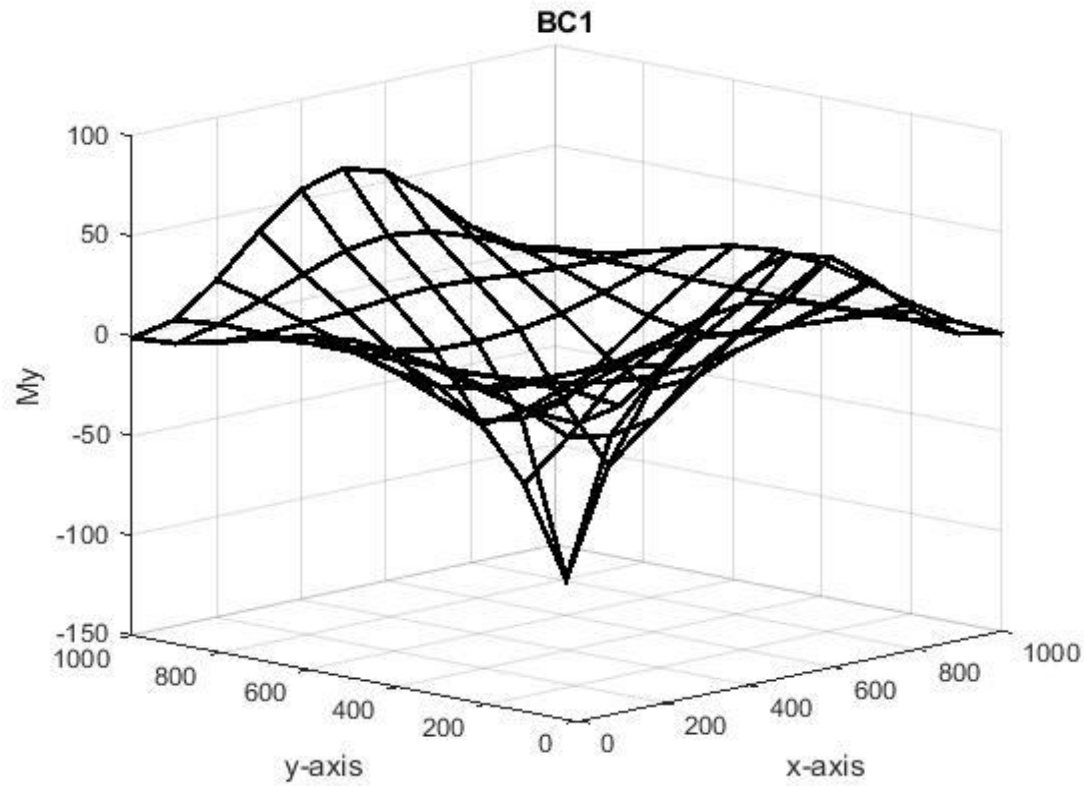

Figure 15 Bending moment acting on face normal to y-direction

Clamped on all sides

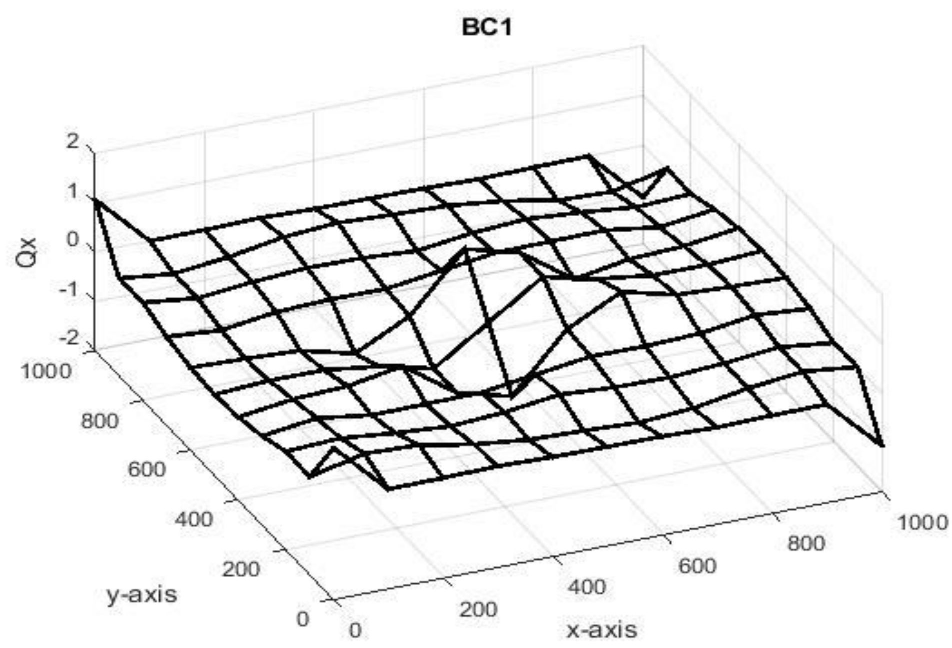

Figure 16 Out of plane Shear Force along x direction

Clamped on all sides

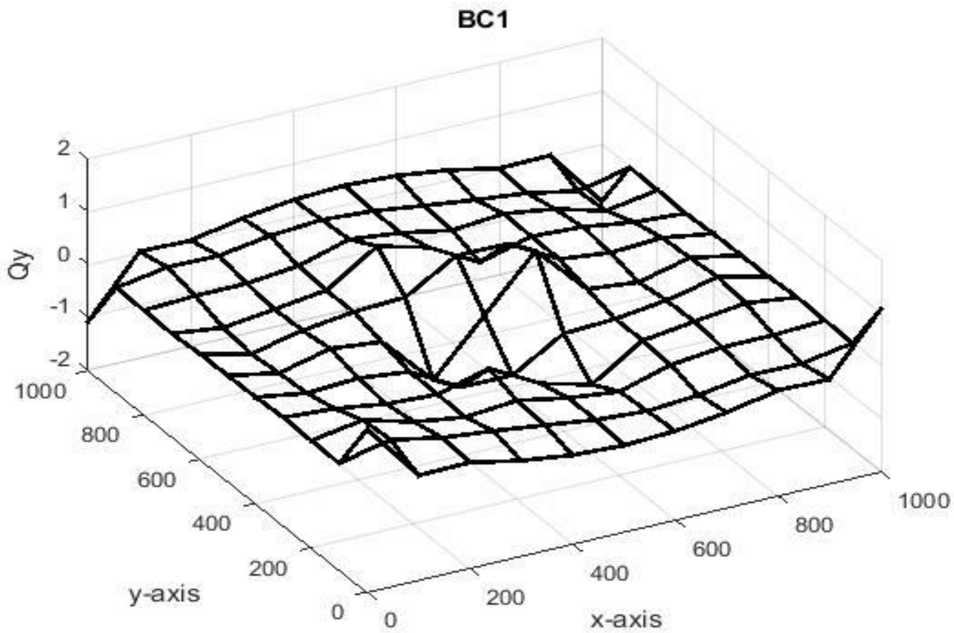

Figure 17 Out of plane Shear Force along y direction

Clamped on all sides

<u>Case 2-</u>

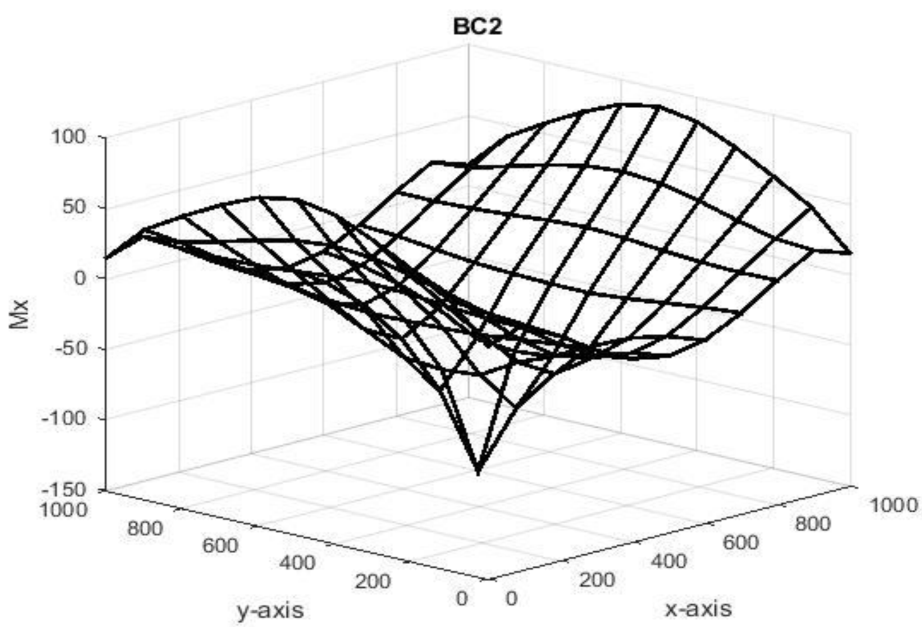

Figure 18 Bending moment acting on face normal to x-direction

Clamped for x=0, x=l

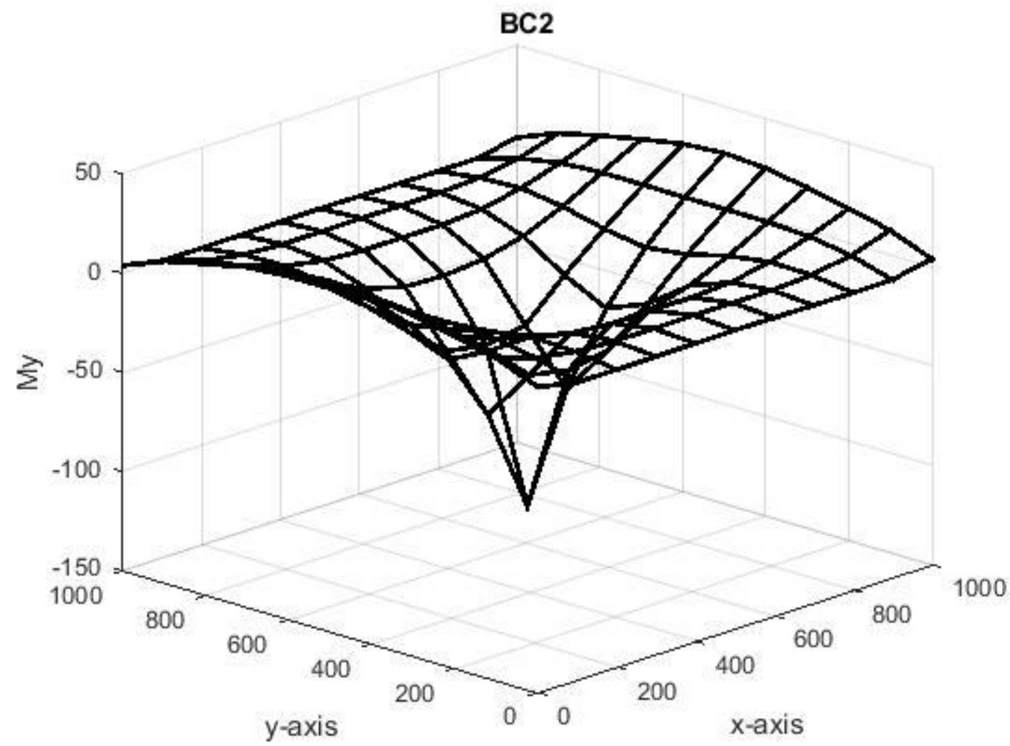

Figure 19 Bending moment acting on face normal to y-direction

Clamped for x=0, x=l

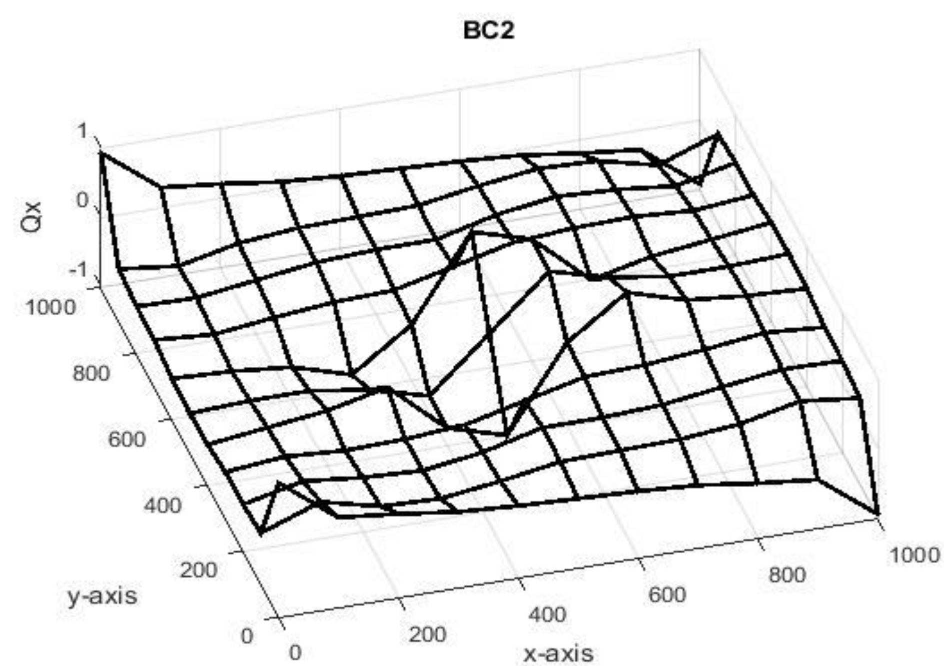

Figure 20 Out of plane Shear Force along x direction

Clamped for x=0, x=l

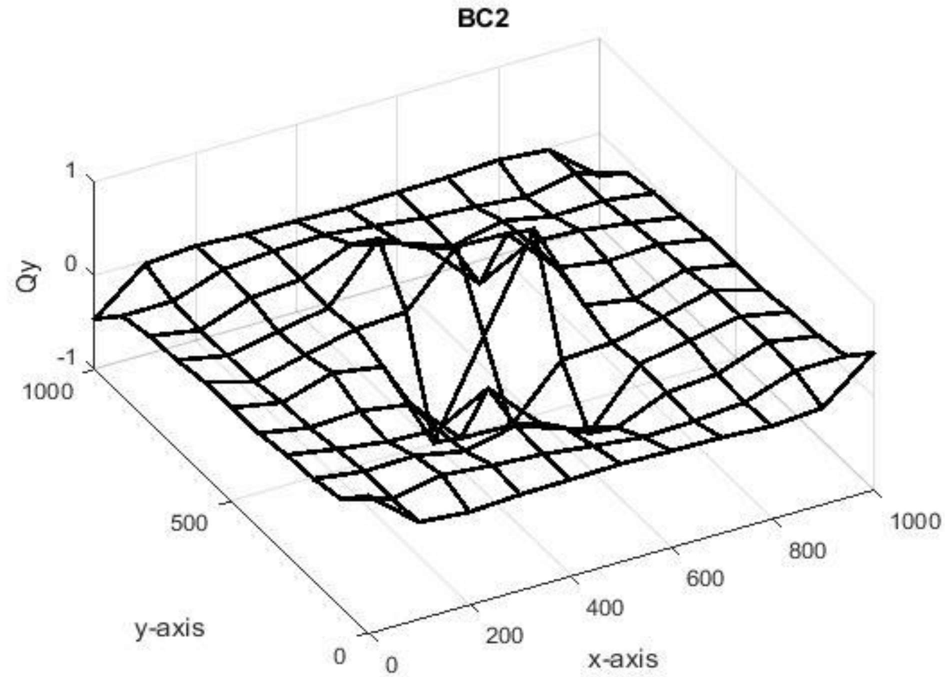

Figure 21 Out of plane Shear Force along y direction

Clamped for x=0, x=l

