A New Approach to Automatic Mesh Generation Over Polygonal Domains Using All Quadrilaterals of Quintic Order to Decic Order Finite Elements of Complete Lagrange Family

¹H.T.Rathod^{*}, ²K.V.Vijayakumar, ³Bharath Rathod

¹Department of Mathematics, Jnana Bharathi Campus, Bangalore University, Bangalore -560056, Karnataka state, India.

²Department of Mathematics, B.M.S.Institute of Technology and Management, Avalahalli, Bangalore-

560064, Karnataka State, India.

³PGDFM student IIFM Bhopal, MP state INDIA

Abstract

This paper presents a novel mesh generation scheme of all quadrilateral elements over a linear polygonal domain. We first decompose the linear polygon into simple sub regions in the shape of quadrilaterals. These simple regions are then quadrangulated to generate first into a fine mesh of four node quadrilateral elements using bilinear transformations. We have already proposed this automatic quadrilateral mesh generation scheme in our recent paper[31]. In this scheme each four node quadrilateral is converted to higher order quadrilaterals by inserting the midside nodes appropriately. Examples were presented to illustrate the simplicity and efficiency of the new mesh generation method for standard and arbitrary shaped domains for linear to quadrilaterals for Complete Lagrange family having 24,32,40,49,60 and 72 nodes. They are actually the Serendipity famiy quadrilaterals with appropriate number of interior nodes to incorporate the complete monomial basis of 5th to 10th degree polynomials in bivariates

We have appended two important MATLAB programs which incorporate the mesh generation scheme for the 72-noded decic order complete Lagrange quadrilateral elements developed in this paper.Other MATLAB programs can be coded on similar lines. These programs provide valuable output on the nodal coordinates ,element connectivity and graphic display of the all quadrilateral meshes for application to finite element analysis.The typical domains include rectangles,arbitrary oriented rectangles,an equilateral triangle, arbitrary quadrilateral,convex and nonconvex polygons.

Keywords: Finite elements of Serendipity and Complete Lagrange families, quintic to decic order quadrilateral mesh generations, convex and nonconvex polygonal domains, uniform refinement, quadrangulation and triangulation.

1. Introduction

In our recent paper[31] we have already given details about the mesh generation process and its importance in finite element analysis. Hence, we are not repeating this topic. We are now continuing our presentation further directly to generate meshes for remaining higher order elements from quintic order to decic order

In this paper, we present a novel mesh generation scheme of all quadrilateral elements for linear polygonal domains. We have explained the scheme with more clarity and precision. This scheme converts the elements in background quadrilateral into quadrilaterals using bilinear transformation We first decompose the linear polygon into simple subregions in the shape of quadrilaterals. These simple subregions are then quadrangulated by using the one to one mapping concept between the original quadrilateral and the square. We propose then an automatic quadrilateral conversion scheme in which each

background quadrilateral with n by m divisions into nm quadrilaterals according to the bilinear mapping scheme. Further, to preserve the mesh conformity a similar procedure is also applied to every quadrilateral of the domain and this fully discretizes the given linear polygonal domain into all quadrilaterals, thus propogating uniform refinement and quadrangulation. In section-2 of this paper, we explain the particular degenerate transformations required in generating the forms of the quadrilateral: rectangles, parallelograms, trapeziums etc shapes, In section-3 of this paper, we present a scheme to discretize the arbitrary quadrilateral into a fine mesh of quadrilateral elements. In section- 4, we explain the procedure to create higher order quadrilateral by inserting midside nodes in each four node element. In section-5, we have presented a method of piecing together of all quadrilateral subregions and eventually creating an all quadrilateral mesh for the given linear polygonal domain. In section-6, we present several examples to illustrate the simplicity and efficiency of the proposed mesh generation method for triangles, rectangles, arbitrary quadrilaterals, convex and non convex polygonal domains.

2. Linear Convex Quadrilateral and Isoparametric Coordinate Transformation with global vertices Let us first consider an arbitrary four noded linear convex quadrilateral element Q_e in the Cartesian space (x, y) with global vertices (x_{n_k}, y_{n_k}) which is mapped into a 2-square in the local parametric space (ξ, η) , where $(n_k, k = 1,2,3,4)$ are global nodenumbers. This is shown in the following figures Fig.1a and Fig.1b



Fig.1a Linear: convex quadrilateral Q_e in (x,y) space, Fig.1b: Standard 2-square in in (ξ,η) space The Isoparametric coordinate transformation from (x,y) space to the (ξ,η) space is given by $\begin{pmatrix} x^e \\ y^e \end{pmatrix} = \sum_{k=1}^{4} \frac{1}{4} (1 + \xi\xi_k) (1 + \eta\eta_k) \begin{pmatrix} x_{n_k}^e \\ y_{n_k}^e \end{pmatrix}$(1)

Where,

 $((x_{n_k}^e, y_{n_k}^e), k = 1,2,3,4)$ are the vertices of the linear convex quadrilateral element ' Q_e ' in the Cartesian space (x, y)

and $(n_k(\xi_k, \eta_k), k = 1, 2, 3, 4) = \{n_1(-1, -1), n_2(1, -1), n_3(1, 1), n_4(-1, 1)\}$ are the vertices of the 2-square in (ξ, η) space.

For the sake of simplicity and without loss of generality, we may assume that $n_1 = 1, n_2 = 2$,

 $n_3 = 3$, $n_4 = 4$ to demonstrate the various quadrilateral shapes described by eqn(1).

From Eqn(1), we obtain

$$\begin{pmatrix} x^{e} \\ y^{e} \end{pmatrix} = \begin{pmatrix} a_{0}^{e} + a_{1}^{e}\xi + a_{2}^{e}\eta + a_{3}^{e}\xi\eta \\ b_{0}^{e} + b_{1}^{e}\xi + b_{2}^{e}\eta + b_{3}^{e}\xi\eta \end{pmatrix}$$
 -----(2) where

$$\begin{pmatrix} a_0^e \\ b_0^e \end{pmatrix} = \begin{pmatrix} \frac{1}{4}(x_1^e + x_2^e + x_3^e + x_4^e) \\ \frac{1}{4}(y_1^e + y_2^e + y_3^e + y_4^e) \end{pmatrix}$$
------(3a)
$$\begin{pmatrix} a_1^e \\ b_1^e \end{pmatrix} = \begin{pmatrix} \frac{1}{4}(-x_1^e + x_2^e + x_3^e - x_4^e) \\ \frac{1}{4}(-y_1^e + y_2^e + y_3^e - y_4^e) \end{pmatrix}$$
------(3b)
$$\begin{pmatrix} a_2^e \\ b_2^e \end{pmatrix} = \begin{pmatrix} \frac{1}{4}(-x_1^e - x_2^e + x_3^e + x_4^e) \\ \frac{1}{4}(-y_1^e - y_2^e + y_3^e + y_4^e) \end{pmatrix}$$
------(3c)
$$\begin{pmatrix} a_3^e \\ b_3^e \end{pmatrix} = \begin{pmatrix} \frac{1}{4}(x_1^e - x_2^e + x_3^e - x_4^e) \\ \frac{1}{4}(y_1^e - y_2^e + y_3^e - y_4^e) \end{pmatrix}$$
------(3d)

The nature of the constants $((a_i^e, b_i^e), i = 0, 1, 2, 3)$ will determine the element geometry. We have briefly listed some of these element geometries.

Rectangular elements

When $a_1^e = a^e$, $a_2^e = 0$, $a_3^e = 0$; $b_1^e = 0$, $b_2^e = b^e$, $b_3^e = 0$

and (a_0^e, b_0^e) as coordinate of the element centroids, we can generate rectangular elements whose sides are parallel to coordinate axes with half side length = a^e and half side width = b^e . This gives

 $x^e = a_0^e + a^e \xi, \quad y^e = b_0^e + b^e \eta$

Example 1:We consider a domain with vertices $\{1(0,0), (\sqrt{5}, 0), (\sqrt{5}, 2\sqrt{5}), (0, 2\sqrt{5})\}$ which is a rectangle of length= $2\sqrt{5}$ units and breadth= $\sqrt{5}$. It is quite easy to show that the transformation

 $x^e = \sqrt{5/4} (1+\xi)$, and $y^e = \sqrt{5} (1+\eta)$ maps the rectangle into a 2-square in (ξ, η) space

Parallelogram elements

(i)When
$$a_3^e = 0$$
, $b_1^e = 0$, $b_3^e = 0$, gives
 $x^e = a_0^e + a_1^e \xi + a_2^e \eta$, $y^e = b_0^e + b_2^e \eta$ ------(4b)

and this will generate a parallelogram whose two sides are parallel to y-axis.

(ii) When
$$a_2^e = 0$$
, $a_3^e = 0$, $b_3^e = 0$, this gives
 $x^e = a_0^e + a_1^e \xi$, $y^e = b_0^e + b_1^e \xi + b_2^e \eta$ ------(4c)

and this will generate parallelogram element whose two sides are parallel to x- axis

(iii)Arbitrary oriented rectangles and parallelograms are generated

When
$$x^e = a_0^e + a_1^e \xi + a_2^e \eta$$
, $y^e = b_0^e + b_1^e \xi + b_2^e \eta$ ------(4d)
with $a_3^e = 0$, $b_3^e = 0$

Example 2: We again consider a domain with vertices $\{1(0,0), (\sqrt{5}, 0), (\sqrt{5}, 2\sqrt{5}), (0, 2\sqrt{5})\}$ which is a rectangle of length= $2\sqrt{5}$ units and breadth= $\sqrt{5}$ and rotate this rectangle about the origin by an angle arctan(2) in anticlockwise direction we obtain the rectangle $\{(0,0), (1,2), (-3,4), (-4,2)\}$ and then use bilinear mapping to transform this to a 2-square are

$$x^{e} = -3/2 + \xi/2 - 2\eta$$
 and $y^{e} = 2 + \xi + \eta$

We have included the mesh generation for example 1 and example 2 in the application section. using higher order elements.Now as an illustration, we present here the mesh generation for rectangles of Examples 1-2 for the four node element. :





(iv)When all the parameters $((a_i^e, b_i^e), i = 0, 1, 2, 3)$ are non-zero, arbitrary quadrilaterals are generated and this is the general case and we have

 $x^e=a^e_0+a^e_1\xi+a^e_2\eta+a^e_3\xi\eta$

$$y^{e} = b_{0}^{e} + b_{1}^{e}\xi + b_{2}^{e}\eta + b_{3}^{e}\xi\eta \qquad -----(4e)$$

This case also covers the trapezium elements when either x^e is linear or y^e is linear and one of these say when either x^e or y^e is nonlinear. That is:

$$\begin{aligned} x^{e} &= a_{0}^{e} + a_{1}^{e}\xi + a_{2}^{e}\eta \\ y^{e} &= b_{0}^{e} + b_{1}^{e}\xi + b_{2}^{e}\eta + b_{3}^{e}\xi\eta \\ \text{and} \\ x^{e} &= a_{0}^{e} + a_{1}^{e}\xi + a_{2}^{e}\eta + a_{3}^{e}\xi\eta \end{aligned}$$
(4f)

$$e^{e} = b_0^e + b_1^e \xi + b_2^e \eta$$
 ------(4g)

This analysis as explained above covers all bilinear mappings for various shapes degenerated from the arbitrary linear convex quadrilateral.

3. Mesh Generation over a Linear Convex Quadrilaterals with global vertices

y

We can map an arbitrary quadrilateral Q_e with global vertices $((x_{n_i}^e, y_{n_i}^e), i = 1,2,3,4)$, where $(n_i = 1,2,3,4)$ are global node numbers in Cartesian space (x, y) into a unit square in the local space (u, v) with local node numbers $(n_i = 1,2,3,4)$ in the parametric space. The mapping is shown in figs.1a and 1c The unit square in uv-space is a convenient choice for division into smaller squares or rectangles



Fig.1a: a linear convex quadrilateral in (x,y) space, Fig.1c:standard 1-square in (u,v) space,

Let us consider the bilinear mapping of an arbitrary quadrilateral Q_e as shown in Fig.1a and Fig.1c with vertices ($(x_{n_i}^e, y_{n_i}^e), i = 1,2,3,4$), into a standard unit square. The above mapping is defined as

We divide the unit square into $m \times n$ rectangles by making m divisions along u-axis and n-divisions along v-axis and this division (u,v) space has a one to one correspondence with a similar division of quadrilateral Q_e in (x,y) space. We now display the above mapping which shows divisions of a quadrilateral Q_e in (x, y) space and the corresponding divisions of a unit square in (u, v) space in Fig.2a and Fig.2b:



Fig.2a: mxn divisions of an arbitrary quadrilateral Q_e in (x, y) space and a typical quadrilateral element 'e' in the interior

[e]:quadrilateral with nodal vertices $< n_1, n_2, n_3, n_4 >$,

 $n_1 = (j-1)(m+1)+i, n_2 = (j-1)(m+1)+i+1, n_3 = j(m+1)+i+1, n_4 = j(m+1)+i.$



Fig.2b: Division of a unit square into 'mn' four node rectangles, in (u,v) space [e]::element'e' [Re]:rectangle with nodal vertices $\langle n_1, n_2, n_3, n_4 \rangle$, $(n_i, i=1,2,3,4)$::node numbers of element vertices $((u_k,v_l), k=1,2,3,...,m+1), l = 1,2,3,...,n + 1)$):local coordinates of unit square $n_1 = (j-1)(m+1)+i, n_2 = (j-1)(m+1)+i+1, n_3 = j(m+1)+i+1, n_4 = j(m+1)+i.$

The vertices of corner nodes for the element (e) has coordinates (in anticlockwise sense) are

$$n_{1}(x^{e}(u_{i}, v_{i}), y^{e}(u_{i}, v_{i})), \quad n_{2}(x^{e}(u_{i+1}, v_{j}), y^{e}(u_{i+1}, v_{j}))$$

$$n_{3}(x^{e}(u_{i+1}, v_{i+1}), y^{e}(u_{i+1}, v_{i+1})); \quad n_{4}(x^{e}(u_{i}, v_{j+1}), y^{e}(u_{i}, v_{j+1})) \quad -------(6)$$
Where
$$n_{1}(u_{i}, v_{j}) = n_{1}((i-1)/m, \quad (j-1)/n),$$

$$n_{2}(u_{i+1}, v_{j}) = n_{2}(i/m, \quad (j-1)/n),$$

$$n_{3}(u_{i+1}, v_{j+1}) = n_{3}(i/m, \quad j/n),;$$

$$n_{4}(u_{i}, v_{j+1}) = n_{4}((i-1)/m, \quad j/n),$$
and the node numbers in both the spaces are
$$n_{1}(u_{i}, v_{j}) = n_{2}(u_{i}, v_{i}) = n_{2$$

 $n_{1} = j(m+1) + i ; n_{2} = j(m+1) + i + 1 ;$ $n_{3} = (j+1)(m+1) + i + 1 ; n_{4} = (j+1)(m+1) + i -----(8)$

All the element nodes and coordinates can be obtained by varying i and j, i = 1, 2, ..., (m + 1); j = 1, 2, ..., (n + 1) and naturally over any typical element $(i + 1) \le (m + 1)$ and $(j + 1) \le (n + 1)$.

