

**An explicit finite element integration scheme for linear eight node convex quadrilaterals  
using automatic mesh generation technique over plane regions**

H.T.Rathod <sup>a\*</sup>, Bharath Rathod <sup>b</sup>, K.T.Shivaram <sup>c</sup>, K. Sugantha Devi <sup>d</sup>, Tara Rathod <sup>e</sup>

<sup>a</sup> Department of Mathematics, Central College Campus,

Bangalore University, Bangalore- 560001,India

E-mail: [htrathod2010@gmail.com](mailto:htrathod2010@gmail.com)

<sup>b</sup> Xavier Institute of Management and Entrepreneurship,

Electronic City Phase-II, Hosur Road, Bangalore- 560100,India

E-mail: [rathodbharath@gmail.com](mailto:rathodbharath@gmail.com)

<sup>c</sup> Department of Mathematics, Dayananda Sagar College of engineering

Shavige Malleswara Hills, Kumar Swamy Layout, Bangalore- 560078,India

E-mail: [shivaramktshiv@gmail.com](mailto:shivaramktshiv@gmail.com)

<sup>d</sup> Department of Mathematics, Dr. T. Thimmaiah Institute of Technology, Oorgam Post,  
Kolar Gold Field, Kolar District, Karnataka state, Pin- 563120, India.

Email: [suganthadevik@yahoo.co.in](mailto:suganthadevik@yahoo.co.in)

<sup>e</sup> Department of Statistics, Jnana Bharathi Campus,

Bangalore University, Bangalore- 560056

E-mail: [rathodtara12@gmail.com](mailto:rathodtara12@gmail.com)

\* Corresponding author.

**Abstract :**

This paper presents an explicit integration scheme to compute the stiffness matrix of an eight node linear convex quadrilateral element for plane problems using symbolic mathematics and an automatic generation of all quadrilateral mesh technique. In finite element analysis, the boundary problems governed by second order linear partial differential equations, the element stiffness matrices are expressed as integrals of the product of global derivatives over the linear convex quadrilateral region. These matrices can be shown to depend on the material properties and the matrix of integrals with integrands as rational functions with polynomial numerator and the linear denominator  $(4 + \xi + \eta)$  in bivariate  $\xi, \eta$  over an eight node 2-square  $(-1 \leq \xi, \eta \leq 1)$ . In this paper, we have computed these integrals in exact and digital forms using the symbolic mathematics capabilities of MATLAB. The proposed explicit finite element integration scheme is illustrated by computing the Prandtl stress function values and the torsional constant for the square cross section by using the eight node linear convex quadrilateral finite elements. An automatic all quadrilateral mesh generation technique for the eight node linear convex quadrilaterals is also developed for this purpose. We have presented a complete program which automatically discretises the arbitrary triangular domain into all eight node linear convex quadrilaterals and applies the so generated nodal coordinate and element connection data to the above mentioned torsion problem.

**Key words:** Explicit Integration, Gauss Legendre Quadrature, Quadrilateral Element, Prandtl's Stress Function for torsion, Symbolic mathematics, all quadrilateral mesh generation technique.

**Key Words:** Explicit Integration, eight node element, Gauss Legendre Quadrature, Linear Convex Quadrilateral, Stress Function, Poisson Equation, Torsion Constant.

## 1. Introduction :

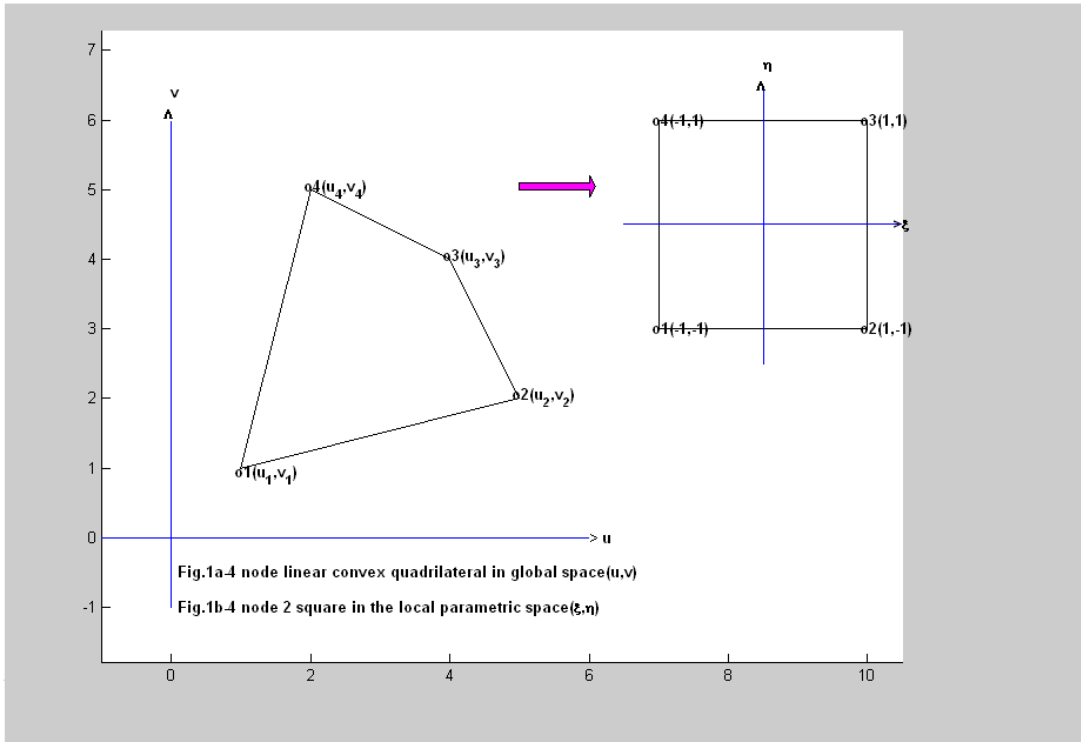
In recent years, the finite element method (FEM) has emerged as a powerful tool for the approximate solution of differential equations governing diverse physical phenomena. Today, finite element analysis is an integral and major component in many fields of engineering design and manufacturing. Its use in industry and research is extensive, and indeed without it many practical problems in science, engineering and emerging technologies such as nanotechnology, biotechnology, aerospace, chemical, etc. would be incapable of solution [1,2,3]. This method is very adaptable and physically satisfying, but it is very difficult to implement. This is in contrast to finite difference methods, which are reasonably easy. In FEM, various integrals are to be determined numerically in the evaluation of stiffness matrix, mass matrix, body force vector, etc. The algebraic integration needed to derive explicit finite element relations for second and higher order continuum mechanics problems generally defies our analytic skill and in most cases, it appears to be a prohibitive task. Hence, from a practical point of view, numerical integration scheme is not only necessary but very important as well. Among various numerical integration schemes, Gaussian quadrature, which can evaluate exactly the  $(2n-1)^{\text{th}}$  order polynomial with  $n$  Gaussian integration points, is mostly used in view of the accuracy and efficiency of calculation. However, the integrands of global derivative products in stiffness matrix computations of practical applications are not always simple polynomials but rational expressions which the Gaussian quadrature cannot evaluate exactly [7-16]. The integration points have to be increased in order to improve the integration accuracy but it is also desirable to make these evaluations by using as few Gaussian points as possible, from the point of view of the computational efficiency. Thus it is an important task to strike a proper balance between accuracy and economy in computation. Therefore, analytical integration is essential to generate a smaller error as well as to save the computational costs of Gaussian quadrature commonly applied for solving science, engineering and technical problems. In explicit integration of stiffness matrix, complications arise from two main sources, firstly the large number of integrations that need to be performed and secondly, in methods which use isoparametric or subparametric or superparametric elements, the presence of determinant of the Jacobian matrix ( we refer this as Jacobian here after ) in the denominator of the element matrix integrands. This problem is considered in the recent work [17-18] for the linear convex quadrilateral which proposes a new discretisation method and use of pre computed universal numeric arrays which do not depend on element size and shape. In this method a linear polygon is discretized into a set of linear triangles and then each of these triangles is further discretised into three linear convex quadrilateral elements by joining the centroid to the mid-point of sides. These quadrilateral elements are then mapped into 2-squares ( $-1 \leq \xi, \eta \leq 1$ ) in the natural space  $(\xi, \eta)$  to

obtain the same expression of the Jacobian, namely  $c(4 + \xi + \eta)$  where  $c$  is some appropriate constant which depends on the geometric data for the triangle.

In the present paper, we propose a similar discretisation method for linear polygon in Cartesian two space  $(x,y)$ . This discretisation is carried in two steps, We first discretise the linear polygon into a set of linear triangles in the Cartesian space  $(x,y)$  and these linear triangles are then mapped into standard triangle in a local  $(u,v)$  space. We further discretise the standard triangles into three linear quadrilaterals by joining the centroid to the midpoints of triangles in  $(u,v)$  space which are finally mapped into 2-square in the local  $(\xi, \eta)$  space. We then establish a derivative product relation between the linear convex quadrilaterals in the Cartesian space,  $(x,y)$  which are interior to an arbitrary triangle and the linear convex quadrilaterals in the local space  $(u,v)$  interior to the standard triangle. In this scheme, all computations in the local space  $(u,v)$  for product of global derivative integrals are free from geometric properties and hence they are pure numbers. We then propose a numerical scheme to integrate the products of global derivatives. We have shown that the matrix product of global derivative integrals is expressible as matrix triple product comprising of geometric properties matrices and the product of local derivative integrals matrix. The explicit integration of the product of local derivative is now possible but it is very tedious and cumbersome for hand calculations. However, we can compute such integrals accurately and efficiently by use of symbolic integration functions available in the leading mathematical softwares, such as, MATLAB, MAPLE, MATHEMATIKA etc. In this paper, we have used MATLAB symbolic mathematics to compute the integrals of the products of local derivatives in  $(u, v)$  space. This has been demonstrated in authors' resent paper [18] for four node linear convex quadrilateral elements under isoparametric transformations between the global spaces  $(x, y)$ ,  $(u, v)$  and local space  $(\xi, \eta)$ . In the present paper, we have proposed these computations for eight node linear convex quadrilateral elements under the subparametric transformations between the global spaces  $(x, y)$ ,  $(u, v)$  and the local space  $(\xi, \eta)$ . The proposed explicit integration scheme is shown as a useful technique in the formation of element stiffness matrices for second order boundary value problems governed by partial differential equations. This is demonstrated by applying the above proposed technique to the linear elastic torsion problem for the square cross section.

## **2. Linear Convex Quadrilateral Elements :**

Let us first consider an arbitrary four noded linear convex quadrilateral in the global (Cartesian) coordinate system  $(u, v)$  as in Fig 1a, which mapped into a 2-square in the local(natural) parametric coordinate  $(\xi, \eta)$  as in Fig 1b.



$$\begin{pmatrix} u \\ v \end{pmatrix} = \sum_{k=1}^4 \begin{pmatrix} u_k \\ v_k \end{pmatrix} M_k(\xi, \eta) \quad \text{----- (1)}$$

Where  $(u_k, v_k)$ ,  $(k=1,2,3,4)$  are the vertices of the original arbitrary linear convex quadrilateral in  $(u, v)$  plane and  $M_k(\xi, \eta)$  denote the well known bilinear basis functions [1-3] in the local parametric space  $(\xi, \eta)$  and they are given by  $M_k(\xi, \eta) = \frac{1}{4} (1 + \xi\xi_k)(1 + \eta\eta_k)$ ,  $k = 1, 2, 3, 4$  -----

----- (2a)

Where  $\{(\xi_k, \eta_k), k = 1, 2, 3, 4\} = \{(-1, -1), (1, -1), (1, 1), (-1, 1)\}$  ----- (2b) described two transformations over a linear convex quadrilateral element from the original global space into the local parametric space.

### 2.1 Isoparametric Transformation :

For the isoparametric coordinate transformation over the linear convex quadrilateral element as shown in Fig 1, we select the field variables, say  $\phi, \psi$ , etc governing the physical problem as

$$\begin{pmatrix} \phi \\ \psi \end{pmatrix} = \sum_{k=1}^4 \begin{pmatrix} \phi_k \\ \psi_k \end{pmatrix} N_k^e(\xi, \eta) \quad \text{----- (3)}$$

Where  $\phi_k, \psi_k$  refer to unknowns at node  $k$  and the shape functions  $N_k^e = M_k$ , and  $M_k$  are defined as in Eqn.(2a-b)

### 2.2 Subparametric Transformation :

For the subparametric transformation over the  $n$  - noded element we define the field variables  $\phi, \psi$  (say) governing the physical problem as

$$\begin{pmatrix} \phi \\ \psi \end{pmatrix} = \sum_{k=1}^{nde} \begin{pmatrix} \phi_k^e \\ \psi_k^e \end{pmatrix} N_k^e(\xi, \eta) \quad \text{----- (4)}$$

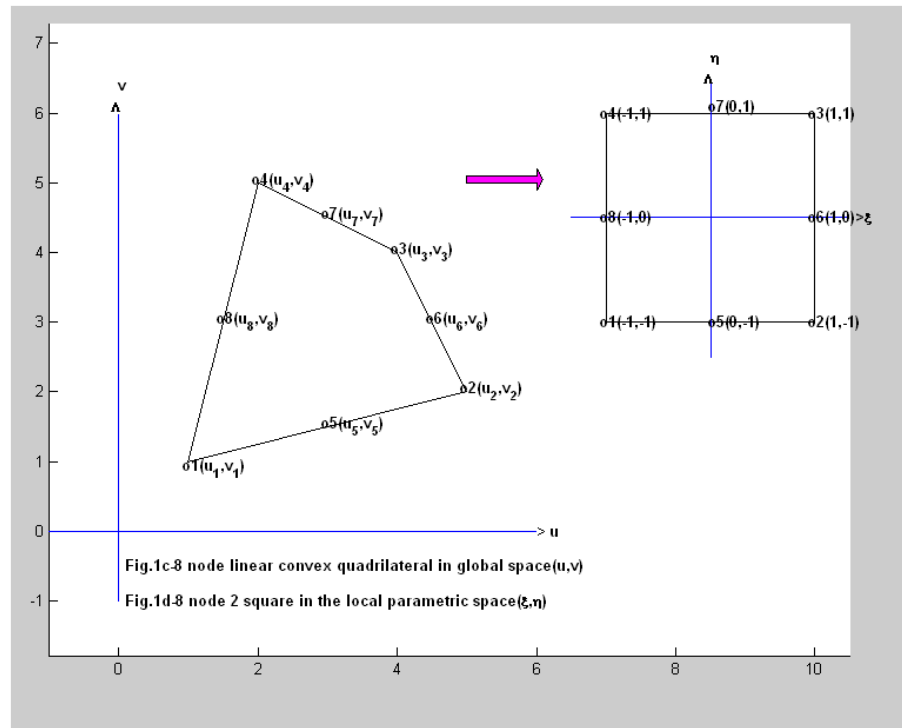
Where  $\phi_k, \psi_k$  refer to unknowns at node  $k$  and  $nde > 4$

In our recent paper, the explicit finite element integration scheme is presented by using the isoparametric transformation over the 4 node linear convex quadrilateral element

In the present paper, we consider the subparametric transformation for a linear convex quadrilateral element for which  $nde = 8$ , a eight noded (serendipity type 2 square)

### 3. Eight Node Linear Convex Quadrilateral Element :

In this section, we give a brief description of the 8- node quadrilateral element under subparametric transformation as shown in Fig 1c , Fig 1d.



We use the transform of Eqns.(1-2) to define the element geometry i.e.

$$\begin{pmatrix} u(\xi, \eta) \\ v(\xi, \eta) \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} = \sum_{k=1}^4 \begin{pmatrix} u_k \\ v_k \end{pmatrix} M_k(\xi, \eta) \quad \text{----- (1)}$$

$$\text{Where } M_k(\xi, \eta) = \frac{1}{4} (1 + \xi \xi_k)(1 + \eta \eta_k), \quad (k = 1, 2, 3, 4) \quad \text{----- (2a)}$$

With  $(u(\xi_k, \eta_k), v(\xi_k, \eta_k)), \quad k = 1, 2, 3, 4$  are the vertices of the linear convex quadrilateral in global  $(u, v)$  space .

$$\{ (\xi_k, \eta_k), k = 1, 2, 3, 4 \} = \{ (-1, -1), (1, -1), (1, 1), (-1, 1) \} \quad \text{----- (2b)}$$

Using the transformation of Eqns.(1-2) and from Fig 1c , Fig 1d we see that there is a one to one correspondence between  $((\xi_k, \eta_k), k = 5,6,7,8) = ((0, 1), (1, 0), (0, 1), (-1, 0))$

and  $((u_k, v_k) = (u(\xi_k, \eta_k), v(\xi_k, \eta_k)), k = 5,6,7,8)$  , where

$$(u_5, v_5) = ( (u_1 + u_2)/2 , (v_1 + v_2)/2 )$$

$$(u_6, v_6) = ( (u_2 + u_3)/2 , (v_2 + v_3)/2 )$$

$$(u_7, v_7) = ( (u_3 + u_4)/2 , (v_3 + v_4)/2 )$$

$$(u_8, v_8) = ( (u_1 + u_4)/2 , (v_1 + v_4)/2 ) \text{ -----(2c)}$$

We then define the variation of physical variables  $\phi^e, \psi^e$  (say) over 8- node element of Fig 1c , 1d by Eqn.(4) with nde = 8

$$\begin{pmatrix} \phi^e \\ \psi^e \end{pmatrix} = \sum_{k=1}^8 N_k^e(\xi, \eta) \begin{pmatrix} \phi_k^e \\ \psi_k^e \end{pmatrix} \text{ ----- (4)}$$

Where  $\phi_k^e, \psi_k^e$  are the nodal values at node k

The shape functions  $N_i^e$  of the 8- node element shown in Fig 1c , Fig 1d are given by  $N_i^e(\xi, \eta) = \frac{1}{4} (1 + \xi\xi_k)(1 + \eta\eta_k)(-1 + \xi\xi_k + \eta\eta_k)$  ,  $i = 1, 2, 3, 4$

$$N_i^e(\xi, \eta) = \frac{1}{2} (1 - \xi^2)(1 + \eta\eta_k) , i = 5, 7$$

$$N_i^e(\xi, \eta) = \frac{1}{2} (1 + \xi\xi_k)(1 - \eta^2) , i = 6, 8 \text{ -----(5)}$$

and  $\{(\xi_k, \eta_k), k = 1(1)8\} = \{(-1, -1), (1, -1), (1, 1), (-1, 1), (0, -1), (1, 0), (0, 1), (-1, 0)\}$

#### 4. Explicit Form of the Jacobian and Global Derivatives :

##### 4.1 Jacobian

Let us consider an arbitrary linear convex quadrilateral in the global Cartesian space (u, v) as in Fig 1a , c which is mapped into a 8- node 2- square in the local parametric space  $(\xi, \eta)$  as in Fig 1b, d

From the Eq.(1) and Eq.(2), we have

$$\frac{\partial u}{\partial \xi} = \sum_{k=1}^4 u_k \frac{\partial M_k}{\partial \xi} = \frac{1}{4} [ (-u_1 + u_2 + u_3 - u_4) + (u_1 - u_2 + u_3 - u_4) \eta ] \text{ ----- (6a)}$$

$$\frac{\partial u}{\partial \eta} = \sum_{k=1}^4 u_k \frac{\partial M_k}{\partial \eta} = \frac{1}{4} [ (-u_1 - u_2 + u_3 + u_4) + (u_1 - u_2 + u_3 - u_4) \xi ] \text{ ----- (6b)}$$

$$\frac{\partial v}{\partial \xi} = \frac{1}{4} [ (-v_1 + v_2 + v_3 - v_4) + (v_1 - v_2 + v_3 - v_4) \eta ] \text{ ----- (6c)}$$

$$\frac{\partial v}{\partial \eta} = \frac{1}{4} [ (-v_1 - v_2 + v_3 + v_4) + (v_1 - v_2 + v_3 - v_4) \xi ] \text{ ----- (6d)}$$

Hence the Jacobian, J can be expressed as [1, 2, 3]

$$J = \frac{\partial(u,v)}{\partial(\xi,\eta)} = \frac{\partial u}{\partial \xi} \frac{\partial v}{\partial \eta} - \frac{\partial u}{\partial \eta} \frac{\partial v}{\partial \xi} = \alpha + \beta\xi + \gamma\eta \text{ ----- (7a)}$$

Where

$$\alpha = \frac{1}{8} [ (u_4 - u_2)(v_1 - v_3) + (u_3 - u_1)(v_4 - v_2) ]$$

$$\beta = \frac{1}{8} [ (u_4 - u_3)(v_2 - v_1) + (u_1 - u_2)(v_4 - v_3) ]$$

$$\gamma = \frac{1}{8} [ (u_4 - u_1)(v_2 - v_3) + (u_3 - u_2)(v_4 - v_1) ] \quad \text{----- (7b)}$$

#### 4.2 Global Derivatives:

If  $N_i^e$  denotes the basis functions of node  $i$  of any order of the element  $e$ , then the chain rule of differentiation from Eq.(1) we can write the global derivative as in [1, 2, 3]

$$\begin{pmatrix} \frac{\partial N_i^e}{\partial u} \\ \frac{\partial N_i^e}{\partial v} \end{pmatrix} = \frac{1}{J} \begin{bmatrix} \frac{\partial v}{\partial \eta} & -\frac{\partial v}{\partial \xi} \\ -\frac{\partial u}{\partial \eta} & \frac{\partial u}{\partial \xi} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \end{bmatrix} \quad \text{----- (8)}$$

Where  $\frac{\partial u}{\partial \xi}$ ,  $\frac{\partial u}{\partial \eta}$ ,  $\frac{\partial v}{\partial \xi}$  and  $\frac{\partial v}{\partial \eta}$  are defined as in Eqs.(6a)–(6d) while  $J$  is defined in Eq.(7a-b) , (  $i, j = 1, 2, 3, \dots, nde$  ) ,  $nde =$  the number of nodes per element. We may recall that the explicit integration for linear convex quadrilateral with  $nde = 4$  is already presented by the authors in their recent paper [18]. We take  $nde = 8$  for the present study.

#### 5. Discretisation of an Arbitrary Triangle :

A linear convex polygon in the physical plane  $(x, y)$  can be always discretised into a finite number of linear triangles. However, we would like to study a particular discretization of these triangles further into linear convex quadrilaterals. This is stated in the following Lemma [ 6 ].

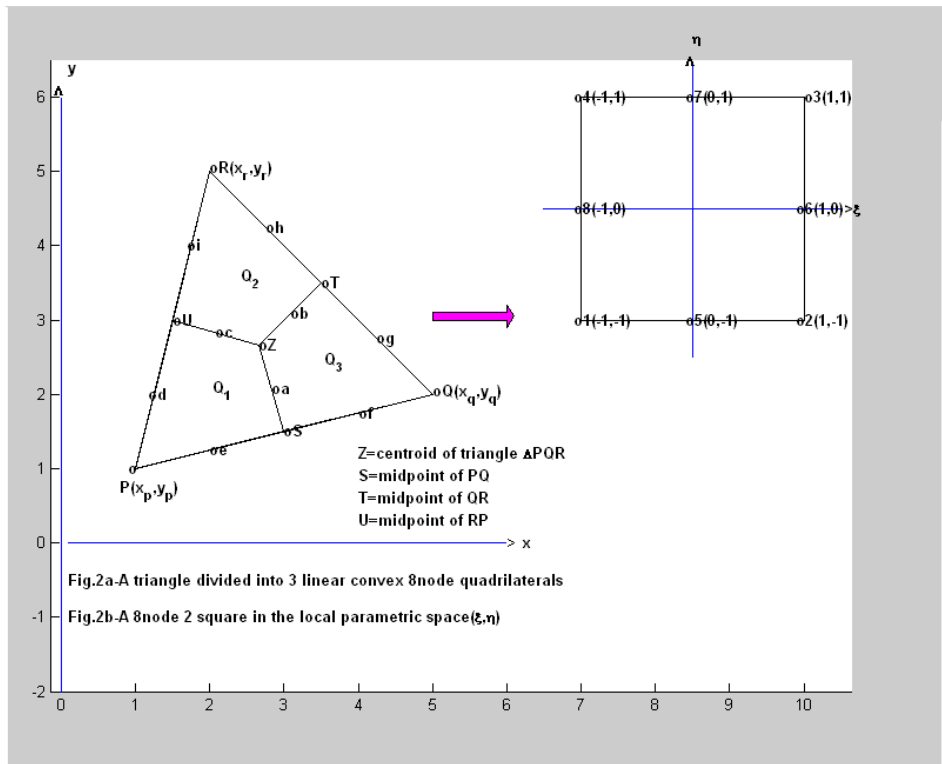
**Lemma 1.** Let  $\Delta PQR$  be an arbitrary triangle with the vertices  $P(x_p, y_p)$  ,  $Q(x_q, y_q)$  and  $R(x_r, y_r)$  and  $S, T, U$  be the midpoints of sides  $PQ, QR$  and  $RP$  respectively, let  $a, b, c, d, e, f, g, h, I$  be the midpoints of sides  $ZS, ZT, ZU, PU, PS, QS, QT, RT, RU$  and let  $Z$  be the centroid of the triangle  $\Delta PQR$ . We can obtain three linear convex 8- node quadrilaterals ( $Q_e, e=1,2,3$ ), where  $Q_1 = ZcUdPeS$  ,  $Q_2 = ZaSfQgT$  and  $Q_3 = ZbThRiU$  from triangle  $\Delta PQR$  as shown in Fig 2a,b. If we map each of these 8- node linear convex quadrilaterals into 8- node 2- squares in which the nodes are oriented in counter clockwise from  $Z$ , then the Jacobian  $J^e$  for each element  $Q_e$ , ( $e=1,2,3$ ) is given by

$$J = J^e = \frac{1}{48} \Delta pqr (4 + \xi + \eta), \quad e = 1, 2, 3 \quad \text{----- (9)}$$

Where  $\Delta pqr$  is the area of the triangle  $\Delta PQR$

$$2\Delta pqr = \begin{vmatrix} 1 & x_p & y_p \\ 1 & x_q & y_q \\ 1 & x_r & y_r \end{vmatrix} = [ (x_p - x_r)(y_q - y_r) - (x_q - x_r)(y_p - y_r) ] \quad \text{----- (10)}$$





Proof : Proof is straight forward and it can be elaborated on the lines of proof given in [17].