<u>Case 5-</u>

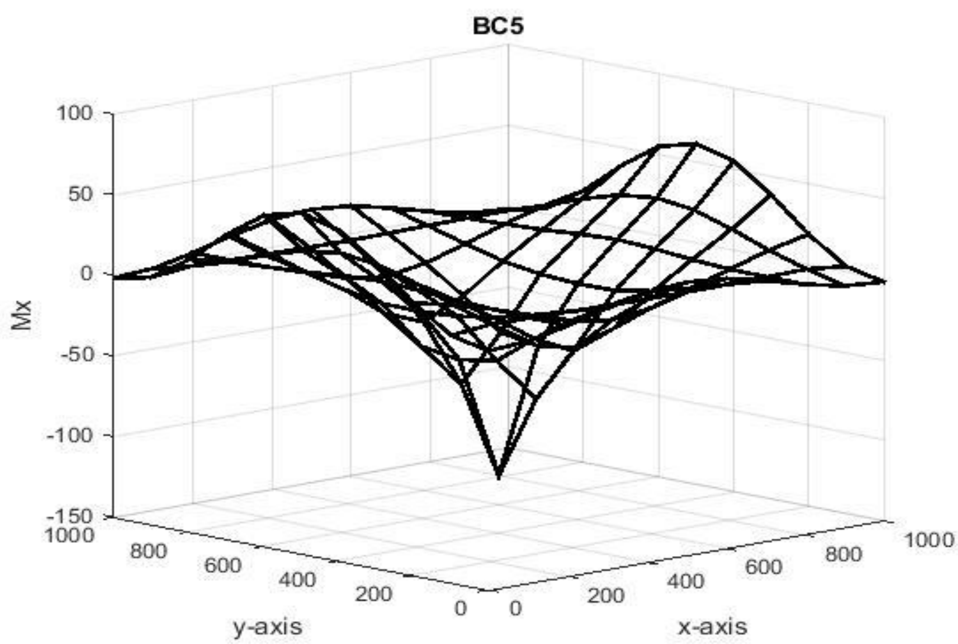

Figure 22 Bending moment acting on face normal to x-direction

Simply Supported for all sides

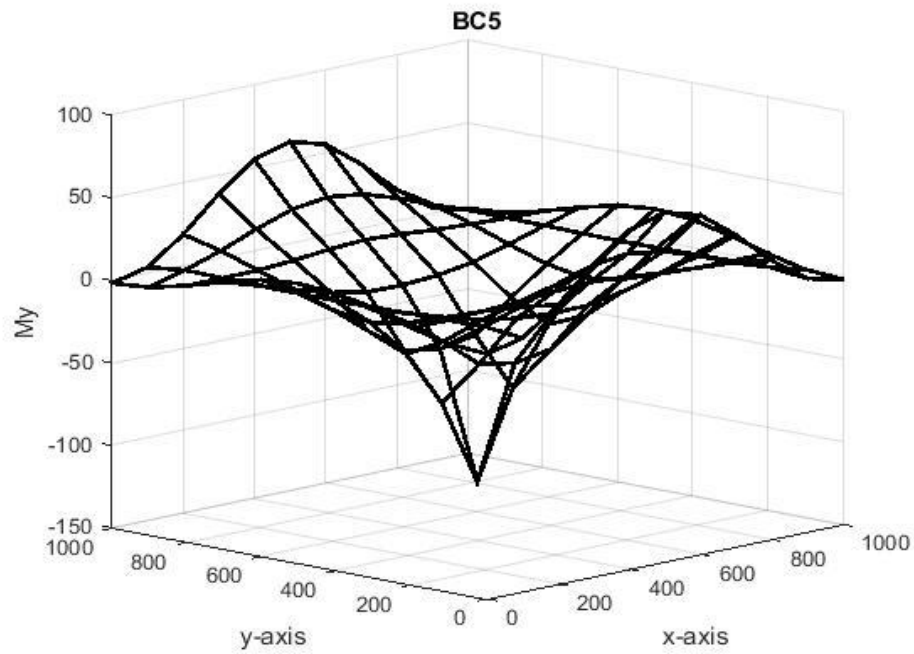

Figure 23 Bending moment acting on face normal to y-direction

Simply Supported for all sides

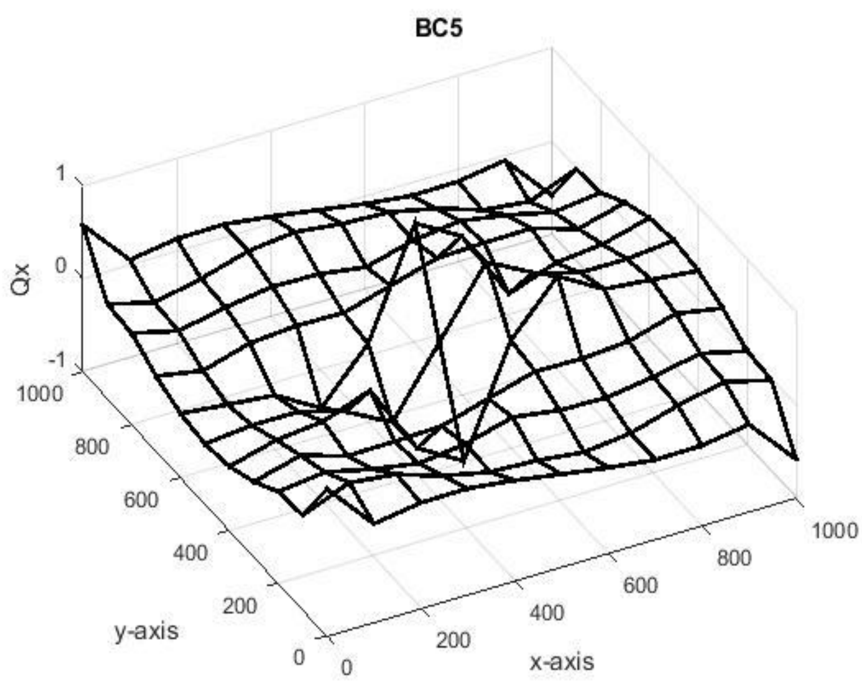

Figure 24 Out of plane Shear Force along x direction

Simply Supported for all sides

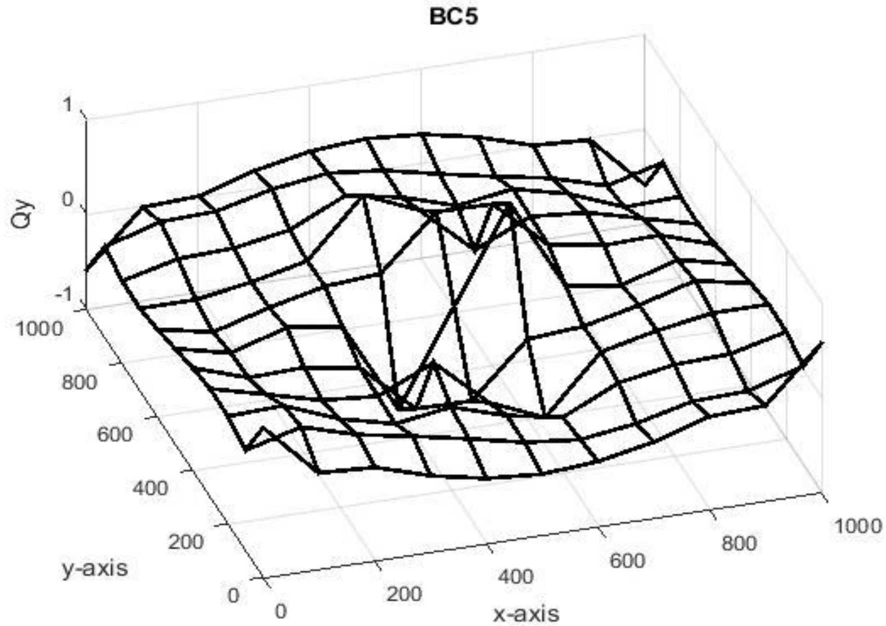

Figure 25 Out of plane Shear Force along x direction

Simply Supported for all sides

<u>Case 8-</u>

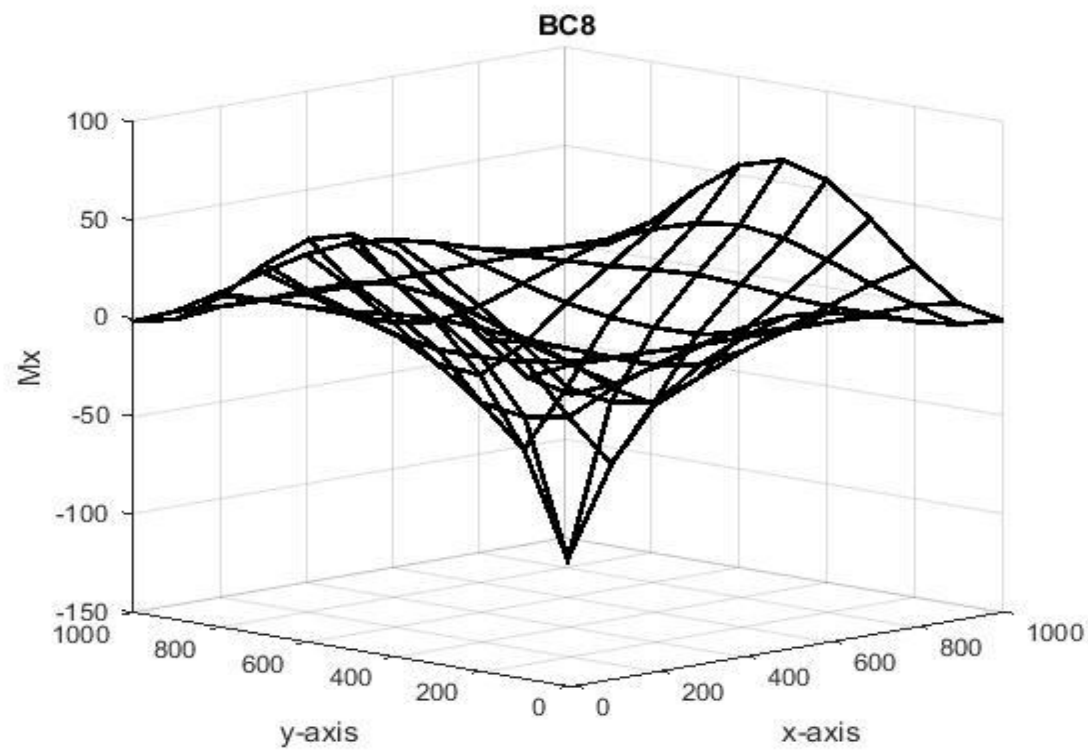

Figure 26 Bending moment acting on face normal to x-direction

Simply Supported for x=0, x=l and Clamped for y=0, y=b

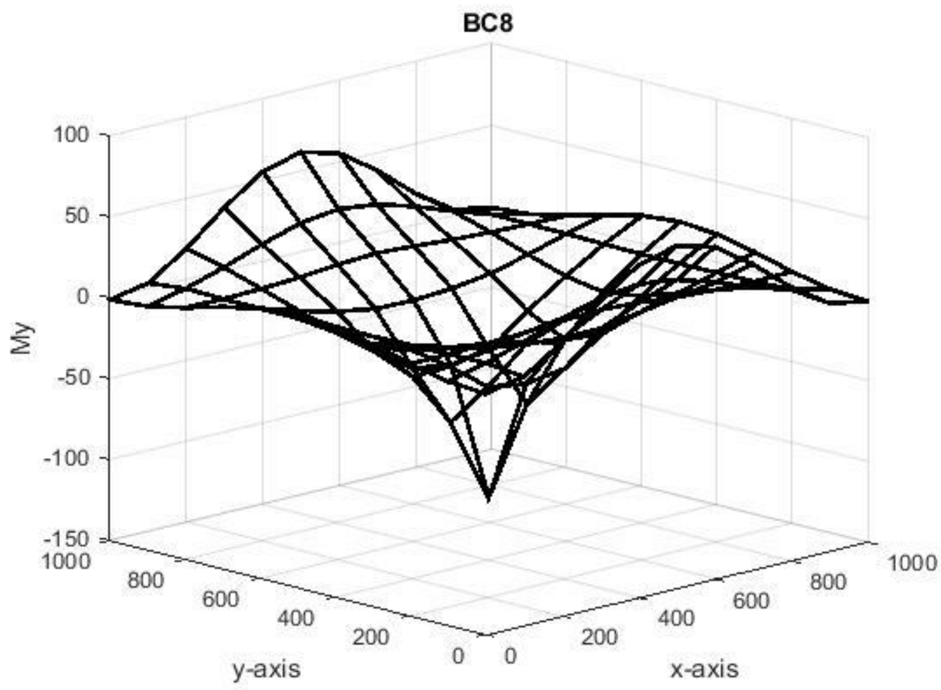

Figure 27 Bending moment acting on face normal to y-direction

