8-Node Quadrilateral Matlab Code for Finite Element Analysis

Akashay kumar, Akash paliwal, Mukul Gaur, Manik Gupta, Naman Jain
Department of Mechanical Engineering, Meerut Institute of Engineering and Technology, Meerut (UP), India

Abstract: The purpose to this work to provide mathematical approach to developed MATLAB code for 8-node quadrilateral element for finite element analysis of 2-D structure to improve accuracy in results. In present investigation 8-node quadrilateral element MATLAB code is proposed for structural analysis of 2-D Isotropic elastic structures. The meshing of cantilever beam is done with 8-node elements which are subjected to static loading condition. In this paper MATLAB code is implemented on cantilever beam and the result obtained is also validated with ANSYS software. Moreover, this code can also be modified for Messerschmitt-Bolkow-Beam and complete MATLAB code is also presented in appendix. The developed MATLAB code can universally adopted for other analysis such as topological optimization, thermal analysis, dynamic loading condition etc.

Keywords: Finite element method; 8-node element, Stiffness matrix, Isotropic element.

1. Introduction

The use of computational method to solve the complicated structure problem gain interest by research now a day. Out of which the finite element method [1-3] comes out to be most efficient mathematical technique to solve the structure analysis problems, which has been adopted by researcher in past few years. The application of finite element method results in reduction of cost and time by analyzing the model on computer rather than on prototype. In finite element method the given design domain is first discretized into finite element which are interconnected with each other through nodes. Then on the basic of real life problem a mathematical model is been developed by application laws of physic by considering some assumption. This mathematical model consists of partial differential equation. Solution of developed differential equation has been done for each element through different techniques and integrated with whole domain. For machining language the equation were transformed into matrix form and by solving them the approximate solution of the given problem is generated. The finite element method has application in different field such as aerospace industry, fracture mechanism, laminate analysis, creep-recovery of composite, vibration analysis etc. Many researchers were working in the field of finite element analysis were; In 2018 Ko and Bathe[4] analyze the 3-D structure by developed new 8-node hexahedral element. Jain in 2017 [5] presented a MATLAB code to solve the structural problem of linear elastic isotropic material. Moreover, MATLAB code is also validated with ANSYS software. In 2017 wang et. Al [6] worked on to improve the accuracy of finite element method solution when mesh of element result in formation of concave quadrangle by using stress quasi-conforming method to developed 8-node and 12-node element. Application of 8-node element to overcome the problem of checker boarding problem was also studied by Jain [7] in 2019.

By the application of higher node element (8-node) the problem of checkerboarding in highly reduce in optimal topology of isotropic structure. In 2019 Raju and Poluraju [8] analyzed the bubble deck slab through ANSYS and found that 30-50% void ratio presentage gives the maximum strength.

2. Linear Quadrilateral 8-Node Element

Quadrilateral 8-node element is the two dimensional element which is better applicable to curve structure as compared with 4-node element. On comparison with 4-node element it has an additional node at middle of each edge of the element. It consist of 8 nodes as shown in figure 1 (left side), each node have two degree of freedom i.e. translation motion in x and y directions. Interpolation function of 8-node element consists of higher terms ($\xi 2\eta$, $\xi\eta 2$) which make it more accuracy in the result as compare to 4-node element (1, ξ, η and $\xi\eta$) [9-10]. It is very difficult to evaluate the shape function directly in term of x and y. Therefore parent element is transformed from global co-ordinate system (x, y) to local co-ordinate system (ξ, η) as the reference square element of side 2 units as shown in figure 1 (right side The interpolation function for x is obtained in terms of ξ and η as shown in equation 1.

$$x = \alpha_1 + \alpha_2\xi + \alpha_3\eta + \alpha_4\xi\eta + \alpha_5\xi^2 + \alpha_6\eta^2 + \alpha_7\xi^2\eta + \alpha_7\xi\eta^2 \quad (1)$$

Above equation can be written into matrix form as shown in equation 2

$$x = \begin{bmatrix} 1 & \xi & \eta & \xi\eta & \xi^2 & \eta^2 & \xi^2\eta & \xi\eta^2 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \end{bmatrix} \quad (2)$$