Lemma 2. Let  $\Delta PQR$  be the arbitrary linear triangle with the vertices  $P(x_p, y_p)$ ,  $Q(x_q, y_q)$  and  $R(x_r, y_r)$  and  $S, T, U$  be the midpoints of sides  $PQ, QR,$  and  $RP$  respectively. Further, let  $a, b, c, d, e, f, g, h, i$  be the midpoints of sides  $ZS, ZT, ZU, PU, PS, QS, QT, RT, RU$  and  $Z$  be the centroid of the  $\Delta PQR$ . Then we obtain three linear convex 8- node quadrilaterals  $Q_e$  ( $e=1,2,3$ ),  $Q_1 = \langle ZcUdPeSa \rangle$ ,  $Q_2 = \langle ZaSfQgTb \rangle$  and  $Q_3 = \langle ZbThRiUc \rangle$ , these quadrilaterals can be mapped into the linear convex 8- node quadrilateral spanning the vertices  $GHEICJF$  with  $G(1/3, 1/3), H(1/6, 5/12), E(0, 1/2), I(0, 1/4), C(0, 0), J(1/4, 0), F(1/2, 0), K(5/12, 1/6)$  in the interior of the right isosceles triangle  $\Delta ABC$  with vertices  $A(1, 0), B(0, 1)$  and  $C(0, 0)$  in the  $(u, v)$  space as shown in Fig3a and Fig3b.

Proof : The sum of the three quadrilaterals  $Q_1, Q_2, Q_3$  is  $Q_1 + Q_2 + Q_3 = \Delta PQR$  as shown in Fig 2a & Fig 3a.

We know that the linear transformations

$$\begin{pmatrix} x^{(1)} \\ y^{(1)} \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} w + \begin{pmatrix} x_q \\ y_q \end{pmatrix} u + \begin{pmatrix} x_r \\ y_r \end{pmatrix} v \quad \text{----- (11)}$$

$$\begin{pmatrix} x^{(2)} \\ y^{(2)} \end{pmatrix} = \begin{pmatrix} x_q \\ y_q \end{pmatrix} w + \begin{pmatrix} x_r \\ y_r \end{pmatrix} u + \begin{pmatrix} x_p \\ y_p \end{pmatrix} v \quad \text{----- (12)}$$

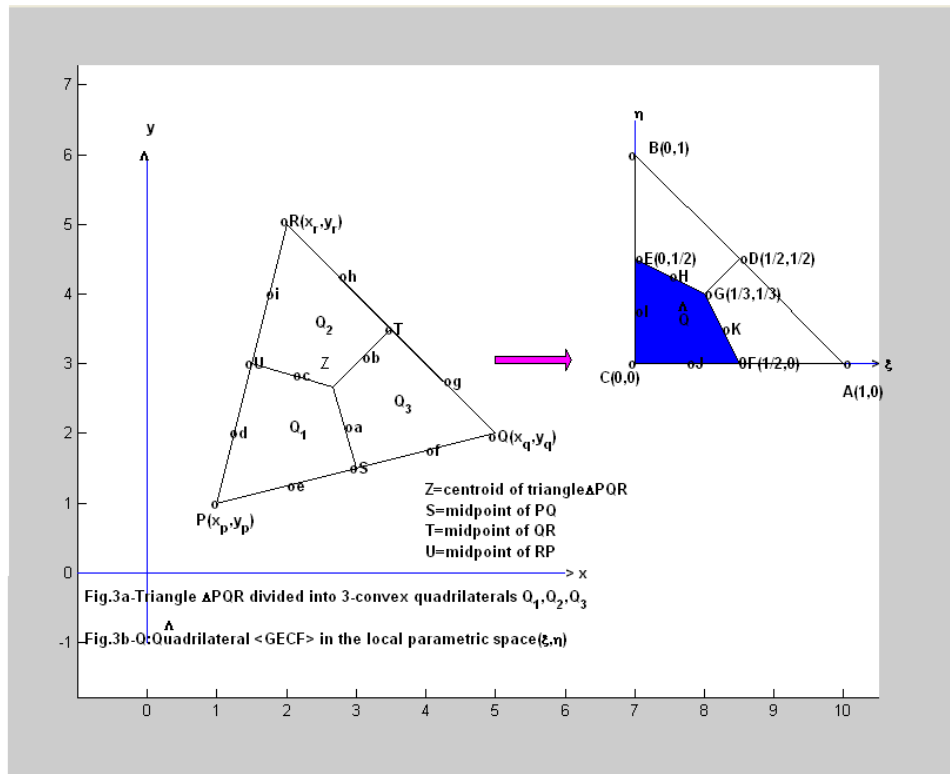
$$\begin{pmatrix} x^{(3)} \\ y^{(3)} \end{pmatrix} = \begin{pmatrix} x_r \\ y_r \end{pmatrix} w + \begin{pmatrix} x_p \\ y_p \end{pmatrix} u + \begin{pmatrix} x_q \\ y_q \end{pmatrix} v \quad \text{----- (13)}$$

with  $w = 1 - u - v$

map the arbitrary triangle  $\Delta PQR$  into a linear right isosceles triangle  $A(1, 0), B(0, 1)$  and  $C(0, 0)$  in the  $uv$ -plane. We can now verify that the vertices  $Z, c, U, d, P, e, S$  in  $xy$  plane is mapped into the linear

convex 8- node quadrilateral spanning the vertices G, H, E, I, C, J, F, K by use of the transformation given in Eqn.(11).

Similarly, we see that the linear convex 8- node quadrilateral  $Q_2$  spanned by vertices Z, a, S, f, Q, g, T, b is mapped into the linear convex 8- node quadrilateral spanned by the vertices G, H, E, I, C, J, F, K by use of the transformation of Eqn.(12). Finally the quadrilateral  $Q_3$  in xy plane is mapped into the quadrilateral GHEICJFK in uv- plane by use of the linear transformation of Eqn.(13), This completes the proof.



We have shown in the foregoing Lemma that an arbitrary linear triangle can be discretised into three linear convex 8- node quadrilaterals. Further, each of these quadrilaterals in  $xy$  plane can be mapped into a unique linear convex 8- node quadrilateral spanned by the vertices  $G(1/3, 1/3)$ ,  $H(1/6, 5/12)$ ,  $E(0, 1/2)$ ,  $I(0, 1/4)$ ,  $C(0, 0)$ ,  $J(1/4, 0)$ ,  $F(1/2, 0)$  and  $K(5/12, 1/6)$  (see Fig 3a, Fig 3b) using a proper linear transformation as given Eqn.(11) – (13).

## 6. Integration over a Triangle Region :

### 6.1 Composite Integration

We shall now establish a composite integration formula for an arbitrary triangular region  $\Delta PQR$  shown in Fig 2a or Fig 3a. Let  $\phi(x, y)$  be an arbitrary and smooth function defined over the region  $\Delta PQR$ . We now consider

$$I_{\Delta PQR} = \iint_{\Delta PQR} \phi(x, y) \, dx \, dy = \sum_{e=1}^3 \iint_{Q_e} \phi(x, y) \, dx \, dy \quad \text{----- (14)}$$

$$= \iint_{\bar{Q}} \sum_{e=1}^3 \left[ \phi(x^{(e)}(u, v), y^{(e)}(u, v)) \frac{\partial(x^{(e)}(u, v), y^{(e)}(u, v))}{\partial(u, v)} \right] \, du \, dv$$

$$= (2 \Delta_{pqr}) \iint_{\widehat{Q}} \{ \sum_{e=1}^3 [\Phi (x^{(e)}(u, v), y^{(e)}(u, v))] \} dudv \quad \text{----- (15)}$$

Where  $(x^{(e)}(u, v), y^{(e)}(u, v)), e = 1,2,3$  are the linear transformations of Eqs.(11)–(13) and  $\widehat{Q}$  is the linear convex 8- node quadrilateral GHEICJFK spanning the vertices G(1/3, 1/3), H(1/6, 5/12), E(0, 1/2), I(0, 1/4), C(0, 0), J(1/4, 0), F(1/2, 0) and K(5/12, 1/6) and  $\Delta_{pqr}$  is the area of triangle  $\Delta$  PQR, Now, we further use the bilinear transformation of Eqns.(1)–(2) in Eqn.(15) and obtain.

$$\|_{\Delta PQR} = (2 \Delta_{pqr}) \int_{-1}^1 \int_{-1}^1 \{ \sum_{e=1}^3 [\Phi (x^{(e)}(u, v), y^{(e)}(u, v)) \frac{\partial(u,v)}{\partial(\xi,\eta)}] \} d\xi d\eta \quad \text{----- (16)}$$

In Eq.(16) we have used the bilinear transformation given in Eqns.(1)- (2)

$$u = u(\xi, \eta) = \frac{1}{3} M_1(\xi, \eta) + \frac{1}{2} M_4(\xi, \eta)$$

$$v = v(\xi, \eta) = \frac{1}{3} M_1(\xi, \eta) + \frac{1}{2} M_2(\xi, \eta) \quad \text{----- (17)}$$

to map the arbitrary linear convex 8- noded quadrilateral into a 2 – square in  $(\xi, \eta) –$  plane. Thus on using Eqn.(17), the integral of Eqn.(16) simplifies to the following.

$$\|_{\Delta PQR} = (2 \Delta_{pqr}) \int_{-1}^1 \int_{-1}^1 [ \sum_{e=1}^3 \left( \frac{4+\xi+\eta}{96} \right) \Phi(x^{(e)}(u, v), y^{(e)}(u, v))] d\xi d\eta \quad \text{----- (18)}$$

We can evaluate Eqn.(18) either analytically or numerically depending on the form of the integrand.

Using Numerical Integration , we have from Eqn.(18)

$$\|_{\Delta PQR} = 2\Delta_{pqr} \sum_{i=1}^N \sum_{j=1}^N \left( \frac{W_i^{(N)} W_j^{(N)} (4+\xi_i^{(N)} + \eta_j^{(N)})}{96} \right) \sum_{e=1}^3 \Phi(x^{(e)}(u_{i,j}^{(N)}, v_{i,j}^{(N)}), y^{(e)}(u_{i,j}^{(N)}, v_{i,j}^{(N)})) \quad \text{----- (19)}$$

Where from Eqn.(17), we write

$$u_{i,j}^{(N)} = u(\xi_i^{(N)}, \eta_j^{(N)})$$

$$v_{i,j}^{(N)} = v(\xi_i^{(N)}, \eta_j^{(N)}) \quad \text{----- (20)}$$

and  $(W_i^{(N)}, \xi_i^{(N)}) , (W_j^{(N)}, \eta_j^{(N)})$  are the weight coefficients and sampling points along  $\xi, \eta$  directions of the  $N^{\text{th}}$  order Gauss Legendre quadrature rules. We could also use Gauss Labatto quadrature rules as well to evaluate the integral of Eqn.(18).

The above composite rule is applied to numerical Integration over polygonal domains using convex quadrangulation and Gauss Legendre Quadrature Rules[27].

In the next section 6.2, we shall apply the above derivations and compute the integral of eqn.(14) by assuming the integrand  $\phi(x, y)$  as the product of global derivatives, which are not explicit function of global variates  $(x, y)$

## 6.2 Global Derivative Integrals :

If  $N_i^{(e)}$  ( $i = 1(1)8$ ) denotes the basis functions for node  $i$  of a linear convex 8- node linear convex quadrilateral element  $e$ , then by use of chain rule of partial differentiation

$$\begin{pmatrix} \frac{\partial N_1^e}{\partial x} \\ \frac{\partial N_1^e}{\partial y} \end{pmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} \end{bmatrix} \begin{pmatrix} \frac{\partial N_1^e}{\partial u} \\ \frac{\partial N_1^e}{\partial v} \end{pmatrix} \quad \text{----- (21)}$$

We note that to transform 8- node linear convex quadrilateral  $Q_e (e = 1,2,3)$  of  $\Delta PQR$  in Cartesian space  $(x,y)$  into  $\hat{Q}$ , the 8- node linear convex quadrilateral spanned by vertices  $(1/3, 1/3)$ ,  $(1/6, 5/12)$ ,  $(0, 1/2)$ ,  $(0, 1/4)$ ,  $(0, 0)$ ,  $(1/4, 0)$ ,  $(1/2, 0)$  and  $(5/12, 1/6)$  in  $uv$ -plane.

We must now use the earlier transformations.

$$\begin{pmatrix} x^1 \\ y^1 \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} + \begin{pmatrix} x_q - x_p \\ y_q - y_p \end{pmatrix} u + \begin{pmatrix} x_r - x_p \\ y_r - y_p \end{pmatrix} v \quad \text{for } Q_1 \text{ in } \Delta PQR \quad \text{----- (11)}$$

$$\begin{pmatrix} x^2 \\ y^2 \end{pmatrix} = \begin{pmatrix} x_q \\ y_q \end{pmatrix} + \begin{pmatrix} x_r - x_q \\ y_r - y_q \end{pmatrix} u + \begin{pmatrix} x_p - x_q \\ y_p - y_q \end{pmatrix} v \quad \text{for } Q_2 \text{ in } \Delta PQR \quad \text{----- (12)}$$

$$\begin{pmatrix} x^3 \\ y^3 \end{pmatrix} = \begin{pmatrix} x_r \\ y_r \end{pmatrix} + \begin{pmatrix} x_p - x_r \\ y_p - y_r \end{pmatrix} u + \begin{pmatrix} x_q - x_r \\ y_q - y_r \end{pmatrix} v \quad \text{for } Q_3 \text{ in } \Delta PQR \quad \text{----- (13)}$$

And we note that the above transformations viz Eqns.(11)-(13) are of the form

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \end{pmatrix} + \begin{pmatrix} x_a - x_c \\ y_a - y_c \end{pmatrix} u + \begin{pmatrix} x_b - x_c \\ y_b - y_c \end{pmatrix} v \quad \text{----- (22)}$$

which can map an arbitrary triangle  $\Delta ABC$ ,  $A(x_a, y_a)$ ,  $B(x_b, y_b)$ ,  $C(x_c, y_c)$  in  $xy$  – plane into a right isosceles triangle in the  $uv$  – plane

Hence, we have from Eqn.(22)

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} (x_a - x_c) & (x_b - x_c) \\ (y_a - y_c) & (y_b - y_c) \end{pmatrix}^{-1} \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix} \quad \text{----- (23)}$$

This gives

$$u = (\alpha_a + \beta_a x + \gamma_a y) / (2 \Delta_{abc})$$

$$v = (\alpha_b + \beta_b x + \gamma_b y) / (2 \Delta_{abc}) \quad \text{----- (24)}$$

where

$$\begin{aligned} \alpha_a &= (x_b y_c - x_c y_b) , & \alpha_b &= (x_c y_a - x_a y_c) , \\ \beta_a &= (y_b - y_c) , & \beta_b &= (y_c - y_a) , \\ \gamma_a &= (x_c - x_b) , & \gamma_b &= (x_a - x_c) , \end{aligned} \quad \text{----- (25a)}$$

and

$$\begin{aligned} \frac{\partial(x,y)}{\partial(u,v)} &= 2\Delta_{abc} = \begin{vmatrix} 1 & x_a & y_a \\ 1 & x_b & y_b \\ 1 & x_c & y_c \end{vmatrix} = 2 * \text{area of the triangle } \Delta ABC \\ &= (\gamma_b \beta_a - \gamma_a \beta_b) \quad \text{----- (25b)} \end{aligned}$$

From Eqn.(21) and Eqn.(24), we obtain

$$\begin{pmatrix} \frac{\partial N_1^e}{\partial x} \\ \frac{\partial N_1^e}{\partial y} \end{pmatrix} = \begin{pmatrix} \beta_a^* & \beta_b^* \\ \gamma_a^* & \gamma_b^* \end{pmatrix} \begin{pmatrix} \frac{\partial N_1^e}{\partial u} \\ \frac{\partial N_1^e}{\partial v} \end{pmatrix} \quad \text{----- (26a)}$$

where  $\beta_a^* = \frac{\beta_a}{(2\Delta_{abc})}$  ,  $\beta_b^* = \frac{\beta_b}{(2\Delta_{abc})}$   
 $\gamma_a^* = \frac{\gamma_a}{(2\Delta_{abc})}$  ,  $\gamma_b^* = \frac{\gamma_b}{(2\Delta_{abc})}$  ----- (26b)

Letting,

$$D_{x,y}^{i,e} = \begin{pmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \end{pmatrix} , P = \begin{pmatrix} \beta_a^* & \beta_b^* \\ \gamma_a^* & \gamma_b^* \end{pmatrix} , D_{u,v}^{i,e} = \begin{pmatrix} \frac{\partial N_i^e}{\partial u} \\ \frac{\partial N_i^e}{\partial v} \end{pmatrix} ----- (27)$$

We obtain from Eqn.(26) and Eqn.(27)

$$D_{x,y}^{i,e} = P D_{u,v}^{i,e} ----- (28)$$

Hence from Eqn.(27) and Eqn.(28)

$$G_{x,y}^{i,j,e} = \begin{pmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \end{pmatrix} \begin{pmatrix} \frac{\partial N_j^e}{\partial x} & \frac{\partial N_j^e}{\partial y} \end{pmatrix} = (D_{x,y}^{i,e}) (D_{x,y}^{j,e})^T$$

$$= \begin{pmatrix} \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} & \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial y} \\ \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial x} & \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} \end{pmatrix} ----- (29a)$$

$$G_{u,v}^{i,j,e} = \begin{pmatrix} \frac{\partial N_i^e}{\partial u} \\ \frac{\partial N_i^e}{\partial v} \end{pmatrix} \begin{pmatrix} \frac{\partial N_j^e}{\partial u} & \frac{\partial N_j^e}{\partial v} \end{pmatrix} = (D_{u,v}^{i,e}) (D_{u,v}^{j,e})^T$$

$$= \begin{pmatrix} \frac{\partial N_i^e}{\partial u} \frac{\partial N_j^e}{\partial u} & \frac{\partial N_i^e}{\partial u} \frac{\partial N_j^e}{\partial v} \\ \frac{\partial N_i^e}{\partial v} \frac{\partial N_j^e}{\partial u} & \frac{\partial N_i^e}{\partial v} \frac{\partial N_j^e}{\partial v} \end{pmatrix} ----- (29b)$$

We have now from Eqn.(28) and Eqn.(29a- b)

$$G_{x,y}^{i,j,e} = (P D_{u,v}^{i,e}) (D_{u,v}^{j,e} P)^T$$

$$= P (D_{u,v}^{i,e}) (D_{u,v}^{j,e})^T P^T$$

$$= P G_{u,v}^{i,j,e} P^T ----- (29c)$$

We now define the submatrices of global derivative integrals in (x,y) and (u,v) space associated with the nodes i and j (i, j = 1, 2, 3, 4, 5, 6, 7, 8) as

$$S^{i,j,e} = \iint_{Q_e} G_{x,y}^{i,j,e} dx dy , ----- (30) \quad K^{i,j,e} = \iint_{\hat{Q}} G_{u,v}^{i,j,e} du dv$$

----- (31)

where, we have already defined the 8- node linear convex quadrilaterals  $Q_e$  (e=1,2,3) in (x,y) space and  $\hat{Q}$  in (u,v) space in Fig 3a- 3b. From Eqns.(29)-(31) , we obtain the following relations connecting the submatrices  $S^{i,j,e}$  and  $K^{i,j,e}$

We now obtain the submatrices  $S^{i,j,e}$  and  $K^{i,j,e}$  in an explicit form from Eqs.(29a)- (29b)

$$S^{i,j,e} = \iint_{Q_e} G_{x,y}^{i,j,e} dx dy = \begin{pmatrix} \iint_{Q_e} \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} dx dy & \iint_{Q_e} \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial y} dx dy \\ \iint_{Q_e} \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial x} dx dy & \iint_{Q_e} \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} dx dy \end{pmatrix}$$

$$= \begin{pmatrix} S_{2i-1,2j-1}^e & S_{2i-1,2j}^e \\ S_{2i,2j-1}^e & S_{2i,2j}^e \end{pmatrix} \text{ (say) } \text{-----(32)}$$

and in similar manner

$$K^{i,j,e} = \iint_{\hat{Q}} G_{u,v}^{i,j,e} du dv = \begin{pmatrix} \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial u} \frac{\partial N_j^e}{\partial u} dudv & \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial u} \frac{\partial N_j^e}{\partial v} dudv \\ \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial v} \frac{\partial N_j^e}{\partial u} dudv & \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial v} \frac{\partial N_j^e}{\partial v} dudv \end{pmatrix}$$

$$= \begin{pmatrix} K_{2i-1,2j-1}^e & K_{2i-1,2j}^e \\ K_{2i,2j-1}^e & K_{2i,2j}^e \end{pmatrix} \text{ (say) } \text{-----(33)}$$

We have now from the above Eqns.(29)-(33)

$$S^{i,j,e} = \iint_{Q_e} G_{x,y}^{i,j,e} dx dy = \iint_{\hat{Q}} (P G_{u,v}^{i,j,e} P^T) \frac{\partial(x,y)}{\partial(u,v)} du dv$$

$$= 2\Delta_{abc} \iint_{\hat{Q}} (P G_{u,v}^{i,j,e} P^T) du dv$$

$$= 2\Delta_{abc} P \left( \iint_{\hat{Q}} G_{u,v}^{i,j,e} du dv \right) P^T$$

$$= 2\Delta_{abc} P (K^{i,j,e}) P^T \quad , \quad (i, j = 1, 2, 3, 4, 5, 6, 7, 8) \text{-----(34)}$$

We can thus obtain the global derivative integrals in the physical space or Cartesian space (x,y) by using the matrix triple product established in Eqn.(34).

We note that  $\hat{Q}$  is the 8- node linear convex quadrilateral in (u, v) space spanned by the vertices (1/3, 1/3), (1/6, 5/12), (0, 1/2), (0, 1/4), (0, 0), (1/4, 0), (1/2, 0) and (5/12, 1/6) in uv- plane hence from Eqn.(33)

$$K^{i,j,e} = \iint_{\hat{Q}} G_{u,v}^{i,j,e} du dv \text{-----(35)}$$

$$= \int_{-1}^1 \int_{-1}^1 G_{u,v}^{i,j,e} \frac{\partial(u,v)}{\partial(\xi,\eta)} d\xi d\eta \text{-----(36)}$$

We now refer to section 6.1 of this paper, in this section, we have derived the necessary relations to integrate Eq.(35). As in Eqns.(15)-(16), we use the transformation of Eqn.(17) to map the 8- node quadrilateral  $\hat{Q}$  to the 8- node 2-square  $-1 \leq \xi, \eta \leq 1$  Using Eqn.(17) in Eqn.(36), we obtain

$$K^{i,j,e} = \iint_{\hat{Q}} G_{u,v}^{i,j,e} \left( \frac{4+\xi+\eta}{96} \right) d\xi d\eta \text{----- (37)}$$

Thus, we have from Eq.(34)

$$S^{i,j,e} = (2\Delta_{abc}) P (K^{i,j,e}) P^T \text{----- (38)}$$

Where  $K^{i,j,e}$  is given in Eqn.(37)

In Eqn.(38) ,  $2\Delta_{abc} = 2 * \text{area of the triangle spanning vertices } A(x_a, y_a), B(x_b, y_b), C(x_c, y_c)$  is a scalar.