We have shown the division of an arbitrary quadrilateral Q_e and a unit square in Fig. 2a and Fig. 2b. respectively. We divide each side of the quadrilateral and unit square (in Cartesian space(x,y) and natural space(u,v)) into m equal division along x and u axes and n equal divisions along y and v axes. This creates $(m+1)^*$ (n + 1) nodes. These nodes are numbered from base line l_{12} (letting l_{ij} as the line joining the vertex (x_i^e, y_i^e) and (x_j^e, y_j^e)) and move upwards upto the line l_{34} in quadrilateral Q_e ; now with respect to the unit square in Fig.2b, we move along the line v = 0 and upwards up to the line v = 1. The nodes along v=0 are 1, 2,...(m+1); and then on $v_1 = 1/n$ are $(m+2), (m+3), \dots, 2(m+1)$; etc and finally on v=1 are $n(m+1)+1, n(m+1)+2, \dots, (n+1)^*(m+1)$ and they are numbered layer by layer. This is shown in the

u=2/m u=(i-1)/m u=i/m u=0 u=1/m u=(m-1)/m u=1 2 3 1 i. i+1 (m+1) ⇒ v=0 m (m+3) (m+4)(m+i+1) (m+i+2)(2m+1)(m+2)2(m+1) 2(m+1)+1 (3m+2) 3(m+1) ⇒v=2/n (j-1)(m+1)+i+1 (j-1)(m+1)+i+2 (i-1)(m+1)+1 i(m+1) | ⇒v=(i-1)/n i(m+1)+1 j(m+1)+i+1 j(m+1)+i+2 (j+1)(m+1) ⇒v=j/m <u>rr</u> = (n-1)(m+1)+1 (n-1)(m+1)+i+1 (n-1)(m+1)+i+2 n(m+1) ⇒v=(n-1)/n n(m+1)+1 n(m+1)+i+1 n(m+1)+i+2 (n+1)(m+1) ⇒v=1

Fig 2c. Matrix rr of node numbers for the division of a unit square

In the present algorithm the four corners of quadrilateral Q_e as well the unit square are defined as $n_1 = 1$, $n_2 = m+1$, $n_3 = (m+1)(n+1)$ and $n_4 = n(m+1)+1$, where m is the number of divisions along x or u axis n is the number of divisions along y or v axis in both the spaces viz: (x,y) and (u,v)

We may further note that when the present algorithm is applied to a single quadrilateral (m=1,n=1) then the boundary nodes are $n_1 = 1$, $n_2 = 2$, $n_3 = 4$ and $n_4 = 3$. This will help us in the graphic display of quintic order to decic order complete Lagrange elements over the standard squares.

4. Mesh Generation Using Higher order Quadrilaterals

In finite element applications, we may have to generate higher order quadrilateral elements. They contain midside nodes. We can obtain quadratic elements by inserting additional nodes at the midpoints of the linear four node element boundaries which gives us Serendipity quadratic elements. In addition to this when a node is also inserted at the centroid of the quadrilateral, we obtain Lagrange quadratic elements.

We next consider cubic elements, they can be obtained by inserting nodes at the trisectional points of the four node element boundaries which gives us Serendipity cubic elements. In addition to this, if we insert nodes in the interior of the elements at trisectional points, we obtain the Lagrange cubic elements. Zienkiewicz[17] intended to define a Serendipity family so that polynomial completeness is realized with necessary minimum nodes and presented the few lower order elements viz.Linear,Quadratic and Cubic elements which have equal number of nodes along each side which are uniformly spaced. It is obvious that the Basis functions for Seredipity elements with nodes placed only along the edges cannot generate complete polynomials beyond cubic, for this reason,Zienkiewicz[17] has suggested a central node for the next Quartic member of this family, and remarks that progression to yet higher order members is difficult and requires some ingenuity. M.Okabe[29], H.T.Rathod and Sridevi. Kilari [30] determined the Basis functions of the Serendipity and Complete Lagrange family elements which allow uniform spacing of

nodes over the element domain for orders 4-10.We have already demonstrated the generation of finite element meshes over polygonal domains upto Quartic order for Serendipity, Complete Lagrange and Lagrange family elements[31]. In this paper we intend to generate finite element meshes for remaining orders viz quintic order to decic order complete Lagrange elements.The present algorithm can also be applied to Lagrange family elements of quintic order to decic order, this presentation is excluded here because they contain a lage number of interior nodes.

These rectangular elements for Complete Lagrange family from Quintic order to Decic order in the local parametric space are depicted in the following figures.

(1) QUINTIC ORDER COMPLETE LAGRANGE ELEMENT OVER A 2-SQUARE





(3) SEPTIC ORDER COMPLETE LAGRANGE ELEMENT OVER A 2-SQUARE



H.T.Rathod, IJECS Volume 09 Issue 10 October, 2020 Page No. 25253-25306



 $\xi - axis$

->



(5) NONIC ORDER COMPLETE LAGRANGE ELEMENT OVER A 2-SQUARE

 $\xi - axis$



(6) DECIC ORDER COMPLETE LAGRANGE ELEMENT OVER A 2-SQUARE

 $\xi - axis$

5. Mesh Generation Algorithm over a linear convex Quadrilateral

We briefly present here the mesh generation algorithm over a Quadrilateral in Cartesian space

(i)Divide the unit square into 'mn' rectangles or squares of uniform size, this is done by making m divisions of equal spacing along u-axis and v-axis. This requires '(m+1)*(n+1) nodes.

(ii)Use the bilinear transformation to obtain (m+1)*(n+1) Cartesian coordinates over the original quadrilateral Q_e in xy – space corresponding to the (m+1)*(n+1) nodes in uv-space. This generates 'mn' rectangles in local uv-space and 'mn' quadrilaterals in the Cartesian xy-space.

(iii)Now insert midside nodes using pigeon hole principle which is elaborated in the MATLAB CODE and then insert the interior nodes if any. This discritises the 'mn' quadrilaterals into higher order quadrilaterals as per the requirements. Please note that the insertion of midsidenodes has to be done on all the four sides of the quadrilateral 'e' in Q_{e} , where e=1,2,3,...mn.

(iv) Using bilinear mapping Cartesian coordinates for all the nodes can be computed

(v)The nodal coordinates and element nodal connectivity data is passed on to the main program.

Then the main program generates the desired mesh for the element type.

6. Quad angulation of an Arbitrary Polygon

Finite element applications to physical problem require mesh generation over polygonal domains. We divide this domain into a coarse mesh of triangles or quadrilaterals or both. Our aim now is to generate a mesh of all quadrilaterals. This is first done by generating quadrilateral meshes over each coarse shape (triangles or quadrilateral) and then piecing together, we obtain an all quadrilateral mesh for the polygonal domains. We have presented in our recent paper[31] the mesh generation of polygonal domains using the 4, 8, 9 12, 16,17,25 noded quadrilaterals elements of Serendipity, Lagrange and Complete Lagrange elements upto Quartic order. In the present paper, we have generated meshes from Quintic order to Decic order elements, the details on the monomial basis and the element geometry are presented in author's paper[30].







Figs.4-5 Piecing together of four quadrilaterals

7. Application Examples

In applications to boundary value problems, we may have to discretize an arbitrary polygonal domain using linear, quadratic, cubic and quartic finite elements. Our purpose is to have codes which automatically generates elements with linear convex quadrilaterals over the domain by assuming the input as coordinates of the vertices. We have choosen four typical examples:

(i)Rectangles described in section 2(Examples 1-2)

- (ii)An Arbitrary Quadrilateral
- (iii)An Equilateral Triangle
- (iv)A Convex Polygon
- (v)A Nonconvex Polygon

We may note that the rectangles and parallelograms of any orientation will be discritsed into finite element meshes of rectangles(or squares) and parallelograms. Two codes written in MATLAB programming and based on the proposed scheme of this paper to generate meshes using Decic order Complete Lagrange appended. elements with 72-nodes are Codes for Linear, Quadratic, Cubic,Quartic,Quintic,Sextic,Septic,Octic,Nonic order finite elements were developed on similar lines and the schemes explained in this paper but they are not included here. Several Figures on Finite Element mesh generation using Quintic to Decic order elements(i.e24,32,40,49,60,72-noded each) are presented immediately after References.

8 Conclusions

Zienkiewicz[17] intended to define a Serendipity family so that polynomial completeness is realized with necessary minimum nodes and presented the few lower order elements viz.Linear,Quadratic and Cubic elements which have equal number of nodes along each side which are uniformly spaced.It is obvious that the Basis functions for Seredipity elements with nodes placed only along the edges cannot generate complete polynomials beyond cubic,for this reason,Zienkiewicz[17] has suggested a central node for the next Quartic member of this family, and remarks that progression to yet higher order members is difficult and requires some ingenuity

This paper presents a novel mesh generation scheme of all quadrilateral elements over a linear polygonal domain. We first decompose the linear polygon into simple sub regions in the shape of quadrilaterals. These simple regions are then quadrangulated to generate first into a fine mesh of four node quadrilateral elements using bilinear transformations. We have already proposed this automatic quadrilateral mesh generation scheme in our recent paper. In this scheme each four node quadrilateral is converted to higher order quadrilaterals by inserting the midside nodes appropriately. Examples were presented to illustrate the simplicity and efficiency of the new mesh generation method for standard and arbitrary shaped domains for linear to quadrilaterals for Complete Lagrange family having 24,32,40,49,60 and 72 nodes. They are actually the Serendipity famiy quadrilaterals with appropriate number of interior nodes to incorporate the complete monomial basis of quintic to decic degree polynomials in bivariates

We have appended two important MATLAB programs which incorporate the mesh generation scheme for the 72-noded decic order complete Lagrange quadrilateral elements developed in this paper see references [29,30]. Other MATLAB programs for lower order elements can be coded on similar lines. These programs provide valuable output on the nodal coordinates ,element connectivity and graphic display of the all quadrilateral meshes for application to finite element analysis. The typical domains include rectangles, arbitrary oriented rectangles, an equilateral triangle, arbitrary quadrilateral, convex and nonconvex polygons. These programs provide valuable output on the nodal coordinates for application to finite element analysis.

References

- [1.] R. Courant, Variational methods for the solution of problems of equilibrium and vibration, Bulletin of the American Mathematical Society, 49:1-23(1943).
- [2.] J.H. Argyris, Matrix displacement analysis of anisotropic shells by triangular elements, Journal of Royal Aeronautical Society 69:801 (1965).
- [3.] R.W. Clough, The finite element method in plane stress analysis, Proceedings of the 2nd ASCE Conference in Electronic Computation, Pittsburgh, PA 1960.
- [4.] Zienkiewicz. O. C, Philips. D. V, An automatic mesh generation scheme for plane and curved surface by isoparametric coordinates, Int. J.Numer. Meth.Eng, 3, 519-528 (1971).
- [5.] Gordan. W. J and Hall. C. A, Construction of curvilinear coordinates systems and application to mesh generation, Int. J.Numer. Meth. Eng 3, 461-477 (1973).
- [6.] Cavendish. J. C, Automatic triangulation of arbitrary planar domains for the finite element method, Int. J. Numer. Meth. Eng 8, 679-696 (1974).
- [7.] Moscardini. A. O, Lewis. B. A and Gross. M A G T H M automatic generation of triangular and higher order meshes, Int. J. Numer. Meth. Eng, Vol 19, 1331-1353(1983).
- [8.] Lewis. R. W, Zheng. Y, and Usmam. A. S, Aspects of adaptive mesh generation based on domain decomposition and Delaunay triangulation, Finite Elements in Analysis and Design 20, 47-70 (1995).
- [9.] W. R. Buell and B. A. Bush, Mesh generation a survey, J. Eng. Industry. ASME Ser B. 95 332-338(1973).
- [10.] Rank. E, Schweingruber and Sommer. M, Adaptive mesh generation and transformation of triangular to quadrilateral meshes, Common. Appl. Numer. Methods 9, 11 121-129(1993).