Interpolation function at nodal values x1, x2, x3, x4, x5, x6, x7 and x8 are calculate with respect to local co-ordinates (ξ, η) that are (−1, −1), (0, −1), (1, −1), (1, 0), (1, 1), (0, 1), (-1, -1), and (-1, 0) respectively at nodes 1, 2, 3, 5, 6, 7 and 8. The eight simultaneous equation form are represented below

$$\begin{aligned}
x1 &= \alpha1 - \alpha2 - \alpha3 + \alpha4 + \alpha5 + \alpha6 - \alpha7 - \alpha8 \\
x2 &= \alpha1 - \alpha3 + \alpha6 \\
x3 &= \alpha1 + \alpha2 - \alpha3 - \alpha4 + \alpha5 + \alpha6 - \alpha7 + \alpha8 \\
x4 &= \alpha1 + \alpha2 + \alpha5 \\
x5 &= \alpha1 + \alpha2 + \alpha3 + \alpha4 + \alpha5 + \alpha6 + \alpha7 + \alpha8 \\
x6 &= \alpha1 + \alpha3 + \alpha6 \\
x7 &= \alpha1 - \alpha2 + \alpha3 - \alpha4 + \alpha5 + \alpha6 + \alpha7 - \alpha8 \\
x8 &= \alpha1 - \alpha2 + \alpha5
\end{aligned} \quad (3)$$

Equation 3 can be rewritten into matrix form

$$\begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{Bmatrix} = \begin{pmatrix} 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 \\ 1 & 0 & -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 \\ 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \end{Bmatrix} \quad (4)$$

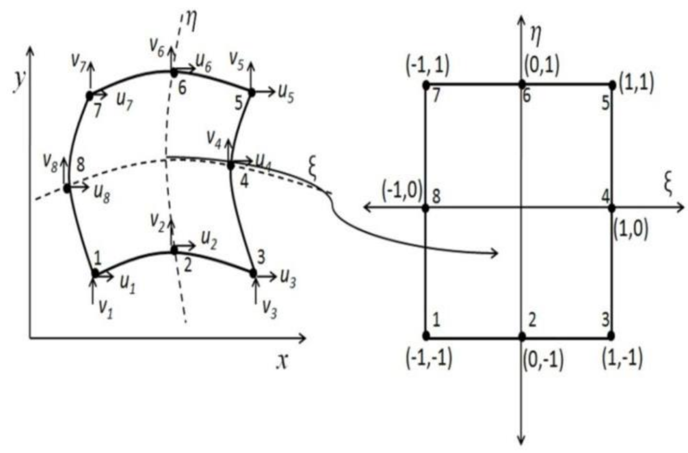

Fig. 1: Geometrical transformation of 8-node element

Corresponding Author,
E-mail address: naman.jain@miet.ac.in; namanjainyati@gmail.com
Phone No-+91-9411260409

Equation 3 can be rewritten into matrix form given eq (4)-

$$\begin{Bmatrix} x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\\ x_7\\ x_8 \end{Bmatrix} = \begin{pmatrix} 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1\\ 1 & 0 & -1 & 0 & 0 & 1 & 0 & 0\\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & 1\\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0\\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0\\ 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1\\ 1 & -1 & 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{Bmatrix} \alpha_1\\ \alpha_2\\ \alpha_3\\ \alpha_4\\ \alpha_5\\ \alpha_6\\ \alpha_7\\ \alpha_8 \end{Bmatrix} \quad (4)$$

Compact form of equation 4 can be written as

$$\{X\} = [A]\{\alpha\} \quad (5)$$

$$\{\alpha\} = [A]^{-1}\{X\} \quad (6)$$

By solving the system of equation 4 value of the parameters αi can be obtained. Hence, the inverse of the matrix [A] is obtained as-

$$[A]^{-1} = 1/4 \begin{pmatrix} -1 & 2 & -1 & 2 & -1 & 2 & -1 & 2\\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & -2\\ 0 & -2 & 0 & 0 & 0 & 2 & 0 & 0\\ 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0\\ 1 & -2 & 1 & 0 & 1 & -2 & 1 & 0\\ 1 & 0 & 1 & -2 & 1 & 0 & 1 & -2\\ -1 & 2 & -1 & 0 & 1 & -2 & 1 & 0\\ -1 & 0 & 1 & -2 & 1 & 0 & -1 & 2 \end{pmatrix} \quad (7)$$