Simply Supported for x=0, x=l and Clamped for y=0, y=b

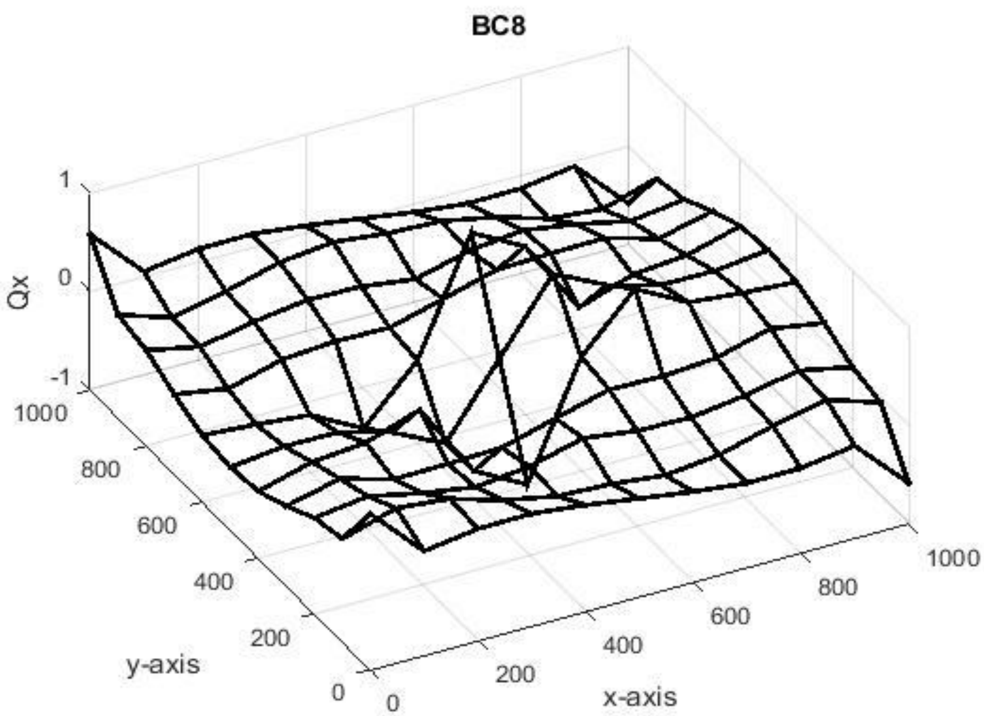

Figure 28 Out of plane Shear Force along x direction

Simply Supported for x=0, x=l and Clamped for y=0, y=b

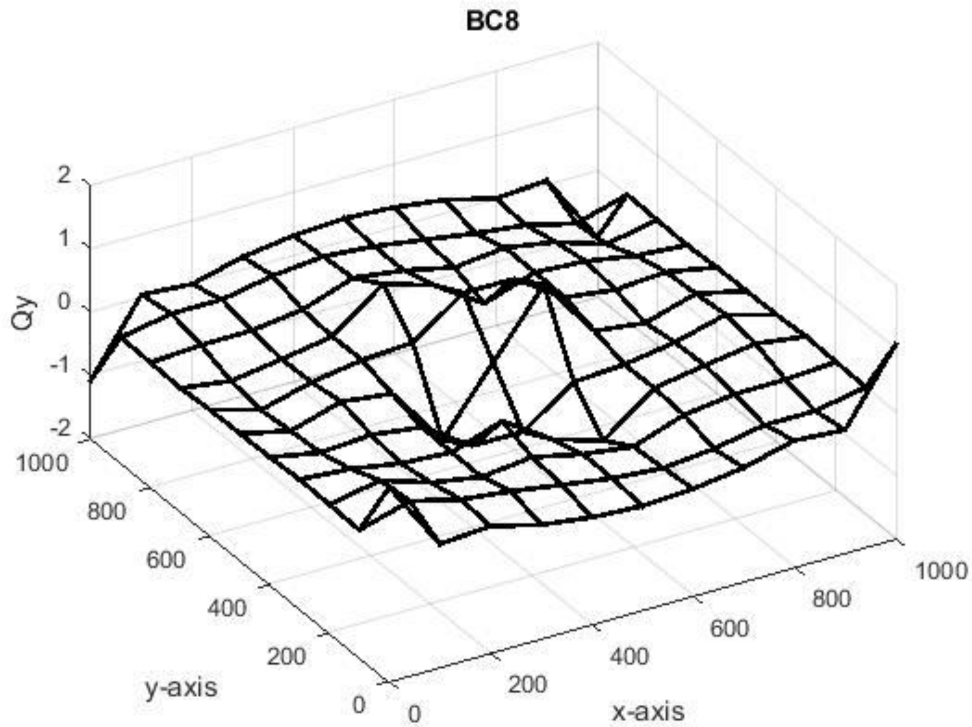

Figure 29 Out of plane Shear Force along x direction

Simply Supported for x=0, x=l and Clamped for y=0, y=b

Conclusions and Future Scope

4.1 Conclusion

These problems that are encountered here are very common in nature we can easily find structures having plates on which constant pressures (may be even for a small-time interval but constant) are applied such as the top plate of table, piston head, leaf valve, thin tin plate against fast moving wind etc. We can see that due to symmetry we can easily predict that in a rectangular plate that may be clamped from all edges, simply supported from all edges, clamped and simply supported etc. Maximum deflection is found to be at the center of the plate in all cases except for BC4 in which there is a free edge and is maximum there and the value of the deflection which are obtained from the finite element method using the MATLAB program is getting more and more accurate. Stress resultant calculated at Gauss points gives us plot of bending moment and out of plane shear which vary maximum at center. Stress/Strain values can be computed for particular point on plate and can be used for further computation.

4.2 Future Scope

Since the present MATLAB code can be appended with the new and extra code without disturbing the original code there is a scope to find out stresses, strains, analysis of plate with patch loading conditions, Bending moment, shear forces, and analysis of skew plates, circular plates, triangular plates etc.

