www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 11 November 2017, Page No. 22869-23058
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i11.09

Finite element solution of Poisson Equation over Polygonal Domains using a novel auto mesh generation technique and an explicit integration scheme for nine node linear convex quadrilateral of Lagrange family.

H. T. Rathod[a*], Md.Shafiqul. Islam[b], H. Y. Shrivalli[c], Bharath Rathod[d] K. Suganth Devi[e]

[a]Department of Mathematics, Central College Campus, Bangalore University,
Bangalore -560001, Karnataka State, India.
[b]Department of Applied Mathematics, University of Dhaka,Dhaka-1000,
Bangladesh
[c]Department of Mathematics,
B.M.S college of Engineering,
Basavanagudi, Bangalore -560019,
India
[d]Xavier Institute of Management and Entrepreneurship, Hosur Road,
Electronic City Phase II, Bangalore, Karnataka 560034, Karnataka State, India.
[e]Department of Mathematics, Dr. T. Thimmaiah Institute of Technology, Oorgam Post,
Kolar Gold Field, Kolar District, Karnataka state, Pin- 563120, India.

Abstract :

This paper presents an explicit integration scheme to compute the stiffness matrix of a nine node linear convex quadrilateral element of Lagrange family using symbolic mathematics and discretisation of polygonal domain by such finite elements using a novel auto mesh generation technique , In finite element analysis, the boundary value 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 the bivariates ξ and η over a nine node 2-square $(-1\leq \xi,\eta \leq 1)$ with nodes at the points {(-1,-1),(1,-1),(1,1),(-1,1),(0,-1),(1,0).(0,1),(-1,0),(0,0)} in local parametric space. In this paper, we have computed these integrals in exact forms using the symbolic mathematics capabilities of MATLAB. The proposed explicit finite element integration scheme can be applied to solve boundary value problems in continuum mechanics over convex polygonal domains. We have also developed a novel auto mesh generation technique of all nine node linear convex quadrilaterals for a polygonal domain $\Omega \subset \mathcal{R}^2$ which provides the nodal coordinates and element connectivity. We have used the explicit integration scheme and this novel auto mesh generation technique to solve the Poisson equation $-\nabla^2 u= f$,where u is unknown physical variable and f *is a known smooth function* in $\Omega \subset \mathcal{R}^2$ with given Dirichlet boundary conditions over the given convex polygonal domain.

Key words: Explicit Integration, Finite Element Method,Quadratic 2-D finite element of Lagrange family, Matlab Symbolic Mathematics, All Quadrilateral Mesh Generation Technique,Poisson Equation,Dirichlet Boundary Conditions ,Polygonal Domain, Gauss Legendre Quadrature Rules

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 problem in science, engineering and emerging technologies such as nanotechnology, biotechnology, aerospace, chemical. etc would be incapable of solution [1,2,3]. 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 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, Gauss Legendre quadrature, which can evaluate exactly the (2n-1)th degree 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 [7-15]. The integration points have to be increased in order 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 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 equivalently the subparametric finite 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 [16] for the four node 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 four node 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 (ξ,η) 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.

Many important problems in engineering,science and applied mathematics are formulated by appropriate differential equations with some boundary conditions imposed on the desired unknown function or the set of functions. There exists a large literature which demonstrates numerical accuracy of the finite element method to deal with such issues [1]. Clough seems to be the first who introduced the fnite elements to standard computational procedures [2]. A further historical development and present day concepts of finite element analysis are widely described in references [1, 3]. In this paper the well-known Laplace and Poisson equations will be examined by means of the finite element method applied to an appropriate 'mesh'. The class of physical situations in which we meet these equations is really broad. Let's recall such problems like heat conduction, seepage through porous media, irrotational flow of ideal fluids, distribution of electrical or magnetic potential, torsion of prismatic shafts, lubrication of pad bearings and others [4]. Therefore, in physics and engineering arises a need of some computational methods that allow to solve accurately such a large variety of physical situations.The considered method completes the above-mentioned task. Particularly, it refers to a standard discrete pattern allowing to find an approximate solution to continuum problem. At the beginning, the continuum domain is discretized by dividing it into a finite number of elements which properties must be determined from an analysis of the physical problem (e. g. as a result of experiments). These studies on particular problem allow to construct so{called the stiffness matrix for each element that, for instance, in elasticity comprising material properties like stress strain relationships [2, 5]. Then the corresponding nodal loads associated with elements must be found. The construction of accurate elements constitutes the subject of a mesh generation recipe proposed by the author within the presented article. In many realistic situations, mesh generation is a time consuming and error prone process because of various levels of geometrical complexity. Over the years, there were developed both semi automatic and fully automatic mesh generators obtained, respectively, by using the mapping methods or, on the contrary, algorithms based on the Delaunay triangulation method [6], the advancing front method [7] and tree methods [8]. It is worth mentioning that the first attempt to create fully automatic mesh generator capable to produce valid finite element meshes over arbitrary domains has been made by Zienkiewicz and Phillips [9].

In the present paper, we propose a similar discretisation method for linear polygon in Cartesian two space (x,y). This discrtisation 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 a standard triangle in a local space (u,v). 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 (ξ,η) 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 quadrilaterals in the local space (u,v) interior to the standard triangle. In this procedure, 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. We have obtained explicit integration of the product of local derivatives which is now possible by use of symbolic integration commands available in leading mathematical softwares MATLAB, MAPLE, MATHEMATIKA etc. In this paper, we have used the MATLAB symbolic mathematics to compute the integrals of the products of local derivatives in (u, v) space .The proposed explicit integration scheme is shown as a useful technique in the formation of element stiffness matrices for second order boundary problems governed by partial differential equations.

This paper presents an explicit integration scheme to compute the stiffness matrix of a nine node linear convex quadrilateral element of Lagrange family using symbolic mathematics and discretisation of polygonal domain by such finite elements using a novel auto mesh generation technique , In finite element analysis, the boundary value 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 the bivariates ξ and η over a nine node 2-square ($-1\leq \xi,\eta \leq 1$) with nodes at the points {(-1,-1),(1,-1),(1,1),(-1,1),(0,-1),(1,0).(0,1),(-1,0),(0,0)}. In this paper, we have computed these integrals in exact forms using the symbolic mathematics capabilities of MATLAB. The proposed explicit finite element integration scheme can be applied to solve boundary value problems in continuum mechanics over convex polygonal domains. We have also developed a novel auto mesh generation technique of all nine node linear convex quadrilaterals for a polygonal domain $\Omega \subset \mathcal{R}^2$ which provides the nodal coordinates and element connectivity. We have used the explicit integration scheme and this novel auto mesh generation technique to solve the Poisson equation $-\nabla^2 u= f$,where u is unknown physical variable and f *is a known smooth function* in $\Omega \subset \mathcal{R}^2$with given Dirichlet boundary conditions over convex polygonal domains.We need a small amount of numerical integration to complete the solution of the Poisson boundary value problem when f *is a known smooth function other than a constant.*

2. POISSON EQUATION

2.1 Statement of the Problem

The Poisson equation

$$-\nabla^2 u = f$$

....(1) is the simplest and most famous elliptic partial differential equations.The source (or load) function is given on some two or three dimensional domain $\Omega \subset \mathcal{R}^2$ or $\mathcal{R}^3$. A solution u satisfying (1.1) will also satisfy boundary conditions on the boundary $\partial\Omega$ of Ω ;for example

$$\alpha u + \beta \frac{\partial u}{\partial n} = g \quad \text{on} \quad \partial\Omega$$

....(2)

where $\partial u / \partial n$ denotes directional derivative in the direction normal to the boundary $\partial\Omega$ (conveniently pointing outwards) and α and β are constants, although variable coefficients are also possible.The combination of (1.1) and (1.2) together is referred to as boundary value problem. If the constant β in (1.2) is zero,then the boundary condition is known as the Dirichlet type, and the boundary value problem is referred as the Dirichlet problem for the Poisson equation. Alternatively, if the constant α in (1.2) is zero,then we correspondingly have a Neumann boundary value problem. A third possibility is that Dirichlet conditions hold on part of the boundary $\partial\Omega_D$ and Neumann conditions(or indeed mixed conditions where α and β are both nonzero) hold on remainder $\partial\Omega \backslash \partial\Omega_D$. The case $\alpha = 0, \beta = 1$ in (1.2) demands special attention.First, since u=constant satisfies the homogeneous problem with $f = 0, g = 0$,it is clear that a solution to a Neumann problem can only be unique up to an additive constant.Second,integrating (1.1) over Ω using Gauss's theorem gives

$$-\int_{\partial\Omega} \frac{\partial u}{\partial n} = --\int_{\Omega} \nabla^2 u = \int_{\Omega} f$$

....(3)

thus a necessary condition for the existence of a solution to the Neumann problem is that the source and boundary data satisfy the compatibility condition:

$$\int_{\partial\Omega} g + \int_{\Omega} f = 0$$

---(4)

2.2 Weak Formulation of the Poisson Boundary Value Problem

A sufficiently smooth function u satisfying both eqns(1) and (2) is known as classical solution to the Poisson boundary value problem. For a Dirichlet problem, u is a classical solution only if it has continuousecond derivatives in Ω (i.e. u is $C^2(\Omega)$ and is continuous up to the boundary i.e.u is in $C^0(\overline{\Omega})$). In case of nonsmooth domains or discontinuous source functions,the function u satisfying eqns(1) and (2) may not be smooth (or regular) enough to be regarded as classical solution. For problems which arise from , perfectly reasonable mathematical models an alternative description of the boundary value problem is required. Since this alternative description is less restrictive in terms of admissible data it is called weak formulation.

To derive a weak formulation of a Poisson problem,we require that for an appropriate set of test functions v,

$$\int_{\Omega} (\nabla^2 u + f)\, v = 0 \quad .$$

....(5)

This formulation exists provided that the integrals are well defined. If u is a classical solution then it must also satisfy eqn (5). If v is sufficiently smooth however,then the smoothness required of u can be reduced by using the derivative of a product rule and the divergence theorem

$$-\int_{\Omega} v\nabla^2 u = \int_{\Omega} \nabla u \,.\nabla v - \int_{\Omega} \nabla.(v\nabla u)$$

$$= \int_{\Omega} \nabla u \,.\nabla v - \int_{\partial\Omega} v \frac{\partial u}{\partial n},$$

so that

$$\int_{\Omega} \nabla u \,.\nabla v = \int_{\Omega} vf + \int_{\partial\Omega} v \frac{\partial u}{\partial n}$$

....(6a)

The point here is that the problem posed byeqn(6) may have a solution u called a weak solution,that is not smooth enough to be a classical solution. If a classical solution does exist then eqn(6)is equivalent to eqns (1) and (2) and the weak solution is classical. The case of Neumann problem ($\alpha = 0$ $, \beta = 1$) in eqn(2) is particularly straight forward. Substituting from eqn(2) into eqn(6)gives us the following formulation: find u defined on Ω such that

$$\int_{\Omega} \nabla u \,.\nabla v = \int_{\Omega} vf + \int_{\partial\Omega} vg$$

....(6b)

for all suitable test functions v .

2.3 Finite Elements for Poisson's Equation with Dirichlet conditions: Implementation and Review Of Theory

2.3.1 Weak Form

Given Poisson Equation:

$-\Delta u$(x)=f(x) for all x∈ Ω
............................(7a)

$u = g(x)$ on $\partial\Omega$
...............................(7b)

We have already obtained in eqn(6) with ($\alpha = 1$,$\beta = 0$) the weak form of the equation by multiplying both sides by a test function v (i.e a function which is infinitely differentiable and has compact support,integrating over the domain Ω and performing integration by parts or by application of Divergence(GREEN) theorem. The result is

$\int_\Omega \nabla u \,.\nabla v \, d\, x = \int_\Omega vf \, dx$
..............................(7c)

$u = g(x)$ on $\partial\Omega$
...............................(7d)

For all test functions v .

2.3.2 Finite Elements

To find an approximation to the solution u , we choose a finite dimensional space V_h and ask that eqn(7a-b) is satisfied only for $v \; in \; V_h$ rather than for all test functions v. Then we look for a function $u_h \in V_h$ which satisfies

$\int_\Omega \nabla u_h \,.\nabla v \, d\, x = \int_\Omega vf \, dx$ for all $v \in V_h$ ....(8)

u_h is called the finite element solution and functions in V_h are called finite elements.

Note that it is also common for the triangles or quadrilaterals in the mesh to be called elements.

If a basis for V_h is $\{\varphi_j\}_{j=1}^{j=N}$ then we can write $u_h = \sum_{\mathrm{j=1}}^{\mathrm{j=N}} \alpha_\mathrm{j}\varphi_\mathrm{j}$. Substituting this in eqn(8) and choosing v to be a basis function φ_i gives the following set of equations

$\sum_{j=1}^{N} \alpha_j \int_\Omega \nabla\varphi_{\,i} \,.\nabla\varphi_j \, d\, x = \int_\Omega f\varphi_i \, dx$,i=1,2,3,...., N ....(9)

This is really a linear system of the form

$Ku{=}f$ ....(10)

Where, $u = (\alpha_1 \,, \alpha_2 \,, \alpha_3 \,, \ldots \ldots \ldots \alpha_N)^T$ and

$K_{i,j} = \int_\Omega \nabla\varphi_{\,i} \,.\nabla\varphi_j \, d\, x$ ....(11a)

$f_i = \int_\Omega f\varphi_i \, dx$ ....(11b)

and K is called stiffness matrix because the linear system looks like Hookes law if f represents forces and u represents displacemewnts.

In general,$\Omega = \sum_{e=1}^{N_e} \Omega^e$, where N_e is the number of elements discritised in the domain Ω. In two dimensions the mesh elements are triangles or quadrilaterals.The choice of finite element spaces are usually piecewise polynomials.

2.3.3 Overview on the implementation of Finite Element Method

Once we have choosen the finite element space (and the element type),then we can implement the finite element method.The implementation is divided into three steps:

1. Mesh Generation:how does one perform a triangulation or quadrangulation of the domain Ω ?
2. Assembling the Stiffness Matrix:how does one compute the entries in the stiffness matrix in an efficient way?
3. Solving the linear System:What kind of methodse suited for solving the linear system?

In this paper,we present new approach to mesh generation [ ] and explicit computations for the entries in the stiffness matrix [ ] which is vital in Assembling the Stiffness Matrix,since we believe that the methods of solving linear system are well researched and standardised.

We shall first take up the derivations regarding the topic on Assembling the Stiffness Matrix. The Mesh Generation topic will be discussed immediately there after.

2.3.4 Assembling the Stiffness Matrix

In order to assemble the stiffness matrix,we need to compute integrals of the form(see eqn(11) in section (2.3.2)

$K_{i,j} = \int_\Omega \nabla\varphi_{\,i} \,.\nabla\varphi_j \, d\, x$
....(11a)

The most obvious way to assemble the stiffness matrix is to compute the integrals $K_{i,j}$ for the nodal pairs i and j ; this is a node oriented computation and we need to know the common support of basis functions $\varphi_{\,i} \; and \; \varphi_j$.This means we need to know which elements contain both i and j.The mesh generator provides us with the information regarding the nodes on a particular element so we would need to do some extra processing to find the elements that contain a particular node.This is an issue which is very complicated.Hence,in practice assembling is focussed on elements rather than on nodes.We note that on a particular element,the basis functions have a simple expression and the elements themselves are very simple domains like triangles and quadrilaterals. It is very easy to make a change of variables for integrals over triangles and quadrilaterals to standard triangles and squares. In the element oriented computation,we rewrite or interpret the integral in eqn(11) as

$K_{i,j} = \sum_{\Omega^e \;\varepsilon \Omega_h^e} \quad K_{i,j}^e \quad =$
....(12a)

where

$K_{i,j}^{e}$ $=\int_{\Omega^e} \nabla\varphi_{\,i}\,.\nabla\varphi_j\,d\,x$ ....(12b)

and Ω_h^e is the set of (mesh) elements in Ω contributing to $K_{i,j}$ and $\Omega=\sum_{e=1}^{N_e}\Omega^e$, $\Omega^e$is an element contained in the set Ω_h^e .This says us that we can compute $K_{i,j}$ by computing the integrals over each element Ω^e and then summing up over all elements Ω_h^e .

Notice that the integrals

$K_{i,j}^e=\int_{\Omega^e}\nabla\varphi_{\,i}\,.\nabla\varphi_j\,d\,$x look like the entries $K_{i,j}=\int_{\Omega}\nabla\varphi_{\,i}\,.\nabla\varphi_j\,d\,x=\sum_{\Omega^e\ \varepsilon\Omega_h^e}\ \int_{\Omega^e}\nabla\varphi_{\,i}\,.\nabla\varphi_j\,d\,x$ except the domain of integration is an element Ω^e. So,we only need to save all entries of $K^e=[K_{i,j}^e\ ]$ which corresponds to nodes on Ω^e. Then if Ω^e has d nodes , we can think of K^e as a dxd matrix. In view of the above,the procedure for computing the stiffness maerix is done on an element by element basis.

We must also compute the integrals

$f_i=\int_{\Omega} f\varphi_i\,dx=\sum_{e=1}^{N_e}f_i^e$ ....(12c)

where

f_i^e $=$ $\int_{\Omega^e} f\varphi_i d\,$x ....(12d)

Now further assume that on an element Ω^{e} , $u_h=\mathrm{u^e}=\sum_{\mathrm{j=1}}^{\mathrm{j=d}}\mathrm{u_j^e}\varphi_{\mathrm{j}}$

From eqn(9) and eqns(12a-d) it follows that $Ku=f$ is equivalent to

$\sum_{e=1}^{N_e}K^e\,\mathrm{u^e}=\sum_{e=1}^{N_e}f^{\,e}$ ....(12e)

Where

$\mathrm{u^e}\ =(\mathrm{u_1^e}\ ,\mathrm{u_2^e},\ \mathrm{u_3^e},$ .... $\mathrm{u_d^e}\ )^T$, $f^{\,\mathrm{e}}\ =(f_1^{\mathrm{e}}\,,f_2^{\mathrm{e}},\ f_3^{\mathrm{e}},$ .... $f_{\mathrm{d}}^{\mathrm{e}}\ )^T$ ....(12f)

d referes to number of nodes per element, N_e referes to the total number of elements in the domain Ω

2.3.5 Computing the Integrals $K_{i,j}^e$ and f_i^e

In order to compute the local/element stiffness matrices,we need to compute the integrals $K_{i,j}^e=\int_{\Omega^e}\nabla\varphi_{\,i}\,.\nabla\varphi_j\,d\,$x .These integrals are computed by making a change of variables to a reference element. We now outline a brief procedure for element oriented computation

(1)For each element Ω^e , compute it's local stiffness matrix K^e. This requires computing the integrals $K_{i,j}^e=\int_{\Omega^e}\nabla\varphi_{\,i}\,.\nabla\varphi_j\,d\,$x which we compute by transforming to a reference element. In two dimensions Ω^e is an arbitrary linear triangle and each triangle will be further discritised three convex quadrilaterals Q_{3e-2} , Q_{3e-1} and Q_{3e} Each triangle will be transformed to the corresponding reference elements:the standard triangle(a right isosceles triangle) and further Each triangle will be transformed to the corresponding reference elements:the standard triangle(a right isosceles triangle) and further Each triangle will be transformed to the corresponding reference elements:the standard triangle(a right isosceles triangle) and further each quadrilateral will be transformed into a standard square(1-square or a 2-square).Since in two dimensional space x= (x,y) the explicit form of $K_{i,j}^e=\int_{\Omega^e}\nabla\varphi_{\,i}\,.\nabla\varphi_j\,d\,$x is given by

$K_{i,j}^e=\int_{\Omega^e}\nabla\varphi_{\,i}\,.\nabla\varphi_j\,$dx $=\int_{\Omega^e}\{\frac{\partial\varphi_i}{\partial x}\frac{\partial\varphi_j}{\partial x}+\frac{\partial\varphi_i}{\partial y}\frac{\partial\varphi_j}{\partial y}\}\,dxdy=\sum_{e=1}^{N_e}\sum_{n=0}^{2}\int_{Q_E}\{\frac{\partial\varphi_i}{\partial x}\frac{\partial\varphi_j}{\partial x}+\frac{\partial\varphi_i}{\partial y}\frac{\partial\varphi_j}{\partial y}\}dxdy$ $=\sum_{e=1}^{N_e}\sum_{n=0}^{2}S_{i,j}^E$ ....(12g)

Where $S_{i,j}^E=\int_{Q_E}\{\frac{\partial\varphi_i}{\partial x}\frac{\partial\varphi_j}{\partial x}+\frac{\partial\varphi_i}{\partial y}\frac{\partial\varphi_j}{\partial y}\}dxdy$ and E=3e+n-2,e=1,2,... N_e andn=0,1,2

and hence we must be careful about the derivatives when we perform the change of variables. These bring extra factors involving the affine transformations (when Ω^e is an arbitrary linear triangle) and bilinear transformations(when Ω^e is an arbitrary linear convex quadrilateral)

f_i^e = $\int_{\Omega^e} f\varphi_i$ dxdy can be computed in a straight forward manner if f is a simple function otherwise we have to apply numerical integration

(2) For each element Ω^e, first compute the local stiffness matrices $S^E=[S_{i,j}^E\ ]$ and then add contribution of $K^e=S^{3e-2}$ $+S^{3e-1}+S^{3e}$, to the global stiffness matrix K. We repeat this procedure for all elements i.e for e=1,2,......, N_e; where N_e is the number of elements Ω^e $which$ are discritised in the domain Ω ,in fact we have $\Omega=\sum_{e=1}^{N_e}\Omega^e$ $=\sum_{e=1}^{N_e}\sum_{n=0}^{2}Q_E$, E=3e+n-2

2.4 Finite Element Types

2.4.1 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 (ξ, η) as in Fig 1b.

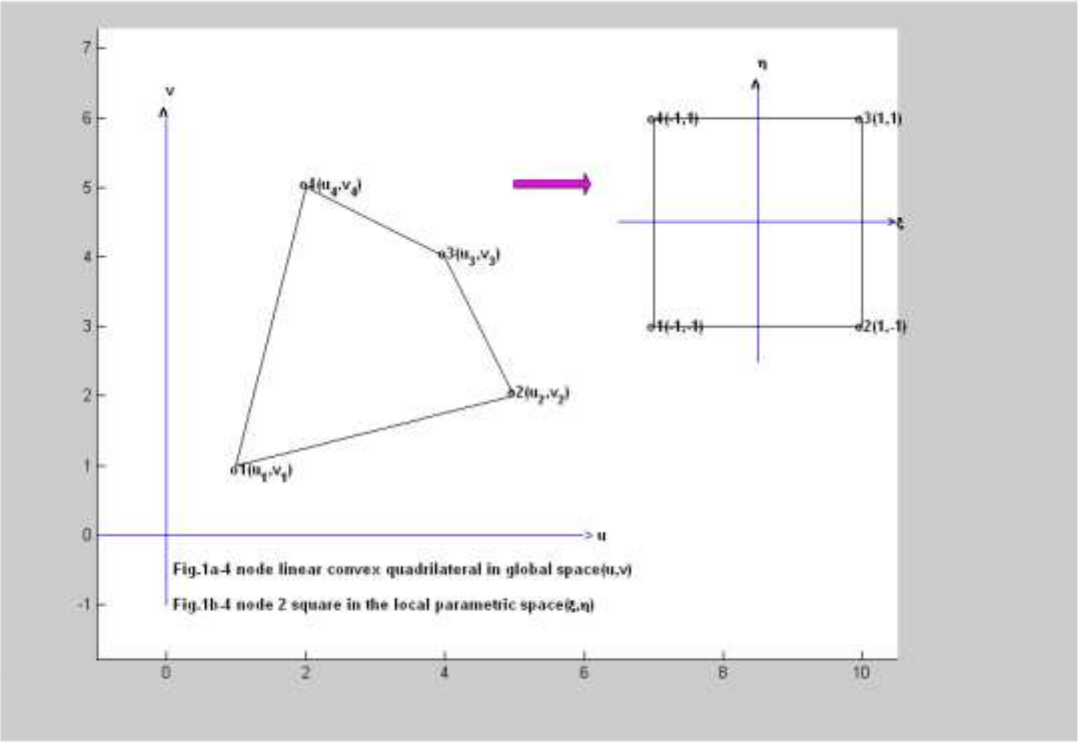


Fig.1a-4 node linear convex quadrilateral in global space(u,v)

Fig.1b-4 node 2 square in the local parametric space(ξ,η)

$$\begin{pmatrix} u \\ v \end{pmatrix} = \sum_{k=1}^{4} \begin{pmatrix} u_k \\ v_k \end{pmatrix} \ M_k(\xi, \eta) \qquad \text{------------------------ (13)}$$

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 (ξ, η) and they are given by

$$M_k(\xi, \eta) = \frac{1}{4}\ (1 + \xi\xi_k)(1 + \eta\eta_k)\ , \quad k = 1, 2,3,4 \qquad \text{----------------------- (14a)}$$

Where $\{\ (\xi_k, \eta_k), k = 1,2,3,4\}$ = {(-1,-1),(1,-1),(1,1),(-1,1)} ----------------------- (14b)

describes a geometric transformation over a linear convex quadrilateral element from the original global space into the local parametric space.

2.4.2 Isoparametric Transformation :

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

$$\begin{pmatrix} \phi \\ \psi \end{pmatrix} = \sum_{k=1}^{4} \begin{pmatrix} \phi_k \\ \psi_k \end{pmatrix} {N_k}^e(\xi, \eta) \qquad \text{------------------------ (15)}$$

Where ϕ_k , ψ_k refer to unknowns at node k and the shape functions ${N_k}^e = M_k$, and M_k are defined as in Eqn.(14a-b)

We have considered the application of explicit stiffness matrix integration scheme and automesh generation technique to find FEM solution of Poisson equation boundary value problems over polygonal domains using linear convex quadrilateral elements under isoparametric transformations[ ].

2.4.3 Subparmetric Transformation :

For the subparametric transformation over the nde – noded element we define the field variables ϕ, ψ (say) governing the physical problem as

$$\begin{pmatrix} \phi \\ \psi \end{pmatrix} = \sum_{k=1}^{nde} \begin{pmatrix} {\phi_k}^e \\ {\psi_k}^e \end{pmatrix} {N_k}^e(\xi, \eta) \qquad \text{------------------------ (16)}$$

Where ϕ_k , ψ_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 for which we set nde=4

In the present paper, we consider the subparametric transformation for a linear convex quadrilateral element for which nde = 9 , a nine noded 2 square of Lagrange type

3. Nine Node Linear Convex Quadrilateral Element :

In this section, we give a brief description of the 9- node quadrilateral element under subparametric transformation as shown in Fig 1c , Fig 1d.

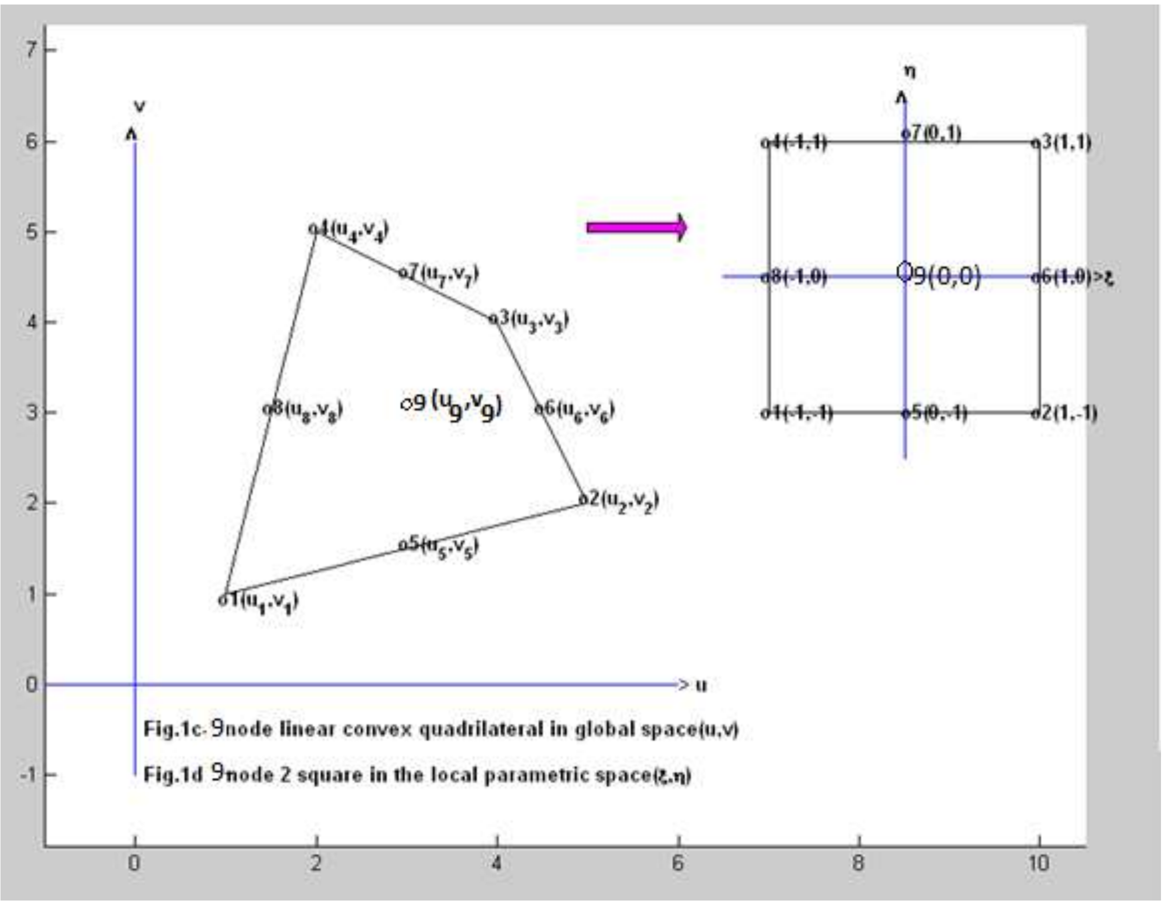


Fig.1c- 9node linear convex quadrilateral in global space(u,v)

Fig.1d 9node 2 square in the local parametric space(ξ,η)

We use the transform of Eqns.(13-14) to define the element geometry i.e.

$\begin{pmatrix} u(\xi,\eta) \\ v(\xi,\eta) \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} = \sum_{k=1}^{4} \begin{pmatrix} u_k \\ v_k \end{pmatrix} M_k(\xi,\eta)$ ------------------------- (13)

Where $M_k(\xi,\eta) = \frac{1}{4}\,(1+\xi\xi_k)(1+\eta\eta_k)$, $(k = 1,2,3,4\,)$ ----------------------- (14a)

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

$\{\, (\xi_k,\eta_k), k = 1,2,3,4\} = \{(-1,-1),(1,-1),(1,1),(-1,1)\}$ ----------------------- (14b)

Using the transformation of Eqns.(13-14) and from Fig 1c , Fig 1d we see that there is a one to one correspondence between $((\xi_k,\eta_k),\ k = 5,6,7,8,9) = ((0,1),(1,0),(0,1),(-1,0),(0,0))$

and $((u_k,v_k) = \big(u(\xi_k\,,\eta_k), v(\,\xi_k\,,\eta_k)\big)$, $k = 5,6,7,8,9)$, where

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

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

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

$(u_8, v_8) = ((u_1 + u_4)/2 \ , (v_1 + v_4)/2 \ )$

$(u_9, v_9) = ((u_1 + u_2 + u_3 + u_4)/4 \ , (v_1 + v_2 + v_3 + v_4)/4 \ )$ -------------------------(14c)

We then define the variation of physical variables ϕ^e, ψ^e (say) over 9- node element of Fig 1c , 1d by Eqn.(16) with nde = 9

$$\begin{pmatrix} \phi^e \\ \psi^e \end{pmatrix} = \sum_{k=1}^{8} N_k{}^e(\xi,\eta) \begin{pmatrix} \phi_k{}^e \\ \psi_k{}^e \end{pmatrix} \qquad \text{------------------------ (16)}$$

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

The shape functions $N_i{}^e$ of the 9- node element shown in Fig 1c , Fig 1d are given by

$N_1{}^e(\xi,\eta) = (\xi^2 - \xi)(\eta^2 - \eta)/4$
$N_2{}^e(\xi,\eta) = (\xi^2 + \xi)(\eta^2 - \eta)/4$
$N_3{}^e(\xi,\eta) = (\xi^2 + \xi)(\eta^2 + \eta)/4$
$N_4{}^e(\xi,\eta) = (\xi^2 - \xi)(\eta^2 + \eta)/4$
$N_5{}^e(\xi,\eta) = (1 - \xi^2)(\eta^2 - \eta)/2$
$N_6{}^e(\xi,\eta) = (\xi^2 + \xi)(1 - \eta^2)/2$
$N_7{}^e(\xi,\eta) = (1 - \xi^2)(\eta^2 + \eta)/2$
$N_8{}^e(\xi,\eta) = (\xi^2 - \xi)(1 - \eta^2)/2$
$N_9{}^e(\xi,\eta) = (1 - \xi^2)(1 - \eta^2)$
-------------------------(17a)

and we may check that

$N_k{}^e(\xi_k,\eta_k)=1$, $N_k{}^e(\xi_j,\eta_j)=0$,when $j \neq k$

$\{ (\xi_k,\eta_k), k = 1(1)9\} = \{(-1,-1), (1,-1), (1,1), (-1,1), (0,-1), (1,0), (0,1), (-1,0), (0,0)\}$ ----------------------(17b)

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 (ξ,η) as in Fig 1b, d

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

$$\frac{\partial u}{\partial \xi} = \sum_{k=1}^{4} u_k \frac{\partial M_k}{\partial \xi} = \frac{1}{4}[(-u_1 + u_2 + u_3 - u_4) + (u_1 - u_2 + u_3 - u_4)\,\eta] \qquad \text{------------ (18a)}$$

$$\frac{\partial u}{\partial \eta} = \sum_{k=1}^{4} u_k \frac{\partial M_k}{\partial \eta} = \frac{1}{4}[(-u_1 - u_2 + u_3 + u_4) + (u_1 - u_2 + u_3 - u_4)\,\xi] \qquad \text{------------ (18b)}$$

$$\frac{\partial v}{\partial \xi} = \frac{1}{4}[(-v_1 + v_2 + v_3 - v_4) + (v_1 - v_2 + v_3 - v_4)\,\eta] \qquad \text{------------ (18c)}$$

$$\frac{\partial v}{\partial \eta} = \frac{1}{4}[(-v_1 - v_2 + v_3 + v_4) + (v_1 - v_2 + v_3 - v_4)\,\xi] \qquad \text{------------ (18d)}$$

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

$$J = \frac{\partial(u,v)}{\partial(\xi,\eta)} = \frac{\partial u}{\partial \xi}\frac{\partial v}{\partial \eta} - \frac{\partial u}{\partial \eta}\frac{\partial v}{\partial \xi} = \alpha + \beta\xi + \gamma\,\eta \qquad \text{-------------- (19a)}$$

Where

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

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

$\gamma = \frac{1}{8}[\,(u_4 - u_1)(v_2 - v_3) + (u_3 - u_2)(v_4 - v_1)\,]$ ---------------- (19b)

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]

$$\begin{pmatrix} \frac{\partial N_i^e}{\partial u} \\ \frac{\partial N_i^e}{\partial v} \end{pmatrix} = \frac{1}{J}\begin{bmatrix} \frac{\partial v}{\partial \eta} & -\frac{\partial v}{\partial \xi} \\ -\frac{\partial u}{\partial \eta} & \frac{\partial u}{\partial \xi} \end{bmatrix}\begin{bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \end{bmatrix} \qquad \text{---------------------------- (20)}$$

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.(18a)–(18d) while J is defined in Eq.(19a-b) , ($i, j = 1,2,3, -----,$ 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 Δ 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 Δ PQR. We can obtain three linear convex 9- node quadrilaterals (Q_e, e=1,2,3),where Q_1 =ZcUdPeS , Q_2 =ZaSfQgT and Q_3= ZbThRiU from triangle Δ PQR as shown in Fig 2a,b.Further each of these Q1,Q2 and Q3 have centrodal points j,k and l. If we map each of these 9- node linear convex quadrilaterals into 9- 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

$J = J^e = \frac{1}{48}\Delta\, pqr\, (4 + \xi + \eta), \qquad e = 1,2,3$ --------------- (21)

Where Δ pqr is the area of the triangle Δ PQR

$$2\Delta\, pqr = \begin{vmatrix} 1 & x_p & y_p \\ 1 & x_q & y_q \\ 1 & x_r & y_r \end{vmatrix} = [\,(x_p - x_r)(y_q - y_r) - (x_q - x_r)(y_p - y_r)\,] \qquad \text{------------ (22)}$$

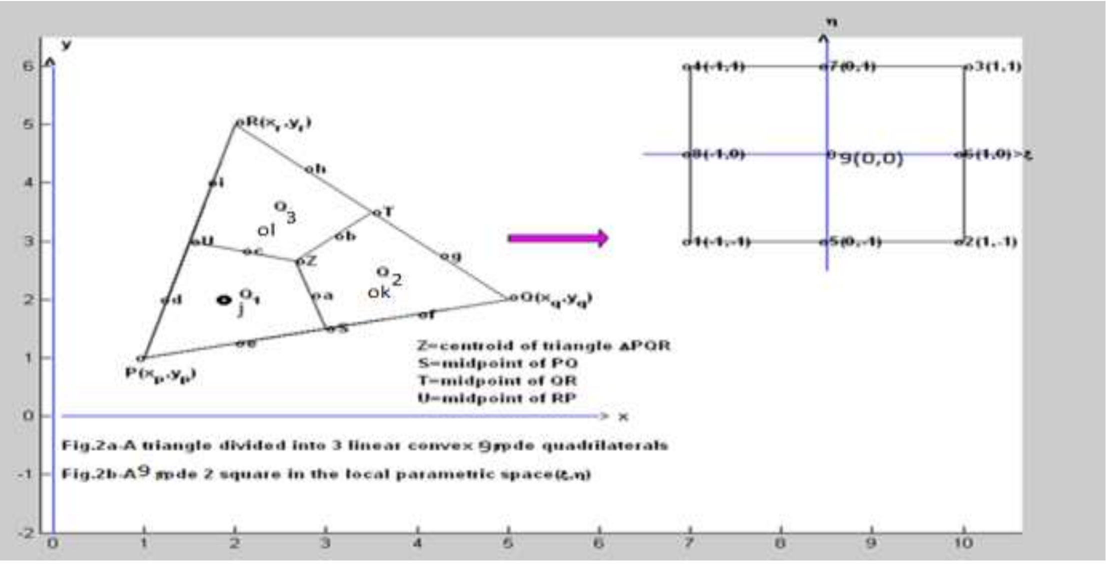


Fig.2a-A triangle divided into 3 linear convex 9 node quadrilaterals

Fig.2b-A 9 node 2 square in the local parametric space(ξ,η)

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

Lemma 2. Let Δ 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 Δ PQR . We further let j,k,l be the centre point of the quadrilaterals Q_e (e=1,2,3), Then we obtain three linear convex 9- node quadrilaterals Q_e (e=1,2,3),Q_1 =<ZcUdPeSaj> , Q_2 =<ZaSfQgTbk > and Q_3= <ZbThRiUcl> , these quadrilaterals can mapped into the linear conv ex 9- node quadrilateral spanning the vertices GHEICJF with G(1/3, 1/3), H(1/6, 5/12), E(0, ½), I(0, ¼), C(0, 0), J(1/4, 0), F(1/2, 0), K(5/12, 1/6) in the interior of the right isosceles triangle Δ 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 = Δ PQR as shown in Fig 2a & Fig 3a.

We know that the linear transformations

$$\begin{pmatrix} x^{(1)} \\ y^{(1)} \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} w + \begin{pmatrix} x_q \\ y_q \end{pmatrix} u + \begin{pmatrix} x_r \\ y_r \end{pmatrix} v \qquad \text{-------------------------} \ (23)$$

$$\begin{pmatrix} x^{(2)} \\ y^{(2)} \end{pmatrix} = \begin{pmatrix} x_q \\ y_q \end{pmatrix} w + \begin{pmatrix} x_r \\ y_r \end{pmatrix} u + \begin{pmatrix} x_p \\ y_p \end{pmatrix} v \qquad \text{--------------------------} \ (24)$$

$$\begin{pmatrix} x^{(3)} \\ y^{(3)} \end{pmatrix} = \begin{pmatrix} x_r \\ y_r \end{pmatrix} w + \begin{pmatrix} x_p \\ y_p \end{pmatrix} u + \begin{pmatrix} x_q \\ y_q \end{pmatrix} v \qquad \text{------------------------} \ (25)$$

with $w = 1 - u - v$

map the arbitrary triangle Δ 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 palne is mapped into the linear convex 9- node quadrilateral spanning the vertices G, H, E, I, C, J, F, K,L by use of the transformation given in Eqn.(23).

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 ,Lby use of the transformation of Eqn.(24). Finally the quadrilateral Q_3 in xy plane is mapped into the quadrilateral GHEICJFKL in uv- plane by use of the linear transformation of Eqn.(25),
This completes the proof.

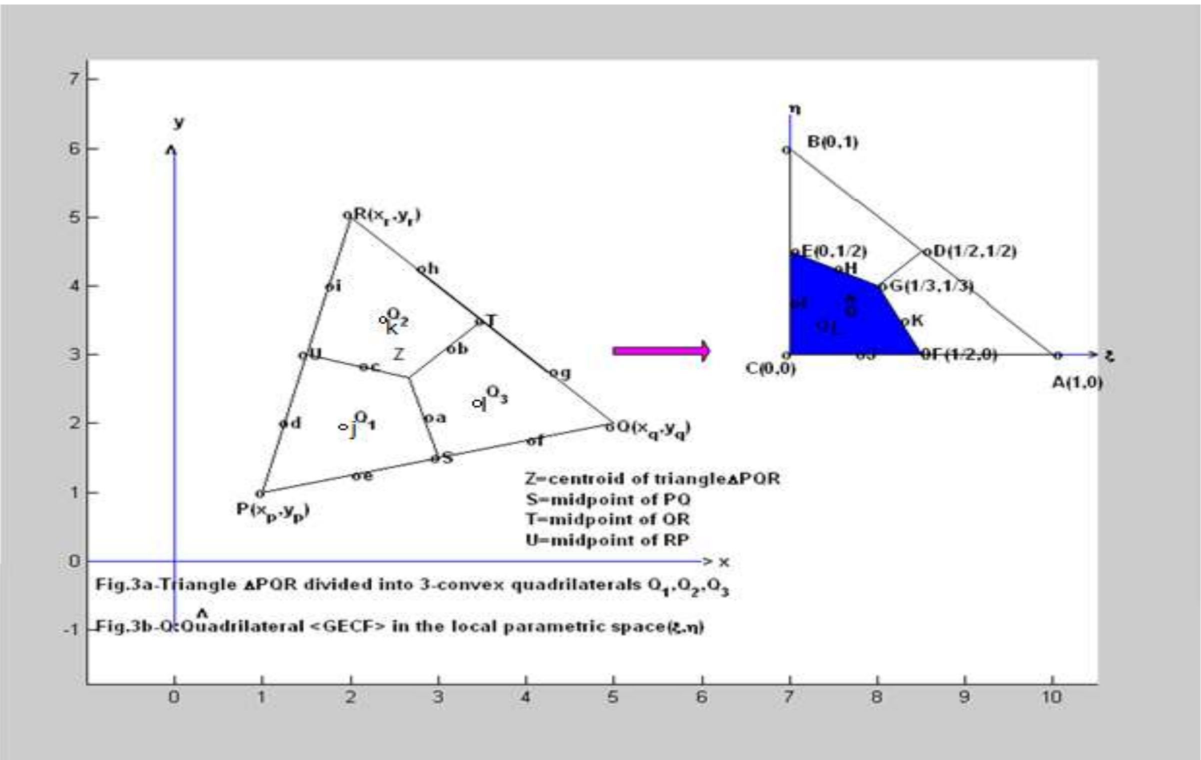


Fig.3a-Triangle ΔPQR divided into 3-convex quadrilaterals Q_1,Q_2,Q_3

Fig.3b-$\hat{Q}$:Quadrilateral <GECF> in the local parametric space(ξ,η)

We have shown in the foregoing Lemma that an arbitrary linear triangle can be discretised into three linear convex 9- node quadrilaterals. Further, each of these quadrilaterals in xy plane can be mapped into a unique linear convex 9- node quadrilateral spanned by the vertices G(1/3, 1/3), H(1/6, 5/12), E(0, ½), I(0, ¼), C(0, 0), J(1/4, 0), F(1/2, 0) , K(5/12, 1/6) and L(5/24,5/24) (see Fig 3a, Fig 3b) using a proper linear transformation as given Eqn.(23) – (25).

6. Integration over a Triangular Region :

6.1 Composite Integration

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

$II_{\Delta PQR} = \iint_{\Delta PQR} \phi(x,y)\, dxdy = \sum_{e=1}^{3} \iint_{Q_e} \phi(x,y)\, dxdy$ --------------------- (26)

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

$= (2\,\Delta_{pqr})\ \iint_{\hat{Q}} \{ \sum_{e=1}^{3} [\phi\,(x^{(e)}(u,v),\ \ y^{(e)}(u,v))\,]\}\ dudv$ -------------------- (27)

Where $(x^{(e)}(u,v),\ \ y^{(e)}(u,v)), e = 1,2,3)$ are the linear transformations of Eqs.(23)–(25) and $\hat{Q}$ is the linear convex 9- node quadrilateral GHEICJFK spanning the vertices G(1/3, 1/3), H(1/6, 5/12), E(0, ½), I(0, ¼), C(0, 0), J(1/4, 0), F(1/2, 0) , K(5/12, 1/6),L(5/24,5/24) and Δ_{pqr} is the area of triangle Δ PQR, Now, we further use the bilinear transformation of Eqns.(1)–(2) in Eqn.(15) and obtain.

$II_{\Delta PQR} = (2\,\Delta_{pqr})\ \int_{-1}^{1}\int_{-1}^{1}\{ \sum_{e=1}^{3} [\phi\,(x^{(e)}(u,v),\ \ y^{(e)}(u,v)) \frac{\partial(u,v)}{\partial(\xi,\eta)}\}\, d\xi\, d\eta$ -------------- (28)

In Eq.(16) we have used the bilinear transformation given in Eqns.(13)- (14)

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

$v = v(\xi,\ \eta) = \frac{1}{3} M_1(\xi,\ \eta) + \frac{1}{2} M_2(\xi,\ \eta)$ ---------------------- (29)

to map the arbitrary linear convex 9- noded quadrilateral into a 2 – square in (ξ, η) – plane. Thus on using Eqn.(29), the integral of Eqn.(28) simplifies to the following.

$II_{\Delta PQR} = (2\,\Delta_{pqr}) \int_{-1}^{1}\int_{-1}^{1}\ [\sum_{e=1}^{3} \left(\frac{4+\xi+\eta}{96}\right) \phi(\,x^{(e)}(u,v),\ \ y^{(e)}(u,v))\ ]d\xi\, d\eta$ ------------- (30)

We can evaluate Eqn.(30) either analytically or numerically depending on the form of the integrand.

Using Numerical Integration , we have from Eqn.(30)

$$II_{\Delta PQR} = 2\Delta_{pqr} \sum_{i=1}^{N}\sum_{j=1}^{N} \left(\frac{W_i^{(N)} W_j^{(N)} (4+\xi_i^{(N)}+\eta_j^{(N)})}{96}\right) \sum_{e=1}^{3} \phi(x^{(e)}\left(u_{i,j}^{(N)}, v_{i,j}^{(N)}\right), y^{(e)}\left(u_{i,j}^{(N)}, v_{i,j}^{(N)}\right))$$

----------------------- (31)

Where from Eqn.(29), we write

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

$v_{i,j}^{(N)} = v(\xi_i^{(N)},\ \eta_j^{(N)})$ ---------------------- (32)

and $(W_i^{(N)},\ \xi_i^{(N)})$, $(W_j^{(N)},\ \xi_j^{(N)})$ are the weight coefficients and sampling points along ξ, η directions of the Nth 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.(26) 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)9) denotes the basis functions for node i of a linear convex 9- node linear convex quadrilateral element e , then by use of chain rule of partial differentiation

$$\begin{pmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \end{pmatrix} = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial u}{\partial y} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i^e}{\partial u} \\ \frac{\partial N_i^e}{\partial v} \end{bmatrix} \qquad \text{-------------------------- (33)}$$

We note that to transform 8- node linear convex quadrilateral $Q_e(e = 1,2,3)$ of ΔPQR in Cartesian space (x,y) into $\widehat{Q}$, the 8- node linear convex quadrilateral spanned by vertices (1/3, 1/3), (1/6, 5/12), (0, ½), (0, ¼), (0, 0), (1/4, 0), (1/2, 0) and (5/12, 1/6) in uv-plane.

We must now use the earlier transformations.

$$\begin{pmatrix} x^1 \\ y^1 \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} + \begin{pmatrix} x_q - x_p \\ y_q - y_p \end{pmatrix} u + \begin{pmatrix} x_r - x_p \\ y_r - y_p \end{pmatrix} v \quad \text{for } Q_1 \text{ in } \Delta PQR \qquad \text{--------------------------- (23)}$$

$$\begin{pmatrix} x^2 \\ y^2 \end{pmatrix} = \begin{pmatrix} x_q \\ y_q \end{pmatrix} + \begin{pmatrix} x_r - x_q \\ y_r - y_q \end{pmatrix} u + \begin{pmatrix} x_p - x_q \\ y_p - y_q \end{pmatrix} v \quad \text{for } Q_2 \text{ in } \Delta PQR \qquad \text{-------------------------- (24)}$$

$$\begin{pmatrix} x^3 \\ y^3 \end{pmatrix} = \begin{pmatrix} x_r \\ y_r \end{pmatrix} + \begin{pmatrix} x_p - x_r \\ y_p - y_r \end{pmatrix} u + \begin{pmatrix} x_q - x_r \\ y_q - y_r \end{pmatrix} v \quad \text{for } Q_3 \text{ in } \Delta PQR \qquad \text{------------------------ (25)}$$

And we note that the above transformations viz Eqns.(23)-(25) are of the form

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \end{pmatrix} + \begin{pmatrix} x_a - x_c \\ y_a - y_c \end{pmatrix} u + \begin{pmatrix} x_b - x_c \\ y_b - y_c \end{pmatrix} v \qquad \text{------------------------ (34)}$$

which can map an arbitrary triangle ΔABC , 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.(34)

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} (x_a - x_c) & (x_b - x_c) \\ (y_a - y_c) & (y_b - y_c) \end{pmatrix}^{-1} \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix} \qquad \text{-------------------------- (35)}$$

This gives

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

$$v = (\propto_b + \beta_b x + \gamma_b y)/(2\,\Delta_{abc}) \qquad \text{------------------------ (36)}$$

where

$\propto_a = (x_b y_c - x_c y_b)$, $\qquad \propto_b = (x_c y_a - x_a y_c)$,

$\beta_a = (y_b - y_c)$, $\qquad \beta_b = (y_c - y_a)$,

$\gamma_a = (x_c - x_b)$, $\qquad \gamma_b = (x_a - x_c)$, $\qquad$ -----------------------(37a)

and

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

$$= (\gamma_b \beta_a - \gamma_a\,\beta_b) \qquad \text{------------------- (37b)}$$

From Eqn.(33) and Eqn.(36), we obtain

$$\begin{pmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \end{pmatrix} = \begin{pmatrix} \beta_a{}^* & \beta_b{}^* \\ \gamma_a{}^* & \gamma_b{}^* \end{pmatrix} \begin{pmatrix} \frac{\partial N_i^e}{\partial u} \\ \frac{\partial N_i^e}{\partial v} \end{pmatrix} \qquad \text{---------------------------- (38a)}$$

where $\beta_a{}^* = \frac{\beta_a}{(2\Delta_{abc})}$, $\beta_b{}^* = \frac{\beta_b}{(2\Delta_{abc})}$

$\gamma_a{}^* = \frac{\gamma_a}{(2\Delta_{abc})}$, $\gamma_b{}^* = \frac{\gamma_b}{(2\Delta_{abc})}$ --------------------------- (38b)

Letting,

$$D_{x,y}^{i,e} = \begin{pmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \end{pmatrix} , \; P = \begin{pmatrix} \beta_a{}^* & \beta_b{}^* \\ \gamma_a{}^* & \gamma_b{}^* \end{pmatrix} , \; D_{u,v}^{i,e} = \begin{pmatrix} \frac{\partial N_i^e}{\partial u} \\ \frac{\partial N_i^e}{\partial v} \end{pmatrix} \qquad \text{---------------------------- (39)}$$

We obtain from Eqn.(38) and Eqn.(39)

$$D_{x,y}^{i,e} = P\, D_{u,v}^{i,e} \qquad \text{----------------------------- (40)}$$

Hence from Eqn.(39) and Eqn.(40)

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

$$= \begin{pmatrix} \frac{\partial N_i^e}{\partial x}\frac{\partial N_j^e}{\partial x} & \frac{\partial N_i^e}{\partial x}\frac{\partial N_j^e}{\partial y} \\ \frac{\partial N_i^e}{\partial y}\frac{\partial N_j^e}{\partial x} & \frac{\partial N_i^e}{\partial y}\frac{\partial N_j^e}{\partial y} \end{pmatrix} \qquad \text{----------------------------- (41a)}$$

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

$$= \begin{pmatrix} \frac{\partial N_i^e}{\partial u}\frac{\partial N_j^e}{\partial u} & \frac{\partial N_i^e}{\partial u}\frac{\partial N_j^e}{\partial v} \\ \frac{\partial N_i^e}{\partial v}\frac{\partial N_j^e}{\partial u} & \frac{\partial N_i^e}{\partial v}\frac{\partial N_j^e}{\partial v} \end{pmatrix} \qquad \text{----------------------------- (41b)}$$

We have now from Eqn.(40) and Eqn.(41a- b)

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

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

$$= P\, G_{u,v}^{i,j,e}\, P^T \qquad \text{--------------------------- (41c)}$$

We now define the submatrices of global derivative integrals in (x,y) and (u,v) space associated with the nodes i and j $(i, j = 1, 2, 3, 4, 5, 6, 7, 8, 9)$ as

$$S^{i,j,e} = \iint_{Q_e} G_{x,y}^{i,j,e} \; dx\, dy , \qquad \text{--------------------------- (42)}$$

$$K^{i,j,e} = \iint_{\hat{Q}} G_{u,v}^{i,j,e} \; du\, dv \qquad \text{--------------------------- (43)}$$

where, we have already defined the 8- node linear convex quadrilaterals Q_e (e=1,2,3) in (x,y) space and $\widehat{Q}$ in (u,v) space in Fig 3a- 3b. From Eqns.(41)-(43) , 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.(41a)- (41b)

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

$$= \begin{pmatrix} S^{e}_{2i-1,2j-1} & S^{e}_{2i-1,2j} \\ S^{e}_{2i,2j-1} & S^{e}_{2i,2j} \end{pmatrix} \quad \text{(say)} \quad \text{------------------(44)}$$

and in similar manner

$$K^{i,j,e} = \iint_{\widehat{Q}} G^{i,j,e}_{u,v} \; du\,dv = \begin{pmatrix} \iint_{\widehat{Q}} \frac{\partial N^e_i}{\partial u}\frac{\partial N^e_j}{\partial u}\,dudv & \iint_{\widehat{Q}} \frac{\partial N^e_i}{\partial u}\frac{\partial N^e_j}{\partial v}\,dudv \\ \iint_{\widehat{Q}} \frac{\partial N^e_i}{\partial v}\frac{\partial N^e_j}{\partial u}\,dudv & \iint_{\widehat{Q}} \frac{\partial N^e_i}{\partial v}\frac{\partial N^e_j}{\partial v}\,dudv \end{pmatrix}$$

$$= \begin{pmatrix} K^{e}_{2i-1,2j-1} & K^{e}_{2i-1,2j} \\ K^{e}_{2i,2j-1} & K^{e}_{2i,2j} \end{pmatrix} \quad \text{(say)} \quad \text{------------------(45)}$$

We have now from the above Eqns.(41)-(45)

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

$$= 2\Delta_{abc} \iint_{\widehat{Q}} (P\,G^{i,j,e}_{u,v}P^{T}) \; du\,dv$$

$$= 2\Delta_{abc}\, P\, (\iint_{\widehat{Q}} G^{i,j,e}_{u,v}\, du\,dv)\; P^{T}$$

$$= 2\Delta_{abc}\; P\,(K^{i,j,e})\; P^{T} \quad , \; (i,j = 1,2,3,4,5,6,7,8,9) \quad \text{----------------(46)}$$

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.(46).

We note that $\widehat{Q}$ is the 8- node linear convex quadrilateral in (u, v) space spanned by the vertices (1/3, 1/3), (1/6, 5/12), (0, ½), (0, ¼), (0, 0), (1/4, 0), (1/2, 0) , (5/12, 1/6) and (5/24,5/24) in uv- plane hence from Eqn.(45)

$$K^{i,j,e} = \iint_{\widehat{Q}} G^{i,j,e}_{u,v} \; du\,dv \quad \text{-------------------------(47)}$$

$$= \int_{-1}^{1}\int_{-1}^{1} G^{i,j,e}_{u,v}\, \frac{\partial(u,v)}{\partial(\xi,\eta)}\; d\xi\, d\eta \quad \text{------------------------(48)}$$

We now refer to section 6.1 of this paper, in this section, we have derived the necessary relations to integrate Eq.(47). As in Eqns.(27)-(28), we use the transformation of Eqn.(29) to map the 9- node quadrilateral $\widehat{Q}$ to the 9- node 2-square $-1 \le \xi,\eta \le 1$ Using Eqn.(29) in Eqn.(48), we obtain

$$K^{i,j,e} = \iint_{\widehat{Q}} G^{i,j,e}_{u,v} \; (\tfrac{4+\xi+\eta}{96})\; d\xi d\eta \quad \text{----------------------- (49)}$$

Thus, we have from Eq.(46)

$$S^{i,j,e} = \; (2\Delta_{abc})\; P\,(K^{i,j,e})\; P^{T} \quad \text{---------------------- (50)}$$

Where $K^{i,j,e}$ is given in Eqn.(49)

In Eqn.(50) , $2\Delta_{abc}$= 2 * 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 nodel coordinates (x_a ,y_a) ,(x_b ,y_b) , (x_c , y_c) the matrix $K^{i,j,e}$ can be explicity computed by the relations obtained in section 2 – 6. We find that $K^{i,j,e}$ is a (2X2) matrix of integrals whose integrands are rational functions with polynomial numerator and the linear denominator $(4 + \xi + \eta)$. Hence these integrals can be explicity 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.(45) is of order $(2xn_{de})$ x $(2xn_{de})$, n_{de} = 8 = for 8-node convex quadrilateral element.

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

We may note that In order to compute the local/element stiffness matrices for the Poisson Boundary Value problem, we need to compute the integrals Eqns(12a-b)

$$K^e_{i,j} = \int_{\Omega^e} \nabla\varphi_i \,.\, \nabla\varphi_j \, d\mathrm{x} = \int_{\Omega^e} \{ \frac{\partial\varphi_i}{\partial x}\frac{\partial\varphi_j}{\partial x} + \frac{\partial\varphi_i}{\partial y}\frac{\partial\varphi_j}{\partial y} \} \, dxdy \, ,$$

,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,(51a)

from the above derivations, we can rewrite $K^e_{i,j}$ in the notations of this sections by taking $\varphi_i = \mathrm{N}_i$ and $\varphi_j = N_j$ and $\Omega^e = Q_e$ so that

$$K^e_{i,j} = \int_{Q_e} \nabla \mathrm{N}_i \,.\, \nabla N_j \, d\mathrm{x} = \int_{Q_e} \{ \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} \} \, dxdy = \mathrm{S}^{\mathrm{e}}_{2\mathrm{i}-1\,,2\mathrm{j}-1} + \mathrm{S}^{\mathrm{e}}_{2\mathrm{i}\,,2\mathrm{j}}$$

....................................(51b)

6.3 Computation of $K^e_{i,j}$

The explicit integration scheme explained above compute four derivative product integrals as given in eqn(44) and they are necessary to compute the stiffness matrix entries of plane stress/plane strain problems in elasticity and sevral other applications in continuum mechanics.But this computation requires matrix triple product as given in eqn (50).Since,we only need the sum of two of these integrals viz : $\mathrm{S}^{\mathrm{e}}_{2\mathrm{i}-1\,,2\mathrm{j}-1} + \mathrm{S}^{\mathrm{e}}_{2\mathrm{i}\,,2\mathrm{j}}$ .We now present an efficient method to compute this sum by using matrix product.