Substituting the value to [A]-1 in equation 6

$$\begin{Bmatrix} \alpha_1\\ \alpha_2\\ \alpha_3\\ \alpha_4\\ \alpha_5\\ \alpha_6\\ \alpha_7\\ \alpha_8 \end{Bmatrix} = 1/4 \begin{pmatrix} -1 & 2 & -1 & 2 & -1 & 2 & -1 & 2\\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & -2\\ 0 & -2 & 0 & 0 & 0 & 2 & 0 & 0\\ 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0\\ 1 & -2 & 1 & 0 & 1 & -2 & 1 & 0\\ 1 & 0 & 1 & -2 & 1 & 0 & 1 & -2\\ -1 & 2 & -1 & 0 & 1 & -2 & 1 & 0\\ -1 & 0 & 1 & -2 & 1 & 0 & -1 & 2 \end{pmatrix} \begin{Bmatrix} x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\\ x_7\\ x_8 \end{Bmatrix} \quad (8)$$

Substituting the value of parameters αi in equation (2)

$$x(\xi,\eta) = 1/4[1 \;\; \xi \;\; \eta \;\; \xi\eta \;\; \xi^2 \;\; \eta^2 \;\; \xi^2\eta \;\; \xi\eta^2] \begin{pmatrix} -1 & 2 & -1 & 2 & -1 & 2 & -1 & 2\\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & -2\\ 0 & -2 & 0 & 0 & 0 & 2 & 0 & 0\\ 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0\\ 1 & -2 & 1 & 0 & 1 & -2 & 1 & 0\\ 1 & 0 & 1 & -2 & 1 & 0 & 1 & -2\\ -1 & 2 & -1 & 0 & 1 & -2 & 1 & 0\\ -1 & 0 & 1 & -2 & 1 & 0 & -1 & 2 \end{pmatrix} \begin{Bmatrix} x_1\\ x_2\\ x_3\\ x_4\\ x_5\\ x_6\\ x_7\\ x_8 \end{Bmatrix} \quad (9)$$

Expanding and rearranging equation (9) leads to

$x(\xi, \eta) = N1(\xi, \eta)x_1 + N2(\xi, \eta)x_2 + N3(\xi, \eta)x_3 + N4(\xi, \eta)x_4 + N5(\xi, \eta)x_5 + N6(\xi, \eta)x_6 + N7(\xi, \eta)x_7 + N8(\xi, \eta)x_8$ (10)

$$\begin{Bmatrix} N_1(\xi,\eta)\\ N_2(\xi,\eta)\\ N_3(\xi,\eta)\\ N_4(\xi,\eta)\\ N_5(\xi,\eta)\\ N_6(\xi,\eta)\\ N_7(\xi,\eta)\\ N_8(\xi,\eta) \end{Bmatrix} = \begin{Bmatrix} -0.25(1-\xi)(1-\eta)(1+\xi+\eta)\\ 0.50(1-\xi^2)(1-\eta)\\ -0.25(1+\xi)(1-\eta)(1-\xi+\eta)\\ 0.50(1+\xi)(1-\eta^2)\\ -0.25(1+\xi)(1+\eta)(1-\xi-\eta)\\ 0.50(1-\xi^2)(1+\eta)\\ -0.25(1-\xi)(1+\eta)(1+\xi-\eta)\\ 0.50(1-\xi)(1-\eta^2) \end{Bmatrix} \quad (11)$$

For the variable y same process will be followed

$y(\xi, \eta) = N1(\xi, \eta)y_1 + N2(\xi, \eta)y_2 + N3(\xi, \eta)y_3 + N4(\xi, \eta)y_4 + N5(\xi, \eta)y_5 + N6(\xi, \eta)y_6 + N7(\xi, \eta)y_7 + N8(\xi, \eta)y_8$

Geometric transformation of isoparametric 8-node element is presented in expressions (10) and (12) represent with center is given by (ξ, η) = (0, 0).