Appendix

```
clc;
clear;
close all;

l=input(' enter the length of the plate ');
h=input(' enter the breadth of the plate ');
ndx=input(' enter the no of divisions on the length ');
ndy=input(' enter the no of divisions on the breadth ');
th=input(' enter the thickness');
E=input('Youngs modulus');
v=input('poissons ratio');
disp(' enter 1 to analyse for Plate clamped on all sides ')
disp(' enter 2 to analyse for Plate clamped on x=0, x=l')
disp(' enter 3 to analyse for Plate clamped on y=0, y=b ')
disp(' enter 4 to analyse for Plate clamped on x=0, y=0 ')
disp(' enter 5 to analyse for Simply supported plate on all sides')
disp(' enter 6 to analyse for Plate clamped on x=0, y=b and Simply supported plate on
x=l, y=0 ')
disp(' enter 7 to analyse for Plate clamped on x=l, y=0 and Simply supported plate on
x=0, y=b ')
disp(' enter 8 to analyse for Plate clamped on x=0, x=l and Simply supported plate on
y=0, y=b ')
disp(' enter 9 to analyse for Plate clamped on y=0, y=b and Simply supported plate on
x=0, x=l ')
q=input('enter your choice ');
c=input('enter load in N');


a=l/(ndx);
s=h/(ndy);
nx=2*(ndx)+1;
ny=2*(ndy)+1;




%nx=no. of nodes along x-axis
%ny=no. of nodes along y-axis


%mesh of 8 noded plate


x=linspace(0,l,nx);
% x2=linspace(0,l,(nx+1)/2);
y1=linspace(0,h,ny);
y2=linspace(0,h,(ny+1)/2);
n=1;
b=1;
k1=1;
```

```
k2=0;
total=((ny+1)/2)*nx+((ny-1)/2)*((nx+1)/2);
nodept=zeros(3,total);

for j=1:total
    k2=k2+1;
    for i=1:3
        if i==1
            nodept(i,j)=n;
        elseif i==2
        nodept(i,j)=x(k1);
        else
            if rem(k1,2)==0
                nodept(i,j)=y2(b);
            else
                nodept(i,j)=y1(b);
            end
        end
    end
    if rem(k2,ny)==0
        k1=k1+1;
        b=0;
    elseif  (rem((k2-ny),(ny+1)/2)==0)&&(k2>((ny-1)/2))
        b=0;
        k2=0;
        k1=k1+1;
    end
    n=n+1;
    b=b+1;
end

ne=((nx-1)/2)*((ny-1)/2);
c1=1;
c2=(ny-1)/2;
c3=ny;
elnn=zeros(ne,9);

for i=1:ne
    elnn(i,1)=i;
    elnn(i,2)=c1;
    elnn(i,3)=elnn(i,2)+c3;
    elnn(i,4)=c1+ny+((ny+1)/2);
    elnn(i,5)=elnn(i,4)+1;
    elnn(i,6)=elnn(i,5)+1;
    elnn(i,7)=elnn(i,3)+1;
    elnn(i,8)=c1+2;
    elnn(i,9)=c1+1;

   if rem(i,c2)==0
        c1=c1+c2+2;
        c3=ny+1;
    end
    c1=c1+2;
    c3=c3-1;
end
```

```matlab
Pos=elnn(:,2:9);
Cor=(nodept(2:3,:))';




Number=size(Pos);
No=Number(1);                   %Plate Element Number


Number=size(Cor);
Node=Number(1);                 %Plate System Node Number


for i=1:Node
    Re(i,:)=[1 1 1 1 1];
end

for i=1:No
    for j=1:Node
        Cor2(1,1,i)=Cor(Pos(i,1),1)+a*(1/2)-a/2*(0.57735026919);
        Cor2(1,2,i)=Cor(Pos(i,1),2)+s*(1/2)-s/2*(0.57735026919);
        Cor2(2,1,i)=Cor(Pos(i,1),1)+a*(1/2)-a/2*(0.57735026919);
        Cor2(2,2,i)=Cor(Pos(i,1),2)+s*(1/2)+s/2*(0.57735026919);
        Cor2(3,1,i)=Cor(Pos(i,1),1)+a*(1/2)+a/2*(0.57735026919);
        Cor2(3,2,i)=Cor(Pos(i,1),2)+s*(1/2)-s/2*(0.57735026919);
        Cor2(4,1,i)=Cor(Pos(i,1),1)+a*(1/2)+a/2*(0.57735026919);
        Cor2(4,2,i)=Cor(Pos(i,1),2)+s*(1/2)+s/2*(0.57735026919);


    end
end




Number=size(Re);
Nom=Number(2);                   %Plane Element node d.o.f Number




sayman=0;
for i=1:Node
    for j=1:Nom
        if Re(i,j)==1
            sayman = sayman +1;
            Re(i,j) = sayman;
        end
    end
end

Item=sayman;
```

```
% Constitutive matrix

    gxy= E/(2*(1+v));
    gxz= gxy/(1.2);
    gyz= gxy/(1.2);



 C=    [E*th/(1-(v^2))      E*th*v/(1-(v^2))      0               0
0                               0                0        0        ;

                       E*th*v/(1-(v^2))    E*th/(1-(v^2))        0               0
0                               0                0        0        ;

                       0                   0                     gxy*th          0
0                               0                0        0        ;

                       0                   0                     0
E*(th)/(1-(v^2))*(th^2)/12      E*(th)*v/(1-(v^2))*(th^2)/12    0                0
0       ;

                       0                   0                     0
E*(th)*v/(1-(v^2))*(th^2)/12    E*(th)/(1-(v^2))*(th^2)/12      0                0
0       ;

                       0                   0                     0               0
0                               gxy*(th^3)/12    0        0        ;

                       0                   0                     0               0
0                               0                gxz*th   0        ;

                       0                   0                     0               0
0                               0                0        gyz*th] ;



K1(8,40,4,:)=0;
K4(40,40,4,:)=0;
K3(40,40,4,:)=0;
K(40,40,:)=0;



for i=1:No
    for j=1:Nom
        R(i,j+0*Nom) = Re(Pos(i,1),j);
        R(i,j+7*Nom) = Re(Pos(i,2),j);
        R(i,j+3*Nom) = Re(Pos(i,3),j);
        R(i,j+6*Nom) = Re(Pos(i,4),j);
        R(i,j+2*Nom) = Re(Pos(i,5),j);
        R(i,j+5*Nom) = Re(Pos(i,6),j);
        R(i,j+1*Nom) = Re(Pos(i,7),j);
        R(i,j+4*Nom) = Re(Pos(i,8),j);
    end
end
```

```
for Elemanno=1:No
    px1(Elemanno)=Cor(Pos(Elemanno,1),1);
    px2(Elemanno)=Cor(Pos(Elemanno,2),1);
    px3(Elemanno)=Cor(Pos(Elemanno,3),1);
    px4(Elemanno)=Cor(Pos(Elemanno,4),1);
    px5(Elemanno)=Cor(Pos(Elemanno,5),1);
    px6(Elemanno)=Cor(Pos(Elemanno,6),1);
    px7(Elemanno)=Cor(Pos(Elemanno,7),1);
    px8(Elemanno)=Cor(Pos(Elemanno,8),1);

    py1(Elemanno)=Cor(Pos(Elemanno,1),2);
    py2(Elemanno)=Cor(Pos(Elemanno,2),2);
    py3(Elemanno)=Cor(Pos(Elemanno,3),2);
    py4(Elemanno)=Cor(Pos(Elemanno,4),2);
    py5(Elemanno)=Cor(Pos(Elemanno,5),2);
    py6(Elemanno)=Cor(Pos(Elemanno,6),2);
    py7(Elemanno)=Cor(Pos(Elemanno,7),2);
    py8(Elemanno)=Cor(Pos(Elemanno,8),2);


end

B=zeros(size(Re));

Number=size(B);
N=Number(1);

%Boundary Conditions

    if q==1

s=0;
for i=1:N
    for j=1:Nom
         if Cor(i,1)==0
             B(i,j)=s+1;
         elseif Cor(i,2)==0
             B(i,j)=s+1;
         elseif Cor(i,1)==l
             B(i,j)=s+1;
        elseif Cor(i,2)==h
             B(i,j)=s+1;
         end
    end
end

elseif q==2

s=0;
for i=1:N
    for j=1:Nom
         if Cor(i,1)==0
             B(i,j)=s+1;
         elseif Cor(i,1)==l
             B(i,j)=s+1;
         end
    end
end
```

```
elseif q==3

s=0;
for i=1:N
    for j=1:Nom
         if Cor(i,2)==0
             B(i,j)=s+1;
         elseif Cor(i,2)==h
             B(i,j)=s+1;
         end
    end
end

elseif q==4

s=0;
for i=1:N
    for j=1:Nom
         if Cor(i,1)==0
             B(i,j)=s+1;
         elseif Cor(i,2)==0
             B(i,j)=s+1;
         end
    end
end

elseif q==5

s=0;
for i=1:N
    for j=1:Nom