- [11.] Ho-Le. K, Finite element mesh generation methods, a review and classification, Computer Aided Design Vol.20, 21-38(1988).
- [12.] Lo. S. H, A new mesh generation scheme for arbitrary planar domains, Int. J. Numer. Meth. Eng. 21, 1403-1426(1985).
- [13.] Peraire. J. Vahdati. M, Morgan. K and Zienkiewicz. O. C, Adaptive remeshing for compressible flow computations, J. Comp. Phys. 72, 449-466(1987).
- [14.] George. P.L, Automatic mesh generation, Application to finite elements, New York , Wiley (1991).
- [15.] George. P. L, Seveno. E, The advancing-front mesh generation method revisited. Int. J. Numer. Meth. Eng 37, 3605-3619(1994).
- [16.] Pepper. D. W, Heinrich. J. C, The finite element method, Basic concepts and applications, London, Taylor and Francis (1992).
- [17.] Zienkiewicz. O. C, Taylor. R. L and Zhu. J. Z, The finite element method, its basis and fundamentals, 6th Edn, Elsevier (2007)
- [18.] Masud. A, Khurram. R. A, A multiscale/stabilized finite element method for the advectiondiffusion equation, Comput. Methods. Appl. Mech. Eng 193, 1997-2018(2004)
- [19.] Johnston. B.P, Sullivan. J. M and Kwasnik. A, Automatic conversion of triangular finite element meshes to quadrilateral elements, Int. J. Numer. Meth. Eng. 31, 67-84(1991).
- [20.] Lo. S. H, Generating quadrilateral elements on plane and over curved surfaces, Comput. Stuct. 31(3) 421-426(1989).
- [21.] Zhu. J, Zienkiewicz. O. C, Hinton. E, Wu. J, A new approach to the development of automatic quadrilateral mesh generation, Int. J. Numer. Meth. Eng 32(4), 849-866(1991).
- [22.] Lau. T. S, Lo. S. H and Lee. C. S, Generation of quadrilateral mesh over analytical curved surfaces, Finite Elements in Analysis and Design, 27, 251-272(1997).
- [23.] Park. C, Noh. J. S, Jang. I. S and Kang. J. M, A new automated scheme of quadrilateral mesh generation for randomly distributed line constraints, Computer Aided Design 39, 258-267(2007).
- [24.] Thompson.E.G, Introduction to the finite element method, John Wiley & Sons Inc.(2005)
- [25.] H.T.Rathod, Bharath Rathod, K.T.Shivaram, K. Sugantha Devi, Tara Rathod, An explicit finite element integration scheme for linear eight node convex quadrilaterals using automatic mesh generation technique over plane regions, *International Journal of Engineering and Computer Science*, vol.3, issue5(2014), pp5657-5713.
- [26.] H.T. Rathod, K. Sugantha devi, An All Quadrilateral Automesh Generation Technique and Explicit Integration Scheme for Finite Elements to Solve Some Elliptic Boundary Value Problems in Two Space, International Journal of Engineering and Computer Science, ISSN 2319-7242, Vol.5 issue 12(December 2016), pp 19674-19778.
- [27.] H.T.Rathod, Bharath Rathod, Sugantha Devi.K,Hariprasad.A.S, Finite element solution of Poisson Equation over Polygonal Domains using a novel auto mesh generation technique and an explicit integration scheme for eight node linear convex quadrilaterals, International Journal of Engineering and Computer Science, ISSN:2319-7242, Volume (6) Issue 10 October 2017, pp 22632-22787.
- [28.] H. T. Rathod Md.Shafiqul Islam. H. Y. Shrivall, Bharath Rathod, K. Sugantha Devi, Finite element solution of Poisson Equation over Polygonal Domains using a novel auto mesh generation technique and an explicit integration scheme for nine node linear convex quadrilateral of Lagrange family.*International Journal Of Engineering And Computer Science ISSN:2319-7242Volume 6 Issue 11 November 2017, Page No. 22869-23058.*
- [29.] .H. T. Rathod, Md.Shafiqul.Islam, Bharath Rathod , K. Sugantha Devi , Finite element solution of Poisson Equation over Polygonal Domains using a novel auto mesh generation technique and an explicit integration scheme for linear convex quadrilaterals of cubic order Serendipity and Lagrange families, International Journal Of Engineering And Computer Science ISSN:2319-7242, Volume 7 Issue 1 January 2018, Page No. 23329-23482
- [30.] M.Okabe,Complete Lagrange family for the cube in Finite Element interpolations,Comput.Meth.Appl.Mech.Engng.29(1981)51-56.

- [31.] .H.T,Rathod and Sridevi.Kilari,General complete Lagrange family for the cube in finite element interpolations,Comput.Meth.Appl.Mech and Engrg 181(2000)295-344.
- [32.] H.T.Rathod,Bharath Rathod,K,Sugantha Devi ,A New Approach to Automatic Mesh Generation over Polygonal Domains with Linear, Quadratic,Cubic and Quartic Order Quadrilaterals of Serendipity, Lagrange and Complete Lagrange Finite Elements,International Journal Of Engineering And Computer Science Volume 9 Issue 10 October 2020, Page No. 25208-25239 ISSN: 2319-7242 DOI: 10.18535/ijecs/v9i10.4536

FIGURES

(1) Quintic order complete Lagrange Elements















(2) Sextic order complete Lagrange Elements

	1 E	Ea	ch M	esh v	vith 1	6 32	node	ed Se	xtic Q	uad	rilate	ral E	lemer	nts 8	Nun	nber	of No	des= 353
	4.5	€1 C	286.28	5284	283282	₩ 2 (3093 0	8307	806305	23 C	32732	6325	324323	<mark>24 C</mark>	845.34	4343	34234′	25
		287	2 95	2 98	2 94	281	3 13	3 16	3 12	304	3 31	3 34	330	<mark>7</mark> 822	3 49	3 52	3 48	<mark>3</mark> 40
	4	288				280			(303			(<mark>3</mark> 21				339
		289	2 99	[13]	2 97	279	3 17	[14]	3 15 (302	3 35	[15]	333 (<mark>3</mark> 20	3 53	[16]	351	338
		2 90	7 92	7 96	7 93	278	(3 10	CR14	3 11	301	3 28	3 32	329	319	3 46	3 50	3 47	337
	3.5	291	1	200	233	277	-010	U 14	v ii (300	300 220	4 002	4 025 (318	-040	-200	-071	<mark>\$</mark> 36
		⋪ 6	2 0920	8207	206 <mark>20</mark> :	97 C	23223	1230	229 <mark>228</mark>	<mark>98 C</mark>	250 <mark>2</mark> 4	<mark>92483</mark>	247 <mark>2</mark> 46	99 3	26826	7 <mark>266</mark>	265 <mark>26</mark> 4	 20
		210	7 18	(2)21	(2 17	204	0 36	0 20	0 35	227	0 254	7 57	2 53	245	7 72	0 75	0 71	263
	3	211		<u>~</u> '' ⁄2	203)3 200	239	200 (226	204 2	201	200	.00 244	212	<u>~</u> 15	-211	262	
		212	2 22	[9]	2 20	202	2 40	[10]	2 38 (225	2 58	[11]	2 56 (243	2 76	[12]	2 74	2 61
	0 F	213	() 15	(7)10	0 16	2 01	<u>ക</u> 33	0 27	0 34	224	0 251	<u>0</u> 55	0 52	242	<u>0</u> 60	0 73	∞ 70	2 60
axis	2.5	214	215	219	210	200	200	201	204 (223	201	200	2 52	241	209	215	210	2 59
		∲ 1-C	13213	101300	129 <mark>0128</mark>	\$12 C	1 55015	<mark>401530</mark>	152 <mark>0151</mark>	<mark>93 C</mark>	173017	2 <mark>01710</mark>	170 <mark>169</mark>	# 4	191<mark>0</mark>19	<mark>0°189°</mark>	188 <mark>0</mark> 187	4 5
\geq	2	<mark>9</mark> 33	C 11	<u>0</u> 11	C 40	q 27	C 50	<u>0</u> 62	7 50	9 50	A 77	<u> 0</u> 00	0 76	9 68	0 05	<u> 00</u>	<u>0</u> 04	4 86
		q 34	941	0 44	0 40	† 26	0.09	02	U 56 (49	U 11	0 00	U 70	† 67	990	90	94	4 85
		q 35	<mark>C</mark> 145	[5]	<mark>0</mark> 43	q 25	<mark>C</mark> 163	[6]	C 161 (48	<mark>C</mark> 181	[7]	C 179 (9 66	<mark>C</mark> 99	[8]	<mark>0</mark> 97	d 84
	1.5	q 36	<u>a</u> 00	a 40	~ ~ ~ ~	q 24	0 50	a co	957	447	0 74	0 70	075	965	a 00	anc	a 00	4 83
		4 37	938	U 4Z	039	q 23	000	000	U5/ (446	074	978	U/5	9 64	92	990	93	9 82
		6 2	40 39	- <mark>-38-C</mark>	37 36	 (<mark>88 (67</mark>	- <mark>66-0</mark>	65 (64 (8-0	<mark>91 </mark>	- <mark>89-(</mark>	88 (87 (• - •	14011	<mark>3112</mark>	111211(h o
	1	41	~~	7 0	~	\$5	~	~~~	~ (63	7 5	7 0	~ (86	0 4 0	2 04	~	4 09
		42	4 9	52	48	3 4	92	9 5	Q 1 (62	95	98	94	85	u 18	O 21	0 17	4 08
	05	43	5 3	[1]	9 51	<mark>\$</mark> 3	7 6	[2]	7 4 (<mark>6</mark> 1	9 9	[3]	9 7 (84	<mark>9</mark> 22	[4]	9 20	4 07
	0.5	44	~~~	~~~	<u></u>	<mark>\$</mark> 2	~ ~~		~ ~~ (60	~~~	<u> </u>	C	83	~	~	~	9 06
		45	4 6	9 0	4 7	<mark>\$</mark> 1	69	4 3	4 0	5 9	92	96	93	82	1 15	d 19	U 16	4 05
	0	॑ —⊂	26 2 7	<mark>~28 (</mark>	29 30	<u>– q</u>	54 (5 5	- 56 9	57 (58 (<mark>.</mark> - C	77 <mark>7</mark> 8	- <mark>79-(</mark>	80 (81 (4-0	10010	<mark>1902</mark>	103 <mark>10</mark> 4	b
		0			0.5				1			1.	.5			2		

3

x axis

____ 2.5



	E a 10	ach M 😤 Gogo	Mesh 68	n with	n 20 32	2 node	ed Sexti	c Qua	drila	teral	Elem	ents &	& Nu	mber	of Nod	es= 435
		<mark>\$69</mark>	~ <mark>8</mark> 67	, 36 <mark>6</mark> 26	5											
		<mark>\$</mark> 70	377	380 880	364 Z	201										
	9	3 71	381	2000 Q	876 36	32,396	889,88									
		372	0 74	[17]	879 3 62	2 <mark>395</mark>	398									
	0		3/4	378	36 75 00 6	1 399	394	386 386	10840	Zoc						
	8		91290	280	8/3/00	0 9 (7 892)	^[18] 397	³⁸⁵ 4	13 <mark>/</mark> 1	40040	500	-				
		292	ືດດ	20026	³⁸ 28 62	0 -00Z	³⁹⁶ 393	384 383 4	17	ॅ ∕412	404	42642	5			
	7	293		3 03 (299 📿	6 6	312211	382 4	10	^{9]} 4 18	5^{403}	4 31 🔏	3424	2 <mark>3</mark> 0		
	'	295	3 04	[13]	202 PE	35 3 18	321 -	142333	$\frac{4}{2}$	¹⁴ 41	102	4 35	201 201	30 <mark>7</mark> 422		
		296	2 97	3 01	202 02	³⁴ 322	<u>~</u> - · (3 1)	7 (309)	مصر محر	38 <mark>0328</mark>		⁰ 428	²⁰ 4 232	33 <mark>4</mark> 27 42	1 0	
	6	₩6 ₇ 2	44.		298 🖓	33 00 0 81 F	^[14] 32	0 307	200 (339 ₍₇	35 32	9 503 4	9218	42904	19	
		215	· '\ V +	212 2	1246	52 - 010 Z - 00-7	' 🕄 19 ₍₃ ,	16 306	340	[15]	838 2	 26 3 54	0757	47346	18 25	
<u>.</u>		2 16 '	2 23	2 26	$\frac{7}{2}$	~ <mark>∕2</mark> ⊰∕2 09	^{.36} 235234	305	3 33	337	2000 03	25 324 3 5	8 116	353 🔾	345	
ax	5	2 17 .	2 27	roı	222	08 24	¹ 244 a		255 <u>2</u>	54253	252-	823 3 8	51 👝 🛛	^{7]} 356	344 343	
\geq		218	<u></u>	[9]	2 25 👌	207 🔈	5 [10]	23	- 1 2 59	9 <mark>0962</mark>		19.07	30720	³⁰ 3 5	2 342	
		219	220	2 24	2 21 9	206		²⁴³ 🔁 3	0 😕 0	3 144	2 58	250 049 9	277	+ 1270	26520	
	4		1 37<mark>0</mark>13	6135	1 34 723		² 42 (239 22	29 20 🖓	۱۱۱ مور 56	J 2 61	248	281	200 🤈	76 268	
		0 38	<mark>0</mark> 46	<u>a</u> 10		132	0 1590580 H	57156	28 🛩 3-01-70	26	0 🛛 25	7 2 47	7 74		279 266	j
	2	439	0.50	4 43	45	1 31 (64 😋 67	a 63 9	55	214-017	^z @175	17440			275 2 6	5 54
	5		950	[5]	C 148 (330 <mark>0</mark>	rai 80		54 ^C	82 🈋	85 <mark>0</mark> 8	B1 97	$3 \qquad \bigcirc 0$	99194 0 may	19319 2	5
		d 42	° 43	<mark>C</mark> 147	<mark>9</mark> 44 ⁽	29			153	¹⁸⁶ [7] 😋	84 07	2 –20 71 – mor	$\sqrt{20}$	3 9 99 9	91 190
	2	6-2	1 5- 214	- N2) ۱۹۰ <mark>۰ مالا</mark>	128		d 62	152	379 🤇	383 🤇	180 9	70	07 m	l 2 02	189
		46	~ .	900	92 (41-	- 73	- <mark>72-71-(</mark>	<mark>70 69 (</mark>	8-9	6 95 (<mark>94 </mark>	ი მიკი მიკი	69 <u> </u>		01 0198	9 88 9 87
		47	9 4	5 7	9 53	$\frac{1}{39}$	77 😮	7 6	6 8	0 00	0 03		9 1			1500
	1	<mark>4</mark> 8	5 8	[1]	5 6	38 (81 [2]	79	رمب 680	1 04	[3]	a 02	90		126 012	² 0 13
		49	(5 1	5 5	5 2	3 7	7/ 78	75	65	<u> </u>		4 02	8 8			25 912
	~	50			~-	3 6			64	ب م		90 90	287			
	U	● ─○ 0	51 32 1	2 33 -	34 35 2	- 2 (5 .3	9 060 061 2	- 062-06 ; 1	3 😼 5	82 8	ය 084 6	085 080 7	0 🕰	0050 8	9 9	109 0 10
		-			-	0		•	-		-	•		-	•	