2. Finite Element Formulations for Plane Stress Problems

1. Plane stress is the state of two dimensional stresses in which stress in the third principle direction is zero i.e. $\sigma_{xx} = 0$. Material can have movement only in directions x and y therefore, u(x, y) and v(x,y) displacement variables play the role. The stress–strain relationships for plane stress [11-16] given as

$$\begin{Bmatrix} \sigma_{xx}\\ \sigma_{yy}\\ \sigma_{xy} \end{Bmatrix} = \frac{E}{1-\mu^2}\begin{bmatrix} 1 & \mu & 0\\ \mu & 1 & 0\\ 0 & 0 & \frac{(1-\mu)}{2} \end{bmatrix}\begin{Bmatrix} \varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{xy} \end{Bmatrix} \quad (13)$$

$$\{\sigma\} = [D]\{\varepsilon\} \quad (14)$$

Where,

$$\left.\begin{aligned} \varepsilon_{xx} &= \frac{\partial u}{\partial x}\\ \varepsilon_{yy} &= \frac{\partial v}{\partial y} \end{aligned}\right\} \quad (15)$$

$$\varepsilon_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$$

Matrix form of above equations is presented as

$$\begin{Bmatrix} \varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{xy} \end{Bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} & 0\\ 0 & \frac{\partial}{\partial y}\\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}\begin{Bmatrix} u\\ v \end{Bmatrix} \quad (16)$$

More simplified form as

$$\{\varepsilon\} = [L]\{U\} \quad (17)$$

Where, $[L]$ is a linear differential operator and $\{U\}$ nodal displacement vector.

2.1 Displacement Field

For 8-node element the interpolation and shape function of displacement field vector (*u* and *v*) is same as that for *x* variable and represented as

$u = N_1u_1 + N_2u_2 + N_3u_3 + N_4u_4 + N_5u_5 + N_6u_6 + N_7u_7 + N_8u_8$ (18)

$v = N_1v_1 + N_2v_2 + N_3v_3 + N_2v_2 + N_2v_2 + N_2v_2 + N_2v_2 + N_nv_n$ (19)

Representing equation 18 and 19 into matrix form

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 & N_5 & 0 & N_6 & 0 & N_7 & 0 & N_8 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 & N_5 & 0 & N_6 & 0 & N_7 & 0 & N_8 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \\ u_5 \\ v_5 \\ u_6 \\ v_6 \\ u_7 \\ v_7 \\ u_8 \\ v_8 \end{Bmatrix} \quad (20)$$

In simplified form as

$\{U\} = [N]\{a\}$ (21)

Where,

$\{a\} = \{u_1, v_1, u_2, v_2, \ldots, u_8, v_8\}$ is the nodal displacements vector, $\{N\}$ is the shape functions depend on the element used.

Substituting for $\{U\}$ using equation (17), the strain displacement equation become

$\{\varepsilon\} = [B]\{a\}$ (22)

Where,

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \cdot & \cdot & \cdot & \frac{\partial N_n}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & \cdot & \cdot & \cdot & 0 & \frac{\partial N_n}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \cdot & \cdot & \cdot & \frac{\partial N_n}{\partial y} & \frac{\partial N_n}{\partial x} \end{bmatrix} \quad (23)$$

The matrix $[B]$ is called the strain matrix; it relates the nodal displacements to the strains. It is formed by the partial derivatives of the shape functions $N_i(x, y)$.