Let $F^{i,j}_{p.q} = \frac{\partial N_i}{\partial p}\frac{\partial N_j}{\partial q}$, $I^{i,j}_{p.q} = \int_{Q_e} F^{i,j}_{p.q} \, dpdq$, then we have from eqns(44-45) :

$$\mathrm{S}^{\mathrm{i,j,e}} = \iint_{\mathrm{Q_e}} \mathrm{G}^{\mathrm{i,j,e}}_{\mathrm{x\,,y}} \;\; \mathrm{dx\,dy} = \begin{pmatrix} \iint_{\mathrm{Q_e}} \frac{\partial \mathrm{N^e_i}}{\partial \mathrm{x}}\frac{\partial \mathrm{N^e_j}}{\partial \mathrm{x}} \, \mathrm{dxdy} & \iint_{\mathrm{Q_e}} \frac{\partial \mathrm{N^e_i}}{\partial \mathrm{x}}\frac{\partial \mathrm{N^e_j}}{\partial \mathrm{y}} \, \mathrm{dxdy} \\ \iint_{\mathrm{Q_e}} \frac{\partial \mathrm{N^e_i}}{\partial \mathrm{y}}\frac{\partial \mathrm{N^e_j}}{\partial \mathrm{x}} \, \mathrm{dxdy} & \iint_{\mathrm{Q_e}} \frac{\partial \mathrm{N^e_i}}{\partial \mathrm{y}}\frac{\partial \mathrm{N^e_j}}{\partial \mathrm{y}} \, \mathrm{dxdy} \end{pmatrix}$$

$$= \begin{pmatrix} \mathrm{S}^{\mathrm{e}}_{2\mathrm{i}-1\,,2\mathrm{j}-1} & \mathrm{S}^{\mathrm{e}}_{2\mathrm{i}-1\,,2\mathrm{j}} \\ \mathrm{S}^{\mathrm{e}}_{2\mathrm{i}\,,2\mathrm{j}-1} & \mathrm{S}^{\mathrm{e}}_{2\mathrm{i}\,,2\mathrm{j}} \end{pmatrix} \;\; \text{(say)}$$

$$= \begin{pmatrix} I^{i,j}_{x,x} & I^{i,j}_{x,y} \\ I^{i,j}_{y,x} & I^{i,j}_{y,y} \end{pmatrix}$$

..(52a)

$$\mathrm{K}^{\mathrm{i,j,e}} = \iint_{\hat{\mathrm{Q}}} \mathrm{G}^{\mathrm{i,j,e}}_{\mathrm{u\,,v}} \;\; \mathrm{du\,dv} = \begin{pmatrix} \iint_{\hat{\mathrm{Q}}} \frac{\partial \mathrm{N^e_i}}{\partial \mathrm{u}}\frac{\partial \mathrm{N^e_j}}{\partial \mathrm{u}} \, \mathrm{dudv} & \iint_{\hat{\mathrm{Q}}} \frac{\partial \mathrm{N^e_i}}{\partial \mathrm{u}}\frac{\partial \mathrm{N^e_j}}{\partial \mathrm{v}} \, \mathrm{dudv} \\ \iint_{\hat{\mathrm{Q}}} \frac{\partial \mathrm{N^e_i}}{\partial \mathrm{v}}\frac{\partial \mathrm{N^e_j}}{\partial \mathrm{u}} \, \mathrm{dudv} & \iint_{\hat{\mathrm{Q}}} \frac{\partial \mathrm{N^e_i}}{\partial \mathrm{v}}\frac{\partial \mathrm{N^e_j}}{\partial \mathrm{v}} \, \mathrm{dudv} \end{pmatrix}$$

$$= \begin{pmatrix} \mathrm{K}^{\mathrm{e}}_{2\mathrm{i}-1\,,2\mathrm{j}-1} & \mathrm{K}^{\mathrm{e}}_{2\mathrm{i}-1\,,2\mathrm{j}} \\ \mathrm{K}^{\mathrm{e}}_{2\mathrm{i}\,,2\mathrm{j}-1} & \mathrm{K}^{\mathrm{e}}_{2\mathrm{i}\,,2\mathrm{j}} \end{pmatrix} \;\; \text{(say)}$$

$$= \begin{pmatrix} I^{i,j}_{u,u} & I^{i,j}_{u,v} \\ I^{i,j}_{v,u} & I^{i,j}_{v,v} \end{pmatrix}$$

...(52b)

Let $\mathrm{P} = \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix}, \; P^T = \begin{pmatrix} P_{11} & P_{21} \\ P_{12} & P_{22} \end{pmatrix}$

...(53)

From eqns(44) , (46) and (52a-b)

$$\mathrm{S}^{\mathrm{i,j,e}} = \iint_{\mathrm{Q_e}} \mathrm{G}^{\mathrm{i,j,e}}_{\mathrm{x\,,y}} \;\; \mathrm{dx\,dy} \; = 2\Delta_{\mathrm{abc}} \, \mathrm{P} \, (\iint_{\hat{\mathrm{Q}}} \mathrm{G}^{\mathrm{i,j,e}}_{\mathrm{u\,,v}} \, \mathrm{du\,dv}) \;\; \mathrm{P^T}$$

$$= 2\Delta_{\mathrm{abc}} \begin{pmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{pmatrix} \begin{pmatrix} I^{i,j}_{u,u} & I^{i,j}_{u,v} \\ I^{i,j}_{v,u} & I^{i,j}_{v,v} \end{pmatrix} \begin{pmatrix} P_{11} & P_{21} \\ P_{12} & P_{22} \end{pmatrix}$$

$$= 2\Delta_{\mathrm{abc}} \begin{pmatrix} \{ \mathrm{P_{11}}(\mathrm{P_{11}}\mathrm{I^{i,j}_{u.u}} + \mathrm{P_{12}}\mathrm{I^{i,j}_{u,v}}) + \mathrm{P_{12}}(\mathrm{P_{11}}\mathrm{I^{i,j}_{v,u}} + \mathrm{P_{12}}\mathrm{I^{i,j}_{v,v}}) \; \} & \{ \mathrm{P_{11}}(\mathrm{P_{21}}\mathrm{I^{i,j}_{u,u}} + \mathrm{P_{22}}\mathrm{I^{i,j}_{u.v}}) + \mathrm{P_{12}}(\mathrm{P_{21}}\mathrm{I^{i.j}_{v,u}} + \mathrm{P_{22}}\mathrm{I^{i,j}_{v.v}}) \} \\ \{ \mathrm{P_{21}}(\mathrm{P_{11}}\mathrm{I^{i,j}_{u,u}} + \mathrm{P_{12}}\mathrm{I^{i,j}_{u,v}}) + \mathrm{P_{22}}(\mathrm{P_{11}}\mathrm{I^{i,j}_{u.u}} + \mathrm{P_{12}}\mathrm{I^{i.j}_{v,v}}) \; \} & \{ \mathrm{P_{21}}(\mathrm{P_{21}}\mathrm{I^{i,j}_{u,u}} + \mathrm{P_{22}}\mathrm{I^{i.j}_{u,v}}) + \mathrm{P_{22}}(\mathrm{P_{21}}\mathrm{I^{i,j}_{v,u}} + \mathrm{P_{22}}\mathrm{I^{i,j}_{v,v}}) \} \end{pmatrix}$$

……………………………………………………………………………………………………(54)

From eqn(51a-b) and eqn(46) , we find

$$\text{trace}\ (S^{i,j,e}) = \text{trace}(\iint_{Q_e} G^{i,j,e}_{x,y}\ dxdy\ ) = (S^e_{2i-1,2j-1} + S^e_{2i,2j}) = K^e_{i,j} = \int_{Q_e} \nabla N_i \,.\, \nabla N_j\ dx = \int_{Q_e}\{\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\}\, dxdy$$

$$= (P_{11}^2 + P_{21}^2)I^{i,j}_{u,u} + (P_{11}\,P_{12} + P_{21}\,P_{22})\left(I^{i,j}_{u,v} + I^{i,j}_{v,u}\right) + (P_{12}^2 + P_{22}^2)I^{i,j}_{v,v} \qquad \text{..........(55)}$$

W e can obtain the above integraql $\int_{Q_e}\{\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\}\, dxdy$ by use of matrix operations which doesnot need the computation matrix triple product.This procedure is presented below

From eqn (44b) and eqn(45),let us do the following:

$$(P^T\,P)\,.* \begin{pmatrix} I^{i,j}_{u,u} & I^{i,j}_{u,v} \\ I^{i,j}_{v,u} & I^{i,j}_{v,v} \end{pmatrix} = \begin{bmatrix} (P_{11}^2 + P_{21}^2)\, I^{i,j}_{u,u} & (\,P_{11}\,P_{12} + P_{21}\,P_{22})I^{i,j}_{u,v} \\ (\,P_{11}\,P_{12} + P_{22}\,P_{21})I^{i,j}_{v,u} & (P_{12}^2 + P_{22}^2)I^{i,j}_{v,v} \end{bmatrix} \text{...................................(56)}$$

We observe from eqn(56) that sum of all the entries gives us the value of the integral i.e

$$\int_{Q_e}\{\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\}\, dxdy = sum(sum\left(\ (P^T\,P).* \begin{pmatrix} I^{i,j}_{u,u} & I^{i,j}_{u,v} \\ I^{i,j}_{v,u} & I^{i,j}_{v,v} \end{pmatrix}\ \right)) \qquad \text{............................(57)}$$

Where,sum is a Matlab function.We note that S=sum(X) gives the sum of the elements of vector X. If X is a matrix then S is a row vector with the sum over each column.It is clear that sum(sum(X)) gives the sum of all the entries in a matrix X.

6.4 Computing of Force Vector Integrals $\int_{\Omega^e} f\varphi_i$ dxdy

We shall now propose numerical integration for the complicated integrands in the force vector integrals over the domain Ω^e which is an arbitrary linear triangle and $\phi(x,y) = f\varphi_i$ .We also refer to the section 2 for the theory necessary to derive the composite numerical integration formla

We shall now establish a composite integration formula for an arbitrary linear triangular region Δ PQR shown in Fig 2a or Fig 3a. We have for an arbitrary smooth function $\phi(x, y)$

$$II_{\Delta PQR} = \iint_{\Delta PQR} \phi(x,y)\ dxdy = \sum_{e=1}^{3} \iint_{Q_e} \phi(x,y)\ dxdy \qquad \text{---------------------- (58)}$$

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

$$= (2\ \Delta_{pqr})\ \iint_{\hat{Q}}\ \{\sum_{e=1}^{3}[\phi\ (x^{(e)}(u,v),\ \ y^{(e)}(u,v))\ ]\}\ dudv \qquad \text{------------------- (59)}$$

Where $(x^{(e)}(u,v),\ \ y^{(e)}(u,v)), e = 1, 2, 3)$ are the transformations of Eqs.(8)–(10) and $\hat{Q}$ is the quadrilateral in uv- plane spanned by vertices G(1/3 ,1/3) , E(0, ½) , C(0, 0) and F(1/2,0) , and Δ_{pqr} is the area of triangle Δ PQR, Now using the transformations defined in Eqs.(1)–(2) we obtain

$$II_{\Delta PQR} = (2\ \Delta_{pqr})\ \iint_{\hat{Q}}\ \{\sum_{e=1}^{3}[\phi\ (x^{(e)}(u,v),\ \ y^{(e)}(u,v))\,\frac{\partial(u,v)}{\partial(\xi,\eta)}\}\ d\xi\ d\eta \qquad \text{--------------- (60)}$$

In Eq.(14) we have used the transformation

$$u(\xi,\ \eta) = \frac{1}{3}N_1(\xi,\ \eta) + \frac{1}{2}N_4(\xi,\ \eta)$$

$$v(\xi,\ \eta) = \frac{1}{3}N_1(\xi,\ \eta) + \frac{1}{2}N_2(\xi,\ \eta) \qquad \text{---------------------- (61)}$$

to map the quadrilateral $\hat{Q}$ into a 2 – square in ξη – plane.

We can now obtain from Eqs.(14)–(15)

$$II_{\Delta PQR} = (2\ \Delta_{pqr}) \int_{-1}^{1}\int_{-1}^{1}\ [\sum_{e=1}^{3}\ \left(\frac{4+\xi+\eta}{96}\right)\phi(\ x^{(e)}(u,v),\ \ y^{(e)}(u,v))\ ]d\xi\ d\eta \qquad \text{------------- (62)}$$

We can evaluate Eq.(16) either analytically or numerically depending on the form of the integrand.

Using Numerical Integration ;

$$II_{\Delta PQR} = 2\Delta_{pqr} \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{W_i^{(N)}W_j^{(N)}(4+\xi_i^{(N)}+\eta_j^{(N)})}{96}\right) \sum_{e=1}^{3}\phi(x^{(e)}\left(u_{i,j}^{(N)}, v_{i,j}^{(N)}\right), y^{(e)}\left(u_{i,j}^{(N)}, v_{i,j}^{(N)}\right)\,)$$

------ ----------------- (63)

Where,

$u_{i,j}^{(N)} = u(\xi_i^{(N)},\ \eta_j^{(N)})$ and $v_{i,j}^{(N)} = v(\xi_i^{(N)},\ \eta_j^{(N)})$ --------- ------------- (64)

and $(W_i^{(N)},\ \xi_i^{(N)})$, $(W_j^{(N)},\ \xi_j^{(N)})$ are the weight coefficients and sampling points of Nth order Gauss Legendre Quadrature rules.

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

The above method will help in integrating $\int_{\Omega^e} f\varphi_i$ dxdy,when the intgrand $f\varphi_i$ is complicated

7 A New Approach To Mesh Generation

The first step in implimenting finite element method isto generate a mesh.In a recent work the author and his co-workers have proposed a new approach to mesh generation which can discretise a convex polygon into an all quadrilateral mesh.This will be presented next.This new approach to mesh generation meets the necessary requirements of regularity on the shape of elements.There are two types of them which usually suffice in finite element computations.The first is called shape regularity. It says that the ratio of the diameter of the element to the radius of the inner circle must be less than some constant. For triangles,the diameter of the triangle is related to the smallest circle which contain the triangle.The inner circle refers to the largest circle which fits inside the triangle. Shape regularity focuses on the shape of individual triangles and doesnot refer to how the shapes of different elements relate to each other. So some elements can be large wshile others might be very small. There is a second type of requirement on the shape of elements.This requirement says that ratio of the maximum diameter of elements to the radius of the inner circle of an element must be less than some constant .If a mesh satisfies this requirement,it is called quasiuniform.This requirement is more important when we perform refinements.We must note that a mesh generation gives us the nodes on a particular element as well as the coordinates of the nodes.We now give an account of this novel mesh generation technique with an aim to use it further in the solution of Poisson problem. Stated in eqn(7a-b).

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 which is applied to torison of square shaft, on considering symmetry of the problem domain, mesh generation for 1/8 of the cross section which is a triangle was discritised into an all quadrilateral mesh. In this paper we consider applications to polygonal domains.

7.1 An automatic indirect quadrilateral mesh generator

A wide range of problems in applied science and engineering can be simulated by partial derivative equations(PDE).In the last few decade,one of the most relevant techniques to solve is the Finite Element Method(FEM).It is well known that a good quality mesh is required in order to obtain an accurate solution.Hence the construction of a mesh is on e of the most important steps.

In the next few sections , we present a novel mesh generation scheme of all quadrilateral elements for convex polygonal domains. This scheme converts the elements in background triangular mesh into quadrilaterals through the operation of splitting. We first decompose the convex polygon into simple subregions in the shape of triangles. These simple subregions are then triangulated to generate a fine mesh of triangles. We propose then an automatic triangular to quadrilateral conversion scheme in which each isolated triangle is split into three quadrilaterals according to the usual scheme, adding three vertices in the middle of edges and a vertex at the barrycentre of the triangular element. Further, to preserve the mesh conformity a similar procedure is also applied to every triangle of the domain and this fully discretizes the given convex polygonal domain into all quadrilaterals, thus propogating uniform refinement. In section 4.2, we present a scheme to discretize the arbitrary and standard triangles into a fine mesh of six node triangular elements. In section 4.3, we explain the procedure to split these triangles into quadrilaterals. In section 4.4,we have presented a method of piecing together of all triangular subregions and eventually creating a all quadrilateral mesh for the given convex polygonal domain. In section 4.5,we present several examples to illustrate the simplicity and efficiency of the proposed mesh generation method for standard and arbitrary triangles,rectangles and convex polygonal domains.

7.2 Division of an Arbitrary Triangle

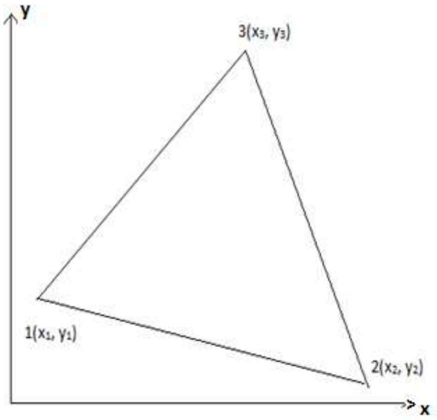
We can map an arbitrary triangle with vertices $(\ (x_i, y_i),\ i = 1, 2, 3)$ into a right isosceles triangle in the (u, v) space as shown in Fig. 4a, b. The necessary transformation is given by the equations.

$$x = x_1 + (x_2 - x_1)u + (x_3 - x_1)v$$

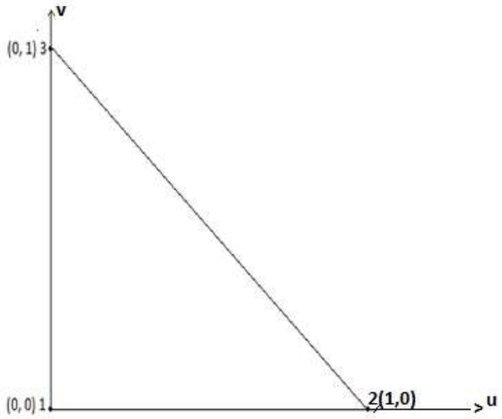
$$y = y_1 + (y_2 - y_1)u + (y_3 - y_1)v$$

(57)

The mapping of eqn.(1) describes a unique relation between the coordinate systems. This is illustrated by using the area coordinates and division of each side into three equal parts in Fig. 5a Fig. 5b. It is clear that all the coordinates of this division can be determined by knowing the coordinates $(\,(x_i, y_i),\ i = 1,2,3)$ of the vertices for the arbitrary triangle. In general , it is well known that by making 'n' equal divisions on all sides and the concept of area coordinates, we can divide an arbitrary triangle into n^2 smaller triangles having the same area which equals Δ/n^2 where Δ is the area of a linear arbitrary triangle with vertices $(\,(x_i, y_i),\ i = 1,2,3)$ in the Cartesian space.



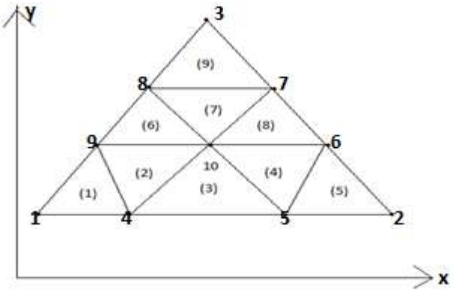
4.a



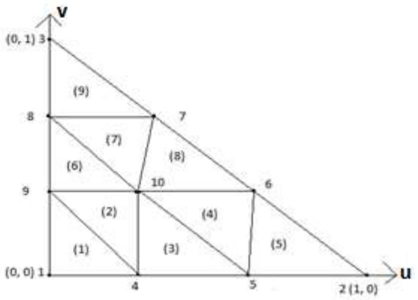
4 b

Fig. 4a An Arbitrary Linear Triangle in the (x, y) space

Fig. 4b A Right Isosceles Triangle in the (u, v) space



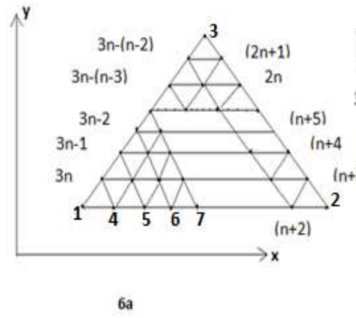
5a



5b

Fig. 5a Division of an arbitrary triangle into Nine triangles in Cartesian space

Fig. 5b Division of a right isosceles triangle into Nine right isosceles triangles in (u, v) space



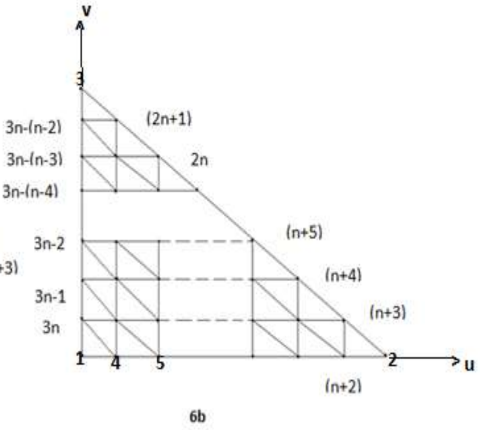


Fig.6a Division of an arbitrary triangle into n^2 triangle in Cartesian space (x, y), where each side is divided into n divisions of equal length

Fig. 6b Division of a right isosceles triangle into n^2 right isosceles triangle in (u, v) space, where each side is divided into n divisions of equal length

We have shown the division of an arbitrary triangle in Fig. 6a , Fig. 6b, We divided each side of the triangles (either in Cartesian space or natural space) into n equal parts and draw lines parallel to the sides of the triangles. This creates (n+1) (n+2) nodes. These nodes are numbered from triangle base line l_{12} (letting l_{ij} as the line joining the vertex (x_i, y_i) and (x_j, y_j)) along the line $v = 0$ and upwards up to the line $v = 1$. The nodes 1, 2, 3 are numbered anticlockwise and then nodes 4, 5, ------, (n+2) are along line $v = 0$ and the nodes (n+3), (n+4), ------, 2n, (2n+1) are numbered along the line l_{23} i.e. $u + v = 1$ and then the node (2n+2), (2n+3), -------, 3n are numbered along the line $u = 0$. Then the interior nodes are numbered in increasing order from left to right along the line $v = \frac{1}{n}, \frac{2}{n}, ---, \frac{n-1}{n}$ bounded on the right by the line $u + v = 1$. Thus the entire triangle is covered by (n+1) (n+2)/2 nodes. This is shown in the $\underline{rr}$ matrix of size $(n+1) \times (n+1)$, only nonzero entries of this matrix refer to the nodes of the triangles

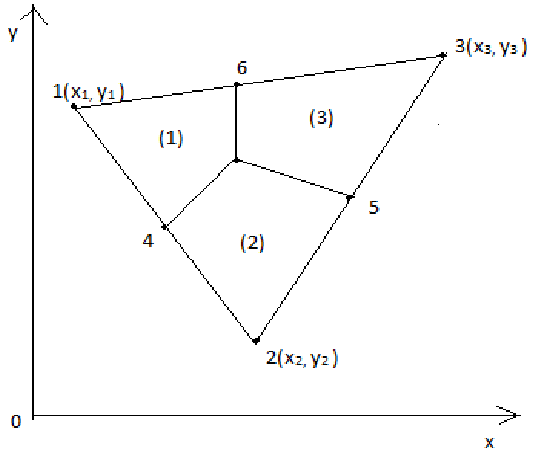
$$\underline{rr} = \begin{bmatrix} 1, & 4, & 5, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, & (n+2) & 2 \\ 3n, & (3n+1), \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots, & 3n+(n-2), & (n+3) & 0 \\ 3n-1, & 3n+(n-1) \ldots\ldots\ldots\ldots\ldots & ,3n+(n-2)+(n-3), & (n+4) \quad 0 & 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ 3n-(n-3), & \frac{(n+1)(n+2)}{2}, & 2n & 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & 0 \\ 3n-(n-2), & (2n+1), & 0 & 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. & 0 \\ 3 & 0 & 0 & 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & 0 \end{bmatrix}$$

..(58)

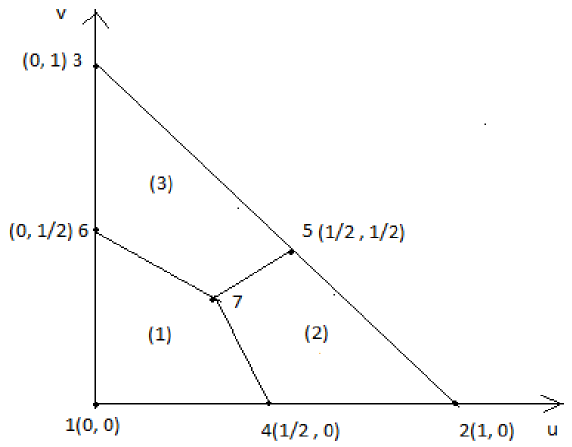
7.3. Quadrangulation of an Arbitrary Triangle

We now consider the quadrangulation of an arbitrary triangle. We first divide the arbitrary triangle into a number of equal size six node triangles. Let us define l_{ij} as the line joining the points (x_i, y_i) and (x_j, y_j) in the Cartesian space (x, y). Then the arbitrary triangle with vertices at $((x_i, y_i), i = 1, 2, 3)$ is bounded by three lines l_{12}, l_{23}, and l_{31} . By dividing the sides l_{12}, l_{23}, l_{31} into $n = 2m$ divisions (m, an integer) creates m^2 six node triangular divisions. Then by joining the centroid of these six node triangles to the midpoints of their sides, we obtain three quadrilaterals for each of these triangle. We have illustrated this process for the two and four divisions of l_{12}, l_{23}, and l_{31} sides of the arbitrary and standard triangles in Figs. 4 and 5

Two Divisions of Each side of an Arbitrary Triangle



7(a)

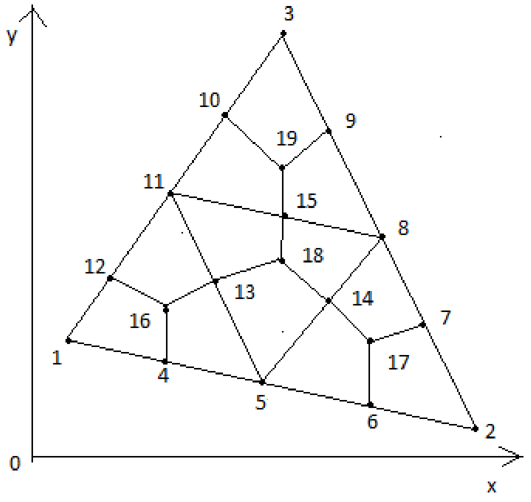


7(b)

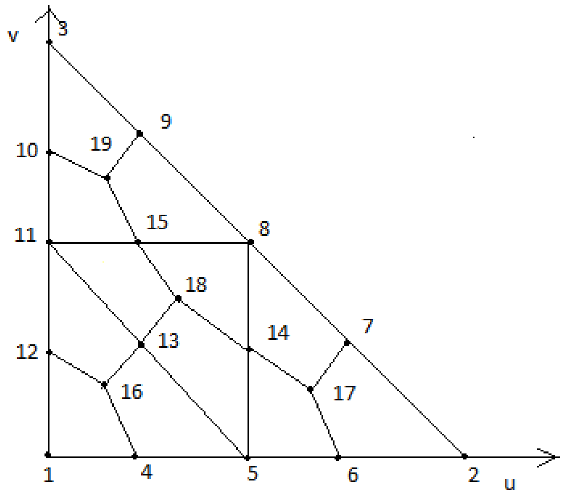
Fig 7(a). Division of an arbitrary triangle into three quadrilaterals

Fig 7(b). Division of a standard triangle into three quadrilaterals

Four Divisions of Each side of an Arbitrary Triangle



8a



8b

Fig 8a. Division of an arbitrary triangle into 4 six node triangles

Fig 8b. Division of a standard triangle into 4 right isosceles triangle

In general, we note that to divide an arbitrary triangle into equal size six node triangle, we must divide each side of the triangle into an even number of divisions and locate points in the interior of triangle at equal spacing. We also do similar divisions and locations of interior points for the standard triangle. Thus n (even) divisions creates ($n/2)^2$ six node triangles in both the spaces. If the entries of the sub matrix $\underline{rr}\,(i\,;\,i+2,\;j\,;\,j+2)$ are nonzero then two six node triangles can be formed. If $\underline{rr}\,(i+1,\;j+2) = \underline{rr}\;(i+2,\;j+1;\;j+2) = 0$ then one six node triangle can be formed. If the sub matrices $\underline{rr}\,(i\,;\,i+2,\;j\,;\,j+2)$ is a (3×3) zero matrix , we cannot form the six node triangles. We now explain the creation of the six node triangles using the $\underline{rr}$ matrix of eqn.( ). We can form six node triangles by using node points of three consecutive rows and columns of $\underline{rr}$ matrix. This procedure is depicted in Fig. 9 for three consecutive rows $i\,,i+1$, $i+2$ and three consecutive columns $j\,,\;j+1,\;j+2$ of the $\underline{rr}$ sub matrix

Formation of six node triangles using sub matrix $\underline{rr}$

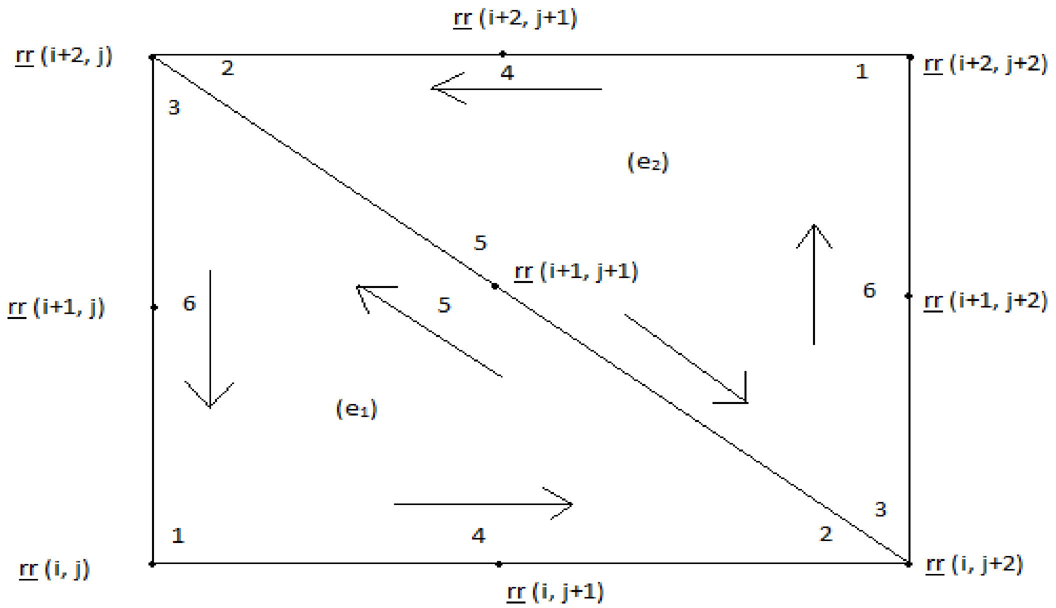


Fig. 9 Six node triangle formation for non zero sub matrix $\underline{rr}$

If the sub matrix ($(\underline{rr}\ (\ k,l), k = i, i+1, i+2), l = j,\ j+1,\ j+2)$ is nonzero, then we can construct two six node triangles. The element nodal connectivity is then given by

(e_1) $< \underline{rr}\ (i,\ j), \underline{rr}\,(i, i+2), \underline{rr}\,(i+2,\ j), \underline{rr}\,(i, j+1), \underline{rr}\,(i+1, j+1), \underline{rr}\,(i+1, j) >$

(e_2) $< \underline{rr}\ (i+2, j+2), \underline{rr}\,(i+2, j), \underline{rr}\,(i, j+2), \underline{rr}\,(i+2, j+1), \underline{rr}\,(i+1, j+1), \underline{rr}\ \ i+1, j+2) >$ ....(59)

If the elements of sub matrix ($(\underline{rr}\ (\ k,l), k = i, i+1, i+2),\ l = j,\ j+1,\ j+2)$ are nonzero, then as standard earlier, we can construct two six node triangles. We can create three quadrilaterals in each of these six node triangles. The nodal connectivity for the 3 quadrilaterals created in (e_1) are given as

$Q_{3n_1-2} < c_1, \underline{rr}\ (i+1,\ j), \underline{rr}\,(i,\ j), \underline{rr}\,(i, j+1) >$

$Q_{3n_1-1} < c_1, \underline{rr}\ (i,\ j+1), \underline{rr}\,(i,\ j+2), \underline{rr}\,(i+1, j+1) >$

$Q_{3n_1} < c_1, \underline{rr}\ (i+1,\ j+1), \underline{rr}\ (i+2,\ j), \underline{rr}\ (i+1, j) >$ ....(60)

and the nodal connectivity for the 3 quadrilaterals created in (e_2) are given as

$Q_{3n_2-2} < c_2, \underline{rr}\ (i+1,\ j+2), \underline{rr}\,(i+2,\ j+2), \underline{rr}\,(i+2,\ j+1) >$

$Q_{3n_2-1} < c_2, \underline{rr}\ (i+2,\ j+1), \underline{rr}\,(i+2,\ j), \underline{rr}\,(i+1,\ j+1) >$

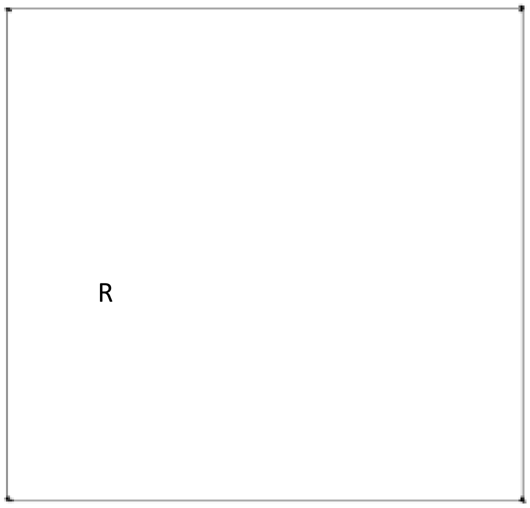
$Q_{3n_1} < c_2, \underline{rr}\ (i+1,\ j+1), \underline{rr}\,(i,\ j+2), \underline{rr}\,(i+1, j+2) >$ -------------------- (61)

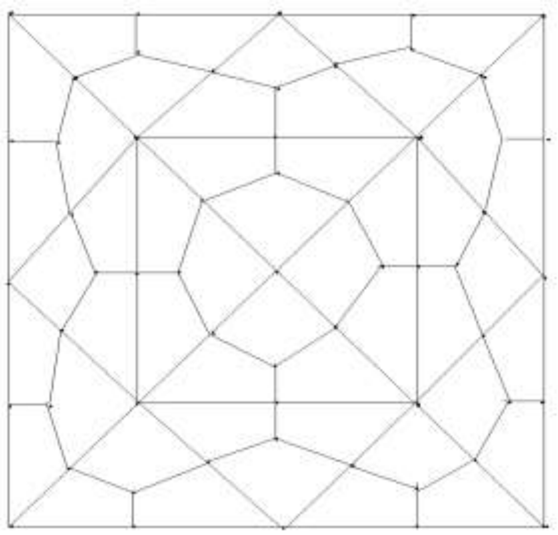
7.4 Quadrangulation of the Polygonal Domain

We can generate polygonal meshes by piecing together triangular with straight sides. Subsection (called LOOPs). The user specifies the shape of these LooPs by designating six coordinates of each LOOP

As an example, consider the geometry shown in Fig. 8(a). This is a a square region which is simply chosen for illustration. We divide this region into four LOOPs as shown in Fig.8(d). These LOOPs 1,2,3 and 4 are triangles each with

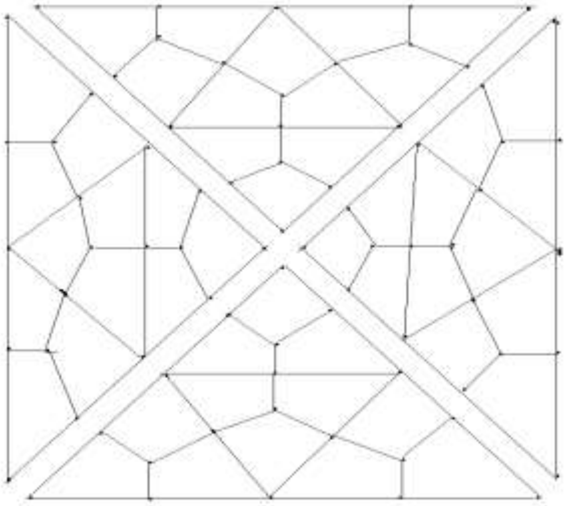
three sides. After the LOOPs are defined, the number of elements for each LOOP is selected to produce the mesh shown in Fig. 8(c).The complete mesh is shown in Fig.8(b)

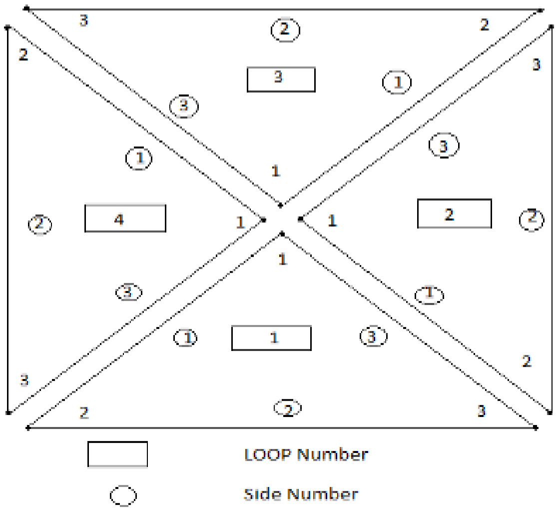




10a **10b**

(i)Fig.10a: Region R to be analyzed **(ii) Fig.10b: Example of completed mesh**





10c **10d**

(iii)Fig.10c:Exploded view showing four loops (iv)Fig.10d:Example of a loop and side numbering scheme

How to define the LOOP geometry, specify the number of elements and piece together the LOOPs will now be explained

Joining LOOPs : A complete mesh is formed by piecing together LOOPs. This piecing is done sequentially thus, the first LOOP formed is the foundation LOOP, with subsequent LOOPs joined either to it or to other LOOPs that have already been defined. As each LOOP is defined, the user must specify for each of the three sides of the current LOOP.

In the present mesh generation code, we aim to create a convex polygon. This requires a simple procedure. We join side 3 0f LOOP 1 to side 1 of LOOP 2, side 3 of LOOP 2 will joined to side 1 of LOOP 3, side 3 of LOOP 3 will be joined to side 1 of LOOP 4. Finally side 3 of LOOP 4 will be joined to side 1 of LOOP 1.

When joining two LOOPs, it is essential that the two sides to be joined have the same number of divisions. Thus the number of divisions remains the same for all the LOOPs. We note that the sides of LOOP (i) and side of LOOP ($i+1$) share the same node numbers. But we have to reverse the sequencing of node numbers of side 3 and assign them as node

numbers for side 1 of LOOP $(i+1)$. This will be required for allowing the anticlockwise numbering for element connectivity.

The auto mesh generation technique discritises a polygonal domain into all four node special quadrilateral elements.We can convert these into nine node special quadrilateral elements by adding one node at the midpoint of each side of the four node special quadrilateral elements,and also the ninth node at the centroid of the quadrilateral elements. We have written codes to carry this conversion schemes in the programs of all four node special quadrilaterals proposed in[ ].We include here some meshes of all nine node special quadrilaterals at initial stagtes of mesh generation which is self explainatory.

Example 1: right isosceles triangle
x=sym([0;1/2;1/2])
y=sym([0 0;1/2])
We use this mesh to solve torsion of a square cross section,due to symmetry considerations mesh generation over the above domain.This is a case of Poisson equation with constant right hand side(=-2).Our main aim is to compute torsional constant

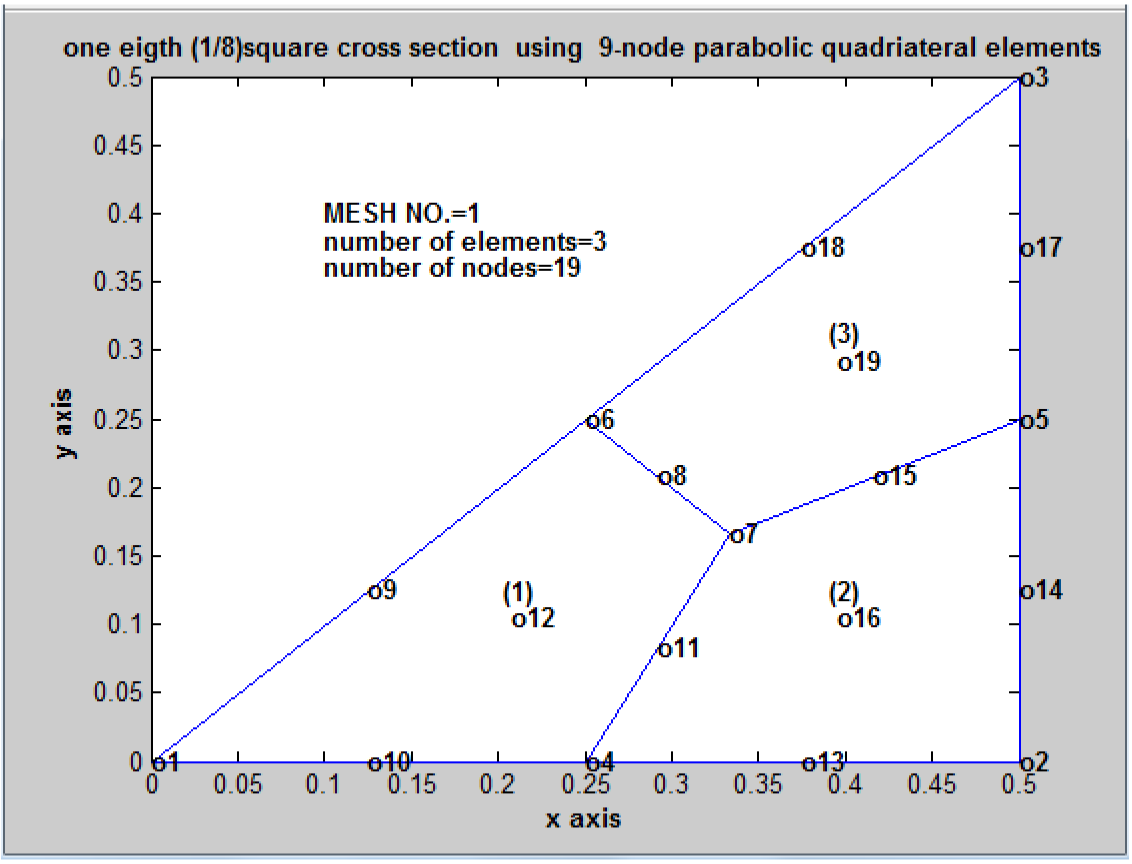


Fig.11 Initial mesh of right isosceles triangle

Example 2: equilateral triangle,each side=2*sqrt(3)
x=sym([-sqrt(3);sqrt(3); 0])
y=sym([-1; -1; 2])
We use this mesh to solve torsion of an arbitrary triangular cross section.This is a case of Poisson equation with constant right hand side(=-1. Our aim are to compute torsional constant and contour lines of Prandtl stress function

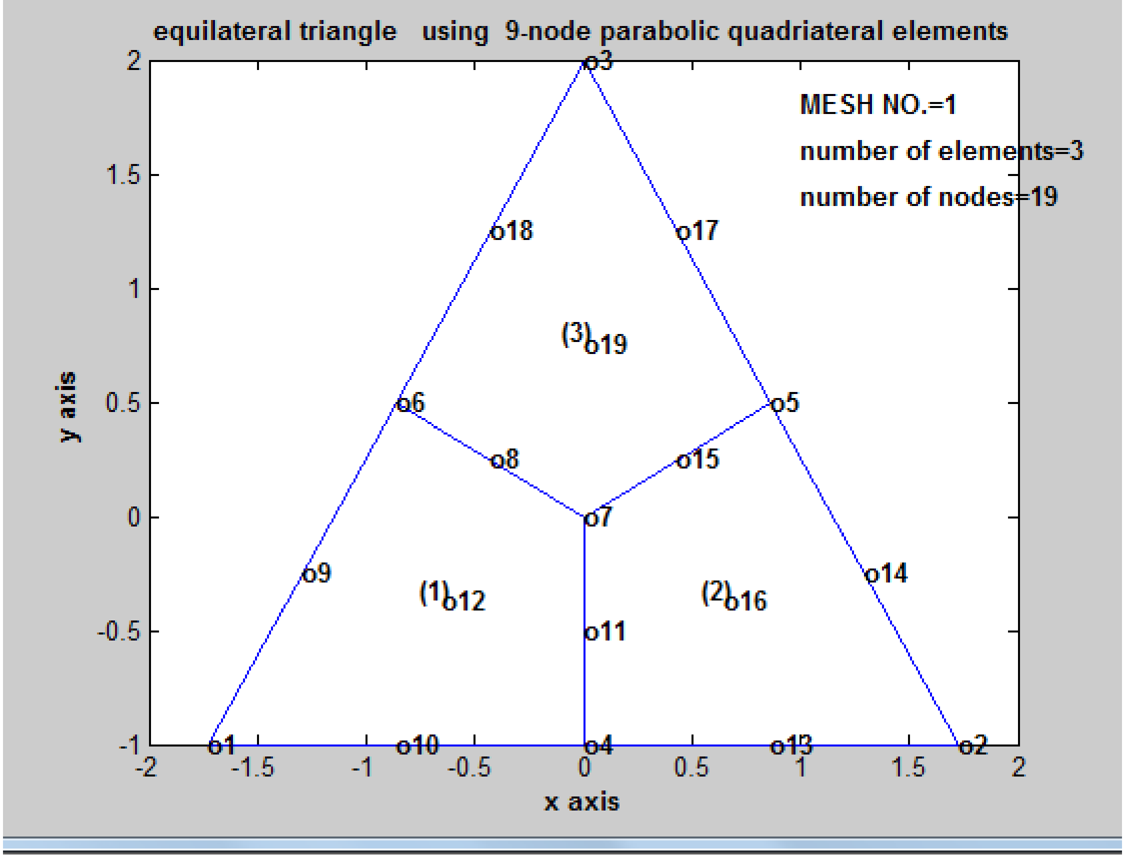


Fig.12 Initial mesh of equilateral triangle,each side=2*sqrt(3)

Example 3 :a square domain with eight triangles(9-boundary nodes)

x=sym([1/2;1/2;1; 1; 1;1/2;0; 0;0])%FOR UNIT SQUARE
y=sym([1/2; 0;0;1/2; 1; 1;1;1/2;0])%FOR UNIT SQUARE
We use this mesh to solve torsion of a square cross section.We would like to draw countour lines of Prandtl stress function over the entire domain.This is a case of Poisson equation with constant right hand side(=-2)

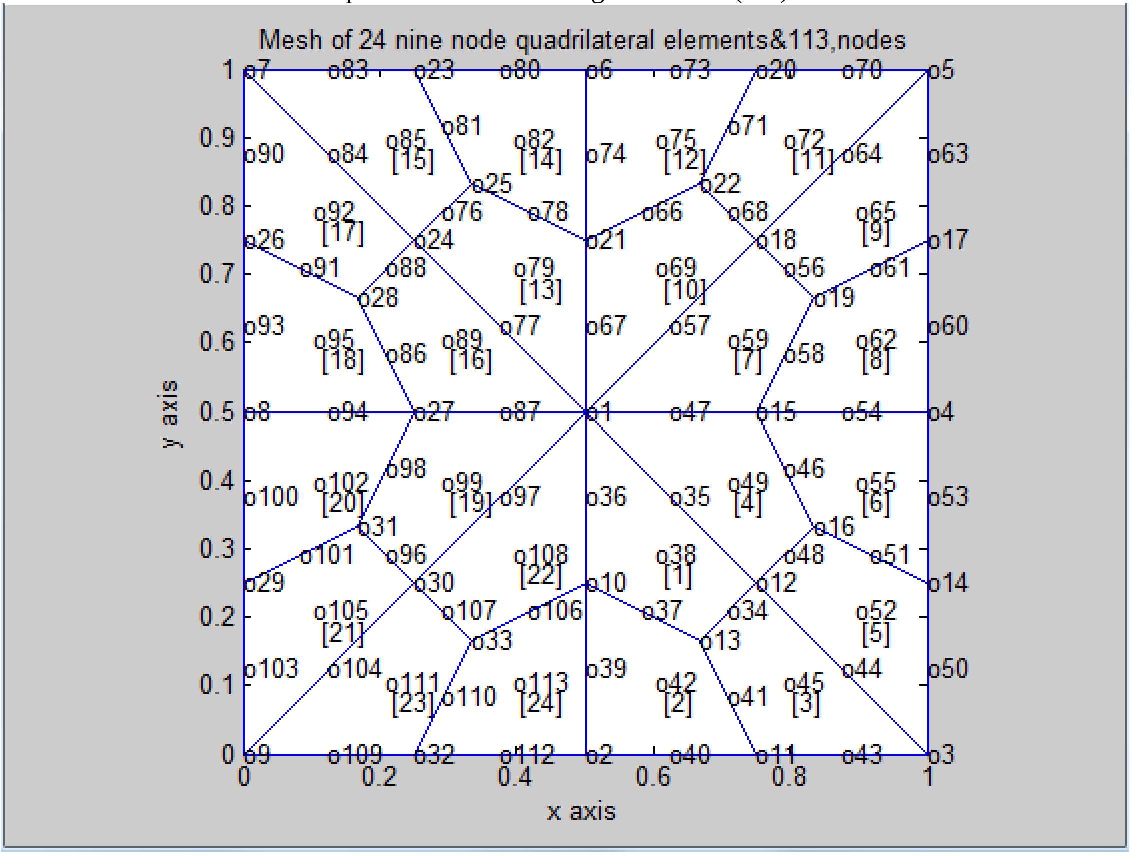


Fig.13 Initial Mesh for a square domain with eight triangles(9-boundary nodes)

Example 4:pentagonal domain with seven triangles(8-boundary nodes)

x=sym([1/2;1/2;1; 1;1/2;0; 0;0])%for MOIN EXAMPLE
y=sym([1/2; 0;0;1/2; 1;1;1/2;0])%for MOIN EXAMPLE
We use this mesh to solve Poisson equation with a nonconstant smooth function on right hand side,with a known analytical solution

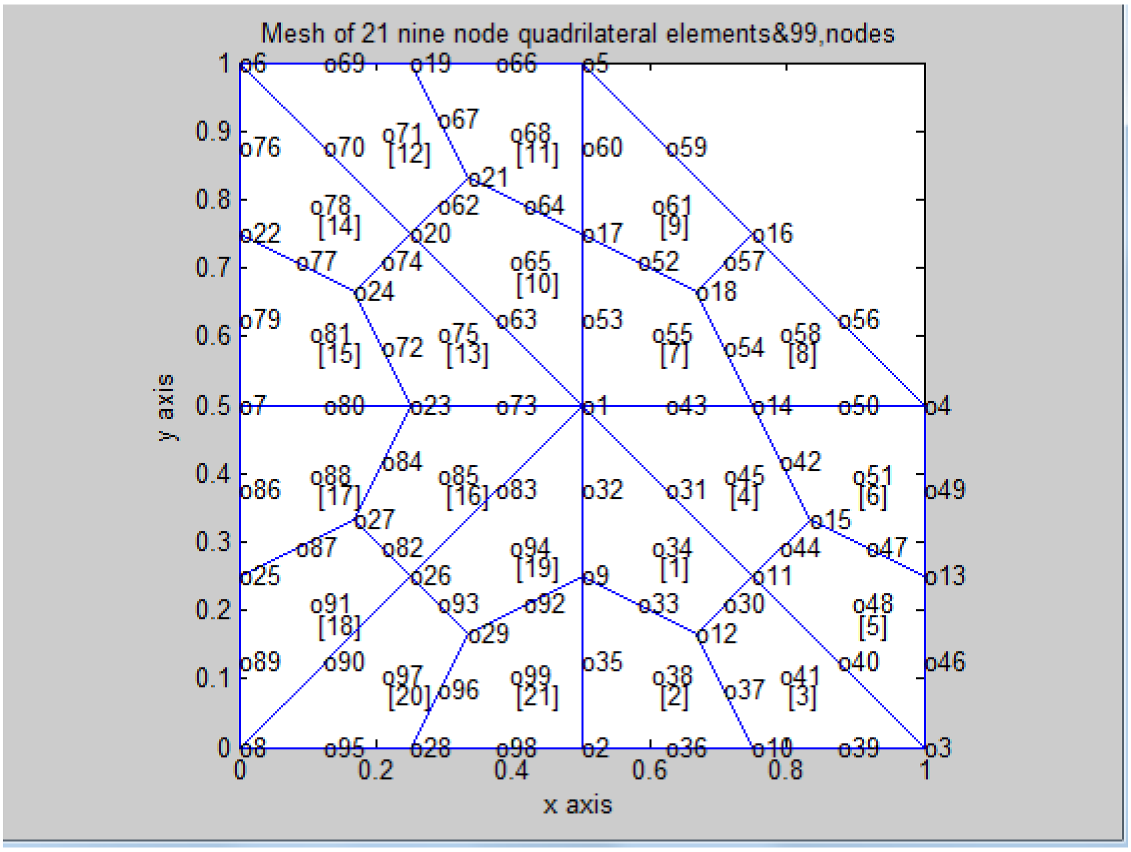


Fig 14:Initial Mesh for a pentagonal domain seven triangles(8-boundary nodes)

Example 5 :a square domain with eight triangles(9-nodes)

x=sym([0;   0; 1/2;1/2;1/2; 0;-1/2;-1/2;-1/2])
y=sym([0;-1/2;-1/2; 0;1/2;1/2; 1/2;   0;-1/2])
We use this mesh to solve torsion of a square cross section.We would like to draw countour lines of Prandtl stress function over the entire domain.This is a case of Poisson equation with constant right hand side(=-2)

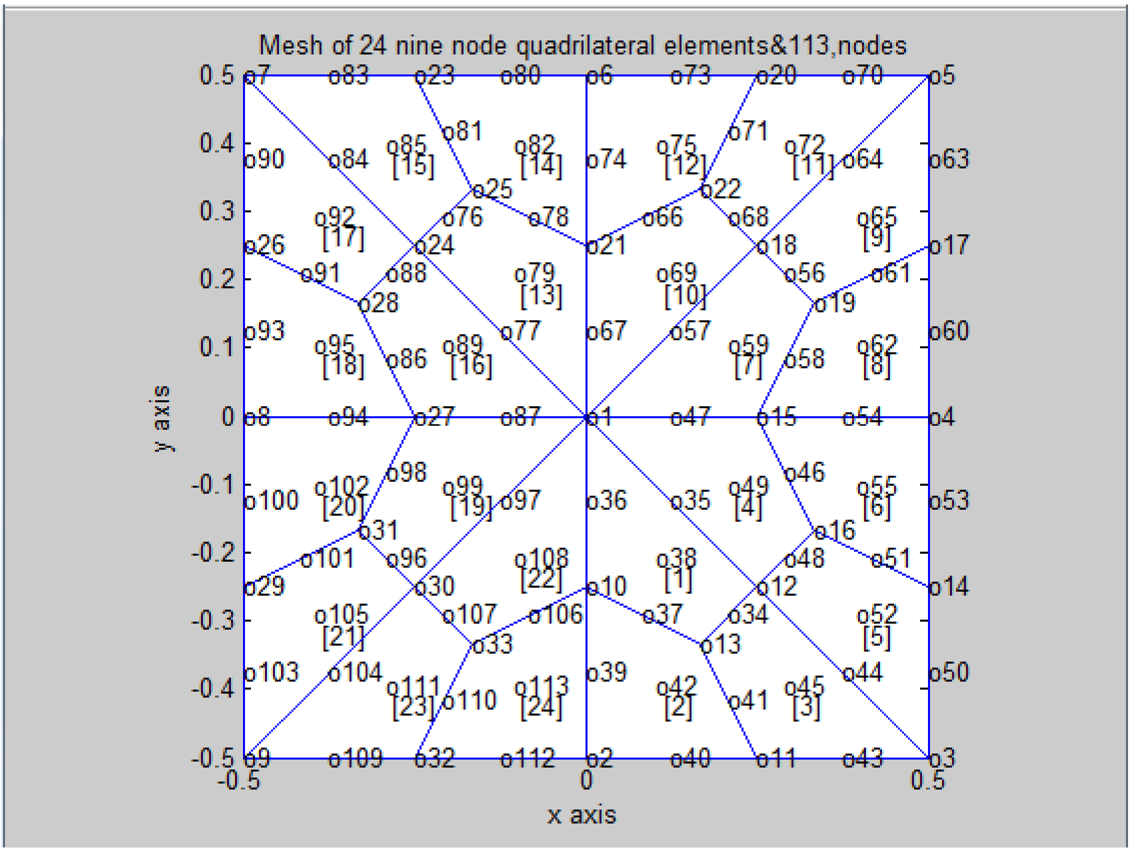


Fig 15: Initial Mesh for a square domain with eight triangles(9-boundary nodes)

7.5 Application Examples

7.5.1 Mesh Generation Over a Standard Triangle & a Square cross sections

Examples 1 &5

Let us use the explicit integration scheme and the auto mesh generation techniques which are developed in the previous sections to solve the Poisson Equation with Dirichlet boundary value problem:

$-\Delta \boldsymbol{u} = \boldsymbol{f},\ \boldsymbol{x} \epsilon \Omega \subset \mathcal{R}^2$ ....(1)

$\boldsymbol{u} = \boldsymbol{g},$ $\boldsymbol{x} \epsilon \partial \Omega$

..(2)

Where Ω is a triangular or polygonal domain and Δ is the standard standard Laplace operator

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 ;

$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + 2G\theta = 0$ in R -------------------- (62a)

$\phi = 0$ on ∂R , the boundary of R --------------------- (62b)

where ϕ(x,y) is known as Prandtl stress function, G is the shear modulus, θ is the angle of twist per unit length, R is the cross sectional region and ∂R is the boundary of R. We choose Gθ = 1 for the sake of simplicity. Then the corresponding torisonal constant is given by the equation

$.t_c = 2 \iint_R \phi(x, y) dx\, dy$ ---------------------------------------(62c)

We take R as the 9-node special quadrilateral meshes described in Examples 1 &5 in previous section.

In a recent paper[26] a new approach to automatic generation of all quadrilateral mesh for finite analysis is proposed and it was applied to discretise the 1/8-th of the square cross section a triangular region into an all quadrilateral mesh. 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 torisonal constant which are expessed in terms of infinite series.This triangular domain is a right isosceles triangle and it was discretised by 8-noded special linear convex quadrilaterals of serendipity family.We have considered this problem again and illustrated the application of 9-node quadrilateral of Lagrange family.

7.5.2 Mesh Generation Over an Arbitrary Triangular Domain

Exmple 2

In applications to boundary value problems due to symmetry considerations or otherwise also, we may have to discretize an arbitrary triangle. Our purpose is to have a code which automatically generates convex quadrangulations of the domain by assuming the input as coordinates of the boundary vertices. We use the theory and procedure developed in section 7.2 and section 7.3 for this purpose..

Let us use the explicit integration scheme and the auto mesh generation techniques which are developed in the previous sections to solve the Poisson Equation with Dirichlet boundary value problem:
$\nabla^2 u = -1,\ x\epsilon\Omega \subset \mathcal{R}^2$
..(9)
$u = 0,$ $x\epsilon\partial\Omega$
..(10)
Where Ω is a regular triangular or polygonal domain and ∇^2 is the standard Laplace operator
Example 2
In this example, we would like to consider the linear elastic torsion of an equilateral triangle which is inscribed in a circle of unit radius.
In our recent we considered torsion of an equilateral triangular cross section and it was discretised by 8-noded special linear convex quadrilaterals of serendipity family.Now we would like to illustrate the t St. Venant Torsion problem for an arbitrary triangular cross section by using 9-node special linear convex quadrilaterals of Lagrange family.

The following MATLAB Codes-I are written for the Poisson Equation with Dirichlet boundary value problem, when the force vector is a constant(either=-1 or=-2)

(1)quadrilateralmesh_over_arbitrarytriangle_q9LGautomeshgen.m
(2)D2LaplaceEquationQ9Ex3automeshgenNew.m
(3)nodaladdresses_special_convex_quadrilaterals_2nd_orderLG.m
(4)coordinate_arbitrarytriangle_2ndorderLAGR.m
(5)coordinate_special_quadrilaterals_in_stdtriangle_2nd_orderLAGR.m
(6)nodaladdresses4Lagrangespecial_convex_quadrilaterals_2nd_order.m
(7)D2LaplaceEquationQ9Ex3automeshgenNewPolygon.m
(8) polygonal_domain_coordinates_2nd_orderLG(see **MATLAB Codes-I I)**
(9) nodaladdresses_special_convex_quadrilaterals_trial_2nd_orderLG.m(see **MATLAB Codes-I I**)

7.5.3 Mesh Generation over a Convex Polygonal Domain

In several physical applications in science and engineering, the boundary value problem require meshes generated over convex polygons. Again our aim is to have a code which automatically generates a mesh of 9 noded convex quadrilaterals of Lagrange family for the complex domains such as those in [27-32]. We use the theory and procedure developed in sections 7.2, 7.3 and 7.4 for this purpose. The following MATLAB codes are written for this purpose.

Example 3(with nonconstant smooth function as right hand side of Poisson equation)

$-\Delta u = 2\pi^2 sin(\pi x)sin(\pi y)\ ,\ (x,y)\epsilon\Omega \subset \mathcal{R}^2$

$u(x,0) = 0, on\ \ y = 0, 0 \leq x \leq 1$

$u(x,1) = 0, on\ \ y = 1, 0 \leq x \leq 1\ ,$

$u(1, y) = 0, on\ x = 1, 0 \leq y \leq 1/2$,

$u(x, y) = sin(\pi x)sin(\pi y)$, **on the line x= 1 - 0.5t ,y=0.5+ 0.5t ,0≤** $t \leq 1$ **......................................(62)**

Where Δ is a standard Laplace operator and Ω is a pentagonal domain joining the vertices {(0,0),(1,0),(1,0.5),(0.5,1),(0,1)}

The exact solution of the above boundary value problem is $u(x, y) = sin(\pi x)sin(\pi y)$.

Example 4(with nonconstant smooth function as right hand side of Poisson equation)

$-\Delta u = 2\pi^2 sin(\pi x)sin(\pi y)$, $(x, y)\epsilon\Omega \subset \mathcal{R}^2$

u =0 , on the boundary $\partial\Omega$ **..(63)**

Where Δ is a standard Laplace operator and Ω is a square domain $[0, 1]^2$ **.**

We have written the following MATLAB Codes-II to solve the Poisson Equations with Dirichlet Boundary Conditions over linear convex polygonal domains when the force vector is a smooth continuous function ($= 2\pi^2 sin(\pi x)sin(\pi y)$)

(1)quadrilateral_mesh4MOINEX_q9LG.m
(2)polygonal_domain_coordinates_2nd_orderLG.m
(3)nodaladdresses_special_convex_quadrilaterals_trial_2nd_orderLG.m
(4)generate_area_coordinate_over_the_standard_triangle.m
(5)glsampleptsweights.m
(6)D2PoissonEquationQ9MoinEx_MeshgridContour.m

8.0 Conclusions

This paper presents the explicit integration scheme for a unique(special) linear convex 9- 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 9- 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 the affine linear transformation from global (x, y) space into a local space (u, v). We then discritise this standard right isosceles triangle in (u, v) space into three unique linear convex 9- node quadrilaterals. We have shown by proving a lemma that any unique linear convex 9-node quadrilateral in (x, y) space can be mapped into one of the unique 9-node quadrilaterals in (u, v) space. We have then mapped these linear convex 9- node quadrilaterals into a 2-square in the local (ξ, η) space by use of the bilinear transformation between (u, v) and (ξ, η) space. Using these two mappings, we have established an integral derivative product relation between the linear convex 9- node quadrilaterals in the global (x ,y) space interior to the arbitrary triangle and the linear convex 9- 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 $P * (K^{i,j,e}) * P^T * (2 *$ area of the arbitrary triangle in (x, y) 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 $(i, j = 1, 2, 3, 4, 5, 6, 7, 8, 9)$. 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, ½), (0, ¼), (0, 0), (1/4, 0), (1/2, 0) , (5/12, 1/6)} and (5/24 ,5/24) 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 a 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 , engineering,medical,business and social sciences. The well known examples are the Laplace and Poisson equations with suitable boundary conditions and the some examples of system of equations are the plane stress, plane strain 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 first demonstrated the proposed explicit integration scheme to solve the St. Venant Torsion problem for an equilateral triangular cross section. Monotonic convergence from below is observed with known analytical solutions for the Prandtl stress function and the torisonal constant. We have demonstrated the proposed explicit integration scheme to solve the Poisson Boundary Value Problem for pentagonal and square domains which are to be considered as simple polygonal domains. Monotonic convergence from below is observed with known analytical solutions for the governing unknown function of Poisson Boundary Value Problem.We have shown the solutions in Tables which list both the FEM and exact solutions.The graphical solutions of nine noded quadrilateral meshes and contour level curves for FEM and exact solutions are also displayed. We conclude that efficient scheme on explicit integration of stiffness matrix and a novel automesh generation technique developed in this paper will be useful for the solution of many physical problems governed by second order partial differential equations.