x axis









(3) Septic order complete Lagrange Elements





Each Mesh with 20,-40-noded Septic Quadrilateral Elements & Number of Nodes= 564

	10	F4 75 475 7 7 7 7 7 7 7 7 7 7
	9	477 484490 477 491 489483468 303020100000000000000000000000000000000
	8	479^{492} $(17)^{480}$ $(480^{40}, 500^{4}, 1$
	7	873
	6	377 390 [13] 386 364 413 341405 395 325242524252425124 325530 518 564 [20] 560 545 378 378 385 385 363 413 41405 395 $43942423224254255557555575559544$ 378 378382384380 362 414 [14] 410 395 $43943432425425425555454543$ 4542
axis	5	271 280 $2696 - 2696 - 361$ 400 $400 - 400 - 393$ 438 $[15]$ 434 419 454 $60459 - 55272$ 280 $280 - 285 -$
Υ		274 288 [9] 284 262 311 293 3293 294 262 311 293 3293 3293 3293 3293 3293 3293
	4	$\begin{array}{c} 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 &$
	3	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$
	2	$\begin{array}{c} 6 & 48 \\ \hline 6 & 48 \\ \hline 48 \\ \hline 47 \\ \hline 46 \\ \hline 45 \\ \hline 44 \\ \hline 43 \\ \hline 7 \\ \hline 84 \\ \hline 83 \\ \hline 82 \\ \hline 81 \\ \hline 81$
	1	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
	0	53 55 59 60 56 98 85 89 90 86 74 91 91 92 91 935 936 94 935 95 59 60 56 99 90 86 74 9151 19120116 90 91 91 91 91 91 91 91 91 91 91 91 91 91
	-	0 1 2 3 4 5 6 7 8 9 10

x axis









(4) Octic order complete Lagrange Elements

Each Mesh with 16.-49-noded Octic Quadrilateral Elements & Number of Nodes= 577 461 453 491 522 553 462 47480797870 452 50215160901 490 53542454632 552 521 56575727563 463 481 **477 451 5**12 **5**08 **4**89 **5**43 **5**39 **5**20 **5**74 **5**70 551 4 464 482 484 476 450 513 515 507 488 544 546 538 519 575 577 569 **\$**50 ု**န္ဒ**ရ၀ **(**487 **(**545 [<mark>18</mark>68 465 483 1<mark>4</mark>375 449 514 1537 518 576 549 466 468727 448 49903050500 486 53634353631 **4**69 517 5656565656565 548 3.5 467 447 485 516 547 **+ 632928272625282<mark>87</mark>36768656863626+ 83989798959393**9 29429282 20 330 822 360 891 422 <mark>\$</mark>359 색 🛯 🕮 134 ^ **3**31 **340494843**39 **3**21 **373807978**70 00001 390 **438**4 44032 421 3) **4**39 **4**20 **3**46 **3**20 **3**81 **377 358 4**12 **4**08 **3**89 **4**43 332 350 <mark>\$33 351 353 345 \$19 382 384 376 \$57 413 415 407 \$88 444 446 438 \$</mark>19 _၂ၛှိ44 **(**318 **(**383 ក្រស្រុវទ **\$**56 **4**14 1496 <mark>\$</mark>87 r<mark>12</mark>37 <mark>4</mark>18 334 352 **4**45 2.5 335 33343424338 317 368727373 355 **399030405**00 386 430 34 35 36 31 417 axis 316 354 385 336 416 303262626262626262014292929292929292929292929292925 >229 260 199 191 291 2 290 228 27282727282 200 202 12 12 1608 190 2424242239 259 3021 **31009**01 **2**27 **2**81 **3**08 ⁄ 289 246 258 312 201 219 215 89 📿 50 277 202 220 222 214 (88 🛛 251 📿 53 📿 45 📿 26 📿 82 📿 84 📿 76 🤇 257 313 315 307 288 1.5 **2**03 **2**21 1<mark>67</mark>44 60<mark>6</mark>81 15**2**13 **0**87 **2**52 **2**25 **2**83 r<mark>,2</mark>75 256 314 **2**87 204 208 12 12 12 12 07 186 23242424238 224 268727 27269 255 **29903030** 286 **3**00 285 85 223 254 205 -919089888786858 012928272625242 9 **1 67 66 65 64 63 62 6** 0 6 246245242232224124 1 47 39 84 122 160
a 33 42 41 40 32
 a 21
 a 71 80 79 78 70
 b 59

 a 43
 a 39
 a 20
 a 81
 a 77
 b 58

 a 44
 a 46
 a 38
 a 19
 a 82
 a 84
 a 76
 b 57

 48 5766656456 38 9510403024 83 37 36 35 <mark>4</mark>9 **6**7 **6**3 **C**105 **1**01 **8**2 0.5 50 68 **1**06 **1**08 **1**00 **4**1 **7**0 **6**2 18 983 [<mark>4</mark>75 **॑**56 51 **6**9 1<mark>مم</mark> ا **0**07 9**وې**ر 80 **C**145 <mark>(</mark>52 5458596055 34 9296979893 **1 30 34 35 36** 31 469 **†**55 79 **q**17 **1 68** <mark>\$</mark>3 33 78 116 354 }12131416-0147484950515256 0 47273747576 ▛──ੑਗ਼ੑਗ਼ੑਗ਼ 2.5 0 0.5 1 1.5 2

x axis













(5) Nonic order complete Lagrange Elements

		Each Mesh with 16,-60-noded Nor	nic Quadrilateral Elements & Numl	ber of Nodes= 729	
4.5 🕶 😘	77 <mark>5765755745735725715</mark>	70 £ 2- <mark>62562462362262162061961</mark>	323-665664663662661660659658	£4-70570470370270170069969	8 ¶5
578		(569	6 17	\$57	697
(5 79	589601600599598588	568 629641640639638628	616 669681680679678668	¢56 709721720719718708	696
580	602 cong cong 597	(567 (642 (640 (648 (637	615 682 680 688 677	655 722 700 708 717	695
4 <mark>(</mark> 581	603 ran 596	5 66 6 43 6 36	614 683 ram 676	654 723 Mai 716	694
582	6 04 [13] 595	(5 65 (6 44 ^[14] 6 35	(613 (684 [10] (675	(653 (724 ^[10] 715	693
583	6 05 606 607 594	(5 64 (6 45 (646 (647 (634	612 685 686 687 674	652 725 26 21 714	692
584	5865905915925935 87	563 626630631632633627	611 6666670671672673667	651 706710711712713707	691
3.5 <mark>(5</mark> 85		5 62	<mark>(6</mark> 10	650	690
06- 44	094084074064054044034 6	92 <mark>#7-04570456045504540453045204510456</mark>	9 <mark>#8-04970496049504940493049204910490</mark>	9 53753653553453353253153	0 20
410		<mark>4</mark> 01	449	489	5 29
411	(421)4337432743174307420	400 (46104737472747104701260	448 501513512511510500	488 541553552551550540	5 28
3/412	4 34 0 41 0 429	(299) (474 (481) (469)	(447 514 s 21 s 20 509	487 554 men 549	5 27
413	435 roj 428	3 98 4 75 ₁₄₀₁ 4 68	446 515 radi 508	486 555 1401 548	(5 26
<mark>4</mark> 14	4 36 ^[9] 4 27	(397 476 ^[10] 467	445 5 16 [11] 507	485 5 56 ^[12] 547	(5 25
<mark>4</mark> 15	4 39 4 38	(3 96 (4 77 4 78 (4 79 (466	444 517 518 519 506	(484 (557 \$58 \$59 (546	(5 24
2.5 <mark>0</mark> 416	418422423424425419	4395 458462463464465459	498502503504505499	483 5 38 5 42 5 43 5 44545 5 39	(5 23
<u>.0</u> (417		<mark>(</mark> 394	442	482	(5 22
α 🕂 🤁	<mark>412462392382372362352</mark> 3	34 <mark>#2-28928828728628528428328</mark> 2	2 <mark>\$3-029028027026025024023022</mark>	4-36936836736636536436336	2₡ 5
> 242		233	281	\$21	3 61
2 <mark>/2</mark> 43	2532652642632622 52	232 093305304303302092	280 (33334533443433342332)	\$20 \$73385384383382372	3 60
244	266 073 079 261	231 306 a13 a12 301	279 346 353 352 341	319 386 agas agay 381	3 59
245	2 67 _{[E1} 2 60	230 307 re1 300	278 347 171 340	318 387 ro1 387 380	\$ 58
246	2 68 ^[0] 2 59	2 29 3 08 ^[0] 2 99	277 3 48 [/] 339	(3 17 (3 88 ^[0] 379	\$57
1.5 <mark>/2</mark> 47	2 69 2 /0 2 /1 2 58	228 309 310 311 298	276 349 350 351 338	\$16 389 390 391 378	\$ 56
248	250254255256257251	227 2902942952962972 91	275 303343353363373 31	\$15 \$370374375376377371	\$ 55
249		226	274	314	3 54
6-4	9 Q48 Q47 Q46 Q45 Q44 Q43 Q4 2	2 🕴 010501040103010201010100099 098	<mark>\$ 053052051058049048047046</mark>	<mark>\$ ~20120009909809709609509</mark>	4♥ 0
150		<mark>4</mark> 1	9 7	(145	9 93
<mark>5</mark> 1	6 1 7 3 7 2 7 1 7 0 6 0	40 0109121020019018008	9 6 0.573 693 683 673 663 56	44 005017016015014004	9 92
<mark>(5</mark> 2	74 <u>k</u> 1 <u>k</u> n 69	⁽³⁹) ⁽¹²² ⁽¹²⁰) ⁽¹²⁰⁾ ⁽¹²⁰⁾ ⁽¹²⁰⁾	95 0170 0177 0176 0165	143 218 025 021 213	q 191
<mark>(</mark> 53	7 5 ran 6 8	3 8 1 23 m 1 16	9 4 0 71 ron 0 64	42 219 14 212	9 90
0.5 <mark>5</mark> 4	7 6 ^[1]	\$ 7 9 24 1 2	9 3 9 72 ^[3] 9 63	(141 (220 ^[4] 221	(189
55	(7 7 (7 8 (7 9 (6 6		9 2 9 73 9 74 9 75 9 62	0140 0221 0222 0223 0210	† 88
56	(58 (62 (63 (64 (65 (59	<mark>\$</mark> 5 0106011001110112011301 07	(g)1 015401580159016001610155	939 2022062072082092 03	9 87
57		<mark>(3</mark> 4	9 0	q 38	1 86
00-2	6 9 7 98 99 90 91 92 9;	3 🖢 - 182 - 183 - 184 - 185 - 186 - 187 - 188 - 189	• 030031032033034035036037	4 ⁴7847948048148248348448	56
0	0.5	1	1.5	2	2.5
			x axis		