2.2 Strain Matrix

From equation (20), when substituting in equation (22) the strain vector matrix [B] is obtained and given as

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 & \frac{\partial N_5}{\partial x} & 0 & \frac{\partial N_6}{\partial x} & 0 & \frac{\partial N_7}{\partial x} & 0 & \frac{\partial N_8}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial y} & 0 & \frac{\partial N_5}{\partial y} & 0 & \frac{\partial N_6}{\partial y} & 0 & \frac{\partial N_7}{\partial y} & 0 & \frac{\partial N_8}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} & \frac{\partial N_5}{\partial y} & \frac{\partial N_5}{\partial x} & \frac{\partial N_6}{\partial y} & \frac{\partial N_6}{\partial x} & \frac{\partial N_7}{\partial y} & \frac{\partial N_7}{\partial x} & \frac{\partial N_8}{\partial y} & \frac{\partial N_8}{\partial x} \end{bmatrix} \quad (23)$$

Subsequently, we will need to express the derivatives $\frac{\partial x}{\partial \xi}, \frac{\partial y}{\partial \xi}, \frac{\partial x}{\partial \eta}$ & $\frac{\partial y}{\partial \eta}$. This can be done in following ways

$$\frac{\partial N}{\partial \xi} = \frac{\partial N}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N}{\partial y}\frac{\partial y}{\partial \xi} \quad (25)$$

$$\frac{\partial N}{\partial \eta} = \frac{\partial N}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial N}{\partial y}\frac{\partial y}{\partial \eta} \quad (26)$$

Quadrilateral 8-node element is isoparametric, therefore the shape functions $N_i(\xi, \eta)$ also define the geometrical transformation [17-19] between the reference and the parent element. The coordinates x and y of any point of the parent element are given as –

$x = N_1x_1 + N_2x_2 + N_3x_3 + N_4x_4 + N_5x_5 + N_6x_6 + N_7x_7 + N_8x_8$ (27)

$y = N_1y_1 + N_2y_2 + N_3y_3 + N_4y_4 + N_5y_5 + N_6y_6 + N_7y_7 + N_8y_8$ (28)

From equation (25) & (26) we get

$$\begin{Bmatrix} \frac{\partial N}{\partial \xi} \\ \frac{\partial N}{\partial \eta} \end{Bmatrix} = J \begin{Bmatrix} \frac{\partial N}{\partial x} \\ \frac{\partial N}{\partial y} \end{Bmatrix} \quad (29)$$

Where J is Jacobian transformation matrix which is given as

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{4} \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^{4} \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^{4} \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^{4} \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix} \quad (30)$$

After deriving and rearranging, the Jacobian is written in the form of a product of two matrices:

$$[J] = 1/4 \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi} & \frac{\partial N_5}{\partial \xi} & \frac{\partial N_6}{\partial \xi} & \frac{\partial N_7}{\partial \xi} & \frac{\partial N_8}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} & \frac{\partial N_5}{\partial \eta} & \frac{\partial N_6}{\partial \eta} & \frac{\partial N_7}{\partial \eta} & \frac{\partial N_8}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} \quad (31)$$

2.3 Stiffness Matrix

The stiffness matrix for the quadrilateral element can be derived from the strain energy [20-21] in the body, given by

$U = \frac{1}{2}\int \sigma^T \varepsilon dv$ (32)

$U = \sum \frac{1}{2} t_e \int \sigma^T \varepsilon dA$ (33)

$\{\sigma\} = [D][B]\{a\}$ (34)

$\{\varepsilon\} = [B]\{a\}$ (35)

Where t_e is the thickness of element e.

$U = \sum \frac{1}{2} t_e \int_{-1}^{1}\int_{-1}^{1} \{a\}[B]^T[D][B]\{a\}dxdy$ (36)

$U = \sum \frac{1}{2} t_e \int_{-1}^{1}\int_{-1}^{1} \{a\}[B]^T[D][B]\{a\}detJd\xi d\eta$ (37)

$= \frac{1}{2}\sum a^T ka$ (38)

Where

$k = t_e \int_{-1}^{1}\int_{-1}^{1} [B]^T[D][B]detJd\xi d\eta$ (39)

Which is the element stiffness matrix of dimension (16x16). Complete developed MATLAB code is presented in Appendix.