The matrices  $P$ ,  $P^T$  depend purely on the nodal coordinates  $(x_a, y_a)$ ,  $(x_b, y_b)$ ,  $(x_c, y_c)$  the matrix  $K^{i,j,e}$  can be explicitly computed by the relations obtained in section 2 – 6. We find that  $K^{i,j,e}$  is a  $(2 \times 2)$  matrix of integrals whose integrands are rational functions with polynomial numerator and the linear denominator  $(4 + \xi + \eta)$ . Hence these integrals can be explicitly computed. The explicit values of these integrals are expressible in terms of logarithmic constants. We have used symbolic mathematics software of MATLAB to compute the explicit values and their conversion to any number of digits can be obtained by using variable precision arithmetic (vpa) command. The matrix  $K^e$  as noted in Eqn.(33) is of order  $(2n_{de}) \times (2n_{de})$ ,  $n_{de} = 8$  for 8-node convex quadrilateral element.

We have computed  $K^e$  for the four node element  $n_{de} = 4$  in our recent paper [18]. In the present paper, we have computed  $K^e$  for the 8- node linear convex quadrilateral  $\hat{Q}$  in  $uv$  – space. This is listed in Table 1A and Table 1B.

### 7. Application Example :

In this section, we examine the application of the proposed explicit integration scheme to the Saint Venant Torsion problem [24]. Exact solutions of this problem for simple cross sections such as circle, ellipse, equilateral triangle and rectangle have been rigorously derived. These problems are described by the following boundary value problem ;

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + 2G\theta = 0 \quad \text{in } R \quad \text{----- (39)}$$

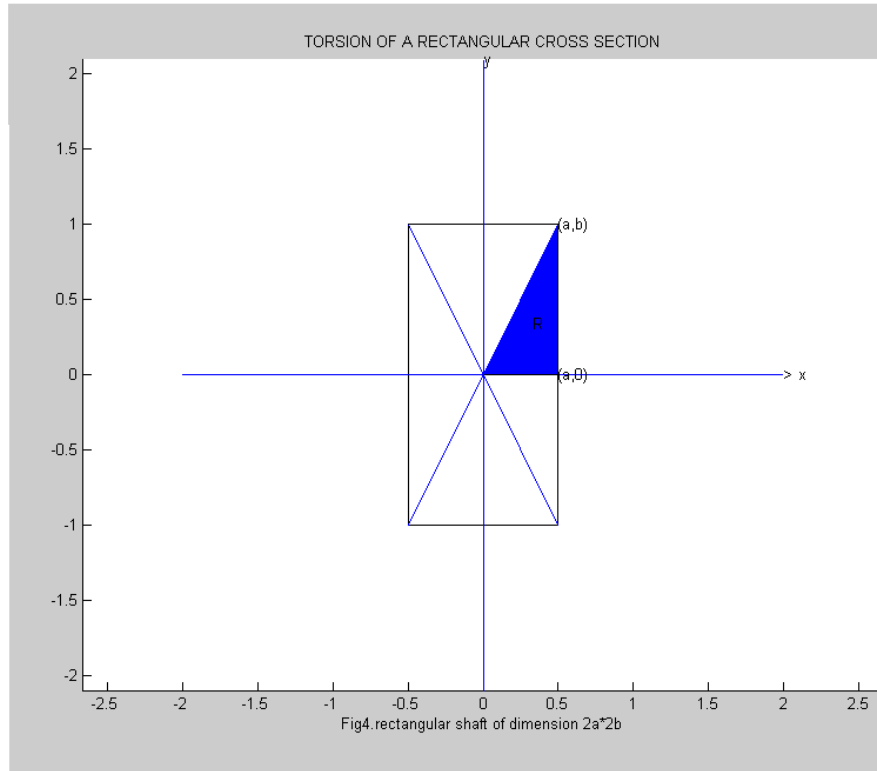
$$\phi = 0 \quad \text{on } \partial R, \text{ the boundary of } R \quad \text{----- (40)}$$

where  $\phi(x,y)$  is known as Prandtl stress function,  $G$  is the shear modulus,  $\theta$  is the angle of twist per unit length,  $R$  is the cross sectional region and  $\partial R$  is the boundary of  $R$ . We choose  $G\theta = 1$  for the sake of simplicity. Then the corresponding torsional constant is given by the equation

$$t_c = 2 \iint_R \phi(x,y) dx dy \quad \text{----- (41)}$$

#### 7.1 Torsion of a rectangular Cross section :

We consider the region  $R$  as the rectangular cross section with the vertices  $(-a, -b)$ ,  $(a, -b)$ ,  $(a, b)$  and  $(-a, b)$  as shown in Fig.4. From the theory of elasticity [24-26], the Prandtl stress function  $\phi$  and the torsional constant  $t_c$  for the rectangular cross section of length  $2b$  and breadth  $2a$  are given by the following expression in series form



$$\Phi = \frac{32a^2}{\pi^3} \sum_{n=1,3,5,\dots} \frac{(-1)^{\frac{(n-1)}{2}}}{n^3} \left[ 1 - \frac{\cosh\left(\frac{n\pi y}{2a}\right)}{\cosh\left(\frac{n\pi b}{2a}\right)} \right] \cos\left(\frac{n\pi x}{2a}\right) \quad \text{----- (42)}$$

$$t_c = \frac{(2a)^3(2b)}{3} \left[ 1 - \frac{192}{\pi^5} \sum_{n=1,3,5,\dots} \frac{1}{n^5} \tan\left(\frac{n\pi b}{2a}\right) \right] \quad \text{----- (43)}$$

These expressions converge rapidly for  $b > a$ . In the present study, we consider the square cross section of unit length

## 7.2 Finite Element procedure :

We consider the rectangular region R of Fig.4 with  $a = b = \frac{1}{2}$ . This cross section has four axes of symmetry, therefore, only one eighth of the cross section needs to be analyzed. We thus have to model the region R, the right isosceles triangular cross section with vertices  $(0, 0)$ ,  $(\frac{1}{2}, 0)$ ,  $(\frac{1}{2}, \frac{1}{2})$  as shown in Fig 4 . We assume that the domain R is discretised by eight node quadrilateral elements  $Q_e$  as explained in section 3 of this paper. We assume that within each eight node quadrilateral element  $Q_e$ , the Prandtl stress function  $\phi^e(x, y)$  is expressed in terms of the natural coordinate variates  $(\xi, \eta)$  such that

$$\phi^e(x, y) = \sum_{i=1}^8 N_i^e(\xi, \eta) \phi_i^e \quad \text{----- (44)}$$

Where  $\phi_i^e$  is the nodal value at the node i and the  $N_i^e(\xi, \eta)$  are as given in Eqn.(5) in section 3 of this paper boundary value problem of Eqn.(39)-(40) in the region R is expressed as

$$[K] \{\phi_N\} = \{F_N\} \quad , \quad (N = 1, 2, \dots, nd) \quad \text{----- (45)}$$

Where  $nd$  = the total number of nodes used in discretisation of region R and



$$[K] = \sum_{e=1}^{n_e} [K_{ij}^e] , \{F_N\} = \sum_{e=1}^{n_e} \{F_i^e\} \quad \text{-----(46)}$$

$$K_{i,j}^e = \iint_{Q_e} \left( \frac{\partial N_i^e}{\partial x} \frac{\partial N_j^e}{\partial x} + \frac{\partial N_i^e}{\partial y} \frac{\partial N_j^e}{\partial y} \right) dx dy \quad \text{-----(47)}$$

$$F_i^e = 2 \iint_{Q_e} N_i^e(\xi, \eta) dx dy \quad \text{-----(48)}$$

$$\{(\phi_N, F_N), N = 1, 2, \dots, n_d\} \quad \text{-----(49)}$$

$n_e$  = the total number of elements used in the discretisation of the region R

### 7.3 Discretisation and Finite Element Model :

In designing a finite element model, we shall discretise the region R, the one octant (a right isosceles triangle) with vertices O(0, 0), A(1/2, 0) and B(1/2, 1/2) into 8- node quadrilateral elements as described in section 2 – 4. This discretisation generates the Jacobian  $(4 + \xi + \eta) * \text{a constant}$  for all such 8- node quadrilaterals of the region R, We shall consider the following boundary value problem

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + 2 = 0 , \text{ within triangle OAB} \quad \text{-----(50a)}$$

$$\phi = 0 , \text{ on side AB} \quad \text{-----(50b)}$$

$$\frac{\partial \phi}{\partial n} = 0 , \text{ on sides OA and OB, the lines of symmetry} \quad \text{-----(50c)}$$

We discretise the region R, the right isosceles triangle OAB into  $1^2, 2^2, 3^2, 4^2, 5^2, 6^2, 7^2, \dots, 25^2$  triangles of equal size, each of these triangles is further discretised into three eight node quadrilaterals as explained in section 2- 5. We have depicted these 25 meshes in FIGURE SECTION. Finite element solutions for these discretisations i.e. for mesh number 1-25 is depicted in Table-4. We have compared these solutions with similar discretisation of four node quadrilaterals which can be obtained from our recent paper[28]. We find rapid converges to the analytical solution as expected and the superiority of eight node quadrilaterals with four node quadrilaterals as expected.

In a recent paper[26] a new approach to automatic generation of all quadrilateral mesh for finite analysis is proposed. We have used this to discretised the 1/8-th of the square cross section into an all quadrilateral mesh. The following MATLAB PROGRAMS are written for this purpose:

```
[eln,nodetel,nodes,nnode]=nodaladdresses_special_convex_quadrilaterals_2nd_order(ndiv);
[coord,gcoord]=coordinate_rtisoscelestriangle00_h0_hh_2ndorder(ndiv);
```

**(1) D2LaplaceEquationQ8Ex3automeshgen.m**

**(2)coordinate\_rtisoscelestriangle00\_h0\_hh\_2ndorder.m**

**(3)nodaladdresses\_special\_convex\_quadrilaterals\_2nd\_order.m**

**(4)quadrilateralmesh\_square\_cross\_section\_q8automeshgen.m**

These are appended for reference

## Conclusions :

This paper presents the explicit integration scheme for a unique linear convex 8- node quadrilateral which can be obtained from an arbitrary linear triangle by joining the centroid to the midpoints of sides of the triangle. The explicit integration scheme proposed for these unique linear convex 8- node quadrilaterals is derived by using the standard transformations in two steps. We first map an arbitrary linear triangle into a standard right isosceles triangle by using a affine linear transformation from global  $(x, y)$  space into a local space  $(u, v)$ . We then discretise this standard right isosceles triangle in  $(u, v)$  space into three unique linear convex 8- node quadrilaterals. We have shown by proving a lemma that any unique linear convex 8-node quadrilateral in  $(x, y)$  space can be mapped into one of the unique 8-node quadrilaterals in  $(u, v)$  space. We have then mapped these linear convex 8- node quadrilaterals into a 2-square in the local  $(\xi, \eta)$  space by use of the bilinear transformation between  $(u, v)$  and  $(\xi, \eta)$  space. Using these two mappings, we have established an integral derivative product relation between the linear convex 8- node quadrilaterals in the global  $(x, y)$  space interior to the arbitrary triangle and the linear convex 8- node quadrilaterals in the local  $(u, v)$  space which are interior to the standard right isosceles triangle. We have then shown that the product of global derivative integrals  $S^{i,j,e}$  in global  $(x, y)$  space can be expressed as a matrix triple product  $P * (K^{i,j,e}) * P^T * (2 * \text{area of the arbitrary triangle in } (x, y) \text{ space})$ , in which  $P$  is a geometric properties matrix and  $K^{i,j,e}$  is the product of global derivative integrals in  $(u, v)$  space, and  $(i, j = 1, 2, 3, 4, 5, 6, 7, 8)$ . We have shown that the explicit integration of the global derivative products in  $(u, v)$  space over the unique 8- node quadrilateral spanning vertices  $\{(1/3, 1/3), (1/6, 5/12), (0, 1/2), (0, 1/4), (0, 0), (1/4, 0), (1/2, 0) \text{ and } (5/12, 1/6)\}$  is now possible by application of symbolic processing capabilities in MATLAB which are based on MAPLE –V mathematical software package. The proposed explicit integration scheme is useful technique for boundary value problems governed by either a single or a system of partial differential equations. The physical applications of such problems are numerous in science, and engineering and business, the well known examples are the Laplace and Poisson equations with suitable boundary conditions and the examples of system of equations are the plane stress, plane stress and axisymmetric stress analysis, flow through porous media, shallow water circulation, dispersion and viscous incompressible flow etc in the areas of solid and fluid mechanics. We have demonstrated the proposed explicit integration scheme to solve the St. Venant Torsion problem for a square cross section. Monotonic convergence from below is observed with known analytical solutions for the Prandtl stress function and the torsional constant which are expressed in terms of infinite series as noted in Eqns.(42)-(43). We hope that the scheme developed in this paper will be useful for the solution of boundary value problems governed by second order partial differential equations.

## REFERENCES:

- [1] Zienkiewicz O.C, Taylor R.L and Zhu J.Z Finite Element Method, its basis and fundamentals, Elsevier, (2005)
- [2] Bathe K.J Finite Element Procedures, Prentice Hall, Englewood Cliffs, N J (1996)
- [3] Reddy J.N Finite Element Method, Third Edition, Tata Mc Graw-Hill (2005)
- [4] Burden R.L and Faires J.D Numerical Analysis, 9<sup>th</sup> Edition, Brooks/Cole, Cengage Learning (2011)
- [5] Stroud A.H and Secrest D, Gaussian quadrature formulas, Prentice Hall, Englewood Cliffs, N J, (1966)
- [6] Stoer J and Bulirsch R, Introduction to Numerical Analysis, Springer-Verlag, New York (1980)
- [7] Chung T.J Finite Element Analysis in Fluid Dynamics, pp. 191-199, Mc Graw Hill, Scarborough, C A, (1978)
- [8] Rathod H.T, Some analytical integration formulae for four node isoparametric element, Computer and structures 30(5), pp.1101-1109, (1988)
- [9] Babu D.K and Pinder G.F, Analytical integration formulae for linear

- isoparametric finite elements, *Int. J. Numer. Methods Eng* 20, pp.1153-1166
- [10] Mizukami A, Some integration formulas for four node isoparametric element, *Computer Methods in Applied Mechanics and Engineering*. 59 pp. 111-121(1986)
- [11] Okabe M, Analytical integration formulas related to convex quadrilateral finite elements, *Computer methods in Applied mechanics and Engineering*. 29, pp.201-218 (1981)
- [12] Griffiths D.V Stiffness matrix of the four node quadrilateral element in closed form, *International Journal for Numerical Methods in Engineering*. 28, pp.687-703(1996)
- [13] Rathod H.T and Shafiqul Islam. Md , Integration of rational functions of bivariate polynomial numerators with linear denominators over a (-1,1) square in a local parametric two dimensional space, *Computer Methods in Applied Mechanics and Engineering*. 161 pp.195-213 (1998)
- [14] Rathod H.T and Sajedul Karim, Md An explicit integration scheme based on recursion and matrix multiplication for the linear convex quadrilateral elements, *International Journal of Computational Engineering Science*. 2(1) pp. 95-135(2001)
- [15] Yagawa G, Ye G.W and Yoshimura S, A numerical integration scheme for finite element method based on symbolic manipulation, *International Journal for Numerical Methods in Engineering*. 29, pp.1539-1549(1990)
- [16] Rathod H.T and Shafiqul Islam Md, Some pre-computed numeric arrays for linear convex quadrilateral finite elements, *Finite Elements in Analysis and Design* 38, pp. 113-136 (2001)
- [17] Hanselman D and Littlefield B, *Mastering MATLAB 7* , Prentice Hall, Happer Saddle River, N J . (2005)
- [18] Hunt B.H, Lipsman R.L and Rosenberg J.M, *A Guide to MATLAB for beginners and experienced users*, Cambridge University Press (2005)
- [19] Char B, Geddes K, Gonnet G, Leong B, Monagan M and Watt S, *First Leaves; A tutorial Introduction to Maple V* , New York : Springer-Verlag (1992)
- [20] Eugene D, *Mathematica , Schaums Outlines Theory and Problems*, Tata Mc Graw Hill (2001)
- [21] Ruskeepaa H, *Mathematica Navigator*, Academic Press (2009)
- [22] Timoshenko S.P and Goodier J.N, *Theory of Elasticity*, 3<sup>rd</sup> Edition, Tata Mc Graw Hill Edition (2010)
- [23] Budynas R.G, *Applied Strength and Applied Stress Analysis*, Second Edition, Tata Mc Graw Hill Edition (2011)
- [24] Roark R.J, *Formulas for stress and strain*, Mc Graw Hill, New York (1965)
- [25] Nguyen S.H , An accurate finite element formulation for linear elastic torsion calculations, *Computers and Structures*. 42, pp.707-711 (1992)
- [26] Rathod H.T, Rathod .Bharath, Shivaram.K.T, Sugantha Devi.K, A new approach to automatic generation of all quadrilateral mesh for finite analysis, *International Journal of Engineering and Computer Science*, Vol. 2, issue 12, pp3488-3530(2013)
- [27] Rathod H.T, Venkatesh.B, Shivaram. K.T, Mamatha.T.M, Numerical Integration over polygonal domains using convex quadrangulation and Gauss Legendre Quadrature Rules, *International Journal of Engineering and Computer Science*, Vol. 2, issue 8, pp2576-2610(2013)
- [28] H.T. Rathod, Bharath Rathod, Shivaram K.T , H. Y. Shrivalli , Tara Rathod , K. Sugantha Devi, An explicit finite element integration scheme using automatic mesh generation technique for linear convex quadrilaterals over plane regions ,to appear in *international Journal of Engineering and Computer Science*, Vol. 3, issue 4, pp (2014)

## TABLES SECTION

TABLE-1

EIGHT NODE SERENDIPITY ELEMENT

$$(K_{i,j}^e, i=1(1)16, j=1(1)16)$$

ANALYTICAL VALUES FOR PRODUCTS OF GLOBAL DERIVATIVE INTEGRALS WITH 32-DIGITS PRECISION

OVER THE EIGHT NODE QUADRILATERAL  $((u_k, v_k), k=1,2,3,4)=((1/3,1/3), (0,1/2), (0,0), (1/2,0))$ , WITH

$(u_5, v_5), (u_6, v_6), (u_7, v_7)$  AND  $(u_8, v_8)$  AS THE MIDPOINT OF SIDES 1-2, 2-3, 3-4, AND 4-1 RESPECTIVELY IN

THE INTERIOR OF THE STANDARD TRIANGLE IN  $(u, v)$  SPACE (see eqn (i))

$$K^{p,q,e} = [K_{2p-1,2q-1}^e \quad K_{2p-1,2q}^e]$$

$$[K_{2p,2q-1}^e \quad K_{2p,2q}^e]$$

where,  $(p, q=1,2,3,4,5,6,7,8)$

$p$	$k^{p,1,e}$
1	$\frac{-58259/630-93266/105 \cdot \log(2)+22599/35 \cdot \log(3)}{-545/126-13124/21 \cdot \log(2)+5589/14 \cdot \log(3)}$
2	$\frac{9223/135+1868/15 \cdot \log(2)-702/5 \cdot \log(3)}{-262/135-1112/15 \cdot \log(2)+243/5 \cdot \log(3)}$
3	$\frac{-359/189+4306/21 \cdot \log(2)-891/7 \cdot \log(3)}{-5954/945+18916/105 \cdot \log(2)-7533/70 \cdot \log(3)}$
4	$\frac{-26893/270-5284/15 \cdot \log(2)+1566/5 \cdot \log(3)}{-479/270-1112/15 \cdot \log(2)+243/5 \cdot \log(3)}$
5	$\frac{-18082/315+37784/105 \cdot \log(2)-6156/35 \cdot \log(3)}{1381/630+53944/105 \cdot \log(2)-11421/35 \cdot \log(3)}$
6	$\frac{-45224/945-40024/105 \cdot \log(2)+9936/35 \cdot \log(3)}{10039/1890-22808/105 \cdot \log(2)+4617/35 \cdot \log(3)}$
7	$\frac{33673/945-14992/105 \cdot \log(2)+1998/35 \cdot \log(3)}{10039/1890-22808/105 \cdot \log(2)+4617/35 \cdot \log(3)}$
8	$\frac{1367/7+22576/21 \cdot \log(2)-5994/7 \cdot \log(3)}{961/630+53944/105 \cdot \log(2)-11421/35 \cdot \log(3)}$

p	$\kappa^{p,2,e}$	
1	$9223/135+1868/15\log(2)-702/5\log(3)$ $-479/270-1112/15\log(2)+243/5\log(3)$	$-262/135-1112/15\log(2)+243/5\log(3)$ $-26893/270-5284/15\log(2)+1566/5\log(3)$
2	$-1892/45-5288/45\log(2)+564/5\log(3)$ $-703/45-2512/45\log(2)+246/5\log(3)$	$-703/45-2512/45\log(2)+246/5\log(3)$ $-4352/45-12008/45\log(2)+1284/5\log(3)$
3	$-268/135+44/5\log(2)-18/5\log(3)$ $-151/270+104/5\log(2)-63/5\log(3)$	$-53/135+104/5\log(2)-63/5\log(3)$ $523/270+188/5\log(2)-126/5\log(3)$
4	$323/5+7288/45\log(2)-804/5\log(3)$ $72/5+1712/45\log(2)-186/5\log(3)$	$72/5+1712/45\log(2)-186/5\log(3)$ $323/5+7288/45\log(2)-804/5\log(3)$
5	$3746/135+592/5\log(2)-504/5\log(3)$ $3178/135+656/5\log(2)-522/5\log(3)$	$3268/135+656/5\log(2)-522/5\log(3)$ $26528/135+2976/5\log(2)-2772/5\log(3)$
6	$4666/135+3248/45\log(2)-384/5\log(3)$ $220/27-16/9\log(2)-6\log(3)$	$202/27-16/9\log(2)-6\log(3)$ $4024/135+1952/45\log(2)-276/5\log(3)$
7	$-526/27-608/9\log(2)+60\log(3)$ $-964/135-2192/45\log(2)+186/5\log(3)$	$-964/135-2192/45\log(2)+186/5\log(3)$ $-6782/135-7696/45\log(2)+768/5\log(3)$
8	$-17782/135-1504/5\log(2)+1548/5\log(3)$ $-2834/135-48/5\log(2)+126/5\log(3)$	$-2834/135-48/5\log(2)+126/5\log(3)$ $-1250/27-48\log(2)+72\log(3)$

p	$\kappa^{p,3,e}$	
1	$-359/189+4306/21\log(2)-891/7\log(3)$ $-5954/945+18916/105\log(2)-7533/70\log(3)$	$-5954/945+18916/105\log(2)-7533/70\log(3)$ $-359/189+4306/21\log(2)-891/7\log(3)$
2	$-268/135+44/5\log(2)-18/5\log(3)$ $53/135+104/5\log(2)-63/5\log(3)$	$-151/270+104/5\log(2)-63/5\log(3)$ $523/270+188/5\log(2)-126/5\log(3)$
3	$2617/1890-6914/105\log(2)+1431/35\log(3)$ $7211/1890-5812/105\log(2)+2241/70\log(3)$	$7211/1890-5812/105\log(2)+2241/70\log(3)$ $2617/1890-6914/105\log(2)+1431/35\log(3)$
4	$523/270+188/5\log(2)-126/5\log(3)$ $-151/270+104/5\log(2)-63/5\log(3)$	$-53/135+104/5\log(2)-63/5\log(3)$ $-268/135+44/5\log(2)-18/5\log(3)$
5	$4042/945-15608/105\log(2)+3132/35\log(3)$ $2117/378-3128/21\log(2)+621/7\log(3)$	$2117/378-3128/21\log(2)+621/7\log(3)$ $-1019/945-20144/105\log(2)+4266/35\log(3)$
6	$88/945+8888/105\log(2)-1872/35\log(3)$ $-7967/1890+6904/105\log(2)-1341/35\log(3)$	$-6707/1890+6904/105\log(2)-1341/35\log(3)$ $-2579/945+7376/105\log(2)-1494/35\log(3)$
7	$-2579/945+7376/105\log(2)-1494/35\log(3)$ $-6707/1890+6904/105\log(2)-1341/35\log(3)$	$-7967/1890+6904/105\log(2)-1341/35\log(3)$ $88/945+8888/105\log(2)-1872/35\log(3)$
8	$-1019/945-20144/105\log(2)+4266/35\log(3)$ $2117/378-3128/21\log(2)+621/7\log(3)$	$2117/378-3128/21\log(2)+621/7\log(3)$ $4042/945-15608/105\log(2)+3132/35\log(3)$