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 with fast convergence and economy for the computational problems.

REFERENCES:

[1] Zienkiewicz O.C, Taylor R.L and Zhu J.Z Finite Element Method, its basis and fundamentals, Elservier, (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, 9th 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 ∨ , 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, 3rd 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, 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)

[27]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)

[28] Rathod.H.T, Bharath Rathod, Shivaram K.T , H. Y. Shrivalli , Tara Rathod , K. Sugantha DeviAn explicit finite element integration scheme using automatic mesh generation technique for linear convex quadrilaterals over plane regions , international Journal of Engineering and Computer Science, Vol. 3,issue 4,pp5400-5435 (2014)

[29] Rathod.H.T, Bharath Rathod, K.T.Shivaram,Sugantha Devi.K,Tara Rathod An explicit finite element integration scheme for linear eight node convex quadrilaterals using automatic mesh generation technique over plane regions**,** international Journal of Engineering and Computer Science, Vol. 3,issue 4,pp5657-5713 (2014)

[30] Rathod.H.T, Sugantha Devi.K ,Finite element solution of Poisson equation over polygonal domains using an explicit integration scheme and a novel auto mesh generation technique,,International Journal of Engineering and Computer Science, ISSN:2319-7242,Volume(5),issue 8,August(2016), pp. 17397-17481

[31] Rathod.H.T, Sugantha Devi.K,Nagabhushana.C.S , Chudamani.H.M ,Finite element analysis of linear elastic torsion for regular polygons, International Journal of Engineering and Computer Science, ISSN:2319-7242,Volume(5),issue 10,October(2016), pp. 18413-1842

[32] Rathod.H.T, Bharath Rathod, Sugantha Devi.K,Hariprasad.A.S, Finite element solution of Poisson Equation over Polygonal Domains using a novel auto mesh generation technique and an explicit integration scheme for eight node linear convex quadrilaterals, International Journal of Engineering and Computer Science, ISSN:2319-7242, Volume (6) Issue 10 October 2017, pp 22632-22787

APPENDIX-1

We now propose to compute the following integrals(see eqns(52a-b)) and they will be listed in Tables 1a-1i

$$\mathbf{K}^{\mathbf{i,j,e}} = \iint_{\hat{Q}} \mathbf{G}^{\mathbf{i,j,e}}_{\mathbf{u\,,v}} \ \mathbf{du\,dv} = \begin{pmatrix} \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial u}\frac{\partial N_j^e}{\partial u}\,\mathbf{dudv} & \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial u}\frac{\partial N_j^e}{\partial v}\,\mathbf{dudv} \\ \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial v}\frac{\partial N_j^e}{\partial u}\,\mathbf{dudv} & \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial v}\frac{\partial N_j^e}{\partial v}\,\mathbf{dudv} \end{pmatrix} = \begin{pmatrix} K^e_{2i-1\,,2j-1} & K^e_{2i-1\,,2j} \\ K^e_{2i\,,2j-1} & K^e_{2i\,,2j} \end{pmatrix} \text{(say)} \qquad \text{-----------}$$

(52a-b)

We map $\hat{Q}$ into a 2-square We map $\hat{Q}$ into a $2-$ square, $-1 \le \xi,\eta \le 1$

in the natural parametric space (ξ,η) by using the bilinear transformation from (u,v) to (ξ,η)

This gives(see eqns(18-20))

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

Let us replace the Greek letters $\boldsymbol{\xi},\boldsymbol{\eta}$ by English letters **r,s** for computing the integrals by using MATLAB programming.With this assumption,we denote the entries of submatrix $\mathbf{K}^{\mathbf{i,j,e}}$ **= intJdnidnjuvrs ,(i,j=1(1)9)**

and we have from eqn(52a-b):

$$\textbf{intJdnidnjuvrs=}\begin{pmatrix} \textbf{intJdnidnjuvrs}(1,1) & \textbf{intJdnidnjuvrs}(1,2) \\ \textbf{intJdnidnjuvrs}(2,1) & \textbf{intJdnidnjuvrs}(2,2) \end{pmatrix}$$

$$K^e_{2i-1\,,2j-1} = \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial u}\frac{\partial N_j^e}{\partial u}\,\mathbf{dudv} = \textbf{intJdnidnjuvrs}(1,1),\ K^e_{2i-1\,,2j} = \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial u}\frac{\partial N_j^e}{\partial v}\,\mathbf{dudv} = \textbf{intJdnidnjuvrs}(1,2),$$

$$K^e_{2i\,,2j-1} = \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial v}\frac{\partial N_j^e}{\partial u}\,\mathbf{dudv} = \textbf{intJdnidnjuvrs}(2,1),\ K^e_{2i\,,2j} = \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial v}\frac{\partial N_j^e}{\partial v}\,\mathbf{dudv} = \textbf{intJdnidnjuvrs}(2,2)$$

Tables 1a-1i

Table-1a

(i=1,j=1(1)9)

```
intJdn1dn1uvrs =[ 0.68800369699992917040751928886589, 0.63026863732902919654693031749554;...
                  0.63026863732902919654693031749554, 0.68800369699992917040751928886589]
intJdn1dn2uvrs = [ 0.049469766679465431322423128490783, -0.092492499790138905674372588717718;...
                  0.074174166876527760992294077948948, -0.058041502241256898516961218298311]
 intJdn1dn3uvrs = [ -0.027639187071864196478244612691051, -0.036406227705135716084660539278176;...
                  -0.036406227705135716084660539278176, -0.027639187071864196478244612691051]
intJdn1dn4uvrs = [ -0.058041502241256898516961218298311, 0.074174166876527760992294077948948;...
                  -0.092492499790138905674372588717718, 0.049469766679465431322423128490783]
intJdn1dn5uvrs =[ -0.42034020155248898256147959938904, 0.23835534506556861836117847035703;...
                  -0.42831132160109804830548819630964, 0.10203781293970562330870803190571]
intJdn1dn6uvrs =[  0.12626870005164588340746772851 13, 0.16865169113202404137334892155088;...
                  0.16865169113202404137334892155088, 0.15224655673422927054938212412807]
intJdn1dn7uvrs =[ 0.15224655673422927054938212412807, 0.16865169113202404137334892155088;...
                  0.16865169113202404137334892155088,  0.12626870005164588340746772851 13]
intJdn1dn8uvrs =[ 0.10203781293970562330870803190571, -0.42831132160109804830548819630964;...
                  0.23835534506556861836117847035703, -0.42034020155248898256147959938904]
intJdn1dn9uvrs =[ -0.61200564253936530143881487152335, -0.72289148243880098858257938459775;...
                  -0.72289148243880098858257938459775, -0.61200564253936530143881487152335]
```

Table-1b

(i=2,j=1(1)9)

```
intJdn2dn1uvrs =[  0.049469766679465431322423128490783,  0.074174166876527760992294077948948;...
                  -0.092492499790138905674372588717718, -0.058041502241256898516961218298311]
intJdn2dn2uvrs =[  0.24822242723570359859415910167674, -0.13943577781062161645243966285652;...
                 -0.13943577781062161645243966285652,  0.22525137401606776447455758470085]
intJdn2dn3uvrs =[ -0.093683290648218768462051538677453, -0.082852283681797324065968279843 49;...
                   0.083814382984869342600698386823177, 0.037904800898126436475032582647755]
intJdn2dn4uvrs =[ 0.00062925700025302140451751959728441,   0.026100052047122979851449223466987;...
                  0.026100052047122979851449223466987, 0.00062925700025302140451751959728447]
intJdn2dn5uvrs =[ -0.46868216752790132335135919627624,  -0.34646194452109534302782806359 57;...
                  0.32020472214557132363883860307097, 0.085836729209541381287229242816599]
intJdn2dn6uvrs =[  0.20486711030201898764584585313023,  0.25718774442435573105221634721679;...
                   -0.40947892224231093561445031944987, -0.28073862677651621463393803503496]
intJdn2dn7uvrs =[ 0.050885917641966498462888866392175,   -0.10335794828229434487750587372274;...
                  -0.10335794828229434487750587372274, -0.0017609016019421764322513358165054]
intJdn2dn8uvrs =[ -0.041736856619169902011341267601507, -0.094765689307712237483661393080961;...
                   -0.094765689307712237483661393080961,   0.033881015844374607003400451583 06]
```

```
intJdn2dn9uvrs =[ 0.050027835935882456394917533267998,   0.40941168025551439401144362446668;...
                   0.40941168025551439401144362446668, -0.042962146348647921061586792195777]
```

Table-1c

(i=3,j=1(1)9)

```
intJdn3dn1uvrs =[ -0.027639187071864196478244612691051, -0.036406227705135716084660539278176]
                  -0.036406227705135716084660539278176, -0.027639187071864196478244612691051;...
intJdn3dn2uvrs =[ -0.093683290648218768462051538677453, 0.083814382984869342600698386823177]
                   -0.08285228368179732406596827984349, 0.037904800898126436475032582647755;...
intJdn3dn3uvrs =[ 0.33203856905137126709551224250652,  0.29812794600013651335576030277 24;...
                   0.29812794600013651335576030277 24, 0.33203856905137126709551224250652]
intJdn3dn4uvrs =[ 0.037904800898126436475032582647755,  -0.082852283681797324065968279843 49;...
                  0.083814382984869342600698386823177, -0.093683290648218768462051538677453]
intJdn3dn5uvrs =[ 0.15710844821544667058002407563081,  0.12945549547911074908308207666547;...
                   0.12945549547911074908308207666547, 0.043059644229262196507731226976332]
intJdn3dn6uvrs =[ 0.14159217838663875388665351330935, -0.35360191630747700837476451284707;...
                   0.31306475035918965829190215381 96, -0.34086853405320051524855521401992]
intJdn3dn7uvrs =[ -0.34086853405320051524855521401992,  0.31306475035918965829190215381 96;...
                   -0.35360191630747700837476451284707, 0.14159217838663875388665351330935]
intJdn3dn8uvrs =[ 0.043059644229262196507731226976332, 0.12945549547911074908308207666547;...
                    0.12945549547911074908308207666547, 0.15710844821544667058002407563081]
intJdn3dn9uvrs =[ -0.24951262900756184435610227568234, -0.48105764260800696388913166477738;...
                   -0.48105764260800696388913166477738, -0.24951262900756184435610227568234]
```

Table-1d

(i=4,j=1(1)9)

```
intJdn4dn1uvrs =[ -0.058041502241256898516961218298311, -0.092492499790138905674372588717718;...
                    0.074174166876527760992294077948948,  0.049469766679465431322423128490783]
intJdn4dn2uvrs =[ 0.00062925700025302140451751959728441,   0.026100052047122979851449223466987;...
                    0.026100052047122979851449223466987, 0.00062925700025302140451751959728447]
intJdn4dn3uvrs =[ 0.037904800898126436475032582647755,  0.083814382984869342600698386823177;...
                  -0.082852283681797324065968279843 49, -0.093683290648218768462051538677453]
intJdn4dn4uvrs =[  0.22525137401606776447455758470085, -0.13943577781062161645243966285652;...
                   -0.13943577781062161645243966285652,  0.24822242723570359859415910167674]
intJdn4dn5uvrs =[   0.033881015844374607003400451583 06, -0.094765689307712237483661393080961;...
                   -0.094765689307712237483661393080961, -0.041736856619169902011341267601507]
intJdn4dn6uvrs =[ -0.0017609016019421764322513358165055, -0.10335794828229434487750587372274;...
                     -0.10335794828229434487750587372274, 0.050885917641966498462888866392175]
intJdn4dn7uvrs =[ -0.28073862677651621463393803503496, -0.40947892224231093561445031944987;...
                    0.25718774442435573105221634721679,  0.20486711030201898764584585313023]
intJdn4dn8uvrs =[ 0.085836729209541381287229242816599,  0.32020472214557132363883860307097;...
                   -0.34646194452109534302782806359 57, -0.46868216752790132335135919627624]
intJdn4dn9uvrs =[ -0.042962146348647921061586792195777,  0.40941168025551439401144362446668;...
                     0.40941168025551439401144362446668, 0.050027835935882456394917533267999]
```

Table-1e

(i=5,j=1(1)9)

```
intJdn5dn1uvrs =[ -0.42034020155248898256147959938904, -0.42831132160109804830548819630964;...
                    0.23835534506556861836117847035703,  0.10203781293970562330870803190571]
intJdn5dn2uvrs =[ -0.46868216752790132335135919627624,  0.32020472214557132363883860307097;...
                    -0.34646194452109534302782806359 57, 0.085836729209541381287229242816599]
intJdn5dn3uvrs =[ 0.15710844821544667058002407563081,  0.12945549547911074908308207666547;...
                    0.12945549547911074908308207666547, 0.043059644229262196507731226976332]
intJdn5dn4uvrs =[   0.033881015844374607003400451583 06, -0.094765689307712237483661393080961;...
                    -0.094765689307712237483661393080961, -0.041736856619169902011341267601507]
intJdn5dn5uvrs =[  1.0663384833344564856750376651589, 0.56556859863119073241392390206822;...
                   0.56556859863119073241392390206822,  1.1211333295329756651950424277428]
intJdn5dn6uvrs =[ -0.58306188548401601151324941680212, -0.54252818687130357695788942931127;...
                   -0.54252818687130357695788942931127, -0.22579554943131879593427777747996]
intJdn5dn7uvrs =[   -0.16767681050348731002396257406 51, -0.0017096478113284630633060979019 75;...
                  -0.0017096478113284630633060979019 75,   0.17132457681637017904834081106242]
intJdn5dn8uvrs =[ 0.090271486053820367920675971147722,  0.41024643803333829631989611851833;...
                   0.41024643803333829631989611851833, 0.090271486053820367920675971147722]
intJdn5dn9uvrs =[  0.29216163161979549627091262301201, -0.35816040869776877564539558371915;...
                  -0.35816040869776877564539558371915,  -1.3461311727311867153221086665701]
```

Table-1f

(i=6,j=1(1)9)

```
intJdn6dn1uvrs =[  0.1262687000516458834074677285113, 0.16865169113202404137334892155088]
                    0.16865169113202404137334892155088, 0.15224655673422927054938212412807;...
intJdn6dn2uvrs =[ 0.20486711030201898764584585313023, -0.40947892224231093561445031944987]
                   0.25718774442435573105221634721679, -0.28073862677651621463393803503496;...
intJdn6dn3uvrs =[  0.14159217838663875388665351330935,   0.3130647503591896582919021538196]
                   -0.35360191630747700837476451284707, -0.34086853405320051524855521401992;...
intJdn6dn4uvrs =[ -0.0017609016019421764322513358165055, -0.10335794828229434487750587372274]
                   -0.10335794828229434487750587372274, 0.050885917641966498462888866392175;...
intJdn6dn5uvrs =[ -0.58306188548401601151324941680212, -0.54252818687130357695788942931127;...
                    -0.54252818687130357695788942931127, -0.22579554943131879593427777747996]
intJdn6dn6uvrs =[ 1.5998510858095308546375928228576,   0.244693272250901099049116631754;...
                    0.244693272250901099049116631754, 0.64675514174853057100283264582228]
intJdn6dn7uvrs =[ -0.074612686016266341101136360446836,   0.42496611860422613701397774222377;...
                    0.42496611860422613701397774222377, -0.074612686016266341101136360446836]
intJdn6dn8uvrs =[   0.17132457681637017904834081106242, -0.001709647811328463063306097901975;...
                  -0.001709647811328463063306097901975,   -0.1676768105034873100239625740651]
intJdn6dn9uvrs =[  -1.5844681782639801295792636158054, -0.09430112713910361521519372896239;...
                  [ -0.09430112713910361521519372896239,  0.23980459065606283692676632470425]
```

Table-1g

(i=7,j=1(1)9)

```
intJdn7dn1uvrs =[ 0.15224655673422927054938212412807, 0.16865169113202404137334892155088;...
                    0.16865169113202404137334892155088,  0.1262687000516458834074677285113]
intJdn7dn2uvrs =[ 0.050885917641966498462888866392175,   -0.10335794828229434487750587372274;...
                   -0.10335794828229434487750587372274, -0.0017609016019421764322513358165054]
intJdn7dn3uvrs =[ -0.34086853405320051524855521401992, -0.35360191630747700837476451284707;...
                    0.3130647503591896582919021538196,  0.14159217838663875388665351330935]
intJdn7dn4uvrs =[ -0.28073862677651621463393803503496, 0.25718774442435573105221634721679;...
                   -0.40947892224231093561445031944987, 0.20486711030201898764584585313023]
intJdn7dn5uvrs =[   -0.1676768105034873100239625740651, -0.001709647811328463063306097901975;...
                   -0.001709647811328463063306097901975,   0.17132457681637017904834081106242]
intJdn7dn6uvrs =[ -0.074612686016266341101136360446836,   0.42496611860422613701397774222377;...
                    0.42496611860422613701397774222377, -0.074612686016266341101136360446836]
intJdn7dn7uvrs =[ 0.64675514174853057100283264582228,  0.244693272250901099049116631754;...
                    0.244693272250901099049116631754, 1.5998510858095308546375928228576]
intJdn7dn8uvrs =[ -0.22579554943131879593427777747996, -0.54252818687130357695788942931127;...
                    -0.54252818687130357695788942931127, -0.58306188548401601151324941680212]
intJdn7dn9uvrs =[   0.23980459065606283692676632470425, -0.094301127139103615215193728962391;...
                   -0.094301127139103615215193728962391,   -1.5844681782639801295792636158054]
```

Table-1h

(i=8,j=1(1)9)

```
intJdn8dn1uvrs =[  0.10203781293970562330870803190571,  0.23835534506556861836117847035703;...
                   -0.42831132160109804830548819630964, -0.42034020155248898256147959938904]
intJdn8dn2uvrs =[ -0.041736856619169902011341267601507, -0.094765689307712237483661393080961;...
                    -0.094765689307712237483661393080961,   0.03388101584437460700340045158306]
intJdn8dn3uvrs =[ 0.043059644229262196507731226976332, 0.12945549547911074908308207666547;...
                    0.12945549547911074908308207666547, 0.15710844821544667058002407563081]
intJdn8dn4uvrs =[ 0.085836729209541381287229242816599,  -0.3464619445210953430278280635957;...
                    0.32020472214557132363883860307097, -0.46868216752790132335135919627624]
intJdn8dn5uvrs =[ 0.090271486053820367920675971147722,  0.41024643803333829631989611851833;...
                    0.41024643803333829631989611851833, 0.090271486053820367920675971147722]
intJdn8dn6uvrs =[   0.17132457681637017904834081106242, -0.001709647811328463063306097901975;...
                   -0.001709647811328463063306097901975,   -0.1676768105034873100239625740651]
intJdn8dn7uvrs =[ -0.22579554943131879593427777747996, -0.54252818687130357695788942931127;...
                   -0.54252818687130357695788942931127, -0.58306188548401601151324941680212]
intJdn8dn8uvrs =[  1.1211333295329756651950424277428, 0.56556859863119073241392390206822;...
                     0.56556859863119073241392390206822,  1.0663384833344564856750376651589]
intJdn8dn9uvrs =[  -1.3461311727311867153221086665701, -0.35816040869776877564539558371915;...
                    -0.35816040869776877564539558371915,  0.29216163161979549627091262301201]
```

Table-1i

(i=9,j=1(1)9)

```
intJdn9dn1uvrs =[ -0.61200564253936530143881487152335, -0.72289148243880098858257938459775;...
                  -0.72289148243880098858257938459775, -0.61200564253936530143881487152335]
intJdn9dn2uvrs =[ 0.050027835935882456394917533267998,   0.40941168025551439401144362446668;...
                   0.40941168025551439401144362446668, -0.042962146348647921061586792195777]
intJdn9dn3uvrs =[ -0.24951262900756184435610227568234, -0.48105764260800696388913166477738;...
                  -0.48105764260800696388913166477738, -0.24951262900756184435610227568234]
intJdn9dn4uvrs =[ -0.042962146348647921061586792195777,  0.40941168025551439401144362446668;...
                   0.40941168025551439401144362446668, 0.050027835935882456394917533267999]
intJdn9dn5uvrs =[  0.29216163161979549627091262301201, -0.35816040869776877564539558371915;...
                  -0.35816040869776877564539558371915,  -1.3461311727311867153221086665701]
intJdn9dn6uvrs =[  -1.5844681782639801295792636158054, -0.09430112713910361521519372896239;...
                  -0.09430112713910361521519372896239,  0.23980459065606283692676632470425]
intJdn9dn7uvrs =[   0.23980459065606283692676632470425, -0.094301127139103615215193728962391;...
                  -0.094301127139103615215193728962391,   -1.5844681782639801295792636158054]
intJdn9dn8uvrs =[  -1.3461311727311867153221086665701, -0.35816040869776877564539558371915;...
                  -0.35816040869776877564539558371915,  0.29216163161979549627091262301201]
intJdn9dn9uvrs =[ 3.2530857106790011221652797407927, 1.2900488362095239461700024258049;...
                  1.2900488362095239461700024258049, 3.2530857106790011221652797407927]
```

SOME SAMPLE RESULTS (TABLES& FIGURES)

Table 2a(PRESENT PAPER)

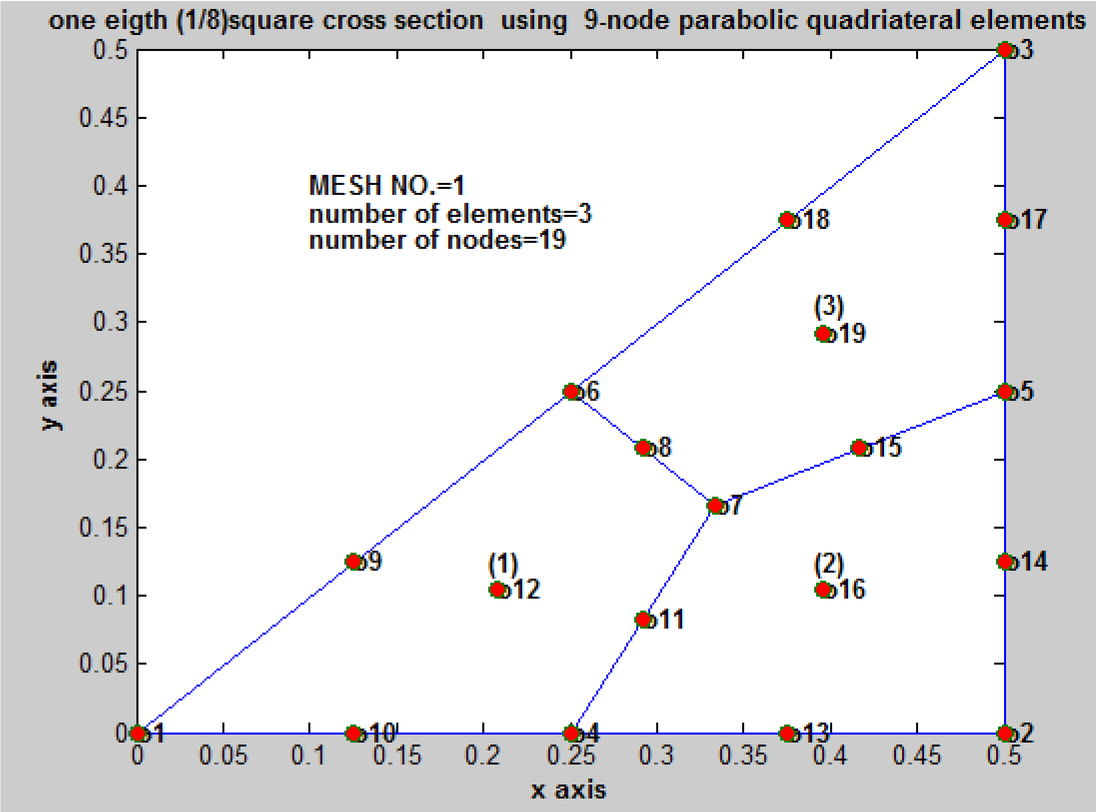
torsion of a square cross section modeled as a right isoscles triangle

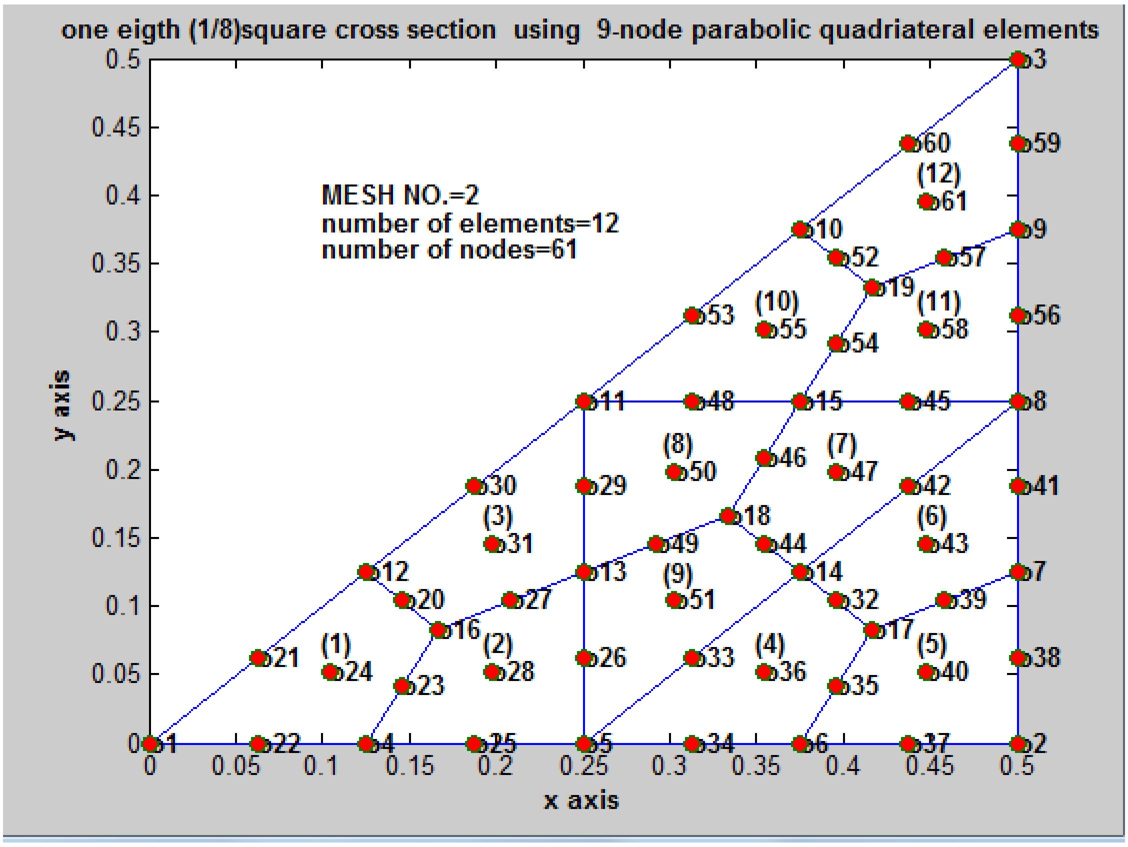
nine node -linear convex quadrilaterals(lagrange family elememts)

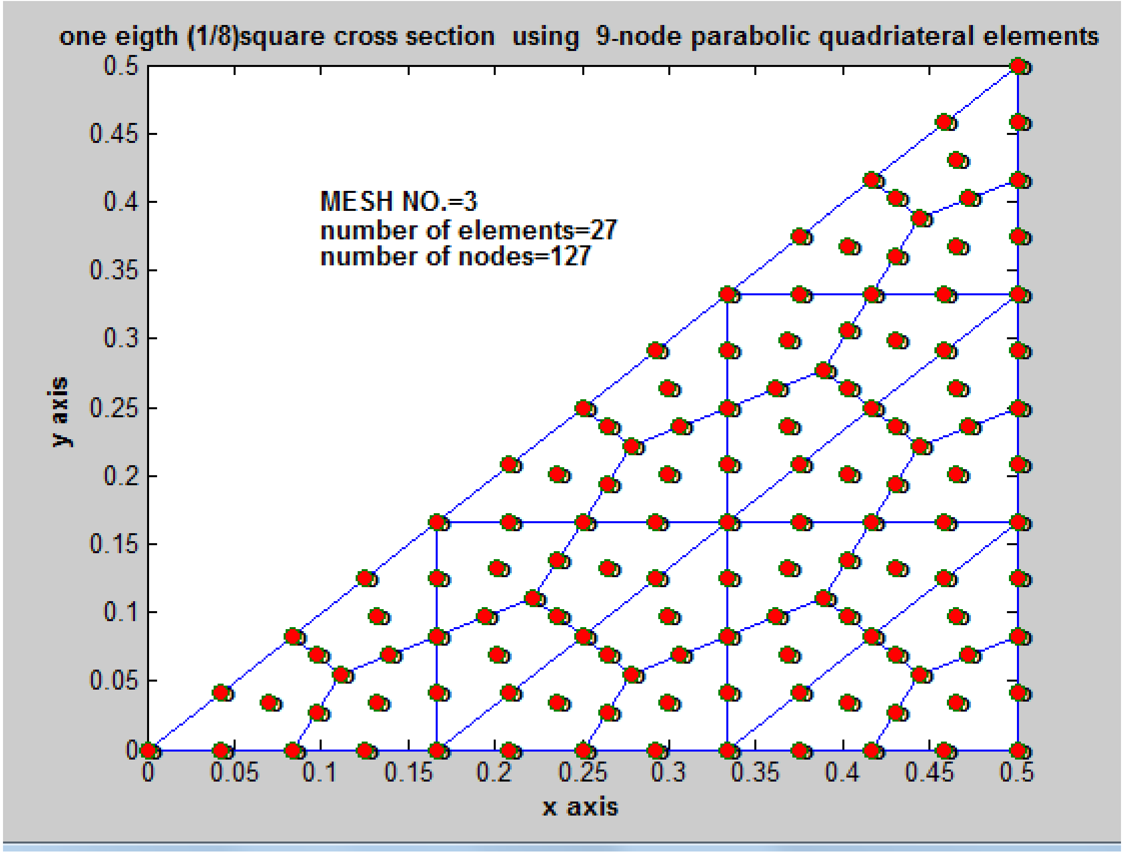
(exact solution of torisonal constant= 0.140577014955156)

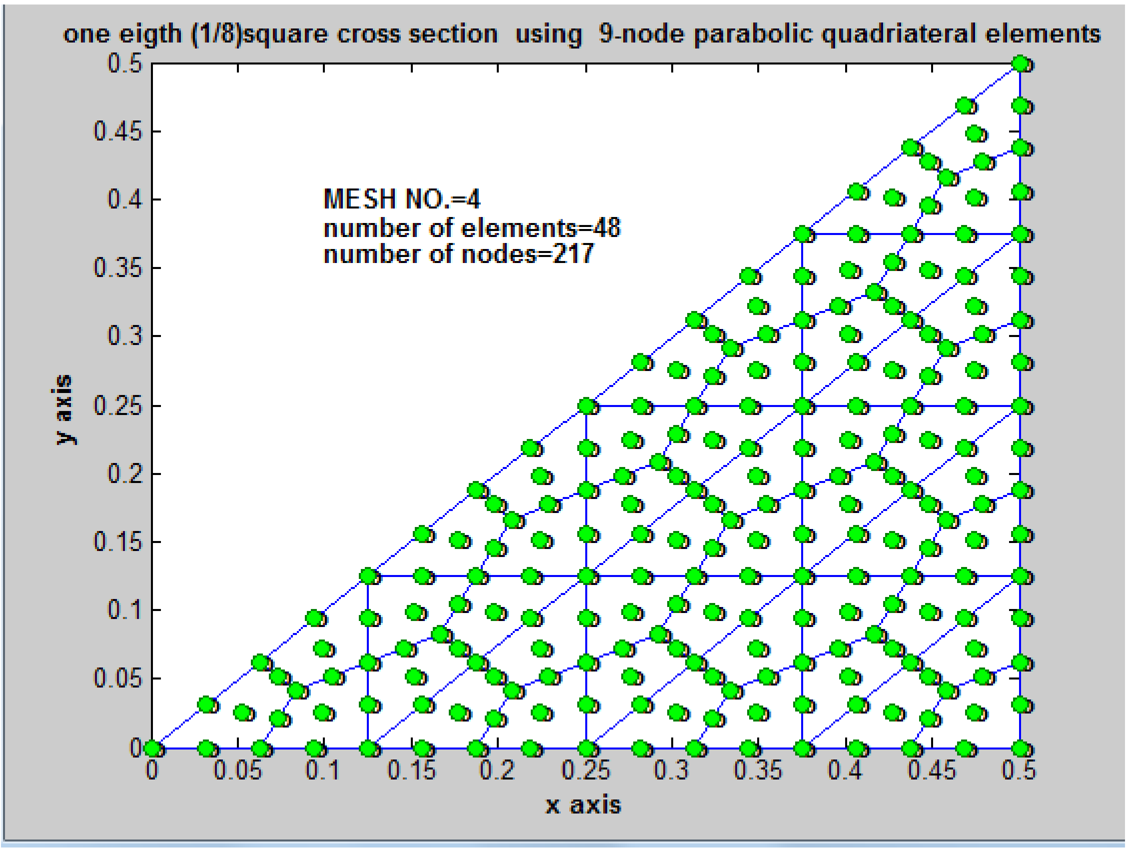
nnode=number of nodes in theregion R ; nel=number of elements in theregion R;

Mesh No	nnode	nel	FEM solution for Torisonal constant	maximum absolute error of Prandtl Stress function values at element nodes
(1)	19	3	0.140226269123952	0.0011496140422405
(2)	61	12	0.140549995991093	0.000283838209277359
(3)	127	27	0.140571151900467	0.000125431921097136
(4)	217	48	0.140575044191592	7.22896125021954e-005
(5)	331	75	0.140576171269977	4.498537060151e-005
(6)	469	108	0.140576593774739	2.98546615531262e-005
(7)	631	147	0.140576781096277	2.23280855678916e-005
(8)	817	192	0.140576874567466	1.8370913205128e-005
(9)	1027	243	0 .140576925494304	1.58338405955379e-005
(10)	1261	300	0.140576955193951	1.38273942157716e-005


one eigth (1/8)square cross section using 9-node parabolic quadriateral elements
MESH NO.=1
number of elements=3
number of nodes=19
y axis
x axis
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
(1)
(2)
(3)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19


one eigth (1/8)square cross section using 9-node parabolic quadriateral elements
MESH NO.=2
number of elements=12
number of nodes=61
y axis
x axis
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61


one eigth (1/8)square cross section using 9-node parabolic quadriateral elements
MESH NO.=3
number of elements=27
number of nodes=127
y axis
x axis
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5


one eigth (1/8)square cross section using 9-node parabolic quadriateral elements
MESH NO.=4
number of elements=48
number of nodes=217
y axis
x axis
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5

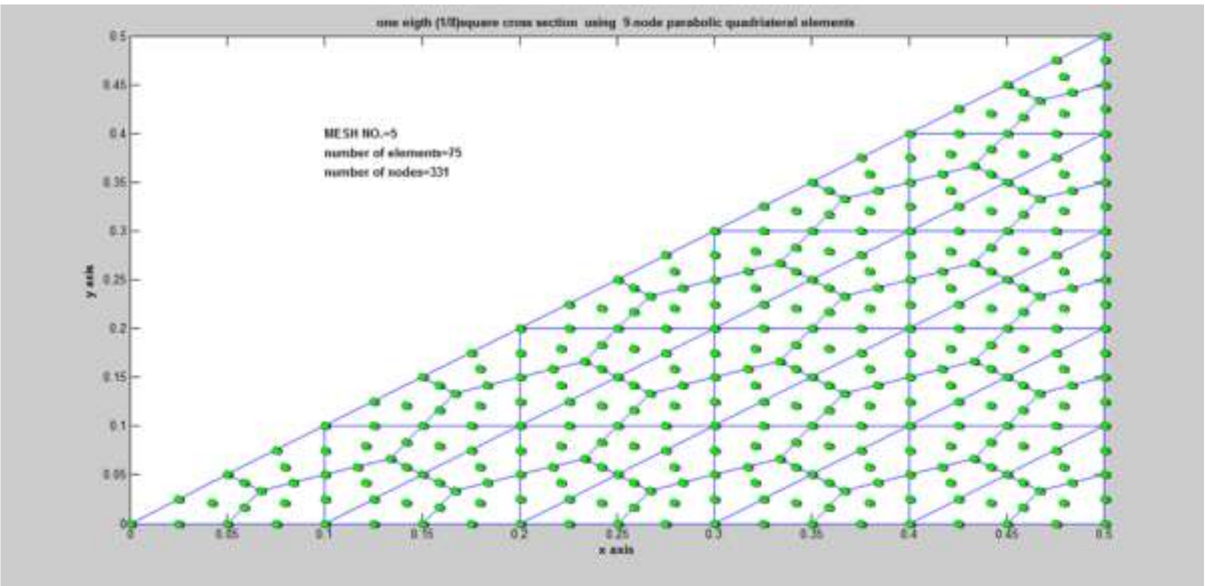


Table-2b

TORSION OF A SQUARE CROSS SECTION MODELED AS A RIGHT ISOSCLES TRIANGLE

EIGHT NODE -LINEAR CONVEX QUADRILATERALS(SERENDIPITY ELEMEMTS)

(exact solution of torisonal constant= 0.140577014955156)

nnode=number of nodes in theregion R ; nel=number of elements in theregion R;

(mesh no.)	nnode	nel	FEM Solution for torisonal constant REFERENCE[32]	maximum absolute error of Prandtl Stress function values at element nodes
(1)	16	3	0.139881192455598	0.00172214821773672
(2)	49	12	0.140511125235056	0.000228103295411064
(3)	100	27	0.140560146231912	0.000111318093756023
(4)	169	48	0.14057033232108	7.57868365891083e-005
(5)	256	75	0.14057364619955	5.71558452171442e-005
(6)	361	108	0.14057504105108	4.25097858819104e-005
(7)	484	147	0.14057573511073	3.45767462262801e-005
(8)	625	192	0.140576123341783	2.94391901602548e-005
(9)	784	243	0.140576360001611	2.42499917089884e-005
(10)	961	300	0.140576514028352	1.94431366115373e-005

(11)	1156	363	0.140576619547978	1.57123956216918e-005
(12)	1369	432	0.14057669486575	1.31847331217476e-005
(13)	1600	507	0.140576750456548	1.16333370254527e-005
(14)	1849	588	0.14057679264143	1.07440118374911e-005
(15)	2116	675	0.140576825408248	1.02449849120308e-005
(16)	2401	768	0.140576851369049	9.94170594790372e-006
(17)	2704	867	0.140576872290657	9.71043506567621e-006
(18)	3025	972	0.140576889401995	9.48027086123348e-006
(19)	3364	1083	0.140576903578949	9.21569212390368e-006
(20)	3721	1200	0.140576915459166	8.90311431203444e-006
(21)	4096	1323	0.140576925515857	8.54159601418455e-006
(22)	4489	1452	0.140576934105913	8.13682739378169e-006

Table-2c

TORSION OF A SQUARE CROSS SECTION MODELED AS A RIGHT ISOSCLES TRIANGLE

FOUR NODE-LINEAR CONVEX QUADRILATERALS {COMPUTED FROM REFERENCE[28])

(exact solution of torisonal constant= 0.140577014955156)

nnode=number of nodes in the region R ;

nel=number of elements in the region R;

			FOR FOUR NODE QUADRILATERAL MESHES	
nnode	nel	nnel	fem solution for torisonal constant] REFERENCE[28]	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-3a(Present Work)

TORSION OF AN EQUILATERAL TRIANGULAR CROSS SECTION(NINE NODED ELEMENTS)

Mesh No	nodes	elements	TORISONAL CONSTANT VALUES FEM SOL	EXACT SOL	maximum absolute error of Prandtl Stress function values at element nodes
1	**19**	**3**	**3.10739407578924**	**3.11769145362398**	**0.00317246382279579**
2	**61**	**12**	**3.11704786750931**	**3.11769145362398**	**0.000396557977849543**
3	**127**	**27**	**3.11756432550257**	**3.11769145362398**	**0.000117498660104104**
4	**217**	**48**	**3.11765122949181**	**3.11769145362398**	**4.95697472316092e-005**
5	**331**	**75**	**3.11767497781946**	**3.11769145362398**	**2.53797105838616e-005**

Table-3b(refer [32])

TORSION OF AN EQUILATERAL TRIANGULAR CROSS SECTION(EIGHT NODED ELEMENTS)

Mesh No	nodes	elements	TORISONAL CONSTANT VALUES FEM SOL	EXACT SOL	maximum absolute error of Prandtl Stress function values at element nodes
1	**16**	**3**	**3.01771523127018**	**3.11769145362398**	**0.0326698575267504**

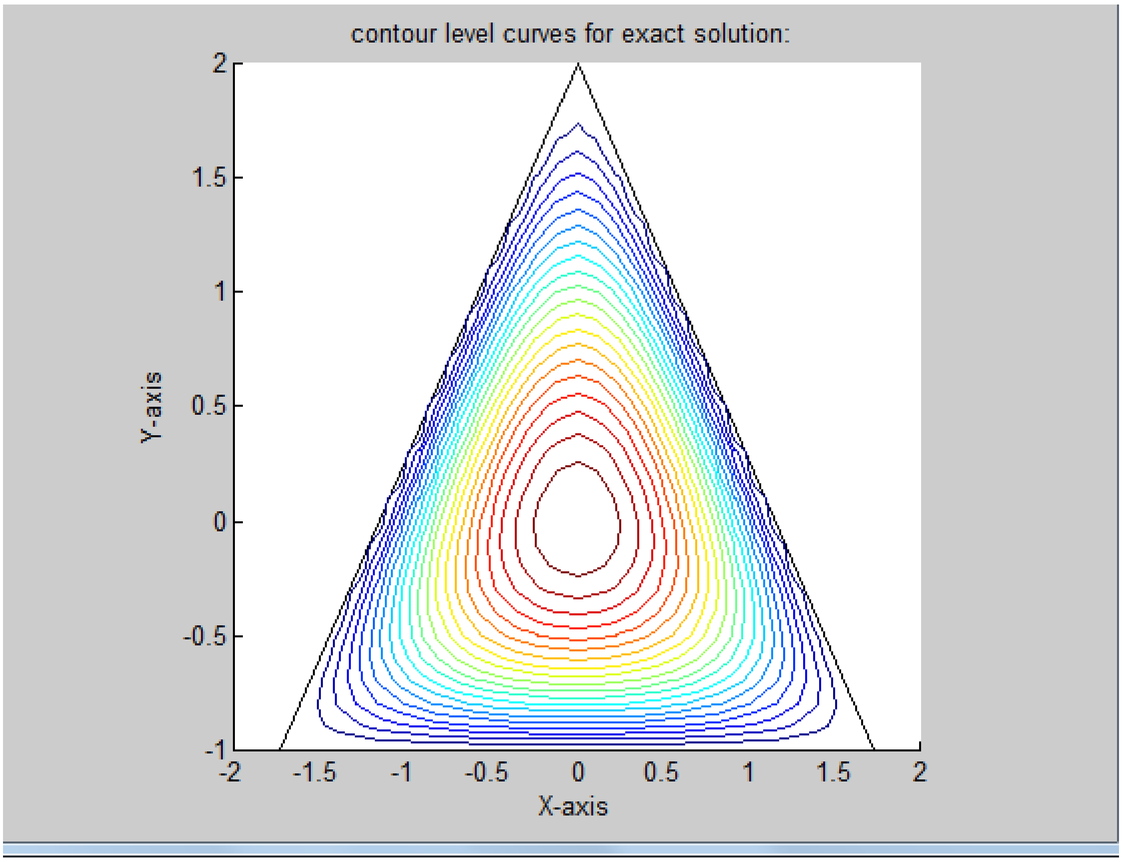
2	**49**	**12**	**3.10973432469345**	**3.11769145362398**	**0.00644969161285464**
3	**100**	**27**	**3.11557019336415**	**3.11769145362398**	**0.00209212772860204**
4	**169**	**48**	**3.1167795362198**	**3.11769145362398**	**0.00131858829294726**
5	**256**	**75**	**3.11719193923821**	**3.11769145362398**	**0.000908905458018264**

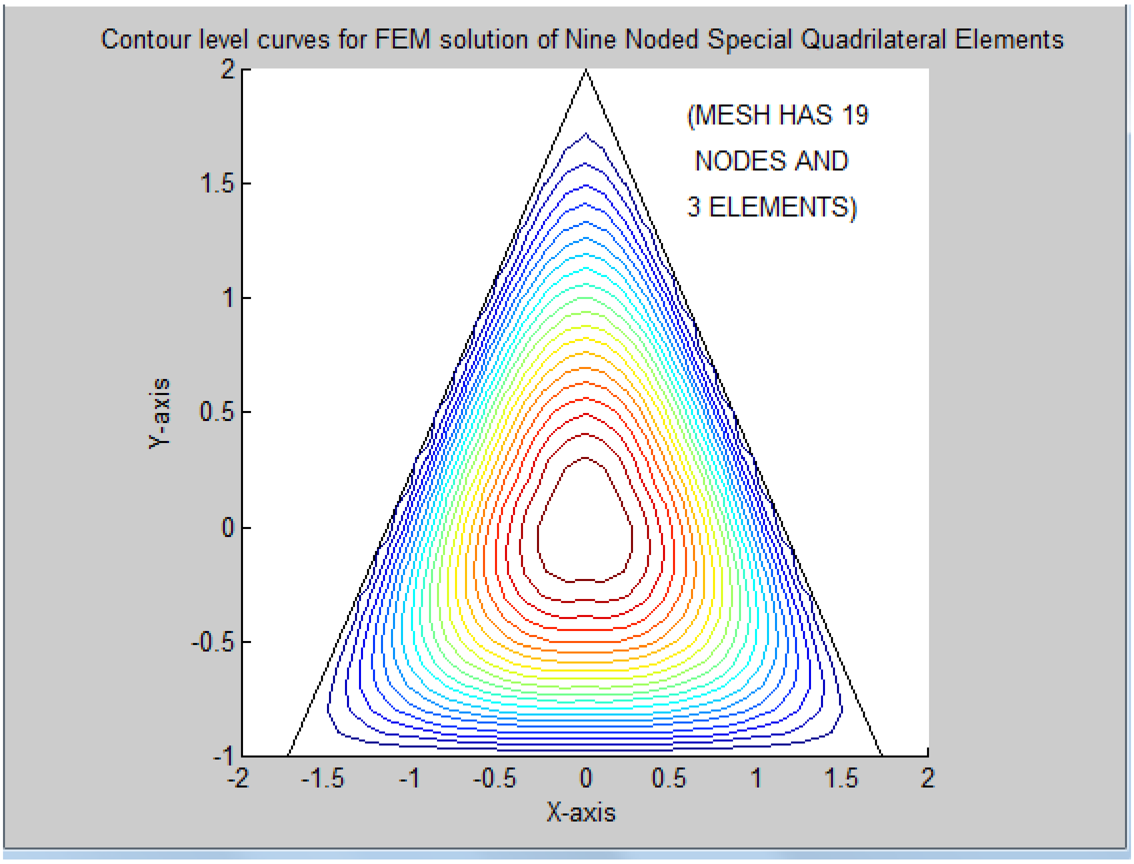
Table-3c(refer [28])

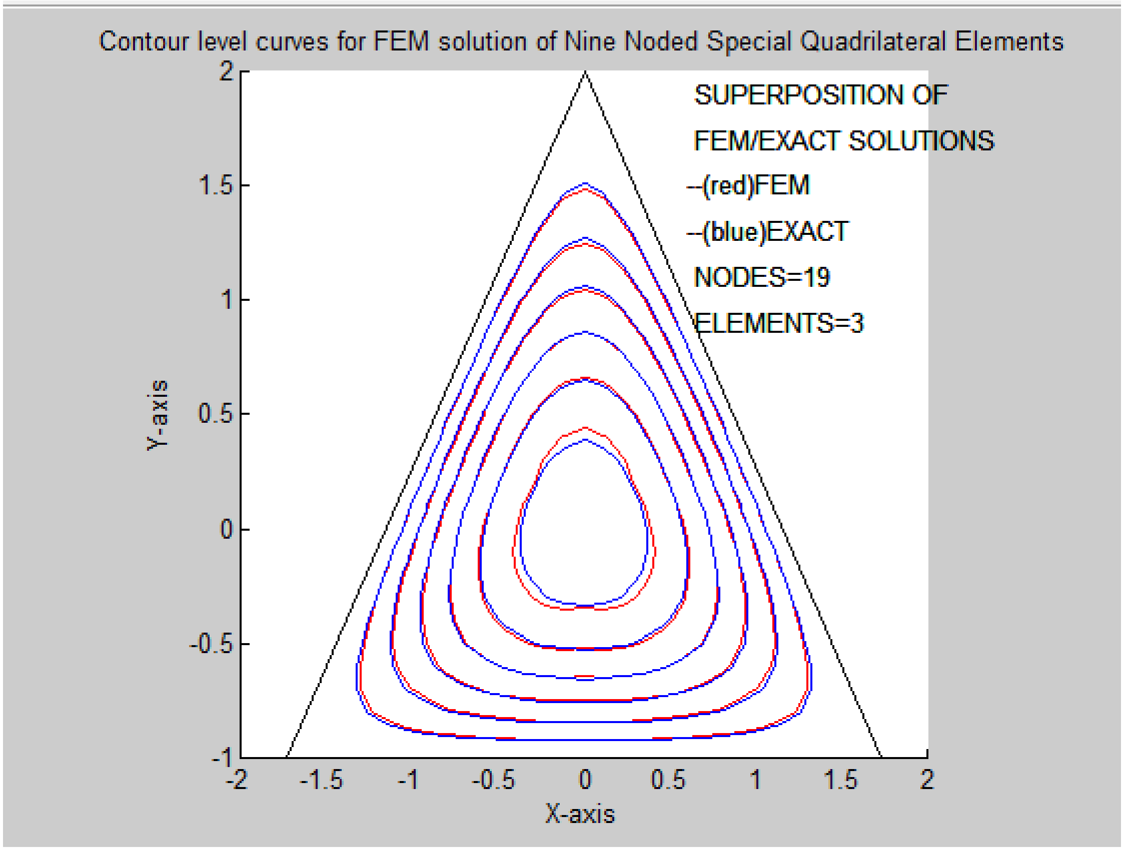
TORSION OF AN EQUILATERAL TRIANGULAR CROSS SECTION(FOUR NODED ELEMENTS)

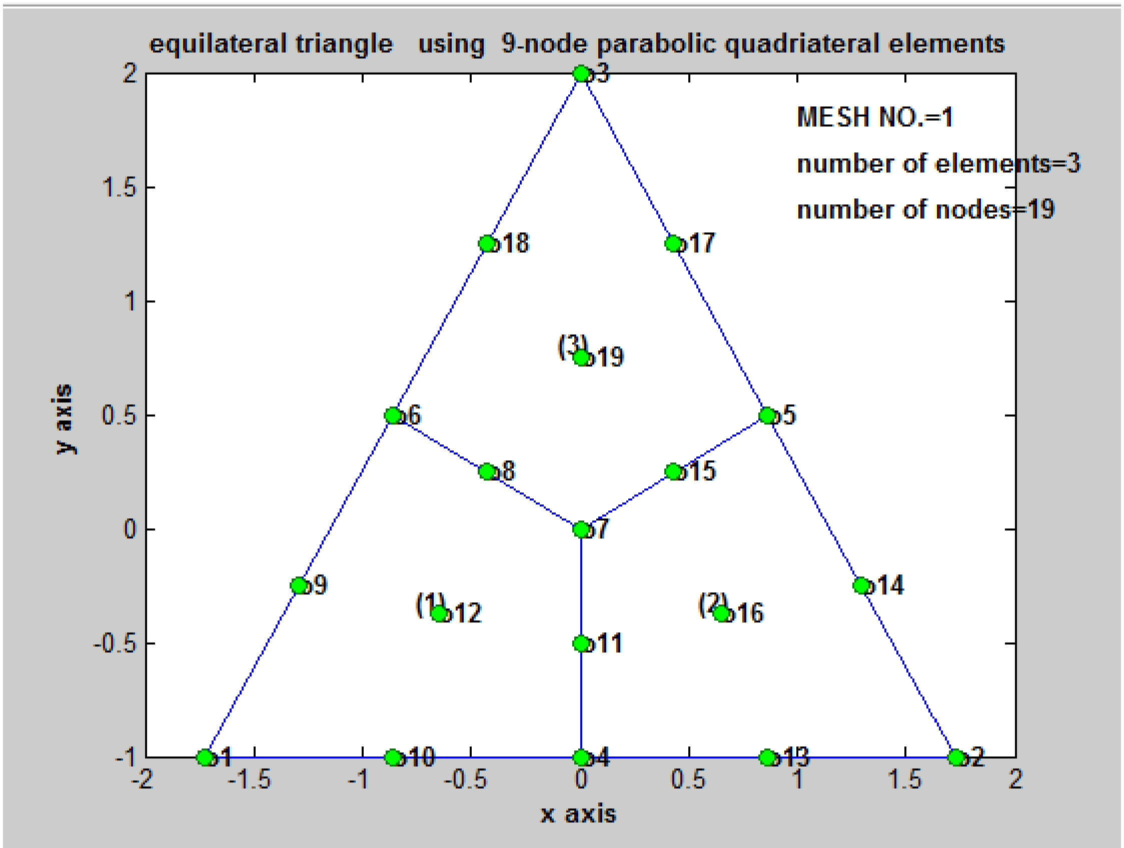
			TORISONAL CONSTANT VALUES		maximum absolute error
Mesh No	nodes	elements	FEM SOL	EXACT SOL	of Prandtl Stress function values at element nodes
1	7	3	1.8722884497	3.1176914536	1.2454030039
2	19	12	2.5125294041	3.1176914536	0.6051620495
3	37	27	2.8307089573	3.1176914536	0.2869824963
4	61	48	2.9539197663	3.1176914536	0.1637716873
5	91	75	3.0125356171	3.1176914536	0.1051558365
6	127	108	3.0446697570	3.1176914536	0.0730216966
7	169	147	3.0641026510	3.1176914536	0.0535888026
8	217	192	3.0767211933	3.1176914536	0.0409702604
9	271	243	3.0853673364	3.1176914536	0.0323241172
10	331	300	3.0915454781	3.1176914536	0.0261459755

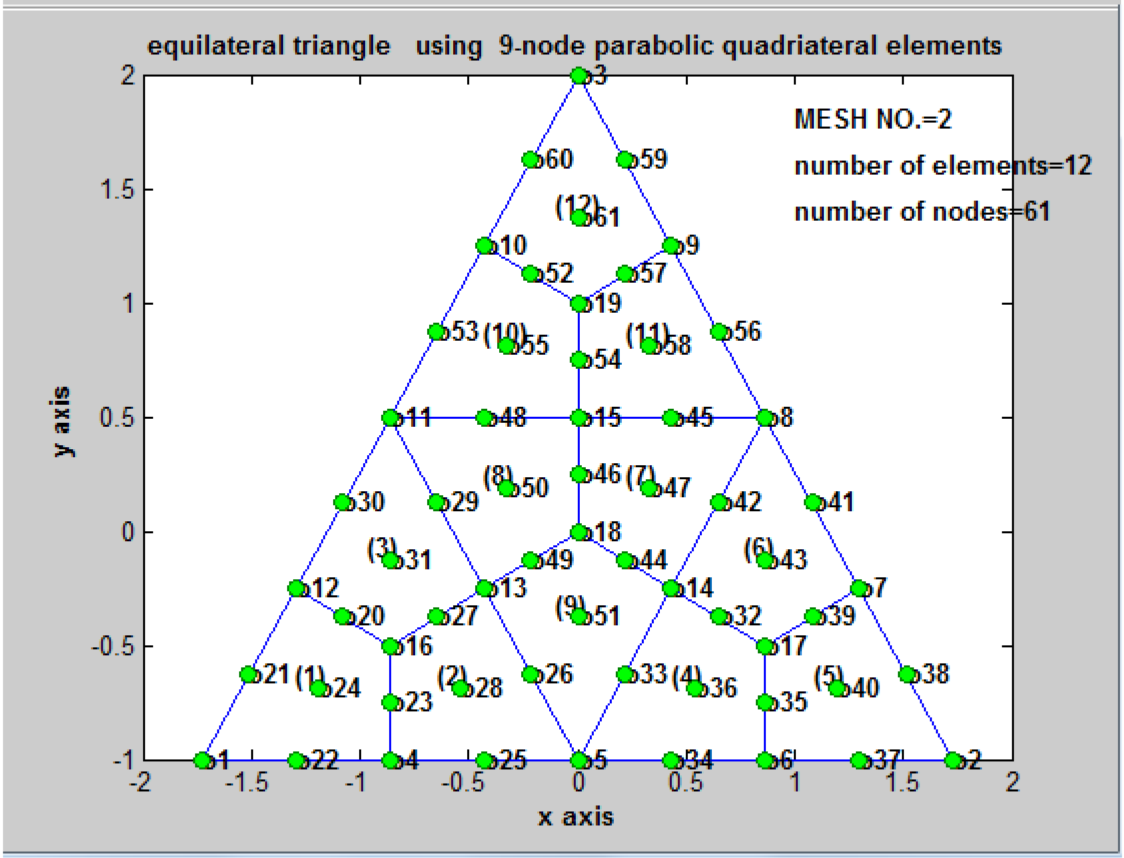
Example:TORSION OF A TRIANGULAR CROSS SECTION

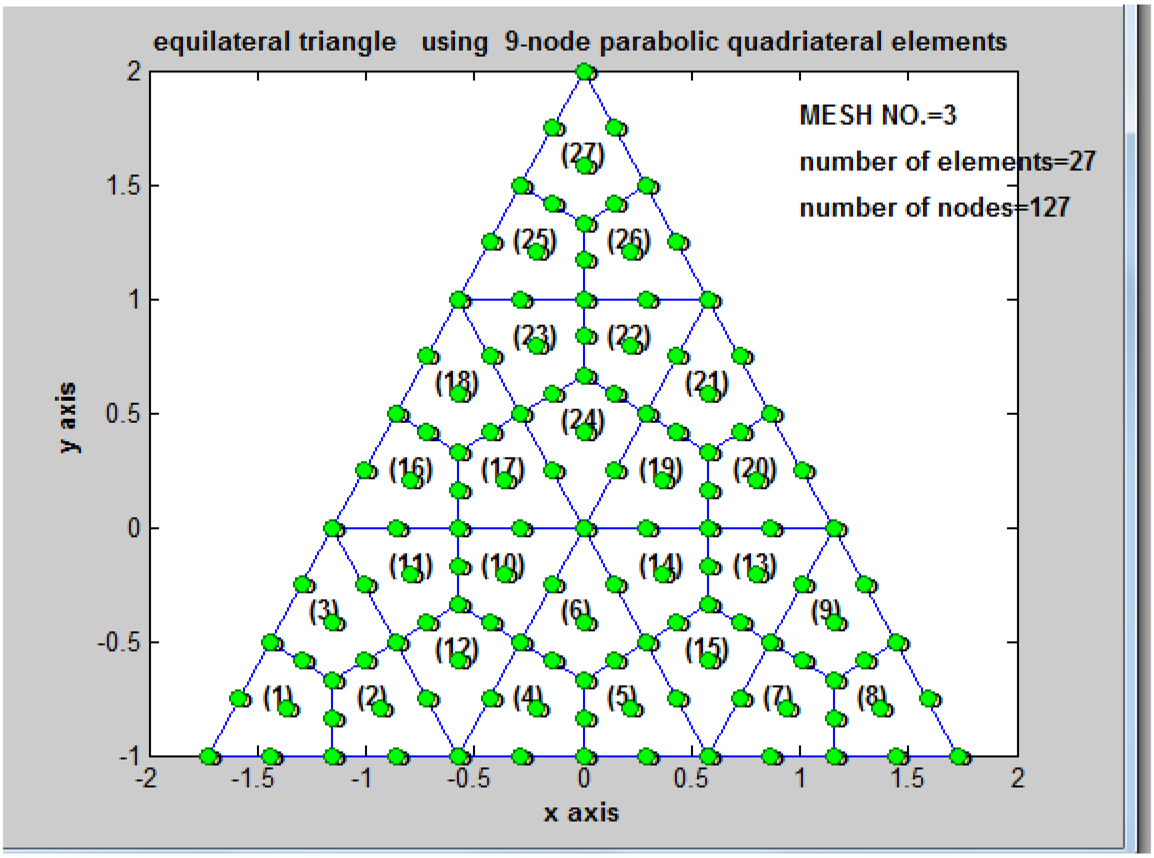

contour level curves for exact solution:
Y-axis
X-axis
2
1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