3 Implementation of MATLAB Code

The novelty of the developed 8-node element matlab code is been validated with the ansys software results for cantilever beam. Cantilever beam is considered as linear isotropic structure subjected to the state of plain stress having 32 mm in length, 20 mm in width and thickness of unity for static loading condition. Intel core i5, 2.5 ghz processor with 4 gb ddr3 ram is used to perform the numerical analysis and matlab 64-bit r2010b is used to developed and execute the code. to reduce the calculation time young's modulus of beam is taken as 1 N/m2 and poisson's ratio 0.3. result obtained with matlab code is presented in figure 2 whereas ansys and matlab code have same value of maximum stress and displacement.

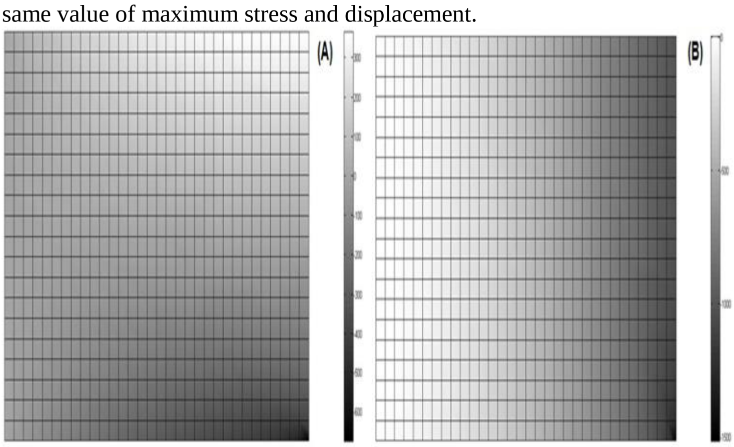

Fig. 2: Result of Cantilever beam (A) displacement in x-direction and (B) Displacement in y-direction

4. Conclusions

In present investigation a mathematical approach has been shown to developed MATLAB code for finite element analysis of isotropic 2-D structure. In present work MATLAB code has been developed for structure analysis of Cantilever beam and meshing of the beam has been done with 8-node quadrilateral element. The developed code is universally adopted for different analysis purpose such as static, fatigue, thermal, vibration, topological optimization etc. The purpose of the work is to replace 4-node quadrilateral element with 8-node quadrilateral element to overcome the problems of meshing of complex structure, geometric discretization error and accuracy in result. Developed MATLAB code is been validate with ANSYS results. With minor changes this code can used for any structural analysis.