p	$\kappa^{p,4,e}$	
1	$-26893/270-5284/15 \cdot \log(2)+1566/5 \cdot \log(3)$ $-262/135-1112/15 \cdot \log(2)+243/5 \cdot \log(3)$	$-479/270-1112/15 \cdot \log(2)+243/5 \cdot \log(3)$ $9223/135+1868/15 \cdot \log(2)-702/5 \cdot \log(3)$
2	$323/5+7288/45 \cdot \log(2)-804/5 \cdot \log(3)$ $72/5+1712/45 \cdot \log(2)-186/5 \cdot \log(3)$	$72/5+1712/45 \cdot \log(2)-186/5 \cdot \log(3)$ $323/5+7288/45 \cdot \log(2)-804/5 \cdot \log(3)$
3	$523/270+188/5 \cdot \log(2)-126/5 \cdot \log(3)$ $-53/135+104/5 \cdot \log(2)-63/5 \cdot \log(3)$	$-151/270+104/5 \cdot \log(2)-63/5 \cdot \log(3)$ $-268/135+44/5 \cdot \log(2)-18/5 \cdot \log(3)$
4	$-4352/45-12008/45 \cdot \log(2)+1284/5 \cdot \log(3)$ $-703/45-2512/45 \cdot \log(2)+246/5 \cdot \log(3)$	$-703/45-2512/45 \cdot \log(2)+246/5 \cdot \log(3)$ $-1892/45-5288/45 \cdot \log(2)+564/5 \cdot \log(3)$
5	$-.467276179069773367865781067869$ $.38224148468296660162159011855e-1$	$.38224148468296660162159011855e-1$ $-.8682585930330713525401530193e-1$
6	$-6782/135-7696/45 \cdot \log(2)+768/5 \cdot \log(3)$ $-964/135-2192/45 \cdot \log(2)+186/5 \cdot \log(3)$	$-964/135-2192/45 \cdot \log(2)+186/5 \cdot \log(3)$ $-526/27-608/9 \cdot \log(2)+60 \cdot \log(3)$
7	$4024/135+1952/45 \cdot \log(2)-276/5 \cdot \log(3)$ $202/27-16/9 \cdot \log(2)-6 \cdot \log(3)$	$220/27-16/9 \cdot \log(2)-6 \cdot \log(3)$ $4666/135+3248/45 \cdot \log(2)-384/5 \cdot \log(3)$
8	$26528/135+2976/5 \cdot \log(2)-2772/5 \cdot \log(3)$ $3268/135+656/5 \cdot \log(2)-522/5 \cdot \log(3)$	$3178/135+656/5 \cdot \log(2)-522/5 \cdot \log(3)$ $3746/135+592/5 \cdot \log(2)-504/5 \cdot \log(3)$

p	$\kappa^{p,5,e}$	
1	$-18082/315+37784/105 \cdot \log(2)-6156/35 \cdot \log(3)$ $961/630+53944/105 \cdot \log(2)-11421/35 \cdot \log(3)$	$1381/630+53944/105 \cdot \log(2)-11421/35 \cdot \log(3)$ $1367/7+22576/21 \cdot \log(2)-5994/7 \cdot \log(3)$
2	$3746/135+592/5 \cdot \log(2)-504/5 \cdot \log(3)$ $3268/135+656/5 \cdot \log(2)-522/5 \cdot \log(3)$	$3178/135+656/5 \cdot \log(2)-522/5 \cdot \log(3)$ $26528/135+2976/5 \cdot \log(2)-2772/5 \cdot \log(3)$
3	$4042/945-15608/105 \cdot \log(2)+3132/35 \cdot \log(3)$ $2117/378-3128/21 \cdot \log(2)+621/7 \cdot \log(3)$	$2117/378-3128/21 \cdot \log(2)+621/7 \cdot \log(3)$ $-1019/945-20144/105 \cdot \log(2)+4266/35 \cdot \log(3)$
4	$-1250/27-48 \cdot \log(2)+72 \cdot \log(3)$ $-2834/135-48/5 \cdot \log(2)+126/5 \cdot \log(3)$	$-2834/135-48/5 \cdot \log(2)+126/5 \cdot \log(3)$ $-17782/135-1504/5 \cdot \log(2)+1548/5 \cdot \log(3)$
5	$-3268/315-45344/105 \cdot \log(2)+9936/35 \cdot \log(3)$ $-10522/315-55456/105 \cdot \log(2)+12744/35 \cdot \log(3)$	$-10522/315-55456/105 \cdot \log(2)+12744/35 \cdot \log(3)$ $-41204/105-160256/105 \cdot \log(2)+46224/35 \cdot \log(3)$
6	$-26512/945+13088/105 \cdot \log(2)-1872/35 \cdot \log(3)$ $-15824/945+15136/105 \cdot \log(2)-2664/35 \cdot \log(3)$	$-15824/945+15136/105 \cdot \log(2)-2664/35 \cdot \log(3)$ $-61288/945+4352/105 \cdot \log(2)+1152/35 \cdot \log(3)$
7	$9944/945+21824/105 \cdot \log(2)-4896/35 \cdot \log(3)$ $5848/945+22528/105 \cdot \log(2)-4932/35 \cdot \log(3)$	$5848/945+22528/105 \cdot \log(2)-4932/35 \cdot \log(3)$ $93272/945+51392/105 \cdot \log(2)-13968/35 \cdot \log(3)$
8	$10456/105-19136/105 \cdot \log(2)+864/35 \cdot \log(3)$ $706/21-6656/21 \cdot \log(2)+1188/7 \cdot \log(3)$	$706/21-6656/21 \cdot \log(2)+1188/7 \cdot \log(3)$ $10456/105-19136/105 \cdot \log(2)+864/35 \cdot \log(3)$

p	$\kappa^{p,6,e}$
1	$\frac{-45224/945-40024/105 \cdot \log(2)+9936/35 \cdot \log(3)}{10039/1890-22808/105 \cdot \log(2)+4617/35 \cdot \log(3)}$ $\frac{10039/1890-22808/105 \cdot \log(2)+4617/35 \cdot \log(3)}{33673/945-14992/105 \cdot \log(2)+1998/35 \cdot \log(3)}$
2	$\frac{4666/135+3248/45 \cdot \log(2)-384/5 \cdot \log(3)}{202/27-16/9 \cdot \log(2)-6 \cdot \log(3)}$ $\frac{220/27-16/9 \cdot \log(2)-6 \cdot \log(3)}{4024/135+1952/45 \cdot \log(2)-276/5 \cdot \log(3)}$
3	$\frac{88/945+8888/105 \cdot \log(2)-1872/35 \cdot \log(3)}{6707/1890+6904/105 \cdot \log(2)-1341/35 \cdot \log(3)}$ $\frac{-7967/1890+6904/105 \cdot \log(2)-1341/35 \cdot \log(3)}{-2579/945+7376/105 \cdot \log(2)-1494/35 \cdot \log(3)}$
4	$\frac{-6782/135-7696/45 \cdot \log(2)+768/5 \cdot \log(3)}{-964/135-2192/45 \cdot \log(2)+186/5 \cdot \log(3)}$ $\frac{-964/135-2192/45 \cdot \log(2)+186/5 \cdot \log(3)}{-526/27-608/9 \cdot \log(2)+60 \cdot \log(3)}$
5	$\frac{-26512/945+13088/105 \cdot \log(2)-1872/35 \cdot \log(3)}{-15824/945+15136/105 \cdot \log(2)-2664/35 \cdot \log(3)}$ $\frac{-15824/945+15136/105 \cdot \log(2)-2664/35 \cdot \log(3)}{-61288/945+4352/105 \cdot \log(2)+1152/35 \cdot \log(3)}$
6	$\frac{-4604/189-10720/63 \cdot \log(2)+912/7 \cdot \log(3)}{382/945-28384/315 \cdot \log(2)+1992/35 \cdot \log(3)}$ $\frac{382/945-28384/315 \cdot \log(2)+1992/35 \cdot \log(3)}{-5716/945-32768/315 \cdot \log(2)+2544/35 \cdot \log(3)}$
7	$\frac{16208/945-15296/315 \cdot \log(2)+528/35 \cdot \log(3)}{7606/945-20992/315 \cdot \log(2)+1236/35 \cdot \log(3)}$ $\frac{7606/945-20992/315 \cdot \log(2)+1236/35 \cdot \log(3)}{16208/945-15296/315 \cdot \log(2)+528/35 \cdot \log(3)}$
8	$\frac{93272/945+51392/105 \cdot \log(2)-13968/35 \cdot \log(3)}{5848/945+22528/105 \cdot \log(2)-4932/35 \cdot \log(3)}$ $\frac{5848/945+22528/105 \cdot \log(2)-4932/35 \cdot \log(3)}{9944/945+21824/105 \cdot \log(2)-4896/35 \cdot \log(3)}$

p	$\kappa^{p,7,e}$
1	$\frac{33673/945-14992/105 \cdot \log(2)+1998/35 \cdot \log(3)}{10039/1890-22808/105 \cdot \log(2)+4617/35 \cdot \log(3)}$ $\frac{10039/1890-22808/105 \cdot \log(2)+4617/35 \cdot \log(3)}{-45224/945-40024/105 \cdot \log(2)+9936/35 \cdot \log(3)}$
2	$\frac{-526/27-608/9 \cdot \log(2)+60 \cdot \log(3)}{-964/135-2192/45 \cdot \log(2)+186/5 \cdot \log(3)}$ $\frac{-964/135-2192/45 \cdot \log(2)+186/5 \cdot \log(3)}{-6782/135-7696/45 \cdot \log(2)+768/5 \cdot \log(3)}$
3	$\frac{-2579/945+7376/105 \cdot \log(2)-1494/35 \cdot \log(3)}{-7967/1890+6904/105 \cdot \log(2)-1341/35 \cdot \log(3)}$ $\frac{-7967/1890+6904/105 \cdot \log(2)-1341/35 \cdot \log(3)}{88/945+8888/105 \cdot \log(2)-1872/35 \cdot \log(3)}$
4	$\frac{4024/135+1952/45 \cdot \log(2)-276/5 \cdot \log(3)}{220/27-16/9 \cdot \log(2)-6 \cdot \log(3)}$ $\frac{202/27-16/9 \cdot \log(2)-6 \cdot \log(3)}{4666/135+3248/45 \cdot \log(2)-384/5 \cdot \log(3)}$
5	$\frac{9944/945+21824/105 \cdot \log(2)-4896/35 \cdot \log(3)}{5848/945+22528/105 \cdot \log(2)-4932/35 \cdot \log(3)}$ $\frac{5848/945+22528/105 \cdot \log(2)-4932/35 \cdot \log(3)}{93272/945+51392/105 \cdot \log(2)-13968/35 \cdot \log(3)}$
6	$\frac{16208/945-15296/315 \cdot \log(2)+528/35 \cdot \log(3)}{7606/945-20992/315 \cdot \log(2)+1236/35 \cdot \log(3)}$ $\frac{7606/945-20992/315 \cdot \log(2)+1236/35 \cdot \log(3)}{16208/945-15296/315 \cdot \log(2)+528/35 \cdot \log(3)}$
7	$\frac{-5716/945-32768/315 \cdot \log(2)+2544/35 \cdot \log(3)}{382/945-28384/315 \cdot \log(2)+1992/35 \cdot \log(3)}$ $\frac{382/945-28384/315 \cdot \log(2)+1992/35 \cdot \log(3)}{-4604/189-10720/63 \cdot \log(2)+912/7 \cdot \log(3)}$
8	$\frac{-61288/945+4352/105 \cdot \log(2)+1152/35 \cdot \log(3)}{-15824/945+15136/105 \cdot \log(2)-2664/35 \cdot \log(3)}$ $\frac{-15824/945+15136/105 \cdot \log(2)-2664/35 \cdot \log(3)}{-26512/945+13088/105 \cdot \log(2)-1872/35 \cdot \log(3)}$



p	$K^{p,8,e}$	
1	$1367/7+22576/21*\log(2)-5994/7*\log(3)$ $1381/630+53944/105*\log(2)-11421/35*\log(3)$	$961/630+53944/105*\log(2)-11421/35*\log(3)$ $-18082/315+37784/105*\log(2)-6156/35*\log(3)$
2	$-17782/135-1504/5*\log(2)+1548/5*\log(3)$ $-2834/135-48/5*\log(2)+126/5*\log(3)$	$-2834/135-48/5*\log(2)+126/5*\log(3)$ $-1250/27-48*\log(2)+72*\log(3)$
3	$-1019/945-20144/105*\log(2)+4266/35*\log(3)$ $2117/378-3128/21*\log(2)+621/7*\log(3)$	$2117/378-3128/21*\log(2)+621/7*\log(3)$ $4042/945-15608/105*\log(2)+3132/35*\log(3)$
4	$26528/135+2976/5*\log(2)-2772/5*\log(3)$ $3178/135+656/5*\log(2)-522/5*\log(3)$	$3268/135+656/5*\log(2)-522/5*\log(3)$ $3746/135+592/5*\log(2)-504/5*\log(3)$
5	$10456/105-19136/105*\log(2)+864/35*\log(3)$ $706/21-6656/21*\log(2)+1188/7*\log(3)$	$706/21-6656/21*\log(2)+1188/7*\log(3)$ $10456/105-19136/105*\log(2)+864/35*\log(3)$
6	$93272/945+51392/105*\log(2)-13968/35*\log(3)$ $5848/945+22528/105*\log(2)-4932/35*\log(3)$	$5848/945+22528/105*\log(2)-4932/35*\log(3)$ $9944/945+21824/105*\log(2)-4896/35*\log(3)$
7	$-61288/945+4352/105*\log(2)+1152/35*\log(3)$ $-15824/945+15136/105*\log(2)-2664/35*\log(3)$	$-15824/945+15136/105*\log(2)-2664/35*\log(3)$ $-26512/945+13088/105*\log(2)-1872/35*\log(3)$
8	$-41204/105-160256/105*\log(2)+46224/35*\log(3)$ $-10522/315-55456/105*\log(2)+12744/35*\log(3)$	$-10522/315-55456/105*\log(2)+12744/35*\log(3)$ $-3268/315-45344/105*\log(2)+9936/35*\log(3)$

TABLE-2

EIGHT NODE SERENDIPITY ELEMENT

$$(K_{i,j}^e, (i=1(1)16, j=1(1)16))$$

NUMERICAL VALUES FOR PRODUCTS OF GLOBAL DERIVATIVE INTEGRALS WITH 32-DIGITS PRECISION

OVER THE EIGHT NODE QUADRILATERAL  $((u_k, v_k), k=1,2,3,4)=((1/3,1/3),(0,1/2),(0,0),(1/2,0))$ , WITH

$(u_5, v_5), (u_6, v_6), (u_7, v_7)$  AND  $(u_8, v_8)$  AS THE MIDPOINT OF SIDES 1-2, 2-3, 3-4, AND 4-1 RESPECTIVELY IN

THE INTERIOR OF THE STANDARD TRIANGLE IN  $(u, v)$  SPACE (see eqn (1))

$$K^{p,q,e} = [K_{2p-1,2q-1}^e \quad K_{2p-1,2q}^e]$$

$$[K_{2p,2q-1}^e \quad K_{2p,2q}^e]$$

where,  $(p, q=1,2,3,4,5,6,7,8)$

p	$\kappa^{p,1,e}$	
1	1.19732437518704939126225670841 1.07234243081152493747384516139	1.07234243081152493747384516139 1.19732437518704939126225670841
2	.39328207524777371271872744686 .66505503018777989604036502927 e-1	.233172169685444656270703169594 .30901830189818397724346918143
3	.39105823773230516010581465792 .34520910581966151866889237468	.34520910581966151866889237468 .39105823773230516010581465792
4	.30901830189818397724346918143 .233172169685444656270703169594	.66505503018777989604036502927 e-1 .39328207524777371271872744686
5	-1.20601914456199564761927515849 -.19480639850558017529001262923	-.86147306517224684195667929589 -.27406792898205548885083220480
6	-.19025279048691687518778377091 -.33047487282879104238539264173	-.33047487282879104238539264173 -.620343126034344229672376860410
7	-.620343126034344229672376860410 -.33047487282879104238539264173	-.33047487282879104238539264173 -.19025279048691687518778377091
8	-.27406792898205548885083220480 -.86147306517224684195667929589	-.19480639850558017529001262923 -1.20601914456199564761927515849

p	$\kappa^{p,2,e}$	
1	.39328207524777371271872744686 .233172169685444656270703169594	.66505503018777989604036502927 e-1 .30901830189818397724346918143
2	.42652636618519994053203031884 -.263513565675283566822536323478	-.263513565675283566822536323478 .45005030410782929514068096460
3	.1595057645371386486635746307258 .15687259169421065039078881847 e-1	.182353925836087731705745548513 .314341351654616447964784833418
4	.20218069152088195770651481813 -.97977735817538970518647437153 e-1	-.97977735817538970518647437153 e-1 .20218069152088195770651481813
5	-.92334437129978410949228855299 -.21347210674508644538200765866	.45319455992158022128465900801 -.574726461686104068369695422 e-2
6	.219362359001080107967460556119 .3242127618106983385904862884652	-.3424539048559683280761803782014 -.768806561449731024667082979907
7	-.390687025888983122842003915751 -.36332930895951737339235932476 e-1	-.36332930895951737339235932476 e-1 -.3376064404514624483888879556 e-1
8	-.8682585930330713525401530193 e-1 .38224148468296660162159011855 e-1	.38224148468296660162159011855 e-1 -.467276179069773367865781067869

p	$\kappa^{p,3,e}$
1	.39105823773230516010581465792 .34520910581966151866889237468
2	.1595057645371386486635746307258 .182353925836087731705745548513
3	.660112740472589759408893364145 .619284819567235241935951286773
4	.314341351654616447964784833418 .15687259169421065039078881847 e-1
5	-.447323988028158265936415185565 -.182789328176641032221385163017
6	.631719461397723832758311823 e-2 -.731811560352895079786782216220
7	-.932211710055872286928957900634 -.65144893686228413120115549553 e-1
8	-.15179959092659718711045271183 -.182789328176641032221385163017

p	$\kappa^{p,4,e}$
1	.30901830189818397724346918143 .66505503018777989604036502927 e-1
2	.20218069152088195770651481813 -.9797735817538970518647437153 e-1
3	.314341351654616447964784833418 .182353925836087731705745548513
4	.45005030410782929514068096460 -.263513565675283566822536323478
5	-.467276179069773367865781067869 .38224148468296660162159011855 e-1
6	-.337606440451462448388879556 e-1 -.36332930895951737339235932476 e-1
7	-.768806561449731024667082979907 -.3424539048559683280761803782014
8	-.574726461686104068369695422 e-2 .45319455992158022128465900801

P	$K^{p,5,e}$
1	-1.20601914456199564761927515849 -.19480639850558017529001262923 -.86147306517224684195667929589 -.27406792898205548885083220480
2	-.92334437129978410949228855299 -.21347210674508644538200765866 .45319455992158022128465900801 -.574726461686104068369695422e-2
3	-.447323988028158265936415185565 -.182789328176641032221385163017 -.182789328176641032221385163017 -.15179959092659718711045271183
4	-.467276179069773367865781067869 .38224148468296660162159011855e-1 .38224148468296660162159011855e-1 -.8682585930330713525401530193e-1
5	2.17177154262400226248727022336 .52992039898580294331102892478 .52992039898580294331102892478 .5882735844715392304142536963
6	-.415943731136358047626104977997 -.446246745737358785845683479200 -.446246745737358785845683479200 .34312587200869545409370986785e-1
7	.91157772830419213711619683500 .9457179332261632804889985221e-1 .9457179332261632804889985221e-1 -.48070367101146296286102539491
8	.376558143167875038936397884560 .37459823838795050721700114125 .37459823838795050721700114125 .376558143167875038936397884560

P	$K^{p,6,e}$
1	-.19025279048691687518778377091 -.33047487282879104238539264173 -.33047487282879104238539264173 -.620343126034344229672376860410
2	.219362359001080107967460556119 .3242127618106983385904862884652 -.3424539048559683280761803782014 -.768806561449731024667082979907
3	.6317194613977723832758311823e-2 -.731811560352895079786782216220 -.65144893686228413120115549553e-1 -.932211710055872286928957900634
4	-.337606440451462448388879556e-1 -.36332930895951737339235932476e-1 -.36332930895951737339235932476e-1 -.390687025888983122842003915751
5	-.415943731136358047626104977997 -.446246745737358785845683479200 -.446246745737358785845683479200 .34312587200869545409370986785e-1
6	.82865433521530100559964914224 .472904354164178470376423509241 .472904354164178470376423509241 1.699831160074343688470918905720
7	.66326947849525293113934929200e-1 .653177200517503508341284619711 .653177200517503508341284619711 .66326947849525293113934929200e-1
8	-.48070367101146296286102539491 .9457179332261632804889985221e-1 .9457179332261632804889985221e-1 .91157772830419213711619683500

p	$\kappa^{p,7,e}$
1	-.620343126034344229672376860410 -.33047487282879104238539264173
2	-.390687025888983122842003915751 -.36332930895951737339235932476e-1
3	-.932211710055872286928957900634 -.731811560352895079786782216220
4	-.768806561449731024667082979907 .3242127618106983385904862884652
5	.91157772830419213711619683500 .9457179332261632804889985221e-1
6	.66326947849525293113934929200e-1 .653177200517503508341284619711
7	1.699831160074343688470918905720 .472904354164178470376423509241
8	.34312587200869545409370986785e-1 -.446246745737358785845683479200

p	$\kappa^{p,8,e}$
1	-.27406792898205548885083220480 -.19480639850558017529001262923
2	-.8682585930330713525401530193e-1 .38224148468296660162159011855e-1
3	-.15179959092659718711045271183 -.182789328176641032221385163017
4	-.574726461686104068369695422e-2 -.21347210674508644538200765866
5	.376558143167875038936397884560 .37459823838795050721700114125
6	-.48070367101146296286102539491 .9457179332261632804889985221e-1
7	.34312587200869545409370986785e-1 -.446246745737358785845683479200
8	.5882735844715392304142536963 .52992039898580294331102892478

Table-3

FOUR NODE-LINEAR CONVEX QUADRILATERALS {COMPUTED FROM REFERENCE[28]}  
 (exact solution of torisonal constant= 0.140577014955156)  
 nnode=number of nodes in theregion R ;  
 nel=number of elements in theregion R;  
 nnel=number of nodes per element

FOR FOUR NODE QUADRILATERAL MESHES					
nnode	nel	nnel	fem solution for torisonal constant	maximum absolute error of Prandtl Stress function values at element nodes	
91	75	4	0.14016582079079	0.00118003991065224	
331	300	4	0.140475648374825	0.000398482465845923	
721	675	4	0.140532182472916	0.000204216075074687	
1261	1200	4	0.140551856423775	0.000128187662553836	
1951	1875	4	0.140560935627972	8.81447150146229e-005	
2791	2700	4	0.140565858672553	6.3733272785143e-005	
3781	3675	4	0.140568823558516	4.76848031509716e-005	
4921	4800	4	0.14057074624724	3.6616338489467e-005	
6211	6075	4	0.140572063602523	2.87090030208158e-005	
7651	7500	4	0.140573005439404	2.28998710912698e-005	