        if Cor(i,1)==0&&Cor(i,2)==0
             B(i,j)=s+1;
        elseif Cor(i,1)==0&&Cor(i,2)==h
             B(i,j)=s+1;
        elseif Cor(i,1)==1&&Cor(i,2)==0
             B(i,j)=s+1;
        elseif Cor(i,1)==1&&Cor(i,2)==h
             B(i,j)=s+1;
        elseif Cor(i,1)==0
             if Cor(i,2)~=0
                 if Cor(i,2)~=h
             B(i,1)=s+1;
             B(i,3)=s+1;
             B(i,4)=s+1;
                 end
             end
         elseif Cor(i,2)==0
              if Cor(i,1)~=0
                 if Cor(i,1)~=1
             B(i,2)=s+1;
             B(i,3)=s+1;
             B(i,5)=s+1;
                 end
              end
         elseif Cor(i,1)==1
              if Cor(i,2)~=0
                 if Cor(i,2)~=h
             B(i,1)=s+1;
             B(i,3)=s+1;
```

```
             B(i,4)=s+1;
                 end
              end
         elseif Cor(i,2)==h
             if Cor(i,1)~=0
                  if Cor(i,1)~=l
              B(i,2)=s+1;
              B(i,3)=s+1;
              B(i,5)=s+1;
                  end
             end

          end
     end
end

elseif q==6

s=0;
for i=1:N
    for j=1:Nom
        if Cor(i,1)==0&&Cor(i,2)==0
             B(i,j)=s+1;
        elseif Cor(i,1)==0&&Cor(i,2)==h
             B(i,j)=s+1;
        elseif Cor(i,1)==l&&Cor(i,2)==0
             B(i,j)=s+1;
        elseif Cor(i,1)==l&&Cor(i,2)==h
             B(i,j)=s+1;

        elseif Cor(i,1)==0
             if Cor(i,2)~=0
                 if Cor(i,2)~=h
             B(i,j)=s+1;
                 end
             end
         elseif Cor(i,2)==0
              if Cor(i,1)~=0
                 if Cor(i,1)~=l
             B(i,2)=s+1;
             B(i,3)=s+1;
             B(i,5)=s+1;
                 end
              end
         elseif Cor(i,1)==l
              if Cor(i,2)~=0
                 if Cor(i,2)~=h
             B(i,1)=s+1;
             B(i,3)=s+1;
             B(i,4)=s+1;
                 end
              end
        elseif Cor(i,2)==h
            if Cor(i,1)~=0
                 if Cor(i,1)~=l
             B(i,j)=s+1;
                 end
            end

         end
```

```
 end
end

elseif q==7

s=0;
for i=1:N
    for j=1:Nom
        if Cor(i,1)==0&&Cor(i,2)==0
             B(i,j)=s+1;
        elseif Cor(i,1)==0&&Cor(i,2)==h
             B(i,j)=s+1;
        elseif Cor(i,1)==l&&Cor(i,2)==0
             B(i,j)=s+1;
        elseif Cor(i,1)==l&&Cor(i,2)==h
             B(i,j)=s+1;

        elseif Cor(i,1)==0
             if Cor(i,2)~=0
                 if Cor(i,2)~=h
             B(i,1)=s+1;
             B(i,3)=s+1;
             B(i,4)=s+1;
                 end
             end
         elseif Cor(i,2)==0
              if Cor(i,1)~=0
                 if Cor(i,1)~=l
             B(i,j)=s+1;
                 end
              end
         elseif Cor(i,1)==l
              if Cor(i,2)~=0
                 if Cor(i,2)~=h
             B(i,j)=s+1;
                 end
              end
        elseif Cor(i,2)==h
            if Cor(i,1)~=0
                 if Cor(i,1)~=l
             B(i,2)=s+1;
             B(i,3)=s+1;
             B(i,5)=s+1;
                 end
            end

         end
    end
end

elseif q==8

s=0;
for i=1:N
    for j=1:Nom
        if Cor(i,1)==0&&Cor(i,2)==0
             B(i,j)=s+1;
        elseif Cor(i,1)==0&&Cor(i,2)==h
             B(i,j)=s+1;
        elseif Cor(i,1)==l&&Cor(i,2)==0
             B(i,j)=s+1;
```

```
        elseif Cor(i,1)==l&&Cor(i,2)==h
             B(i,j)=s+1;

        elseif Cor(i,1)==0
             if Cor(i,2)~=0
                 if Cor(i,2)~=h
             B(i,j)=s+1;
                 end
             end
         elseif Cor(i,2)==0
              if Cor(i,1)~=0
                 if Cor(i,1)~=l
             B(i,2)=s+1;
             B(i,3)=s+1;
             B(i,5)=s+1;
                 end
              end
         elseif Cor(i,1)==l
              if Cor(i,2)~=0
                 if Cor(i,2)~=h
             B(i,j)=s+1;
                 end
              end
        elseif Cor(i,2)==h
            if Cor(i,1)~=0
                 if Cor(i,1)~=l
             B(i,2)=s+1;
             B(i,3)=s+1;
             B(i,5)=s+1;
                 end
            end

         end
    end
end


elseif q==9

s=0;
for i=1:N
    for j=1:Nom
        if Cor(i,1)==0&&Cor(i,2)==0
             B(i,j)=s+1;
        elseif Cor(i,1)==0&&Cor(i,2)==h
             B(i,j)=s+1;
        elseif Cor(i,1)==l&&Cor(i,2)==0
             B(i,j)=s+1;
        elseif Cor(i,1)==l&&Cor(i,2)==h
             B(i,j)=s+1;

        elseif Cor(i,1)==0
             if Cor(i,2)~=0
                 if Cor(i,2)~=h
             B(i,1)=s+1;
             B(i,3)=s+1;
             B(i,4)=s+1;
                 end
             end
         elseif Cor(i,2)==0
              if Cor(i,1)~=0
```

```
         if Cor(i,1)~=l
             B(i,j)=s+1;
                 end
              end
         elseif Cor(i,1)==l
              if Cor(i,2)~=0
                 if Cor(i,2)~=h
             B(i,1)=s+1;
             B(i,3)=s+1;
             B(i,4)=s+1;
                 end
              end
        elseif Cor(i,2)==h
            if Cor(i,1)~=0
                 if Cor(i,1)~=l
             B(i,j)=s+1;
                 end
            end

         end
    end
end



    end




% Gauss Points

W=[-0.57735026919 ;
   -0.57735026919;
   0.57735026919;
   0.57735026919];

O=[-0.57735026919 ;
   0.57735026919;
   -0.57735026919 ;
   0.57735026919];


for s=1:No
    for k=1:4


        e=W(k);
        n=O(k);

       X1=px1(s);     X2=px7(s);     X3=px5(s);     X4=px3(s);
        X5=px8(s);     X6=px6(s);     X7=px4(s);     X8=px2(s);

        Y1=py1(s);     Y2=py7(s);     Y3=py5(s);     Y4=py3(s);
        Y5=py8(s);     Y6=py6(s);     Y7=py4(s);     Y8=py2(s);
```

```
% Derative of shape functions with e

   dx1= (-(1-n)*(-n-e-1)/4.0-(1-e)*(1-n)/4.0);
   dx2= (-(n+1)*(n-e-1)/4.0-(1-e)*(n+1)/4.0);
   dx3= ((n+1)*(n+e-1)/4.0+(e+1)*(n+1)/4.0);
   dx4= ((1-n)*(-n+e-1)/4.0+(e+1)*(1-n)/4.0);
   dx5=  -(1-n^2)/2;
   dx6=  -e*(n+1);
   dx7=  (1-n^2)/2;
   dx8=  -e*(1-n);


   % Derative of shape functions with n

   dy1= (-(1-e)*(-n-e-1)/4.0-(1-e)*(1-n)/4.0);
   dy2= ((1-e)*(n-e-1)/4.0+(1-e)*(n+1)/4.0);
   dy3= ((e+1)*(n+e-1)/4.0+(e+1)*(n+1)/4.0);
   dy4= (-(e+1)*(-n+e-1)/4.0-(e+1)*(1-n)/4.0);
   dy5= -(1-e)*n;
   dy6= (1-e^2)/2;
   dy7= -(e+1)*n;
   dy8= -(1-e^2)/2;



   J11 = dx1*X1+dx2*X2+dx3*X3+dx4*X4+dx5*X5+dx6*X6+dx7*X7+dx8*X8;
   J12 = dx1*Y1+dx2*Y2+dx3*Y3+dx4*Y4+dx5*Y5+dx6*Y6+dx7*Y7+dx8*Y8;
   J21 = dy1*X1+dy2*X2+dy3*X3+dy4*X4+dy5*X5+dy6*X6+dy7*X7+dy8*X8;
   J22 = dy1*Y1+dy2*Y2+dy3*Y3+dy4*Y4+dy5*Y5+dy6*Y6+dy7*Y7+dy8*Y8;


   %Jacobian matrix

   Jacobi=[ J11    J12;
           J21    J22];