APPENDIX: MATLAB code for cantilever beam under static loading meshing by 8-node elements

```
NXE=32;   % number of elements in the horizontal  direction
NYE=20;   % number of elements in the  vertical direction

% Nodal connectivity and coordinate calculation

X_origin=NXE;
Y_origin=NYE;
nnd = 0;
k = 0;
for i = 1:NXE
for j=1:NYE
k = k + 1;
n1 = (i-1)*(3*NYE+2)+2*j - 1;
n2 = i*(3*NYE+2)+j - NYE - 1;
n3 = i*(3*NYE+2)+2*j-1;
n4 = n3 + 1;
n5 = n3 + 2;
n6 = n2 + 1;
n7 = n1 + 2;
n8 = n1 + 1;
Coordinate(n1,:) = [(i-1) Y_origin-(j-1)];
Coordinate(n3,:) = [i Y_origin-(j-1)];
Coordinate(n2,:) = [(Coordinate(n1,1)+Coordinate(n3,1))/2 (Coordinate(n1,2)+Coordinate(n3,2))/2];
Coordinate(n5,:) = [i Y_origin-j];
Coordinate(n4,:) = [(Coordinate(n3,1)+ Coordinate(n5,1))/2 (Coordinate(n3,2)+ Coordinate(n5,2))/2];
Coordinate(n7,:) = [(i-1) Y_origin-j];
Coordinate(n6,:) = [(Coordinate(n5,1)+ Coordinate(n7,1))/2 (Coordinate(n5,2)+ Coordinate(n7,2))/2];
Coordinate(n8,:) = [(Coordinate(n1,1)+ Coordinate(n7,1))/2 (Coordinate(n1,2)+ Coordinate(n7,2))/2];
nel = k;
nnd = n5;
node(k,:) = [n1 n2 n3 n4 n5 n6 n7 n8];
end
end

% Input data
nel = length(node);            % number of elements
nnel = 8;                      % number of nodes per element
ndof = 2;                         % number of dofs per node (UX,UY)
nnode = length(Coordinate) ;% total number of nodes in system
sdof = nnode*ndof;            % total system dofs
edof = nnel*ndof;             % degrees of freedom per element
stiffness = zeros(sdof,sdof);   % system stiffness matrix
displacement = zeros(sdof,1);
force = zeros(sdof,1);
E =2.1E10;                      % Young’s modulus
nu = 0.3;                          % Poisson's ratio

% Computation of element stiffness matrices and their assembly
D = E/(1-nu^2)*[1 nu 0 ; nu 1 0; 0 0 (1-nu)/2] ;                %
Constituent Matrix for Plane stress
Gausspointx=[-1 1 1 -1]/sqrt(3);
Gausspointy=[1 1 -1 -1]/sqrt(3);
Gaussweight=[1 1 1 1 1 1 1 1];
for iel=1:nel           % loop for the total number of elements
for i=1:nnel
nd(i)=node(iel,i); % extract connected node for (iel)-th element
xx(i)=Coordinate(nd(i),1);                         % extract x
value of the node
yy(i)=Coordinate(nd(i),2);                         % extract y
value of the node
end
K = zeros(edof,edof);       % initialization of stiffness matrix
for int=1:4
xi = Gausspointx(int);
            % sampling point in x-axis
wtx = Gaussweight(int);
            % weight in x-axis
eta = Gausspointy(int);
             % sampling point in y-axis
wty = Gaussweight(int) ;     % weight in y-axis

DU1=0.25*(1.-eta)*(2.*xi+eta); DU2=-1.*(1.-eta)*xi;
DU3=0.25*(1.-eta)*(2.*xi-eta); DU4=0.5*(1-eta^2);
DU5=0.25*(1.+eta)*(2.*xi+eta); DU6=-1.*(1.+eta)*xi;
DU7=0.25*(1.+eta)*(2.*xi-eta); DU8=-0.5*(1.-eta^2);
DV1=0.25*(1.-xi)*(2.*eta+xi); DV2=-0.5*(1. - xi^2);
DV3=-0.25*(1.+xi)*(xi-2.*eta); DV4=-1.*(1.+xi)*eta;
DV5=0.25*(1.+xi)*(xi+2.*eta); DV6=0.5*(1.-xi^2);
DV7=-0.25*(1.-xi)*(xi-2.*eta); DV8=-1.*(1.-xi)*eta;
j=[DU1 DU2 DU3 DU4 DU5 DU6 DU7 DU8;
DV1 DV2 DV3 DV4 DV5 DV6 DV7 DV8];
J=j*[xx;yy]';
jac1=inv(J);
deriv=jac1*j;
for m=1:nnel
k=2*m;
l=k-1;
x=deriv(1,m);
bee(1,l)=x;
bee(3,k)=x;
y=deriv(2,m);
bee(2,k)=y;
bee(3,l)=y;
end
K=K + det(J)*bee'*D*bee;
end
EDOF = [2*nd(1)-1; 2*nd(1); 2*nd(2)-1; 2*nd(2); 2*nd(3)-1;
2*nd(3); 2*nd(4)-1; 2*nd(4);
2*nd(5)-1; 2*nd(5); 2*nd(6)-1; 2*nd(6); 2*nd(7)-1; 2*nd(7);
2*nd(8)-1; 2*nd(8)];
stiffness(EDOF,EDOF)=stiffness(EDOF,EDOF)+K;
end

% Boundary condition and calculation of displacement
force(4050,1) = -50;
fixeddofs   = [1:1:2*41];
alldofs     = [1:4050];
freedofs    = setdiff(alldofs,fixeddofs);
displacement(freedofs,:)  =  stiffness(freedofs,freedofs)  \
force(freedofs,:);
displacement(fixeddofs,:)= 0;
disp('The maximum displacement UX')
max(abs(UX))
disp('The maximum displacement UY')
max(abs(UY))
```

```
%Display of result
for iel=1:nel
for i=1:nnel
nd(i)=node(iel,i);
X(i,iel)=Coordinate(nd(i),1);
Y(i,iel)=Coordinate(nd(i),2);
end
profile1(:,iel) = -UX(nd') ;
profile2(:,iel) = -UY(nd') ;
end
f3 = figure ;
set(f3,'name','Postprocessing','numbertitle','off') ;
plot(X,Y,'k')
fill(X,Y,profile1)
colormap(gray);
axis off ;
cbar = colorbar;
set(figure,'name','Postprocessing','numbertitle','off') ;
plot(X,Y,'k')
fill(X,Y,profile2)
colormap(gray);
axis off ;
cbar = colorbar;
% For Messerschmitt-Bolkow-Beam replaces Boundary condition and calculation following line
force(1682) =1;
fixeddofs   = [41 42 3402];
alldofs     = [1:2*(nely+1)*(nelx+1)];
freedofs    = setdiff(alldofs,fixeddofs);
displacement(freedofs,:) = stiffness(freedofs,freedofs) \ force(freedofs,:);
displacement(fixeddofs,:)= 0;
disp('The maximum displacement UX')
max(abs(UX))
disp('The maximum displacement UY')
max(abs(UY))
```