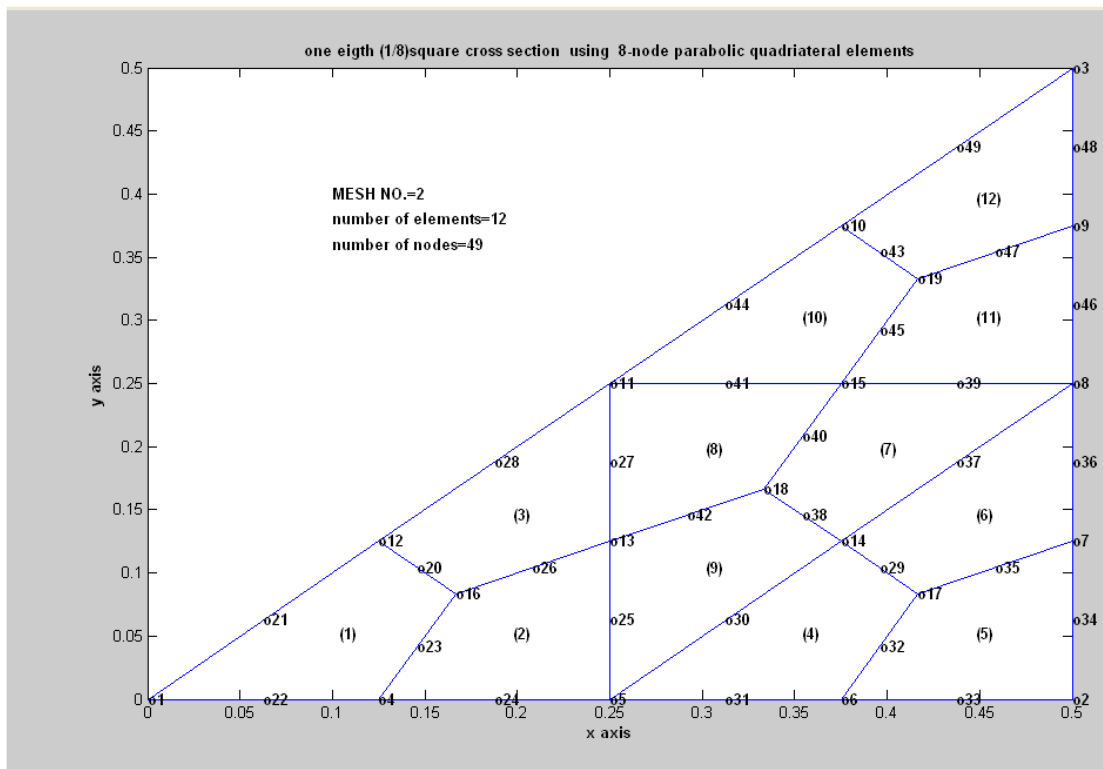
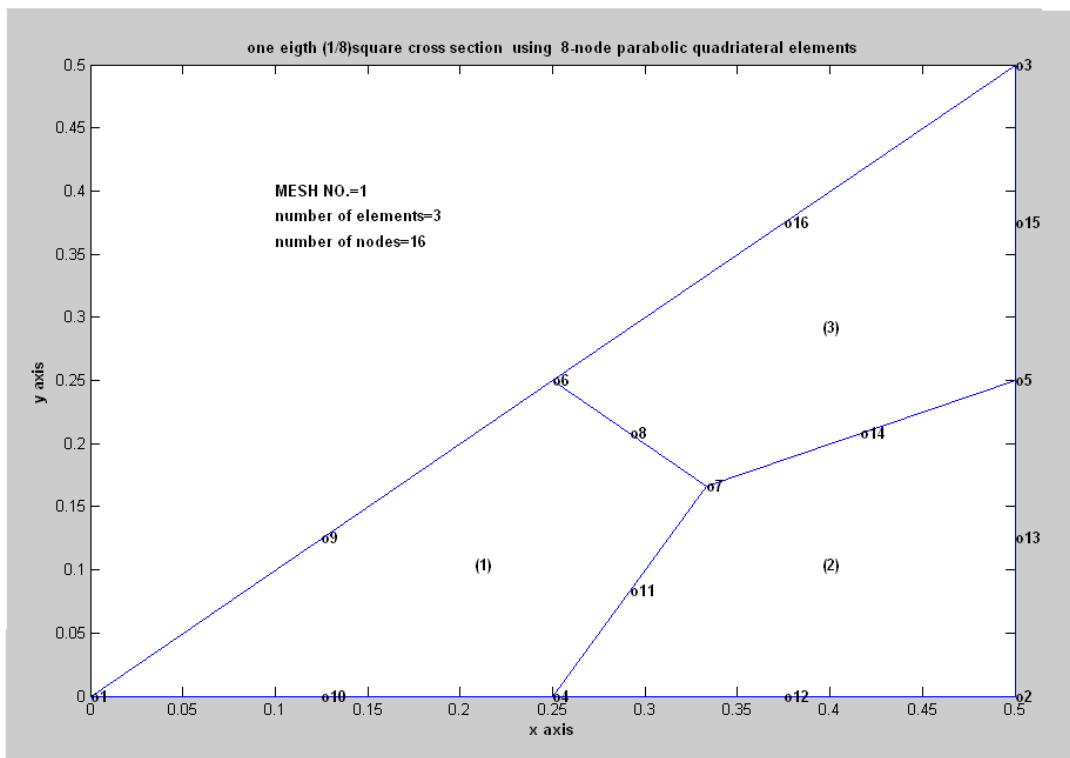
Table-4

EIGHT NODE -LINEAR CONVEX QUADRILATERALS(SERENDIPITY ELEMEMENTS)  
 (exact solution of torisonal constant= 0.140577014955156)  
 nnode=number of nodes in theregion R ;  
 nel=number of elements in theregion R;  
 nnel=number of nodes per element  
 NOTE:GRAPHIC DISPLAY OF MESH NO.1-25 ARE APPENDED IN FIGURES SECTION

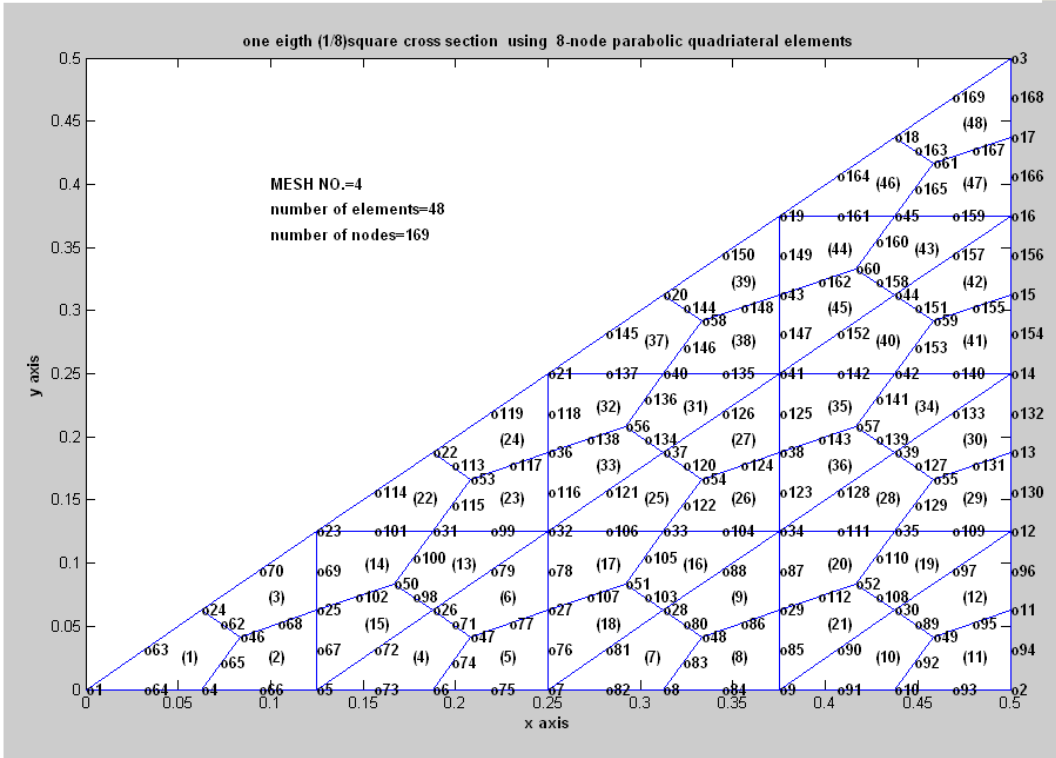
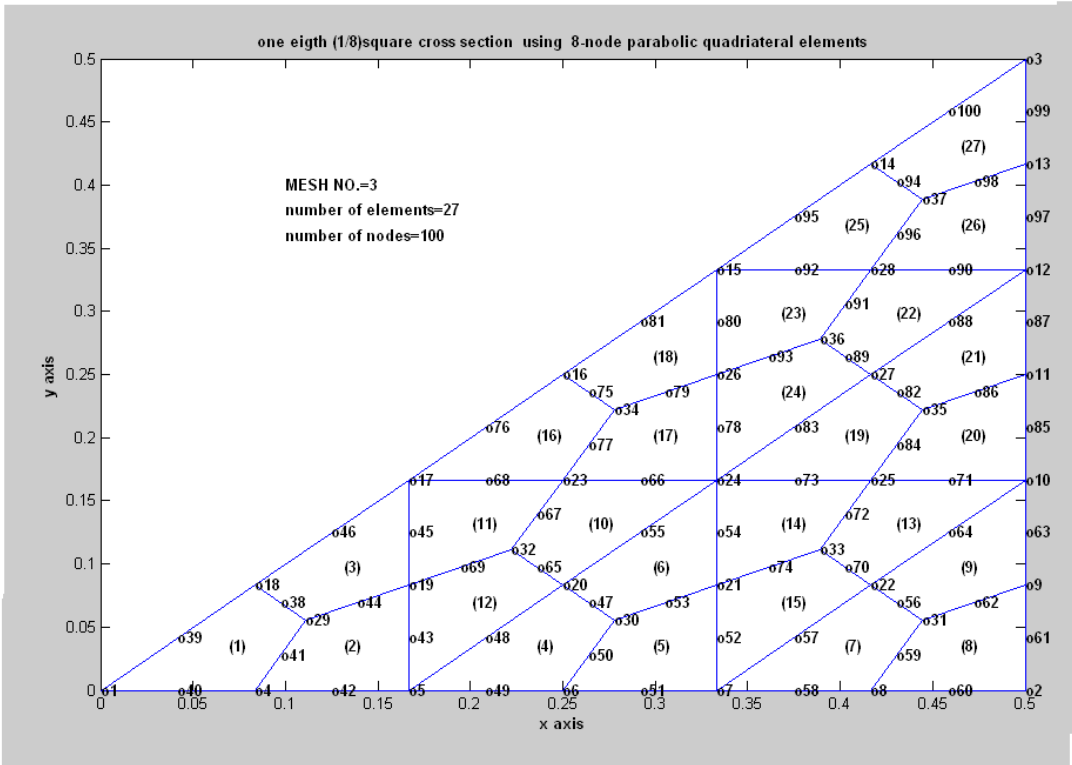
FOR EIGHT NODE QUADRILATERAL MESHES					
(mesh no.)	nnode	nel	nnel	fem solution for torisonal constant	maximum absolute error of Prandtl Stress function values at element nodes
(1)	16	3	8	0.139881192455598	0.00172214821773672
(2)	49	12	8	0.140511125235056	0.000228103295411064
(3)	100	27	8	0.140560146231912	0.000111318093756023
(4)	169	48	8	0.14057033232108	7.57868365891083e-005

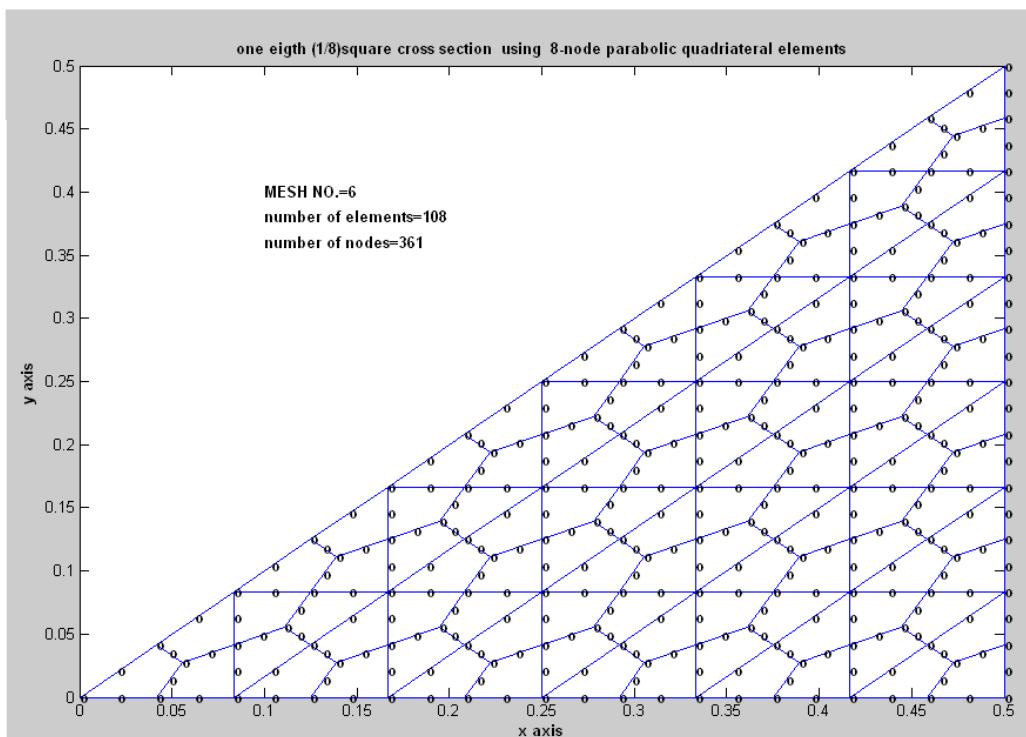
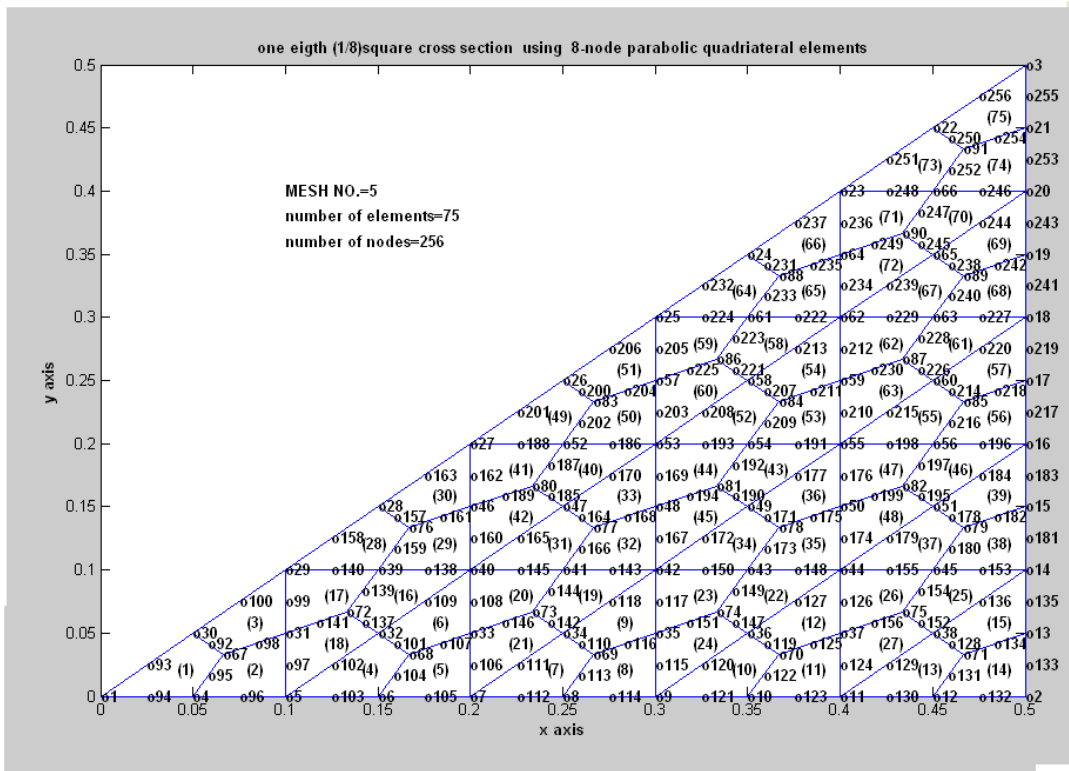
(5)	256	75	8	0.14057364619955	5.71558452171442e-005
(6)	361	108	8	0.14057504105108	4.25097858819104e-005
(7)	484	147	8	0.14057573511073	3.45767462262801e-005
(8)	625	192	8	0.140576123341783	2.94391901602548e-005
(9)	784	243	8	0.140576360001611	2.42499917089884e-005
(10)	961	300	8	0.140576514028352	1.94431366115373e-005
(11)	1156	363	8	0.140576619547978	1.57123956216918e-005
(12)	1369	432	8	0.14057669486575	1.3184733121217476e-005
(13)	1600	507	8	0.140576750456548	1.16333370254527e-005
(14)	1849	588	8	0.14057679264143	1.07440118374911e-005
(15)	2116	675	8	0.140576825408248	1.02449849120308e-005
(16)	2401	768	8	0.140576851369049	9.94170594790372e-006
(17)	2704	867	8	0.140576872290657	9.71043506567621e-006
(18)	3025	972	8	0.140576889401995	9.48027086123348e-006
(19)	3364	1083	8	0.140576903578949	9.21569212390368e-006
(20)	3721	1200	8	0.140576915459166	8.90311431203444e-006
(21)	4096	1323	8	0.140576925515857	8.54159601418455e-006
(22)	4489	1452	8	0.140576934105913	8.13682739378169e-006
(23)	4900	1587	8	0.140576941502974	7.6974375320801e-006
(24)	5329	1728	8	0.140576947919399	7.23283976170421e-006
(25)	5776	1875	8	0.14057695352226	6.7520539368107e-006

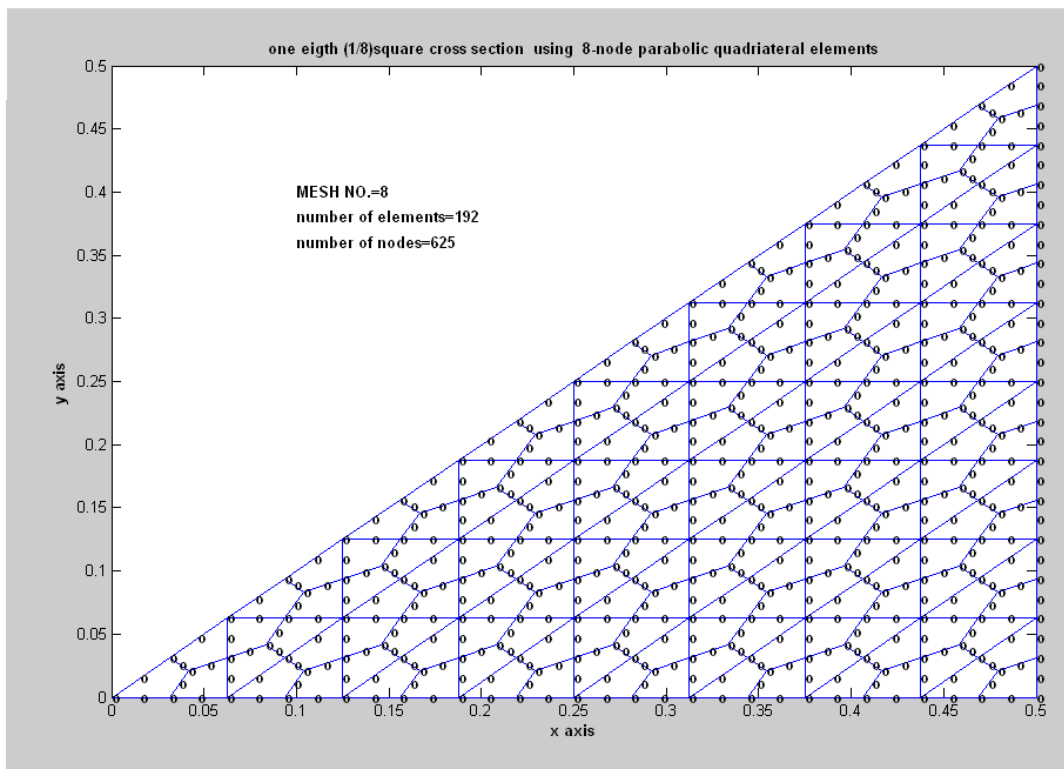
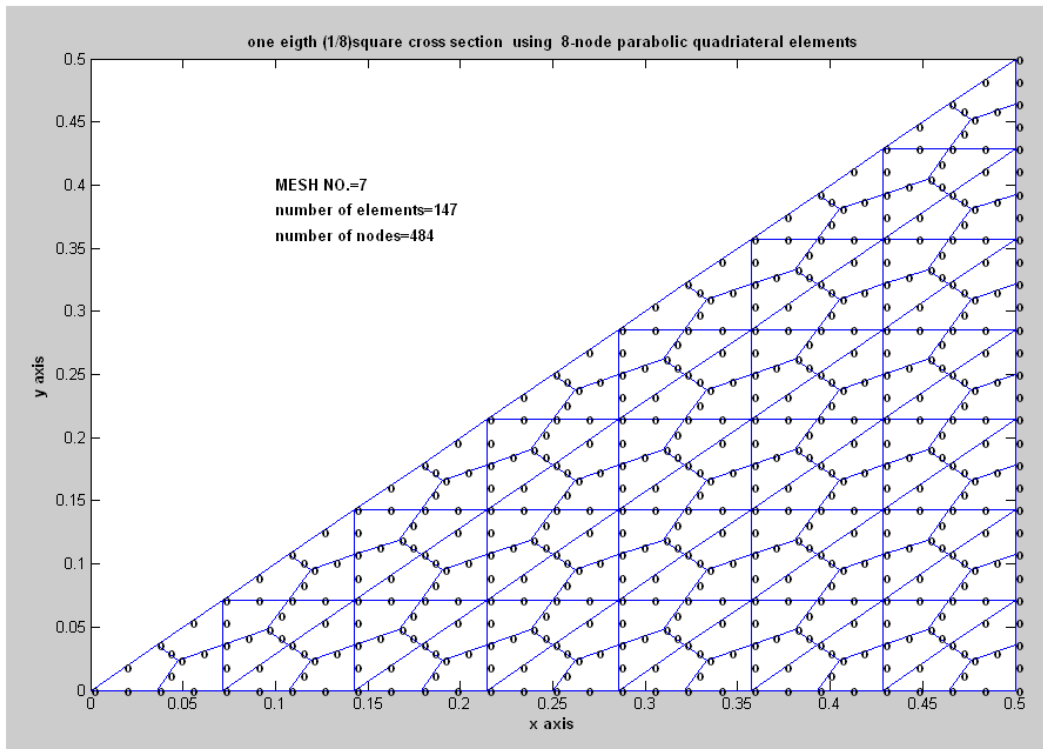
## FIGURES SECTION