H.T.Rathod, IJECS Volume 09 Issue 10 October, 2020 Page No. 25253-25306







(6) Decic order complete Lagrange Elements







Each Mesh with 4,-72-noded Decic Quadrilateral Elements & Number of Nodes= 245







MATLAB CODES

Code (1)

```
function[]=FEMmeshExample4triangleNquadrilateral72node(gdata)
%This example generates NE elements for a rectangular structure of
    %length = Ly units and width = Lx units with Nx divisions on the x
   % Ny=NE/(2*Nx); %Divisions on y axis
   % cla
   N=0;
    switch gdata
    case 1
    Lx=1;
    Ly=1;
    Nx=8;
    NE=144;
    X=[0;10;8; 0]
    Y = [0; 0; 7; 10]
    hdata=gdata
    case 2
    Lx=1;
    Ly=1;
    Nx=4;
    NE=40;
    X=[0;10;8; 0]
    Y = [0; 0;7;10]
    hdata=gdata
    case 3
    Lx=1;
    Ly=1;
    Nx=2;
    NE=8;
    X=[0;10;8; 0]
    Y = [0; 0; 7; 10]
    hdata=gdata
    case 4
    Lx=1;
    Ly=1;
    Nx=16;
    NE=288;
    X=[0;10;8; 0]
    Y = [0; 0; 7; 10]
    hdata=gdata
    case 5
    Lx=1;
    Ly=1;
    Nx=16;
    NE=288;
    X=[0;1;1;0]
    Y = [0; 0; 1; 1]
    hdata=gdata
    case 6
    Lx=1;
    Ly=1;
    Nx=8;
    NE=144;
    X = [0; 1; 1; 0]
    Y = [0; 0; 1; 1]
    hdata=gdata
   case 7
    Lx=1;
    Ly=1;
```

```
Nx=4;
    NE = 40;
    X = [0; 1; 1; 0]
    Y = [0; 0; 1; 1]
    hdata=gdata
    case 8
    Lx=1;
    Ly=1;
    Nx=2;
    NE=8;
    X=[0;1;1;0]
    Y = [0; 0; 1; 1]
    hdata=gdata
        case 9%beginning-Q1
    Lx=1;
    Ly=1;
    Nx=10;
    NE=160;
    X = [0; 10; 5; 0]
    Y = [0; 0; 10; 10]
    hdata=9
         case 10%beginning-Q2
    Lx=1;
    Ly=1;
    Nx=10;
    NE=160;
    X = [-10; 0; 0; -5]
    Y = [0;0;10;10]
    hdata=9
          case 11%beginning-Q3
    Lx=1;
    Ly=1;
    Nx=10;
    NE=160;
    X = [0; -10; -5; 0]
    Y = [0; 0; -10; -10]
    hdata=9
           case 12%beginning-Q4
    Lx=1;
    Ly=1;
    Nx=10;
    NE=160;
    X=[10;0; 0; 5]
    Y = [0; 0; -10; -10]
    hdata=9
        case 13%6-node convex polygonQ1
                      Nx=2;
     Lx=1;
             Ly=1;
                                 NE=8;
            Ly=1;
                      Nx=2;
     Lx=1;
                                NE=8;N=2;
                 Nx=4; NE=32;N=4
Lx=1;
        Ly=1;
     A1= 0;A2= .05;A3=0.1;A4=0.7;A5= 1; A6=0.75;A7=0.5;A8= 0.25;A9=0.95/2 ;
    B1=0.25;B2=0.125;B3= 0;B4=0.2;B5=0.5; B6=0.85;B7= 1;B8=0.625; ;B9=0.825/2;
    X = [A9; A4; A5; A6]
    Y = [B9; B4; B5; B6]
    hdata=10
       case 14%6-node convex polygonQ2
            Ly=1; Nx=2;
Ly=1; Nx=2;
                               NE=8;
     Lx=1;
     Lx=1;
                                  NE=8;N=2;
Lx=1; Ly=1; Nx=4; NE=32;N=4
A1= 0;A2= .05;A3=0.1;A4=0.7;A5= 1; A6=0.75;A7=0.5;A8= 0.25;A9=0.95/2;
     B1=0.25;B2=0.125;B3= 0;B4=0.2;B5=0.5; B6=0.85;B7= 1;B8=0.625; ;B9=0.825/2;
    X = [A8; A9; A6; A7]
    Y=[B8;B9;B6;B7]
    hdata=10
       case 15%6-node convex polygonQ3
     Lx=1;
              Ly=1;
                        Nx=2;
                                  NE=8;
```

```
Lx=1;
            Ly=1;
                    Nx=2;
                            NE=8;N=2;
Lx=1; Ly=1; Nx=4; NE=32;N=4
A1= 0;A2= .05;A3=0.1;A4=0.7;A5= 1; A6=0.75;A7=0.5;A8= 0.25;A9=0.95/2 ;
    B1=0.25;B2=0.125;B3= 0;B4=0.2;B5=0.5; B6=0.85;B7= 1;B8=0.625; ;B9=0.825/2;
   X = [A9; A8; A1; A2]
   Y=[B9;B8;B1;B2]
   hdata=10
      case 16%6-node convex polygonQ4
    Lx=1; Ly=1; Nx=2; NE=8;
Lx=1; Ly=1; Nx=2; NE=8;N=2;
Lx=1; Ly=1; Nx=4; NE=32;N=4
    A1= 0;A2= .05;A3=0.1;A4=0.7;A5= 1; A6=0.75;A7=0.5;A8= 0.25;A9=0.95/2 ;
    B1=0.25;B2=0.125;B3= 0;B4=0.2;B5=0.5; B6=0.85;B7= 1;B8=0.625; ;B9=0.825/2;
   X = [A4; A9; A2; A3]
   Y = [B4; B9; B2; B3]
   hdata=10
 %======6-node convex polygon discritised into a coarse mesh of four quadrilateral
with no subdivisions========
      case 17%6-node convex polygonQ1
    Lx=1; Ly=1; Nx=1; NE=2;
    A1= 0;A2= .05;A3=0.1;A4=0.7;A5= 1; A6=0.75;A7=0.5;A8= 0.25;A9=0.95/2 ;
    B1=0.25;B2=0.125;B3= 0;B4=0.2;B5=0.5; B6=0.85;B7= 1;B8=0.625; ;B9=0.825/2;
   X = [A9; A4; A5; A6]
   Y = [B9; B4; B5; B6]
   hdata=11
      case 18%6-node convex polygonQ2
                   Nx=1; NE=2;
    Lx=1; Ly=1;
    A1= 0;A2= .05;A3=0.1;A4=0.7;A5= 1; A6=0.75;A7=0.5;A8= 0.25;A9=0.95/2 ;
    B1=0.25;B2=0.125;B3= 0;B4=0.2;B5=0.5; B6=0.85;B7= 1;B8=0.625; ;B9=0.825/2;
   X = [A8; A9; A6; A7]
   Y = [B8; B9; B6; B7]
   hdata=11
      case 19%6-node convex polygonQ3
    Lx=1; Ly=1; Nx=1; NE=2;
    A1= 0;A2= .05;A3=0.1;A4=0.7;A5= 1; A6=0.75;A7=0.5;A8= 0.25;A9=0.95/2 ;
    B1=0.25;B2=0.125;B3= 0;B4=0.2;B5=0.5; B6=0.85;B7= 1;B8=0.625; ;B9=0.825/2;
   X = [A9; A8; A1; A2]
   Y=[B9;B8;B1;B2]
   hdata=11
       case 20%6-node convex polygonQ4
                    Nx=1;
    Lx=1; Ly=1;
                             NE=2;
    A1= 0;A2= .05;A3=0.1;A4=0.7;A5= 1; A6=0.75;A7=0.5;A8= 0.25;A9=0.95/2 ;
    B1=0.25;B2=0.125;B3= 0;B4=0.2;B5=0.5; B6=0.85;B7= 1;B8=0.625; ;B9=0.825/2;
   X = [A4; A9; A2; A3]
   Y = [B4; B9; B2; B3]
   hdata=11
  %_____
      case 21%standard square
   Lx=1; Ly=1;
                   Nx=1; NE=2;
  X=[-1; 1;1; -1]
  Y=[-1;-1;1; 1]
  hdata=12
      case 22% arbitrary quadrilateral
   Lx=1; Ly=1; Nx=1; NE=2;
  X = [-1; 2; 3; 1]
  Y = [2; 1; 3; 4]
  hdata=13
  case 23%Q1<11,5,6,7 >==>FIRST QUADRILATERAL OF NINE NODE NONCONVEX POLYGON
      Ly=1; Nx=2; NE=8;N=2;
Ly=1; Nx=4; NE=32;N=4;
Lx=1;
Lx=1;
               Nx=8; NE=256;N=8;
      Ly=1;
%Lx=1;
```

```
A1=0.25;A2=0.75;A3=0.75;A4= 1;A5=0.75;A6=0.75;A7=0.5;A8= 0;A9=0.25;A10=1.75/2;A11=
0.5;
B1=
      0;B2= 0.5;B3= 0;B4=0.5;B5=0.75;B6=0.85;B7= 1;B8=0.75;B9=
0.5;B10=1.25/2;B11=1.25/2;
  X=[A11; A5; A6; A7];
  Y=[B11; B5; B6; B7];
hdata=14
case 24%Q2<9,11,7,8>==>SECOND QUADRILATERAL OF NINE NODE NONCONVEX POLYGON
   Lx=1; Ly=1; Nx=2; NE=8;
          Ly=1; Nx=2;
                          NE=8;N=2;
   Lx=1;
      Ly=1; Nx=4; NE=32;N=4;
Lx=1;
%Lx=1; Ly=1; Nx=8; NE=256;N=8;
A1=0.25;A2=0.75;A3=0.75;A4= 1;A5=0.75;A6=0.75;A7=0.5;A8= 0;A9=0.25;A10=1.75/2;A11=
0.5:
B1= 0;B2= 0.5;B3= 0;B4=0.5;B5=0.75;B6=0.85;B7= 1;B8=0.75;B9=
0.5;B10=1.25/2;B11=1.25/2;
   X=[A9; A11; A7; A8];
   Y=[B9; B11; B7; B8];
hdata=14
case 25%Q3<11,9,1,2>==>THIRD QUADRILATERAL OF NINE NODE NONCONVEX POLYGON
          Ly=1; Nx=2; NE=8;
   Lx=1;
   Lx=1;
                           NE=8;N=2;
          Ly=1;
                  Nx=2;
Lx=1; Ly=1; Nx=4; NE=32;N=4
   A1=0.25;A2=0.75;A3=0.75;A4= 1;A5=0.75;A6=0.75;A7=0.5;A8=
0;A9=0.25;A10=1.75/2;A11= 0.5;
   B1= 0;B2= 0.5;B3= 0;B4=0.5;B5=0.75;B6=0.85;B7= 1;B8=0.75;B9=
0.5;B10=1.25/2;B11=1.25/2;
  X=[A11; A9; A1; A2];
  Y=[B11; B9; B1; B2];
hdata=14
case 26%Q4<5;11;2;10>==>FOURTH QUADRILATERAL OF NINE NODE NONCONVEX POLYGON
   Lx=1; Ly=1; Nx=2;
                          NE=8:
          Ly=1; Nx=2;
                           NE=8;N=2;
   Lx=1;
      Ly=1; Nx=4;
Ly=1; Nx=4;
                       NE=32;N=4 ;
Lx=1;
                        NE=256;N=8;
%Lx=1;
 A1=0.25;A2=0.75;A3=0.75;A4= 1;A5=0.75;A6=0.75;A7=0.5;A8= 0;A9=0.25;A10=1.75/2;A11=
0.5;
 B1= 0;B2= 0.5;B3= 0;B4=0.5;B5=0.75;B6=0.85;B7= 1;B8=0.75;B9=
0.5;B10=1.25/2;B11=1.25/2;
  X=[A5; A11; A2; A10];
  Y=[B5; B11; B2; B10];
  hdata=14
case 27%Q5<10;2;3;4>==>FIFTH QUADRILATERAL OF NINE NODE NONCONVEX POLYGON
   Lx=1; Ly=1; Nx=2; NE=8;
   Lx=1;
          Ly=1; Nx=2;
                           NE=8;N=2;
Lx=1; Ly=1; Nx=4; NE=32;N=4;
      Ly=1; Nx=4; NE=256;N=8;
%Lx=1;
 A1=0.25;A2=0.75;A3=0.75;A4= 1;A5=0.75;A6=0.75;A7=0.5;A8= 0;A9=0.25;A10=1.75/2;A11=
0.5;
     0;B2= 0.5;B3= 0;B4=0.5;B5=0.75;B6=0.85;B7= 1;B8=0.75;B9=
 B1=
0.5;B10=1.25/2;B11=1.25/2;
   X=[A10; A2; A3; A4];
   Y=[B10; B2; B3; B4];
   hdata=14
  case 28%arbitrary quadrilateral
   Lx=1;
           Ly=1;
                  Nx=4; NE=40;
  X = [-1; 2; 3; 1]
  Y = [2; 1; 3; 4]
```

```
hdata=15
   case 29
    Lx=1;
    Ly=1;
    Nx=4;
    NE=32;
    X=[0;1;1;0]
    Y=[0;0;1;1]
    hdata=16
    case 30%Q1<7;6;1;4>==>FIRST QUADRILATERAL OF EQUILATERAL TRIANGLE
    Lx=1; Ly=1; Nx=2; NE=8;
Lx=1; Ly=1; Nx=2; NE=8;N=2;
Lx=1; Ly=1; Nx=4; NE=32;N=4;
%Lx=1; Ly=1; Nx=4; NE=256;N=8;
A1=-0.5;A2=0.5;A3= 0;A4=0;A5= 0.25;A6= -0.25;A7=0;
  B1= 0;B2= 0;B3=sqrt(3)/2;B4=0;B5=sqrt(3)/4;B6=sqrt(3)/4;B7=sqrt(3)/6;
    X = [A7; A6; A1; A4];
    Y=[B7; B6; B1; B4];
    hdata=17
     case 31%Q2<7;4;2;5>==>SECOND QUADRILATERAL OF EQUILATERAL TRIANGLE
    Lx=1; Ly=1; Nx=2; NE=8;
    Lx=1; Ly=1; Nx=2; NE=8;N=2;
Lx=1; Ly=1; Nx=4; NE=32;N=4;
%Lx=1; Ly=1; Nx=4; NE=256;N=8;
  A1=-0.5;A2=0.5;A3=
                             0;A4=0;A5= 0.25;A6= -0.25;A7=0;
  B1= 0;B2= 0;B3=sqrt(3)/2;B4=0;B5=sqrt(3)/4;B6=sqrt(3)/4;B7=sqrt(3)/6;
    X=[A7; A4; A2; A5];
    Y=[B7; B4; B2; B5];
    hdata=17
      case 32%Q3<7;5;3;6>==>THIRD QUADRILATERAL OF EQUILATERAL TRIANGLE
    Lx=1; Ly=1; Nx=2; NE=8;
    Lx=1; Ly=1; Nx=2; NE=8;N=2;
Lx=1; Ly=1; Nx=4; NE=32;N=4;
%Lx=1; Ly=1; Nx=4; NE=256;N=8;
  A1=-0.5;A2=0.5;A3= 0;A4=0;A5= 0.25;A6= -0.25;A7=0;
  B1= 0;B2= 0;B3=sqrt(3)/2;B4=0;B5=sqrt(3)/4;B6=sqrt(3)/4;B7=sqrt(3)/6;
    X=[A7; A5; A3; A6];
    Y=[B7; B5; B3; B6];
    hdata=17
    case 33%Decic element over a 2-square
        Lx=1; Ly=1; Nx=1; NE=2;
    X = [-1; 1; 1; -1]
    Y = [-1; -1; 1; 1]
    hdata=18
    case 34
                 Ly=1; Nx=1; NE=2;
Ly=1; Nx=2; NE=8;N=2;
Ly=1; Nx=4; NE=32;N=4;
Ly=1; Nx=4; NE=32;N=0;
         Lx=1;
         Lx=1;
         Lx=1;
         Lx=1;
    % Lx=1; Ly=1; Nx=8; NE=128;N=0;
Lx=1; Ly=1; Nx=2; NE=8;N=0;
X=[0;sqrt(5); sqrt(5); 0]
              0;2*sqrt(5);2*sqrt(5)]
    Y=[0;
    hdata=19
        case 35
                 Ly=1; Nx=1; NE=2;
Ly=1; Nx=2; NE=8;N=2;
         Lx=1;
         Lx=1;
                   Ly=1;
Ly=1;
                             Nx=4; NE=32;N=4;
Nx=4; NE=32;N=0;
          Lx=1;
                                       NE=32;N=4 ;
         Lx=1;
         % Lx=1; Ly=1; Nx=8; NE=128;N=0;
Lx=1; Ly=1; Nx=2; NE=8;N=0;
    X = [0; 1; -3; -4]
    Y = [0; 2; 4; 2]
    hdata=20
```