References

[1] TR Chandrupatla, AD Belegunda. Introduction to finite element in engineering. Prentice hall upper Inc, 2004.

[2] DV Hutton. Fundamental of finite element analysis. The McGraw-Hill companies, 2004.

[3] A Khennane. Introduction to finite element analysis using MATLAB and Abaqus, Taylor & Francis Group, LLC, 2013.

[4] Y Ko, KJ Bathe. A new 8-node element for analysis of three dimensional solids, Computers and Structures 202, 2018, 85-104.

[5] N Jain. MATLAB code for structure analysis of 2-D structure subjected to static and self-weight loading conditions, International Journal of Trend in Research and Development, 48 (4) 2017, 78-94.

[6] C Wang, Y Wang, C Yang, X Zhang, P Hu. 8-node and 12 node plane element based on assumed stress quasi-conforming method immune to distorted mesh, Engineering Computations, 34(4), 2017.

[7] N Jain. Effect of higher order element on numerical instability in topological optimization of linear static loading structure, Journal of theoretical and applied mechanics, 48 (4), 2018, 78-94.

[8] BG Raju, Ppoluraju. Numerical study of bubble deck slab using ansys,7: 6C2, 2019.

[9] IM Smith, DV Griffiths. Programming the Finite Element Method, 2nd edn. Wiley, Chichester, U.K, 1988.

[10] OC Zienkiewicz. The Finite Element Method, 3rd edn. McGraw-Hill, York, London, 1977.

[11] SM Lepi. Practical guide to finite elements: a solid mechanics approach, Marcel Dekker, 1977.

[12] S Timoshenko, J Goodier. Theory of Elasticity, 3rd edn. McGraw-Hill, New York, 1970.

[13] S Timoshenko, S Woinowsky-Krieger. Theory of Plates and Shells. McGraw-Hill, New York, 1959.

[14] RD Cook. Finite Element Modeling for Stress Analysis, Wiley, New York, 1995.

[15] GE Mase. Schaum's Outline Series: Theory and Problems of Continuum Mechanics. McGraw-Hill, New York, 1970.

[16] AS Saada. Elasticity: Theory and Applications, 2nd edn. Krieger Publishing, Melbourne, FL, 1993.

[17] YW Kwon, H Bang. The Finite Element Method Using Matlab, 2nd edn. CRC Press, London, U.K, 2000.

[18] DL Logan. A First Course in the Finite Element Method Using Algor, 2nd edn. Brooks/Cole Thompson Learning, Pacific Groove, CA, 2001.

[19] JR Reddy. An Introduction to the Finite Element Method, 3rd edn. McGraw-Hill, New York, 2006.

[20] M Mcguire, GH Gallagher, RD Ziemian. Matrix Structural Analysis, 2nd edn. Wiley, New York, 2000.

[21] JL Meek. Computer Methods in Structural Analysis. E & FN SPON, London, U.K, 1991.