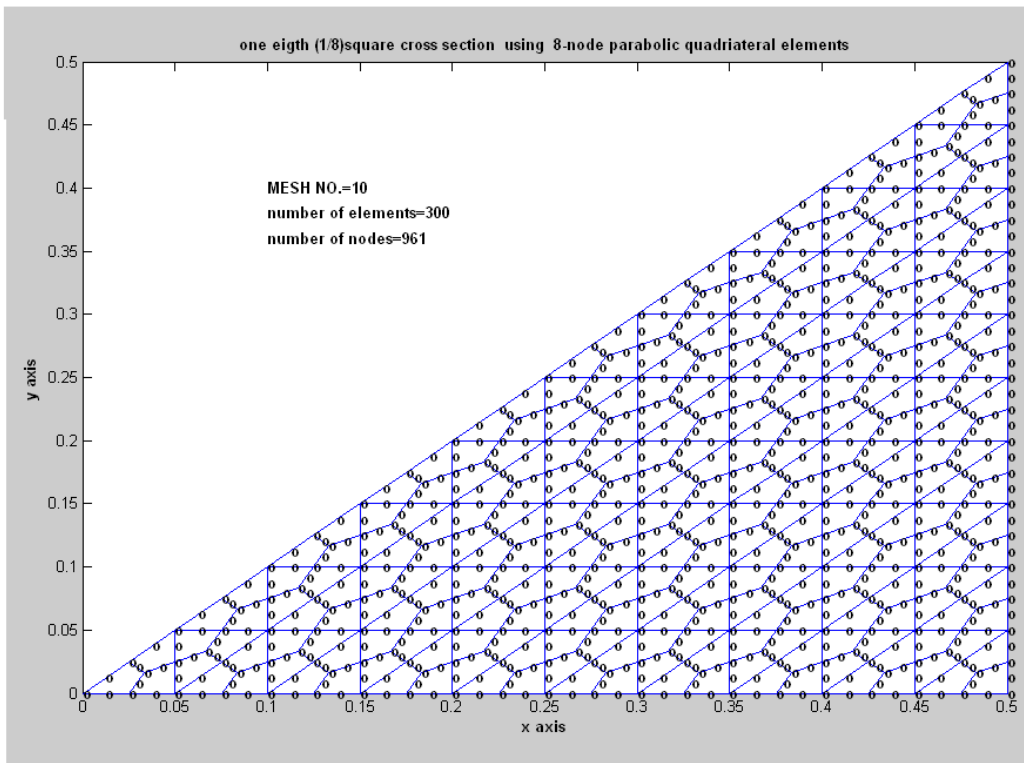
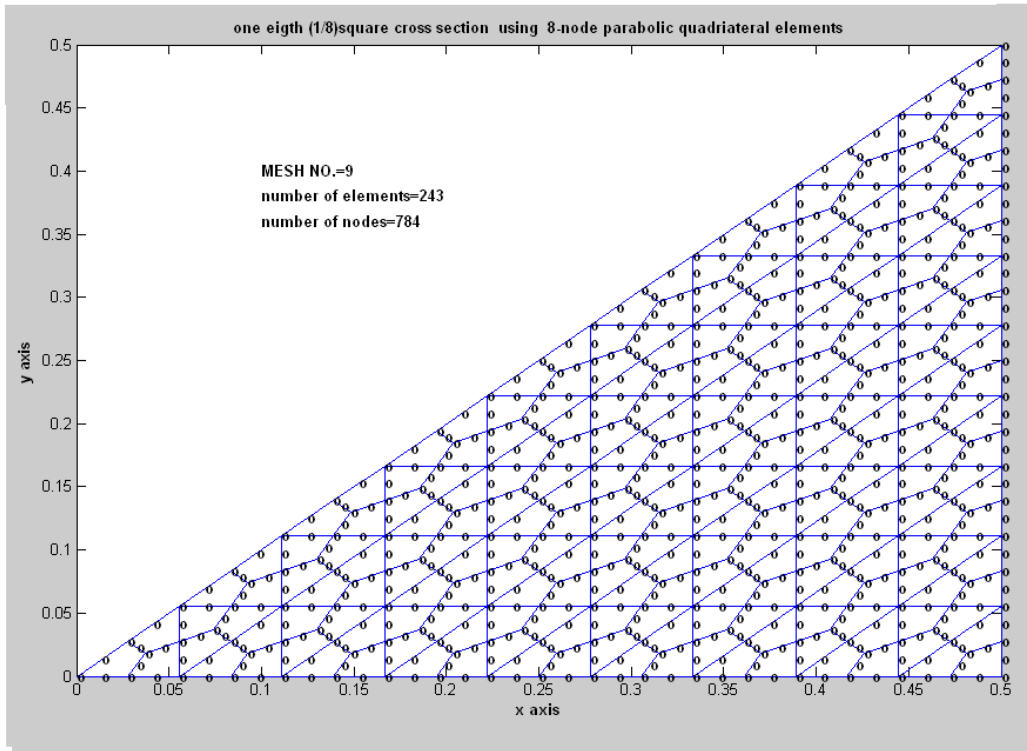


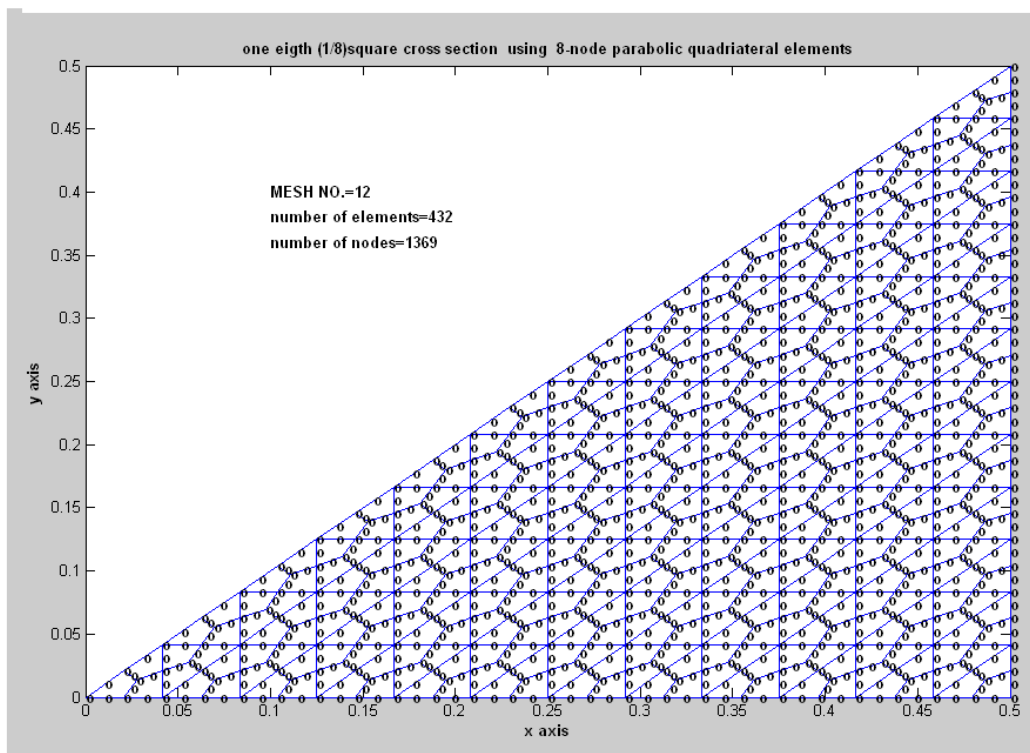
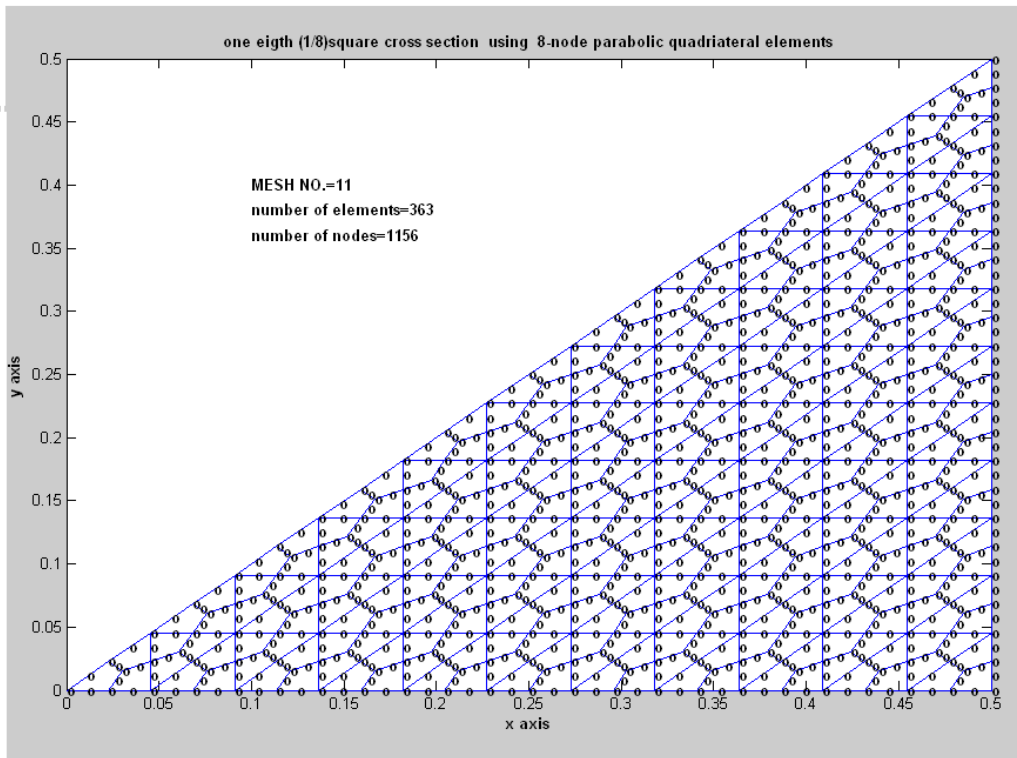


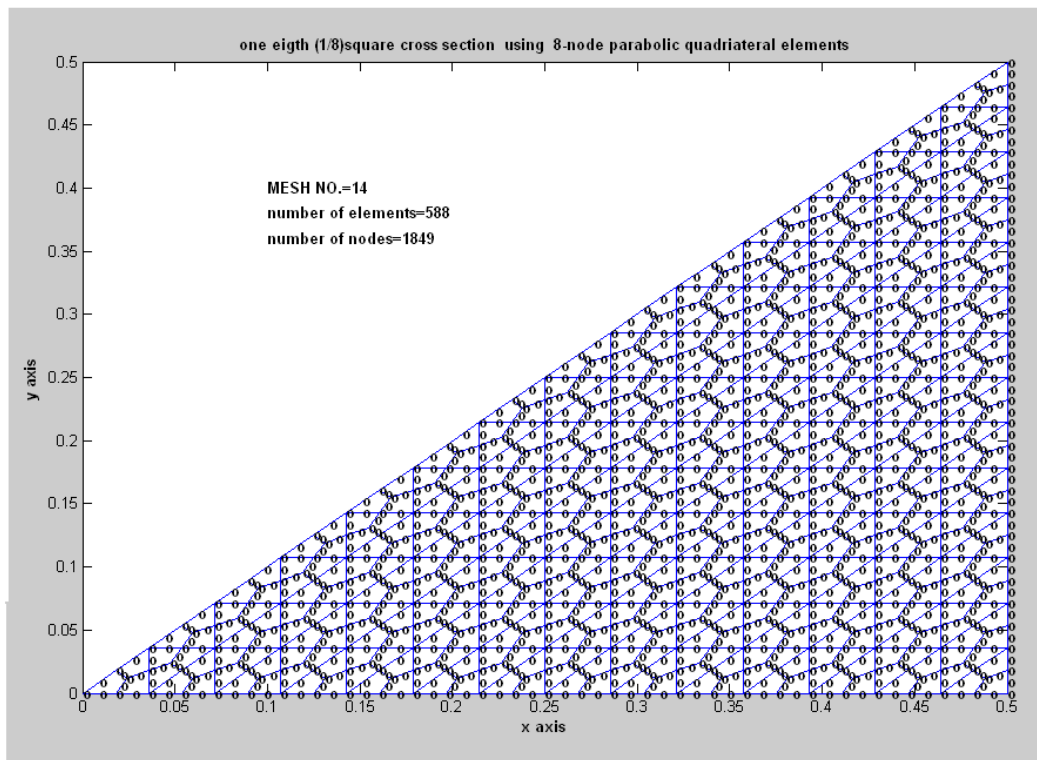
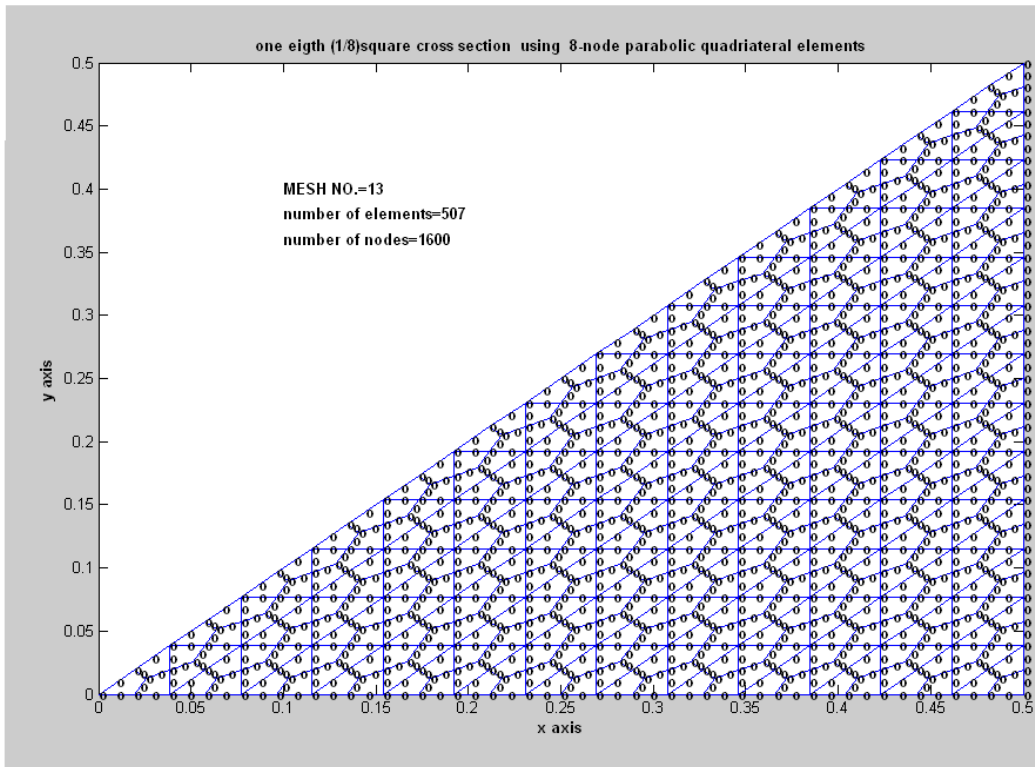


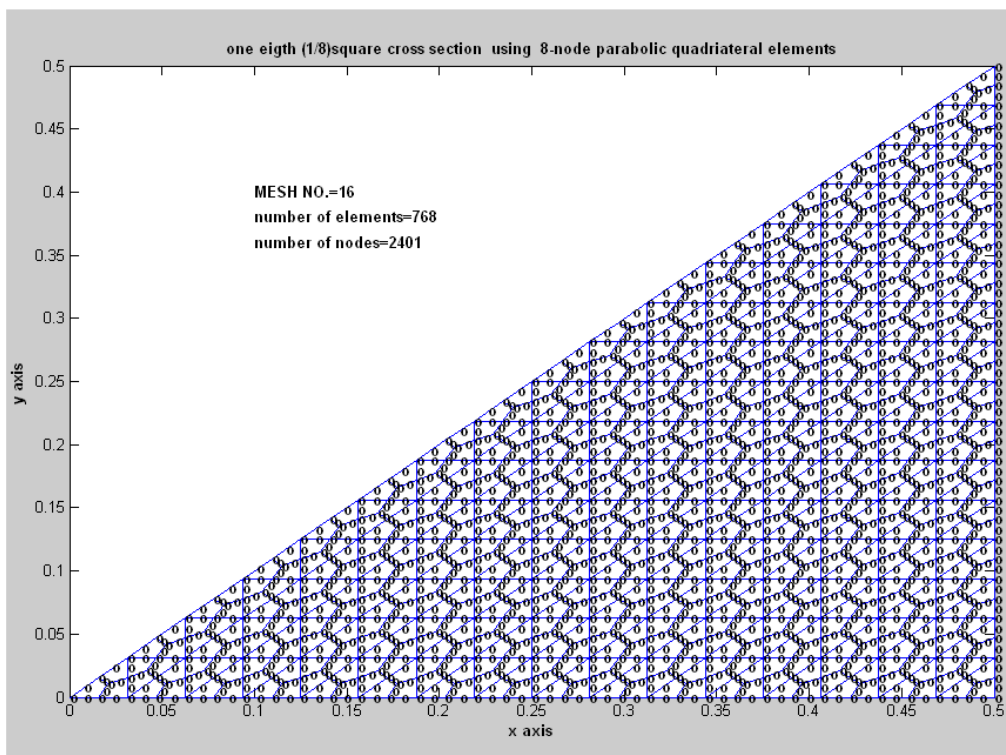
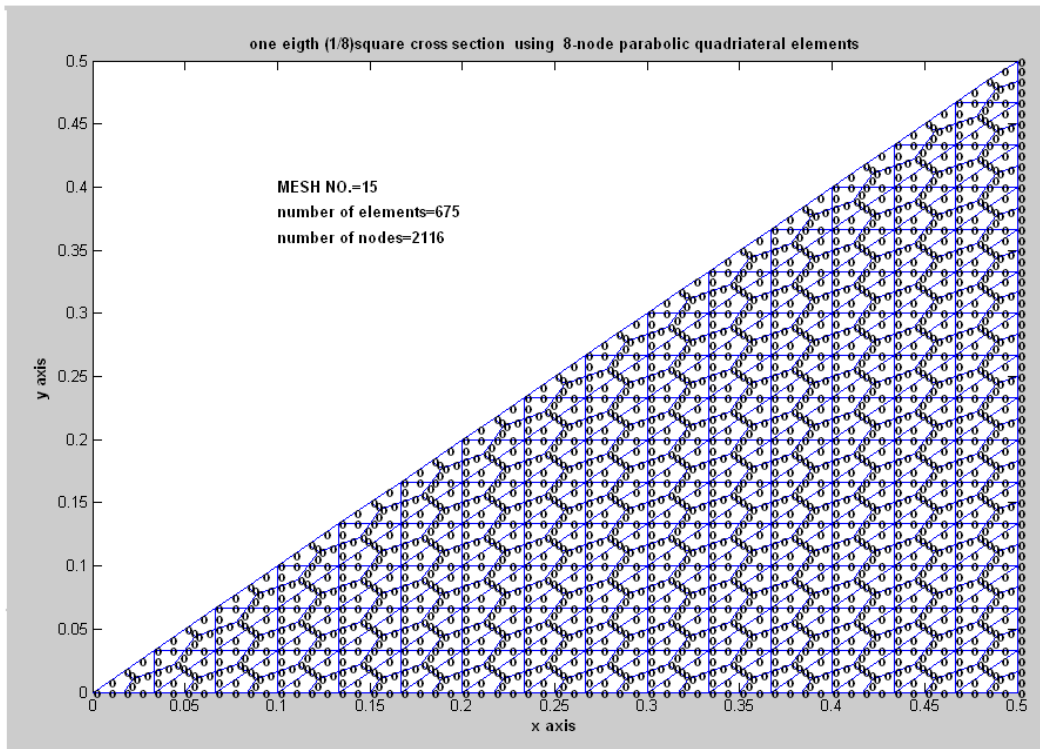


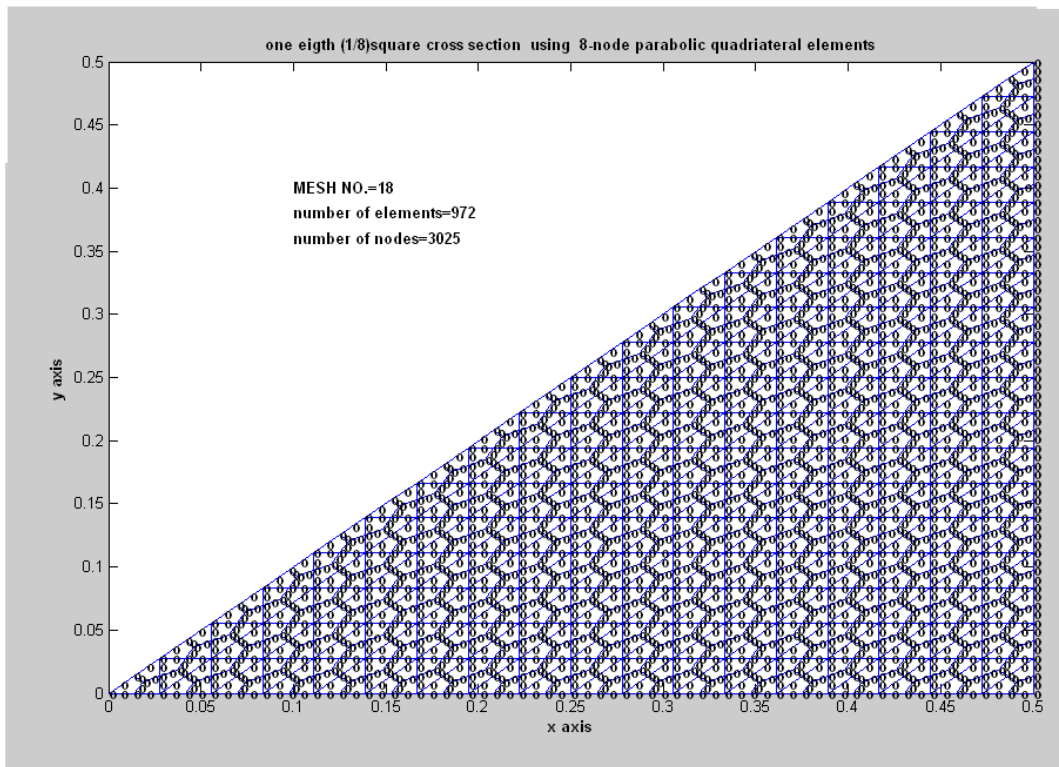
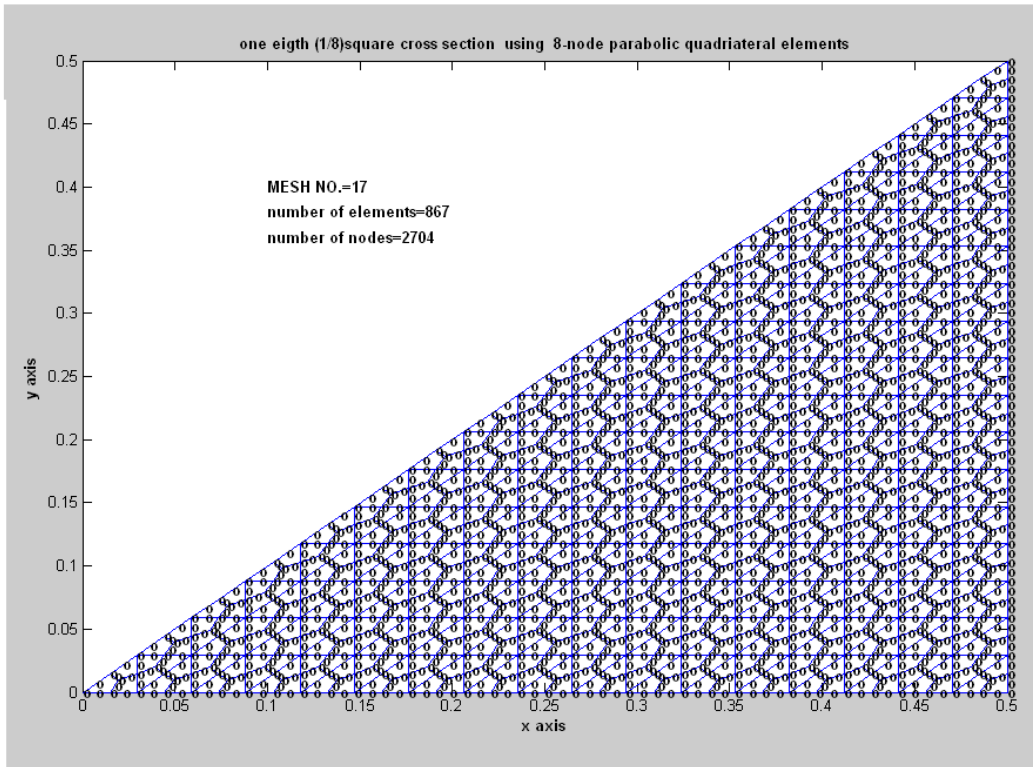