end

```
[xygcoord, xycoords, xycoordrgqd, xycoordrgqdm, cT, qT, nNodes]=femTriangularMeshGenerator4tr
iangleNquadrilateral72node(Lx,Ly,Nx,NE,X,Y)
   % return
    [nnode,dimension]=size(xygcoord)
    disp(['Number of nodes = ',num2str(nnode)])
    disp('Connectivity Table')
    disp(cT)
    disp(qT)
    axis square
   %axis equal
    z=1;
    for i=1:NE
        figure(2*hdata-
1), patch('Vertices', xycoords(z:z+2,:), 'Faces', [1,2,3], 'FaceColor', 'none', 'EdgeColor', 'g
')
       if Nx<9
       midx=mean(xycoords(z:z+2,1));
       midy=mean(xycoords(z:z+2,2));
       text(midx,midy,['[',num2str(i),']']);
       end
       hold on
        z = z + 3;
    end
    ex****
    hold on
xlabel('x axis')
ylabel('y axis')
st1='FEM MESH WITH ';
st2=num2str(NE);
st3='; 3-node Linear ';
st4='Triangular';
st5=' Elements'
st6='& Nodes='
st7=num2str(nNodes);
title([st1, st2, st3, st4, st5, st6, st7])
    figure(2*hdata-1),scatter(xycoords(:,1),xycoords(:,2),'MarkerFaceColor','r')
    hold on
    %put node numbers
if Nx<9
for jj=1:nNodes
text(xygcoord(jj,1),xygcoord(jj,2),['.',num2str(jj)]);
end
end
2****
                                                                     * * * * * * * * * * * * * * * * * * *
*****
disp('nodal connectivity for 60 noded decic convex quadrilaterals ')
disp(qT) %
  z=1;
   for i=1:NE/2
figure(2*hdata),patch('Vertices',xycoordrgqd(z:z+3,:),'Faces',[1,2,3,4],'FaceColor','no
ne', 'EdgeColor', 'r')
        xx=xycoordrgqd(z:z+3,1); yy=xycoordrgqd(z:z+3,2);
        hold on
```

```
patch(xx,yy,'w')
        if (Nx<9) & (Nx~=N)
        midx=mean(xycoordrgqd(z:z+3,1));
        midy=mean(xycoordrgqd(z:z+3,2));
        text(midx,midy,['[',num2str(i),']']);
        end
        hold on
        z = z + 4;
    end
    figure(2*hdata),scatter(xycoordrgqd(:,1),xycoordrgqd(:,2),'MarkerFaceColor','r')
    figure(2*hdata),scatter(xycoordrgqdm(:,1),xycoordrgqdm(:,2),'MarkerFaceColor','y')
    hold on
    %put node numbers
if (Nx<9) & (Nx~=N)
% figure(2*hdata),scatter(xygcoord(:,1),xygcoord(:,2),'MarkerFaceColor','w')
for jj=1:nnode
text(xygcoord(jj,1),xygcoord(jj,2),num2str(jj));
end
end
8 8
hold on
xlabel('x axis')
ylabel('y axis')
st1='Each Mesh with ';
st2=num2str(NE/2);
st3=',-72-noded Decic ';
st4='Quadrilateral';
st5=' Elements ';
st6='& Number of Nodes= ';
st7=num2str(nnode);
title([st1, st2, st3, st4, st5, st6, st7])
   end
```

code(2)

function

[xygcoord, xycoords, xycoordrgqd, xycoordrgqdm, cT,qT,nNodes]=femTriangularMeshGenerator4triangleNquadrilateral72 node(Lx,Ly,Nx,NE,X,Y)

```
This function generates triangular mesh for a rectangular
   shape structure for finite element analysis
2
    coords = x and y coordinates of each element nodes
e
÷
   сТ
           =
               nodal connectivity for triangles
   qT
              nodal connectivity for quadrilaterals
e
           =
   nNodes =
8
               Number of nodes
           =
               width of the rectangular structure
8
   Lx
           _
              Height of the rectangular structure
2
   Lv
응
   Nx
               Number of divisions on x- axis
              Number of elements
응
   NE
           =
응
x1=X(1,1);
x2=X(2,1);
x3=X(3,1);
x4=X(4,1);
```

y1=Y(1,1); y2=Y(2,1); y3=Y(3,1);

```
y4=Y(4,1);
    if mod((NE/Nx), 2) \sim = 0
        errordlg('The No of divisions on X axis must divide No of Elements twice')
    end
   Ny=NE/(2*Nx);
                   %Divisions on v axis
    nNodes =(Nx+1)*(Ny+1); %No of nodes
    m=1;
    j = (1:Nx);
    k=linspace(Nx*2,NE,Ny);
    for i=1:Ny
        cT(m:2:k(i),1) = j'; %node 1 of 1st element
        cT(m+1:2:k(i),1) = j'; %node 1 of 2nd element
        cT(m:2:k(i),2)=(j+1)'; %%node 2 of 1st element
        cT(m+1:2:k(i),2)=(j+Nx+2)';%node 2 of 2nd element
        cT(m:2:k(i),3) = (j+Nx+2)'; %node 3 of 1st element
        cT(m+1:2:k(i),3)=(j+1+Nx)';
                                        %%node 3 of 1st element
        m=k(i)+1;
        j=j+Nx+1;
    end
    for m=1:NE/2
        qT(m,1)=cT(2*m-1,1);
        qT(m,2)=cT(2*m-1,2);
        qT(m, 3) = cT(2*m-1, 3);
        qT(m,4)=cT(2*m,3);
    end
     αT
    ax=linspace(0,Lx,Nx+1); %%%x coordinates
    by=linspace(0,Ly,Ny+1); %%%y coordinates
    X1 = [];
    Y1=[];
    for i1=1:Ny+1
             General Nodal Coordinates layer by layer
        by1(1:Nx+1)=by(i1);
        X1=[X1 ax];
        Y1=[Y1 by1];
    end
   gcoord(:,1)=X1';
   gcoord(:,2)=Y1';
   NN=(1:nNodes)':
   [NN gcoord]
    j=1:3;
    %each element coordinates for triangles
    for n=1:NE
        X(j,1) = X1(cT(n,:));
        Y(j,1)=Y1(cT(n,:));
        j=j+3;
    end
                  %x and y coordinates for triangles
    coords=[X Y];
      j=1:4;
    %each element coordinates for quadrilaterals
    for n=1:NE/2
        XX(j,1) = X1(qT(n,:));
        YY(j,1)=Y1(qT(n,:));
        j=j+4;
    \operatorname{end}
    coord=[XX YY]; %x and y coordinates for quadrilaterals
 8~~~
% mesh generation of 16-node special quadrilaterals
nnd=nNodes;
for inum=1:nnd
    for jnum=1:nnd
        trisect(inum,jnum,1)=0;
        trisect(inum,jnum,2)=0;
        trisect(inum,jnum,3)=0;
```