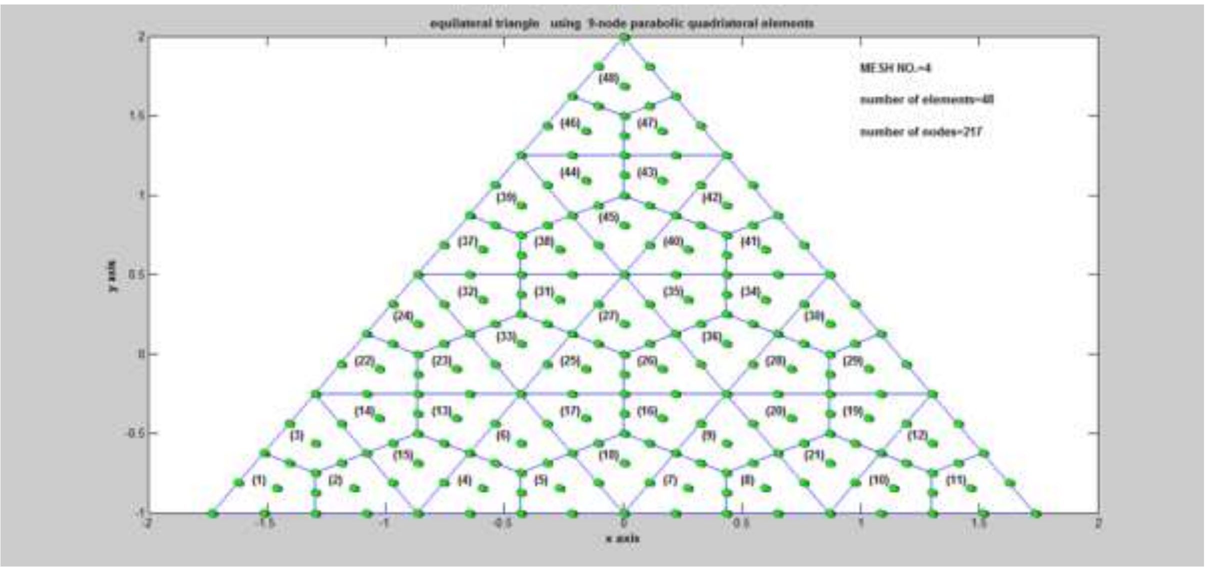

Contour level curves for FEM solution of Nine Noded Special Quadrilateral Elements
(MESH HAS 19
NODES AND
3 ELEMENTS)
Y-axis
X-axis
2
1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

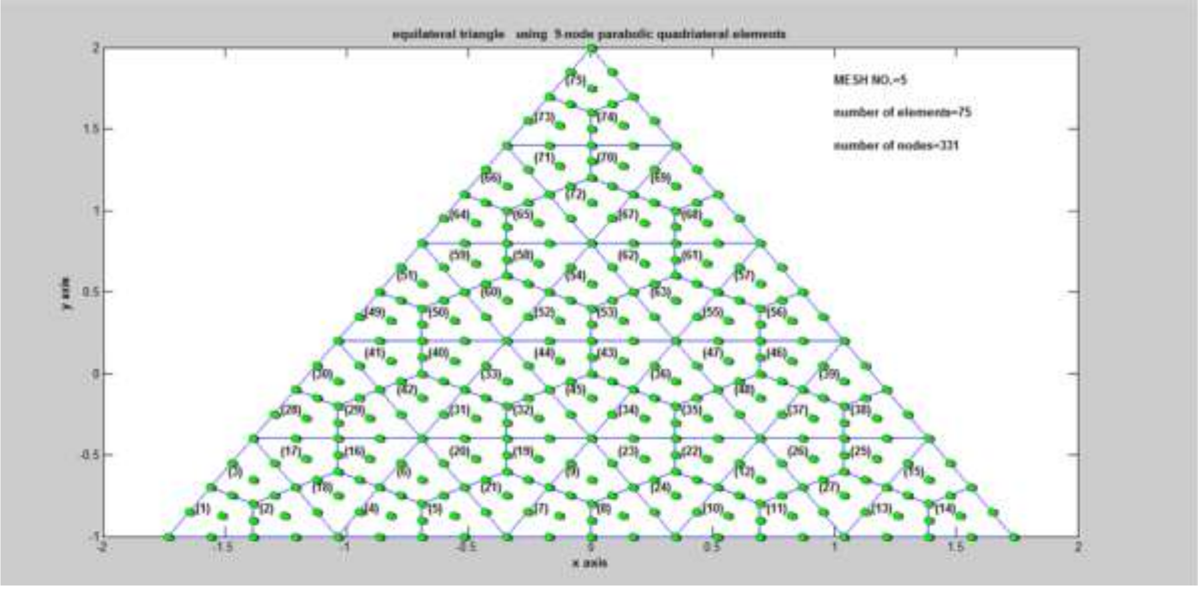

Contour level curves for FEM solution of Nine Noded Special Quadrilateral Elements
SUPERPOSITION OF
FEM/EXACT SOLUTIONS
--(red)FEM
--(blue)EXACT
NODES=19
ELEMENTS=3
Y-axis
X-axis
2
1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

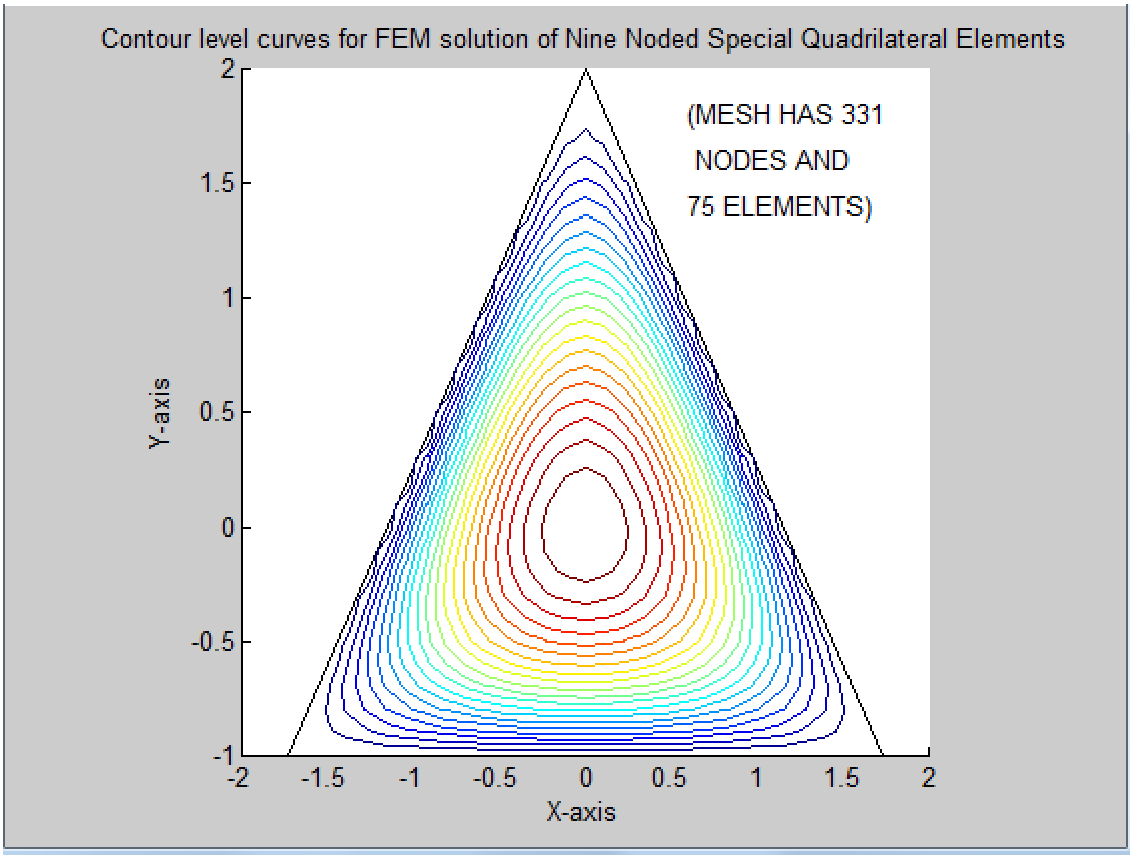

equilateral triangle using 9-node parabolic quadriateral elements
MESH NO.=1
number of elements=3
number of nodes=19
y axis
x axis
2
1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
(1)
(2)
(3)

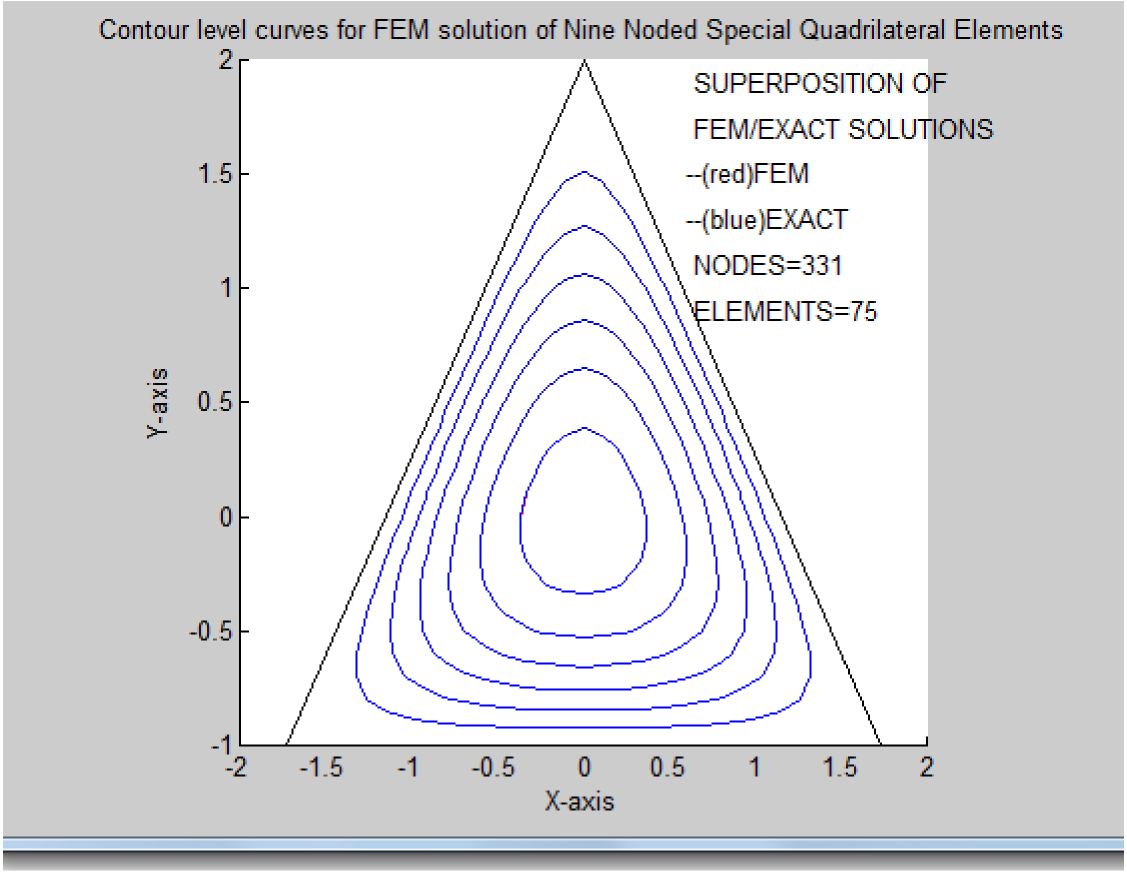

equilateral triangle using 9-node parabolic quadriateral elements
MESH NO.=2
number of elements=12
number of nodes=61
y axis
x axis
2
1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61


equilateral triangle using 9-node parabolic quadriateral elements
MESH NO.=3
number of elements=27
number of nodes=127
(27)
(25)
(26)
(23)
(22)
(18)
(21)
(24)
(16)
(17)
(19)
(20)
(11)
(10)
(14)
(13)
(3)
(6)
(9)
(12)
(15)
(1)
(2)
(4)
(5)
(7)
(8)
2
1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
y axis
x axis


equilateral triangle using 9-node parabolic quadriateral elements
MESH NO.=4
number of elements=48
number of nodes=217
y axis
x axis


equilateral triangle using 9-node parabolic quadrilateral elements
MESH NO.=5
number of elements=75
number of nodes=331
(75)
(73) (74)
(71) (70)
(66) (69)
(72)
(64) (65) (67) (68)
(59) (58) (62) (61)
(51) (57)
(60) (54) (63)
(49) (50) (52) (53) (55) (56)
(41) (40) (44) (43) (47) (46)
(30) (33) (36) (39)
(42) (45) (48)
(28) (29) (31) (32) (34) (35) (37) (38)
(17) (16) (20) (19) (23) (22) (26) (25)
(3) (6) (9) (12) (15)
(18) (21) (24) (27)
(1) (2) (4) (5) (7) (8) (10) (11) (13) (14)
y axis
2
1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
x axis


Contour level curves for FEM solution of Nine Noded Special Quadrilateral Elements
(MESH HAS 331
NODES AND
75 ELEMENTS)
Y-axis
2
1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
X-axis



Example5:TORSION OF A SQUARE CROSS SECTION

Table 4a
Torsion of a square cross section::Prandtl Stress Function Values
Mesh:Number of nodes=113,Number of nine noded elements=24

NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION
38	0.12058111589899	0.12051563971659	42	0.05815465754034	0.05825586477343	45	0.04329079617995	0.04418561671400
49	0.12058111589899	0.12051581611974	52	0.04329079617995	0.04418530139887	55	0.05815465754034	0.05825576227700
59	0.12058111589899	0.12051581611974	62	0.05815465754034	0.05825576227700	65	0.04329079617995	0.04418530139887
69	0.12058111589899	0.12051563971659	72	0.04329079617995	0.04418561671400	75	0.05815465754034	0.05825586477343
79	0.12058111589899	0.12051563971659	82	0.05815465754034	0.05825586477343	85	0.04329079617995	0.04418561671400
89	0.12058111589899	0.12051581611974	92	0.04329079617995	0.04418530139887	95	0.05815465754034	0.05825576227700
99	0.12058111589899	0.12051581611974	102	0.05815465754034	0.05825576227700	105	0.04329079617995	0.04418530139887

108	0.12058111589899	0.12051563971659	111	0.04329079617995	0.04418561671400	113	0.05815465754034	0.05825586477343

Table 4b
Torsion of a square cross section::Prandtl Stress Function Values
Mesh:Number of nodes=417,Number of nine noded elements=96

NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION
118	0.14058402174602	0.14057991011659	122	0.12606659834304	0.12607100298720	125	0.11819964167791	0.11822264404702
130	0.07829162319465	0.07828581395383	134	0.03230269835713	0.03230342146107	137	0.03061129750487	0.03065536488900
141	0.05313961558180	0.05317475469774	144	0.08633821959216	0.08635250522867	145	0.09538194827035	0.09536508048004
149	0.05554916029394	0.05549859083375	152	0.02370683846389	0.02371656300946	155	0.01561138492251	0.01583498199359
159	0.14058402174602	0.14057977294577	162	0.11819964167791	0.11822247696936	165	0.12606659834304	0.12607077188216
169	0.05554916029394	0.05549881852549	172	0.01561138492251	0.01583580314745	175	0.02370683846389	0.02371741735483
179	0.05313961558180	0.05317474613552	182	0.09538194827035	0.09536494479211	183	0.08633821959216	0.08635246347495
187	0.07829162319465	0.07828576878682	190	0.03061129750487	0.03065601040301	193	0.03230269835713	0.03230400294771
197	0.14058402174602	0.14057977294577	200	0.12606659834304	0.12607077188216	203	0.11819964167791	0.11822247696936
207	0.07829162319465	0.07828576878682	210	0.03230269835713	0.03230400294771	213	0.03061129750487	0.03065601040301
217	0.05313961558180	0.05317474613552	220	0.08633821959216	0.08635246347495	221	0.09538194827035	0.09536494479211
225	0.05554916029394	0.05549881852549	228	0.02370683846389	0.02371741735483	231	0.01561138492251	0.01583580314745
235	0.14058402174602	0.14057991011659	238	0.11819964167791	0.11822264404702	241	0.12606659834304	0.12607100298720
245	0.05554916029394	0.05549859083375	248	0.01561138492251	0.01583498199359	251	0.02370683846389	0.02371656300946
255	0.05313961558180	0.05317475469774	258	0.09538194827035	0.09536508048004	259	0.08633821959216	0.08635250522867
263	0.07829162319465	0.07828581395383	266	0.03061129750487	0.03065536488900	269	0.03230269835713	0.03230342146107
273	0.14058402174602	0.14057991011659	276	0.12606659834304	0.12607100298720	279	0.11819964167791	0.11822264404702
283	0.07829162319465	0.07828581395383	286	0.03230269835713	0.03230342146107	289	0.03061129750487	0.03065536488900

```
  293  0.05313961558180  0.05317475469774  296  0.08633821959216  0.08635250522867  297  0.09538194827035
0.09536508048004
  301  0.05554916029394  0.05549859083375  304  0.02370683846389  0.02371656300946  307  0.01561138492251
0.01583498199359
  311  0.14058402174602  0.14057977294577  314  0.11819964167791  0.11822247696936  317  0.12606659834304
0.12607077188216
  321  0.05554916029394  0.05549881852549  324  0.01561138492251  0.01583580314745  327  0.02370683846389
0.02371741735483
  331  0.05313961558180  0.05317474613552  334  0.09538194827035  0.09536494479211  335  0.08633821959216
0.08635246347495
  339  0.07829162319465  0.07828576878682  342  0.03061129750487  0.03065601040301  345  0.03230269835713
0.03230400294771
  349  0.14058402174602  0.14057977294577  352  0.12606659834304  0.12607077188216  355  0.11819964167791
0.11822247696936
  359  0.07829162319465  0.07828576878682  362  0.03230269835713  0.03230400294771  365  0.03061129750487
0.03065601040301
  369  0.05313961558180  0.05317474613552  372  0.08633821959216  0.08635246347495  373  0.09538194827035
0.09536494479211
  377  0.05554916029394  0.05549881852549  380  0.02370683846389  0.02371741735483  383  0.01561138492251
0.01583580314745
  386  0.14058402174602  0.14057991011659  389  0.11819964167791  0.11822264404702  391  0.12606659834304
0.12607100298720
  395  0.05554916029394  0.05549859083375  398  0.01561138492251  0.01583498199359  401  0.02370683846389
0.02371656300946
  405  0.05313961558180  0.05317475469774  408  0.09538194827035  0.09536508048004  409  0.08633821959216
0.08635250522867
  412  0.07829162319465  0.07828581395383  415  0.03061129750487  0.03065536488900  417  0.03230269835713
0.03230342146107
------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------
```

Table 4c

Torsion of a square cross section::Prandtl Stress Function Values

Mesh:Number of nodes=913,Number of nine noded elements=96

```
_______________________________________________________________________________________________________________
_______

NODE   FEM SOLUTION EXACT SOLUTION          NODE   FEM SOLUTION EXACT SOLUTION          NODE   FEM
SOLUTION EXACT SOLUTION

------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------

  246  0.14433333219977  0.14433234669041  250  0.13796948006707  0.13797008232081  253  0.13412472870848
0.13412887952449
  258  0.11788082756962  0.11787916908081  262  0.09949468254560  0.09949520367557  265  0.09692684724613
0.09693472253870
  270  0.05652158769702  0.05651970194102  274  0.02219663795132  0.02219628932800  277  0.02170285319259
0.02171367177965
  281  0.10458047793871  0.10458730848097  284  0.11919187393044  0.11919400576816  285  0.12447574614932
0.12447195716961
  289  0.04021618706311  0.04022427127012  292  0.06919860279300  0.06920166800843  293  0.07193694214824
0.07193138178694
  297  0.10241983457569  0.10241275116633  300  0.08689301113676  0.08689467865435  303  0.07691875779399
0.07693039408998
  307  0.04997507629917  0.04996682333415  310  0.01980942844874  0.01980680670951  313  0.01786726473065
0.01788595244207
  317  0.03085537563365  0.03087043871473  320  0.05207864094458  0.05208350462548  321  0.06041612925117
0.06040621499874
  325  0.03130926700361  0.03128641504819  328  0.01290305114353  0.01290752628843  331  0.00818320410613
0.00828323169950
```

335	0.14433333219977	0.14433246600812	338	0.13412472870848	0.13412908168048	341	0.13796948006707	0.13797018573330
345	0.10241983457569	0.10241283022985	348	0.07691875779399	0.07693025249116	351	0.08689301113676	0.08689462104669
355	0.03130926700361	0.03128590027657	358	0.00818320410613	0.00828304838643	361	0.01290305114353	0.01290762963338
365	0.10458047793871	0.10458713486162	368	0.12447574614932	0.12447168365388	369	0.11919187393044	0.11919374815765
373	0.03085537563365	0.03087119931359	376	0.06041612925117	0.06040655226683	377	0.05207864094458	0.05208397856475
381	0.11788082756962	0.11787909394632	384	0.09692684724613	0.09693460947652	387	0.09949468254560	0.09949499186990
391	0.04997507629917	0.04996672489415	394	0.01786726473065	0.01788638812152	397	0.01980942844874	0.01980732637832
401	0.04021618706311	0.04022471665081	404	0.07193694214824	0.07193152460283	405	0.06919860279300	0.06920182672954
409	0.05652158769702	0.05651944930300	412	0.02170285319259	0.02171413599477	415	0.02219663795132	0.02219665480144
419	0.14433333219977	0.14433246600812	422	0.13796948006707	0.13797018573330	425	0.13412472870848	0.13412908168048
429	0.11788082756962	0.11787909394632	432	0.09949468254560	0.09949499186990	435	0.09692684724613	0.09693460947652
439	0.05652158769702	0.05651944930300	442	0.02219663795132	0.02219665480144	445	0.02170285319259	0.02171413599477
449	0.10458047793871	0.10458713486162	452	0.11919187393044	0.11919374815765	453	0.12447574614932	0.12447168365388
457	0.04021618706311	0.04022471665081	460	0.06919860279300	0.06920182672954	461	0.07193694214824	0.07193152460283
465	0.10241983457569	0.10241283022985	468	0.08689301113676	0.08689462104669	471	0.07691875779399	0.07693025249116
475	0.04997507629917	0.04996672489415	478	0.01980942844874	0.01980732637832	481	0.01786726473065	0.01788638812152
485	0.03085537563365	0.03087119931359	488	0.05207864094458	0.05208397856475	489	0.06041612925117	0.06040655226683
493	0.03130926700361	0.03128590027657	496	0.01290305114353	0.01290762963338	499	0.00818320410613	0.00828304838643
503	0.14433333219977	0.14433234669041	506	0.13412472870848	0.13412887952449	509	0.13796948006707	0.13797008232081
513	0.10241983457569	0.10241275116633	516	0.07691875779399	0.07693039408998	519	0.08689301113676	0.08689467865435
523	0.03130926700361	0.03128641504819	526	0.00818320410613	0.00828323169950	529	0.01290305114353	0.01290752628843
533	0.10458047793871	0.10458730848097	536	0.12447574614932	0.12447195716961	537	0.11919187393044	0.11919400576816
541	0.03085537563365	0.03087043871473	544	0.06041612925117	0.06040621499874	545	0.05207864094458	0.05208350462548
549	0.11788082756962	0.11787916908081	552	0.09692684724613	0.09693472253870	555	0.09949468254560	0.09949520367557
559	0.04997507629917	0.04996682333415	562	0.01786726473065	0.01788595244207	565	0.01980942844874	0.01980680670951
569	0.04021618706311	0.04022427127012	572	0.07193694214824	0.07193138178694	573	0.06919860279300	0.06920166800843
577	0.05652158769702	0.05651970194102	580	0.02170285319259	0.02171367177965	583	0.02219663795132	0.02219628932800
587	0.14433333219977	0.14433234669041	590	0.13796948006707	0.13797008232081	593	0.13412472870848	0.13412887952449
597	0.11788082756962	0.11787916908081	600	0.09949468254560	0.09949520367557	603	0.09692684724613	0.09693472253870
607	0.05652158769702	0.05651970194102	610	0.02219663795132	0.02219628932800	613	0.02170285319259	0.02171367177965
617	0.10458047793871	0.10458730848097	620	0.11919187393044	0.11919400576816	621	0.12447574614932	0.12447195716961
625	0.04021618706311	0.04022427127012	628	0.06919860279300	0.06920166800843	629	0.07193694214824	0.07193138178694

633	0.10241983457569	0.10241275116633	636	0.08689301113676	0.08689467865435	639	0.07691875779399	0.07693039408998
643	0.04997507629917	0.04996682333415	646	0.01980942844874	0.01980680670951	649	0.01786726473065	0.01788595244207
653	0.03085537563365	0.03087043871473	656	0.05207864094458	0.05208350462548	657	0.06041612925117	0.06040621499874
661	0.03130926700361	0.03128641504819	664	0.01290305114353	0.01290752628843	667	0.00818320410613	0.00828323169950
671	0.14433333219977	0.14433246600812	674	0.13412472870848	0.13412908168048	677	0.13796948006707	0.13797018573330
681	0.10241983457569	0.10241283022985	684	0.07691875779399	0.07693025249116	687	0.08689301113676	0.08689462104669
691	0.03130926700361	0.03128590027657	694	0.00818320410613	0.00828304838643	697	0.01290305114353	0.01290762963338
701	0.10458047793871	0.10458713486162	704	0.12447574614932	0.12447168365388	705	0.11919187393044	0.11919374815765
709	0.03085537563365	0.03087119931359	712	0.06041612925117	0.06040655226683	713	0.05207864094458	0.05208397856475
717	0.11788082756962	0.11787909394632	720	0.09692684724613	0.09693460947652	723	0.09949468254560	0.09949499186990
727	0.04997507629917	0.04996672489415	730	0.01786726473065	0.01788638812152	733	0.01980942844874	0.01980732637832
737	0.04021618706311	0.04022471665081	740	0.07193694214824	0.07193152460283	741	0.06919860279300	0.06920182672954
745	0.05652158769702	0.05651944930300	748	0.02170285319259	0.02171413599477	751	0.02219663795132	0.02219665480144
755	0.14433333219977	0.14433246600812	758	0.13796948006707	0.13797018573330	761	0.13412472870848	0.13412908168048
765	0.11788082756962	0.11787909394632	768	0.09949468254560	0.09949499186990	771	0.09692684724613	0.09693460947652
775	0.05652158769702	0.05651944930300	778	0.02219663795132	0.02219665480144	781	0.02170285319259	0.02171413599477
785	0.10458047793871	0.10458713486162	788	0.11919187393044	0.11919374815765	789	0.12447574614932	0.12447168365388
793	0.04021618706311	0.04022471665081	796	0.06919860279300	0.06920182672954	797	0.07193694214824	0.07193152460283
801	0.10241983457569	0.10241283022985	804	0.08689301113676	0.08689462104669	807	0.07691875779399	0.07693025249116
811	0.04997507629917	0.04996672489415	814	0.01980942844874	0.01980732637832	817	0.01786726473065	0.01788638812152
821	0.03085537563365	0.03087119931359	824	0.05207864094458	0.05208397856475	825	0.06041612925117	0.06040655226683
829	0.03130926700361	0.03128590027657	832	0.01290305114353	0.01290762963338	835	0.00818320410613	0.00828304838643
838	0.14433333219977	0.14433234669041	841	0.13412472870848	0.13412887952449	843	0.13796948006707	0.13797008232081
847	0.10241983457569	0.10241275116633	850	0.07691875779399	0.07693039408998	853	0.08689301113676	0.08689467865435
857	0.03130926700361	0.03128641504819	860	0.00818320410613	0.00828323169950	863	0.01290305114353	0.01290752628843
867	0.10458047793871	0.10458730848097	870	0.12447574614932	0.12447195716961	871	0.11919187393044	0.11919400576816
875	0.03085537563365	0.03087043871473	878	0.06041612925117	0.06040621499874	879	0.05207864094458	0.05208350462548
882	0.11788082756962	0.11787916908081	885	0.09692684724613	0.09693472253870	887	0.09949468254560	0.09949520367557
891	0.04997507629917	0.04996682333415	894	0.01786726473065	0.01788595244207	897	0.01980942844874	0.01980680670951
901	0.04021618706311	0.04022427127012	904	0.07193694214824	0.07193138178694	905	0.06919860279300	0.06920166800843
908	0.05652158769702	0.05651970194102	911	0.02170285319259	0.02171367177965	913	0.02219663795132	0.02219628932800

Table 4d
Torsion of a square cross section::Prandtl Stress Function Values
Mesh:Number of nodes=1601,Number of nine noded elements=384

NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION
422	0.14564882193950	0.14564850166381	426	0.14208649815471	0.14208667583111	429	0.13985527731080	0.13985664851894
434	0.13101272754489	0.13101216481656	438	0.12108203921134	0.12108216535664	441	0.11925791310345	0.11926045648225
446	0.09848195119661	0.09848119842851	450	0.08091235217421	0.08091240989557	453	0.07979086601672	0.07979439352486
458	0.04405900279918	0.04405820257296	462	0.01688874318962	0.01688855310157	465	0.01668146633670	0.01668587125885
469	0.12310296984080	0.12310494525266	472	0.13130305077015	0.13130350351770	473	0.13452375325154	0.13452255122710
477	0.08852501053921	0.08852758581299	480	0.10382002358479	0.10382074166801	481	0.10622121710458	0.10621925474092
485	0.03167976690994	0.03168267057947	488	0.05597058300803	0.05597148564585	489	0.05713384426848	0.05713141455742
493	0.12127673851649	0.12127476157203	496	0.11225360105090	0.11225422334320	499	0.10550458118647	0.10550777492662
503	0.09162247079606	0.09161992564901	506	0.07547731092996	0.07547778781461	509	0.07128937674598	0.07129386969292
513	0.04132710409536	0.04132442898115	516	0.01590236529074	0.01590128555616	519	0.01513165919958	0.01513719627633
523	0.07669344895523	0.07669747832015	526	0.08951378454228	0.08951452455303	527	0.09631729400316	0.09631403414148
531	0.02793612832763	0.02794030007448	534	0.04899094354889	0.04899214129568	535	0.05232725976334	0.05232342964798
539	0.07377571467254	0.07377119917764	542	0.06125082843839	0.06125156071092	545	0.05304852373486	0.05305503958376
549	0.03411036569805	0.03410551405794	552	0.01328822289106	0.01328675186047	555	0.01174226234385	0.01175263634241
559	0.02023930647921	0.02024814929477	562	0.03476795312534	0.03477095544141	563	0.04116262484960	0.04115676184611
567	0.02025431875552	0.02024087103077	570	0.00820203998379	0.00820401150452	573	0.00509985809361	0.00515649923925
577	0.14564882193950	0.14564856305511	580	0.13985527731080	0.13985644002880	583	0.14208649815471	0.14208650531155
587	0.12127673851649	0.12127453045709	590	0.10550458118647	0.10550785252030	593	0.11225360105090	0.11225403661523
597	0.07377571467254	0.07377110019203	600	0.05304852373486	0.05305544972870	603	0.06125082843839	0.06125190256852
607	0.02025431875552	0.02024164756596	610	0.00509985809361	0.00515581715289	613	0.00820203998379	0.00820439611147
617	0.12310296984080	0.12310494605850	620	0.13452375325154	0.13452258444372	621	0.13130305077015	0.13130376864518
625	0.07669344895523	0.07669738967103	628	0.09631729400316	0.09631405816666	629	0.08951378454228	0.08951450419124
633	0.02023930647921	0.02024856279696	636	0.04116262484960	0.04115654922890	637	0.03476795312534	0.03477030198099
641	0.13101272754489	0.13101202830954	644	0.11925791310345	0.11926032639161	647	0.12108203921134	0.12108207323659

651 0.09162247079606 0.09161961957364 654 0.07128937674598 0.07129426878241 657 0.07547731092996 0.07547792258245

661 0.03411036569805 0.03410597899031 664 0.01174226234385 0.01175277769061 667 0.01328822289106 0.01328682492777

671 0.08852501053921 0.08852757556337 674 0.10622121710458 0.10621908086777 675 0.10382002358479 0.10382079970573

679 0.02793612832763 0.02794103507246 682 0.05232725976334 0.05232314207035 683 0.04899094354889 0.04899180933108

687 0.09848195119661 0.09848098696107 690 0.07979086601672 0.07979458493004 693 0.08091235217421 0.08091263927133

697 0.04132710409536 0.04132468682377 700 0.01513165919958 0.01513732656289 703 0.01590236529074 0.01590115159531

707 0.03167976690994 0.03168348397695 710 0.05713384426847 0.05713092908146 711 0.05597058300803 0.05597123208075

715 0.04405900279918 0.04405855502350 718 0.01668146633670 0.01668579391966 721 0.01688874318962 0.01688851372320

725 0.14564882193950 0.14564856305511 728 0.14208649815471 0.14208650531155 731 0.13985527731080 0.13985644002880

735 0.13101272754489 0.13101202830954 738 0.12108203921134 0.12108207323659 741 0.11925791310345 0.11926032639161

745 0.09848195119661 0.09848098696107 748 0.08091235217421 0.08091263927133 751 0.07979086601672 0.07979458493004

755 0.04405900279918 0.04405855502350 758 0.01688874318962 0.01688851372320 761 0.01668146633670 0.01668579391966

765 0.12310296984080 0.12310494605850 768 0.13130305077015 0.13130376864518 769 0.13452375325154 0.13452258444372

773 0.08852501053921 0.08852757556337 776 0.10382002358479 0.10382079970573 777 0.10622121710458 0.10621908086777

781 0.03167976690994 0.03168348397695 784 0.05597058300803 0.05597123208075 785 0.05713384426848 0.05713092908146

789 0.12127673851649 0.12127453045709 792 0.11225360105090 0.11225403661523 795 0.10550458118647 0.10550785252030

799 0.09162247079606 0.09161961957364 802 0.07547731092996 0.07547792258245 805 0.07128937674598 0.07129426878241

809 0.04132710409536 0.04132468682377 812 0.01590236529074 0.01590115159531 815 0.01513165919958 0.01513732656289

819 0.07669344895523 0.07669738967103 822 0.08951378454228 0.08951450419124 823 0.09631729400316 0.09631405816666

827 0.02793612832763 0.02794103507246 830 0.04899094354889 0.04899180933108 831 0.05232725976334 0.05232314207035

835 0.07377571467255 0.07377110019203 838 0.06125082843839 0.06125190256852 841 0.05304852373487 0.05305544972870

845 0.03411036569805 0.03410597899031 848 0.01328822289106 0.01328682492777 851 0.01174226234385 0.01175277769061

855 0.02023930647921 0.02024856279696 858 0.03476795312534 0.03477030198099 859 0.04116262484960 0.04115654922890

863 0.02025431875552 0.02024164756596 866 0.00820203998379 0.00820439611147 869 0.00509985809361 0.00515581715289

873 0.14564882193950 0.14564850166381 876 0.13985527731080 0.13985664851894 879 0.14208649815471 0.14208667583111

883 0.12127673851649 0.12127476157203 886 0.10550458118647 0.10550777492662 889 0.11225360105090 0.11225422334320

893 0.07377571467255 0.07377119917764 896 0.05304852373487 0.05305503958376 899 0.06125082843839 0.06125156071092

903 0.02025431875552 0.02024087103077 906 0.00509985809361 0.00515649923925 909 0.00820203998379 0.00820401150452

913 0.12310296984080 0.12310494525266 916 0.13452375325154 0.13452255122710 917 0.13130305077015 0.13130350351770

921 0.07669344895523 0.07669747832015 924 0.09631729400316 0.09631403414148 925 0.08951378454228 0.08951452455303

929 0.02023930647921 0.02024814929477 932 0.04116262484960 0.04115676184611 933 0.03476795312534 0.03477095544141

937 0.13101272754489 0.13101216481656 940 0.11925791310345 0.11926045648225 943 0.12108203921134 0.12108216535664

947 0.09162247079606 0.09161992564901 950 0.07128937674598 0.07129386969292 953 0.07547731092996 0.07547778781461

957 0.03411036569805 0.03410551405794 960 0.01174226234385 0.01175263634241 963 0.01328822289106 0.01328675186047

967 0.08852501053921 0.08852758581299 970 0.10622121710458 0.10621925474092 971 0.10382002358479 0.10382074166801

975 0.02793612832763 0.02794030007448 978 0.05232725976334 0.05232342964798 979 0.04899094354889 0.04899214129568

983 0.09848195119661 0.09848119842851 986 0.07979086601672 0.07979439352486 989 0.08091235217421 0.08091240989557

993 0.04132710409536 0.04132442898115 996 0.01513165919958 0.01513719627633 999 0.01590236529074 0.01590128555616

1003 0.03167976690994 0.03168267057947 1006 0.05713384426848 0.05713141455742 1007 0.05597058300803 0.05597148564585

1011 0.04405900279918 0.04405820257296 1014 0.01668146633670 0.01668587125885 1017 0.01688874318962 0.01688855310157

1021 0.14564882193950 0.14564850166381 1024 0.14208649815471 0.14208667583111 1027 0.13985527731080 0.13985664851894

1031 0.13101272754489 0.13101216481656 1034 0.12108203921134 0.12108216535664 1037 0.11925791310345 0.11926045648225

1041 0.09848195119661 0.09848119842851 1044 0.08091235217421 0.08091240989557 1047 0.07979086601672 0.07979439352486

1051 0.04405900279918 0.04405820257296 1054 0.01688874318962 0.01688855310157 1057 0.01668146633670 0.01668587125885

1061 0.12310296984080 0.12310494525266 1064 0.13130305077015 0.13130350351770 1065 0.13452375325154 0.13452255122710

1069 0.08852501053921 0.08852758581299 1072 0.10382002358479 0.10382074166801 1073 0.10622121710458 0.10621925474092

1077 0.03167976690994 0.03168267057947 1080 0.05597058300803 0.05597148564585 1081 0.05713384426848 0.05713141455742

1085 0.12127673851649 0.12127476157203 1088 0.11225360105090 0.11225422334320 1091 0.10550458118647 0.10550777492662

1095 0.09162247079606 0.09161992564901 1098 0.07547731092997 0.07547778781461 1101 0.07128937674598 0.07129386969292

1105 0.04132710409536 0.04132442898115 1108 0.01590236529074 0.01590128555616 1111 0.01513165919958 0.01513719627633

1115 0.07669344895523 0.07669747832015 1118 0.08951378454228 0.08951452455303 1119 0.09631729400316 0.09631403414148

1123 0.02793612832763 0.02794030007448 1126 0.04899094354889 0.04899214129568 1127 0.05232725976334 0.05232342964798

1131 0.07377571467255 0.07377119917764 1134 0.06125082843839 0.06125156071092 1137 0.05304852373487 0.05305503958376

1141 0.03411036569805 0.03410551405794 1144 0.01328822289106 0.01328675186047 1147 0.01174226234385 0.01175263634241

1151 0.02023930647921 0.02024814929477 1154 0.03476795312534 0.03477095544141 1155 0.04116262484960 0.04115676184611

1159 0.02025431875552 0.02024087103077 1162 0.00820203998379 0.00820401150452 1165 0.00509985809361 0.00515649923925

1169 0.14564882193950 0.14564856305511 1172 0.13985527731080 0.13985644002880 1175 0.14208649815471 0.14208650531155

1179 0.12127673851649 0.12127453045709 1182 0.10550458118647 0.10550785252030 1185 0.11225360105090 0.11225403661523

1189 0.07377571467255 0.07377110019203 1192 0.05304852373486 0.05305544972870 1195 0.06125082843839 0.06125190256852

1199 0.02025431875552 0.02024164756596 1202 0.00509985809361 0.00515581715289 1205 0.00820203998379 0.00820439611147

1209 0.12310296984080 0.12310494605850 1212 0.13452375325154 0.13452258444372 1213 0.13130305077015 0.13130376864518

1217 0.07669344895523 0.07669738967103 1220 0.09631729400316 0.09631405816666 1221 0.08951378454228 0.08951450419124

1225 0.02023930647921 0.02024856279696 1228 0.04116262484960 0.04115654922890 1229 0.03476795312534 0.03477030198099

1233 0.13101272754489 0.13101202830954 1236 0.11925791310345 0.11926032639161 1239 0.12108203921134 0.12108207323659

1243 0.09162247079606 0.09161961957364 1246 0.07128937674598 0.07129426878241 1249 0.07547731092997 0.07547792258245
1253 0.03411036569805 0.03410597899031 1256 0.01174226234385 0.01175277769061 1259 0.01328822289106 0.01328682492777
1263 0.08852501053921 0.08852757556337 1266 0.10622121710458 0.10621908086777 1267 0.10382002358479 0.10382079970573
1271 0.02793612832763 0.02794103507246 1274 0.05232725976334 0.05232314207035 1275 0.04899094354889 0.04899180933108
1279 0.09848195119661 0.09848098696107 1282 0.07979086601672 0.07979458493004 1285 0.08091235217421 0.08091263927133
1289 0.04132710409536 0.04132468682377 1292 0.01513165919958 0.01513732656289 1295 0.01590236529074 0.01590115159531
1299 0.03167976690994 0.03168348397695 1302 0.05713384426848 0.05713092908146 1303 0.05597058300803 0.05597123208075
1307 0.04405900279918 0.04405855502350 1310 0.01668146633670 0.01668579391966 1313 0.01688874318962 0.01688851372320
1317 0.14564882193950 0.14564856305511 1320 0.14208649815471 0.14208650531155 1323 0.13985527731080 0.13985644002880
1327 0.13101272754489 0.13101202830954 1330 0.12108203921134 0.12108207323659 1333 0.11925791310345 0.11926032639161
1337 0.09848195119661 0.09848098696107 1340 0.08091235217421 0.08091263927133 1343 0.07979086601672 0.07979458493004
1347 0.04405900279918 0.04405855502350 1350 0.01688874318962 0.01688851372320 1353 0.01668146633670 0.01668579391966
1357 0.12310296984080 0.12310494605850 1360 0.13130305077015 0.13130376864518 1361 0.13452375325154 0.13452258444372
1365 0.08852501053921 0.08852757556337 1368 0.10382002358479 0.10382079970573 1369 0.10622121710458 0.10621908086777
1373 0.03167976690994 0.03168348397695 1376 0.05597058300803 0.05597123208075 1377 0.05713384426848 0.05713092908146
1381 0.12127673851649 0.12127453045709 1384 0.11225360105090 0.11225403661523 1387 0.10550458118647 0.10550785252030
1391 0.09162247079606 0.09161961957364 1394 0.07547731092996 0.07547792258245 1397 0.07128937674598 0.07129426878241
1401 0.04132710409536 0.04132468682377 1404 0.01590236529074 0.01590115159531 1407 0.01513165919958 0.01513732656289
1411 0.07669344895523 0.07669738967103 1414 0.08951378454228 0.08951450419124 1415 0.09631729400316 0.09631405816666
1419 0.02793612832763 0.02794103507246 1422 0.04899094354889 0.04899180933108 1423 0.05232725976334 0.05232314207035
1427 0.07377571467255 0.07377110019203 1430 0.06125082843839 0.06125190256852 1433 0.05304852373486 0.05305544972870
1437 0.03411036569805 0.03410597899031 1440 0.01328822289106 0.01328682492777 1443 0.01174226234385 0.01175277769061
1447 0.02023930647921 0.02024856279696 1450 0.03476795312534 0.03477030198099 1451 0.04116262484960 0.04115654922890
1455 0.02025431875552 0.02024164756596 1458 0.00820203998379 0.00820439611147 1461 0.00509985809361 0.00515581715289
1464 0.14564882193950 0.14564850166381 1467 0.13985527731080 0.13985664851894 1469 0.14208649815471 0.14208667583111
1473 0.12127673851649 0.12127476157203 1476 0.10550458118647 0.10550777492662 1479 0.11225360105090 0.11225422334320
1483 0.07377571467255 0.07377119917764 1486 0.05304852373487 0.05305503958376 1489 0.06125082843839 0.06125156071092
1493 0.02025431875552 0.02024087103077 1496 0.00509985809361 0.00515649923925 1499 0.00820203998379 0.00820401150452
1503 0.12310296984080 0.12310494525266 1506 0.13452375325154 0.13452255122710 1507 0.13130305077015 0.13130350351770
1511 0.07669344895523 0.07669747832015 1514 0.09631729400316 0.09631403414148 1515 0.08951378454228 0.08951452455303
1519 0.02023930647921 0.02024814929477 1522 0.04116262484960 0.04115676184611 1523 0.03476795312534 0.03477095544141
1526 0.13101272754489 0.13101216481656 1529 0.11925791310345 0.11926045648225 1531 0.12108203921134 0.12108216535664

1535	0.09162247079606	0.09161992564901	1538	0.07128937674598	0.07129386969292	1541	0.07547731092997	0.07547778781461
1545	0.03411036569805	0.03410551405794	1548	0.01174226234385	0.01175263634241	1551	0.01328822289106	0.01328675186047
1555	0.08852501053921	0.08852758581299	1558	0.10622121710458	0.10621925474092	1559	0.10382002358479	0.10382074166801
1563	0.02793612832763	0.02794030007448	1566	0.05232725976334	0.05232342964798	1567	0.04899094354889	0.04899214129568
1570	0.09848195119661	0.09848119842851	1573	0.07979086601672	0.07979439352486	1575	0.08091235217421	0.08091240989557
1579	0.04132710409536	0.04132442898115	1582	0.01513165919958	0.01513719627633	1585	0.01590236529074	0.01590128555616
1589	0.03167976690994	0.03168267057947	1592	0.05713384426848	0.05713141455742	1593	0.05597058300803	0.05597148564585
1596	0.04405900279918	0.04405820257296	1599	0.01668146633670	0.01668587125885	1601	0.01688874318962	0.01688855310157

Table 4e

Torsion of a square cross section::Prandtl Stress Function Values

Mesh:Number of nodes=2481,Number of nine noded elements=600

NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION
646	0.14625828214478	0.14625799334796	650	0.14398352999959	0.14398344465129	653	0.14253524587494	0.14253564500567
658	0.13696332947999	0.13696293746972	662	0.13072998931971	0.13072987760096	665	0.12944879535718	0.12944967327349
670	0.11673713166364	0.11673663554891	674	0.10602115913807	0.10602101141591	677	0.10503329287795	0.10503461041355
682	0.08391873690481	0.08391817314443	686	0.06788706591390	0.06788688630284	689	0.06730349895817	0.06730515385109
694	0.03605924523629	0.03605867528529	698	0.01362510608026	0.01362484287590	701	0.01351922271267	0.01352125998214
705	0.13177364211724	0.13177436324512	708	0.13701663213118	0.13701672541751	709	0.13915146714621	0.13915080245345
713	0.11011388417765	0.11011489278544	716	0.11961108317550	0.11961128162549	717	0.12140866920393	0.12140766383844
721	0.07573336777530	0.07573456786920	724	0.09026847026665	0.09026875898306	725	0.09154179389298	0.09154050438680
729	0.02600341715387	0.02600476506314	732	0.04664722489382	0.04664757101135	733	0.04724465149876	0.04724320568487
737	0.13042525501443	0.13042425445322	740	0.12456395040140	0.12456399956828	743	0.11991353923327	0.11991473153453
747	0.11138341046992	0.11138216099422	750	0.10126335998705	0.10126336575411	753	0.09766250042398	0.09766417558510
757	0.08032438624530	0.08032298689430	760	0.06507691428851	0.06507683804024	763	0.06294155634344	0.06294360755461
767	0.03466765313827	0.03466621657555	770	0.01312479947849	0.01312402508314	773	0.01274071714261	0.01274309887747
777	0.10061218807971	0.10061366302537	780	0.10909460241907	0.10909473031363	781	0.11405271354075	0.11405130281864
785	0.06964965378226	0.06965129896048	788	0.08279056508540	0.08279080947843	789	0.08632424541746	0.08632249084902
793	0.02413166908930	0.02413330939773	796	0.04313148049314	0.04313181767747	797	0.04479717860004	0.04479525024587

801	0.09789612928671	0.09789431467968	804	0.08924555254664	0.08924562616294	807	0.08260013408198	0.08260222305115
811	0.07120017028542	0.07119808804983	814	0.05792581151041	0.05792580050629	817	0.05393158489629	0.05393438477477
821	0.03111299708174	0.03111096543047	824	0.01184501107289	0.01184408484193	827	0.01112203233083	0.01112544586079
831	0.05796566866476	0.05796812695941	834	0.06849919642491	0.06849942867776	835	0.07476009548352	0.07475775859274
839	0.02050178906139	0.02050427002337	842	0.03633194224359	0.03633249126146	843	0.03932752253618	0.03932495408980
847	0.05526377695512	0.05526060037598	850	0.04534846995487	0.04534871604109	853	0.03881186270325	0.03881586882921
857	0.02480313499499	0.02479981603325	860	0.00956655088363	0.00956543089302	863	0.00835406477241	0.00836052242587
867	0.01438388530817	0.01438903034313	870	0.02496912805505	0.02497051748330	871	0.02990618899601	0.02990229756286
875	0.01427479196745	0.01426593988594	878	0.00571792128896	0.00571894559182	881	0.00351053591177	0.00354544447856
885	0.14625828214478	0.14625800572427	888	0.14253524587494	0.14253563018103	891	0.14398352999959	0.14398344064660
895	0.13042525501443	0.13042426542688	898	0.11991353923327	0.11991471180672	901	0.12456395040140	0.12456398526623
905	0.09789612928671	0.09789432420388	908	0.08260013408198	0.08260219169659	911	0.08924555254664	0.08924559062991
915	0.05526377695512	0.05526060445204	918	0.03881186270325	0.03881579588808	921	0.04534846995487	0.04534860549301
925	0.01427479196745	0.01426588254934	928	0.00351053591177	0.00354461935573	931	0.00571792128896	0.00571769637168
935	0.13177364211724	0.13177436378611	938	0.13915146714621	0.13915078118767	939	0.13701663213118	0.13701672053274
943	0.10061218807971	0.10061364601075	946	0.11405271354076	0.11405126470678	947	0.10909460241907	0.10909471747749
951	0.05796566866476	0.05796805592010	954	0.07476009548352	0.07475768010314	955	0.06849919642491	0.06849939524539
959	0.01438388530817	0.01438846454530	962	0.02990618899601	0.02990204180532	963	0.02496912805505	0.02497037634989
967	0.13696332947999	0.13696293955392	970	0.12944879535718	0.12944963926203	973	0.13072998931971	0.13072985440945
977	0.11138341046992	0.11138214338022	980	0.09766250042398	0.09766410748812	983	0.10126335998705	0.10126330308289
987	0.07120017028542	0.07119802316049	990	0.05393158489629	0.05393420943977	993	0.05792581151041	0.05792562099281
997	0.02480313499499	0.02479950701517	1000	0.00835406477241	0.00835905930447	1003	0.00956655088363	0.00956393024905
1007	0.11011388417766	0.11011485658400	1010	0.12140866920393	0.12140761543446	1011	0.11961108317550	0.11961124960253
1015	0.06964965378226	0.06965117955198	1018	0.08632424541746	0.08632238377176	1019	0.08279056508540	0.08279072767688
1023	0.02050178906139	0.02050356024535	1026	0.03932752253618	0.03932462391872	1027	0.03633194224359	0.03633220614761
1031	0.11673713166364	0.11673660904545	1034	0.10503329287795	0.10503452803291	1037	0.10602115913807	0.10602093985522
1041	0.08032438624530	0.08032289486677	1044	0.06294155634344	0.06294339547718	1047	0.06507691428851	0.06507663138857
1051	0.03111299708174	0.03111058744699	1054	0.01112203233083	0.01112388050008	1057	0.01184501107289	0.01184251532034
1061	0.07573336777530	0.07573442927389	1064	0.09154179389298	0.09154038701742	1065	0.09026847026665	0.09026865799470
1069	0.02413166908930	0.02413255125063	1072	0.04479717860004	0.04479489148713	1073	0.04313148049314	0.04313148419444
1077	0.08391873690481	0.08391807222744	1080	0.06730349895817	0.06730492749000	1083	0.06788706591391	0.06788667076170
1087	0.03466765313827	0.03466581145387	1090	0.01274071714261	0.01274149683065	1093	0.01312479947849	0.01312242845916

1097 0.02600341715387 0.02600398772927 1100 0.04724465149876 0.04724283663401 1101 0.04664722489382 0.04664721834150
1105 0.03605924523629 0.03605826127415 1108 0.01351922271267 0.01351964367429 1111 0.01362510608026 0.01362323737381
1115 0.14625828214478 0.14625800572427 1118 0.14398352999959 0.14398344064660 1121 0.14253524587494 0.14253563018103
1125 0.13696332947999 0.13696293955392 1128 0.13072998931971 0.13072985440945 1131 0.12944879535718 0.12944963926203
1135 0.11673713166364 0.11673660904545 1138 0.10602115913807 0.10602093985522 1141 0.10503329287795 0.10503452803291
1145 0.08391873690481 0.08391807222744 1148 0.06788706591391 0.06788667076170 1151 0.06730349895817 0.06730492749000
1155 0.03605924523629 0.03605826127415 1158 0.01362510608026 0.01362323737381 1161 0.01351922271267 0.01351964367429
1165 0.13177364211724 0.13177436378611 1168 0.13701663213118 0.13701672053274 1169 0.13915146714621 0.13915078118767
1173 0.11011388417766 0.11011485658400 1176 0.11961108317550 0.11961124960253 1177 0.12140866920393 0.12140761543446
1181 0.07573336777530 0.07573442927389 1184 0.09026847026665 0.09026865799470 1185 0.09154179389298 0.09154038701742
1189 0.02600341715387 0.02600398772927 1192 0.04664722489382 0.04664721834150 1193 0.04724465149876 0.04724283663401
1197 0.13042525501443 0.13042426542688 1200 0.12456395040140 0.12456398526623 1203 0.11991353923326 0.11991471180672
1207 0.11138341046992 0.11138214338022 1210 0.10126335998705 0.10126330308289 1213 0.09766250042398 0.09766410748812
1217 0.08032438624530 0.08032289486677 1220 0.06507691428851 0.06507663138857 1223 0.06294155634344 0.06294339547718
1227 0.03466765313827 0.03466581145387 1230 0.01312479947849 0.01312242845916 1233 0.01274071714261 0.01274149683065
1237 0.10061218807971 0.10061364601075 1240 0.10909460241907 0.10909471747749 1241 0.11405271354075 0.11405126470678
1245 0.06964965378226 0.06965117955198 1248 0.08279056508540 0.08279072767688 1249 0.08632424541746 0.08632238377176
1253 0.02413166908930 0.02413255125063 1256 0.04313148049314 0.04313148419444 1257 0.04479717860004 0.04479489148713
1261 0.09789612928671 0.09789432420388 1264 0.08924555254664 0.08924559062991 1267 0.08260013408198 0.08260219169659
1271 0.07120017028542 0.07119802316049 1274 0.05792581151041 0.05792562099281 1277 0.05393158489629 0.05393420943977
1281 0.03111299708174 0.03111058744699 1284 0.01184501107289 0.01184251532034 1287 0.01112203233083 0.01112388050008
1291 0.05796566866476 0.05796805592010 1294 0.06849919642491 0.06849939524539 1295 0.07476009548352 0.07475768010314
1299 0.02050178906139 0.02050356024535 1302 0.03633194224359 0.03633220614761 1303 0.03932752253618 0.03932462391872
1307 0.05526377695512 0.05526060445204 1310 0.04534846995487 0.04534860549301 1313 0.03881186270325 0.03881579588808
1317 0.02480313499499 0.02479950701517 1320 0.00956655088363 0.00956393024905 1323 0.00835406477241 0.00835905930447
1327 0.01438388530817 0.01438846454530 1330 0.02496912805505 0.02497037634989 1331 0.02990618899601 0.02990204180532
1335 0.01427479196745 0.01426588254934 1338 0.00571792128896 0.00571769637168 1341 0.00351053591177 0.00354461935573
1345 0.14625828214478 0.14625799334796 1348 0.14253524587494 0.14253564500567 1351 0.14398352999959 0.14398344465129
1355 0.13042525501443 0.13042425445322 1358 0.11991353923326 0.11991473153453 1361 0.12456395040140 0.12456399956828
1365 0.09789612928671 0.09789431467968 1368 0.08260013408198 0.08260222305115 1371 0.08924555254664 0.08924562616294
1375 0.05526377695512 0.05526060037598 1378 0.03881186270325 0.03881586882921 1381 0.04534846995487 0.04534871604109
1385 0.01427479196745 0.01426593988594 1388 0.00351053591177 0.00354544447856 1391 0.00571792128896 0.00571894559182

1395	0.13177364211724	0.13177436324512	1398	0.13915146714621	0.13915080245345	1399	0.13701663213118	0.13701672541751
1403	0.10061218807971	0.10061366302537	1406	0.11405271354075	0.11405130281864	1407	0.10909460241907	0.10909473031363
1411	0.05796566866476	0.05796812695941	1414	0.07476009548352	0.07475775859274	1415	0.06849919642491	0.06849942867776
1419	0.01438388530817	0.01438903034313	1422	0.02990618899601	0.02990229756286	1423	0.02496912805505	0.02497051748330
1427	0.13696332947999	0.13696293746972	1430	0.12944879535718	0.12944967327349	1433	0.13072998931971	0.13072987760096
1437	0.11138341046992	0.11138216099422	1440	0.09766250042398	0.09766417558510	1443	0.10126335998705	0.10126336575411
1447	0.07120017028542	0.07119808804983	1450	0.05393158489629	0.05393438477477	1453	0.05792581151041	0.05792580050629
1457	0.02480313499499	0.02479981603325	1460	0.00835406477241	0.00836052242587	1463	0.00956655088363	0.00956543089302
1467	0.11011388417766	0.11011489278544	1470	0.12140866920393	0.12140766383844	1471	0.11961108317550	0.11961128162549
1475	0.06964965378226	0.06965129896048	1478	0.08632424541746	0.08632249084902	1479	0.08279056508540	0.08279080947843
1483	0.02050178906139	0.02050427002337	1486	0.03932752253618	0.03932495408980	1487	0.03633194224359	0.03633249126146
1491	0.11673713166365	0.11673663554891	1494	0.10503329287795	0.10503461041355	1497	0.10602115913807	0.10602101141591
1501	0.08032438624530	0.08032298689430	1504	0.06294155634344	0.06294360755461	1507	0.06507691428851	0.06507683804024
1511	0.03111299708174	0.03111096543047	1514	0.01112203233083	0.01112544586079	1517	0.01184501107289	0.01184408484193
1521	0.07573336777530	0.07573456786920	1524	0.09154179389298	0.09154050438680	1525	0.09026847026665	0.09026875898306
1529	0.02413166908930	0.02413330939773	1532	0.04479717860004	0.04479525024587	1533	0.04313148049314	0.04313181767747
1537	0.08391873690481	0.08391817314443	1540	0.06730349895817	0.06730515385109	1543	0.06788706591391	0.06788688630284
1547	0.03466765313827	0.03466621657555	1550	0.01274071714261	0.01274309887747	1553	0.01312479947849	0.01312402508314
1557	0.02600341715387	0.02600476506314	1560	0.04724465149876	0.04724320568487	1561	0.04664722489382	0.04664757101135
1565	0.03605924523629	0.03605867528529	1568	0.01351922271267	0.01352125998214	1571	0.01362510608026	0.01362484287590
1575	0.14625828214478	0.14625799334796	1578	0.14398352999959	0.14398344465129	1581	0.14253524587494	0.14253564500567
1585	0.13696332947999	0.13696293746972	1588	0.13072998931971	0.13072987760096	1591	0.12944879535718	0.12944967327349
1595	0.11673713166365	0.11673663554891	1598	0.10602115913807	0.10602101141591	1601	0.10503329287795	0.10503461041355
1605	0.08391873690481	0.08391817314443	1608	0.06788706591391	0.06788688630284	1611	0.06730349895817	0.06730515385109
1615	0.03605924523629	0.03605867528529	1618	0.01362510608026	0.01362484287590	1621	0.01351922271267	0.01352125998214
1625	0.13177364211725	0.13177436324512	1628	0.13701663213118	0.13701672541751	1629	0.13915146714621	0.13915080245345
1633	0.11011388417766	0.11011489278544	1636	0.11961108317550	0.11961128162549	1637	0.12140866920393	0.12140766383844
1641	0.07573336777530	0.07573456786920	1644	0.09026847026665	0.09026875898306	1645	0.09154179389298	0.09154050438680
1649	0.02600341715387	0.02600476506314	1652	0.04664722489382	0.04664757101135	1653	0.04724465149876	0.04724320568487
1657	0.13042525501443	0.13042425445322	1660	0.12456395040140	0.12456399956828	1663	0.11991353923327	0.11991473153453
1667	0.11138341046992	0.11138216099422	1670	0.10126335998705	0.10126336575411	1673	0.09766250042398	0.09766417558510
1677	0.08032438624530	0.08032298689430	1680	0.06507691428851	0.06507683804024	1683	0.06294155634344	0.06294360755461