           InvJ=Jacobi^-1;

           g11=InvJ(1,1);   g12=InvJ(1,2);
           g21=InvJ(2,1);   g22=InvJ(2,2);

   Hx1=g11*dx1+g12*dy1;
   Hx2=g11*dx2+g12*dy2;
   Hx3=g11*dx3+g12*dy3;
   Hx4=g11*dx4+g12*dy4;
   Hx5=g11*dx5+g12*dy5;
   Hx6=g11*dx6+g12*dy6;
   Hx7=g11*dx7+g12*dy7;
   Hx8=g11*dx8+g12*dy8;

   Hy1=g21*dx1+g22*dy1;
   Hy2=g21*dx2+g22*dy2;
   Hy3=g21*dx3+g22*dy3;
   Hy4=g21*dx4+g22*dy4;
   Hy5=g21*dx5+g22*dy5;
   Hy6=g21*dx6+g22*dy6;
   Hy7=g21*dx7+g22*dy7;
   Hy8=g21*dx8+g22*dy8;
```

```
 %Shape functions

N1=1/4*(1-(e))*(1-(n))*(-(e)-(n)-1);
N2=1/4*(1-(e))*(1+(n))*(-(e)+(n)-1);
N3=1/4*(1+(e))*(1+(n))*((e)+(n)-1);
N4=1/4*(1+(e))*(1-(n))*((e)-(n)-1);

N5=1/2*(1-(n)^2)*(1-(e));
N6=1/2*(1-(e)^2)*(1+(n));
N7=1/2*(1-(n)^2)*(1+(e));
N8=1/2*(1-(e)^2)*(1-(n));

% Strain-Displacement matrix

 A(:,:,k,s)=[   Hx1     0       0       0       0       Hx2      0      0     0     0
Hx3    0      0     0      0        Hx4     0      0     0       0        Hx5   0      0
0       0      Hx6       0     0       0        0       Hx7    0     0      0       0
Hx8     0       0       0       0    ;

                0       Hy1     0       0       0       0        Hy2    0     0     0
0      Hy3    0     0      0        0       Hy4     0     0       0        0     Hy5    0
0       0      0       Hy6   0       0        0       0      Hy7   0      0       0
0       Hy8     0       0       0    ;

                Hy1     Hx1     0       0       0       Hy2      Hx2    0     0     0
Hy3    Hx3    0     0      0        Hy4     Hx4     0     0       0        Hy5   Hx5    0
0       0      Hy6       Hx6   0       0        0       Hy7    Hx7   0      0       0
Hy8     Hx8     0       0       0    ;

                0       0       0       Hx1     0       0        0      0     Hx2   0
0      0      0     Hx3    0        0       0      0     Hx4     0        0     0      0
Hx5     0      0         0     0       Hx6      0       0      0     0      Hx7     0
0       0       0       Hx8     0    ;

                0       0       0       0       Hy1     0        0      0     0     Hy2
0      0      0     0      Hy3      0       0      0     0       Hy4      0     0      0
0       Hy5    0         0     0       0        Hy6     0      0     0      0       Hy7
0       0       0       0       Hy8  ;

                0       0       0       Hy1     Hx1     0        0      0     Hy2   Hx2
0      0      0     Hy3    Hx3      0       0      0     Hy4     Hx4      0     0      0
Hy5     Hx5    0         0     0       Hy6      Hx6     0      0     0      Hy7     Hx7
0       0       0       Hy8     Hx8  ;

                0       0       Hx1     N1      0       0        0      Hx2   N2    0
0      0      Hx3   N3     0        0       0      Hx4   N4      0        0     0
Hx5     N5      0      0        0     Hx6     N6        0       0      0     Hx7    N7
0       0       0       Hx8     N8      0    ;

                0       0       Hy1     0       N1      0        0      Hy2   0     N2
0      0      Hy3   0      N3       0       0      Hy4   0       N4       0     0
Hy5     0      N5      0        0     Hy6     0         N6      0      0     Hy7    0
N7      0       0       Hy8     0       N8  ];



    K1(:,:,k,s)=C*A(:,:,k,s);
    K4(:,:,k,s)=(A(:,:,k,s)')*K1(:,:,k,s);
    K3(:,:,k,s)=det(Jacobi)*K4(:,:,k,s);
```

```
  K(:,:,s)=K3(:,:,1,s)+K3(:,:,2,s)+K3(:,:,3,s)+K3(:,:,4,s);     % Element stiffness
matrix

    end
end









% For cutoff of a element where, H denotes element number

% H=9;
%
% for i=1:No
%     for j=1:40
%         for k=1:40
%              if i==H
%                     K(j,k,i)=1e-16;
%              end
%         end
%     end
% end

%Assembly of Global stiffness matrix

Ksis(Item,Item)=0;
    for n=1:No
        for sat=1:40
            for sut=1:40
               if (R(n,sat)~=0)
                  if (R(n,sut)~=0)
                    Ksis(R(n,sat),R(n,sut))=Ksis(R(n,sat),R(n,sut)) + K(sut,sat,n);
                 end
              end
           end
       end
    end






B1 = reshape(B',[],1) ;
```

```
count=0;
for i=1:N
    for j=1:Nom
        if B(i,j)==0
            count=count+1;
        end
    end
end


Number=size(B);
o=Number(1);
disp(o);

% Load in z direction at (l/2,b/2)

B((o+1)/2,3)=-c;
asd=count;
AS(asd)=0;

B2 = reshape(B',[],1) ;


Number=size(B1);
   o=Number(1);



for i=o:-1:1
        if B2(i,1)==1
           B2(i,:)= [] ;
        end
end



   Number=size(B1);
   om=Number(1);


   K2=Ksis;

%Reduce stiffness matrix
   for m=om:-1:1
       if B1(m,1)==1
           K2(m,:)=[];
           K2(:,m)=[];
       end
   end


   DF1=K2\B2;
   DF2=vec2mat(DF1,5);
```

```
Number=size(B2);
mx=Number(1);




  f=1; j=1;
for i=1:N
    for j=1:Nom

            if B(i,j)~=1
            B(i,j)=DF1(f);
            f=f+1;
            else
            B(i,j)=0;
            end

    end
end

% maximum displacement in z direction
 maxdis = min(B(:,3));
 disp(maxdis);



  for i=1:No
           S(i,1) = Pos(i,1);
           S(i,2) = Pos(i,7);
           S(i,3) = Pos(i,5);
           S(i,4) = Pos(i,3);
           S(i,5) = Pos(i,8);
           S(i,6) = Pos(i,6);
           S(i,7) = Pos(i,4);
           S(i,8) = Pos(i,2);
  end



  % Calculating stress and Strain

   for i=1:No
       for l=1:4
           for j=1:8
               for k=1:Nom

           U1(j,k)=B(S(i,j),k);
           U = reshape(U1',[],1) ;

               end
           end
           Strain(:,:,l,i)=A(:,:,l,i)*U;
           Stress(:,:,l,i)=C*Strain(:,:,l,i);
       end
   end
```

```
% Plotting stress resultant

Cor3=[];
Nx=[];
Ny=[];
Mx=[];
My=[];
Qx=[];
Qy=[];

for i=1:No
    Cor3=[Cor3;Cor2(:,:,i)];
         for j=1:4
             Mx=[Mx;Stress(4,:,j,i)];
             My=[My;Stress(5,:,j,i)];
             Qx=[Qx;Stress(7,:,j,i)];
             Qy=[Qy;Stress(8,:,j,i)];
         end
end

xx=Cor3(:,1);
yy=Cor3(:,2);
zz1=Mx;
zz2=My;
zz3=Qx;
zz4=Qy;




xi=linspace(0,1000,11);
yi=linspace(0,1000,11);
F1 = scatteredInterpolant(xx,yy,zz1);
[xq,yq]=meshgrid(xi,yi) ;
figure
vq1=F1(xq,yq);
surf(xq,yq,vq1,'LineWidth',2)
alpha(0.0);
xlabel('x-axis')
ylabel('y-axis')
zlabel('Mx')
grid on

F2 = scatteredInterpolant(xx,yy,zz2);
[xq,yq]=meshgrid(xi,yi) ;
figure
vq1=F2(xq,yq);
surf(xq,yq,vq1,'LineWidth',2)
alpha(0.0);
xlabel('x-axis')
ylabel('y-axis')
zlabel('My')
grid on

F3 = scatteredInterpolant(xx,yy,zz3);
[xq,yq]=meshgrid(xi,yi) ;
figure
vq1=F3(xq,yq);
surf(xq,yq,vq1,'LineWidth',2)
alpha(0.0);
```

```
xlabel('x-axis')
ylabel('y-axis')
zlabel('Qx')
grid on

F4 = scatteredInterpolant(xx,yy,zz4);
[xq,yq]=meshgrid(xi,yi) ;
figure
vq1=F4(xq,yq);
surf(xq,yq,vq1,'LineWidth',2)
alpha(0.0);
xlabel('x-axis')
ylabel('y-axis')
zlabel('Qy')
grid on
```