trisect(inum,jnum,4)=0;

```
trisect(inum,jnum,5)=0;
   end
end
nd=nnd;mm=NE/2;
for mmm=1:mm
mmm1=qT(mmm,1);
mmm2=qT(mmm, 2);
mmm3=qT(mmm,3);
mmm4=qT(mmm, 4);
%trisectional points and midpoints: side-1 of 4-node quadrilateral
if((trisect(mmm1,mmm2,1)==0))
    nd=nd+1;
   trisect(mmm1,mmm2,1)=nd;
end
if(trisect(mmm1,mmm2,2)==0)
    nd=nd+1;
   trisect(mmm1,mmm2,2)=nd;
end
if((trisect(mmm1,mmm2,3)==0))
    nd=nd+1;
   trisect(mmm1,mmm2,3)=nd;
end
if(trisect(mmm1,mmm2,4)==0)
   nd=nd+1;
   trisect(mmm1,mmm2,4)=nd;
end
if((trisect(mmm1,mmm2,5)==0) &&(trisect(mmm2,mmm1,5)==0))
    nd=nd+1;
   trisect(mmm1,mmm2,5)=nd;
   trisect(mmm2,mmm1,5)=nd;
end
if((trisect(mmm1,mmm2,4)~=0) &&(trisect(mmm2,mmm1,4)==0))
    nd=nd+1;
   trisect(mmm2,mmm1,4)=nd;
end
if((trisect(mmm1,mmm2,3)~=0) &&(trisect(mmm2,mmm1,3)==0))
    nd=nd+1;
   trisect(mmm2,mmm1,3)=nd;
end
if((trisect(mmm1,mmm2,2)~=0)&&(trisect(mmm2,mmm1,2)==0)))
   nd=nd+1;
   trisect(mmm2,mmm1,2)=nd;
end
if((trisect(mmm1,mmm2,1)~=0)&&(trisect(mmm2,mmm1,1)==0)))
   nd=nd+1;
   trisect(mmm2,mmm1,1)=nd;
end
% trisectional points and midpoints: side-2 of 4-node quadrilateral
if ((trisect(mmm2,mmm3,1)==0))
   nd=nd+1;
   trisect(mmm2,mmm3,1)=nd;
end
if(trisect(mm2,mm3,2) == 0)
  nd=nd+1:
   trisect(mmm2,mmm3,2)=nd;
end
if(trisect(mmm2,mmm3,3)==0)
  nd=nd+1;
   trisect(mmm2,mmm3,3)=nd;
end
if(trisect(mmm2,mmm3,4)==0)
   nd=nd+1;
   trisect(mmm2,mmm3,4)=nd;
end
if((trisect(mmm2,mmm3,5)==0) &&(trisect(mmm3,mmm2,5)==0))
   nd=nd+1:
   trisect(mmm2,mmm3,5)=nd;
   trisect(mmm3,mmm2,5)=nd;
end
if ((trisect(mmm2,mmm3,4)~=0) && (trisect(mmm3,mmm2,4)==0))
    nd=nd+1;
   trisect(mmm3,mmm2,4)=nd;
end
if((trisect(mmm2,mmm3,3)~=0)&&(trisect(mmm3,mmm2,3)==0)))
    nd=nd+1;
```

```
trisect(mmm3,mmm2,3)=nd;
end
if((trisect(mmm2,mmm3,2)~=0)&&(trisect(mmm3,mmm2,2)==0)))
   nd=nd+1;
   trisect(mmm3,mmm2,2)=nd;
end
if((trisect(mmm2,mmm3,1)~=0) &&(trisect(mmm3,mmm2,1)==0))
    nd=nd+1;
   trisect(mmm3,mmm2,1)=nd;
end
% trisectional points and midpoints: side-3 of 4-node quadrilateral
if((trisect(mmm3,mmm4,1)==0))
   nd=nd+1;
   trisect(mmm3,mmm4,1)=nd;
end
if(trisect(mmm3,mmm4,2)==0)
   nd=nd+1;
   trisect(mmm3,mmm4,2)=nd;
end
if(trisect(mmm3,mmm4,3)==0)
    nd=nd+1;
    trisect(mmm3,mmm4,3)=nd;
end
if(trisect(mmm3,mmm4,4)==0)
   nd=nd+1;
   trisect(mmm3,mmm4,4)=nd;
end
if((trisect(mmm3,mmm4,5)==0)&&(trisect(mmm4,mmm3,5)==0))
    nd=nd+1;
   trisect(mmm3,mmm4,5)=nd;
   trisect(mmm4,mmm3,5)=nd;
end
if((trisect(mmm3,mmm4,4)~=0)&&(trisect(mmm4,mmm3,4)==0)))
   nd=nd+1;
   trisect(mmm4,mmm3,4)=nd;
end
if((trisect(mmm3,mmm4,3)~=0) &&(trisect(mmm4,mmm3,3)==0))
   nd=nd+1;
   trisect(mmm4,mmm3,3)=nd;
end
if ((trisect(mmm3,mmm4,2) ~=0) && (trisect(mmm4,mmm3,2) ==0))
    nd=nd+1;
   trisect(mmm4,mmm3,2)=nd;
end
if((trisect(mmm3,mmm4,1)~=0)&&(trisect(mmm4,mmm3,1)==0)))
   nd=nd+1;
   trisect(mmm4,mmm3,1)=nd;
end
% trisectional points and midpoints: side-4 of 4-node quadrilateral
if((trisect(mmm4,mmm1,1)==0))
   nd=nd+1;
   trisect(mmm4,mmm1,1)=nd;
end
if(trisect(mmm4,mmm1,2)==0)
    nd=nd+1;
   trisect(mmm4,mmm1,2)=nd;
end
if(trisect(mmm4,mmm1,3)==0)
   nd=nd+1;
   trisect(mmm4,mmm1,3)=nd;
end
if(trisect(mmm4,mmm1,4)==0)
    nd=nd+1;
   trisect(mmm4,mmm1,4)=nd;
end
if((trisect(mmm4,mmm1,5)==0)&&(trisect(mmm1,mmm4,5)==0))
    nd=nd+1;
   trisect(mmm4,mmm1,5)=nd;
   trisect(mmm1,mmm4,5)=nd;
end
```

```
nd=nd+1;
   trisect(mmm1,mmm4,4)=nd;
end
if((trisect(mmm4,mmm1,3)~=0)&&(trisect(mmm1,mmm4,3)==0)))
    nd=nd+1;
   trisect(mmm1,mmm4,3)=nd;
end
if((trisect(mmm4,mmm1,2)~=0)&&(trisect(mmm1,mmm4,2)==0))
    nd=nd+1;
   trisect(mmm1,mmm4,2)=nd;
end
if((trisect(mmm4,mmm1,1)~=0)&&(trisect(mmm1,mmm4,1)==0)))
    nd=nd+1;
   trisect(mmm1,mmm4,1)=nd;
end
%assign matrices trisect to mid-side nodes of element
   qT(mmm, 5) = trisect(mmm1, mmm2, 1);
   qT(mmm, 6) = trisect(mmm1, mmm2, 2);
   qT(mmm,7)=trisect(mmm1,mmm2,3);
   qT(mmm, 8) = trisect(mmm1, mmm2, 4);
   qT(mmm,9)=trisect(mmm1,mmm2,5);
   qT(mmm,10)=trisect(mmm2,mmm1,4);
   qT(mmm, 11) = trisect(mmm2, mmm1, 3);
   qT(mmm, 12) = trisect(mmm2, mmm1, 2);
   qT(mmm, 13) = trisect(mmm2, mmm1, 1);
   qT(mmm,14)=trisect(mmm2,mmm3,1);
   qT(mmm, 15) = trisect(mmm2, mmm3, 2);
   qT(mmm, 16) = trisect(mmm2, mmm3, 3);
   qT(mmm, 17) = trisect(mmm2, mmm3, 4);
   qT(mmm, 18) = trisect(mmm2, mmm3, 5);
   qT(mmm, 19) = trisect(mmm3, mmm2, 4);
   qT(mmm,20)=trisect(mmm3,mmm2,3);
   qT(mmm,21)=trisect(mmm3,mmm2,2);
   qT(mmm, 22) = trisect(mmm3, mmm2, 1);
   qT(mmm, 23) = trisect(mmm3, mmm4, 1);
   qT(mmm,24)=trisect(mmm3,mmm4,2);
   qT(mmm, 25) = trisect(mmm3, mmm4, 3);
   qT(mmm, 26) = trisect(mmm3, mmm4, 4);
   qT(mmm, 27) = trisect(mmm3, mmm4, 5);
   qT(mmm, 28) = trisect(mmm4, mmm3, 4);
   qT(mmm, 29) = trisect(mmm4, mmm3, 3);
   qT(mmm, 30) = trisect(mmm4, mmm3, 2);
   qT(mmm, 31)=trisect(mmm4,mmm3,1);
   qT(mmm, 32) = trisect(mmm4, mmm1, 1);
   qT(mmm, 33) = trisect(mmm4, mmm1, 2);
   qT(mmm, 34) = trisect(mmm4, mmm1, 3);
   qT(mmm, 35) = trisect(mmm4, mmm1, 4);
   qT(mmm, 36) = trisect(mmm4, mmm1, 5);
   qT(mmm, 37) = trisect(mmm1, mmm4, 4);
   qT(mmm, 38) = trisect(mmm1, mmm4, 3);
   qT(mmm, 39) = trisect(mmm1, mmm4, 2);
   qT(mmm, 40) = trisect(mmm1, mmm4, 1);
   for icc=41:72
   nd=nd+1;
   qT(mmm,icc)=nd;
   end
end%for
nnode=nd;
nel=mm;
% % spqd=nodes;
MM=(1:mm)';
  disp([MM qT])
  [nel,nnel]=size(qT)
 % return
```

```
mmm5=qT (mmm, 5) ;
mmm6=qT (mmm, 6);
mmm7=qT (mmm, 7);
T Rathod IIECS Volume 09 Issue 10 Octo
```

for mmm=1:nel

mmm1=qT(mmm, 1); mmm2=qT(mmm, 2); mmm3=qT(mmm, 3); mmm4=qT(mmm, 4);