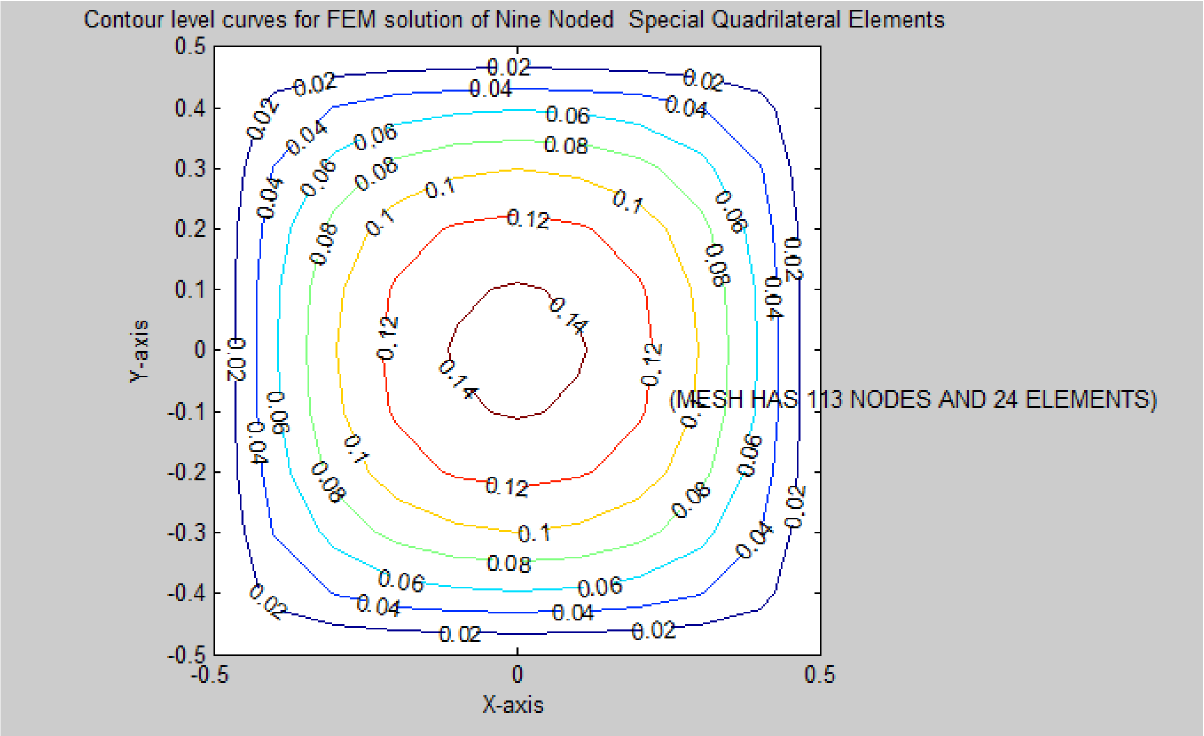
1687	0.03466765313827	0.03466621657555	1690	0.01312479947849	0.01312402508314	1693	0.01274071714261	0.01274309887747
1697	0.10061218807971	0.10061366302537	1700	0.10909460241907	0.10909473031363	1701	0.11405271354076	0.11405130281864
1705	0.06964965378226	0.06965129896048	1708	0.08279056508540	0.08279080947843	1709	0.08632424541746	0.08632249084902
1713	0.02413166908930	0.02413330939773	1716	0.04313148049314	0.04313181767747	1717	0.04479717860004	0.04479525024587
1721	0.09789612928672	0.09789431467968	1724	0.08924555254664	0.08924562616294	1727	0.08260013408198	0.08260222305115
1731	0.07120017028542	0.07119808804983	1734	0.05792581151041	0.05792580050629	1737	0.05393158489629	0.05393438477477
1741	0.03111299708174	0.03111096543047	1744	0.01184501107289	0.01184408484193	1747	0.01112203233083	0.01112544586079
1751	0.05796566866476	0.05796812695941	1754	0.06849919642491	0.06849942867776	1755	0.07476009548352	0.07475775859274
1759	0.02050178906139	0.02050427002337	1762	0.03633194224359	0.03633249126146	1763	0.03932752253618	0.03932495408980
1767	0.05526377695512	0.05526060037598	1770	0.04534846995487	0.04534871604109	1773	0.03881186270325	0.03881586882921
1777	0.02480313499499	0.02479981603325	1780	0.00956655088363	0.00956543089302	1783	0.00835406477241	0.00836052242587
1787	0.01438388530817	0.01438903034313	1790	0.02496912805505	0.02497051748330	1791	0.02990618899601	0.02990229756286
1795	0.01427479196745	0.01426593988594	1798	0.00571792128896	0.00571894559182	1801	0.00351053591177	0.00354544447856
1805	0.14625828214478	0.14625800572427	1808	0.14253524587494	0.14253563018103	1811	0.14398352999959	0.14398344064660
1815	0.13042525501443	0.13042426542688	1818	0.11991353923327	0.11991471180672	1821	0.12456395040140	0.12456398526623
1825	0.09789612928672	0.09789432420388	1828	0.08260013408198	0.08260219169659	1831	0.08924555254664	0.08924559062991
1835	0.05526377695512	0.05526060445204	1838	0.03881186270325	0.03881579588808	1841	0.04534846995487	0.04534860549301
1845	0.01427479196745	0.01426588254934	1848	0.00351053591177	0.00354461935573	1851	0.00571792128896	0.00571769637168
1855	0.13177364211725	0.13177436378611	1858	0.13915146714621	0.13915078118767	1859	0.13701663213118	0.13701672053274
1863	0.10061218807971	0.10061364601075	1866	0.11405271354076	0.11405126470678	1867	0.10909460241907	0.10909471747749
1871	0.05796566866476	0.05796805592010	1874	0.07476009548352	0.07475768010314	1875	0.06849919642491	0.06849939524539
1879	0.01438388530817	0.01438846454530	1882	0.02990618899601	0.02990204180532	1883	0.02496912805505	0.02497037634989
1887	0.13696332947999	0.13696293955392	1890	0.12944879535718	0.12944963926203	1893	0.13072998931971	0.13072985440945
1897	0.11138341046992	0.11138214338022	1900	0.09766250042398	0.09766410748812	1903	0.10126335998705	0.10126330308289
1907	0.07120017028542	0.07119802316049	1910	0.05393158489629	0.05393420943977	1913	0.05792581151041	0.05792562099281
1917	0.02480313499499	0.02479950701517	1920	0.00835406477241	0.00835905930447	1923	0.00956655088363	0.00956393024905
1927	0.11011388417766	0.11011485658400	1930	0.12140866920393	0.12140761543446	1931	0.11961108317550	0.11961124960253
1935	0.06964965378226	0.06965117955198	1938	0.08632424541746	0.08632238377176	1939	0.08279056508540	0.08279072767688
1943	0.02050178906139	0.02050356024535	1946	0.03932752253618	0.03932462391872	1947	0.03633194224359	0.03633220614761
1951	0.11673713166365	0.11673660904545	1954	0.10503329287795	0.10503452803291	1957	0.10602115913807	0.10602093985522
1961	0.08032438624531	0.08032289486677	1964	0.06294155634344	0.06294339547718	1967	0.06507691428851	0.06507663138857
1971	0.03111299708174	0.03111058744699	1974	0.01112203233083	0.01112388050008	1977	0.01184501107289	0.01184251532034

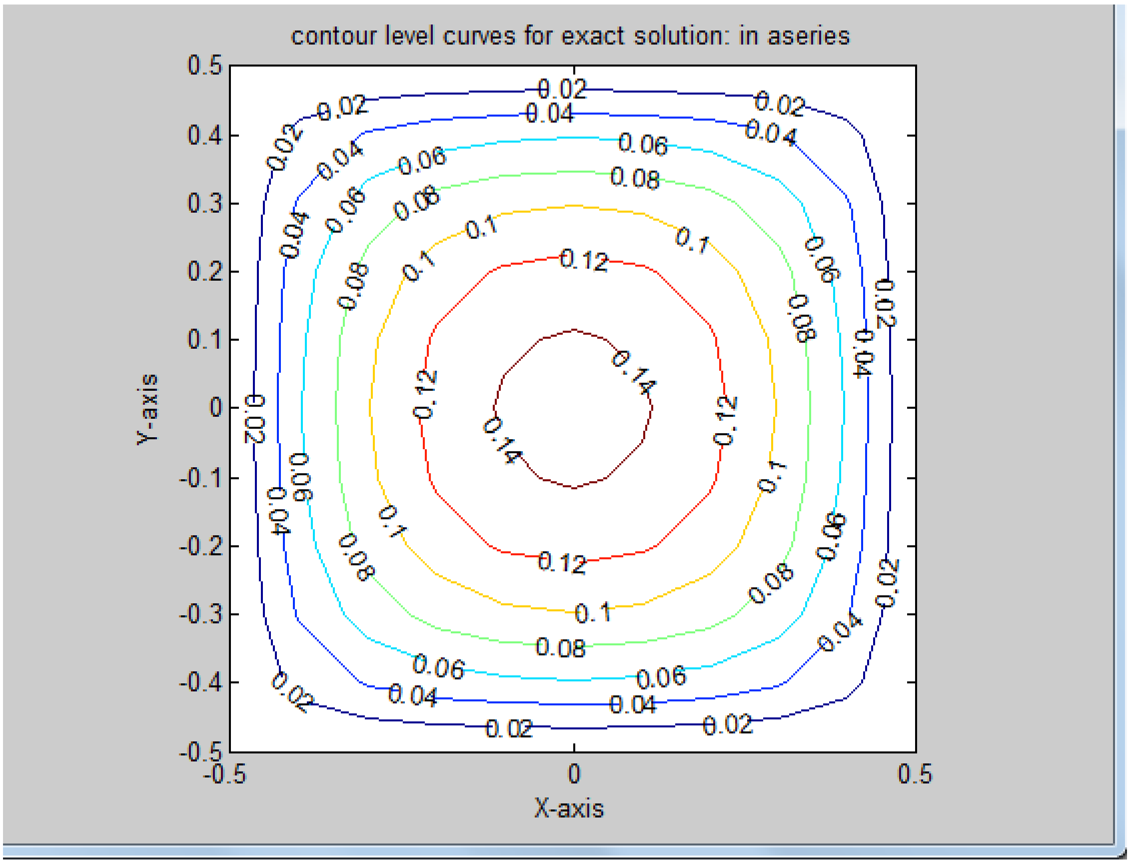
1981 0.07573336777530 0.07573442927389 1984 0.09154179389298 0.09154038701742 1985 0.09026847026665 0.09026865799470
1989 0.02413166908930 0.02413255125063 1992 0.04479717860004 0.04479489148713 1993 0.04313148049314 0.04313148419444
1997 0.08391873690481 0.08391807222744 2000 0.06730349895817 0.06730492749000 2003 0.06788706591391 0.06788667076170
2007 0.03466765313827 0.03466581145387 2010 0.01274071714261 0.01274149683065 2013 0.01312479947849 0.01312242845916
2017 0.02600341715387 0.02600398772927 2020 0.04724465149876 0.04724283663401 2021 0.04664722489382 0.04664721834150
2025 0.03605924523629 0.03605826127415 2028 0.01351922271267 0.01351964367429 2031 0.01362510608026 0.01362323737381
2035 0.14625828214478 0.14625800572427 2038 0.14398352999959 0.14398344064660 2041 0.14253524587494 0.14253563018103
2045 0.13696332947999 0.13696293955392 2048 0.13072998931971 0.13072985440945 2051 0.12944879535718 0.12944963926203
2055 0.11673713166365 0.11673660904545 2058 0.10602115913807 0.10602093985522 2061 0.10503329287795 0.10503452803291
2065 0.08391873690481 0.08391807222744 2068 0.06788706591390 0.06788667076170 2071 0.06730349895817 0.06730492749000
2075 0.03605924523629 0.03605826127415 2078 0.01362510608026 0.01362323737381 2081 0.01351922271267 0.01351964367429
2085 0.13177364211724 0.13177436378611 2088 0.13701663213118 0.13701672053274 2089 0.13915146714621 0.13915078118767
2093 0.11011388417766 0.11011485658400 2096 0.11961108317550 0.11961124960253 2097 0.12140866920393 0.12140761543446
2101 0.07573336777530 0.07573442927389 2104 0.09026847026665 0.09026865799470 2105 0.09154179389298 0.09154038701742
2109 0.02600341715387 0.02600398772927 2112 0.04664722489382 0.04664721834150 2113 0.04724465149876 0.04724283663401
2117 0.13042525501443 0.13042426542688 2120 0.12456395040140 0.12456398526623 2123 0.11991353923326 0.11991471180672
2127 0.11138341046992 0.11138214338022 2130 0.10126335998705 0.10126330308289 2133 0.09766250042398 0.09766410748812
2137 0.08032438624530 0.08032289486677 2140 0.06507691428851 0.06507663138857 2143 0.06294155634344 0.06294339547718
2147 0.03466765313827 0.03466581145387 2150 0.01312479947849 0.01312242845916 2153 0.01274071714261 0.01274149683065
2157 0.10061218807971 0.10061364601075 2160 0.10909460241907 0.10909471747749 2161 0.11405271354075 0.11405126470678
2165 0.06964965378226 0.06965117955198 2168 0.08279056508540 0.08279072767688 2169 0.08632424541746 0.08632238377176
2173 0.02413166908930 0.02413255125063 2176 0.04313148049314 0.04313148419444 2177 0.04479717860003 0.04479489148713
2181 0.09789612928671 0.09789432420388 2184 0.08924555254664 0.08924559062991 2187 0.08260013408198 0.08260219169659
2191 0.07120017028542 0.07119802316049 2194 0.05792581151041 0.05792562099281 2197 0.05393158489629 0.05393420943977
2201 0.03111299708174 0.03111058744699 2204 0.01184501107289 0.01184251532034 2207 0.01112203233083 0.01112388050008
2211 0.05796566866476 0.05796805592010 2214 0.06849919642491 0.06849939524539 2215 0.07476009548352 0.07475768010314
2219 0.02050178906139 0.02050356024535 2222 0.03633194224359 0.03633220614761 2223 0.03932752253618 0.03932462391872
2227 0.05526377695512 0.05526060445204 2230 0.04534846995487 0.04534860549301 2233 0.03881186270325 0.03881579588808
2237 0.02480313499499 0.02479950701517 2240 0.00956655088363 0.00956393024905 2243 0.00835406477241 0.00835905930447
2247 0.01438388530817 0.01438846454530 2250 0.02496912805505 0.02497037634989 2251 0.02990618899601 0.02990204180532
2255 0.01427479196745 0.01426588254934 2258 0.00571792128896 0.00571769637168 2261 0.00351053591177 0.00354461935573
2264 0.14625828214478 0.14625799334796 2267 0.14253524587494 0.14253564500567 2269 0.14398352999959 0.14398344465129

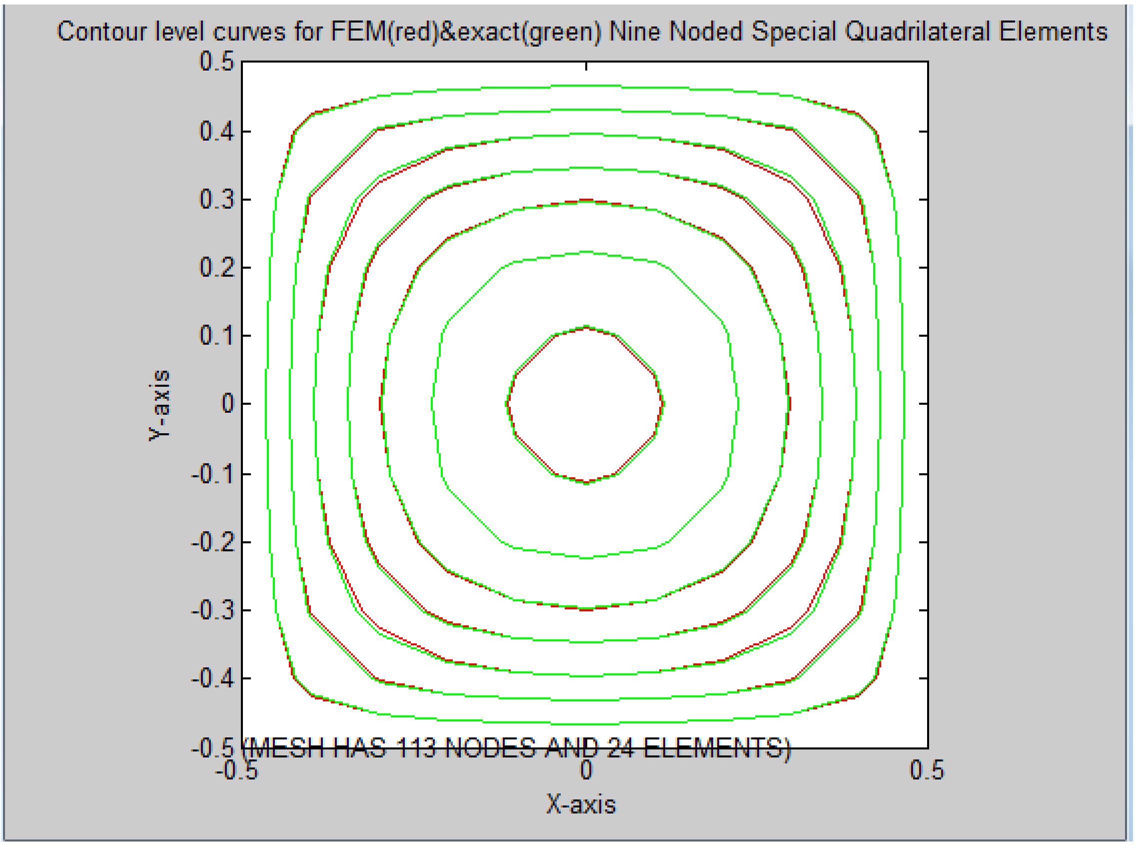
```
 2273    0.13042525501443    0.13042425445322  2276    0.11991353923327    0.11991473153453  2279    0.12456395040140
0.12456399956828
 2283    0.09789612928671    0.09789431467968  2286    0.08260013408198    0.08260222305115  2289    0.08924555254664
0.08924562616294
 2293    0.05526377695512    0.05526060037598  2296    0.03881186270325    0.03881586882921  2299    0.04534846995487
0.04534871604109
 2303    0.01427479196745    0.01426593988594  2306    0.00351053591177    0.00354544447856  2309    0.00571792128896
0.00571894559182
 2313    0.13177364211724    0.13177436324512  2316    0.13915146714621    0.13915080245345  2317    0.13701663213118
0.13701672541751
 2321    0.10061218807971    0.10061366302537  2324    0.11405271354075    0.11405130281864  2325    0.10909460241907
0.10909473031363
 2329    0.05796566866476    0.05796812695941  2332    0.07476009548352    0.07475775859274  2333    0.06849919642491
0.06849942867776
 2337    0.01438388530817    0.01438903034313  2340    0.02990618899601    0.02990229756286  2341    0.02496912805505
0.02497051748330
 2344    0.13696332947999    0.13696293746972  2347    0.12944879535718    0.12944967327349  2349    0.13072998931971
0.13072987760096
 2353    0.11138341046992    0.11138216099422  2356    0.09766250042398    0.09766417558510  2359    0.10126335998705
0.10126336575411
 2363    0.07120017028542    0.07119808804983  2366    0.05393158489629    0.05393438477477  2369    0.05792581151041
0.05792580050629
 2373    0.02480313499499    0.02479981603325  2376    0.00835406477241    0.00836052242587  2379    0.00956655088363
0.00956543089302
 2383    0.11011388417766    0.11011489278544  2386    0.12140866920393    0.12140766383844  2387    0.11961108317550
0.11961128162549
 2391    0.06964965378226    0.06965129896048  2394    0.08632424541746    0.08632249084902  2395    0.08279056508540
0.08279080947843
 2399    0.02050178906139    0.02050427002337  2402    0.03932752253618    0.03932495408980  2403    0.03633194224359
0.03633249126146
 2406    0.11673713166364    0.11673663554891  2409    0.10503329287795    0.10503461041355  2411    0.10602115913807
0.10602101141591
 2415    0.08032438624530    0.08032298689430  2418    0.06294155634344    0.06294360755461  2421    0.06507691428851
0.06507683804024
 2425    0.03111299708174    0.03111096543047  2428    0.01112203233083    0.01112544586079  2431    0.01184501107289
0.01184408484193
 2435    0.07573336777530    0.07573456786920  2438    0.09154179389298    0.09154050438680  2439    0.09026847026665
0.09026875898306
 2443    0.02413166908930    0.02413330939773  2446    0.04479717860004    0.04479525024587  2447    0.04313148049314
0.04313181767747
 2450    0.08391873690481    0.08391817314443  2453    0.06730349895817    0.06730515385109  2455    0.06788706591391
0.06788688630284
 2459    0.03466765313827    0.03466621657555  2462    0.01274071714261    0.01274309887747  2465    0.01312479947849
0.01312402508314
 2469    0.02600341715387    0.02600476506314  2472    0.04724465149876    0.04724320568487  2473    0.04664722489382
0.04664757101135
 2476    0.03605924523629    0.03605867528529  2479    0.01351922271267    0.01352125998214  2481    0.01362510608026
0.01362484287590
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
----------------------------------------------
```

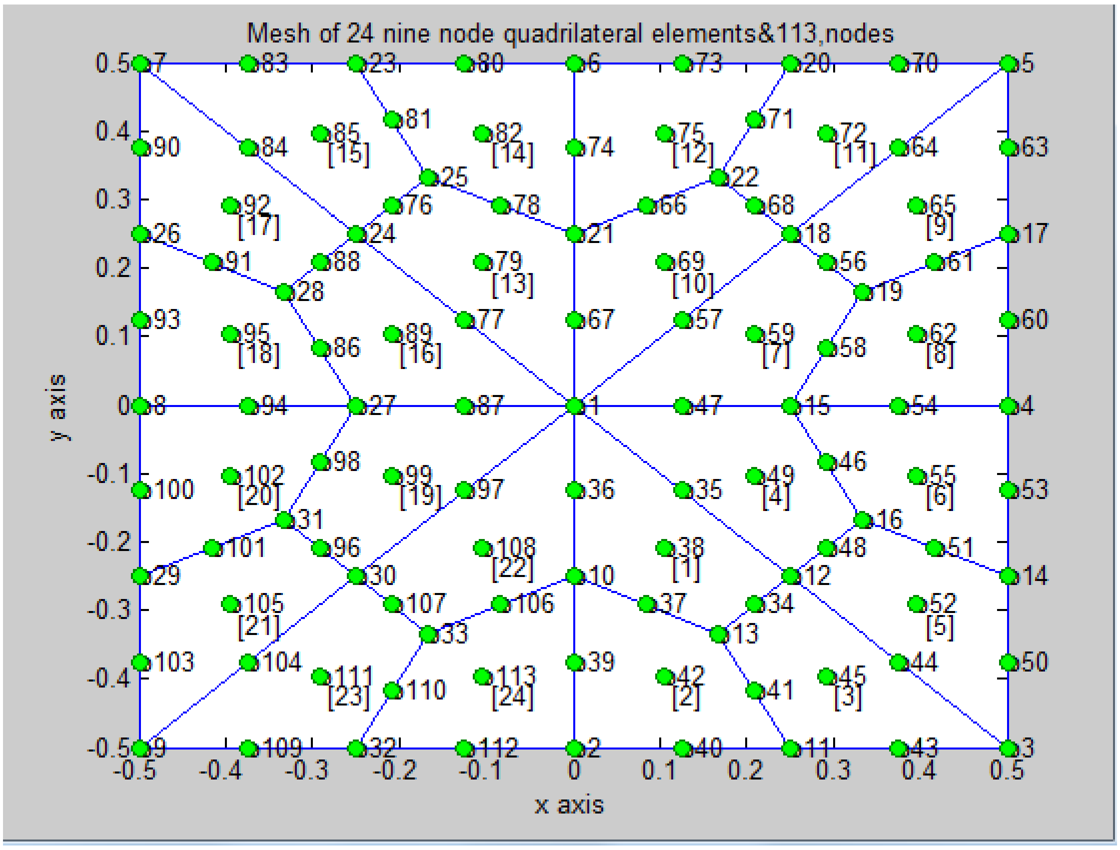
Mesh No.1

Mesh with 24 nine noded quadrilateral elements & no. of nodes = 113



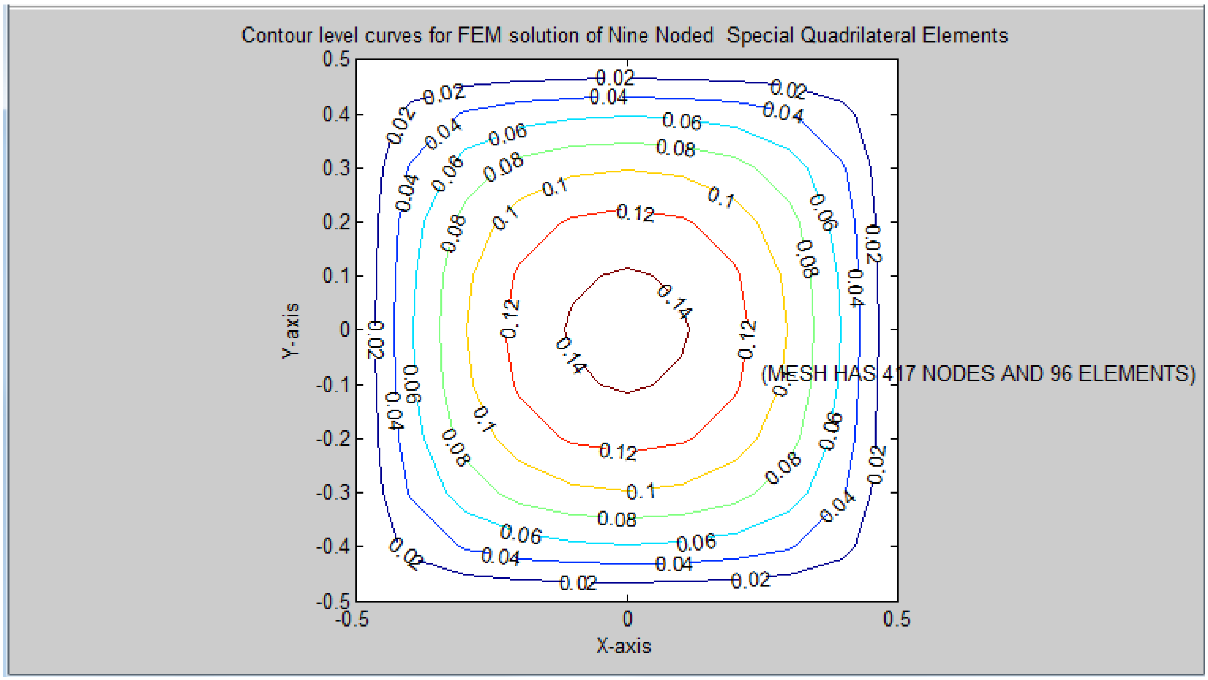


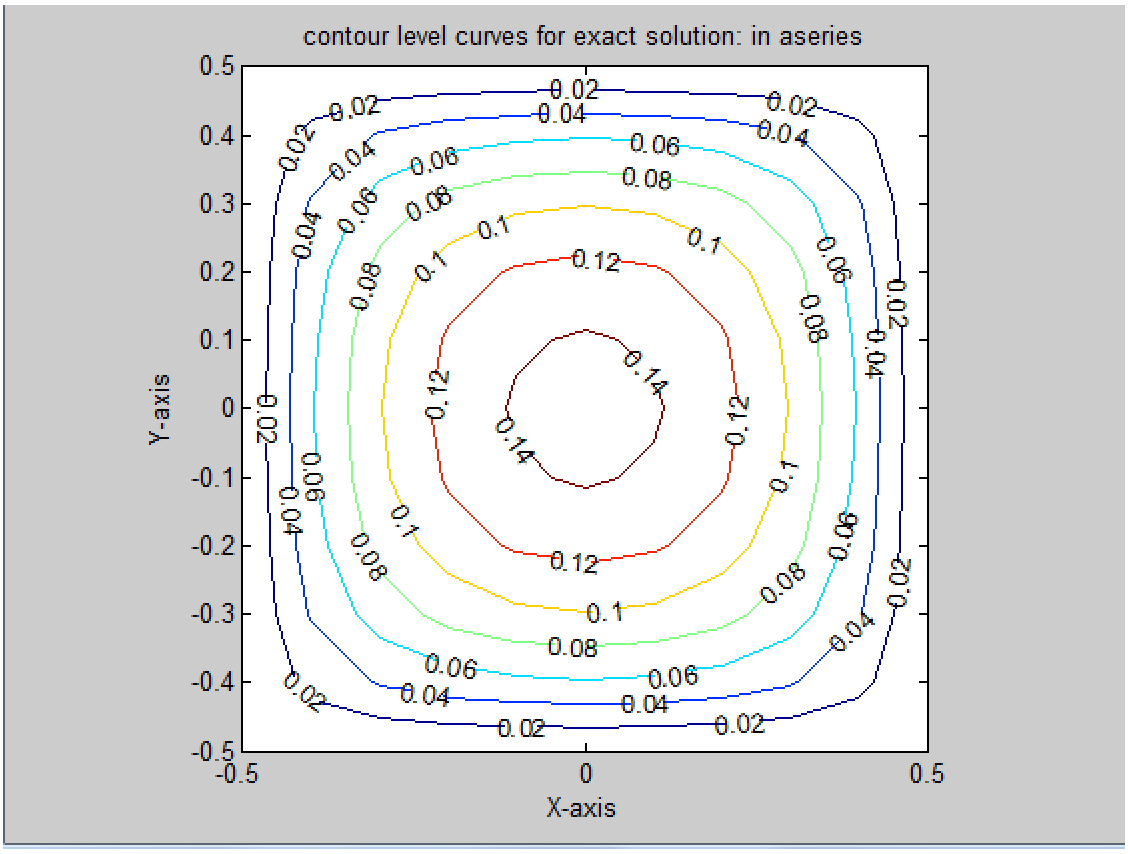


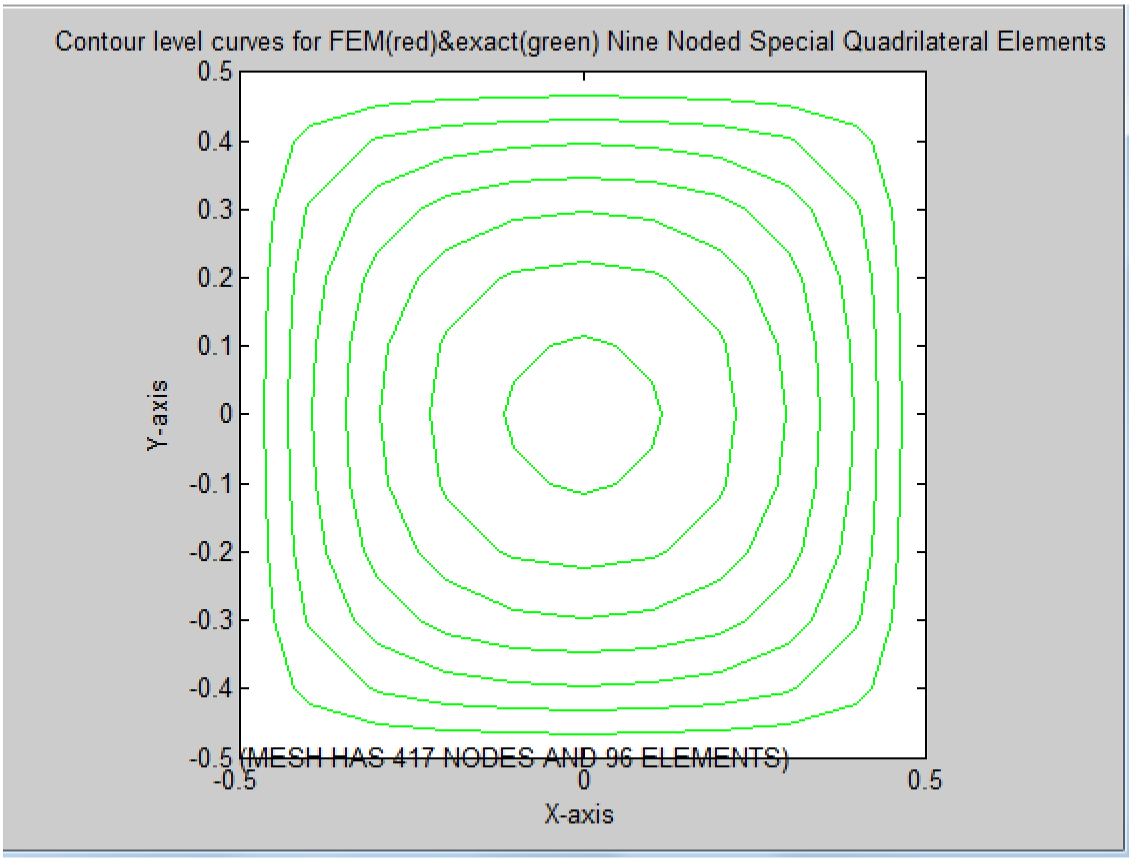


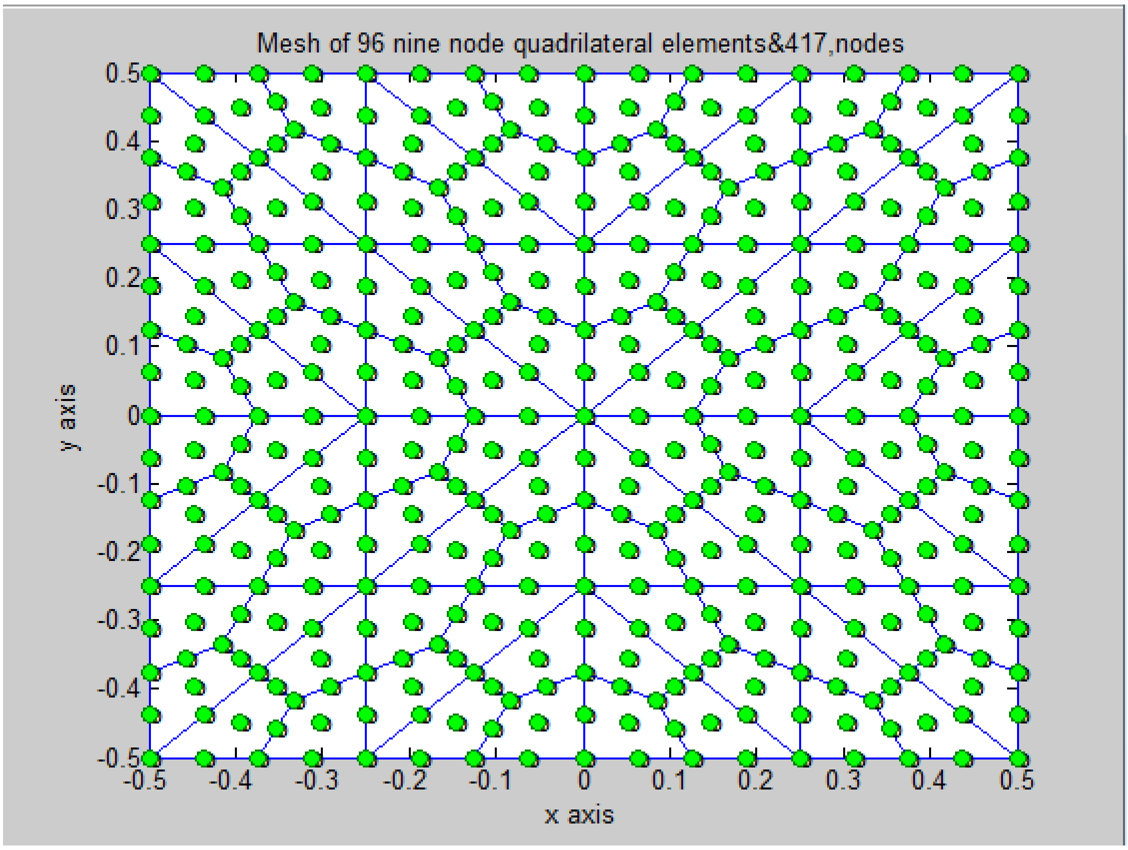
Mesh No.2

Mesh with 96 nine noded quadrilateral elements & no. of nodes = 417


Contour level curves for FEM solution of Nine Noded Special Quadrilateral Elements
(MESH HAS 417 NODES AND 96 ELEMENTS)
Y-axis
X-axis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.5
0
0.5
0.02
0.04
0.06
0.08
0.1
0.12
0.14

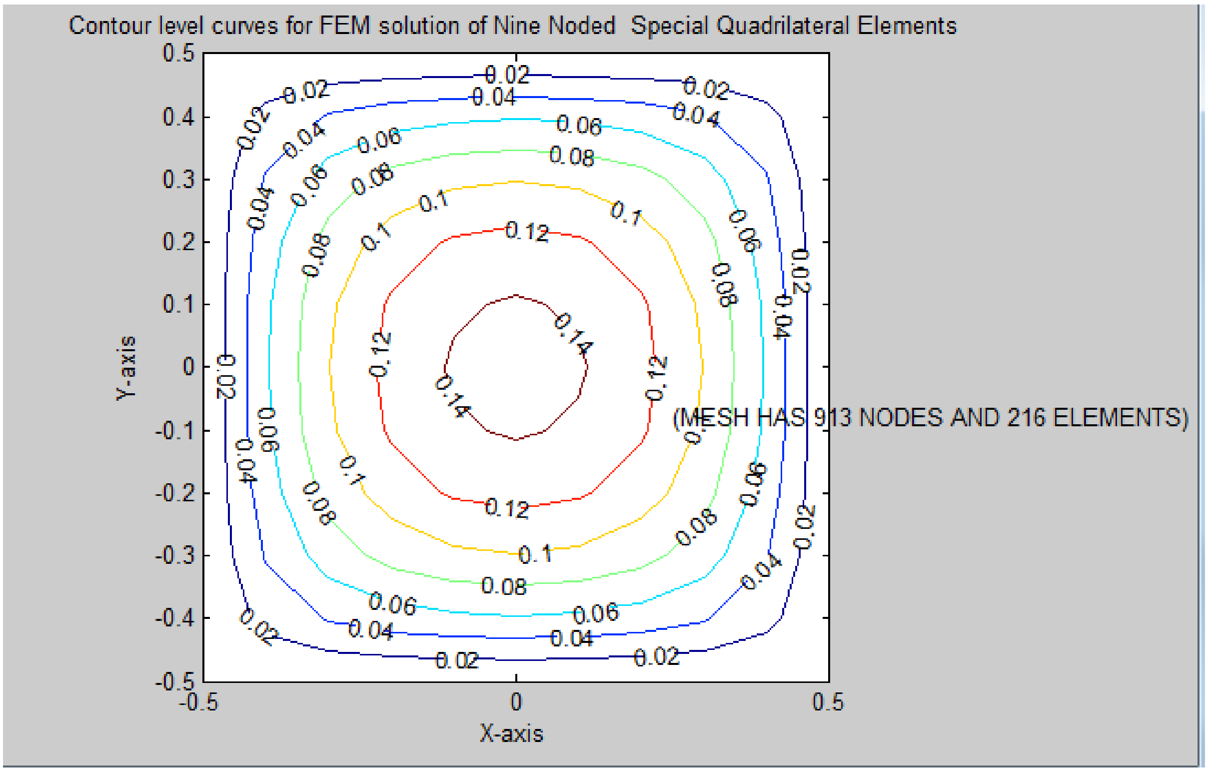

contour level curves for exact solution: in aseries
Y-axis
X-axis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.5
0
0.5
0.02
0.04
0.06
0.08
0.1
0.12
0.14

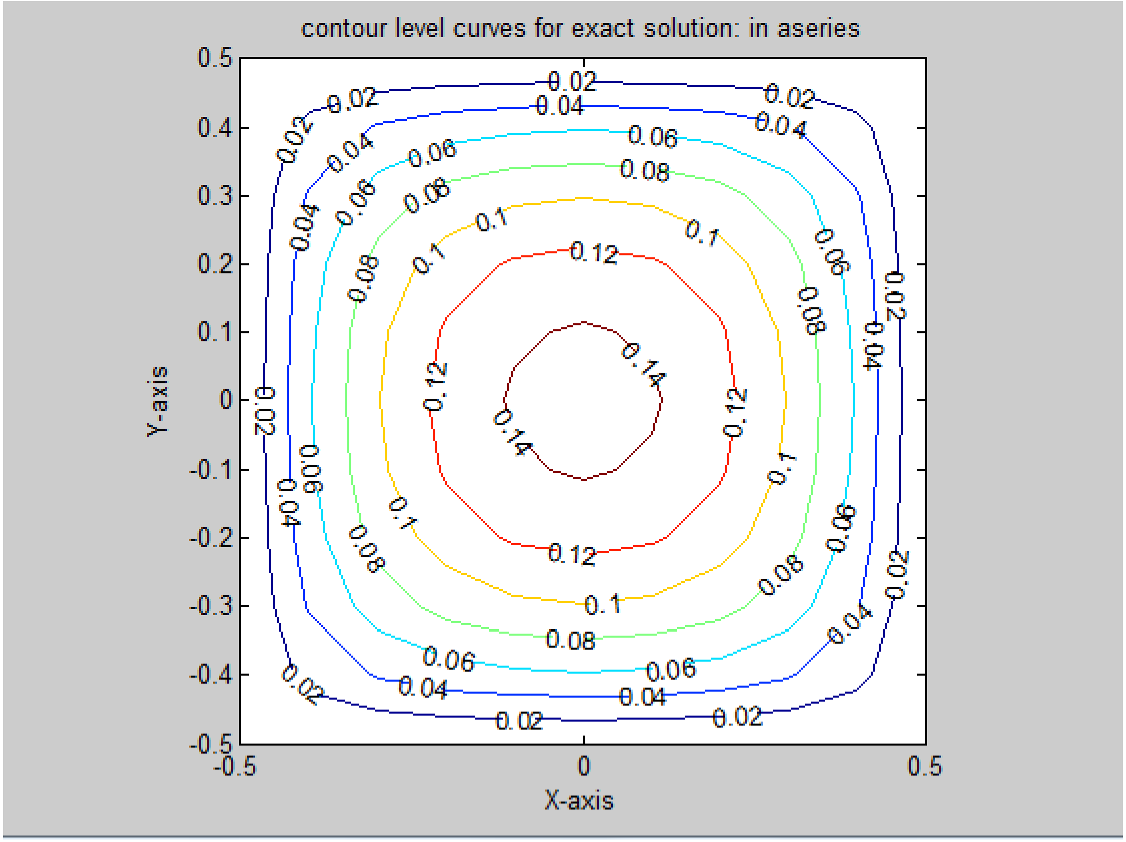


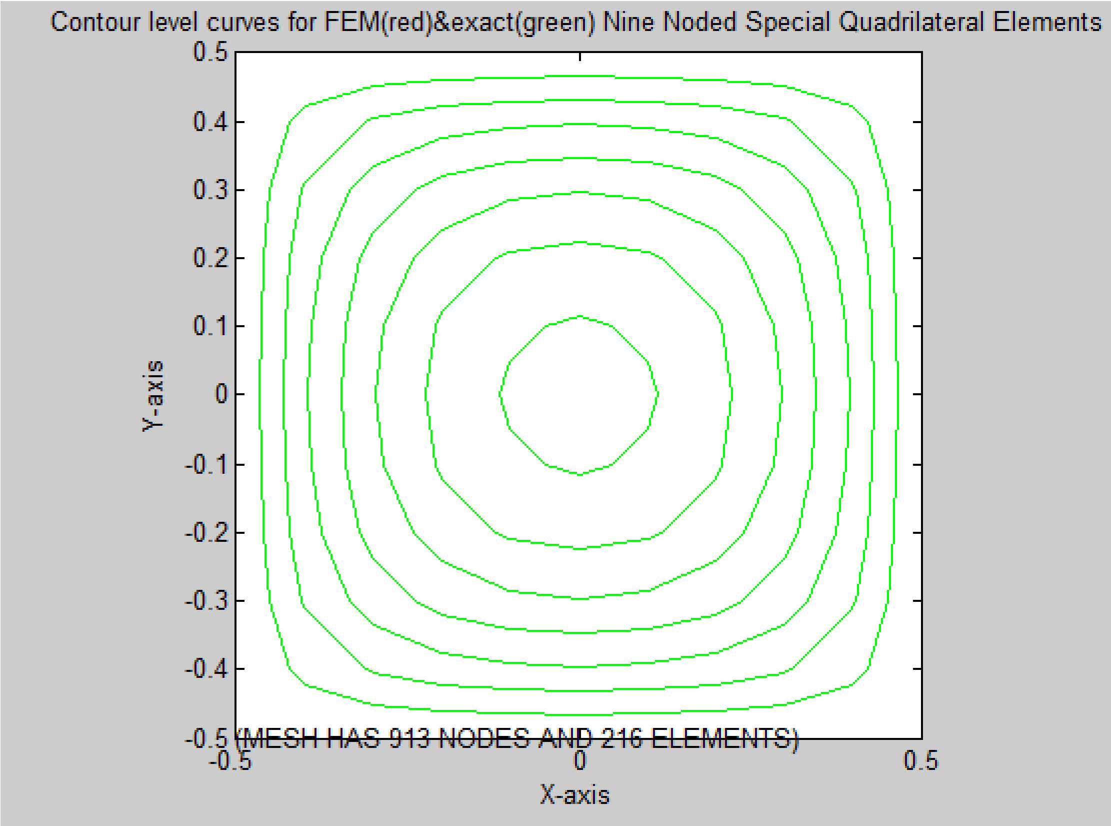


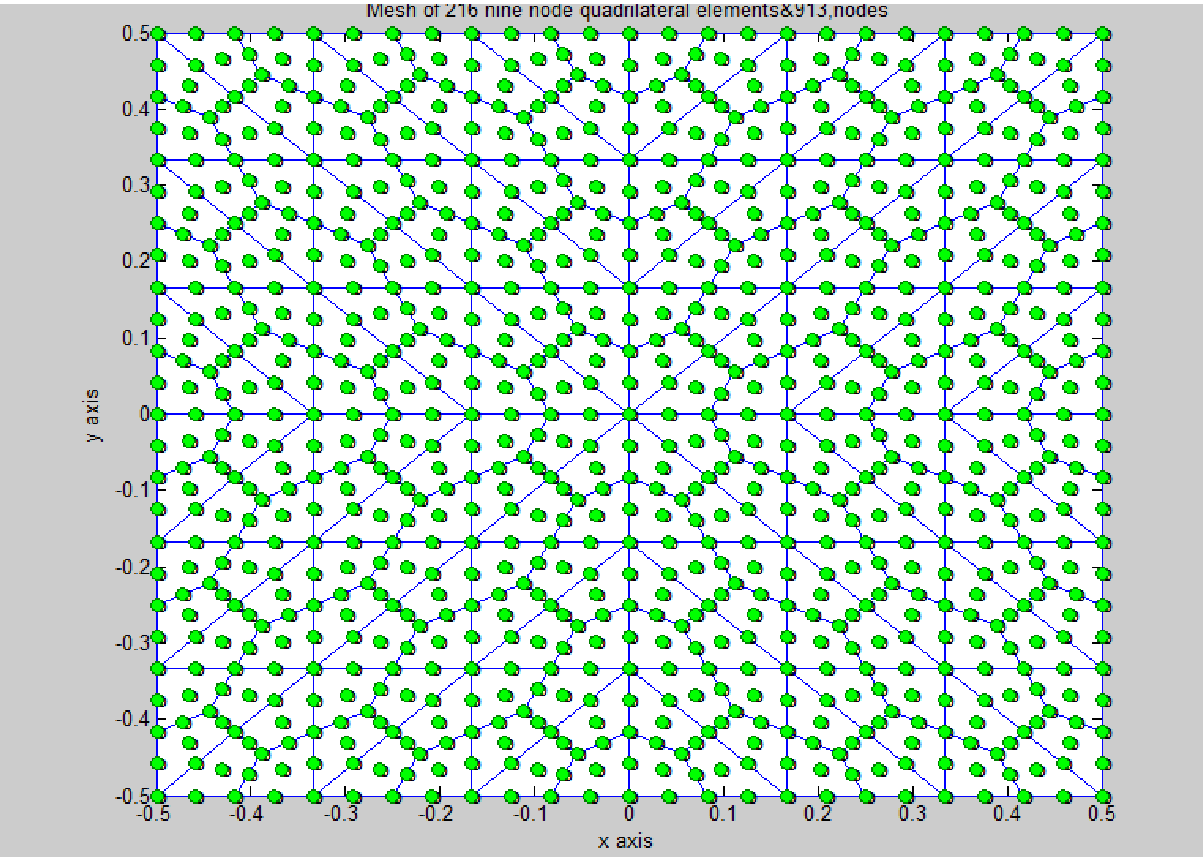
Mesh No.3

Mesh with 216 nine noded quadrilateral elements & no. of nodes = 913


Contour level curves for FEM solution of Nine Noded Special Quadrilateral Elements
(MESH HAS 913 NODES AND 216 ELEMENTS)
Y-axis
X-axis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.5
0
0.5
0.02
0.04
0.06
0.08
0.1
0.12
0.14

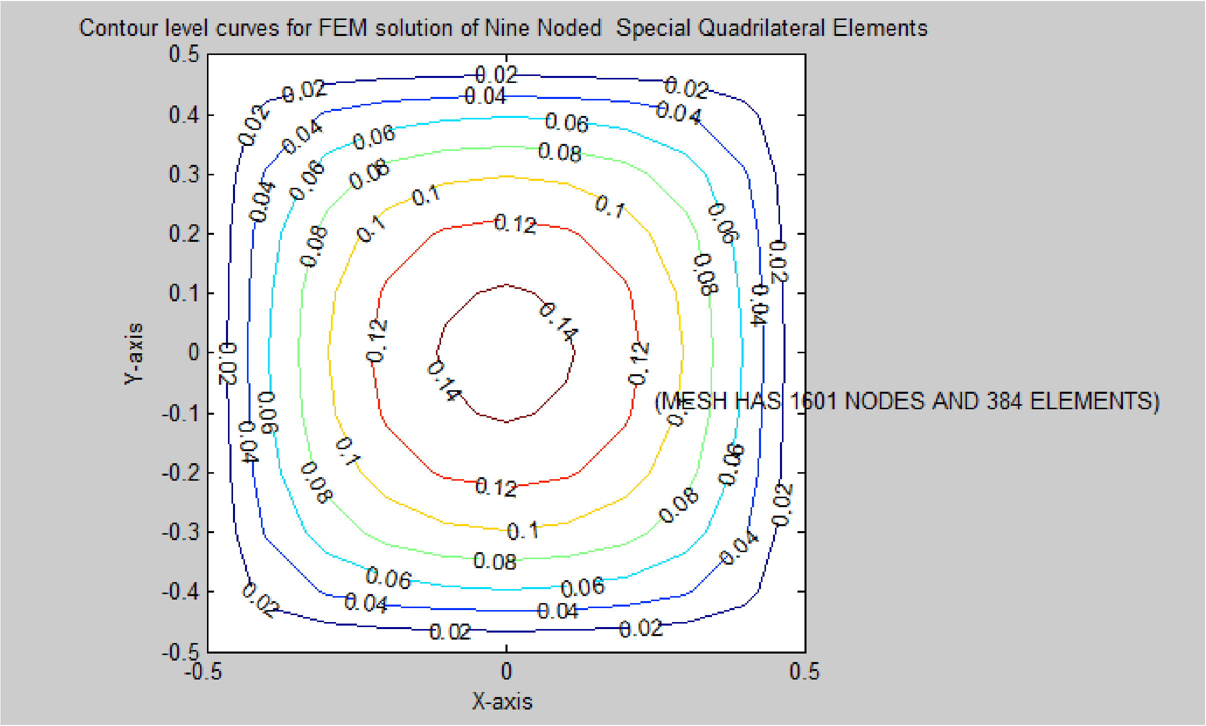

contour level curves for exact solution: in aseries
Y-axis
X-axis
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.5
0
0.5
0.02
0.04
0.06
0.08
0.1
0.12
0.14

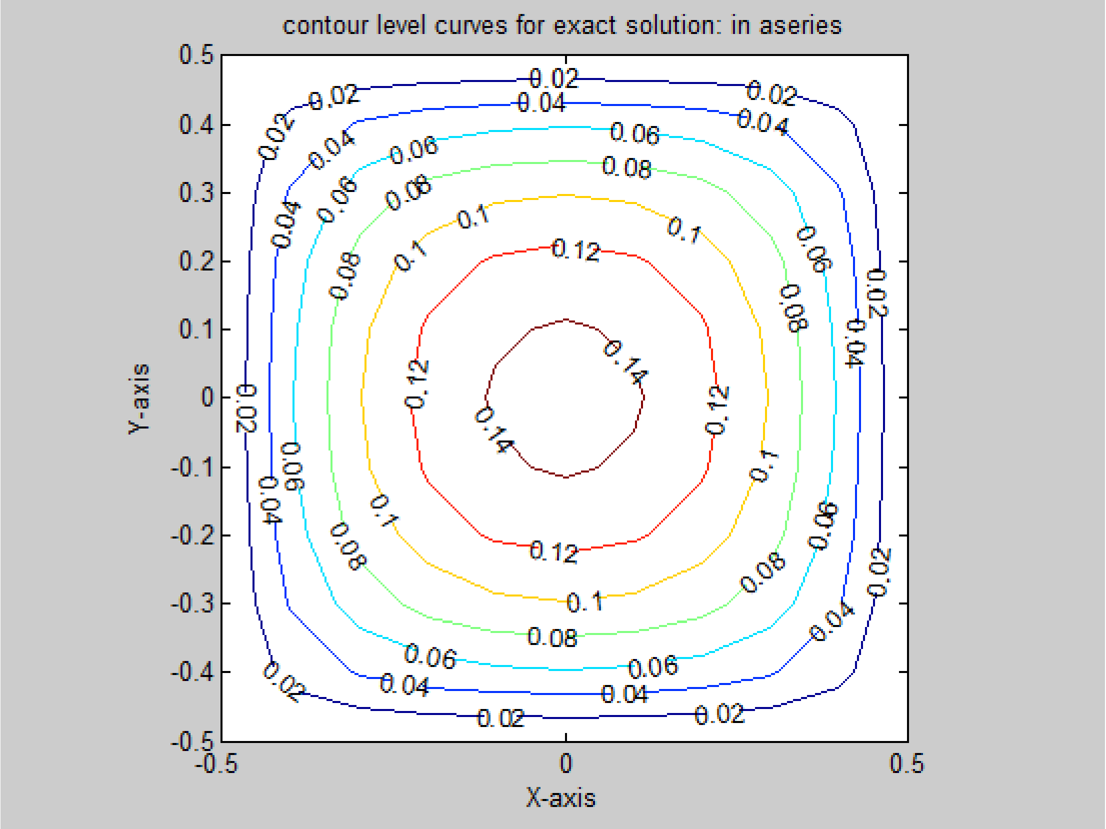


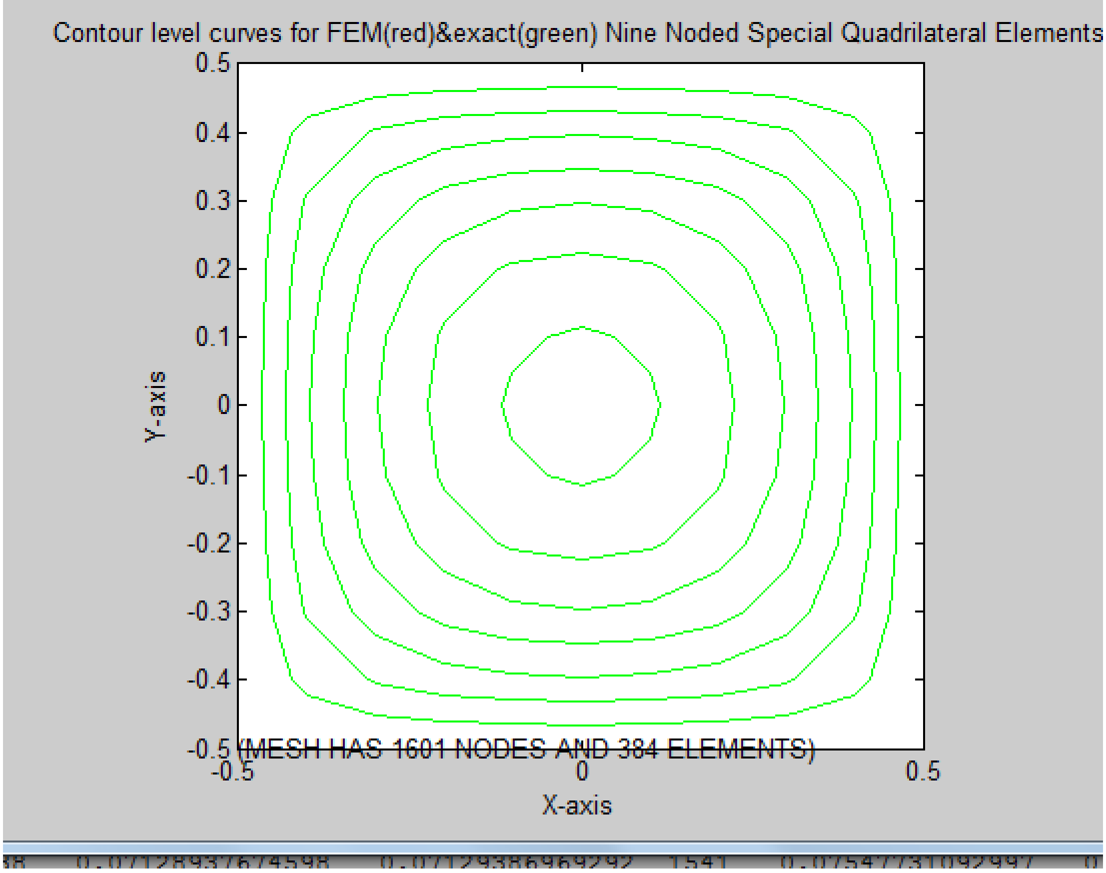


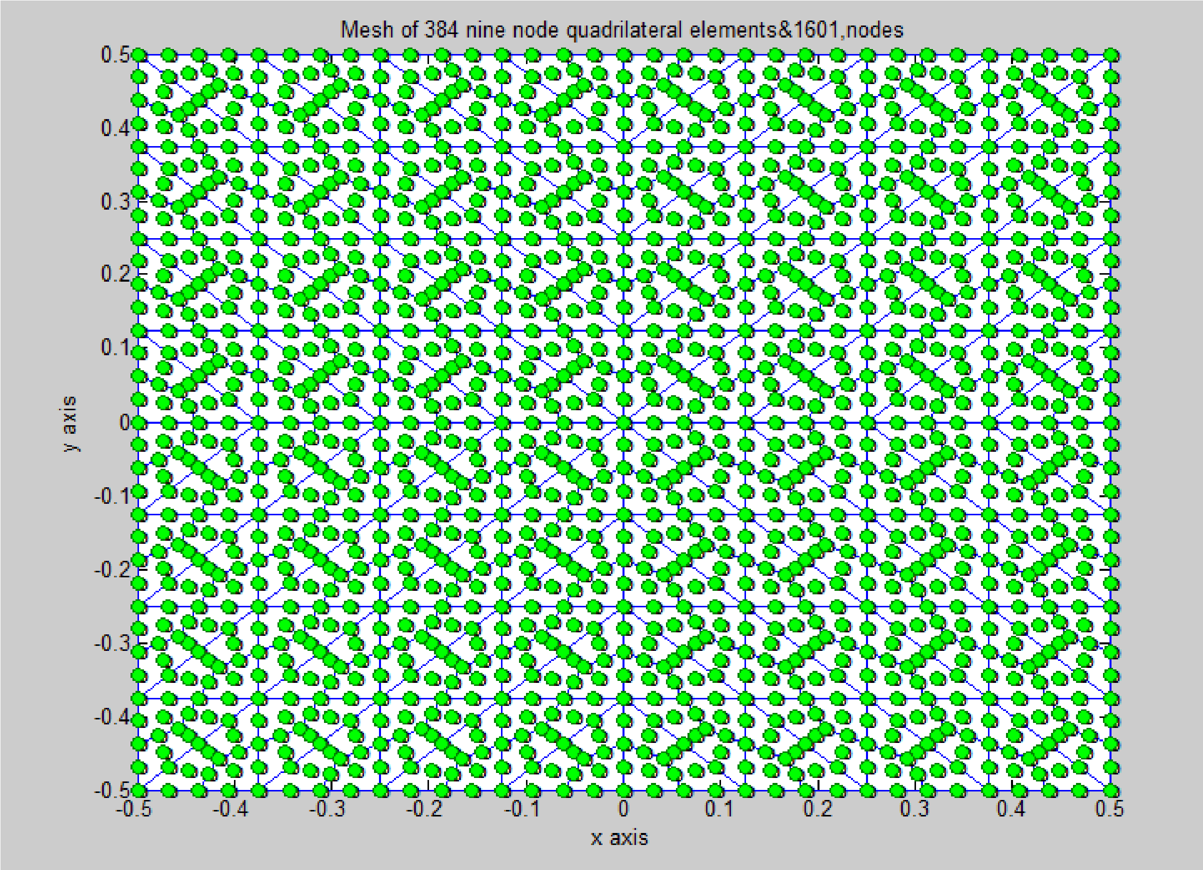
Mesh No.4

Mesh with 384 nine noded quadrilateral elements & no. of nodes = 1601


Contour level curves for FEM solution of Nine Noded Special Quadrilateral Elements
(MESH HAS 1601 NODES AND 384 ELEMENTS)
X-axis
Y-axis
-0.5
0
0.5
0.4
0.3
0.2
0.1
-0.1
-0.2
-0.3
-0.4
0.02
0.04
0.06
0.08
0.1
0.12
0.14

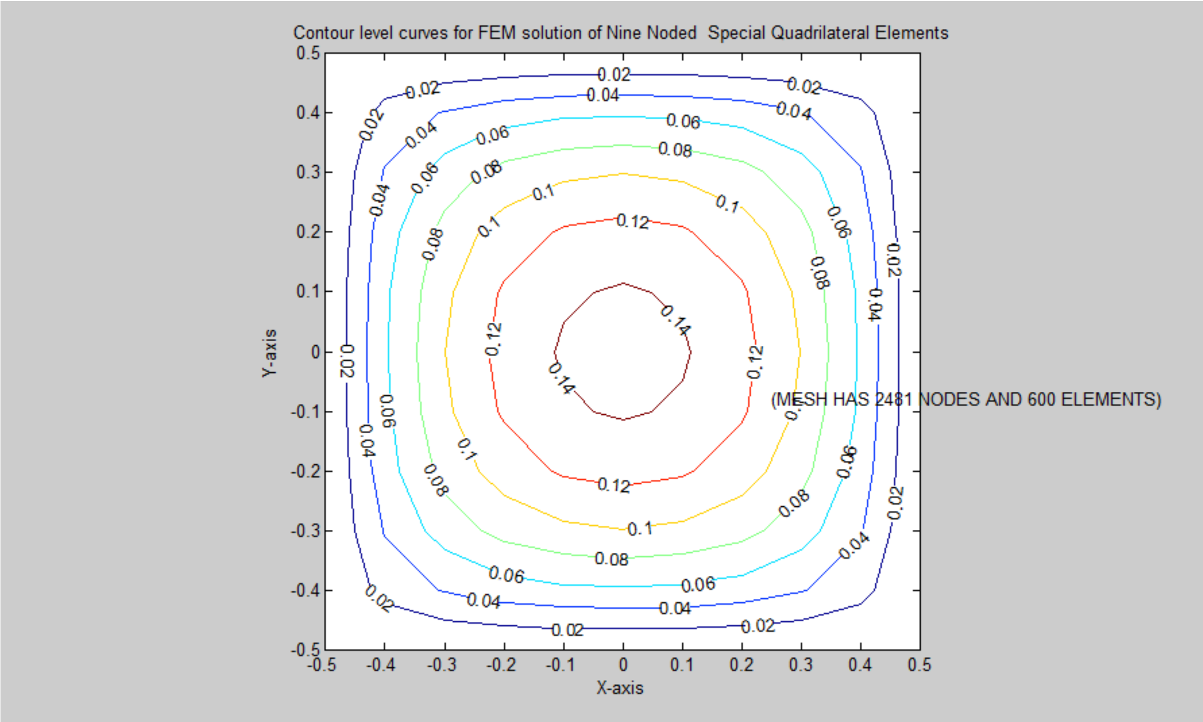

contour level curves for exact solution: in aseries
X-axis
Y-axis
-0.5
0
0.5
0.4
0.3
0.2
0.1
-0.1
-0.2
-0.3
-0.4
0.02
0.04
0.06
0.08
0.1
0.12
0.14