References

Bakker MCM, Rosmanit M, Hofmeyer H (2008). “approximate large deflection analysis of simply supported rectangular plates under transverse loading using plate post-buckling solutions”, Struct. Design, 46(11): 1224-1235

Defu W, sheikh E (2005). “Large deflection mathematical analysis of rectangular plates”, journal of engineering mechanics, 131(8): 809- 821.

Jain NK (2009). “Analysis of Stress Concentration and Deflection in Isotropic and Orthotropic Rectangular Plates with Central Circular Hole under Transverse Static Loading”, 52th international meet World Academy of Science, Engineering and Technology, Bangalore.

Yuwaraj M. Ghugal, Atteshamuddin S. Sayyad (2010).” A static flexure of thick isotropic plates using trigonometric shear deformation theory.”, Journal of Solid Mechanics Vol. 2, No. 1 (2010) pp. 79-90

Vanam B. C. L., Rajyalakshmi M. Inala R. (2012.). “Static analysis of an isotropic rectangular plate using finite element analysis”, Journal of Mechanical Engineering Research Vol. 4(4), pp. 148-162.

A.S. Sayyad, Y. M. Ghugal (2012). “Bending and free vibration analysis of thick isotropic plates by using exponential shear deformation theory”, Applied and Computational Mechanics, 6, 65–82

A. Mahi, E. A. A. Bedia, A. Tounsi (2014). “A new hyperbolic shear deformation theory for bending and free vibration analysis of isotropic, functionally graded, sandwich and laminated composite plates”, Applied Mathematical Modelling 39, 2489–2508

Yuwaraj M. Ghugal(2017). “A Unified Shear Deformation Theory for the Bending of Isotropic, Functionally Graded, Laminated and Sandwich Beams and Plates” , International Journal of Applied Mechanics Vol. 9, No. 1, 1750007.