mmm8=qT(mmm,8);	
$mmm U = \alpha' l' (mmm) U l \cdot$	
11111119-Q1 (1111111, 9);	
mmm10=qT(mmm,10);	
mmm11=qT(mmm,11);	
mmm12-cm (mmm 12)	0
nunun12-q1 (nunun,12),	0
mmm13=qT(mmm,13);	
mmm14=aT(mmm,14);	
mmm15=q1(mmm,15);	
mmm16=qT(mmm,16);	
mmm17-cm (mmm 17)	
nununi /-qi (nunun, i /),	
mmm18=qT(mmm,18);	
mmm19=cT (mmm, 19);	
mmm20=q'l'(mmm,20);	
mmm21=aT(mmm,21);	
mmz = qT (mmm, 22);	
mmm23=qT(mmm,23);	
$mmm^2 A = \alpha T (mmm^2 A)$	
mmm25=q'l'(mmm,25);	
mmm26=aT(mmm,26);	
······································	
mmz = qT (mmm, 2);	
mmm28=qT(mmm,28);	
mmm29-cr (mmm 29)	
iuiuii2)-q1 (iuiuii, 2)),	
mmm30=qT(mmm,30);	
mmm31=aT(mmm,31);	
mmm 22 - mm (mmm 22) -	
mm32=qT(mm, 32);	
mmm33=qT(mmm,33);	
mm34-cm(mmm 34)	
110101134-Q1 (1101011, 34);	
mmm35=qT(mmm,35);	
$mmm36=\alpha T (mmm, 36)$	
mmm3/=q'l'(mmm,3/);	
mmm38=qT(mmm,38);	
$mm30-\alpha (mmm 30)$	
IIIIIIII.39-Q1 (IIIIIII, 39);	
mmm40=qT(mmm,40);	
mmm41 = cT (mmm, 41)	
1000011 011 (10000, 112),	
mm42 = q'1' (mmm, 42);	
mmm43=qT(mmm,43);	
mmm (1 - qT (mmm (1)))	
nunun44-q1 (nunun, 44);	
mmm45=qT(mmm,45);	
$mmm46=\alpha T (mmm, 46)$	
mmm4 /=q'l'(mmm,4 /);	
mmm48=qT(mmm,48);	
mmm (0 = cm (mmm (0))	
111111149-q1 (1111111,49);	
mmm50=qT(mmm,50);	
$mmm51 = \alpha T (mmm 51)$	
mmm52=q'l'(mmm,52);	
mmm53=aT(mmm,53);	
nuuns4=q1 (nunun, 54);	
mmm55=qT(mmm,55);	
mmmb6-cm(mmm b6)	
mmm56=qT(mmm,56);	
mmm56=qT(mmm,56); mmm57=qT(mmm,57);	
mmm56=qT(mmm,56); mmm57=qT(mmm,57); mmm58=qT(mmm,58);	
<pre>mmm56=qT(mmm,56); mmm57=qT(mmm,57); mmm58=qT(mmm,58);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 58); mmm59=qT (mmm, 59);</pre>	
<pre>mmm56=qT(mmm,56); mmm57=qT(mmm,57); mmm58=qT(mmm,58); mmm59=qT(mmm,59); mmm60=qT(mmm,60);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60);</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm58=qT (mmm,58); mmm59=qT (mmm,59); mmm60=qT (mmm,60); mmm61=qT (mmm,61);</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm58=qT (mmm,58); mmm59=qT (mmm,59); mmm60=qT (mmm,60); mmm61=qT (mmm,61); mmm62=qT (mmm,62);</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm58=qT (mmm,58); mmm59=qT (mmm,59); mmm60=qT (mmm,60); mmm61=qT (mmm,61); mmm62=qT (mmm,63);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm63=qT (mmm, 63); mmm64=qT (mmm, 64);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm64=qT (mmm, 64);</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm58=qT (mmm,58); mmm59=qT (mmm,59); mmm60=qT (mmm,60); mmm61=qT (mmm,61); mmm62=qT (mmm,63); mmm64=qT (mmm,64); mmm65=qT (mmm,65);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm59=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm65=qT (mmm, 65);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm64=qT (mmm, 64); mmm65=qT (mmm, 65); mmm66=qT (mmm, 66);</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm58=qT (mmm,58); mmm59=qT (mmm,59); mmm60=qT (mmm,60); mmm61=qT (mmm,61); mmm63=qT (mmm,63); mmm64=qT (mmm,64); mmm65=qT (mmm,65); mmm66=qT (mmm,66); mmm67=qT (mmm,67);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm66=qT (mmm, 66); mmm67=qT (mmm, 66);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm64=qT (mmm, 64); mmm65=qT (mmm, 65); mmm66=qT (mmm, 66); mmm67=qT (mmm, 67); mmm68=qT (mmm, 67);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm63=qT (mmm, 63); mmm63=qT (mmm, 64); mmm65=qT (mmm, 65); mmm66=qT (mmm, 66); mmm67=qT (mmm, 68); mmm69=qT (mmm, 68);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm63=qT (mmm, 62); mmm63=qT (mmm, 63); mmm65=qT (mmm, 66); mmm67=qT (mmm, 66); mmm68=qT (mmm, 68); mmm69=qT (mmm, 69); mmm70=qT (mmm, 70);</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm58=qT (mmm,58); mmm59=qT (mmm,59); mmm60=qT (mmm,60); mmm61=qT (mmm,61); mmm62=qT (mmm,63); mmm63=qT (mmm,63); mmm65=qT (mmm,64); mmm65=qT (mmm,65); mmm67=qT (mmm,66); mmm68=qT (mmm,69); mmm69=qT (mmm,69); mmm70=qT (mmm,70);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm57=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm65=qT (mmm, 65); mmm66=qT (mmm, 65); mmm67=qT (mmm, 67); mmm68=qT (mmm, 68); mmm69=qT (mmm, 69); mmm70=qT (mmm, 70); mmm71=qT (mmm, 71); mmm71=qT (mmm, 71);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm64=qT (mmm, 64); mmm65=qT (mmm, 66); mmm68=qT (mmm, 68); mmm69=qT (mmm, 68); mmm69=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm64=qT (mmm, 63); mmm65=qT (mmm, 65); mmm66=qT (mmm, 66); mmm67=qT (mmm, 67); mmm69=qT (mmm, 69); mmm70=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72); %</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm65=qT (mmm, 66); mmm67=qT (mmm, 66); mmm68=qT (mmm, 68); mmm68=qT (mmm, 69); mmm70=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72); %</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 57); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm65=qT (mmm, 64); mmm65=qT (mmm, 66); mmm68=qT (mmm, 68); mmm69=qT (mmm, 69); mmm70=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72); %</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm57=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm65=qT (mmm, 65); mmm66=qT (mmm, 66); mmm67=qT (mmm, 68); mmm69=qT (mmm, 69); mmm70=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72); %</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm57=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm64=qT (mmm, 64); mmm65=qT (mmm, 66); mmm67=qT (mmm, 66); mmm69=qT (mmm, 69); mmm70=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72); %</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm58=qT (mmm,57); mmm59=qT (mmm,59); mmm60=qT (mmm,60); mmm61=qT (mmm,61); mmm63=qT (mmm,62); mmm63=qT (mmm,63); mmm64=qT (mmm,63); mmm65=qT (mmm,66); mmm66=qT (mmm,66); mmm69=qT (mmm,69); mmm69=qT (mmm,70); mmm70=qT (mmm,70); mmm71=qT (mmm,71); mmm72=qT (mmm,71); % xi1=gcoord (mmm1,1); xi2=gcoord (mmm2,1);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm57=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm65=qT (mmm, 65); mmm66=qT (mmm, 65); mmm67=qT (mmm, 66); mmm67=qT (mmm, 69); mmm70=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72); % xi1=gcoord (mmm1, 1); xi2=gcoord (mmm2, 1);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm57=qT (mmm, 57); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm64=qT (mmm, 64); mmm65=qT (mmm, 66); mmm67=qT (mmm, 66); mmm69=qT (mmm, 68); mmm69=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72); % xi1=gcoord (mmm1, 1); xi2=gcoord (mmm1, 1); xi3=gcoord (mmm3, 1);</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm58=qT (mmm,57); mmm59=qT (mmm,59); mmm60=qT (mmm,60); mmm61=qT (mmm,61); mmm63=qT (mmm,62); mmm63=qT (mmm,63); mmm65=qT (mmm,63); mmm65=qT (mmm,66); mmm68=qT (mmm,66); mmm69=qT (mmm,68); mmm69=qT (mmm,69); mmm70=qT (mmm,70); mmm71=qT (mmm,71); mmm72=qT (mmm,72); % xi1=gcoord (mmm1,1); xi2=gcoord (mmm3,1); xi4=gcoord (mmm4,1);</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm57=qT (mmm,57); mmm59=qT (mmm,59); mmm60=qT (mmm,60); mmm61=qT (mmm,61); mmm62=qT (mmm,61); mmm63=qT (mmm,63); mmm64=qT (mmm,64); mmm65=qT (mmm,66); mmm67=qT (mmm,66); mmm67=qT (mmm,69); mmm70=qT (mmm,70); mmm71=qT (mmm,71); mmm72=qT (mmm,72); % xi1=gcoord (mmm1,1); xi2=gcoord (mmm3,1); xi4=gcoord (mmm4,1); %</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm57=qT (mmm,57); mmm59=qT (mmm,59); mmm60=qT (mmm,60); mmm61=qT (mmm,61); mmm62=qT (mmm,61); mmm63=qT (mmm,63); mmm64=qT (mmm,63); mmm65=qT (mmm,66); mmm67=qT (mmm,66); mmm69=qT (mmm,68); mmm70=qT (mmm,70); mmm71=qT (mmm,71); mmm72=qT (mmm,72); % xi1=gcoord (mmm1,1); xi3=gcoord (mmm3,1); xi4=gcoord (mmm1,2);</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm57=qT (mmm,57); mmm59=qT (mmm,59); mmm60=qT (mmm,60); mmm61=qT (mmm,61); mmm62=qT (mmm,61); mmm63=qT (mmm,63); mmm65=qT (mmm,63); mmm65=qT (mmm,66); mmm66=qT (mmm,66); mmm67=qT (mmm,67); mmm69=qT (mmm,70); mmm71=qT (mmm,71); mmm72=qT (mmm,72); % xi1=gcoord (mmm1,1); xi3=gcoord (mmm3,1); xi4=gcoord (mmm1,2); %</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm57=qT (mmm,57); mmm59=qT (mmm,58); mmm60=qT (mmm,60); mmm61=qT (mmm,61); mmm62=qT (mmm,61); mmm63=qT (mmm,63); mmm64=qT (mmm,63); mmm65=qT (mmm,66); mmm67=qT (mmm,67); mmm68=qT (mmm,69); mmm70=qT (mmm,70); mmm71=qT (mmm,71); mmm72=qT (mmm,72); % xi1=gcoord (mmm1,1); xi3=gcoord (mmm3,1); xi4=gcoord (mmm4,1); % yi1=gcoord (mmm1,2); yi2=gcoord (mmm2,2);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 57); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm63=qT (mmm, 64); mmm65=qT (mmm, 66); mmm67=qT (mmm, 66); mmm69=qT (mmm, 68); mmm69=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72); % xi1=gcoord (mmm1, 1); xi2=gcoord (mmm1, 1); xi4=gcoord (mmm1, 2); yi1=gcoord (mmm1, 2); yi2=gcoord (mmm1, 2); yi3=gcoord (mmm1, 2); yi3=gcoord (mmm1, 2);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm57=qT (mmm, 57); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm63=qT (mmm, 63); mmm66=qT (mmm, 65); mmm66=qT (mmm, 66); mmm67=qT (mmm, 69); mmm70=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72); % xi1=gcoord (mmm1, 1); xi3=gcoord (mmm1, 2); yi2=gcoord (mmm1, 2); yi3=gcoord (mmm2, 2); yi3=gcoord (mmm2, 2); yi3=gcoord (mmm2, 2);</pre>	
<pre>mmm56=qT (mmm, 56); mmm57=qT (mmm, 57); mmm57=qT (mmm, 57); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 61); mmm63=qT (mmm, 63); mmm64=qT (mmm, 64); mmm65=qT (mmm, 66); mmm67=qT (mmm, 66); mmm67=qT (mmm, 70); mmm70=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72); % xi1=gcoord (mmm1, 1); xi3=gcoord (mmm1, 1); xi4=gcoord (mmm1, 2); yi1=gcoord (mmm1, 2); yi1=gcoord (mmm3, 2); yi4=gcoord (mmm4, 2);</pre>	
<pre>mmmb6=qT (mmm, 56); mmm57=qT (mmm, 57); mmm58=qT (mmm, 57); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 61); mmm63=qT (mmm, 63); mmm65=qT (mmm, 64); mmm65=qT (mmm, 66); mmm68=qT (mmm, 68); mmm69=qT (mmm, 69); mmm70=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72); % xi1=gcoord (mmm1, 1); xi2=gcoord (mmm1, 1); xi4=gcoord (mmm1, 2); yi1=gcoord (mmm1, 2); yi2=gcoord (mmm1, 2); yi3=gcoord (mmm1, 2); yi4=gcoord (mmm4, 2); %</pre>	
<pre>mmmb6=qT (mmm, 56); mmm57=qT (mmm, 57); mmm57=qT (mmm, 58); mmm59=qT (mmm, 59); mmm60=qT (mmm, 60); mmm61=qT (mmm, 61); mmm62=qT (mmm, 62); mmm63=qT (mmm, 63); mmm65=qT (mmm, 66); mmm67=qT (mmm, 66); mmm67=qT (mmm, 69); mmm70=qT (mmm, 70); mmm71=qT (mmm, 70); mmm71=qT (mmm, 71); mmm72=qT (mmm, 72); % xi1=gcoord (mmm1, 1); xi2=gcoord (mmm1, 1); yi2=gcoord (mmm1, 2); yi3=gcoord (mmm3, 2); yi4=gcoord (mmm4, 2); %</pre>	
<pre>mmm56=qT (mmm,56); mmm57=qT (mmm,57); mmm57=qT (mmm,57); mmm59=qT (mmm,59); mmm60=qT (mmm,60); mmm61=qT (mmm,61); mmm62=qT (mmm,61); mmm63=qT (mmm,63); mmm64=qT (mmm,63); mmm65=qT (mmm,66); mmm67=qT (mmm,66); mmm67=qT (mmm,70); mmm70=qT (mmm,70); mmm71=qT (mmm,71); mmm72=qT (mmm,72); % xi1=gcoord (mmm1,1); xi3=gcoord (mmm1,1); xi4=gcoord (mmm1,2); yi1=gcoord (mmm1,2); yi2=gcoord (mmm3,2); yi4=gcoord (mmm4,2); %</pre>	

%compute element local coordinates for midside and interior nodes % 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

H.T.Rathod, IJECS Volume 09 Issue 10 October, 2020 Page No. 25253-25306

```
22 23 24 25 26
8

      28
      29
      30
      31
      32
      33
      34
      35
      36
      37
      38
      39
      40
      41
      42
      43
      44
      45
      46
      47
      4

      54
      55
      56
      57
      58
      59
      60
      61
      62
      63
      64
      65
      66
      67
      68
      69
      70
      71
      72

27
                                                                                                 51 52
                                                                                    48 49
                                                                                            50
53
   ui=[0;1;1;0; 0.1; 0.2; 0.3;0.4;0.5;0.6;0.7;0.8;0.9; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1;
1/4; 1/4;3/8;5/8;5/8;3/8;1/2;5/8;1/2;3/8];
   1; 1; 1; 1;0.9;0.8;0.7;0.6;0.5;0.4;0.3;0.2;0.1;1/4;1/4;3/4;3/4; 1/4; 1/4;1/4; 1/4;
1/4;4/12;5/12;1/2;7/12;8/12; 3/4; 3/4; 3/4; 3/4;
3/4;8/12;7/12;1/2;5/12;4/12;3/8;3/8;5/8;5/8;3/8;1/2;5/8;1/2];
mi=[mmm1;mmm2;mmm4;mmm5;mmm6;mmm7;mmm8;mmm9;mmm10;mmm11;mmm12;mmm13;mmm14;mmm15;mmm16;mmm17;mmm18;mmm19;
mmm20;mmm21;mmm22;mmm23;mmm24;mmm25;mmm26;mmm27;mmm28;mmm29;mmm30;mmm31;mmm32;mmm33;mmm34;mmm35;mmm36;mmm37;m
mm38; mmm39; mmm40; mmm41; mmm42; mmm44; mmm45; mmm46; mmm47; mmm48; mmm49; mmm50; mmm51; mmm52; mmm53; mmm54; mmm55; mm
m56;mmm57;mmm58;mmm60;mmm61;mmm62;mmm63;mmm64;mmm65;mmm66;mmm67;mmm68;mmm69;mmm70;mmm71;mmm72];
   for ii=5:72
       mii=mi(ii,1)
       uii=ui(ii,1);
       vii=vi(ii,1);
       gcoord(mii,1)=xi1+uii*(xi2-xi1)+vii*(xi4-xi1)+uii*vii*(xi1-xi2+xi3-xi4);
       gcoord (mii, 2)=yi1+uii*(yi2-yi1)+vii*(yi4-yi1)+uii*vii*(yi1-yi2+yi3-yi4);
   end
end%for nel
disp(gcoord)
[nnode, dimension] = size (gcoord)
  j=1:68;
   %each element coordinates for quadrilaterals at midside and interior nodes
    for n=1:NE/2
       XM(j,1)=gcoord(qT(n,5:72),1);
       YM(j,1)=gcoord(qT(n,5:72),2);
       j=j+68;
   end
   coordm=[XM YM];
                   %x and y coordinates for quadrilaterals at midside nodes
for ii=1:nnode
       r=gcoord(ii,1);s=gcoord(ii,2);
       xygcoord(ii,1)=x1+r*(x2-x1)+s*(x4-x1)+r*s*(x1-x2+x3-x4);
       xygcoord(ii,2)=y1+r*(y2-y1)+s*(y4-y1)+r*s*(y1-y2+y3-y4);
   end
   kk=0;
   for n=1:NE
       for jj=1:3
           kk=kk+1
           rr=coords(kk,1);ss=coords(kk,2);
           xycoords(kk,1)=x1+rr*(x2-x1)+ss*(x4-x1)+rr*ss*(x1-x2+x3-x4);
           xycoords(kk,2)=y1+rr*(y2-y1)+ss*(y4-y1)+rr*ss*(y1-y2+y3-y4);
       end
   end
   kk=0;
   for n=1:NE/2
       for jj=1:4
            kk=kk+1
           rr=coord(kk,1);ss=coord(kk,2);
           xycoordrgqd(kk,1)=x1+rr*(x2-x1)+ss*(x4-x1)+rr*ss*(x1-x2+x3-x4);
           xycoordrgqd(kk,2)=y1+rr*(y2-y1)+ss*(y4-y1)+rr*ss*(y1-y2+y3-y4);
       end
   end
    %coordinates for the element midside and interior nodes
   kk=0;
   for n=1:NE/2
       for jj=1:68
           kk=kk+1;
           rr=coordm(kk,1);ss=coordm(kk,2);
           xycoordrgqdm(kk,1)=x1+rr*(x2-x1)+ss*(x4-x1)+rr*ss*(x1-x2+x3-x4);
           xycoordrgqdm(kk,2)=y1+rr*(y2-y1)+ss*(y4-y1)+rr*ss*(y1-y2+y3-y4);
       end
   end
```

end