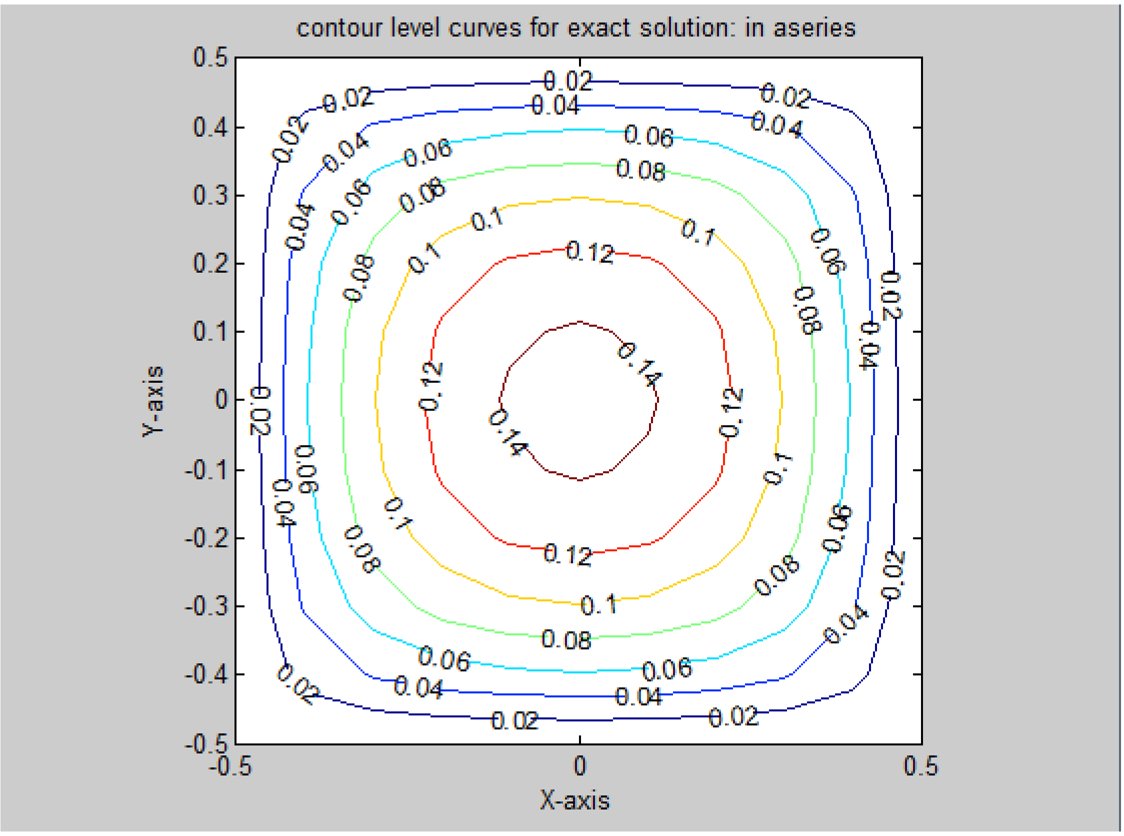


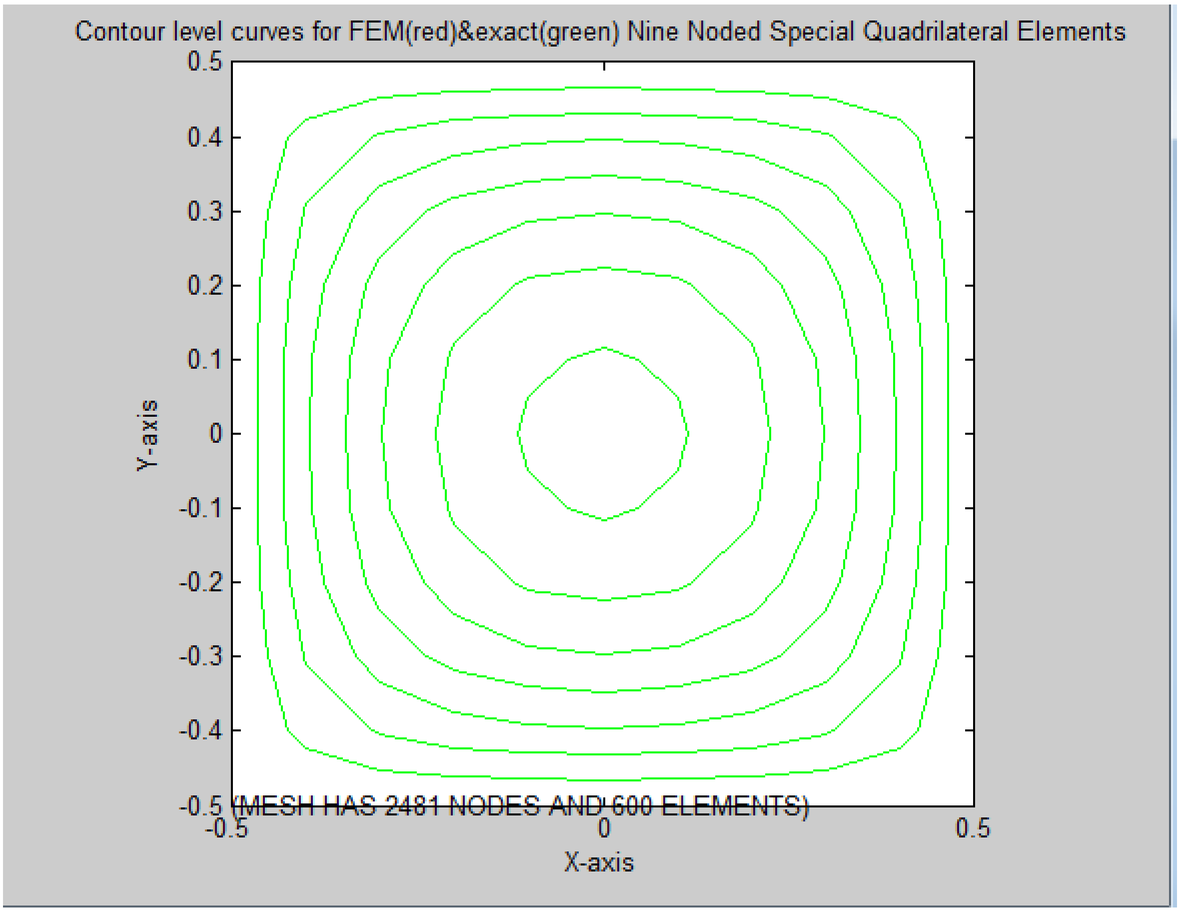


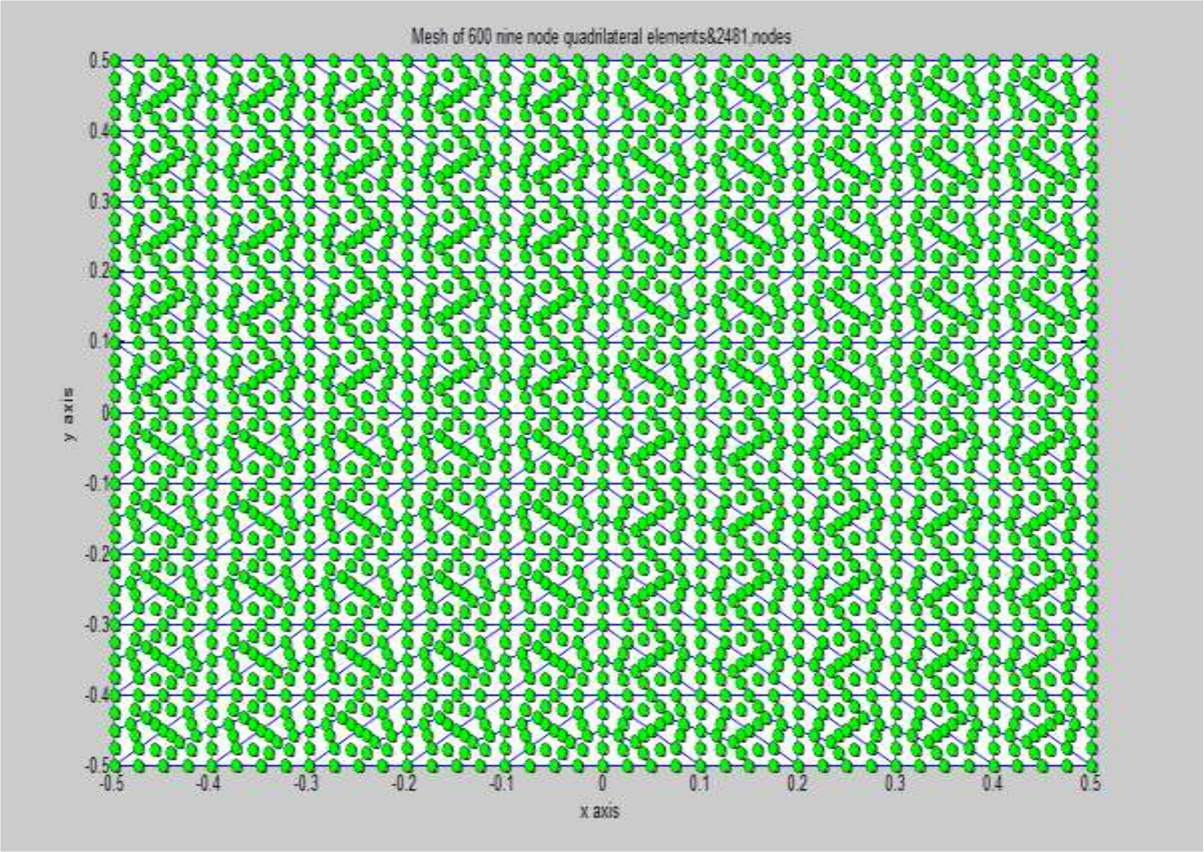
Mesh No.5

Mesh with 600 nine noded quadrilateral elements & no. of nodes = 2481


Contour level curves for FEM solution of Nine Noded Special Quadrilateral Elements
(MESH HAS 2481 NODES AND 600 ELEMENTS)
0.02
0.04
0.06
0.08
0.1
0.12
0.14
X-axis
Y-axis
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5


contour level curves for exact solution: in aseries
0.02
0.04
0.06
0.08
0.1
0.12
0.14
X-axis
Y-axis
-0.5
0
0.5
-0.4
-0.3
-0.2
-0.1
0.1
0.2
0.3
0.4





Solution of Poisson Boundary Value Problems Over Polygonal Domains

Example 3(pentagonal domain,refer section 7.5.2)

Table-5a

Mesh:number of nodes=2171,number of nine node elements= 525

fem computed values and exact values at centroid points

NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION
566	0.97339958515641	0.98932210182092	570	0.95128879706595	0.96715571192955	573	0.93817819513447	0.95300104853184
578	0.88500594743287	0.90065277997236	582	0.82919571310024	0.84391705351846	585	0.81814735147050	0.83156603115381
590	0.71011969844479	0.72382128880320	594	0.62610883478952	0.63806991399311	597	0.61825668593475	0.62873153678521
602	0.46598426618935	0.47613712672652	606	0.36208882054412	0.36976404559162	609	0.35818126088466	0.36435241896587
614	0.17690823155835	0.18184534524342	618	0.06378624279181	0.06526309611003	621	0.06363375240841	0.06430794778558
625	0.83842186999125	0.85177670957122	628	0.88604947680321	0.90023092787438	629	0.90539039234496	0.92086345838978
633	0.65646682745044	0.66717453160046	636	0.73289843597119	0.74518526832117	637	0.74846649660232	0.76226428361846
641	0.41053710106666	0.41726466199836	644	0.50826615132126	0.51719568269362	645	0.51852140520908	0.52904936975901
649	0.12481313491872	0.12651002002600	652	0.23424557185042	0.23857938012980	653	0.23831180355192	0.24404741748384
657	0.82511110496604	0.83832983438577	660	0.77308797097766	0.78552007992821	663	0.73267392432981	0.74324008236778
667	0.66256766386496	0.67373464520468	670	0.58427563230040	0.59391705351846	673	0.55409502597724	0.56194993744403
677	0.43541701487237	0.44318961476565	680	0.33850562840281	0.34417728784688	683	0.32144462583117	0.32565221730781
687	0.16586680549736	0.16926209694974	690	0.05987946679785	0.06074705121667	693	0.05721403528346	0.05747738918909
697	0.57417840859553	0.58216061586144	700	0.64094709214368	0.65023092787438	701	0.68273685835609	0.69394532362209
705	0.35946587895553	0.36409521212307	708	0.44490196186630	0.45129264217493	709	0.47350325860941	0.48163260958091
713	0.10945050000013	0.11038963222158	716	0.20533875887391	0.20817868831867	717	0.21807345125743	0.22217433998223
721	0.54952152304417	0.55769816034822	724	0.48464462950258	0.49162745378198	727	0.43507174056087	0.44016076289038
731	0.36143500154923	0.36685961542790	734	0.28106288593779	0.28490005914351	737	0.25261886583279	0.25507490766725
741	0.13784329931033	0.14011029528825	744	0.04977249279193	0.05028466170064	747	0.04500592766131	0.04502054327026
751	0.27260533083388	0.27528558608466	754	0.33735981941603	0.34121393349943	755	0.38144239886448	0.38707029384531
759	0.08308995827599	0.08346353808552	762	0.15586612644483	0.15740001602869	763	0.17585406406789	0.17855329010351
767	0.25149612002628	0.25461884087082	770	0.19559888458441	0.19773482763574	773	0.15856891026921	0.15952908885283
777	0.09602824615462	0.09724352171812	780	0.03467986620585	0.03490005914351	783	0.02828773746520	0.02815677289956
787	0.04838860559931	0.04836745131698	790	0.09076720244006	0.09121393349943	791	0.11605387179739	0.11745420013553
795	0.04449130719781	0.04485787470676	798	0.01607078784449	0.01609919563440	801	0.00864157206935	0.00853682141769
805	0.97386435069014	0.98932210182092	808	0.93855726981600	0.95300104853184	811	0.95272931193375	0.96715571192955
815	0.82539562673207	0.83832983438577	818	0.73286050846179	0.74324008236778	821	0.77449833403681	0.78552007992821
825	0.54959545755408	0.55769816034822	828	0.43507596060127	0.44016076289038	831	0.48567114575503	0.49162745378198
835	0.25144384079420	0.25461884087082	838	0.15850350099382	0.15952908885283	841	0.19617345173084	0.19773482763574
845	0.04443254204524	0.04485787470676	848	0.00858368179911	0.00853682141769	851	0.01618249531988	0.01609919563440
855	0.83984776287391	0.85177670957122	858	0.90720788005848	0.92086345838978	859	0.88651317081068	0.90023092787438
863	0.57496561675682	0.58216061586144	866	0.68399553729074	0.69394532362209	867	0.64085763216907	0.65023092787438
871	0.27282119980680	0.27528558608466	874	0.38206595977850	0.38707029384531	875	0.33695306955408	0.34121393349943
879	0.04834918369049	0.04836745131698	882	0.11629010887523	0.11745420013553	883	0.09042756536073	0.09121393349943
887	0.88763836897989	0.90065277997236	890	0.82017306966673	0.83156603115380	893	0.83268063276447	0.84391705351846
897	0.66418083232643	0.67373464520468	900	0.55504530409589	0.56194993744403	903	0.58666317242474	0.59391705351846
907	0.36202536997823	0.36685961542790	910	0.25271759530647	0.25507490766725	913	0.28218201733783	0.28490005914351
917	0.09610574114357	0.09724352171812	920	0.02812097307258	0.02815677289956	923	0.03492600749796	0.03490005914351
927	0.65894418163777	0.66717453160046	930	0.75196970225667	0.76226428361846	931	0.73468830577483	0.74518526832117
935	0.36033783242736	0.36409521212307	938	0.47536521559936	0.48163260958091	939	0.44530734289948	0.45129264217493
943	0.08309231886438	0.08346353808552	946	0.17648183010941	0.17855329010351	947	0.15556391558919	0.15740001602869
951	0.71429962000763	0.72382128880320	954	0.62113604447764	0.62873153678521	957	0.63083333624229	0.63806991399311
961	0.43736920633676	0.44318961476565	964	0.32223468971500	0.32565221730781	967	0.34074323802412	0.34417728784688
971	0.13825243617810	0.14011029528825	974	0.04481125801626	0.04502054327026	977	0.05020451441466	0.05028466170064
981	0.41294460457120	0.41726466199836	984	0.52268215554921	0.52904936975901	985	0.51045001444616	0.51719568269362
989	0.10968343957875	0.11038963222158	992	0.21946982814940	0.22217433998223	993	0.20544263961316	0.20817868831867
997	0.47057067245229	0.47613712672652	1000	0.36064963920186	0.36435241896587	1003	0.36661318359047	0.36976404559162
1007	0.16696611260443	0.16926209694974	1010	0.05712203332544	0.05747738918909	1013	0.06060608874387	0.06074705121667
1017	0.12568970617443	0.12651002002600	1020	0.24140360808720	0.24404741748384	1021	0.23547509532103	0.23857938012980
1025	0.17977794383315	0.18184534524342	1028	0.06394425799922	0.06430794778558	1031	0.06523140277810	0.06526309611003
1035	0.98037649712264	0.99572243068691	1038	0.96660809290317	0.98114968333084	1041	0.96669838344885	0.98114968333084
1045	0.91354883852343	0.92682090032251	1048	0.86529084474814	0.87693549776453	1051	0.90093032737360	0.91325655105360
1055	0.75723858795105	0.76719568269362	1058	0.67915247782640	0.68688075570814	1061	0.74708046864716	0.75596750452669
1065	0.52651258083929	0.53247200627604	1068	0.42609958293322	0.42958933950340	1071	0.51993637518262	0.52467909152139
1075	0.24342459883105	0.24562625993355	1078	0.12996806669906	0.13024672562314	1081	0.24138491744629	0.24203143338379
1085	0.92648238808501	0.93940855633098	1088	0.94726712576208	0.96093900491343	1089	0.94712609371525	0.96093900491343
1093	0.80999853004457	0.81970554599445	1096	0.86350708818213	0.87481355623834	1097	0.82777457798311	0.83849250294927
1101	0.61429373417767	0.61976404559162	1104	0.69538425364551	0.70305526149420	1105	0.62730184605860	0.63396851267565
1109	0.35815431506836	0.35915572225626	1112	0.45901751301874	0.46247701928097	1113	0.36497342022337	0.36738726726297

1117	0.91373453791467	0.92682090032251	1120	0.90069929493313	0.91325655105360	1123	0.86545175736366	0.87693549776453
1127	0.85162923933802	0.86268718550614	1130	0.80665024537306	0.81625372946776	1133	0.80692838006340	0.81625372946776
1137	0.70657490515227	0.71410763827738	1140	0.63399102807601	0.63935030566746	1143	0.67004746759671	0.67567135895653
1147	0.49246827769827	0.49562625993355	1150	0.39909040976747	0.39986281933291	1153	0.46799614794364	0.46894956815147

NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION
1157	0.80981406936323	0.81970554599445	1160	0.82805457840924	0.83849250294927	1161	0.86329989709724	0.87481355623834
1165	0.70877876861560	0.71525554840415	1168	0.75540323282716	0.76334148643527	1169	0.75507877531569	0.76334148643527
1173	0.53912345016228	0.54079135425760	1176	0.60956888929016	0.61346928671611	1177	0.57338426443085	0.57714823342704
1181	0.75759268855316	0.76719568269362	1184	0.74673214624403	0.75596750452669	1187	0.67936896453094	0.68688075570814
1191	0.70676086644772	0.71410763827738	1194	0.66966819207946	0.67567135895653	1197	0.63430026914298	0.63935030566746
1201	0.58774016151010	0.59111776274607	1204	0.52814120344415	0.52923579309253	1207	0.52825607560608	0.52923579309253
1211	0.61394321993248	0.61976404559162	1214	0.62769243704886	0.63396851267565	1215	0.69513335802393	0.70305526149420
1219	0.53893847721870	0.54079135425760	1222	0.57383511680181	0.57714823342704	1223	0.60919035193814	0.61346928671611
1227	0.52700158123462	0.53247200627604	1230	0.51949428454624	0.52467909152139	1233	0.42636348052741	0.42958933950340
1237	0.49282158462950	0.49562625993355	1240	0.46780969431196	0.46894956815147	1243	0.39914194398419	0.39986281933291
1247	0.35767155330089	0.35915572225626	1250	0.36547902610722	0.36738726726297	1251	0.45870642906768	0.46247701928097
1255	0.24400693934599	0.24562625993355	1258	0.24114873831592	0.24203143338379	1261	0.13000697466807	0.13024672562314
1265	0.97387704962309	0.98932210182092	1268	0.95254404447292	0.96715571192955	1271	0.93861638615850	0.95300104853184
1275	0.88757007468902	0.90065277997236	1278	0.83232956248008	0.84391705351846	1281	0.82037898979156	0.83156603115380
1285	0.71415987413367	0.72382128880320	1288	0.63035184166918	0.63806991399311	1291	0.62146880074990	0.62873153678521
1295	0.47037639998466	0.47613712672652	1298	0.36605095854763	0.36976404559162	1301	0.36107454785849	0.36435241896587
1305	0.17953250463562	0.18184534524342	1308	0.06479250704588	0.06526309611003	1311	0.06429958697921	0.06430794778558
1315	0.83987465140379	0.85177670957122	1318	0.88686565077639	0.90023092787438	1319	0.90704628816827	0.92086345838978
1323	0.65905733643477	0.66717453160046	1326	0.73517539900436	0.74518526832117	1327	0.75167505089209	0.76226428361846
1331	0.41313274150346	0.41726466199836	1334	0.51102510547620	0.51719568269362	1335	0.52228479981672	0.52904936975901
1339	0.12585719833827	0.12651002002600	1342	0.23606838331681	0.23857938012980	1343	0.24093456616925	0.24404741748384
1347	0.82547229924464	0.83832983438577	1350	0.77401434917184	0.78552007992821	1353	0.73297137774705	0.74324008236778
1357	0.66416699597564	0.67373464520468	1360	0.58609706714044	0.59391705351846	1363	0.55528532780608	0.56194993744403
1367	0.43726787048210	0.44318961476565	1370	0.34015186271974	0.34417728784688	1373	0.32258346770288	0.32565221730781
1377	0.16681073635697	0.16926209694974	1380	0.06023305056468	0.06074705121667	1383	0.05745760463761	0.05747738918909
1387	0.57496301365784	0.58216061586144	1390	0.64142893172425	0.65023092787438	1391	0.68378307471938	0.69394532362209
1395	0.36044724147169	0.36409521212307	1398	0.44591309585203	0.45129264217493	1399	0.47503999624731	0.48163260958091
1403	0.10983555686392	0.11038963222158	1406	0.20602122635880	0.20817868831867	1407	0.21907276212998	0.22217433998223
1411	0.54970715219894	0.55769816034822	1414	0.48507463933154	0.49162745378198	1417	0.43519732448210	0.44016076289038
1421	0.36203009763207	0.36685961542790	1424	0.28162484644632	0.28490005914351	1427	0.25295333133180	0.25507490766725
1431	0.13817518341735	0.14011029528825	1434	0.04989702217619	0.05028466170064	1437	0.04508521089877	0.04502054327026
1441	0.27283735622861	0.27528558608466	1444	0.33752684299921	0.34121393349943	1445	0.38184428031474	0.38707029384531
1449	0.08320609413157	0.08346353808552	1452	0.15606349007450	0.15740001602869	1453	0.17618570301572	0.17855329010351
1457	0.25154763974305	0.25461884087082	1460	0.19569786515867	0.19773482763574	1463	0.15859344549972	0.15952908885283
1467	0.09610774271335	0.09724352171812	1470	0.03471127242090	0.03490005914351	1473	0.02830318941479	0.02815677289956
1477	0.04840577030850	0.04836745131698	1480	0.09078823236948	0.09121393349943	1481	0.11611595748800	0.11745420013553
1485	0.04449479989720	0.04485787470676	1488	0.01607329340308	0.01609919563440	1491	0.00864187416378	0.00853682141769
1495	0.97338703109168	0.98932210182092	1498	0.93811558614037	0.95300104853184	1501	0.95145735508867	0.96715571192955
1505	0.82503439644358	0.83832983438577	1508	0.73256319760435	0.74324008236778	1511	0.77357206083738	0.78552007992821
1515	0.54940986989902	0.55769816034822	1518	0.43495028334971	0.44016076289038	1521	0.48524109405880	0.49162745378198
1525	0.25139216084576	0.25461884087082	1528	0.15847884606033	0.15952908885283	1531	0.19607431893860	0.19773482763574
1535	0.04442901344292	0.04485787470676	1538	0.00858337292818	0.00853682141769	1541	0.01617997675236	0.01609919563440
1545	0.83839350179801	0.85177670957122	1548	0.90556238127299	0.92086345838978	1549	0.88569785421839	0.90023092787438
1553	0.57418112231669	0.58216061586144	1556	0.68294964208784	0.69394532362209	1557	0.64037595617481	0.65023092787438
1561	0.27258897482204	0.27528558608466	1564	0.38166390665600	0.38707029384531	1565	0.33678588189982	0.34121393349943
1569	0.04833197811391	0.04836745131698	1572	0.11622792130129	0.11745420013553	1573	0.09040646181470	0.09121393349943
1577	0.88508665695370	0.90065277997236	1580	0.81793744398727	0.83156603115381	1583	0.82951112305299	0.84391705351846
1587	0.66258193226048	0.67373464520468	1590	0.55385514579817	0.56194993744403	1593	0.58484211056500	0.59391705351846
1597	0.36143002112248	0.36685961542790	1600	0.25238287529952	0.25507490766725	1603	0.28161967643738	0.28490005914351
1607	0.09602614622625	0.09724352171812	1610	0.02810549542909	0.02815677289956	1613	0.03489456309838	0.03490005914351
1617	0.65635267079838	0.66717453160046	1620	0.74877988510885	0.76226428361846	1621	0.73241397757643	0.74518526832117
1625	0.35935601607851	0.36409521212307	1628	0.47382854613248	0.48163260958091	1629	0.44429606895827	0.45129264217493
1633	0.08297606177439	0.08346353808552	1636	0.17614988063279	0.17855329010351	1637	0.15536636026601	0.15740001602869
1641	0.71028585686771	0.72382128880320	1644	0.61791918191712	0.62873153678521	1647	0.62654006323795	0.63806991399311
1651	0.43551824810234	0.44318961476565	1654	0.32109504085301	0.32565221730781	1657	0.33909574283249	0.34417728784688
1661	0.13792018243823	0.14011029528825	1664	0.04473188839974	0.04502054327026	1667	0.05007982740923	0.05028466170064
1671	0.41034688995027	0.41726466199836	1674	0.51894475732878	0.52904936975901	1675	0.50769484189608	0.51719568269362
1679	0.10929775494143	0.11038963222158	1682	0.21846899638585	0.22217433998223	1683	0.20475936076866	0.20817868831867

1687	0.46621915537101	0.47613712672652	1690	0.35774820828696	0.36435241896587	1693	0.36259309376253	0.36976404559162
1697	0.16601975365279	0.16926209694974	1700	0.05687791216569	0.05747738918909	1703	0.06025114905852	0.06074705121667
1707	0.12463704105198	0.12651002002600	1710	0.23879668617424	0.24404741748384	1711	0.23365010213536	0.23857938012980
1715	0.17716617541089	0.18184534524342	1718	0.06326251800604	0.06430794778558	1721	0.06412645376133	0.06526309611003
1725	0.97300345442538	0.98932210182092	1728	0.95096654046697	0.96715571192955	1731	0.93728794848522	0.95300104853184
1735	0.88477612818715	0.90065277997236	1738	0.82901027535080	0.84391705351846	1741	0.81763527394378	0.83156603115380
1745	0.70999243708722	0.72382128880320	1748	0.62600924200233	0.63806991399311	1751	0.61798226056600	0.62873153678521
1755	0.46592103120957	0.47613712672652	1758	0.36204332409295	0.36976404559162	1761	0.35805607294350	0.36435241896587
1765	0.17688758932963	0.18184534524342	1768	0.06377900868302	0.06526309611003	1771	0.06361388647389	0.06430794778558
1775	0.83764800205995	0.85177670957122	1778	0.88509390424311	0.90023092787438	1779	0.90487791951048	0.92086345838978

NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION
1783	0.65604570584651	0.66717453160046	1786	0.73236438048395	0.74518526832117	1787	0.74817892526272	0.76226428361846
1791	0.41033507521464	0.41726466199836	1794	0.50799324115202	0.51719568269362	1795	0.51837403170486	0.52904936975901
1799	0.12475966191614	0.12651002002600	1802	0.23414117983754	0.23857938012980	1803	0.23825534098463	0.24404741748384
1807	0.82385868143697	0.83832983438577	1810	0.77208159014525	0.78552007992821	1813	0.73142221985174	0.74324008236778
1817	0.66188261881603	0.67373464520468	1820	0.58374080417887	0.59391705351846	1823	0.55343513560535	0.56194993744403
1827	0.43507932283752	0.44318961476565	1830	0.33826282314507	0.34417728784688	1833	0.32114641982727	0.32565221730781
1837	0.16575701544912	0.16926209694974	1840	0.05984095269659	0.06074705121667	1843	0.05716687493918	0.05747738918909
1847	0.57335339889101	0.58216061586144	1850	0.63989360151120	0.65023092787438	1851	0.68184685144877	0.69394532362209
1855	0.35907512499342	0.36409521212307	1858	0.44437212720648	0.45129264217493	1859	0.47305230728508	0.48163260958091
1863	0.10934765200378	0.11038963222158	1866	0.20513772645295	0.20817868831867	1867	0.21790169162563	0.22217433998223
1871	0.54849503835245	0.55769816034822	1874	0.48384975970416	0.49162745378198	1877	0.43424966312297	0.44016076289038
1881	0.36093823623152	0.36685961542790	1884	0.28070716665863	0.28490005914351	1887	0.25225355151169	0.25507490766725
1891	0.13768315822163	0.14011029528825	1894	0.04971636807213	0.05028466170064	1897	0.04494850792711	0.04502054327026
1901	0.27216572996089	0.27528558608466	1904	0.33676011342850	0.34121393349943	1905	0.38084231585983	0.38707029384531
1909	0.08297524683275	0.08346353808552	1912	0.15564145419216	0.15740001602869	1913	0.17562806892062	0.17855329010351
1917	0.25102354877776	0.25461884087082	1920	0.19526232258410	0.19773482763574	1923	0.15827621105049	0.15952908885283
1927	0.09587757664251	0.09724352171812	1930	0.03462712597554	0.03490005914351	1933	0.02824204065362	0.02815677289956
1937	0.04830653243007	0.04836745131698	1940	0.09060613723327	0.09121393349943	1941	0.11585914521303	0.11745420013553
1945	0.04441016252778	0.04485787470676	1948	0.01604240625689	0.01609919563440	1951	0.00862613760696	0.00853682141769
1954	0.97298892461233	0.98932210182092	1957	0.93722665756930	0.95300104853184	1959	0.95113531320749	0.96715571192955
1963	0.82378200127156	0.83832983438577	1966	0.73131146114767	0.74324008236778	1969	0.77256572313822	0.78552007992821
1973	0.54838337477857	0.55769816034822	1976	0.43412824536170	0.44016076289038	1979	0.48444624370510	0.49162745378198
1983	0.25091965740453	0.25461884087082	1986	0.15818619928977	0.15952908885283	1989	0.19573780461830	0.19773482763574
1993	0.04434788553983	0.04485787470676	1996	0.00856794174998	0.00853682141769	1999	0.01615160081058	0.01609919563440
2003	0.83761979383501	0.85177670957122	2006	0.90505029714716	0.92086345838978	2007	0.88474253711801	0.90023092787438
2011	0.57335612136279	0.58216061586144	2014	0.68205965275040	0.69394532362209	2015	0.63932244415864	0.65023092787438
2019	0.27214941899623	0.27528558608466	2022	0.38106386061436	0.38707029384531	2023	0.33618623078726	0.34121393349943
2027	0.04824991950654	0.04836745131698	2030	0.11603322508603	0.11745420013553	2031	0.09024542645912	0.09121393349943
2034	0.88485712084928	0.90065277997236	2037	0.81742552980635	0.83156603115380	2039	0.82932588057632	0.84391705351846
2043	0.66189692887712	0.67373464520468	2046	0.55319527947461	0.56194993744403	2049	0.58430732239187	0.59391705351846
2053	0.36093328730045	0.36685961542790	2056	0.25201759511959	0.25507490766725	2059	0.28126398501877	0.28490005914351
2063	0.09587549772273	0.09724352171812	2066	0.02805980602024	0.02815677289956	2069	0.03484183017893	0.03490005914351
2073	0.65593161764960	0.66717453160046	2076	0.74849243448231	0.76226428361846	2077	0.73188001199110	0.74518526832117
2081	0.35896528909856	0.36409521212307	2084	0.47337762628923	0.48163260958091	2085	0.44376626140775	0.45129264217493
2089	0.08286136351539	0.08346353808552	2092	0.17592390692018	0.17855329010351	2093	0.15514171323342	0.15740001602869
2096	0.71015870262304	0.72382128880320	2099	0.61764482445799	0.62873153678521	2101	0.62644054782220	0.63806991399311
2105	0.43518058901209	0.44318961476565	2108	0.32079685964177	0.32565221730781	2111	0.33885296414335	0.34417728784688
2115	0.13776005621482	0.14011029528825	2118	0.04467447479599	0.04502054327026	2121	0.05002370809796	0.05028466170064
2125	0.41014489936254	0.41726466199836	2128	0.51879743656815	0.52904936975901	2129	0.50742197740106	0.51719568269362
2133	0.10919491670242	0.11038963222158	2136	0.21829725441003	0.22217433998223	2137	0.20455834614804	0.20817868831867
2140	0.46615596674986	0.47613712672652	2143	0.35762305178595	0.36435241896587	2145	0.36254763063804	0.36976404559162
2149	0.16590997709098	0.16926209694974	2152	0.05683075666280	0.05747738918909	2155	0.06021263983495	0.06074705121667
2159	0.12458357827927	0.12651002002600	2162	0.23874024422357	0.24404741748384	2163	0.23354572958019	0.23857938012980
2166	0.17714554848392	0.18184534524342	2169	0.06324265731456	0.06430794778558	2171	0.06411922505963	0.06526309611003

Example 3(pentagonal domain,refer section 7.5.2)

Table-5b

Mesh:number of nodes=8541,number of nine node elements= 2100

fem computed values and exact values at centroid points

```
        NODE FEM SOLUTION EXACT SOLUTION       NODE FEM SOLUTION EXACT SOLUTION       NODE FEM
SOLUTION EXACT SOLUTION
--------------------------------------------------------------------------------------------------------------------------------------------------

 2176   0.99334058161356   0.99732465707428  2180   0.98773832500749   0.99174663408490  2183   0.98420116458847
0.98811523439862
 2188   0.97077941391612   0.97482011001528  2192   0.95613342885394   0.96014371300095  2195   0.95273139590986
0.95662803121455
 2200   0.92432005015106   0.92831225660525  2204   0.90099207633806   0.90489886716863  2207   0.89781199393043
0.90158547103557
 2212   0.85510896710957   0.85894627454584  2216   0.82367417800256   0.82737240803946  2219   0.82079656240239
0.82434288436907
 2224   0.76485257811161   0.76843018452787  2228   0.72608608179689   0.72947329433276  2231   0.72358340024676
0.72680224005222
 2236   0.65577673729797   0.65899279309326  2240   0.61063497951629   0.61361212713653  2243   0.60856982639828
0.61136531246694
 2248   0.53057339974713   0.53332881202100  2252   0.48017139077361   0.48264179290830  2255   0.47859474697517
0.48087454188358
 2260   0.39233698400499   0.39453250557993  2264   0.33792254275057   0.33978721594239  2267   0.33687180965164
0.33854304414794
 2272   0.24449460611102   0.24602149947317  2276   0.18742272234921   0.18856595003087  2279   0.18691731212896
0.18787549310543
 2284   0.09074670863061   0.09145262755084  2288   0.03248568338437   0.03270156461507  2291   0.03246524833412
0.03258182389965
 2295   0.95788543152270   0.96174642343239  2298   0.97057022124839   0.97445771976688  2299   0.97593941693685
0.97993850223311
 2303   0.90793986777333   0.91169622366786  2306   0.92933828940258   0.93317442234844  2307   0.93445530736108
0.93842300923754
 2311   0.83564988130072   0.83919703713035  2314   0.86523472317840   0.86891327362344  2315   0.86997093949328
0.87380042730712
 2319   0.74279680440135   0.74603403440339  2322   0.77983898556723   0.78325659634400  2323   0.78407589857134
0.78766197887905
 2327   0.63166970875088   0.63450119780442  2330   0.67525587222792   0.67831354218297  2331   0.67888797261794
0.68212867843074
 2335   0.50501035100814   0.50734483592675  2338   0.55406478752745   0.55666815741981  2339   0.55700230971734
0.55979910606418
 2343   0.36594795267937   0.36769596040760  2346   0.41925805364900   0.42131575514528  2347   0.42142968808664
0.42368542183959
 2351   0.21792658245035   0.21899319003028  2354   0.27417068776018   0.27558916071225  2355   0.27552714403215
0.27713919639802
 2359   0.06461838220790   0.06489808051768  2362   0.12240343354141   0.12307664651450  2363   0.12292932621447
0.12376888416887
 2367   0.95401974310133   0.95782627229675  2370   0.93962381270555   0.94340572588151  2373   0.92748301893207
0.93109948672708
 2377   0.90838965218024   0.91212918069330  2380   0.88546294146503   0.88912395204085  2383   0.87404492771478
0.87752579051652
 2387   0.84040656308697   0.84397244147798  2390   0.80951360592261   0.81294899566772  2393   0.79910147970160
0.80234449700232
 2397   0.75174470101210   0.75503429976952  2400   0.71364654047794   0.71675653699822  2403   0.70449771414630
0.70740681914320
 2407   0.64458863916650   0.64750470778557  2410   0.60022439157505   0.60291515360926  2413   0.59256404168798
0.59505043764815
 2417   0.52158043895676   0.52403140095101  2420   0.47204463181774   0.47422799817777  2423   0.46605960022986
0.46804193951901
```

```
NODE     FEM SOLUTION    EXACT SOLUTION      NODE   FEM SOLUTION      EXACT SOLUTION  NODE    FEM SOLUTION      EXACT SOLUTION
----------------------------------------------------------------------------------------------------------------------------------

2427   0.38575599428770   0.38765470186453  2430   0.33227227858115   0.33386377555864  2433   0.32810503524999
0.32950869549679
 2437   0.24047202829078   0.24173265746595  2440   0.18436332120879   0.18527871875492  2443   0.18210469257135
0.18286185381476
```

2447 0.08932280900539 0.08985835277587 2450 0.03198587855304 0.03213148499064 2453 0.03164165015162 0.03171234640812
2457 0.87918855801572 0.88264418273435 2460 0.89990679964170 0.90343795880678 2461 0.91353895404213 0.91724757291113
2465 0.80921786483889 0.81245524963461 2468 0.83786090806754 0.84122455084753 2469 0.85052885511205 0.85408319411027
2473 0.71933835375618 0.72226097190448 2476 0.75520120724341 0.75829740258220 2477 0.76659150890425 0.76988845253080
2481 0.61176315034994 0.61428223199932 2484 0.65396259866028 0.65669845562050 2485 0.66379431414585 0.66673650213674
2489 0.48914268572837 0.49117782485650 2492 0.53663882734720 0.53892941322420 2493 0.54467076373270 0.54716728628849
2497 0.35449984784243 0.35597898953996 2500 0.40612159747114 0.40789014007015 2501 0.41215906883045 0.41412499590771
2505 0.21115453907683 0.21201477007442 2508 0.26562921307264 0.26680725795781 2509 0.26953079110845 0.27088557372561
2513 0.06262792778365 0.06283004333297 2516 0.11862244968087 0.11915469567202 2517 0.12030836878469 0.12097604970072
2521 0.87006895247377 0.87348645716948 2524 0.84811166353999 0.85145585438060 2527 0.82873272748207 0.83185851269382
2531 0.80497943343654 0.80821720591697 2534 0.77539422141053 0.77850808099961 2537 0.75769970979813 0.76058972529065
2541 0.72008687583216 0.72304696473567 2544 0.68360155498880 0.68639085494422 2547 0.66802779537566 0.67059269459825
2551 0.61748010820591 0.62007290762727 2554 0.57499276181545 0.57737240803946 2557 0.56192254773555 0.56408344619528
2561 0.49968518162233 0.50183059762910 2564 0.45224209682686 0.45413713625960 2567 0.44199478816842 0.44368459126136
2571 0.36960312214542 0.37123155283703 2574 0.31837276561172 0.31971950099029 2577 0.31119557285935 0.31236074918584
2581 0.23043711869592 0.23149155516722 2584 0.17667999931820 0.17742931051841 2587 0.17274090933625 0.17334554879948
2591 0.08560989058489 0.08605146713278 2594 0.03065734543324 0.03077022156744 2597 0.03001758925830 0.03006200570078
2601 0.76283068479011 0.76570811750848 2604 0.78982889614418 0.79282208776554 2605 0.81011640922291 0.81333559813481
2609 0.67813124059071 0.68070344723054 2612 0.71193392335769 0.71466641012403 2613 0.73019616718998 0.73315771736796
2617 0.57674987852778 0.57893759895656 2620 0.61652587053722 0.61891327362344 2621 0.63231310822646 0.63492706038855
2625 0.46118010231389 0.46291638561253 2628 0.50595102063749 0.50792043827084 2629 0.51887592044648 0.52106239198028
2633 0.33426457461125 0.33549663452332 2636 0.38292926456467 0.38442091603673 2637 0.39267698414699 0.39436743817451
2641 0.19912302855139 0.19981584284266 2644 0.25048571552726 0.25145567502992 2645 0.25682243274306 0.25796184921040
2649 0.05906516650497 0.05921492196044 2652 0.11187200737583 0.11229876077033 2653 0.11465200223041 0.11520438339244
2657 0.74970155463807 0.75256098042718 2660 0.72215034674627 0.72489771365710 2663 0.69760845630351 0.70010671028975
2667 0.67066275281687 0.67325581377473 2670 0.63668653038324 0.63912395204085 2673 0.61507143496223 0.61726635234273
2677 0.57512611708225 0.57737285457920 2680 0.53555978333525 0.53761283759456 2683 0.51740289519321 0.51922684821146
2687 0.46543953547128 0.46727305951329 2690 0.42125557471553 0.42286391085199 2693 0.40700167085405 0.40840225586211
2697 0.34429880338124 0.34566745093179 2700 0.29658283786209 0.29770267121937 2703 0.28657881738633 0.28752144456405
2707 0.21467953119569 0.21555036250386 2710 0.16460260612670 0.16521100380282 2713 0.15908892381969 0.15956090107186
2717 0.07976254366742 0.08012570878043 2720 0.02856375517595 0.02865129316875 2723 0.02764698935499 0.02767143864301
2727 0.62018983166664 0.62238474456056 2730 0.65110168180828 0.65343795880678 2731 0.67578660377608 0.67837420599256

2735	0.52749473385578	0.52933760084375	2738	0.56386921635523	0.56588839277437	2739	0.58522270627061	0.58748360721149
2743	0.42181874428685	0.42325640862338	2746	0.46276278267121	0.46440477643596	2747	0.48026041727337	0.48212721227457
2751	0.30575611989114	0.30675323891523	2754	0.35026469532000	0.35148597323061	2755	0.36347849353214	0.36489924528297
2759	0.18215409707036	0.18269678640934	2762	0.22913570186943	0.22991241884926	2763	0.23774482159934	0.23868624784134
2767	0.05403547322678	0.05414173268619	2770	0.10234442829512	0.10267765768026	2771	0.10614381370189	0.10659600282361
2775	0.60468857615167	0.60688687027071	2778	0.57405680314604	0.57611969630745	2781	0.54693242344979	0.54874086390296
2785	0.51857151213369	0.52045596566075	2788	0.48289968141067	0.48461545485338	2791	0.46010399864735	0.46158516200956
2795	0.41969273308363	0.42120970788190	2798	0.37985601061571	0.38117837255433	2801	0.36194853390279	0.36306370151415
2805	0.31047794371435	0.31159186918012	2808	0.26745331358831	0.26835541366451	2811	0.25487207006623	0.25560240774812
2815	0.19360490860759	0.19430160454500	2818	0.14844653827143	0.14892465386972	2821	0.14149736380452	0.14184733440758
2825	0.07193752579728	0.07222698955594	2828	0.02576181566167	0.02582687484406	2831	0.02459138844595	0.02459950892961
2835	0.46521221664358	0.46670355422739	2838	0.49728985412145	0.49892946161930	2839	0.52368373222076	0.52557435787858
2843	0.37203286579995	0.37317445414644	2846	0.40814123340315	0.40945392773414	2847	0.42977968213217	0.43132046051419
2851	0.26968534110175	0.27045656050940	2854	0.30893968109371	0.30989627924849	2855	0.32529198457781	0.32644602194140
2859	0.16067616466993	0.16107912875876	2862	0.20211540762585	0.20270795588095	2863	0.21278200786731	0.21353339889618
2867	0.04766745631641	0.04773539426710	2870	0.09028239936285	0.09052828989049	2871	0.09500554364726	0.09536287489342
2875	0.44920933795424	0.45072372357340	2878	0.41831054040702	0.41968523126722	2881	0.39143728548365	0.39257771720566
2885	0.36357434432303	0.36477477532757	2888	0.32906701689750	0.33010695766590	2891	0.30794749804801	0.30878531389558
2895	0.26897926835903	0.26984386149511	2898	0.23170821332601	0.23240035520468	2901	0.21686064597223	0.21738959135769
2905	0.16773940678389	0.16826849623220	2908	0.12861628836541	0.12897128470575	2911	0.12040379624563	0.12064101560588
2915	0.06233152885986	0.06254980214093	2918	0.02232204246482	0.02236651314621	2921	0.02092700317564	0.02092185766528
2925	0.31304993092995	0.31390370611341	2928	0.34343217214392	0.34442096443183	2929	0.36867865957778	0.36989316754540
2933	0.22694370087301	0.22750034398998	2936	0.25997450355529	0.26067591038448	2937	0.27906214584497	0.27995461412740
2941	0.13522170497335	0.13549516836715	2944	0.17009391095331	0.17051215028973	2945	0.18255452397328	0.18312264899348
2949	0.04011937257473	0.04015365201645	2952	0.07598548384765	0.07614981515745	2953	0.08151552089047	0.08178159649211
2957	0.29846608743287	0.29935787718661	2960	0.27013961756392	0.27090721391756	2963	0.24632732879816	0.24690360704919
2967	0.22082583206706	0.22145140237965	2970	0.19022926857720	0.19072282870714	2973	0.17348048851471	0.17382392175333
2977	0.13772172248926	0.13809205909104	2980	0.10560181447181	0.10584221448118	2983	0.09632798456139	0.09646411461538
2987	0.05118207573550	0.05133243100633	2990	0.01832950738564	0.01835541366451	2993	0.01674416599000	0.01672904082956
2997	0.17858261469602	0.17894231397116	3000	0.20457352936949	0.20503683547307	3001	0.22592745393499	0.22656979459748
3005	0.10641709694516	0.10657486724767	3008	0.13385909099648	0.13411776966098	3009	0.14780756966396	0.14820281172333
3013	0.03157700924775	0.03158319359083	3016	0.05980556078418	0.05989627924849	3017	0.06600664731696	0.06618658376761

3021 0.16720052603148 0.16760607478254 3024 0.14403527715194 0.14434907319406 3027 0.12579564903185 0.12597813030688
3031 0.10428897756775 0.10451533715378 3034 0.07996819354586 0.08010695766590 3037 0.06985989729566 0.06991194697701
3041 0.03876287567213 0.03885108505778 3044 0.01388221892430 0.01389234298027 3047 0.01214531439682 0.01212429948811
3051 0.07496757346626 0.07503033919484 3054 0.09429855427499 0.09442096443183 3055 0.10939369540792 0.10963372937162
3059 0.02224929565374 0.02223505212040 3062 0.04213847331718 0.04216789816007 3063 0.04885920849328 0.04896183769007
3067 0.06825813052354 0.06836510074929 3070 0.05234125770196 0.05239920169315 3073 0.04164392477659 0.04163831517961
3077 0.02537677407466 0.02541309645575 3080 0.00908858451002 0.00908719670580 3083 0.00724210075300 0.00722101765502
3087 0.01236212965521 0.01233940987285 3090 0.02341267755268 0.02340120347172 3091 0.03049075027066 0.03053148867358
3095 0.01134563678258 0.01134935307775 3098 0.00406359805619 0.00405829348976 3101 0.00215028090079 0.00213993040208
3105 0.99340016148415 0.99732465707428 3108 0.98425607534638 0.98811523439862 3111 0.98791498713410 0.99174663408490
3115 0.95407068542295 0.95782627229675 3118 0.92752856394070 0.93109948672708 3121 0.93981241829438 0.94340572588151
3125 0.87010826793930 0.87348645716948 3128 0.82876593245569 0.83185851269382 3131 0.84829626887066 0.85145585438060
3135 0.74972742543434 0.75256098042718 3138 0.69762829372471 0.70010671028975 3141 0.72231880695013 0.72489771365710
3145 0.60470123505738 0.60688687027071 3148 0.54693997671328 0.54874086390296 3151 0.57420073257832 0.57611969630745
3155 0.44921111921489 0.45072372357340 3158 0.39143556799624 0.39257771720566 3161 0.41842614667333 0.41968523126722
3165 0.29846083361278 0.29935787718661 3168 0.24632046199327 0.24690360704919 3171 0.27022680750596 0.27090721391756
3175 0.16719251236662 0.16760607478254 3178 0.12578773852043 0.12597813030688 3181 0.14409563826861 0.14434907319406
3185 0.06825101943238 0.06836510074929 3188 0.04163811896346 0.04163831517961 3191 0.05237627145581 0.05239920169315
3195 0.01134156902717 0.01134935307775 3198 0.00214659035564 0.00213993040208 3201 0.00407056736573 0.00405829348976
3205 0.95810100151452 0.96174642343239 3208 0.97617376392937 0.97993850223311 3209 0.97066280402546 0.97445771976688
3213 0.87938089257817 0.88264418273435 3216 0.91375498144432 0.91724757291113 3217 0.89996535870019 0.90343795880678
3221 0.76298750577900 0.76570811750848 3224 0.81030107795426 0.81333559813481 3225 0.78985145989764 0.79282208776554
3229 0.62030510020036 0.62238474456056 3232 0.67593140346321 0.67837420599256 3233 0.65109100947003 0.65343795880678
3237 0.46528712087175 0.46670355422739 3240 0.52378718272395 0.52557435787858 3241 0.49725361211608 0.49892946161930
3245 0.31309160222364 0.31390370611341 3248 0.36874597865702 0.36989316754540 3249 0.34338175031761 0.34442096443183
3253 0.17860081014248 0.17894231397116 3256 0.22596783574804 0.22656979459748 3257 0.20452165598241 0.20503683547307
3261 0.07497121502628 0.07503033919484 3264 0.10941665938459 0.10963372937162 3265 0.09425666637298 0.09442096443183
3269 0.01235883115460 0.01233940987285 3272 0.03050307655420 0.03053148867358 3273 0.02338912104186 0.02340120347172
3277 0.97112711430432 0.97482011001528 3280 0.95304925408245 0.95662803121455 3283 0.95659697076022 0.96014371300095
3287 0.90869748811317 0.91212918069330 3290 0.87431727277896 0.87752579051652 3293 0.88590036830300 0.88912395204085
3297 0.80522999121515 0.80821720591697 3300 0.75790989984245 0.76058972529065 3303 0.77577364293088 0.77850808099961
3307 0.67084574926376 0.67325581377473 3310 0.61521338463054 0.61726635234273 3313 0.63698952225524 0.63912395204085

```
 3317  0.51868812460457  0.52045596566075  3320  0.46018321738304  0.46158516200956  3323  0.48312190200184
0.48461545485338
 3327  0.36363639374190  0.36477477532757  3330  0.30797876102638  0.30878531389558  3333  0.32921702097809
0.33010695766590
 3337  0.22085118104918  0.22145140237965  3340  0.17348229890662  0.17382392175333  3343  0.19032252924468
0.19072282870714
 3347  0.10429549588653  0.10451533715378  3350  0.06984877762312  0.06991194697701  3353  0.08002003506255
0.08010695766590
 3357  0.02537705414739  0.02541309645575  3360  0.00723042464068  0.00722101765502  3363  0.00910313072002
0.00908719670580
 3367  0.90840061828366  0.91169622366786  3370  0.93496449926962  0.93842300923754  3371  0.92968106601069
0.93317442234844
 3375  0.80960487939862  0.81245524963461  3378  0.85097429598320  0.85408319411027  3379  0.83812784044364
0.84122455084753
 3383  0.67842421370036  0.68070344723054  3386  0.73055361462649  0.73315771736796  3387  0.71211327995951
0.71466641012403
 3391  0.52768991103310  0.52933760084375  3394  0.58548197655326  0.58748360721149  3395  0.56396310729927
0.56588839277437
 3399  0.37214291165368  0.37317445414644  3402  0.42994770407557  0.43132046051419  3403  0.40816569616198
0.40945392773414
 3407  0.22699158286591  0.22750034398998  3410  0.27915903506577  0.27995461412740  3411  0.25995476521958
0.26067591038448

______________________________________________________________________________________________
NODE      FEM SOLUTION    EXACT SOLUTION      NODE    FEM SOLUTION       EXACT SOLUTION   NODE     FEM SOLUTION       EXACT SOLUTION
-------------------------------------------------------------------------------------------------------------------------------------

3415  0.10642711625423  0.10657486724767  3418  0.14785807879662  0.14820281172333  3419  0.13382194548187
0.13411776966098
 3423  0.02224356222977  0.02223505212040  3426  0.04888428461718  0.04896183769007  3427  0.04210605742403
0.04216789816007
 3431  0.92493706025101  0.92831225660525  3434  0.89836721604963  0.90158547103557  3437  0.90171938922176
0.90489886716863
 3441  0.84093346835521  0.84397244147798  3444  0.79955717070555  0.80234449700232  3447  0.81015692949172
0.81294899566772
 3451  0.72049708761776  0.72304696473567  3454  0.66836075016158  0.67059269459825  3457  0.68411970610766
0.68639085494422
 3461  0.57540968901433  0.57737285457920  3464  0.51761153523077  0.51922684821146  3467  0.53593775470885
0.53761283759456
 3471  0.41986174055855  0.42120970788190  3474  0.36205229443550  0.36306370151415  3477  0.38010399032747
0.38117837255433
 3481  0.26906255025343  0.26984386149511  3484  0.21689182430046  0.21738959135769  3487  0.23185427926689
0.23240035520468
 3491  0.13775354664231  0.13809205909104  3494  0.09631989904021  0.09646411461538  3497  0.10567842873424
0.10584221448118
 3501  0.03877128521354  0.03885108505778  3504  0.01212756341706  0.01212429948811  3507  0.01390543251703
0.01389234298027
 3511  0.83631912674934  0.83919703713035  3514  0.87072501290964  0.87380042730712  3515  0.86579605602245
0.86891327362344
 3519  0.71986784528011  0.72226097190448  3522  0.76721843097846  0.76988845253080  3523  0.75562909291145
0.75829740258220
 3527  0.57712018815576  0.57893759895656  3530  0.63278508725824  0.63492706038855  3531  0.61680678328490
0.61891327362344
 3535  0.42203921765886  0.42325640862338  3538  0.48057603320274  0.48212721227457  3539  0.46290819272137
0.46440477643596
 3543  0.26978887999812  0.27045656050940  3546  0.32547666160817  0.32644602194140  3547  0.30898277178097
0.30989627924849
 3551  0.13525021138881  0.13549516836715  3554  0.18264868208835  0.18312264899348  3555  0.17007772762420
0.17051215028973
 3559  0.03157179970437  0.03158319359083  3562  0.06604961054356  0.06618658376761  3563  0.05977021264311
0.05989627924849
 3567  0.85596463368754  0.85894627454584  3570  0.82155171694284  0.82434288436907  3573  0.82462913781667
0.82737240803946
 3577  0.75244161997646  0.75503429976952  3580  0.70508331526222  0.70740681914320  3583  0.71444094283774
0.71675653699822
```

```
 3587  0.61799025205142  0.62007290762727 3590  0.56231800845817  0.56408344619528 3593  0.57558511638601
0.57737240803946
 3597  0.46576428723944  0.46727305951329 3600  0.40722143272580  0.40840225586211 3603  0.42164659190629
0.42286391085199
 3607  0.31065110524867  0.31159186918012 3610  0.25495885665912  0.25560240774813 3613  0.26767900278522
0.26835541366451
 3617  0.16781239409964  0.16826849623220 3620  0.12041037340620  0.12064101560588 3623  0.12872876185629
0.12897128470575
 3627  0.05120298675869  0.05133243100633 3630  0.01672226540450  0.01672904082956 3633  0.01836246796274
0.01835541366451
 3637  0.74362521744667  0.74603403440339 3640  0.78503155193037  0.78766197887905 3641  0.78057537497946
0.78325659634400
 3645  0.61237275478330  0.61428223199932 3648  0.66454305905736  0.66673650213674 3649  0.65449439992949
0.65669845562050
 3653  0.46156410894456  0.46291638561253 3656  0.51939667169441  0.52106239198028 3657  0.50627324980206
0.50792043827084
 3661  0.30594978465170  0.30675323891523 3664  0.36379171335988  0.36489924528297 3665  0.35040998919120
0.35148597323061
 3669  0.16073974811424  0.16107912875876 3672  0.21294157032176  0.21353339889618 3673  0.20214186258569
0.20270795588095
 3677  0.04011858697351  0.04015365201645 3680  0.08158337163093  0.08178159649211 3681  0.07595516864977
0.07614981515745
 3685  0.76590203821447  0.76843018452787 3688  0.72448739347680  0.72680224005222 3691  0.72721731925663
0.72947329433276
 3695  0.64539353105753  0.64750470778557 3698  0.59321459332757  0.59505043764815 3701  0.60110114306808
0.60291515360926
 3705  0.50022669004910  0.50183059762910 3708  0.44238650004671  0.44368459126136 3711  0.45283519239326
0.45413713625960
 3715  0.34460410303287  0.34566745093179 3718  0.28675642445394  0.28752144456405 3721  0.29692512523239
0.29770267121937
 3725  0.19374053476560  0.19430160454500 3728  0.14153449276814  0.14184733440758 3731  0.14861041606705
0.14892465386972
 3735  0.06237084227248  0.06254980214093 3738  0.02090352191231  0.02092185766528 3741  0.02236624488034
0.02236651314621
 3745  0.63259290478878  0.63450119780442 3748  0.67998615784739  0.68212867843074 3749  0.67610989177818
0.67831354218297
 3753  0.48975845050780  0.49117782485650 3756  0.54546795100316  0.54716728628849 3757  0.53720647378764
0.53892941322420
 3761  0.33459546513951  0.33549663452332 3764  0.39317356075698  0.39436743817451 3765  0.38322840128281
0.38442091603673
 3769  0.18227573847985  0.18269678640934 3772  0.23799999146220  0.23868624784134 3773  0.22923394919102
0.22991241884926
 3777  0.04767643265866  0.04773539426710 3780  0.09510801947749  0.09536287489342 3781  0.09026785427850
0.09052828989049
 3785  0.65695806409589  0.65899279309326 3788  0.60955498504248  0.61136531246694 3791  0.61187256691260
0.61361212713653
 3795  0.52241573223481  0.52403140095101 3798  0.46669669601863  0.46804193951901 3801  0.47291826862975
0.47422799817777
 3805  0.37009877788852  0.37123155283703 3808  0.31151318311644  0.31236074918584 3811  0.31888529022730
0.31971950099029
 3815  0.21490807690298  0.21555036250386 3818  0.15917881134575  0.15956090107186 3821  0.16484074079721
0.16521100380282
 3825  0.07200342421116  0.07222698955594 3828  0.02456977956320  0.02459950892961 3831  0.02581946983527
0.02582687484406
 3835  0.50594527442494  0.50734483592675 3838  0.55816456961395  0.55979910606418 3839  0.55496194759432
0.55666815741981
 3843  0.35503537030340  0.35597898953996 3846  0.41291516008399  0.41412499590771 3847  0.40664470946953
0.40789014007015
 3851  0.19933636199933  0.19981584284266 3854  0.25721787049295  0.25796184921040 3855  0.25069684566971
0.25145567502992
 3859  0.05406170736194  0.05414173268619 3862  0.10629515794212  0.10659600282361 3863  0.10236042648608
0.10267765768026
 3867  0.53180328192645  0.53332881202100 3870  0.47957238782935  0.48087454188358 3873  0.48142173319283
0.48264179290830
```

3877 0.38652524475412 0.38765470186453 3880 0.32863530959046 0.32950869549679 3883 0.33303706931904 0.33386377555864

3887 0.23080471811872 0.23149155516722 3890 0.17291679016355 0.17334554879948 3893 0.17702915626056 0.17742931051841

3897 0.07986709410779 0.08012570878043 3900 0.02763200761754 0.02767143864301 3903 0.02863840153812 0.02865129316875

3907 0.36678735368105 0.36769596040760 3910 0.42255236442126 0.42368542183959 3911 0.42010209999875 0.42131575514528

3915 0.21151371142542 0.21201477007442 3918 0.27013827254319 0.27088557372561 3919 0.26601578214978 0.26680725795781

3923 0.05911998871361 0.05921492196044 3926 0.11487517007891 0.11520438339244 3927 0.11194029543873 0.11229876077033

3931 0.39350363997864 0.39453250557993 3934 0.33772497982181 0.33854304414794 3937 0.33905950912374 0.33978721594239

3941 0.24105653898883 0.24173265746595 3944 0.18242246656016 0.18286185381476 3947 0.18489107049108 0.18527871875492

3951 0.08577304238898 0.08605146713278 3954 0.03001631770860 0.03006200570078 3957 0.03075539557209 0.03077022156744

3961 0.21852999918917 0.21899319003028 3964 0.27647084221925 0.27713919639802 3965 0.27483612165672 0.27558916071225