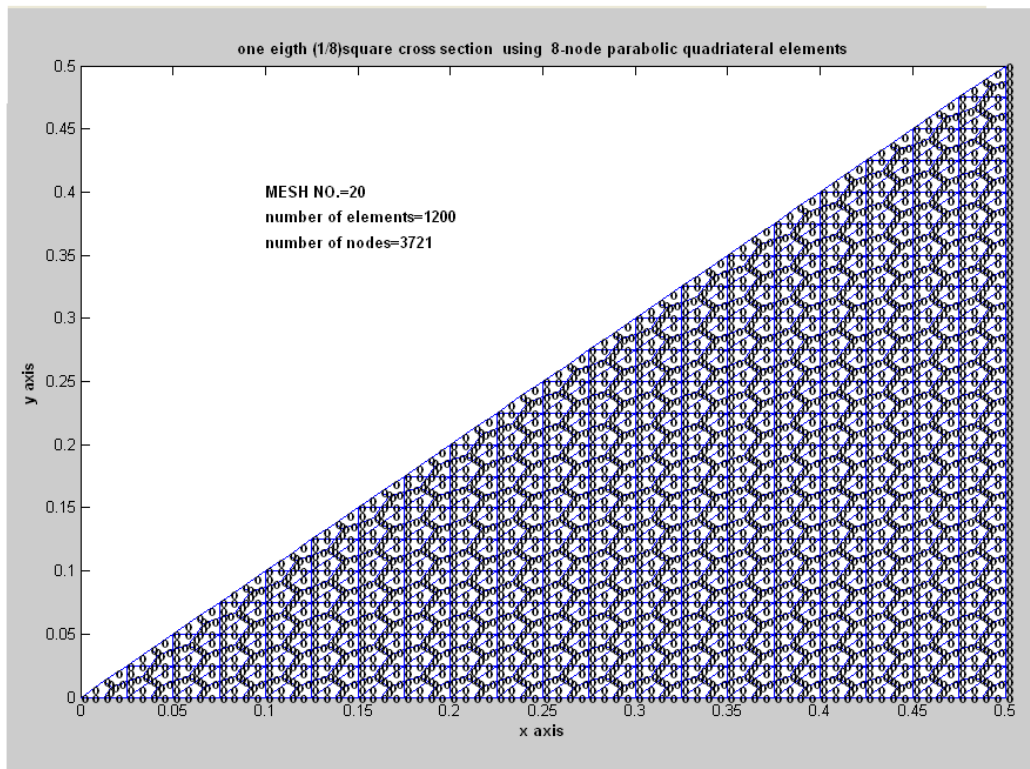
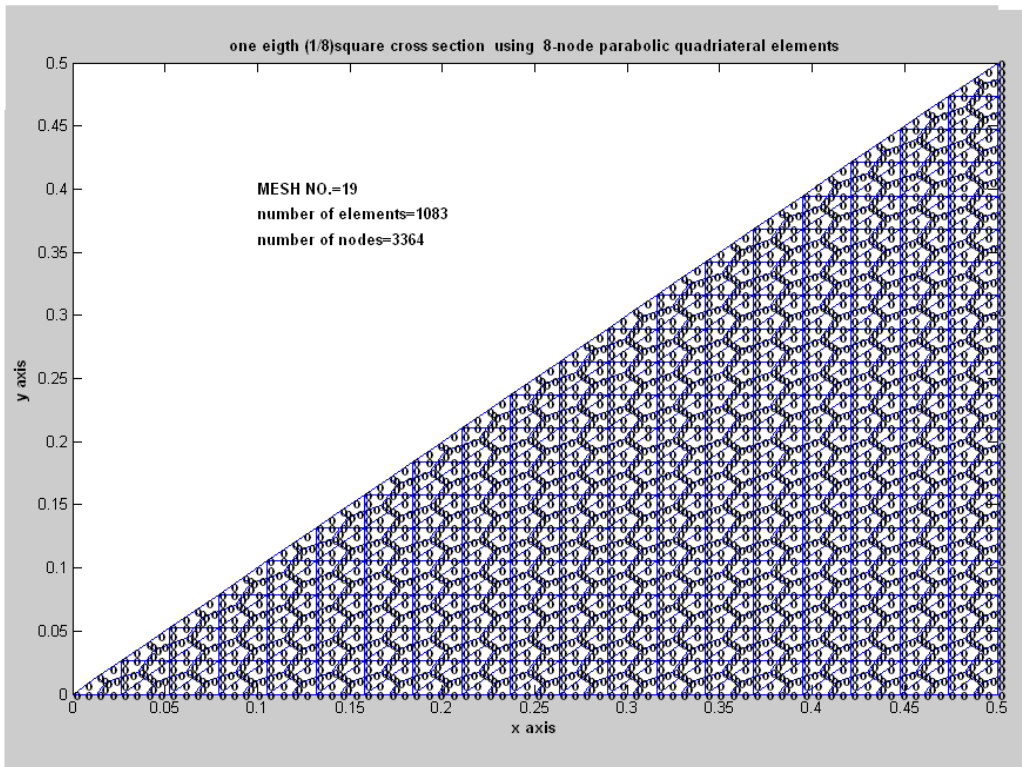


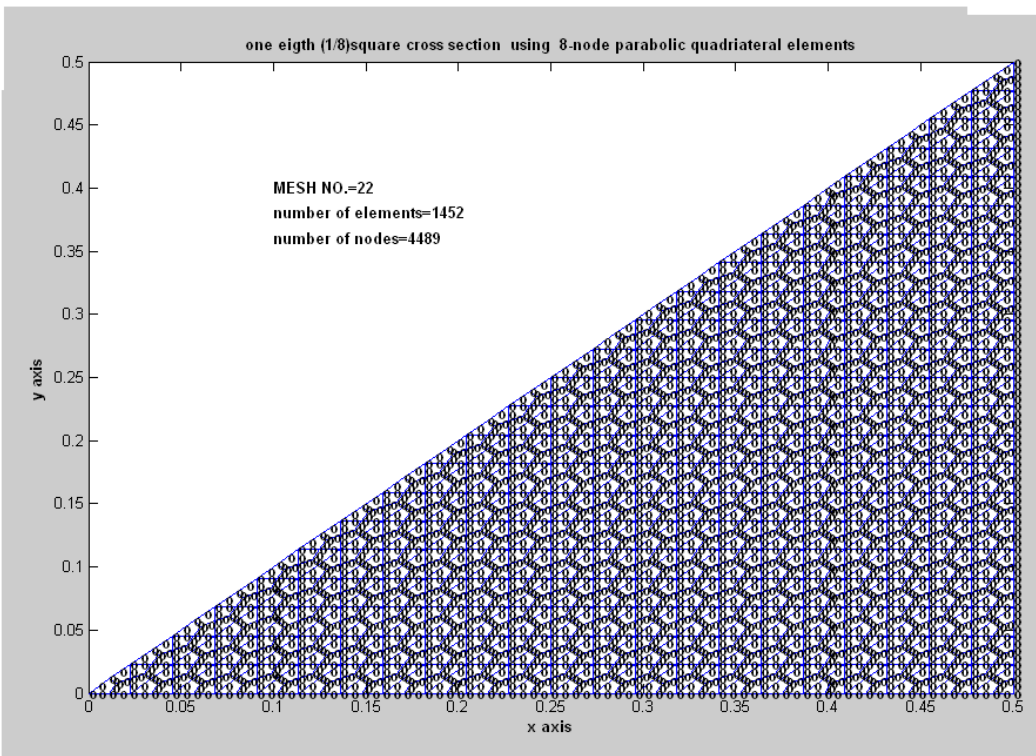
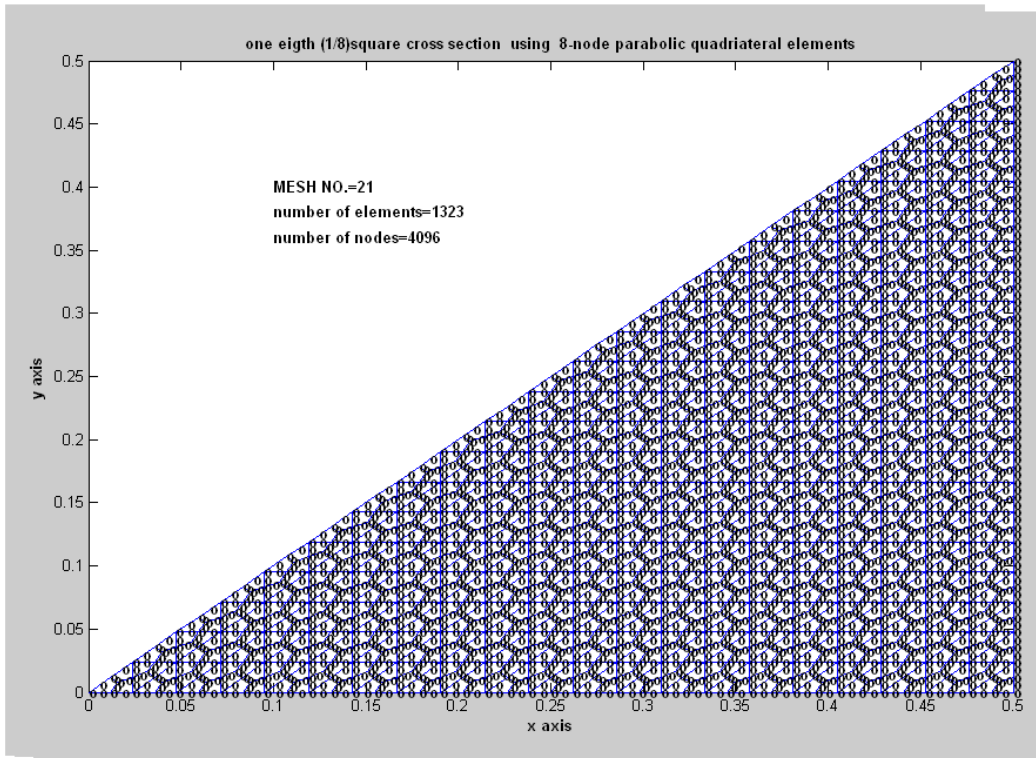


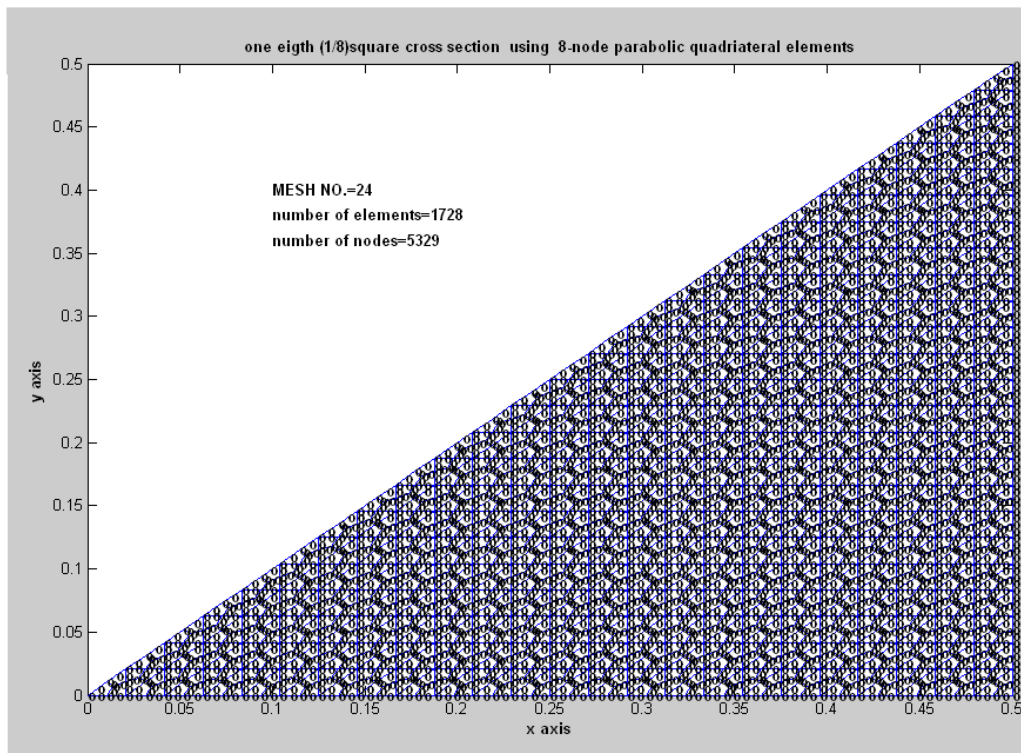
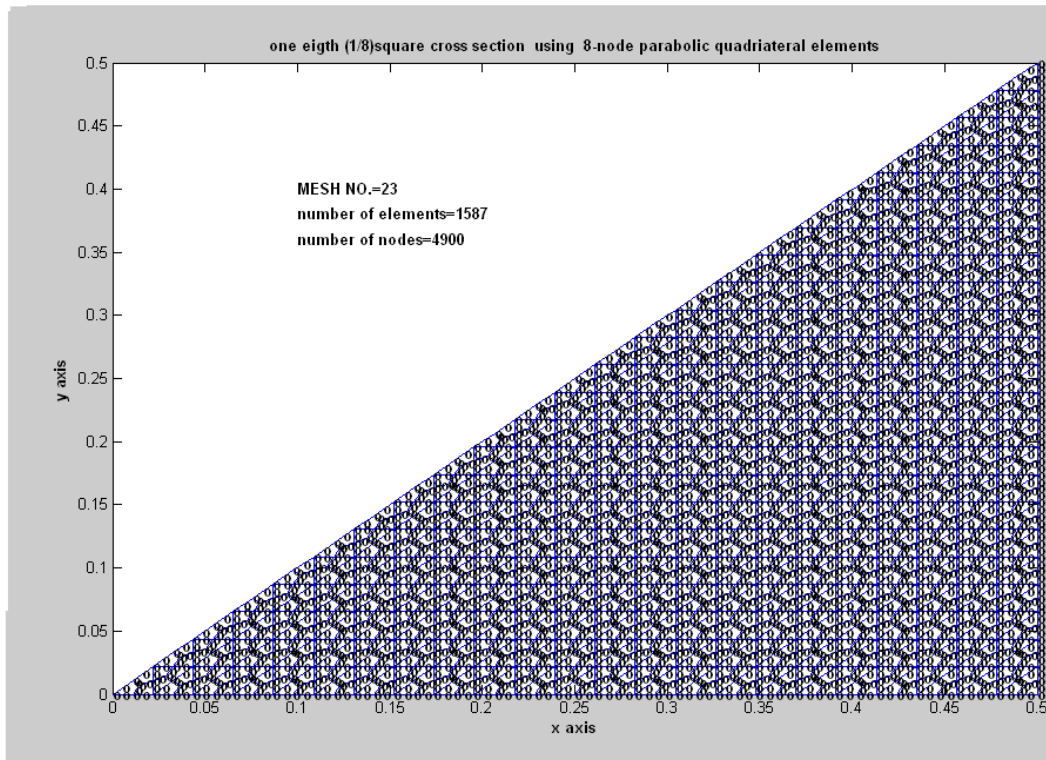


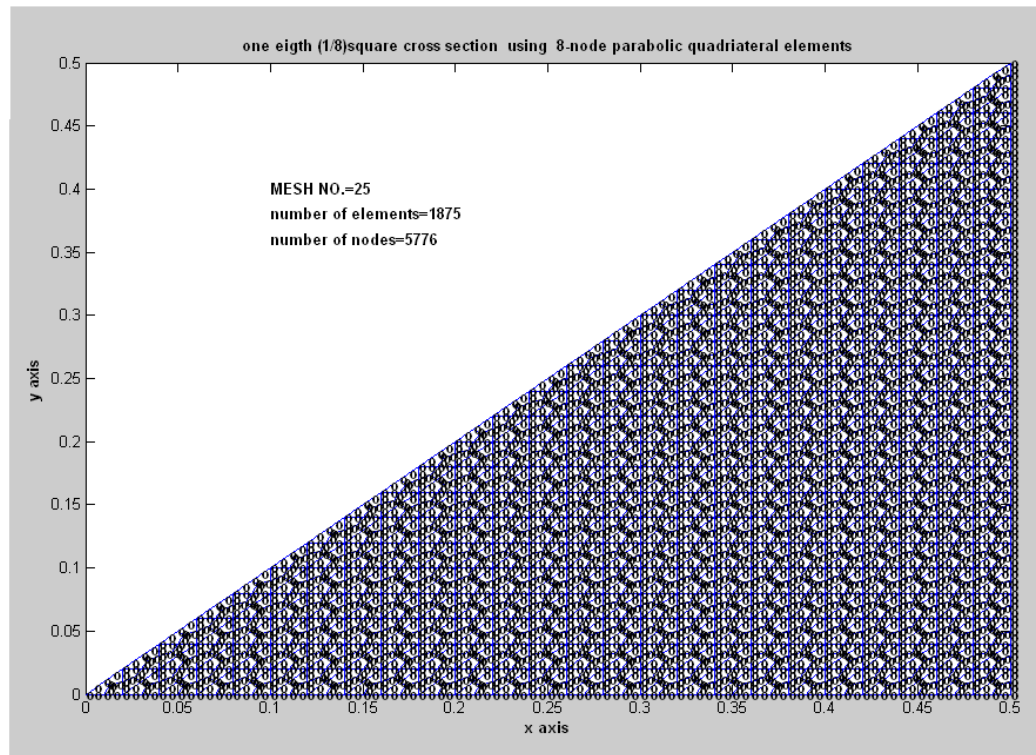












## COMPUTER PROGRAMS

### PROGRAM-1

```
function []=D2LaplaceEquationQ8Ex3automeshgen(ndiv)
syms coord
ndof=1;

[eln,nodetel,nodes,nnode]=nodaladdresses_special_convex_quadrilaterals_2nd_order(ndiv);
[coord,gcoord]=coordinate_rtisoscelestriangle00_h0_hh_2ndorder(ndiv);
[nel,nnel]=size(nodes);
format long g
for i=1:nel
N(i,1)=i;
end
for i=1:nel
NN(i,1)=i;
end
sdof=nnode*ndof;
ff=(zeros(sdof,1));ss=(zeros(sdof,sdof));
format long g
for i=1:nel
N(i,1)=i;
end
table1=[N nodes];
[nel,nnel]=size(nodes);
nnn=0;
for nn=1:nnode
if coord(nn,1)==(1/2)
nnn=nnn+1;
bcdof(nnn,1)=nn;
end
end
```

```

end
bcdof;
mm=length(bcdof);

format long g
k1 =double(0.14057701495515551037840396020329);
xi={zeros(nnode,1)};
a0=8/pi^3;
for m=1:nnode
    x={gcoord(m,1)};y={gcoord(m,2)};rr={0};
for n=1:2:99
    rr=rr+(-1)^((n-1)/2)*(1-(cosh(n*pi*y)/cosh(n*pi/2)))*cos(n*pi*x)/n^3;
end
xi(m,1)={a0*rr};
end

%
% disp(gcoord)

%
%quadtype=3:quadrilateral elements of special shape
%quadtype=0:quadrilateral elements of arbitrary shape
%for el=1:nel
%    elmtype(el,1)=quadtype;%change 0 to 3 to take advantage of special shape
%end

for L=1:nel
    for M=1:3
        LM=nodetel(L,M);
        xx(L,M)=gcoord(LM,1);
        yy(L,M)=gcoord(LM,2);
    end
end
%

table2=[N xx yy];
%disp([xx yy])
intJdn1dn1uvrs =[vpa(sym(' 1.19732437518704939126225670841')),vpa(sym(' 1.07234243081152493747384516139'))];...
    vpa(sym(' 1.07234243081152493747384516139')),vpa(sym(' 1.19732437518704939126225670841'))];

intJdn1dn2uvrs =[vpa(sym(' .39328207524777371271872744686')),vpa(sym(' .66505503018777989604036502927e-1'))];...
    vpa(sym(' .233172169685444656270703169594')),vpa(sym(' .30901830189818397724346918143'))];

intJdn1dn3uvrs =[vpa(sym(' .39105823773230516010581465792')),vpa(sym(' .34520910581966151866889237468'))];...
    vpa(sym(' .34520910581966151866889237468')),vpa(sym(' .39105823773230516010581465792'))];

intJdn1dn4uvrs =[vpa(sym(' .30901830189818397724346918143')),vpa(sym(' .233172169685444656270703169594'))];...
    vpa(sym(' .66505503018777989604036502927e-1')),vpa(sym(' .39328207524777371271872744686'))];

intJdn1dn5uvrs =[vpa(sym(' -1.20601914456199564761927515849')),vpa(sym(' -.19480639850558017529001262923'))];...
    vpa(sym(' -.86147306517224684195667929589')),vpa(sym(' -.27406792898205548885083220480'))];

intJdn1dn6uvrs =[vpa(sym(' -.19025279048691687518778377091')),vpa(sym(' -.33047487282879104238539264173'))];...
    vpa(sym(' -.33047487282879104238539264173')),vpa(sym(' -.620343126034344229672376860410'))];

intJdn1dn7uvrs =[vpa(sym(' -.620343126034344229672376860410')),vpa(sym(' -.33047487282879104238539264173'))];...
    vpa(sym(' -.33047487282879104238539264173')),vpa(sym(' -.19025279048691687518778377091'))];

intJdn1dn8uvrs =[vpa(sym(' -.27406792898205548885083220480')),vpa(sym(' -.86147306517224684195667929589'))];...
    vpa(sym(' -.19480639850558017529001262923')),vpa(sym(' -1.20601914456199564761927515849'))];

intJdn2dn1uvrs =[vpa(sym(' .39328207524777371271872744686')),vpa(sym(' .233172169685444656270703169594'))];...
    vpa(sym(' .66505503018777989604036502927e-1')),vpa(sym(' .30901830189818397724346918143'))];

intJdn2dn2uvrs =[vpa(sym(' .42652636618519994053203031884')),vpa(sym(' -.263513565675283566822536323478'))];...
    vpa(sym(' -.263513565675283566822536323478')),vpa(sym(' .45005030410782929514068096460'))];

intJdn2dn3uvrs =[vpa(sym(' .1595057645371386486635746307258')),vpa(sym(' .15687259169421065039078881847e-1'))];...
    vpa(sym(' .182353925836087731705745548513')),vpa(sym(' .314341351654616447964784833418'))];

```

intJdn2dn4uvrs =[vpa(sym(' .20218069152088195770651481813')),vpa(sym(' -.97977735817538970518647437153e-1'))];...  
vpa(sym(' -.97977735817538970518647437153e-1')),vpa(sym(' .20218069152088195770651481813'))];

intJdn2dn5uvrs =[vpa(sym(' -.92334437129978410949228855299')),vpa(sym(' -.21347210674508644538200765866'))];...  
vpa(sym(' .45319455992158022128465900801')),vpa(sym(' -.574726461686104068369695422e-2'))];

intJdn2dn6uvrs =[vpa(sym(' .219362359001080107967460556119')),vpa(sym(' .3242127618106983385904862884652'))];...  
vpa(sym(' -.3424539048559683280761803782014')),vpa(sym(' -.768806561449731024667082979907'))];

intJdn2dn7uvrs =[vpa(sym(' -.390687025888983122842003915751')),vpa(sym(' -.36332930895951737339235932476e-1'))];...  
vpa(sym(' -.36332930895951737339235932476e-1')),vpa(sym(' -.337606440451462448388879556e-1'))];

intJdn2dn8uvrs =[vpa(sym(' -.8682585930330713525401530193e-1')),vpa(sym(' .38224148468296660162159011855e-1'))];...  
vpa(sym(' .38224148468296660162159011855e-1')),vpa(sym(' -.467276179069773367865781067869'))];

intJdn3dn1uvrs =[vpa(sym(' .39105823773230516010581465792')),vpa(sym(' .34520910581966151866889237468'))];...  
vpa(sym(' .34520910581966151866889237468')),vpa(sym(' .39105823773230516010581465792'))];

intJdn3dn2uvrs =[vpa(sym(' .1595057645371386486635746307258')),vpa(sym(' .182353925836087731705745548513'))];...  
vpa(sym(' .15687259169421065039078881847e-1')),vpa(sym(' .314341351654616447964784833418'))];

intJdn3dn3uvrs =[vpa(sym(' .660112740472589759408893364145')),vpa(sym(' .619284819567235241935951286773'))];...  
vpa(sym(' .619284819567235241935951286773')),vpa(sym(' .660112740472589759408893364145'))];

intJdn3dn4uvrs =[vpa(sym(' .314341351654616447964784833418')),vpa(sym(' .15687259169421065039078881847e-1'))];...  
vpa(sym(' .182353925836087731705745548513')),vpa(sym(' .1595057645371386486635746307258'))];

intJdn3dn5uvrs =[vpa(sym(' -.447323988028158265936415185565')),vpa(sym(' -.182789328176641032221385163017'))];...  
vpa(sym(' -.182789328176641032221385163017')),vpa(sym(' -.15179959092659718711045271183'))];

intJdn3dn6uvrs =[vpa(sym(' .6317194613977723832758311823e-2')),vpa(sym(' -.731811560352895079786782216220'))];...  
vpa(sym(' -.65144893686228413120115549553e-1')),vpa(sym(' -.932211710055872286928957900634'))];

intJdn3dn7uvrs =[vpa(sym(' -.932211710055872286928957900634')),vpa(sym(' -.65144893686228413120115549553e-1'))];...  
vpa(sym(' -.731811560352895079786782216220')),vpa(sym(' .6317194613977723832758311823e-2'))];

intJdn3dn8uvrs =[vpa(sym(' -.15179959092659718711045271183')),vpa(sym(' -.182789328176641032221385163017'))];...  
vpa(sym(' -.182789328176641032221385163017')),vpa(sym(' -.447323988028158265936415185565'))];

intJdn4dn1uvrs =[vpa(sym(' .30901830189818397724346918143')),vpa(sym(' .6650550301877989604036502927e-1'))];...  
vpa(sym(' .233172169685444656270703169594')), vpa(sym(' .39328207524777371271872744686'))];

intJdn4dn2uvrs =[vpa(sym(' .20218069152088195770651481813')),vpa(sym(' -.97977735817538970518647437153e-1'))];...  
vpa(sym(' -.97977735817538970518647437153e-1')),vpa(sym(' .20218069152088195770651481813'))];

intJdn4dn3uvrs =[vpa(sym(' .314341351654616447964784833418')),vpa(sym(' .182353925836087731705745548513'))];...  
vpa(sym(' .15687259169421065039078881847e-1')),vpa(sym(' .1595057645371386486635746307258'))];

intJdn4dn4uvrs =[vpa(sym(' .45005030410782929514068096460')),vpa(sym(' -.263513565675283566822536323478'))];...  
vpa(sym(' -.263513565675283566822536323478')),vpa(sym(' .42652636618519994053203031884'))];

intJdn4dn5uvrs =[vpa(sym(' -.467276179069773367865781067869')),vpa(sym(' .38224148468296660162159011855e-1'))];...  
vpa(sym(' .38224148468296660162159011855e-1')),vpa(sym(' -.8682585930330713525401530193e-1'))];

intJdn4dn6uvrs =[vpa(sym(' -.337606440451462448388879556e-1')),vpa(sym(' -.36332930895951737339235932476e-1'))];...  
vpa(sym(' -.36332930895951737339235932476e-1')),vpa(sym(' -.390687025888983122842003915751'))];

intJdn4dn7uvrs =[vpa(sym(' -.768806561449731024667082979907')),vpa(sym(' -.3424539048559683280761803782014'))];...  
vpa(sym(' .3242127618106983385904862884652')),vpa(sym(' .219362359001080107967460556119'))];

intJdn4dn8uvrs =[vpa(sym(' -.574726461686104068369695422e-2')),vpa(sym(' .45319455992158022128465900801'))];...  
vpa(sym(' -.21347210674508644538200765866')),vpa(sym(' -.92334437129978410949228855299'))];

intJdn5dn1uvrs =[vpa(sym(' -.12061914456199564761927515849')),vpa(sym(' -.86147306517224684195667929589'))];...  
vpa(sym(' -.19480639850558017529001262923')),vpa(sym(' -.27406792898205548885083220480'))];

intJdn5dn2uvrs =[vpa(sym(' -.92334437129978410949228855299')),vpa(sym(' .45319455992158022128465900801'))];...  
vpa(sym(' -.21347210674508644538200765866')),vpa(sym(' -.574726461686104068369695422e-2'))];

intJdn5dn3uvrs =[vpa(sym(' -.447323988028158265936415185565')),vpa(sym(' -.182789328176641032221385163017'))];...