3969 0.06273108978262 0.06283004333297 3972 0.12064675151906 0.12097604970072 3973 0.11877945665763 0.11915469567202

3977 0.24544388035689 0.24602149947317 3980 0.18748656483604 0.18787549310543 3983 0.18826751471593 0.18856595003087

3987 0.08958474980303 0.08985835277587 3990 0.03166653224722 0.03171234640812 3993 0.03212164337912 0.03213148499063

3997 0.06481531500436 0.06489808051768 4000 0.12349137045074 0.12376888416887 4001 0.12272578246886 0.12307664651450

4005 0.09123154210633 0.09145262755084 4008 0.03254755442367 0.03258182389965 4011 0.03270960208527 0.03270156461507

4015 0.99502681304118 0.99892946161930 4018 0.99144684500663 0.99527176114534 4021 0.99145252885017 0.99527176114534

4025 0.97781678753018 0.98151533055183 4028 0.96514654473699 0.96871197133940 4031 0.97430766550146 0.97792139401507

4035 0.93652974473584 0.93993303458354 4038 0.91508084111518 0.91829927782838 4041 0.93317773124210 0.93649135662911

4045 0.87217999069080 0.87520646784491 4048 0.84247876891025 0.84527500843665 4051 0.86907184234161 0.87200179400632

4055 0.78634632711711 0.78892941322420 4058 0.74912124195824 0.75143726303029 4061 0.78356264930119 0.78604065320706

4065 0.68113358410524 0.68322629814331 4068 0.63729712699851 0.63909663833082 4071 0.67874668848994 0.68072458280648

4075 0.55912030048628 0.56069988410404 4078 0.50974626700513 0.51101933335552 4081 0.55719207473544 0.55864681398184

4085 0.42329439000941 0.42436717806191 4088 0.36959090876189 0.37035903641707 4091 0.42187431897275 0.42281330655449

4095 0.27697799146748 0.27758514370197 4098 0.22025802690581 0.22057927085086 4101 0.27609981536605 0.27656873228285

4105 0.12373911871909 0.12396804185182 4108 0.06534769478314 0.06536811157567 4111 0.12345401070192 0.12351411794341

4115 0.98094792561712 0.98461545485338 4118 0.98641291483774 0.99015336892751 4119 0.98640399313874 0.99015336892751

4123 0.94984879190468 0.95323977537202 4126 0.96428959777396 0.96781064138278 4127 0.95512408746914 0.95860121870711

4131 0.89536743235576 0.89839216899956 4134 0.91842983721456 0.92163720386783 4135 0.90032748373926 0.90344512506710

4139 0.81884054228978 0.82142316583367 4142 0.84995876735102 0.85276999965514 4143 0.82335885720284 0.82604321408547

4147 0.72214477940606 0.72422799817777 4150 0.76055586057892 0.76290476786958 4151 0.72610590568199 0.72830137769280

4155 0.60764977781510 0.60919993363180 4158 0.65241263838062 0.65425428876330 4159 0.61095241119837 0.61262634428765

4163 0.47815890643194 0.47917134490115 4166 0.52817788366046 0.52949389772218 4167 0.48071878201163 0.48186641709586

4171	0.33683858419180	0.33734396738052	4174	0.39089143906823	0.39169560962964	4175	0.33859101677669	0.33924133949222
4179	0.18713971885122	0.18721006180255	4182	0.24390221614697	0.24425247566482	4183	0.18804499511754	0.18826301423282
4187	0.97782856562787	0.98151533055183	4190	0.97429251500323	0.97792139401507	4193	0.96515701900046	0.96871197133940
4197	0.96091016858390	0.96440477643596	4200	0.94844691669347	0.95182461554134	4203	0.94846649289968	0.95182461554134
4207	0.92034452947939	0.92354737604830	4210	0.89925968393075	0.90229075610819	4213	0.90843981090102	0.91150017878385
4217	0.85712675193463	0.85994917418431	4220	0.82793761250254	0.83053950372832	4223	0.84605924571770	0.84873158252905
4227	0.77280756881833	0.77517616964422	4230	0.73622983675918	0.73833761236408	4233	0.76285508524838	0.76506439793375
4237	0.66945392927852	0.67131575514528	4240	0.62638373711517	0.62795539858139	4243	0.66086681533574	0.66255878875816
4247	0.54959715132309	0.55092531878541	4250	0.50108856235558	0.50211083882114	4253	0.54259190667192	0.54373878329680
4257	0.41616964010575	0.41696927266073	4260	0.36340936023827	0.36390264379878	4263	0.41092228339339	0.41153012442510
4267	0.27242892131708	0.27274605920147	4270	0.21667608237209	0.21673395796246	4273	0.26909804776313	0.26918822809988
4277	0.94983701840468	0.95323977537202	4280	0.95514244224786	0.95860121870711	4281	0.96427593222055	0.96781064138278
4285	0.91973381402113	0.92286391085199	4288	0.93372535586516	0.93697046278006	4289	0.93370258332826	0.93697046278006
4293	0.86700311772884	0.86976407403706	4296	0.88933729508589	0.89226838443267	4297	0.88015264587337	0.88305896175701
4301	0.79293897510345	0.79524775913787	4304	0.82306779916383	0.82559569719157	4305	0.80493999741182	0.80740361839084
4309	0.69935751830788	0.70114980513271	4312	0.73654240230669	0.73859410389057	4313	0.70990889716920	0.71186731832090
4317	0.58855122423102	0.58978721594239	4320	0.63188153399583	0.63340587249850	4321	0.59738757033782	0.59880248232173
4325	0.46323102221101	0.46390210810402	4328	0.51164733969700	0.51262108637194	4329	0.47012961139263	0.47099314189629
4333	0.32646225915264	0.32659419076131	4336	0.37877616616554	0.37921386780707	4337	0.33124987069223	0.33158638718074
4341	0.93655303469214	0.93993303458354	4344	0.93315348821213	0.93649135662911	4347	0.91509588704228	0.91829927782838
4351	0.92035632422983	0.92354737604830	4354	0.90841162921006	0.91150017878385	4357	0.89928321703429	0.90229075610819
4361	0.88152386368509	0.88442091603673	4364	0.86132757187920	0.86406484144128	4367	0.86135900604502	0.86406484144128
4371	0.82100959197258	0.82351707784760	4374	0.79305747819098	0.79535336003562	4377	0.80225561892547	0.80456278271128
4381	0.74029847383585	0.74233551610538	4384	0.70527595737138	0.70705763927942	4387	0.72342385563312	0.72524971808015
4391	0.64136889696627	0.64287519028632	4394	0.60013374668902	0.60135181285439	4397	0.62679621566288	0.62807859842407

NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION
4401	0.52664313030611	0.52758514370197	4404	0.48020462205389	0.48083870902465	4407	0.51473800341861	0.51544209920142
4411	0.39892342153697	0.39930419992497	4414	0.34839625100632	0.34848576036653	4417	0.39000195357232	0.39011370484219
4421	0.89534418999001	0.89839216899956	4424	0.90035484088913	0.90344512506710	4425	0.91841177711832	0.92163720386783
4429	0.86699135351090	0.86976407403706	4432	0.88018397590795	0.88305896175701	4433	0.88931067686230	0.89226838443267
4437	0.81732496804774	0.81971950099029	4440	0.83837496461328	0.84092895610379	4441	0.83834043800484	0.84092895610379

```
 4445  0.74756527667827  0.74949071333610  4448  0.77595572717152  0.77809248866810  4449  0.76675267508901
0.76888306599244
 4453  0.65942357340593  0.66080697690249  4456  0.69445831299690  0.69609680182049  4457  0.67630312828046
0.67790472301976
 4461  0.55505856155597  0.55585197960493  4464  0.59588011313139  0.59696090149921  4465  0.56920724763916
0.57023411592954
 4469  0.43702735012925  0.43721006180255  4472  0.48263268271167  0.48312584258339  4473  0.44808782015787
0.44852245240661
 4477  0.87221423183098  0.87520646784491  4480  0.86903911684139  0.87200179400632  4483  0.84249802563493
0.84527500843665
 4487  0.85715006110828  0.85994917418431  4490  0.84602316074486  0.84873158252905  4493  0.82796453119096
0.83053950372832
 4497  0.82102139179843  0.82351707784760  4500  0.80221706094106  0.80456278271128  4503  0.79309139417191
0.79535336003562
 4507  0.76471793812138  0.76680725795781  4510  0.73869842759143  0.74058297699238  4513  0.73873851034598
0.74058297699238
 4517  0.68962281666635  0.69121609848970  4520  0.65702653734740  0.65836756052593  4523  0.66624190011570
0.66757698320160
 4527  0.59757775704516  0.59860490466207  4530  0.55920319402277  0.55994094972268  4533  0.57738031618143
0.57813302852341
 4537  0.49083279504009  0.49125407142589  4540  0.44761073876259  0.44772673439978  4543  0.47432876276753
0.47445351996946
 4547  0.81880641194258  0.82142316583367  4550  0.82339455225006  0.82604321408547  4551  0.84993676809322
0.85276999965514
 4555  0.79291574091914  0.79524775913787  4558  0.80497911979355  0.80740361839084  4559  0.82303799692817
0.82559569719157
 4563  0.74755351222176  0.74949071333610  4566  0.76679426107527  0.76888306599244  4567  0.77591885580463
0.77809248866810
 4571  0.68383663061004  0.68527871875492  4574  0.70978142733271  0.71143006073271  4575  0.70973838822524
0.71143006073271
 4579  0.60332935282917  0.60419288780832  4582  0.63534175942524  0.63645928627674  4583  0.62612096679202
0.62724986360108
 4587  0.50800972489191  0.50822982276265  4590  0.54530236011833  0.54581677190537  4591  0.52711613880043
0.52762469310464
 4595  0.78639068493159  0.78892941322420  4598  0.78352225481629  0.78604065320706  4601  0.74914424374362
0.75143726303029
 4605  0.77284182388672  0.77517616964422  4608  0.76281198940750  0.76506439793375  4611  0.73625948216247
0.73833761236408
 4615  0.74032178376783  0.74233551610538  4618  0.72337911967704  0.72524971808015  4621  0.70531152192911
0.70705763927942
 4625  0.68963461270257  0.69121609848970  4628  0.66619669307139  0.66757698320160  4631  0.65706709170705
0.65836756052593
 4635  0.62202829744040  0.62307664651450  4638  0.59267034229525  0.59346628743564  4641  0.59271630271997
0.59346628743564
 4645  0.53915879086648  0.53959497962927  4648  0.50460699717786  0.50474248207132  4651  0.51382355473292
0.51395190474699
 4655  0.72210060694629  0.72422799817777  4658  0.72614906700597  0.72830137769280  4659  0.76053047025201
0.76290476786958
 4663  0.69932338664678  0.70114980513271  4666  0.70995485045428  0.71186731832090  4667  0.73651015224922
0.73859410389057
 4671  0.65940032985057  0.66080697690249  4674  0.67635075674522  0.67790472301976  4675  0.69441999228591
0.69609680182049
 4679  0.60331758014079  0.60419288780832  4682  0.62616888269477  0.62724986360108  4683  0.63529854424532
0.63645928627674
 4687  0.53246060141787  0.53270156461507  4690  0.56064991302639  0.56115003081760  4691  0.56059771027762
0.56115003081760
 4695  0.68118697434592  0.68322629814331  4698  0.67869962552265  0.68072458280648  4701  0.63732331499869
0.63909663833082
 4705  0.66949828875321  0.67131575514528  4708  0.66081776691490  0.66255878875816  4711  0.62641538261020
0.62795539858139
 4715  0.64140313942540  0.64287519028632  4718  0.62674645890412  0.62807859842407  4721  0.60017001753329
0.60135181285439
 4725  0.59760105088557  0.59860490466207  4728  0.57732968433127  0.57813302852341  4731  0.55924461656657
0.55994094972268
```

```
 4735  0.53917057618884  0.53959497962927  4738  0.51380566540680  0.51395190474699  4741  0.50462021050412
0.50474248207132
 4745  0.60759665410718  0.60919993363180  4748  0.61100198159759  0.61262634428765  4749  0.65238448661704
0.65425428876330
 4753  0.58850703121115  0.58978721594239  4756  0.59743923110204  0.59880248232173  4757  0.63184761810439
0.63340587249850
 4761  0.55502440182276  0.55585197960493  4764  0.56925951358007  0.57023411592954  4765  0.59584150371821
0.59696090149921
 4769  0.50798645772799  0.50822982276265  4772  0.52717290829824  0.52762469310464  4773  0.54525485496645
0.54581677190537
 4777  0.55918141583417  0.56069988410404  4780  0.55713950382518  0.55864681398184  4783  0.50977500690658
0.51101933335552
 4787  0.54965052129185  0.55092531878541  4790  0.54253816921144  0.54373878329680  4793  0.50112136451921
0.50211083882114
 4797  0.52668745230757  0.52758514370197  4800  0.51468303473914  0.51544209920142  4803  0.48024209744211
0.48083870902465
 4807  0.49086700000702  0.49125407142589  4810  0.47430629661935  0.47445351996946  4813  0.44761963657204
0.44772673439978
 4817  0.47809814189699  0.47917134490115  4820  0.48077355772382  0.48186641709586  4821  0.52814765612888
0.52949389772218
 4825  0.46317784377942  0.46390210810402  4828  0.47018547584135  0.47099314189629  4829  0.51161280889031
0.51262108637194
 4833  0.43698309768245  0.43721006180255  4836  0.44814858751267  0.44852245240661  4837  0.48258955358369
0.48312584258339
 4841  0.42336173897254  0.42436717806191  4844  0.42181760411534  0.42281330655449  4847  0.36962145611278
0.37035903641707
 4851  0.41623069971002  0.41696927266073  4854  0.41086376047893  0.41153012442510  4857  0.36344390531763
0.36390264379878
 4861  0.39897670947888  0.39930419992497  4864  0.38997546051901  0.39011370484219  4867  0.34840162033190
0.34848576036653
 4871  0.33677168571069  0.33734396738052  4874  0.33864945604107  0.33924133949222  4875  0.39086011007001
0.39169560962964
 4879  0.32640137034276  0.32659419076131  4882  0.33131370658782  0.33158638718074  4883  0.37873668031651
0.37921386780707
 4887  0.27705000209212  0.27758514370197  4890  0.27603899962893  0.27656873228285  4893  0.22029123562660
0.22057927085086
 4897  0.27249613409728  0.27274605920147  4900  0.26906848388411  0.26918822809988  4903  0.21667908574784
0.21673395796246
 4907  0.18706831644009  0.18721006180255  4910  0.18811103335281  0.18826301423282  4911  0.24386540953059
0.24425247566482
 4915  0.12381486314169  0.12396804185182  4918  0.12342289826914  0.12351411794341  4921  0.06535193451641
0.06536811157567
 4925  0.99340098703516  0.99732465707428  4928  0.98790318110680  0.99174663408490  4931  0.98425992645794
0.98811523439862
 4935  0.97112287059706  0.97482011001528  4938  0.95657369933589  0.96014371300095  4941  0.95306272636163
0.95662803121455
 4945  0.92492799457488  0.92831225660525  4948  0.90168523460476  0.90489886716863  4951  0.89838991986372
0.90158547103557
 4955  0.85595097521769  0.85894627454584  4958  0.82458494925494  0.82737240803946  4961  0.82158310175220
0.82434288436907
 4965  0.76588413324302  0.76843018452787  4968  0.72716419524131  0.72947329433276  4971  0.72452669911322
0.72680224005222
 4975  0.65693636401366  0.65899279309326  4978  0.61181182584336  0.61361212713653  4981  0.60960125666201
0.61136531246694
 4985  0.53177833131957  0.53332881202100  4988  0.48135488076607  0.48264179290830  4991  0.47962449552957
0.48087454188358
 4995  0.39347605771222  0.39453250557993  4998  0.33898821533728  0.33978721594239  5001  0.33778164855326
0.33854304414794
 5005  0.24541465974831  0.24602149947317  5008  0.18819377633652  0.18856595003087  5011  0.18754597193925
0.18787549310543
 5015  0.09119942857943  0.09145262755084  5018  0.03265413913607  0.03270156461507  5021  0.03259241425791
0.03258182389965
 5025  0.95810210368111  0.96174642343239  5028  0.97068604788359  0.97445771976688  5029  0.97616337905771
0.97993850223311
```

```
 5033  0.90840679922120  0.91169622366786 5036  0.92971526951376  0.93317442234844 5037  0.93494478217763
0.93842300923754
 5041  0.83633024194855  0.83919703713035 5044  0.86584037980463  0.86891327362344 5045  0.87069644746591
0.87380042730712
 5049  0.74364100666843  0.74603403440339 5052  0.78062873960010  0.78325659634400 5053  0.78499485173404
0.78766197887905
 5057  0.63261299408752  0.63450119780442 5060  0.67617099796910  0.67831354218297 5061  0.67994223853224
0.68212867843074
 5065  0.50596918067447  0.50734483592675 5068  0.55502930428800  0.55666815741981 5069  0.55811452389205
0.55979910606418
 5073  0.36681448612610  0.36769596040760 5076  0.42017405326855  0.42131575514528 5077  0.42249743121674
0.42368542183959
 5081  0.21855945263131  0.21899319003028 5084  0.27491084460140  0.27558916071225 5085  0.27641244963595
0.27713919639802
 5089  0.06483638937585  0.06489808051768 5092  0.12279952946751  0.12307664651450 5093  0.12342986653026
0.12376888416887
 5097  0.95407597204653  0.95782627229675 5100  0.93977821887416  0.94340572588151 5103  0.92753682186429
0.93109948672708
 5107  0.90869763350055  0.91212918069330 5110  0.88585613517419  0.88912395204085 5113  0.87433475619261
0.87752579051652
 5117  0.84092847966583  0.84397244147798 5120  0.81010376389910  0.81294899566772 5123  0.79958346345245
0.80234449700232
 5127  0.75243163580962  0.75503429976952 5130  0.71438017124085  0.71675653699822 5133  0.70511779082886
0.70740681914320
 5137  0.64537881596949  0.64750470778557 5140  0.60103428046333  0.60291515360926 5143  0.59325642595960
0.59505043764815
 5147  0.52239666460521  0.52403140095101 5150  0.47284697361376  0.47422799817777 5153  0.46674487301648
0.46804193951901
 5157  0.38650229208461  0.38765470186453 5160  0.33296307529686  0.33386377555864 5163  0.32868863234357
0.32950869549679
 5167  0.24103016210641  0.24173265746595 5170  0.18481603289174  0.18527871875492 5173  0.18247952191042
0.18286185381476
 5177  0.08955835303091  0.08985835277587 5180  0.03207186833303  0.03213148499063 5183  0.03171163828291
0.03171234640812
 5187  0.87937804853335  0.88264418273435 5190  0.90000962832353  0.90343795880678 5191  0.91374027062784
0.91724757291113
 5195  0.80960756898529  0.81245524963461 5198  0.83818114551888  0.84122455084753 5199  0.85095058578519
0.85408319411027
 5203  0.71987602421419  0.72226097190448 5206  0.75569014038183  0.75829740258220 5207  0.76718632145501
0.76988845253080
 5211  0.61238624760273  0.61428223199932 5214  0.65456171156152  0.65669845562050 5215  0.66450336144254
0.66673650213674
 5219  0.48977694842968  0.49117782485650 5222  0.53727841765917  0.53892941322420 5223  0.54542166261369
0.54716728628849
 5227  0.35505842308205  0.35597898953996 5230  0.40671952912138  0.40789014007015 5231  0.41286342741279
0.41412499590771
 5235  0.21154069322610  0.21201477007442 5238  0.26609161291909  0.26680725795781 5239  0.27008231652486
0.27088557372561
 5243  0.06275241198938  0.06283004333297 5246  0.11885340899469  0.11915469567202 5247  0.12058862538306
0.12097604970072
 5251  0.87011761208896  0.87348645716948 5254  0.84824303546311  0.85145585438060 5257  0.82877775134840
0.83185851269382
 5261  0.80523378200298  0.80821720591697 5264  0.77571279575582  0.77850808099961 5267  0.75793043588551
0.76058972529065
 5271  0.72049525693565  0.72304696473567 5274  0.68405277141801  0.68639085494422 5277  0.66838953051936
0.67059269459825
 5281  0.61798287658402  0.62007290762727 5284  0.57551377529350  0.57737240803946 5287  0.56235436444403
0.56408344619528
 5291  0.50021398466918  0.50183059762910 5294  0.45276123342619  0.45413713625960 5297  0.44242957927974
0.44368459126136
 5301  0.37008108179998  0.37123155283703 5304  0.31881055606334  0.31971950099029 5307  0.31156196692518
0.31236074918584
 5311  0.23078249258020  0.23149155516722 5314  0.17695555699360  0.17742931051841 5317  0.17297011853035
0.17334554879948
```

```
 5321  0.08574969836923  0.08605146713278  5324  0.03070795017638  0.03077022156744  5327  0.03005951088124
0.03006200570078
 5331  0.76298168459307  0.76570811750848  5334  0.78991241619665  0.79282208776554  5335  0.81028279262883
0.81333559813481
 5339  0.67842456132368  0.68070344723054  5342  0.71218048712253  0.71466641012403  5343  0.73052687078548
0.73315771736796
 5347  0.57712673865458  0.57893759895656  5350  0.61687862597292  0.61891327362344  5351  0.63275057706064
0.63492706038855
 5355  0.46157674954546  0.46291638561253  5358  0.50634800577965  0.50792043827084  5359  0.51935528289254
0.52106239198028
 5363  0.33461393609108  0.33549663452332  5366  0.38330428022596  0.38442091603673  5367  0.39312634874791
0.39436743817451
 5371  0.19936024848584  0.19981584284266  5374  0.25077203351043  0.25145567502992  5375  0.25716603431936
0.25796184921040
________________________________________________________________________________________________________
NODE    FEM SOLUTION   EXACT SOLUTION    NODE   FEM SOLUTION    EXACT SOLUTION  NODE   FEM SOLUTION    EXACT SOLUTION
------------------------------------------------------------------------------------------------------------------------------------------------------


 5379  0.05914023957805  0.05921492196044  5382  0.11201204582541  0.11229876077033  5383  0.11482082145584
0.11520438339244
 5387  0.74973987943191  0.75256098042718  5390  0.72225175191698  0.72489771365710  5393  0.69764253770002
0.70010671028975
 5397  0.67085209126765  0.67325581377473  5400  0.63691804696172  0.63912395204085  5403  0.61523572849457
0.61726635234273
 5407  0.57540981172112  0.57737285457920  5410  0.53586367206990  0.53761283759456  5413  0.51764148997934
0.51922684821146
 5417  0.46575824678195  0.46727305951329  5420  0.42157178858509  0.42286391085199  5423  0.40725833535582
0.40840225586211
 5427  0.34459211612765  0.34566745093179  5430  0.29685151497560  0.29770267121937  5433  0.28679945411673
0.28752144456405
 5437  0.21489052621010  0.21555036250386  5440  0.16477026159637  0.16521100380282  5443  0.15922699626947
0.15956090107186
 5447  0.07984707552428  0.08012570878043  5450  0.02859429184637  0.02865129316875  5453  0.02767215722494
0.02767143864301
 5457  0.62029757155577  0.62238474456056  5460  0.65116268643770  0.65343795880678  5461  0.67591065501821
0.67837420599256
 5465  0.52768932170762  0.52933760084375  5468  0.56403767500920  0.56588839277437  5469  0.58545347499080
0.58748360721149
 5473  0.42204566030996  0.42325640862338  5476  0.46298388741915  0.46440477643596  5477  0.48054054287120
0.48212721227457
 5481  0.30596319385756  0.30675323891523  5484  0.35048503463795  0.35148597323061  5485  0.36375018198101
0.36489924528297
 5489  0.18229586942729  0.18269678640934  5492  0.22930659795085  0.22991241884926  5493  0.23795353135306
0.23868624784134
 5497  0.05408030846324  0.05414173268619  5500  0.10242810076297  0.10267765768026  5501  0.10624573153712
0.10659600282361
 5505  0.60471554957239  0.60688687027071  5508  0.57412643058830  0.57611969630745  5511  0.54695527888294
0.54874086390296
 5515  0.51869569450710  0.52045596566075  5518  0.48304685387847  0.48461545485338  5521  0.46020597291984
0.46158516200956
 5525  0.41986246693869  0.42120970788190  5528  0.38003013890176  0.38117837255433  5531  0.36208205026641
0.36306370151415
 5535  0.31064507712474  0.31159186918012  5538  0.26760828015470  0.26835541366451  5541  0.25499500982463
0.25560240774813
 5545  0.19372802923901  0.19430160454500  5548  0.14854472289795  0.14892465386972  5551  0.14157628560553
0.14184733440758
 5555  0.07198716794386  0.07222698955594  5558  0.02577975869893  0.02582687484406  5561  0.02460588081670
0.02459950892961
 5565  0.46527933987584  0.46670355422739  5568  0.49732900495310  0.49892946161930  5569  0.52376533843559
0.52557435787858
 5573  0.37214292765411  0.37317445414644  5576  0.40824039653270  0.40945392773414  5577  0.42991893105523
0.43132046051419
 5581  0.26979682689351  0.27045656050940  5584  0.30905506538397  0.30989627924849  5585  0.32544177910610
0.32644602194140
```

5589 0.16075556255870 0.16107912875876 5592 0.20221011454155 0.20270795588095 5593 0.21290156934113 0.21353339889618
5597 0.04769289801568 0.04773539426710 5600 0.09032973824332 0.09052828989049 5601 0.09506467622530 0.09536287489342
5605 0.44922587128508 0.45072372357340 5608 0.41835189863426 0.41968523126722 5611 0.39145046750412 0.39257771720566
5615 0.36364378217454 0.36477477532757 5618 0.32914585154408 0.33010695766590 5621 0.30800053156615 0.30878531389558
5625 0.26906254123650 0.26984386149511 5628 0.23178809990786 0.23240035520468 5631 0.21692010851857 0.21738959135769
5635 0.16780516552202 0.16826849623220 5638 0.12866940698333 0.12897128470575 5641 0.12044467098237 0.12064101560588
5645 0.06235871211223 0.06254980214093 5648 0.02233189651797 0.02236651314621 5651 0.02093467112007 0.02092185766528
5655 0.31308507652018 0.31390370611341 5658 0.34345350638841 0.34442096443183 5659 0.36872453716067 0.36989316754540
5663 0.22699375638061 0.22750034398998 5666 0.26002240955315 0.26067591038448 5667 0.27913156207029 0.27995461412740
5671 0.13526122931589 0.13549516836715 5674 0.17013979840271 0.17051215028973 5675 0.18261603758560 0.18312264899348
5679 0.04013247962483 0.04015365201645 5682 0.07600968118226 0.07614981515745 5683 0.08154712891057 0.08178159649211
5687 0.29847456837467 0.29935787718661 5690 0.27015993352716 0.27090721391756 5693 0.24633354917788 0.24690360704919
5697 0.22085704976881 0.22145140237965 5700 0.19026238481788 0.19072282870714 5703 0.17350185169698 0.17382392175333
5707 0.13775165352299 0.13809205909104 5710 0.10562677373622 0.10584221448118 5713 0.09634573301499 0.09646411461538
5717 0.05119522788892 0.05133243100633 5720 0.01833430847464 0.01835541366451 5723 0.01674767076749 0.01672904082956
5727 0.17859697094358 0.17894231397116 5730 0.20458280185453 0.20503683547307 5731 0.22594829314588 0.22656979459748
5735 0.10643291065978 0.10657486724767 5738 0.13387613217980 0.13411776966098 5739 0.14783345239639 0.14820281172333
5743 0.03158272140957 0.03158319359083 5746 0.05981591032307 0.05989627924849 5747 0.06602128107526 0.06618658376761
5751 0.16720388960744 0.16760607478254 5754 0.14404278128169 0.14434907319406 5757 0.12579780092000 0.12597813030688
5761 0.10429877209940 0.10451533715378 5764 0.07997716670339 0.08010695766590 5767 0.06986526972843 0.06991194697701
5771 0.03876798449206 0.03885108505778 5774 0.01388411636019 0.01389234298027 5777 0.01214654553557 0.01212429948811
5781 0.07497134093841 0.07503033919484 5784 0.09430131898186 0.09442096443183 5785 0.10940040641216 0.10963372937162
5789 0.02225111722529 0.02223505212040 5792 0.04214159674913 0.04216789816007 5793 0.04886440657189 0.04896183769007
5797 0.06825896903621 0.06836510074929 5800 0.05234284842476 0.05239920169315 5803 0.04164431913420 0.04163831517961
5807 0.02537803401566 0.02541309645575 5810 0.00908908034681 0.00908719670580 5813 0.00724234587251 0.00722101765502
5817 0.01236240266410 0.01233940987285 5820 0.02341301312183 0.02340120347172 5821 0.03049173915845 0.03053148867358
5825 0.01134569275074 0.01134935307775 5828 0.00406363800065 0.00405829348976 5831 0.00215028578726 0.00213993040208
5835 0.99333974943415 0.99732465707428 5838 0.98419705274772 0.98811523439862 5841 0.98774902554749 0.99174663408490
5845 0.95401440022491 0.95782627229675 5848 0.92747472538734 0.93109948672708 5851 0.93965796463381 0.94340572588151
5855 0.87005960160253 0.87348645716948 5858 0.82872091443137 0.83185851269382 5861 0.84816489778746 0.85145585438060
5865 0.74968912026023 0.75256098042718 5868 0.69759423623109 0.70010671028975 5871 0.72221742461258 0.72489771365710
5875 0.60467428732105 0.60688687027071 5878 0.54691714567674 0.54874086390296 5881 0.57413113070058 0.57611969630745

```
 5885  0.44919460602876  0.45072372357340  5888  0.39142240307322  0.39257771720566  5891  0.41838480709974
0.41968523126722
 5895  0.29845236489817  0.29935787718661  5898  0.24631425129256  0.24690360704919  5901  0.27020650236450
0.27090721391756
 5905  0.16718915498111  0.16760607478254  5908  0.12578559115888  0.12597813030688  5911  0.14408813938025
0.14434907319406
 5915  0.06825018329848  0.06836510074929  5918  0.04163772603706  0.04163831517961  5921  0.05237468254271
0.05239920169315
 5925  0.01134151344451  0.01134935307775  5928  0.00214658554163  0.00213993040208  5931  0.00407052755891
0.00405829348976
 5935  0.95788416968549  0.96174642343239  5938  0.97595043182602  0.97993850223311  5939  0.97054698142102
0.97445771976688
 5943  0.87919139162859  0.88264418273435  5946  0.91355364130196  0.91724757291113  5947  0.89986251104225
0.90343795880678
 5951  0.76283652510963  0.76570811750848  5954  0.81013470554739  0.81333559813481  5955  0.78976795403546
0.79282208776554
 5959  0.62019738703402  0.62238474456056  5962  0.67580737820644  0.67837420599256  5963  0.65103003061012
0.65343795880678
 5967  0.46522001895664  0.46670355422739  5970  0.52370560067217  0.52557435787858  5971  0.49721448384758
0.49892946161930
 5975  0.31305646963029  0.31390370611341  5978  0.36870011708311  0.36989316754540  5979  0.34336043063490
0.34442096443183
 5983  0.17858646053970  0.17894231397116  5986  0.22594700526224  0.22656979459748  5987  0.20451239127584
0.20503683547307
 5991  0.07496745017152  0.07503033919484  5994  0.10940995227352  0.10963372937162  5995  0.09425390499550
0.09442096443183
 5999  0.01235855856571  0.01233940987285  6002  0.03050208871199  0.03053148867358  6003  0.02338878627136
0.02340120347172
 6007  0.97078439288720  0.97482011001528  6010  0.95271759008610  0.95662803121455  6013  0.95615424725158
0.96014371300095
 6017  0.90838948155540  0.91212918069330  6020  0.87402743585375  0.87752579051652  6023  0.88550715617081
0.88912395204085
 6027  0.80497565606000  0.80821720591697  6030  0.75767919517171  0.76058972529065  6033  0.77545508945266
0.77850808099961
 6037  0.67065643857802  0.67325581377473  6040  0.61504911918451  0.61726635234273  6043  0.63675803588063
0.63912395204085
 6047  0.51856396703077  0.52045596566075  6050  0.46008126469634  0.46158516200956  6053  0.48297475331306
0.48461545485338
 6057  0.36356697177468  0.36477477532757  6060  0.30792574027547  0.30878531389558  6063  0.32913820034548
0.33010695766590
 6067  0.22081997180960  0.22145140237965  6070  0.17346094212580  0.17382392175333  6073  0.19028942006606
0.19072282870714
 6077  0.10428570504410  0.10451533715378  6080  0.06984340761970  0.06991194697701  6083  0.08001106469011
0.08010695766590
 6087  0.02537579507604  0.02541309645575  6090  0.00723017976709  0.00722101765502  6093  0.00910263519338
0.00908719670580
 6097  0.90793352252260  0.91169622366786  6100  0.93447620499474  0.93842300923754  6101  0.92930418520677
0.93317442234844
 6105  0.80921518985826  0.81245524963461  6108  0.85055257274678  0.85408319411027  6109  0.83780760923612
0.84122455084753
 6113  0.67813092416220  0.68070344723054  6116  0.73022293913337  0.73315771736796  6117  0.71186674317481
0.71466641012403
 6121  0.52749535049097  0.52933760084375  6124  0.58525123814512  0.58748360721149  6125  0.56379467619454
0.56588839277437
 6129  0.37203286656419  0.37317445414644  6132  0.42980847585933  0.43132046051419  6133  0.40806655174824
0.40945392773414
 6137  0.22694153600890  0.22750034398998  6140  0.27908963008581  0.27995461412740  6141  0.25990686951561
0.26067591038448
 6145  0.10641130626934  0.10657486724767  6148  0.14783220136143  0.14820281172333  6149  0.13380490909525
0.13411776966098
 6153  0.02224174141878  0.02223505212040  6156  0.04887908822129  0.04896183769007  6157  0.04210293544202
0.04216789816007
 6161  0.92433074429161  0.92831225660525  6164  0.89778883620854  0.90158547103557  6167  0.90102251221929
0.90489886716863
```

```
 6171  0.84041156057415  0.84397244147798  6174  0.79907520604220  0.80234449700232  6177  0.80956678391697
0.81294899566772
 6181  0.72008873923211  0.72304696473567  6184  0.66799905018894  0.67059269459825  6187  0.68366852887148
0.68639085494422
 6191  0.57512602713626  0.57737285457920  6194  0.51737296921596  0.51922684821146  6197  0.53563389896686
0.53761283759456
 6201  0.41969202754981  0.42120970788190  6204  0.36191879450935  0.36306370151415  6207  0.37992988033508
0.38117837255433
 6211  0.26897928801878  0.26984386149511  6214  0.21683236982079  0.21738959135769  6217  0.23177440126938
0.23240035520468
 6221  0.13772362043723  0.13809205909104  6224  0.09630215390715  0.09646411461538  6227  0.10565347305006
0.10584221448118
 6231  0.03876617771951  0.03885108505778  6234  0.01212633269196  0.01212429948811  6237  0.01390353555276
0.01389234298027
 6241  0.83563860279016  0.83919703713035  6244  0.87000120477374  0.87380042730712  6245  0.86519059087458
0.86891327362344
 6249  0.71933021386964  0.72226097190448  6252  0.76662365752429  0.76988845253080  6253  0.75514019122765
0.75829740258220
 6257  0.57674336672263  0.57893759895656  6260  0.63234766042192  0.63492706038855  6261  0.61645406458505
0.61891327362344
 6265  0.42181232439137  0.42325640862338  6268  0.48029593667475  0.48212721227457  6269  0.46268711318728
0.46440477643596
 6273  0.26967740465010  0.27045656050940  6276  0.32532688118874  0.32644602194140  6277  0.30886740042990
0.30989627924849
 6281  0.13521069158418  0.13549516836715  6284  0.18258717477578  0.18312264899348  6285  0.17003184617815
0.17051215028973
 6289  0.03156608860349  0.03158319359083  6292  0.06603497899190  0.06618658376761  6293  0.05975986513870
0.05989627924849
 6297  0.85512511081971  0.85894627454584  6300  0.82076461723539  0.82434288436907  6303  0.82371347823864
0.82737240803946
 6307  0.75175472918507  0.75503429976952  6310  0.70446328460444  0.70740681914320  6313  0.71370735606981
0.71675653699822
 6317  0.61748753164791  0.62007290762727  6320  0.56188623477584  0.56408344619528  6323  0.57506415429724
0.57737240803946
 6327  0.46544560635663  0.46727305951329  6330  0.40696479094625  0.40840225586211  6333  0.42133040480374
0.42286391085199
 6337  0.31048398480705  0.31159186918012  6340  0.25483592614262  0.25560240774812  6343  0.26752404568581
0.26835541366451
 6347  0.16774664075390  0.16826849623220  6350  0.12036950248746  0.12064101560588  6353  0.12867564699424
0.12897128470575
 6357  0.05118983626878  0.05133243100633  6360  0.01671876118048  0.01672904082956  6363  0.01835766746113
0.01835541366451
____________________________________________________________________________________________________
NODE     FEM SOLUTION    EXACT SOLUTION      NODE    FEM SOLUTION      EXACT SOLUTION   NODE     FEM SOLUTION      EXACT SOLUTION
----------------------------------------------------------------------------------------------------------------------------------


 6367  0.74278085984327  0.74603403440339  6370  0.78411478044799  0.78766197887905  6371  0.77978590267694
0.78325659634400
 6375  0.61174971669141  0.61428223199932  6378  0.66383408093170  0.66673650213674  6379  0.65389534115386
0.65669845562050
 6383  0.46116749757074  0.46291638561253  6386  0.51891735625086  0.52106239198028  6387  0.50587630326412
0.50792043827084
 6391  0.30574272286167  0.30675323891523  6394  0.36352004392264  0.36489924528297  6395  0.35018966657129
0.35148597323061
 6399  0.16066035457794  0.16107912875876  6402  0.21282201496322  0.21353339889618  6403  0.20204716208885
0.20270795588095
 6407  0.04010548113445  0.04015365201645  6410  0.08155176593843  0.08178159649211  6411  0.07593097368107
0.07614981515745
 6415  0.76487376970285  0.76843018452787  6418  0.72354344556413  0.72680224005222  6421  0.72613327442253
0.72947329433276
 6425  0.64460343198234  0.64750470778557  6428  0.59252227701694  0.59505043764815  6431  0.60029132666501
0.60291515360926
 6435  0.49969793914365  0.50183059762910  6438  0.44195174664674  0.44368459126136  6441  0.45231610412309
0.45413713625960
```

6445	0.34431080730961	0.34566745093179	6448	0.28653579726356	0.28752144456405	6451	0.29665645762551	0.29770267121937
6455	0.19361741827596	0.19430160454500	6458	0.14145557395914	0.14184733440758	6461	0.14851223325742	0.14892465386972
6465	0.06234366060318	0.06254980214093	6468	0.02089585455991	0.02092185766528	6471	0.02235639136823	0.02236651314621
6475	0.63164947747888	0.63450119780442	6478	0.67893450651977	0.68212867843074	6479	0.67519513349108	0.67831354218297
6483	0.48912425389147	0.49117782485650	6486	0.54471714354694	0.54716728628849	6487	0.53656695126091	0.53892941322420
6491	0.33424611978273	0.33549663452332	6494	0.39272422922810	0.39436743817451	6495	0.38285341135504	0.38442091603673
6499	0.18213396697126	0.18269678640934	6502	0.23779128421142	0.23868624784134	6503	0.22906305762730	0.22991241884926
6507	0.04765099178690	0.04773539426710	6510	0.09504888791857	0.09536287489342	6511	0.09022051721331	0.09052828989049
6515	0.65580245060685	0.65899279309326	6518	0.60852283564187	0.61136531246694	6521	0.61068889809000	0.61361212713653
6525	0.52159960953525	0.52403140095101	6528	0.46601149734664	0.46804193951901	6531	0.47211601485454	0.47422799817777
6535	0.36962084991486	0.37123155283703	6538	0.31114679962933	0.31236074918584	6541	0.31844751378577	0.31971950099029
6545	0.21469707899585	0.21555036250386	6548	0.15904073635313	0.15956090107186	6551	0.16467307746678	0.16521100380282
6555	0.07195378203398	0.07222698955594	6558	0.02455528745723	0.02459950892961	6561	0.02580152668765	0.02582687484406
6565	0.50498631624292	0.50734483592675	6568	0.55705534184376	0.55979910606418	6569	0.55399787402542	0.55666815741981
6573	0.35447683305891	0.35597898953996	6576	0.41221089116796	0.41412499590771	6577	0.40604683491674	0.40789014007015
6581	0.19909912803163	0.19981584284266	6584	0.25687426017956	0.25796184921040	6585	0.25041052424562	0.25145567502992
6589	0.05401687049839	0.05414173268619	6592	0.10619323420992	0.10659600282361	6593	0.10227675197403	0.10267765768026
6597	0.53060299721860	0.53332881202100	6600	0.47854186613431	0.48087454188358	6603	0.48023070333593	0.48264179290830
6607	0.38577904722337	0.38765470186453	6610	0.32805174580049	0.32950869549679	6613	0.33234632984488	0.33386377555864
6617	0.23045931843141	0.23149155516722	6620	0.17268755493708	0.17334554879948	6623	0.17675355110472	0.17742931051841
6627	0.07978255421812	0.08012570878043	6630	0.02760683820746	0.02767143864301	6633	0.02860786150154	0.02865129316875
6637	0.36592069306966	0.36769596040760	6640	0.42148790503636	0.42368542183959	6641	0.41918659838911	0.42131575514528
6645	0.21112749389759	0.21201477007442	6648	0.26958675041728	0.27088557372561	6649	0.26555336793920	0.26680725795781
6653	0.05904490202172	0.05921492196044	6656	0.11470631215272	0.11520438339244	6657	0.11180023684655	0.11229876077033
6661	0.39236973322681	0.39453250557993	6664	0.33681433320123	0.33854304414794	6667	0.33798578667242	0.33978721594239
6671	0.24049837209089	0.24173265746595	6674	0.18204751887546	0.18286185381476	6677	0.18443816431392	0.18527871875492
6681	0.08563318546318	0.08605146713278	6684	0.02997438541896	0.03006200570078	6687	0.03070476977465	0.03077022156744
6691	0.21789680156027	0.21899319003028	6694	0.27558910938239	0.27713919639802	6695	0.27409644703531	0.27558916071225
6699	0.06260651911050	0.06283004333297	6702	0.12036627947197	0.12097604970072	6703	0.11854838411046	0.11915469567202
6707	0.24452969912611	0.24602149947317	6710	0.18685670478664	0.18787549310543	6713	0.18748833856653	0.18856595003087
6717	0.08934887244715	0.08985835277587	6720	0.03159646874695	0.03171234640812	6723	0.03203547263814	0.03213148499064
6727	0.06459618410689	0.06489808051768	6730	0.12299279140657	0.12376888416887	6731	0.12232935167714	0.12307664651450

```
 6735  0.09078052896232  0.09145262755084  6738  0.03241834582225  0.03258182389965  6741  0.03252851479562
0.03270156461507
 6745  0.99328568852392  0.99732465707428  6748  0.98768858130163  0.99174663408490  6751  0.98406237434894
0.98811523439862
 6755  0.97073695138961  0.97482011001528  6758  0.95609491585281  0.96014371300095  6761  0.95262387073321
0.95662803121455
 6765  0.92428745468862  0.92831225660525  6768  0.90096267928284  0.90489886716863  6771  0.89772994564208
0.90158547103557
 6775  0.85508432175218  0.85894627454584  6778  0.82365208275637  0.82737240803946  6781  0.82073491160435
0.82434288436907
 6785  0.76483424953605  0.76843018452787  6788  0.72606976694042  0.72947329433276  6791  0.72353788932054
0.72680224005222
 6795  0.65576339174501  0.65899279309326  6798  0.61062322251079  0.61361212713653  6801  0.60853703634661
0.61136531246694
 6805  0.53056399457083  0.53332881202100  6808  0.48016325280361  0.48264179290830  6811  0.47857205384628
0.48087454188358
 6815  0.39233074497103  0.39453250557993  6818  0.33791734336368  0.33978721594239  6821  0.33685731256465
0.33854304414794
 6825  0.24449100046894  0.24602149947317  6828  0.18742001294941  0.18856595003087  6831  0.18690975838290
0.18787549310543
 6835  0.09074542123557  0.09145262755084  6838  0.03248522577850  0.03270156461507  6841  0.03246397303981
0.03258182389965
 6845  0.95773173027225  0.96174642343239  6848  0.97040094841752  0.97445771976688  6849  0.97584997616007
0.97993850223311
 6853  0.90782226019678  0.91169622366786  6856  0.92920803031892  0.93317442234844  6857  0.93438644318426
0.93842300923754
 6861  0.83556124851164  0.83919703713035  6864  0.86513598649390  0.86891327362344  6865  0.86991871105526
0.87380042730712
 6869  0.74273112911691  0.74603403440339  6872  0.77976532929378  0.78325659634400  6873  0.78403691782092
0.78766197887905
 6877  0.63162211944265  0.63450119780442  6880  0.67520199689347  0.67831354218297  6881  0.67885944888690
0.68212867843074
 6885  0.50497707400937  0.50734483592675  6888  0.55402651720197  0.55666815741981  6889  0.55698204141391
0.55979910606418
 6893  0.36592621762438  0.36769596040760  6896  0.41923225403205  0.42131575514528  6897  0.42141602112044
0.42368542183959
 6901  0.21791451797173  0.21899319003028  6904  0.27415516192622  0.27558916071225  6905  0.27551891808632
0.27713919639802
 6909  0.06461492770353  0.06489808051768  6912  0.12239682535319  0.12307664651450  6913  0.12292582464166
0.12376888416887
 6917  0.95377653477886  0.95782627229675  6920  0.93940333871684  0.94340572588151  6923  0.92719689582682
0.93109948672708
 6927  0.90820330266157  0.91212918069330  6930  0.88529495395757  0.88912395204085  6933  0.87382722805326
0.87752579051652
 6937  0.84026590410269  0.84397244147798  6940  0.80938754505691  0.81294899566772  6943  0.79893831786042
0.80234449700232
 6947  0.75164024711846  0.75503429976952  6950  0.71355358393672  0.71675653699822  6953  0.70437752590078
0.70740681914320
 6957  0.64451267468683  0.64750470778557  6960  0.60015747662553  0.60291515360926  6963  0.59247759680783
0.59505043764815
 6967  0.52152695364527  0.52403140095101  6970  0.47199835273043  0.47422799817777  6973  0.46599985234486
0.46804193951901
 6977  0.38572053844533  0.38765470186453  6980  0.33224272784145  0.33386377555864  6983  0.32806690159925
0.32950869549679
 6987  0.24045154712130  0.24173265746595  6990  0.18434792786887  0.18527871875492  6993  0.18208483412119
0.18286185381476
 6997  0.08931549824420  0.08985835277587  7000  0.03198327905741  0.03213148499064  7003  0.03163829805248
0.03171234640812
 7007  0.87892936448820  0.88264418273435  7010  0.89961935914056  0.90343795880678  7011  0.91330896048908
0.91724757291113
 7015  0.80902314449967  0.81245524963461  7018  0.83764374986074  0.84122455084753  7019  0.85035482576557
0.85408319411027
 7023  0.71919445608241  0.72226097190448  7026  0.75503967447727  0.75829740258220  7027  0.76646188286991
0.76988845253080
```

```
 7031  0.61165910612382  0.61428223199932 7034  0.65384472602130  0.65669845562050 7035  0.66369961931525
0.66673650213674
 7039  0.48907005370497  0.49117782485650 7042  0.53655525183557  0.53892941322420 7043  0.54460356366540
0.54716728628849
 7047  0.35445246455639  0.35597898953996 7050  0.40606533288588  0.40789014007015 7051  0.41211379942239
0.41412499590771
 7055  0.21112825807088  0.21201477007442 7058  0.26559538502907  0.26680725795781 7059  0.26950356181116
0.27088557372561
 7063  0.06262040528240  0.06283004333297 7066  0.11860805888325  0.11915469567202 7067  0.12029678205909
0.12097604970072
 7071  0.86974006541748  0.87348645716948 7074  0.84781568128131  0.85145585438060 7077  0.82839330563567
0.83185851269382
 7081  0.80473225818668  0.80821720591697 7084  0.77517301979761  0.77850808099961 7087  0.75744660043051
0.76058972529065
 7091  0.71990399890957  0.72304696473567 7094  0.68343900121770  0.68639085494422 7097  0.66784213246185
0.67059269459825
 7101  0.61734750874676  0.62007290762727 7104  0.57487606730164  0.57737240803946 7107  0.56178945580074
0.56408344619528
 7111  0.49959203676272  0.50183059762910 7114  0.45216155667149  0.45413713625960 7117  0.44190303045663
0.44368459126136
 7121  0.36954147930158  0.37123155283703 7124  0.31832141271706  0.31971950099029 7127  0.31113711209226
0.31236074918584
 7131  0.23040154949969  0.23149155516722 7134  0.17665327269085  0.17742931051841 7137  0.17271049794778
0.17334554879948
 7141  0.08559720120326  0.08605146713278 7144  0.03065283365927  0.03077022156744 7147  0.03001245800281
0.03006200570078
 7151  0.76254353312430  0.76570811750848 7154  0.78950799711948  0.79282208776554 7155  0.80983216111092
0.81333559813481
 7159  0.67792006102126  0.68070344723054 7162  0.71169646113861  0.71466641012403 7163  0.72998535625304
0.73315771736796
 7167  0.57659777653437  0.57893759895656 7170  0.61635332323921  0.61891327362344 7171  0.63215964912851
0.63492706038855
 7175  0.46107423373626  0.46291638561253 7178  0.50582908009546  0.50792043827084 7179  0.51876731791060
0.52106239198028
 7183  0.33419565296951  0.33549663452332 7186  0.38284736883685  0.38442091603673 7187  0.39260397172641
0.39436743817451
 7191  0.19908485189684  0.19981584284266 7194  0.25043655608075  0.25145567502992 7195  0.25677857557896
0.25796184921040
 7199  0.05905424583069  0.05921492196044 7202  0.11185111274668  0.11229876077033 7203  0.11463335373380
0.11520438339244
 7207  0.74936528651765  0.75256098042718 7210  0.72185012638878  0.72489771365710 7213  0.69728424367923
0.70010671028975
 7217  0.67041543702155  0.67325581377473 7220  0.63646712061711  0.63912395204085 7223  0.61483510942524
0.61726635234273
 7227  0.57494764767716  0.57737285457920 7230  0.53540295375981  0.53761283759456 7233  0.51723431654779
0.51922684821146
 7237  0.46531462394481  0.46727305951329 7240  0.42114768366975  0.42286391085199 7243  0.40688587048184
0.40840225586211
 7247  0.34421635252814  0.34566745093179 7250  0.29651420025416  0.29770267121937 7253  0.28650522425058
0.28752144456405
 7257  0.21463203539737  0.21555036250386 7260  0.16456693229928  0.16521100380282 7263  0.15905069882203
0.15956090107186
 7267  0.07974561383454  0.08012570878043 7270  0.02855773629795  0.02865129316875 7273  0.02764054368188
0.02767143864301
 7277  0.61992819687701  0.62238474456056 7280  0.65080675920260  0.65343795880678 7281  0.67551065684328
0.67837420599256
 7285  0.52730733231715  0.52933760084375 7288  0.56365622011511  0.56588839277437 7289  0.58502291414007
0.58748360721149
 7293  0.42168884460756  0.42325640862338 7296  0.46261295417090  0.46440477643596 7297  0.48011961006751
0.48212721227457
 7301  0.30567179883949  0.30675323891523 7304  0.35016440618432  0.35148597323061 7305  0.36338411440685
0.36489924528297
 7309  0.18210747532265  0.18269678640934 7312  0.22907563430659  0.22991241884926 7313  0.23768824276869
0.23868624784134
```

7317	0.05402214805117	0.05414173268619	7320	0.10231892790653	0.10267765768026	7321	0.10611978188827	0.10659600282361
7325	0.60439684044469	0.60688687027071	7328	0.57379866813206	0.57611969630745	7331	0.54666544709102	0.54874086390296

NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION
7335	0.51836235216337	0.52045596566075	7338	0.48271624920406	0.48461545485338	7341	0.45991476876172	0.46158516200956
7345	0.41954704956588	0.42120970788190	7348	0.37973035829527	0.38117837255433	7351	0.36181914907555	0.36306370151415
7355	0.31038211013594	0.31159186918012	7358	0.26737361105835	0.26835541366451	7361	0.25479010265924	0.25560240774812
7365	0.19354982422775	0.19430160454500	7368	0.14840518651426	0.14892465386972	7371	0.14145486932345	0.14184733440758
7375	0.07191791250704	0.07222698955594	7378	0.02575484372556	0.02582687484406	7381	0.02458422826972	0.02459950892961
7385	0.46500609647573	0.46670355422739	7388	0.49705501577986	0.49892946161930	7389	0.52345465503882	0.52557435787858
7393	0.37189072868642	0.37317445414644	7396	0.40797700565945	0.40945392773414	7397	0.42961911006265	0.43132046051419
7401	0.26959340464540	0.27045656050940	7404	0.30883020731601	0.30989627924849	7405	0.32518477276545	0.32644602194140
7409	0.16062544423221	0.16107912875876	7412	0.20205001508663	0.20270795588095	7413	0.21271789834895	0.21353339889618
7417	0.04765297441786	0.04773539426710	7420	0.09025467850060	0.09052828989049	7421	0.09497835029008	0.09536287489342
7425	0.44898839802654	0.45072372357340	7428	0.41811724784681	0.41968523126722	7431	0.39124509202019	0.39257771720566
7435	0.36342135524413	0.36477477532757	7438	0.32893528834432	0.33010695766590	7441	0.30781679043119	0.30878531389558
7445	0.26887903507625	0.26984386149511	7448	0.23162494423771	0.23240035520468	7451	0.21677813607318	0.21738959135769
7455	0.16768193993143	0.16826849623220	7458	0.12857317443333	0.12897128470575	7461	0.12036111062754	0.12064101560588
7465	0.06231109310584	0.06254980214093	7468	0.02231477933405	0.02236651314621	7471	0.02091981685690	0.02092185766528
7475	0.31290914868967	0.31390370611341	7478	0.34326919030893	0.34442096443183	7479	0.36851328471605	0.36989316754540
7483	0.22685300227236	0.22750034398998	7486	0.25986636453136	0.26067591038448	7487	0.27895222295823	0.27995461412740
7491	0.13517178851355	0.13549516836715	7494	0.17002950729769	0.17051215028973	7495	0.18248898323402	0.18312264899348
7499	0.04010513602059	0.04015365201645	7502	0.07595822528841	0.07614981515745	7503	0.08148776335326	0.08178159649211
7507	0.29832075833375	0.29935787718661	7510	0.27001471545899	0.27090721391756	7513	0.24620836379735	0.24690360704919
7517	0.22073103102201	0.22145140237965	7520	0.19015060515188	0.19072282870714	7523	0.17340566712166	0.17382392175333
7527	0.13766751625408	0.13809205909104	7530	0.10556117280139	0.10584221448118	7533	0.09628935917072	0.09646411461538
7537	0.05116282497377	0.05133243100633	7540	0.01832266655513	0.01835541366451	7543	0.01673766873206	0.01672904082956
7547	0.17850229138996	0.17894231397116	7550	0.20447763689378	0.20503683547307	7551	0.22582570894014	0.22656979459748
7555	0.10637299691540	0.10657486724767	7558	0.13380215025787	0.13411776966098	7559	0.14774708713344	0.14820281172333
7563	0.03156444530567	0.03158319359083	7566	0.05978149843489	0.05989627924849	7567	0.06598107260715	0.06618658376761
7571	0.16712090137518	0.16760607478254	7574	0.14396928073577	0.14434907319406	7577	0.12573630394852	0.12597813030688
7581	0.10424356531174	0.10451533715378	7584	0.07993416564021	0.08010695766590	7587	0.06982932020765	0.06991194697701

7591 0.03874676803442 0.03885108505778 7594 0.01387649588025 0.01389234298027 7597 0.01214017482223 0.01212429948811
7601 0.07493388208108 0.07503033919484 7604 0.09425502537327 0.09442096443183 7605 0.10934450226927 0.10963372937162
7609 0.02223970618647 0.02223505212040 7612 0.04212010355896 0.04216789816007 7613 0.04883843765621 0.04896183769007
7617 0.06822631458330 0.06836510074929 7620 0.05231742903013 0.05239920169315 7623 0.04162465306533 0.04163831517961
7627 0.02536550039027 0.02541309645575 7630 0.00908457940672 0.00908719670580 7633 0.00723886322331 0.00722101765502
7637 0.01235653615453 0.01233940987285 7640 0.02340196099947 0.02340120347172 7641 0.03047697866431 0.03053148867358
7645 0.01134041002791 0.01134935307775 7648 0.00406174121339 0.00405829348976 7651 0.00214929559358 0.00213993040208
7654 0.99328473757371 0.99732465707428 7657 0.98405836468225 0.98811523439862 7659 0.98769930965085 0.99174663408490
7663 0.95377121753276 0.95782627229675 7666 0.92718862010278 0.93109948672708 7669 0.93943751215047 0.94340572588151
7673 0.86973071868275 0.87348645716948 7676 0.82838149070448 0.83185851269382 7679 0.84786891773321 0.85145585438060
7683 0.74935284355018 0.75256098042718 7686 0.69727001285005 0.70010671028975 7689 0.72191719627342 0.72489771365710
7693 0.60438253976317 0.60688687027071 7696 0.54665015793457 0.54874086390296 7699 0.57387298544562 0.57611969630745
7703 0.44897365654321 0.45072372357340 7706 0.39123020142281 0.39257771720566 7709 0.41819150642125 0.41968523126722
7713 0.29830702986512 0.29935787718661 7716 0.24619528156377 0.24690360704919 7719 0.27008159523994 0.27090721391756
7723 0.16710952727640 0.16760607478254 7726 0.12572624383891 0.12597813030688 7729 0.14402214044019 0.14434907319406
7733 0.06821836617806 0.06836510074929 7736 0.04161845361438 0.04163831517961 7739 0.05235085298307 0.05239920169315
7743 0.01133628649768 0.01134935307775 7746 0.00214560019824 0.00213993040208 7749 0.00406867064770 0.00405829348976
7753 0.95773049721305 0.96174642343239 7756 0.97586103324992 0.97993850223311 7757 0.97037774556828 0.97445771976688
7761 0.87893220571327 0.88264418273435 7764 0.91332366230203 0.91724757291113 7765 0.89957508117134 0.90343795880678
7769 0.76254936853512 0.76570811750848 7772 0.80985045619991 0.81333559813481 7773 0.78944705026045 0.79282208776554
7777 0.61993574287818 0.62238474456056 7780 0.67553142276636 0.67837420599256 7781 0.65073509735019 0.65343795880678
7785 0.46501389028749 0.46670355422739 7788 0.52347651427054 0.52557435787858 7789 0.49697963554531 0.49892946161930
7793 0.31291568166348 0.31390370611341 7796 0.36853473540198 0.36989316754540 7797 0.34319744203464 0.34442096443183
7801 0.17850613411482 0.17894231397116 7804 0.22584525624925 0.22656979459748 7805 0.20441649505552 0.20503683547307
7809 0.07493375751904 0.07503033919484 7812 0.10936075726695 0.10963372937162 7813 0.09421037446083 0.09442096443183
7817 0.01235296485788 0.01233940987285 7820 0.03048831659321 0.03053148867358 7821 0.02337806932233 0.02340120347172
7824 0.97074196485829 0.97482011001528 7827 0.95261009154045 0.95662803121455 7829 0.95611576203014 0.96014371300095
7833 0.90820314615313 0.91212918069330 7836 0.87380974414705 0.87752579051652 7839 0.88533917930337 0.88912395204085
7843 0.80472848076331 0.80821720591697 7846 0.75742608235406 0.76058972529065 7849 0.77523388689752 0.77850808099961
7853 0.67040911584783 0.67325581377473 7856 0.61481278576542 0.61726635234273 7859 0.63653861979636 0.63912395204085
7863 0.51835479902608 0.52045596566075 7866 0.45989202727945 0.46158516200956 7869 0.48279131416613 0.48461545485338
7873 0.36341397651239 0.36477477532757 7876 0.30779502740668 0.30878531389558 7879 0.32900646656067 0.33010695766590

7883 0.22072516702919 0.22145140237965 7886 0.17338611782254 0.17382392175333 7889 0.19021075356790 0.19072282870714

7893 0.10424029105444 0.10451533715378 7896 0.06981282937455 0.06991194697701 7899 0.07997703548651 0.08010695766590

7903 0.02536452096988 0.02541309645575 7906 0.00722694211699 0.00722101765502 7909 0.00909862994009 0.00908719670580

7913 0.90781592986354 0.91169622366786 7916 0.93440736173297 0.93842300923754 7917 0.92917394539014 0.93317442234844

7921 0.80902047178824 0.81245524963461 7924 0.85037854977521 0.85408319411027 7925 0.83759045490975 0.84122455084753

7929 0.67791973991678 0.68070344723054 7932 0.73001212545625 0.73315771736796 7933 0.71162927611068 0.71466641012403

7937 0.52730794235396 0.52933760084375 7940 0.58505143966441 0.58748360721149 7941 0.56358167241911 0.56588839277437

7945 0.37189072389300 0.37317445414644 7948 0.42964789763743 0.43132046051419 7949 0.40790231747467 0.40945392773414

7953 0.22685083385153 0.22750034398998 7956 0.27897970286979 0.27995461412740 7957 0.25979872620324 0.26067591038448

7961 0.10636720454444 0.10657486724767 7964 0.14777171648785 0.14820281172333 7965 0.13374796616132 0.13411776966098

7969 0.02223215158963 0.02223505212040 7972 0.04885831659526 0.04896183769007 7973 0.04208456499109 0.04216789816007

7976 0.92429816748018 0.92831225660525 7979 0.89770680154709 0.90158547103557 7981 0.90099312960711 0.90489886716863

7985 0.84027090784352 0.84397244147798 7988 0.79891204683234 0.80234449700232 7991 0.80944072736129 0.81294899566772

7995 0.71990586042253 0.72304696473567 7998 0.66781338358518 0.67059269459825 8001 0.68350597289562 0.68639085494422

8005 0.57494755238461 0.57737285457920 8008 0.51720438490200 0.51922684821146 8011 0.53547706459084 0.53761283759456