vpa(sym('-.182789328176641032221385163017')),vpa(sym('-.15179959092659718711045271183')));

intJdn5dn4uvrs=[vpa(sym('-.467276179069773367865781067869')),vpa(sym(' .38224148468296660162159011855e-1'))];...  
vpa(sym(' .38224148468296660162159011855e-1')),vpa(sym('-.8682585930330713525401530193e-1')));

intJdn5dn5uvrs=[vpa(sym('2.17177154262400226248727022336')),vpa(sym(' .52992039898580294331102892478'))];...  
vpa(sym(' .52992039898580294331102892478')),vpa(sym(' .5882735844715392304142536963')));

intJdn5dn6uvrs=[vpa(sym('-.415943731136358047626104977997')),vpa(sym('-.446246745737358785845683479200'))];...  
vpa(sym('-.446246745737358785845683479200')),vpa(sym('-.34312587200869545409370986785e-1')));

intJdn5dn7uvrs=[vpa(sym(' .91157772830419213711619683500')),vpa(sym(' .9457179332261632804889985221e-1'))];...  
vpa(sym(' .9457179332261632804889985221e-1')),vpa(sym('-.48070367101146296286102539491')));

intJdn5dn8uvrs=[vpa(sym(' .376558143167875038936397884560')),vpa(sym(' .37459823838795050721700114125'))];...  
vpa(sym(' .37459823838795050721700114125')),vpa(sym(' .376558143167875038936397884560')));

intJdn6dn1uvrs=[vpa(sym('-.19025279048691687518778377091')),vpa(sym('-.33047487282879104238539264173'))];...  
vpa(sym('-.33047487282879104238539264173')),vpa(sym('-.620343126034344229672376860410')));

intJdn6dn2uvrs=[vpa(sym(' .219362359001080107967460556119')),vpa(sym('-.3424539048559683280761803782014'))];...  
vpa(sym(' .3242127618106983385904862884652')),vpa(sym('-.768806561449731024667082979907')));

intJdn6dn3uvrs=[vpa(sym(' .631719461397723832758311823e-2')),vpa(sym('-.65144893686228413120115549553e-1'))];...  
vpa(sym('-.731811560352895079786782216220')),vpa(sym('-.932211710055872286928957900634')));

intJdn6dn4uvrs=[vpa(sym('-.3376064404514624483888879556e-1')),vpa(sym('-.36332930895951737339235932476e-1'))];...  
vpa(sym('-.36332930895951737339235932476e-1')),vpa(sym('-.390687025888983122842003915751')));

intJdn6dn5uvrs=[vpa(sym('-.415943731136358047626104977997')),vpa(sym('-.446246745737358785845683479200'))];...  
vpa(sym('-.446246745737358785845683479200')),vpa(sym('-.34312587200869545409370986785e-1')));

intJdn6dn6uvrs=[vpa(sym(' .82865433521530100559964914224')),vpa(sym(' .472904354164178470376423509241'))];...  
vpa(sym(' .472904354164178470376423509241')),vpa(sym('1.699831160074343688470918905720')));

intJdn6dn7uvrs=[vpa(sym(' .66326947849525293113934929200e-1')),vpa(sym(' .653177200517503508341284619711'))];...  
vpa(sym(' .653177200517503508341284619711')),vpa(sym(' .66326947849525293113934929200e-1')));

intJdn6dn8uvrs=[vpa(sym('-.48070367101146296286102539491')),vpa(sym(' .9457179332261632804889985221e-1'))];...  
vpa(sym(' .9457179332261632804889985221e-1')),vpa(sym(' .91157772830419213711619683500')));

intJdn7dn1uvrs=[vpa(sym('-.620343126034344229672376860410')),vpa(sym('-.33047487282879104238539264173'))];...  
vpa(sym('-.33047487282879104238539264173')),vpa(sym('-.19025279048691687518778377091')));

intJdn7dn2uvrs=[vpa(sym('-.390687025888983122842003915751')),vpa(sym('-.36332930895951737339235932476e-1'))];...  
vpa(sym('-.36332930895951737339235932476e-1')),vpa(sym('-.337606440451462448388879556e-1')));

intJdn7dn3uvrs=[vpa(sym('-.932211710055872286928957900634')),vpa(sym('-.731811560352895079786782216220'))];...  
vpa(sym('-.65144893686228413120115549553e-1')),vpa(sym(' .631719461397723832758311823e-2')));

intJdn7dn4uvrs=[vpa(sym('-.768806561449731024667082979907')),vpa(sym(' .3242127618106983385904862884652'))];...  
vpa(sym('-.3424539048559683280761803782014')),vpa(sym(' .219362359001080107967460556119')));

intJdn7dn5uvrs=[vpa(sym(' .91157772830419213711619683500')),vpa(sym(' .9457179332261632804889985221e-1'))];...  
vpa(sym(' .9457179332261632804889985221e-1')),vpa(sym('-.48070367101146296286102539491')));

intJdn7dn6uvrs=[vpa(sym(' .66326947849525293113934929200e-1')),vpa(sym(' .653177200517503508341284619711'))];...  
vpa(sym(' .653177200517503508341284619711')),vpa(sym(' .66326947849525293113934929200e-1')));

intJdn7dn7uvrs=[vpa(sym('1.699831160074343688470918905720')),vpa(sym(' .472904354164178470376423509241'))];...  
vpa(sym(' .472904354164178470376423509241')),vpa(sym(' .82865433521530100559964914224')));

intJdn7dn8uvrs=[vpa(sym('-.34312587200869545409370986785e-1')),vpa(sym('-.446246745737358785845683479200'))];...  
vpa(sym('-.446246745737358785845683479200')),vpa(sym('-.415943731136358047626104977997')));

intJdn8dn1uvrs=[vpa(sym('-.27406792898205548885083220480')),vpa(sym('-.19480639850558017529001262923'))];...  
vpa(sym('-.86147306517224684195667929589')),vpa(sym(' -1.20601914456199564761927515849')));

intJdn8dn2uvrs=[vpa(sym('-.8682585930330713525401530193e-1')),vpa(sym(' .38224148468296660162159011855e-1'))];...  
vpa(sym(' .38224148468296660162159011855e-1')),vpa(sym('-.467276179069773367865781067869')));

```

intJdn8dn3uvrs =[vpa(sym(' -.15179959092659718711045271183')),vpa(sym(' -.182789328176641032221385163017'))];...
vpa(sym(' -.182789328176641032221385163017')),vpa(sym(' -.447323988028158265936415185565'))];

intJdn8dn4uvrs =[vpa(sym(' -.574726461686104068369695422e-2')),vpa(sym(' -.21347210674508644538200765866'))];...
vpa(sym(' .45319455992158022128465900801')),vpa(sym(' -.92334437129978410949228855299'))];

intJdn8dn5uvrs =[vpa(sym(' .376558143167875038936397884560')),vpa(sym(' .37459823838795050721700114125'))];...
vpa(sym(' .37459823838795050721700114125')),vpa(sym(' .376558143167875038936397884560'))];

intJdn8dn6uvrs =[vpa(sym(' -.48070367101146296286102539491')),vpa(sym(' .9457179332261632804889985221e-1'))];...
vpa(sym(' .9457179332261632804889985221e-1')),vpa(sym(' .91157772830419213711619683500'))];

intJdn8dn7uvrs =[vpa(sym(' .34312587200869545409370986785e-1')),vpa(sym(' -.446246745737358785845683479200'))];...
vpa(sym(' -.446246745737358785845683479200')),vpa(sym(' -.415943731136358047626104977997'))];

intJdn8dn8uvrs =[vpa(sym(' .5882735844715392304142536963')),vpa(sym(' .52992039898580294331102892478'))];...
vpa(sym(' .52992039898580294331102892478')),vpa(sym(' 2.17177154262400226248727022336'))];

```

```

%=====
intJdndn=[intJdn1dn1uvrs intJdn1dn2uvrs intJdn1dn3uvrs intJdn1dn4uvrs intJdn1dn5uvrs intJdn1dn6uvrs intJdn1dn7uvrs
intJdn1dn8uvrs;...
intJdn2dn1uvrs intJdn2dn2uvrs intJdn2dn3uvrs intJdn2dn4uvrs intJdn2dn5uvrs intJdn2dn6uvrs intJdn2dn7uvrs intJdn2dn8uvrs;...
intJdn3dn1uvrs intJdn3dn2uvrs intJdn3dn3uvrs intJdn3dn4uvrs intJdn3dn5uvrs intJdn3dn6uvrs intJdn3dn7uvrs intJdn3dn8uvrs;...
intJdn4dn1uvrs intJdn4dn2uvrs intJdn4dn3uvrs intJdn4dn4uvrs intJdn4dn5uvrs intJdn4dn6uvrs intJdn4dn7uvrs intJdn4dn8uvrs;...
intJdn5dn1uvrs intJdn5dn2uvrs intJdn5dn3uvrs intJdn5dn4uvrs intJdn5dn5uvrs intJdn5dn6uvrs intJdn5dn7uvrs intJdn5dn8uvrs;...
intJdn6dn1uvrs intJdn6dn2uvrs intJdn6dn3uvrs intJdn6dn4uvrs intJdn6dn5uvrs intJdn6dn6uvrs intJdn6dn7uvrs intJdn6dn8uvrs;...
intJdn7dn1uvrs intJdn7dn2uvrs intJdn7dn3uvrs intJdn7dn4uvrs intJdn7dn5uvrs intJdn7dn6uvrs intJdn7dn7uvrs intJdn7dn8uvrs;...
intJdn8dn1uvrs intJdn8dn2uvrs intJdn8dn3uvrs intJdn8dn4uvrs intJdn8dn5uvrs intJdn8dn6uvrs intJdn8dn7uvrs intJdn8dn8uvrs];

```

```
intJdndn=double(intJdndn);
```

```
for iel=1:nel
index=zeros(nnel*ndof,1);
```

```

X=xx(iel,1:3);
Y=yy(iel,1:3);
%disp([X Y])
xa=X(1,1);
xb=X(1,2);
xc=X(1,3);
ya=Y(1,1);
yb=Y(1,2);
yc=Y(1,3);
bta=yb-yc;btb=yc-ya;
gma=xc-xb;gmb=xa-xc;
delabc=gmb*bta-gma*btb;
G=[bta btb;gma gmb]/delabc;
GT=[bta gma;btb gmb]/delabc;
Q=GT*G;

```

```

sk(1:8,1:8)=(zeros(8,8));
for i=1:8
for j=i:8
sk(i,j)=(delabc*sum(sum(Q.*(intJdndn(2*i-1:2*i,2*j-1:2*j)))));
sk(j,i)=sk(i,j);
end
end
f=[ -7/432; -1/72; -5/432; -1/72; 11/216; 13/216; 13/216; 11/216]*(2*delabc);

```

```

%_____
edof=nnel*ndof;
k=0;
for i=1:nnel
nd(i,1)=nodes(iel,i);
start=(nd(i,1)-1)*ndof;

```



```

    for j=1:ndof
        k=k+1;
        index(k,1)=start+j;
    end
end
%-----
for i=1:edof
    ii=index(i,1);
    ff(ii,1)=ff(ii,1)+f(i,1);
    for j=1:edof
        jj=index(j,1);
        ss(ii,jj)=ss(ii,jj)+sk(i,j);
    end
end
end%for iel
%-----
for ii=1:mm
    kk=bc dof(ii,1);
    ss(kk,1:nnode)=zeros(1,nnode);
    ss(1:nnode,kk)=zeros(nnode,1);
    ff(kk,1)=0;
end
for ii=1:mm
    kk=bc dof(ii,1);
    ss(kk,kk)=1;
end
phi=ss\ff;

for l=1:nnode
    NN(l,1)=l;
    phi_xi(l,1)=phi(l,1)-xi(l,1);
end
MAXPHI_XI=max(abs(phi_xi));
%disp('-----')
%disp('number of nodes,elements & nodes per element')
%[nnode nel nnel ndof]
%disp('element number   nodal connectivity for quadrilateral element')
%table1
%disp('-----')
%disp('element number   coordinates of the triangle spanning the quadrilateral element')
%table2
%disp('-----')
%disp('node number       Prandtl Stress Values')
%disp('       fem-computed values       anlytical(theoretical)-values       ')

%disp([NN phi xi phi_xi])
t=0;
for iii=1:nnode
    t=t+phi(iii,1)*ff(iii,1);
end
T=8*t;

%disp('-----')
%disp('number of nodes,elements & nodes per element')
%disp([nnode nel nnel ])
%disp('torisonal constants(fem=phi&exact=xi) error(max(abs(phi_xi)))')
disp('-----')
disp([nnode nel nnel ])
disp([T k1 MAXPHI_XI ])
disp('-----')

```

## PROGRAM-2

```

function[coord,gcoord]=coordinate_rtisoscelestriangle00_h0_hh_2ndorder(n)
%cartesian coordinates ((xi,yi),i=1,2,3) for the right isosceles triangle
%with vertices (x1,y1)=(0,0),(x2,y2)=(1,0) and (x3,y3)=(1,1)
syms ui vi wi xi yi
[ui,vi,wi]=coordinate_special_quadrilaterals_in_stdtriangle_2nd_order(n);
%disp([ui vi wi])

```

```

N=length(ui);
NN=(1:N)';
for i=1:N
    xi(i,1)=(ui(i,1)+vi(i,1))/2;
    yi(i,1)=vi(i,1)/2;
end
%disp('_____')
%disp('NN xi yi')
%disp(['NN xi yi'])
%disp('_____')
coord(:,1)=(xi(:,1));
coord(:,2)=(yi(:,1));
gcoord(:,1)=double(xi(:,1));
gcoord(:,2)=double(yi(:,1));
%disp(gcoord);

```

### PROGRAM-3

```

function[eln,nodetel,nodes,nnode]=nodaladdresses_special_convex_quadrilaterals_2nd_order(n)
%division of a standard triangle(right isoscles triangle)
%into eight node special_convex_quadrilaterals
for nelm=1:3*(n/2)^2
    spqd(nelm,1:8)=0;
end
%disp('vertex nodes of triangle')
elm(1,1)=1;
elm(n+1,1)=2;
elm((n+1)*(n+2)/2,1)=3;
%disp('vertex nodes of triangle')
kk=3;
for k=2:n
    kk=kk+1;
    elm(k,1)=kk;
end
%disp('left edge nodes')
nni=1;
for i=0:(n-2)
    nni=nni+(n-i)+1;
    elm(nni,1)=3*n-i;
end
%disp('right edge nodes')
nni=n+1;
for i=0:(n-2)
    nni=nni+(n-i);
    elm(nni,1)=(n+3)+i;
end

%disp('interior nodes')
nni=1;jj=0;
for i=0:(n-3)
    nni=nni+(n-i)+1;
    for j=1:(n-2-i)
        jj=jj+1;
        nnj=nni+j;
        elm(nnj,1)=3*n+jj;
    end
end
%disp(eln)
%disp(length(eln))

jj=0;kk=0;
for j=0:n-1
    jj=jj+1;
    for k=1:(n+1)-j
        kk=kk+1;
        row_nodes(jj,k)=elm(kk,1);
    end
end
end

```

```

row_nodes(n+1,1)=3;
%for jj=(n+1):-1:1
% disp(row_nodes(jj,:))
%end
[row_nodes]
rr=row_nodes;
%rr
%disp('element computations')
if rem(n,2)==0
ne=0;N=n+1;

for k=1:2:n-1
N=N-2;
i=k;
for j=1:2:N
ne=ne+1;
eln(ne,1)=rr(i,j);
eln(ne,2)=rr(i,j+2);
eln(ne,3)=rr(i+2,j);
eln(ne,4)=rr(i,j+1);
eln(ne,5)=rr(i+1,j+1);
eln(ne,6)=rr(i+1,j);
end%i
%me=ne;
%N-2
if (N-2)>0
for jj=1:2:N-2
ne=ne+1;
eln(ne,1)=rr(i+2,jj+2);
eln(ne,2)=rr(i+2,jj);
eln(ne,3)=rr(i,jj+2);
eln(ne,4)=rr(i+2,jj+1);
eln(ne,5)=rr(i+1,jj+1);
eln(ne,6)=rr(i+1,jj+2);
end%jj
end
end%k
end
%ne
%for kk=1:ne
%[eln(kk,1:6)];
%end
%add node numbers for element centroids

nnd=(n+1)*(n+2)/2;
for kkk=1:ne
nnd=nnd+1;
eln(kkk,7)=nnd;
end
%for kk=1:ne
%[eln(kk,1:7)]
%end
%to generate special quadrilaterals
mm=0;
for iel=1:ne
for jel=1:3
mm=mm+1;
switch jel
case 1
nodes(mm,1:4)=[eln(iel,7) eln(iel,6) eln(iel,1) eln(iel,4)];
nodetel(mm,1:3)=[eln(iel,2) eln(iel,3) eln(iel,1)];
case 2
nodes(mm,1:4)=[eln(iel,7) eln(iel,4) eln(iel,2) eln(iel,5)];
nodetel(mm,1:3)=[eln(iel,3) eln(iel,1) eln(iel,2)];
case 3
nodes(mm,1:4)=[eln(iel,7) eln(iel,5) eln(iel,3) eln(iel,6)];
nodetel(mm,1:3)=[eln(iel,1) eln(iel,2) eln(iel,3)];
end
end
end
end

```

```

%for mmm=1:mm
%spqd(:,1:4)
%end
%mesh generation of eight node special quadrilaterals

for inum=1:nnd
    for jnum=1:nnd
        mdpt(inum,jnum)=0;
    end
end
nd=nnd;
for mmm=1:mm
    mmm1=nodes(mmm,1);
    mmm2=nodes(mmm,2);
    mmm3=nodes(mmm,3);
    mmm4=nodes(mmm,4);
    %midpoint side-1 of 4-node special quadrilateral
    if((mdpt(mmm1,mmm2)==0)&(mdpt(mmm2,mmm1)==0))
        nd=nd+1;
        mdpt(mmm1,mmm2)=nd;
        mdpt(mmm2,mmm1)=nd;
    end
    %midpoint side-2 of 4-node special quadrilateral
    if((mdpt(mmm2,mmm3)==0)&(mdpt(mmm3,mmm2)==0))
        nd=nd+1;
        mdpt(mmm2,mmm3)=nd;
        mdpt(mmm3,mmm2)=nd;
    end
    %midpoint side-3 of 4-node special quadrilateral
    if((mdpt(mmm3,mmm4)==0)&(mdpt(mmm4,mmm3)==0))
        nd=nd+1;
        mdpt(mmm3,mmm4)=nd;
        mdpt(mmm4,mmm3)=nd;
    end
    %midpoint side-4 of 4-node special quadrilateral
    if((mdpt(mmm4,mmm1)==0)&(mdpt(mmm1,mmm4)==0))
        nd=nd+1;
        mdpt(mmm4,mmm1)=nd;
        mdpt(mmm1,mmm4)=nd;
    end
    nodes(mmm,5)=mdpt(mmm1,mmm2);
    nodes(mmm,6)=mdpt(mmm2,mmm3);
    nodes(mmm,7)=mdpt(mmm3,mmm4);
    nodes(mmm,8)=mdpt(mmm4,mmm1);
end
nnode=nd;
nel=mm;

```

## PROGRAM-4

```

function[]=quadrilateralmesh_square_cross_section_q8automeshgen(mmesh,nmesh)

clf
for mesh=mmesh:nmesh
    figure(mesh)
    ndiv=2*mesh;
    [eln,nodetel,nodes,nnode]=nodaladdresses_special_convex_quadrilaterals_2nd_order(ndiv);
    [coord,gcoord]=coordinate_rtisoscelestriangle00_h0_hh_2ndorder(ndiv);

    [nel,nnel]=size(nodes)

    for i=1:nel
        NN(i,1)=i;
    end

    table1=[NN nodes]

```

```

for elm=1:nel
nn1=nodes(elm,1);nn2=nodes(elm,2);nn3=nodes(elm,3);nn4=nodes(elm,4);
nn5=nodes(elm,5);nn6=nodes(elm,6);nn7=nodes(elm,7);nn8=nodes(elm,8);
  for j=1:2
    gcoord(nn5,j)=(gcoord(nn1,j)+gcoord(nn2,j))/2;
    gcoord(nn6,j)=(gcoord(nn2,j)+gcoord(nn3,j))/2;
    gcoord(nn7,j)=(gcoord(nn3,j)+gcoord(nn4,j))/2;
    gcoord(nn8,j)=(gcoord(nn4,j)+gcoord(nn1,j))/2;
  end
end

```

```

[nnode,dimension]=size(gcoord)
%plot the mesh for the generated data
%x and y coordinates
xcoord(1:nnode,1)=gcoord(1:nnode,1);
ycoord(1:nnode,1)=gcoord(1:nnode,2);
%extract coordinates for each element

```

```

for i=1:nel
for j=1:nnel
x(1,j)=xcoord(nodes(i,j),1);
y(1,j)=ycoord(nodes(i,j),1);
end;%j loop
xvec(1,1:5)=[x(1,1),x(1,2),x(1,3),x(1,4),x(1,1)];
yvec(1,1:5)=[y(1,1),y(1,2),y(1,3),y(1,4),y(1,1)];
%axis equal
axis tight
xmin=0;xmax=1/2;ymin=0;ymax=1/2;
axis([xmin,xmax,ymin,ymax]);
plot(xvec,yvec);%plot element
hold on;
%place element number
midx=mean(xvec(1,1:4));
midy=mean(yvec(1,1:4));
if mesh<=5
text(midx,midy,['\bf(',num2str(i),'\bf)']);
end
end;%i loop
xlabel('\bfx axis')
ylabel('\bfy axis')
st1='\bfone eighth (1/8)square cross section ';
st2=' using ';
st3='8-node parabolic ';
st4='quadriateral';
st5=' elements'
title([st1,st2,st3,st4,st5])
text(0.1,0.4,['\bfMESH NO.=',num2str(mesh)])
text(0.1,0.38,['\bfnumber of elements=',num2str(nel)])
text(0.1,0.36,['\bfnumber of nodes=',num2str(nnode)])

```

```

%put node numbers
for jj=1:nnode
if mesh<=5
text(gcoord(jj,1),gcoord(jj,2),['\bfo',num2str(jj)]);
else
text(gcoord(jj,1),gcoord(jj,2),['\bfo']);
end
end
hold on
%axis off
end%for nmesh-the number of meshes

```