8015 0.41954633861949 0.42120970788190 8018 0.36178940474056 0.36306370151415 8021 0.37980422338075 0.38117837255433

8025 0.26887905084036 0.26984386149511 8028 0.21674985671286 0.21738959135769 8031 0.23169112897478 0.23240035520468

8035 0.13766941210633 0.13809205909104 8038 0.09626352703091 0.09646411461538 8041 0.10561282981490 0.10584221448118

8045 0.03875006946801 0.03885108505778 8048 0.01212119292267 0.01212429948811 8051 0.01389781229068 0.01389234298027

8055 0.83554997686047 0.83919703713035 8058 0.86994898671378 0.87380042730712 8059 0.86509186355358 0.86891327362344

8063 0.71918631580695 0.72226097190448 8066 0.76649403340071 0.76988845253080 8067 0.75497865883709 0.75829740258220

8071 0.57659126075344 0.57893759895656 8074 0.63219419827888 0.63492706038855 8075 0.61628151301878 0.61891327362344

8079 0.42168242015385 0.42325640862338 8082 0.48015512485972 0.48212721227457 8083 0.46253727942992 0.46440477643596

8087 0.26958546470219 0.27045656050940 8090 0.32521966537506 0.32644602194140 8091 0.30875792245831 0.30989627924849

8095 0.13516077320369 0.13549516836715 8098 0.18252163151776 0.18312264899348 8099 0.16996744003065 0.17051215028973

8103 0.03155352418059 0.03158319359083 8106 0.06600940330038 0.06618658376761 8107 0.05973580186757 0.05989627924849

8110 0.85510047455317 0.85894627454584 8113 0.82070297261133 0.82434288436907 8115 0.82369138967416 0.82737240803946

8119 0.75165027724396 0.75503429976952 8122 0.70434309629989 0.70740681914320 8125 0.71361440048697 0.71675653699822

8129 0.61735492974641 0.62007290762727 8132 0.56175313953112 0.56408344619528 8135 0.57494745733007 0.57737240803946

8139 0.46532069087636 0.46727305951329 8142 0.40684898662129 0.40840225586211 8145 0.42122251027938 0.42286391085199

8149 0.31038814772085 0.31159186918012 8152 0.25475395566946 0.25560240774812 8155 0.26744434024554 0.26835541366451

8159 0.16768917171235 0.16826849623220 8162 0.12032681523608 0.12064101560588 8165 0.12863253142675 0.12897128470575

```
 8169   0.05117058476899   0.05133243100633  8172   0.01671226367370   0.01672904082956  8175   0.01835082636863
0.01835541366451
 8179   0.74271518696958   0.74603403440339  8182   0.78407580408290   0.78766197887905  8183   0.77971225016119
0.78325659634400
 8187   0.61164567097957   0.61428223199932  8190   0.66373938581085   0.66673650213674  8191   0.65377746730851
0.65669845562050
 8195   0.46106162587878   0.46291638561253  8198   0.51880875096260   0.52106239198028  8199   0.50575435927539
0.50792043827084
 8203   0.30565839883329   0.30675323891523  8206   0.36342566161479   0.36489924528297  8207   0.35008937390543
0.35148597323061
 8211   0.16060963224000   0.16107912875876  8214   0.21275790307868   0.21353339889618  8215   0.20198176709039
0.20270795588095
 8219   0.04009124403672   0.04015365201645  8222   0.08152400734626   0.08178159649211  8223   0.07590371407911
0.07614981515745
 8226   0.76485544482560   0.76843018452787  8229   0.72349793673123   0.72680224005222  8231   0.72611696196935
0.72947329433276
 8235   0.64452746733114   0.64750470778557  8238   0.59243583093585   0.59505043764815  8241   0.60022441111368
0.60291515360926
 8245   0.49960479196474   0.50183059762910  8248   0.44185998628344   0.44368459126136  8251   0.45223556179730
0.45413713625960
 8255   0.34422835372303   0.34566745093179  8258   0.28646220157008   0.28752144456405  8261   0.29658781770874
0.29770267121937
 8265   0.19356233188804   0.19430160454500  8268   0.14141307790202   0.14184733440758  8271   0.14847087999312
0.14892465386972
 8275   0.06232322407717   0.06254980214093  8278   0.02088866796780   0.02092185766528  8281   0.02234912796337
0.02236651314621
 8285   0.63160188834391   0.63450119780442  8288   0.67890598398654   0.68212867843074  8289   0.67514125897584
0.67831354218297
 8293   0.48905162016337   0.49117782485650  8296   0.54464994231534   0.54716728628849  8297   0.53648337405354
0.53892941322420
 8301   0.33417719593368   0.33549663452332  8304   0.39265121464044   0.39436743817451  8305   0.38277151307764
0.38442091603673
 8309   0.18208734356322   0.18269678640934  8312   0.23773470342324   0.23868624784134  8313   0.22900298793202
0.22991241884926
 8317   0.04763650934842   0.04773539426710  8320   0.09502169356117   0.09536287489342  8321   0.09019279531639
0.09052828989049
 8324   0.65578910592768   0.65899279309326  8327   0.60849004566024   0.61136531246694  8329   0.61067714133864
0.61361212713653
 8333   0.52154612321040   0.52403140095101  8336   0.46595174800117   0.46804193951901  8339   0.47206973464879
0.47422799817777
 8343   0.36955920522734   0.37123155283703  8346   0.31108833699221   0.31236074918584  8349   0.31839615927849
0.31971950099029
 8353   0.21464958156739   0.21555036250386  8356   0.15900251000686   0.15956090107186  8359   0.16463740240370
0.16521100380282
 8363   0.07193416802876   0.07222698955594  8366   0.02454812701611   0.02459950892961  8369   0.02579455449737
0.02582687484406
 8373   0.50495303849376   0.50734483592675  8376   0.55703507324544   0.55979910606418  8377   0.55395960318317
0.55666815741981
 8381   0.35442944833147   0.35597898953996  8384   0.41216562047774   0.41412499590771  8385   0.40599056875178
0.40789014007015
 8389   0.19906095008433   0.19981584284266  8392   0.25683040157566   0.25796184921040  8393   0.25036136316333
0.25145567502992
 8397   0.05400354484553   0.05414173268619  8400   0.10616920155098   0.10659600282361  8401   0.10225125067349
0.10267765768026
 8404   0.53059359163660   0.53332881202100  8407   0.47851917227222   0.48087454188358  8409   0.48022256473861
0.48264179290830
 8413   0.38574359025944   0.38765470186453  8416   0.32801361091950   0.32950869549679  8419   0.33231677806009
0.33386377555864
```

NODE   FEM SOLUTION   EXACT SOLUTION   NODE   FEM SOLUTION   EXACT SOLUTION   NODE   FEM SOLUTION   EXACT SOLUTION

```
-----------------------------------------------------------------------------------------------------------------------------------------------------------

8423   0.23042374804694   0.23149155516722  8426   0.17265714251349   0.17334554879948  8429   0.17672682356054
0.17742931051841
 8433   0.07976562379289   0.08012570878043  8436   0.02760039230531   0.02767143864301  8439   0.02860184241238
0.02865129316875
```

8443	0.36589895709787	0.36769596040760	8446	0.42147423727744	0.42368542183959	8447	0.41916079785422	0.42131575514528
8451	0.21110121196948	0.21201477007442	8454	0.26955952013709	0.27088557372561	8455	0.26551953875861	0.26680725795781
8459	0.05903398096757	0.05921492196044	8462	0.11468766300961	0.11520438339244	8463	0.11177934149543	0.11229876077033
8466	0.39236349341003	0.39453250557993	8469	0.33679983527836	0.33854304414794	8471	0.33798058650729	0.33978721594239
8475	0.24047789009550	0.24173265746595	8478	0.18202765968486	0.18286185381476	8481	0.18442277031928	0.18527871875492
8485	0.08562049563395	0.08605146713278	8488	0.02996925398428	0.03006200570078	8491	0.03070025784039	0.03077022156744
8495	0.21788473641525	0.21899319003028	8498	0.27558088270819	0.27713919639802	8499	0.27408092040773	0.27558916071225
8503	0.06259899632818	0.06283004333297	8506	0.12035469227660	0.12097604970072	8507	0.11853399278333	0.11915469567202
8510	0.24452609282903	0.24602149947317	8513	0.18684915048285	0.18787549310543	8515	0.18748562863578	0.18856595003087
8519	0.08934156135770	0.08985835277587	8522	0.03159311651568	0.03171234640812	8525	0.03203287302417	0.03213148499064
8529	0.06459272938985	0.06489808051768	8532	0.12298928946282	0.12376888416887	8533	0.12232274309259	0.12307664651450
8536	0.09077924129546	0.09145262755084	8539	0.03241707042506	0.03258182389965	8541	0.03252805709123	0.03270156461507

Example 4 (square domain,refer section 7.5.2)

Table -6a(MESH NO.1- square domain,
Mesh: (number of nodes=2481,number of nine node elements= 600)
fem computed values and exact values at centroid points

NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION
646	0.96902996923803	0.98932210182092	650	0.94740423458267	0.96715571192955	653	0.93403439514655	0.95300104853184
658	0.88186168899164	0.90065277997236	662	0.82646073680609	0.84391705351846	665	0.81527146245855	0.83156603115381
670	0.70801250657272	0.72382128880320	674	0.62434905325806	0.63806991399311	677	0.61642736549350	0.62873153678521
682	0.46476233780122	0.47613712672652	686	0.36116992366837	0.36976404559162	689	0.35723293843446	0.36435241896587
694	0.17647165816887	0.18184534524342	698	0.06363100012132	0.06526309611003	701	0.06347405798344	0.06430794778558
705	0.83522529710944	0.85177670957122	708	0.88236347870189	0.90023092787438	709	0.90189058959931	0.92086345838978
713	0.65440092565543	0.66717453160046	716	0.73043595874941	0.74518526832117	717	0.74609495893444	0.76226428361846
721	0.40940173559384	0.41726466199836	724	0.50680019128999	0.51719568269362	725	0.51709523311369	0.52904936975901
729	0.12448981584229	0.12651002002600	732	0.23362623893643	0.23857938012980	733	0.23770586772619	0.24404741748384
737	0.82156689268265	0.83832983438577	740	0.77005160519013	0.78552007992821	743	0.72959964890093	0.74324008236778
747	0.66027879736022	0.67373464520468	750	0.58238469070505	0.59391705351846	753	0.55220750151548	0.56194993744403
757	0.43412242686107	0.44318961476565	760	0.33753808095102	0.34417728784688	763	0.32048702072224	0.32565221730781
767	0.16541010239666	0.16926209694974	770	0.05971733650313	0.06074705121667	773	0.05705418458479	0.05747738918909
777	0.57208532006535	0.58216061586144	780	0.63841881453065	0.65023092787438	781	0.68019784936268	0.69394532362209

785 0.35834424244575 0.36409521212307 788 0.44344239588810 0.45129264217493 789 0.47201948286052 0.48163260958091

793 0.10913477858500 0.11038963222158 796 0.20473224688998 0.20817868831867 797 0.21745285764904 0.22217433998223

801 0.54731399982079 0.55769816034822 804 0.48284451781850 0.49162745378198 807 0.43338017168823 0.44016076289038

811 0.36022278148629 0.36685961542790 814 0.28016324237411 0.28490005914351 817 0.25178133740352 0.25507490766725

821 0.13742175114996 0.14011029528825 824 0.04962312287989 0.05028466170064 827 0.04486743765352 0.04502054327026

831 0.27165875835472 0.27528558608466 834 0.33611753290053 0.34121393349943 835 0.38009082057636 0.38707029384531

839 0.08282677757508 0.08346353808552 842 0.15535903493286 0.15740001602869 843 0.17529862376390 0.17855329010351

847 0.25056343427960 0.25461884087082 850 0.19491159151197 0.19773482763574 853 0.15799886624346 0.15952908885283

857 0.09570853902865 0.09724352171812 860 0.03456678508065 0.03490005914351 863 0.02819428133358 0.02815677289956

867 0.04822502747283 0.04836745131698 870 0.09045123143407 0.09121393349943 871 0.11565623152721 0.11745420013553

875 0.04433538440485 0.04485787470676 878 0.01601570191411 0.01609919563440 881 0.00861208187306 0.00853682141769

885 0.96901580187087 0.98932210182092 888 0.93397311480852 0.95300104853184 891 0.94757307526631 0.96715571192955

895 0.82149020541956 0.83832983438577 898 0.72948888573366 0.74324008236778 901 0.77053572240536 0.78552007992821

905 0.54720233437172 0.55769816034822 908 0.43325875283685 0.44016076289038 911 0.48344099789862 0.49162745378198

915 0.25045954251627 0.25461884087082 918 0.15790885430320 0.15952908885283 921 0.19538707281805 0.19773482763574

925 0.04427310739175 0.04485787470676 928 0.00855388601390 0.00853682141769 931 0.01612489644984 0.01609919563440

935 0.83519705672752 0.85177670957122 938 0.90206292287087 0.92086345838978 939 0.88201209430657 0.90023092787438

943 0.57208803442553 0.58216061586144 946 0.68041063701139 0.69394532362209 947 0.63784765143169 0.65023092787438

951 0.27164244563678 0.27528558608466 954 0.38031236206917 0.38707029384531 955 0.33554364893400 0.34121393349943

959 0.04816841442641 0.04836745131698 962 0.11583031095364 0.11745420013553 963 0.09009052050843 0.09121393349943

967 0.88194258967577 0.90065277997236 970 0.81506166980062 0.83156603115380 973 0.82677626226379 0.84391705351846

977 0.66029308796868 0.67373464520468 980 0.55196763381299 0.56194993744403 983 0.58295119095369 0.59391705351846

987 0.36021782791205 0.36685961542790 990 0.25154537855326 0.25507490766725 993 0.28072005667386 0.28490005914351

997 0.09570645954372 0.09724352171812 1000 0.02801204659017 0.02815677289956 1003 0.03478148906288 0.03490005914351

1007 0.65428680822753 0.66717453160046 1010 0.74640841843486 0.76226428361846 1011 0.72995155643589 0.74518526832117

1015 0.35823439934388 0.36409521212307 1018 0.47234478940420 0.48163260958091 1019 0.44283652175676 0.45129264217493

1023 0.08271289345299 0.08346353808552 1026 0.17559445947110 0.17855329010351 1027 0.15485929257602 0.15740001602869

1031 0.70817872237942 0.72382128880320 1034 0.61608989890361 0.62873153678521 1037 0.62478032204907 0.63806991399311

1041 0.43422367968658 0.44318961476565 1044 0.32013745287289 0.32565221730781 1047 0.33812821174445 0.34417728784688

1051 0.13749864694906 0.14011029528825 1054 0.04459340398641 0.04502054327026 1057 0.04993046210185 0.05028466170064

1061 0.40921154418880 0.41726466199836 1064 0.51751861320615 0.52904936975901 1065 0.50622890714730 0.51719568269362

1069 0.10898204095685 0.11038963222158 1072 0.21784841462334 0.22217433998223 1073 0.20415286225039 0.20817868831867

```
 1077  0.46499725085411  0.47613712672652  1080  0.35679990295295  0.36435241896587  1083  0.36167421405766
0.36976404559162
 1087  0.16556305913541  0.16926209694974  1090  0.05671806493676  0.05747738918909  1093  0.06008902189307
0.06074705121667
 1097  0.12431372788149  0.12651002002600  1100  0.23819076136347  0.24404741748384  1101  0.23303078035169
0.23857938012980
 1105  0.17672960993770  0.18184534524342  1108  0.06310282648947  0.06430794778558  1111  0.06397121389282
0.06526309611003
 1115  0.96902996923804  0.98932210182092  1118  0.94740423458267  0.96715571192955  1121  0.93403439514655
0.95300104853184
 1125  0.88186168899164  0.90065277997236  1128  0.82646073680610  0.84391705351846  1131  0.81527146245856
0.83156603115380
 1135  0.70801250657272  0.72382128880320  1138  0.62434905325807  0.63806991399311  1141  0.61642736549350
0.62873153678521
 1145  0.46476233780123  0.47613712672652  1148  0.36116992366837  0.36976404559162  1151  0.35723293843446
0.36435241896587
 1155  0.17647165816887  0.18184534524342  1158  0.06363100012132  0.06526309611003  1161  0.06347405798344
0.06430794778558
 1165  0.83522529710944  0.85177670957122  1168  0.88236347870190  0.90023092787438  1169  0.90189058959931
0.92086345838978
 1173  0.65440092565544  0.66717453160046  1176  0.73043595874942  0.74518526832117  1177  0.74609495893445
0.76226428361846
 1181  0.40940173559384  0.41726466199836  1184  0.50680019128999  0.51719568269362  1185  0.51709523311369
0.52904936975901
 1189  0.12448981584229  0.12651002002600  1192  0.23362623893643  0.23857938012980  1193  0.23770586772619
0.24404741748384
 1197  0.82156689268265  0.83832983438577  1200  0.77005160519013  0.78552007992821  1203  0.72959964890093
0.74324008236778
 1207  0.66027879736022  0.67373464520468  1210  0.58238469070505  0.59391705351846  1213  0.55220750151548
0.56194993744403
 1217  0.43412242686107  0.44318961476565  1220  0.33753808095102  0.34417728784688  1223  0.32048702072224
0.32565221730781
 1227  0.16541010239666  0.16926209694974  1230  0.05971733650313  0.06074705121667  1233  0.05705418458479
0.05747738918909
 1237  0.57208532006535  0.58216061586144  1240  0.63841881453065  0.65023092787438  1241  0.68019784936268
0.69394532362209
 1245  0.35834424244575  0.36409521212307  1248  0.44344239588810  0.45129264217493  1249  0.47201948286052
0.48163260958091
 1253  0.10913477858500  0.11038963222158  1256  0.20473224688998  0.20817868831867  1257  0.21745285764904
0.22217433998223
 1261  0.54731399982079  0.55769816034822  1264  0.48284451781850  0.49162745378198  1267  0.43338017168824
0.44016076289038
 1271  0.36022278148629  0.36685961542790  1274  0.28016324237411  0.28490005914351  1277  0.25178133740352
0.25507490766725
 1281  0.13742175114996  0.14011029528825  1284  0.04962312287989  0.05028466170064  1287  0.04486743765352
0.04502054327026
```

NODE FEM SOLUTION EXACT SOLUTION NODE FEM SOLUTION EXACT SOLUTION NODE FEM SOLUTION EXACT SOLUTION

```
---------------------------------------------------------------------------------------------------------------------------------------------

 1291  0.27165875835472  0.27528558608466  1294  0.33611753290053  0.34121393349943  1295  0.38009082057636
0.38707029384531
 1299  0.08282677757508  0.08346353808552  1302  0.15535903493286  0.15740001602869  1303  0.17529862376390
0.17855329010351
 1307  0.25056343427959  0.25461884087082  1310  0.19491159151197  0.19773482763574  1313  0.15799886624346
0.15952908885283
 1317  0.09570853902865  0.09724352171812  1320  0.03456678508065  0.03490005914351  1323  0.02819428133358
0.02815677289956
 1327  0.04822502747283  0.04836745131698  1330  0.09045123143407  0.09121393349943  1331  0.11565623152721
0.11745420013553
 1335  0.04433538440485  0.04485787470676  1338  0.01601570191411  0.01609919563440  1341  0.00861208187306
0.00853682141769
```

1345 0.96901580187087 0.98932210182092 1348 0.93397311480852 0.95300104853184 1351 0.94757307526631 0.96715571192955

1355 0.82149020541957 0.83832983438577 1358 0.72948888573366 0.74324008236778 1361 0.77053572240536 0.78552007992821

1365 0.54720233437172 0.55769816034822 1368 0.43325875283685 0.44016076289038 1371 0.48344099789862 0.49162745378198

1375 0.25045954251627 0.25461884087082 1378 0.15790885430320 0.15952908885283 1381 0.19538707281805 0.19773482763574

1385 0.04427310739175 0.04485787470676 1388 0.00855388601390 0.00853682141769 1391 0.01612489644984 0.01609919563440

1395 0.83519705672752 0.85177670957122 1398 0.90206292287087 0.92086345838978 1399 0.88201209430657 0.90023092787438

1403 0.57208803442553 0.58216061586144 1406 0.68041063701139 0.69394532362209 1407 0.63784765143169 0.65023092787438

1411 0.27164244563678 0.27528558608466 1414 0.38031236206917 0.38707029384531 1415 0.33554364893400 0.34121393349943

1419 0.04816841442641 0.04836745131698 1422 0.11583031095364 0.11745420013553 1423 0.09009052050843 0.09121393349943

1427 0.88194258967577 0.90065277997236 1430 0.81506166980062 0.83156603115380 1433 0.82677626226380 0.84391705351846

1437 0.66029308796868 0.67373464520468 1440 0.55196763381299 0.56194993744403 1443 0.58295119095369 0.59391705351846

1447 0.36021782791205 0.36685961542790 1450 0.25154537855326 0.25507490766725 1453 0.28072005667386 0.28490005914351

1457 0.09570645954372 0.09724352171812 1460 0.02801204659017 0.02815677289956 1463 0.03478148906288 0.03490005914351

1467 0.65428680822753 0.66717453160046 1470 0.74640841843487 0.76226428361846 1471 0.72995155643589 0.74518526832117

1475 0.35823439934388 0.36409521212307 1478 0.47234478940420 0.48163260958091 1479 0.44283652175676 0.45129264217493

1483 0.08271289345299 0.08346353808552 1486 0.17559445947110 0.17855329010351 1487 0.15485929257602 0.15740001602869

1491 0.70817872237943 0.72382128880320 1494 0.61608989890361 0.62873153678521 1497 0.62478032204907 0.63806991399311

1501 0.43422367968658 0.44318961476565 1504 0.32013745287289 0.32565221730781 1507 0.33812821174445 0.34417728784688

1511 0.13749864694906 0.14011029528825 1514 0.04459340398641 0.04502054327026 1517 0.04993046210185 0.05028466170064

1521 0.40921154418880 0.41726466199836 1524 0.51751861320615 0.52904936975901 1525 0.50622890714730 0.51719568269362

1529 0.10898204095685 0.11038963222158 1532 0.21784841462334 0.22217433998223 1533 0.20415286225039 0.20817868831867

1537 0.46499725085411 0.47613712672652 1540 0.35679990295295 0.36435241896587 1543 0.36167421405767 0.36976404559162

1547 0.16556305913541 0.16926209694974 1550 0.05671806493676 0.05747738918909 1553 0.06008902189307 0.06074705121667

1557 0.12431372788149 0.12651002002600 1560 0.23819076136347 0.24404741748384 1561 0.23303078035170 0.23857938012980

1565 0.17672960993771 0.18184534524342 1568 0.06310282648947 0.06430794778558 1571 0.06397121389282 0.06526309611003

1575 0.96902996923804 0.98932210182092 1578 0.94740423458267 0.96715571192955 1581 0.93403439514655 0.95300104853184

1585 0.88186168899165 0.90065277997236 1588 0.82646073680610 0.84391705351846 1591 0.81527146245856 0.83156603115380

1595 0.70801250657273 0.72382128880320 1598 0.62434905325807 0.63806991399311 1601 0.61642736549350 0.62873153678521

1605 0.46476233780123 0.47613712672652 1608 0.36116992366837 0.36976404559162 1611 0.35723293843446 0.36435241896587

1615 0.17647165816887 0.18184534524342 1618 0.06363100012132 0.06526309611003 1621 0.06347405798344 0.06430794778558

1625 0.83522529710944 0.85177670957122 1628 0.88236347870190 0.90023092787438 1629 0.90189058959931 0.92086345838978

1633 0.65440092565544 0.66717453160046 1636 0.73043595874942 0.74518526832117 1637 0.74609495893445 0.76226428361846

1641 0.40940173559384 0.41726466199836 1644 0.50680019128999 0.51719568269362 1645 0.51709523311369 0.52904936975901
1649 0.12448981584229 0.12651002002600 1652 0.23362623893643 0.23857938012980 1653 0.23770586772619 0.24404741748384
1657 0.82156689268266 0.83832983438577 1660 0.77005160519014 0.78552007992821 1663 0.72959964890093 0.74324008236778
1667 0.66027879736022 0.67373464520468 1670 0.58238469070505 0.59391705351846 1673 0.55220750151549 0.56194993744403
1677 0.43412242686107 0.44318961476565 1680 0.33753808095102 0.34417728784688 1683 0.32048702072224 0.32565221730781
1687 0.16541010239666 0.16926209694974 1690 0.05971733650313 0.06074705121667 1693 0.05705418458479 0.05747738918909
1697 0.57208532006536 0.58216061586144 1700 0.63841881453065 0.65023092787438 1701 0.68019784936268 0.69394532362209
1705 0.35834424244575 0.36409521212307 1708 0.44344239588810 0.45129264217493 1709 0.47201948286053 0.48163260958091
1713 0.10913477858500 0.11038963222158 1716 0.20473224688998 0.20817868831867 1717 0.21745285764904 0.22217433998223
1721 0.54731399982079 0.55769816034822 1724 0.48284451781850 0.49162745378198 1727 0.43338017168823 0.44016076289038
1731 0.36022278148629 0.36685961542790 1734 0.28016324237411 0.28490005914351 1737 0.25178133740352 0.25507490766725
1741 0.13742175114996 0.14011029528825 1744 0.04962312287989 0.05028466170064 1747 0.04486743765352 0.04502054327026
1751 0.27165875835472 0.27528558608466 1754 0.33611753290053 0.34121393349943 1755 0.38009082057636 0.38707029384531
1759 0.08282677757508 0.08346353808552 1762 0.15535903493286 0.15740001602869 1763 0.17529862376390 0.17855329010351
1767 0.25056343427960 0.25461884087082 1770 0.19491159151197 0.19773482763574 1773 0.15799886624346 0.15952908885283
1777 0.09570853902865 0.09724352171812 1780 0.03456678508065 0.03490005914351 1783 0.02819428133358 0.02815677289956
1787 0.04822502747283 0.04836745131698 1790 0.09045123143407 0.09121393349943 1791 0.11565623152721 0.11745420013553
1795 0.04433538440485 0.04485787470676 1798 0.01601570191411 0.01609919563440 1801 0.00861208187306 0.00853682141769
1805 0.96901580187087 0.98932210182092 1808 0.93397311480852 0.95300104853184 1811 0.94757307526631 0.96715571192955
1815 0.82149020541957 0.83832983438577 1818 0.72948888573366 0.74324008236778 1821 0.77053572240537 0.78552007992821
1825 0.54720233437172 0.55769816034822 1828 0.43325875283685 0.44016076289038 1831 0.48344099789862 0.49162745378198
1835 0.25045954251627 0.25461884087082 1838 0.15790885430320 0.15952908885283 1841 0.19538707281805 0.19773482763574
1845 0.04427310739175 0.04485787470676 1848 0.00855388601390 0.00853682141769 1851 0.01612489644984 0.01609919563440
1855 0.83519705672753 0.85177670957122 1858 0.90206292287087 0.92086345838978 1859 0.88201209430657 0.90023092787438
1863 0.57208803442553 0.58216061586144 1866 0.68041063701139 0.69394532362209 1867 0.63784765143169 0.65023092787438
1871 0.27164244563679 0.27528558608466 1874 0.38031236206917 0.38707029384531 1875 0.33554364893400 0.34121393349943
1879 0.04816841442641 0.04836745131698 1882 0.11583031095364 0.11745420013553 1883 0.09009052050843 0.09121393349943
1887 0.88194258967577 0.90065277997236 1890 0.81506166980062 0.83156603115381 1893 0.82677626226379 0.84391705351846
1897 0.66029308796868 0.67373464520468 1900 0.55196763381300 0.56194993744403 1903 0.58295119095369 0.59391705351846
1907 0.36021782791205 0.36685961542790 1910 0.25154537855326 0.25507490766725 1913 0.28072005667386 0.28490005914351
1917 0.09570645954372 0.09724352171812 1920 0.02801204659017 0.02815677289956 1923 0.03478148906288 0.03490005914351
1927 0.65428680822753 0.66717453160046 1930 0.74640841843486 0.76226428361846 1931 0.72995155643589 0.74518526832117

```
 1935  0.35823439934388  0.36409521212307  1938  0.47234478940420  0.48163260958091  1939  0.44283652175677
0.45129264217493
 1943  0.08271289345299  0.08346353808552  1946  0.17559445947110  0.17855329010351  1947  0.15485929257602
0.15740001602869
 1951  0.70817872237942  0.72382128880320  1954  0.61608989890361  0.62873153678521  1957  0.62478032204907
0.63806991399311
 1961  0.43422367968659  0.44318961476565  1964  0.32013745287289  0.32565221730781  1967  0.33812821174445
0.34417728784688
 1971  0.13749864694906  0.14011029528825  1974  0.04459340398641  0.04502054327026  1977  0.04993046210185
0.05028466170064
 1981  0.40921154418880  0.41726466199836  1984  0.51751861320615  0.52904936975901  1985  0.50622890714730
0.51719568269362
 1989  0.10898204095685  0.11038963222158  1992  0.21784841462334  0.22217433998223  1993  0.20415286225039
0.20817868831867
 1997  0.46499725085411  0.47613712672652  2000  0.35679990295295  0.36435241896587  2003  0.36167421405767
0.36976404559162
 2007  0.16556305913542  0.16926209694974  2010  0.05671806493676  0.05747738918909  2013  0.06008902189307
0.06074705121667
 2017  0.12431372788149  0.12651002002600  2020  0.23819076136347  0.24404741748384  2021  0.23303078035170
0.23857938012980
 2025  0.17672960993771  0.18184534524342  2028  0.06310282648947  0.06430794778558  2031  0.06397121389282
0.06526309611003
 2035  0.96902996923804  0.98932210182092  2038  0.94740423458267  0.96715571192955  2041  0.93403439514654
0.95300104853184
 2045  0.88186168899164  0.90065277997236  2048  0.82646073680609  0.84391705351846  2051  0.81527146245856
0.83156603115380
 2055  0.70801250657272  0.72382128880320  2058  0.62434905325807  0.63806991399311  2061  0.61642736549350
0.62873153678521
 2065  0.46476233780123  0.47613712672652  2068  0.36116992366837  0.36976404559162  2071  0.35723293843446
0.36435241896587
 2075  0.17647165816887  0.18184534524342  2078  0.06363100012132  0.06526309611003  2081  0.06347405798344
0.06430794778558
 2085  0.83522529710943  0.85177670957122  2088  0.88236347870189  0.90023092787438  2089  0.90189058959931
0.92086345838978
 2093  0.65440092565543  0.66717453160046  2096  0.73043595874941  0.74518526832117  2097  0.74609495893445
0.76226428361846
 2101  0.40940173559384  0.41726466199836  2104  0.50680019128999  0.51719568269362  2105  0.51709523311369
0.52904936975901
 2109  0.12448981584229  0.12651002002600  2112  0.23362623893643  0.23857938012980  2113  0.23770586772619
0.24404741748384
 2117  0.82156689268265  0.83832983438577  2120  0.77005160519013  0.78552007992821  2123  0.72959964890092
0.74324008236778
 2127  0.66027879736021  0.67373464520468  2130  0.58238469070504  0.59391705351846  2133  0.55220750151548
0.56194993744403
 2137  0.43412242686107  0.44318961476565  2140  0.33753808095102  0.34417728784688  2143  0.32048702072224
0.32565221730781
 2147  0.16541010239666  0.16926209694974  2150  0.05971733650313  0.06074705121667  2153  0.05705418458479
0.05747738918909
 2157  0.57208532006535  0.58216061586144  2160  0.63841881453064  0.65023092787438  2161  0.68019784936267
0.69394532362209
 2165  0.35834424244574  0.36409521212307  2168  0.44344239588809  0.45129264217493  2169  0.47201948286052
0.48163260958091
 2173  0.10913477858500  0.11038963222158  2176  0.20473224688998  0.20817868831867  2177  0.21745285764903
0.22217433998223
 2181  0.54731399982078  0.55769816034822  2184  0.48284451781849  0.49162745378198  2187  0.43338017168823
0.44016076289038
 2191  0.36022278148628  0.36685961542790  2194  0.28016324237410  0.28490005914351  2197  0.25178133740351
0.25507490766725
 2201  0.13742175114996  0.14011029528825  2204  0.04962312287989  0.05028466170064  2207  0.04486743765352
0.04502054327026
 2211  0.27165875835472  0.27528558608466  2214  0.33611753290053  0.34121393349943  2215  0.38009082057636
0.38707029384531
 2219  0.08282677757508  0.08346353808552  2222  0.15535903493286  0.15740001602869  2223  0.17529862376390
0.17855329010351
```

```
 2227  0.25056343427959  0.25461884087082  2230  0.19491159151197  0.19773482763574  2233  0.15799886624346
0.15952908885283
 2237  0.09570853902865  0.09724352171812  2240  0.03456678508065  0.03490005914351  2243  0.02819428133358
0.02815677289956
 2247  0.04822502747283  0.04836745131698  2250  0.09045123143407  0.09121393349943  2251  0.11565623152720
0.11745420013553
 2255  0.04433538440485  0.04485787470676  2258  0.01601570191411  0.01609919563440  2261  0.00861208187306
0.00853682141769
 2264  0.96901580187087  0.98932210182092  2267  0.93397311480851  0.95300104853184  2269  0.94757307526631
0.96715571192955
 2273  0.82149020541956  0.83832983438577  2276  0.72948888573365  0.74324008236778  2279  0.77053572240536
0.78552007992821
 2283  0.54720233437172  0.55769816034822  2286  0.43325875283685  0.44016076289038  2289  0.48344099789861
0.49162745378198
 2293  0.25045954251627  0.25461884087082  2296  0.15790885430320  0.15952908885283  2299  0.19538707281804
0.19773482763574
____________________________________________________________________________________________________________
___________________
   NODE FEM SOLUTION        EXACT SOLUTION      NODE  FEM SOLUTION      EXACT SOLUTION       NODE
FEM SOLUTION     EXACT SOLUTION
---------------------------------------------------------------------------------------------------------------------------------------

 2303  0.04427310739175  0.04485787470676  2306  0.00855388601390  0.00853682141769  2309  0.01612489644984
0.01609919563440
 2313  0.83519705672752  0.85177670957122  2316  0.90206292287086  0.92086345838978  2317  0.88201209430656
0.90023092787438
 2321  0.57208803442552  0.58216061586144  2324  0.68041063701138  0.69394532362209  2325  0.63784765143168
0.65023092787438
 2329  0.27164244563678  0.27528558608466  2332  0.38031236206917  0.38707029384531  2333  0.33554364893400
0.34121393349943
 2337  0.04816841442641  0.04836745131698  2340  0.11583031095364  0.11745420013553  2341  0.09009052050843
0.09121393349943
 2344  0.88194258967576  0.90065277997236  2347  0.81506166980061  0.83156603115380  2349  0.82677626226379
0.84391705351846
 2353  0.66029308796867  0.67373464520468  2356  0.55196763381299  0.56194993744403  2359  0.58295119095368
0.59391705351846
 2363  0.36021782791205  0.36685961542790  2366  0.25154537855326  0.25507490766725  2369  0.28072005667385
0.28490005914351
 2373  0.09570645954372  0.09724352171812  2376  0.02801204659017  0.02815677289956  2379  0.03478148906288
0.03490005914351
 2383  0.65428680822753  0.66717453160046  2386  0.74640841843486  0.76226428361846  2387  0.72995155643588
0.74518526832117
 2391  0.35823439934388  0.36409521212307  2394  0.47234478940419  0.48163260958091  2395  0.44283652175676
0.45129264217493
 2399  0.08271289345299  0.08346353808552  2402  0.17559445947109  0.17855329010351  2403  0.15485929257602
0.15740001602869
 2406  0.70817872237942  0.72382128880320  2409  0.61608989890361  0.62873153678521  2411  0.62478032204907
0.63806991399311
 2415  0.43422367968658  0.44318961476565  2418  0.32013745287289  0.32565221730781  2421  0.33812821174445
0.34417728784688
 2425  0.13749864694906  0.14011029528825  2428  0.04459340398641  0.04502054327026  2431  0.04993046210185
0.05028466170064
 2435  0.40921154418880  0.41726466199836  2438  0.51751861320615  0.52904936975901  2439  0.50622890714730
0.51719568269362
 2443  0.10898204095685  0.11038963222158  2446  0.21784841462334  0.22217433998223  2447  0.20415286225039
0.20817868831867
 2450  0.46499725085411  0.47613712672652  2453  0.35679990295295  0.36435241896587  2455  0.36167421405766
0.36976404559162
 2459  0.16556305913541  0.16926209694974  2462  0.05671806493676  0.05747738918909  2465  0.06008902189307
0.06074705121667
 2469  0.12431372788149  0.12651002002600  2472  0.23819076136347  0.24404741748384  2473  0.23303078035170
0.23857938012980
 2476  0.17672960993771  0.18184534524342  2479  0.06310282648947  0.06430794778558  2481  0.06397121389282
0.06526309611003
```

Table -6b(MESH NO.2- square domain,)
Mesh:nodes=9761,number of nine node elements= 2400
fem computed values and exact values at centroid points

NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION	NODE	FEM SOLUTION	EXACT SOLUTION
2486	0.99222025782418	0.99732465707428	2490	0.98667620482132	0.99174663408490	2493	0.98309568598507	0.98811523439862
2498	0.96981113299079	0.97482011001528	2502	0.95521985947286	0.96014371300095	2505	0.95178445547329	0.95662803121455
2510	0.92349490938071	0.92831225660525	2514	0.90021827612323	0.90489886716863	2517	0.89701313985609	0.90158547103557
2522	0.85441776063309	0.85894627454584	2526	0.82303074280470	0.82737240803946	2529	0.82013464161417	0.82434288436907
2534	0.76428581390750	0.76843018452787	2538	0.72556361416198	0.72947329433276	2541	0.72304754023008	0.72680224005222
2546	0.65532539474758	0.65899279309326	2550	0.61022479835642	0.61361212713653	2553	0.60815017109950	0.61136531246694
2558	0.53022951717413	0.53332881202100	2562	0.47986604166877	0.48264179290830	2565	0.47828294938187	0.48087454188358
2570	0.39209404968079	0.39453250557993	2574	0.33771612581818	0.33978721594239	2577	0.33666132942475	0.33854304414794
2582	0.24434779669720	0.24602149947317	2586	0.18731108619539	0.18856595003087	2589	0.18680357908013	0.18787549310543
2594	0.09069302279800	0.09145262755084	2598	0.03246652968041	0.03270156461507	2601	0.03244574318887	0.03258182389965
2605	0.95688725786317	0.96174642343239	2608	0.96951084671644	0.97445771976688	2609	0.97491837166128	0.97993850223311
2613	0.90709694147438	0.91169622366786	2616	0.92843854627146	0.93317442234844	2617	0.93358441883596	0.93842300923754
2621	0.83495005301815	0.83919703713035	2624	0.86448265974883	0.86891327362344	2625	0.86924014679600	0.87380042730712
2629	0.74222817257936	0.74603403440339	2632	0.77922249405823	0.78325659634400	2633	0.78347480549538	0.78766197887905
2637	0.63122143767913	0.63450119780442	2640	0.67476364108903	0.67831354218297	2641	0.67840666624677	0.68212867843074
2645	0.50467320182057	0.50734483592675	2648	0.55368693225990	0.55666815741981	2649	0.55663199120823	0.55979910606418
2653	0.36571459674768	0.36769596040760	2656	0.41898650333195	0.42131575514528	2657	0.42116308866008	0.42368542183959
2661	0.21779176758239	0.21899319003028	2664	0.27399939874190	0.27558916071225	2665	0.27535877454550	0.27713919639802
2669	0.06457901048541	0.06489808051768	2672	0.12232849455682	0.12307664651450	2673	0.12285561409929	0.12376888416887
2677	0.95295614791020	0.95782627229675	2680	0.93862469698219	0.94340572588151	2683	0.92645515251912	0.93109948672708
2687	0.90749357193983	0.91212918069330	2690	0.88462597450644	0.88912395204085	2693	0.87318773083238	0.87752579051652
2697	0.83966359898847	0.84397244147798	2700	0.80882440132879	0.81294899566772	2703	0.79839839498352	0.80234449700232
2707	0.75114086933070	0.75503429976952	2710	0.71309153634517	0.71675653699822	2713	0.70393342397164	0.70740681914320
2717	0.64411130955465	0.64750470778557	2720	0.59979162353681	0.60291515360926	2723	0.59212523361580	0.59505043764815
2727	0.52121888683705	0.52403140095101	2730	0.47172417703671	0.47422799817777	2733	0.46573536283935	0.46804193951901
2737	0.38550169779036	0.38765470186453	2740	0.33205648248967	0.33386377555864	2743	0.32788702666642	0.32950869549679

```
 2747  0.24031880460599  0.24173265746595  2750  0.18424689356003  0.18527871875492  2753  0.18198718437679
0.18286185381476
 2757  0.08926686261119  0.08985835277587  2760  0.03196592229619  0.03213148499064  2763  0.03162151840217
0.03171234640812
 2767  0.87828607331939  0.88264418273435  2770  0.89893924942841  0.90343795880678  2771  0.91259504084072
0.91724757291113
 2775  0.80847657255828  0.81245524963461  2778  0.83706121893118  0.84122455084753  2779  0.84974527502340
0.85408319411027
 2783  0.71874152387087  0.72226097190448  2786  0.75455203889612  0.75829740258220  2787  0.76595300458065
0.76988845253080
 2791  0.61129618845498  0.61428223199932  2794  0.65344848720843  0.65669845562050  2795  0.66328704900118
0.66673650213674
 2799  0.48879355991055  0.49117782485650  2802  0.53624674936785  0.53892941322420  2803  0.54428292828387
0.54716728628849
 2807  0.35425925640497  0.35597898953996  2810  0.40584121873458  0.40789014007015  2811  0.41188119364617
0.41412499590771
 2815  0.21101593878857  0.21201477007442  2818  0.26545295758449  0.26680725795781  2819  0.26935587964225
0.27088557372561
 2823  0.06258750529141  0.06283004333297  2826  0.11854548479397  0.11915469567202  2827  0.12023193402276
0.12097604970072
 2831  0.86911658023816  0.87348645716948  2834  0.84722612084224  0.85145585438060  2837  0.82783396111787
0.83185851269382
 2841  0.80419884533594  0.80821720591697  2844  0.77467293430426  0.77850808099961  2847  0.75697090071761
0.76058972529065
 2851  0.71945869563899  0.72304696473567  2854  0.68302605006921  0.68639085494422  2857  0.66744844969248
0.67059269459825
 2861  0.61698756287882  0.62007290762727  2864  0.57454736535095  0.57737240803946  2867  0.56147552358709
0.56408344619528
 2871  0.49931450255423  0.50183059762910  2874  0.45191420534220  0.45413713625960  2877  0.44166645822800
0.44368459126136
 2881  0.36934365306398  0.37123155283703  2884  0.31815288443604  0.31971950099029  2887  0.31097576160220
0.31236074918584
 2891  0.23028127722975  0.23149155516722  2894  0.17656167696692  0.17742931051841  2897  0.17262274660091
0.17334554879948
 2901  0.08555308192753  0.08605146713278  2904  0.03063708611253  0.03077022156744  2907  0.02999736787712
0.03006200570078
 2911  0.76206479419880  0.76570811750848  2914  0.78899909235611  0.79282208776554  2915  0.80929546859688
0.81333559813481
 2919  0.67752087369727  0.68070344723054  2922  0.71126763094265  0.71466641012403  2923  0.72953418496732
0.73315771736796
 2927  0.57627628239051  0.57893759895656  2930  0.61600293992088  0.61891327362344  2931  0.63179174274857
0.63492706038855
 2935  0.46082830401801  0.46291638561253  2938  0.50555506304756  0.50792043827084  2939  0.51848005724971
0.52106239198028
 2943  0.33402328472204  0.33549663452332  2946  0.38264762922417  0.38442091603673  2947  0.39239483427316
0.39436743817451
 2951  0.19898444992189  0.19981584284266  2954  0.25030931920570  0.25145567502992  2955  0.25664546503135
0.25796184921040
 2959  0.05902480884369  0.05921492196044  2962  0.11179513825895  0.11229876077033  2963  0.11457482294849
0.11520438339244
 2967  0.74890286093071  0.75256098042718  2970  0.72141555782597  0.72489771365710  2973  0.69687450801903
0.70010671028975
 2977  0.67002705843163  0.67325581377473  2980  0.63610623849959  0.63912395204085  2983  0.61449419702198
0.61726635234273
 2987  0.57463213857907  0.57737285457920  2990  0.53511436483774  0.53761283759456  2993  0.51696126796860
0.51922684821146
 2997  0.46507037664575  0.46727305951329  3000  0.42092972803036  0.42286391085199  3003  0.40667939714304
0.40840225586211
 3007  0.34404172811010  0.34566745093179  3010  0.29636530729526  0.29770267121937  3013  0.28636404858138
0.28752144456405
 3017  0.21452565348082  0.21555036250386  3020  0.16448587377678  0.16521100380282  3023  0.15897379832128
0.15956090107186
 3027  0.07970654884309  0.08012570878043  3030  0.02854379079365  0.02865129316875  3033  0.02762731078678
0.02767143864301
```

3037 0.61958470100313 0.62238474456056 3040 0.65043845998146 0.65343795880678 3041 0.67511955118286 0.67837420599256
3045 0.52702945583474 0.52933760084375 3048 0.56335384630963 0.56588839277437 3049 0.58470236662992 0.58748360721149
3053 0.42147552840622 0.42325640862338 3056 0.46237556455209 0.46440477643596 3057 0.47986830092227 0.48212721227457
3061 0.30552189633557 0.30675323891523 3064 0.34999085208029 0.35148597323061 3065 0.36320057824556 0.36489924528297
3069 0.18202000904966 0.18269678640934 3072 0.22896485030310 0.22991241884926 3073 0.23757117298980 0.23868624784134
3077 0.05399648258722 0.05414173268619 3080 0.10227013497208 0.10267765768026 3081 0.10606824163226 0.10659600282361
3085 0.60406665127214 0.60688687027071 3088 0.57349132716021 0.57611969630745 3091 0.54637833908859 0.54874086390296
3095 0.51809295350119 0.52045596566075 3098 0.48246949206155 0.48461545485338 3101 0.45968394870908 0.46158516200956
3105 0.41933777595133 0.42120970788190 3108 0.37954341072045 0.38117837255433 3111 0.36164409253974 0.36306370151415
3115 0.31023210029099 0.31159186918012 3118 0.26724560879435 0.26835541366451 3121 0.25467015070025 0.25560240774812
3125 0.19345827715838 0.19430160454500 3128 0.14833540079574 0.14892465386972 3131 0.14138944121077 0.14184733440758
3135 0.07188426447786 0.07222698955594 3138 0.02574283049908 0.02582687484406 3141 0.02457296315122 0.02459950892961
3145 0.46477295496442 0.46670355422739 3148 0.49680165733630 0.49892946161930 3149 0.52318285444849 0.52557435787858
3153 0.37171121504260 0.37317445414644 3156 0.40777744026914 0.40945392773414 3157 0.42940526813645 0.43132046051419
3161 0.26946697392493 0.27045656050940 3164 0.30868393771694 0.30989627924849 3165 0.32502817933085 0.32644602194140
3169 0.16055156526690 0.16107912875876 3172 0.20195648366331 0.20270795588095 3173 0.21261782751313 0.21353339889618
3177 0.04763128069916 0.04773539426710 3180 0.09021344349983 0.09052828989049 3181 0.09493424757558 0.09536287489342
3185 0.44876583229496 0.45072372357340 3188 0.41791314814037 0.41968523126722 3191 0.39105707308292 0.39257771720566
3195 0.36324795903845 0.36477477532757 3198 0.32878025044228 0.33010695766590 3201 0.30767384491554 0.30878531389558
3205 0.26875447009083 0.26984386149511 3208 0.23151858612986 0.23240035520468 3211 0.21668001201626 0.21738959135769
3215 0.16760580881077 0.16826849623220 3218 0.12851511866418 0.12897128470575 3221 0.12030752830845 0.12064101560588
3225 0.06228308973104 0.06254980214093 3228 0.02230478035931 0.02236651314621 3231 0.02091058695089 0.02092185766528
3235 0.31276401907905 0.31390370611341 3238 0.34310798749630 0.34442096443183 3239 0.36833766406521 0.36989316754540
3243 0.22675059951287 0.22750034398998 3246 0.25974796627684 0.26067591038448 3247 0.27882332914463 0.27995461412740
3251 0.13511187768744 0.13549516836715 3254 0.16995368844841 0.17051215028973 3255 0.18240648507284 0.18312264899348
3259 0.04008753369117 0.04015365201645 3262 0.07592477194276 0.07614981515745 3263 0.08145137322014 0.08178159649211
3267 0.29818361339199 0.29935787718661 3270 0.26989200073845 0.27090721391756 3273 0.24609781036530 0.24690360704919
3277 0.22063233319767 0.22145140237965 3280 0.19006628979064 0.19072282870714 3283 0.17332967030747 0.17382392175333
3287 0.13760712216228 0.13809205909104 3290 0.10551510385168 0.10584221448118 3293 0.09624782255263 0.09646411461538
3297 0.05114059633334 0.05133243100633 3300 0.01831472885771 0.01835541366451 3303 0.01673051109242 0.01672904082956
3307 0.17842417210075 0.17894231397116 3310 0.20438735908536 0.20503683547307 3311 0.22572487192832 0.22656979459748
3315 0.10632724956390 0.10657486724767 3318 0.13374427299829 0.13411776966098 3319 0.14768246561211 0.14820281172333

```
 3323   0.03155099819062   0.03158319359083  3326   0.05975594500328   0.05989627924849  3327   0.06595254776156   0.06618658376761
 3331   0.16704823403916   0.16760607478254  3334   0.14390717803154   0.14434907319406  3337   0.12568254611556   0.12597813030688
 3341   0.10419905860160   0.10451533715378  3344   0.07990020783705   0.08010695766590  3347   0.06979991882288   0.06991194697701
 3351   0.03873037907639   0.03885108505778  3354   0.01387064311684   0.01389234298027  3357   0.01213510692484   0.01212429948811
 3361   0.07490237908617   0.07503033919484  3364   0.09421517806205   0.09442096443183  3365   0.10929788916778   0.10963372937162
 3369   0.02223044301073   0.02223505212040  3372   0.04210250228809   0.04216789816007  3373   0.04881785099964   0.04896183769007
 3377   0.06819774708013   0.06836510074929  3380   0.05229562904974   0.05239920169315  3383   0.04160741578786   0.04163831517961
 3387   0.02535497731832   0.02541309645575  3390   0.00908082127859   0.00908719670580  3393   0.00723589149981   0.00722101765502
 3397   0.01235146661909   0.01233940987285  3400   0.02339232870776   0.02340120347172  3401   0.03046435728973   0.03053148867358
 3405   0.01133575997113   0.01134935307775  3408   0.00406008048682   0.00405829348976  3411   0.00214842152390   0.00213993040208
 3415   0.99221932836816   0.99732465707428  3418   0.98309167568237   0.98811523439862  3421   0.98668693308122   0.99174663408490
 3425   0.95295082955850   0.95782627229675  3428   0.92644687608543   0.93109948672708  3431   0.93865886796193   0.94340572588151
 3435   0.86910723316961   0.87348645716948  3438   0.82782214597077   0.83185851269382  3441   0.84727935655760   0.85145585438060
 3445   0.74889041785613   0.75256098042718  3448   0.69686027711868   0.70010671028975  3451   0.72148262747026   0.72489771365710
 3455   0.60405235055409   0.60688687027071  3458   0.54636304990749   0.54874086390296  3461   0.57356564439101   0.57611969630745
 3465   0.44875109079872   0.45072372357340  3468   0.39104218247676   0.39257771720566  3471   0.41798740668539   0.41968523126722
 3475   0.29816988491876   0.29935787718661  3478   0.24608472812864   0.24690360704919  3481   0.26995888050896   0.27090721391756
 3485   0.16703685993886   0.16760607478254  3488   0.12567248600502   0.12597813030688  3491   0.14396003773267   0.14434907319406
 3495   0.06818979867454   0.06836510074929  3498   0.04160121633675   0.04163831517961  3501   0.05232905300201   0.05239920169315
 3505   0.01133163644087   0.01134935307775  3508   0.00214472612857   0.00213993040208  3511   0.00406700992112   0.00405829348976
 3515   0.95688601997811   0.96174642343239  3518   0.97492942164403   0.97993850223311  3519   0.96948764090913   0.97445771976688
 3523   0.87828891312704   0.88264418273435  3526   0.91260974042798   0.91724757291113  3527   0.89889497049777   0.90343795880678
 3531   0.76207062915696   0.76570811750848  3534   0.80931376299857   0.81333559813481  3535   0.78893814519631   0.79282208776554
 3539   0.61959224685032   0.62238474456056  3542   0.67514031687660   0.67837420599256  3543   0.65036679802795   0.65343795880678
 3547   0.46478074872185   0.46670355422739  3550   0.52320471360022   0.52557435787858  3551   0.49672627706633   0.49892946161930
 3555   0.31277055203343   0.31390370611341  3558   0.36835911472255   0.36989316754540  3559   0.34303623920935   0.34442096443183
 3563   0.17842801481912   0.17894231397116  3566   0.22574441922746   0.22656979459748  3567   0.20432621724276   0.20503683547307
 3571   0.07490225452255   0.07503033919484  3574   0.10931414416260   0.10963372937162  3575   0.09417052714843   0.09442096443183
 3579   0.01234789532233   0.01233940987285  3582   0.03047569521821   0.03053148867358  3583   0.02336843703049   0.02340120347172
 3587   0.96981613431525   0.97482011001528  3590   0.95177066914879   0.95662803121455  3593   0.95524069436949   0.96014371300095
 3597   0.90749341222041   0.91212918069330  3600   0.87317024487622   0.87752579051652  3603   0.88467019673043   0.88912395204085
 3607   0.80419506691251   0.80821720591697  3610   0.75695038198089   0.76058972529065  3613   0.77473380040201   0.77850808099961
```

```
 3617 0.67002073692227 0.67325581377473 3620 0.61447187313654 0.61726635234273 3623 0.63617773733711
0.63912395204085
 3627 0.51808540024645 0.52045596566075 3630 0.45966120714697 0.46158516200956 3633 0.48254455690289
0.48461545485338
 3637 0.36324058026463 0.36477477532757 3640 0.30765208186242 0.30878531389558 3643 0.32885142861520
0.33010695766590
 3647 0.22062646919018 0.22145140237965 3650 0.17331012099883 0.17382392175333 3653 0.19012643819186
0.19072282870714
 3657 0.10419578434016 0.10451533715378 3660 0.06978342798753 0.06991194697701 3663 0.07994307767962
0.08010695766590
 3667 0.02535399789740 0.02541309645575 3670 0.00722397039338 0.00722101765502 3673 0.00909487181176
0.00908719670580
 3677 0.90709060625559 0.91169622366786 3680 0.93360532957939 0.93842300923754 3681 0.92840445600196
0.93317442234844
 3685 0.80847389830887 0.81245524963461 3688 0.84976899668931 0.85408319411027 3689 0.83700792236443
0.84122455084753
 3693 0.67752055207338 0.68070344723054 3696 0.72956095339486 0.73315771736796 3697 0.71120044537980
0.71466641012403
 3701 0.52703006568930 0.52933760084375 3704 0.58473089188547 0.58748360721149 3705 0.56327929842806
0.56588839277437
 3709 0.37171121018362 0.37317445414644 3712 0.42943405561522 0.43132046051419 3713 0.40770275201807
0.40945392773414
 3717 0.22674843106898 0.22750034398998 3720 0.27885080902175 0.27995461412740 3721 0.25968032792515
0.26067591038448
 3725 0.10632145718634 0.10657486724767 3728 0.14770709495547 0.14820281172333 3729 0.13369008889443
0.13411776966098
 3733 0.02222288841314 0.02223505212040 3736 0.04883772993655 0.04896183769007 3737 0.04206696371893
0.04216789816007
 3741 0.92350561388795 0.92831225660525 3744 0.89698999047115 0.90158547103557 3747 0.90024871982412
0.90489886716863
 3751 0.83966860014066 0.84397244147798 3754 0.79837212224349 0.80234449700232 3757 0.80887758144332
0.81294899566772
 3761 0.71946055628078 0.72304696473567 3764 0.66741970023035 0.67059269459825 3767 0.68309302099243
0.68639085494422
 3771 0.57463204298217 0.57737285457920 3774 0.51693133611601 0.51922684821146 3777 0.53518847540156
0.53761283759456
 3781 0.41933706489562 0.42120970788190 3784 0.36161434812994 0.36306370151415 3787 0.37961727570893
0.38117837255433
 3791 0.26875448581558 0.26984386149511 3794 0.21665173262963 0.21738959135769 3797 0.23158477083223
0.23240035520468
 3801 0.13760901800209 0.13809205909104 3804 0.09622199040555 0.09646411461538 3807 0.10556676085517
0.10584221448118
 3811 0.03873368050789 0.03885108505778 3814 0.01211612502477 0.01212429948811 3817 0.01389195952650
0.01389234298027
 3821 0.83493877829309 0.83919703713035 3824 0.86927041784236 0.87380042730712 3825 0.86443853304870
0.86891327362344
 3829 0.71873338253417 0.72226097190448 3832 0.76598515353555 0.76988845253080 3833 0.75449102199636
0.75829740258220
 3837 0.57626976623476 0.57893759895656 3840 0.63182629134691 0.63492706038855 3841 0.61593112926234
0.61891327362344
 3845 0.42146910381680 0.42325640862338 3848 0.47990381551652 0.48212721227457 3849 0.46229988965442
0.46440477643596
 3853 0.26945903393227 0.27045656050940 3856 0.32506307186800 0.32644602194140 3857 0.30861165280230
0.30989627924849
 3861 0.13510086236184 0.13549516836715 3864 0.18243913333154 0.18312264899348 3865 0.16989162116219
0.17051215028973
 3869 0.03154007706325 0.03158319359083 3872 0.06598087844899 0.06618658376761 3873 0.05971024843177
0.05989627924849
 3877 0.85443390908940 0.85894627454584 3880 0.82010269967238 0.82434288436907 3883 0.82307004644734
0.82737240803946
 3887 0.75115089791520 0.75503429976952 3890 0.70389899332529 0.70740681914320 3893 0.71315235169825
0.71675653699822
 3897 0.61699498332999 0.62007290762727 3900 0.56143920694359 0.56408344619528 3903 0.57461875494642
0.57737240803946
```

3907 0.46507644337861 0.46727305951329 3910 0.40664251314570 0.40840225586211 3913 0.42100455448056 0.42286391085199
3917 0.31023813780239 0.31159186918012 3920 0.25463400366018 0.25560240774813 3923 0.26731633792192 0.26835541366451
3927 0.16761304056642 0.16826849623220 3930 0.12027323290142 0.12064101560588 3933 0.12857447563835 0.12897128470575
3937 0.05114835612349 0.05133243100633 3940 0.01670510603268 0.01672904082956 3943 0.01834288866937 0.01835541366451
3947 0.74221222887518 0.74603403440339 3950 0.78351368950118 0.78766197887905 3951 0.77916941286540 0.78325659634400
3955 0.61128275274184 0.61428223199932 3958 0.66332681466618 0.66673650213674 3959 0.65338122775616 0.65669845562050
3963 0.46081569595128 0.46291638561253 3966 0.51852148999830 0.52106239198028 3967 0.50548034195991 0.50792043827084
3971 0.30550849625027 0.30675323891523 3974 0.36324212533979 0.36489924528297 3975 0.34991581970211 0.35148597323061
3979 0.16053575324700 0.16107912875876 3982 0.21265783220071 0.21353339889618 3983 0.20188823563127 0.20270795588095
3987 0.04007364170248 0.04015365201645 3990 0.08148761720184 0.08178159649211 3991 0.07587026072438 0.07614981515745
3995 0.76430700728982 0.76843018452787 3998 0.72300758630562 0.72680224005222 4001 0.72561080789225 0.72947329433276
4005 0.64412610147557 0.64750470778557 4008 0.59208346724510 0.59505043764815 4011 0.59985855752404 0.60291515360926
4015 0.49932725748757 0.50183059762910 4018 0.44162341386815 0.44368459126136 4021 0.45198821027772 0.45413713625960
4025 0.34405372920198 0.34566745093179 4028 0.28632102582855 0.28752144456405 4031 0.29643892467392 0.29770267121937
4035 0.19347078477984 0.19430160454500 4038 0.14134764976411 0.14184733440758 4041 0.14840109424652 0.14892465386972
4045 0.06229522069314 0.06254980214093 4048 0.02087943805906 0.02092185766528 4051 0.02233912898534 0.02236651314621
4055 0.63120120598912 0.63450119780442 4058 0.67845320050523 0.68212867843074 4059 0.67470290227880 0.67831354218297
4063 0.48877512614259 0.49117782485650 4066 0.54432930660981 0.54716728628849 4067 0.53617487125099 0.53892941322420
4071 0.33400482759522 0.33549663452332 4074 0.39244207706030 0.39436743817451 4075 0.38257177333654 0.38442091603673
4079 0.18199987725370 0.18269678640934 4082 0.23761763359201 0.23868624784134 4083 0.22889220387811 0.22991241884926
4087 0.04761481562196 0.04773539426710 4090 0.09497759082990 0.09536287489342 4091 0.09015156030067 0.09052828989049
4095 0.65535110836850 0.65899279309326 4098 0.60810318001220 0.61136531246694 4101 0.61027871692874 0.61361212713653
4105 0.52123805617800 0.52403140095101 4108 0.46568725833644 0.46804193951901 4111 0.47179555884666 0.47422799817777
4115 0.36936137889655 0.37123155283703 4118 0.31092698643296 0.31236074918584 4121 0.31822763094392 0.31971950099029
4125 0.21454319960882 0.21555036250386 4128 0.15892560947633 0.15956090107186 4131 0.16455634385352 0.16521100380282
4135 0.07190051998691 0.07222698955594 4138 0.02453686189349 0.02459950892961 4141 0.02578254126645 0.02582687484406
4145 0.50464916622740 0.50734483592675 4148 0.55668502293624 0.55979910606418 4149 0.55362001801467 0.55666815741981
4153 0.35423624013678 0.35597898953996 4156 0.41193301464990 0.41412499590771 4157 0.40576645450523 0.40789014007015
4161 0.19896054808087 0.19981584284266 4164 0.25669729099407 0.25796184921040 4165 0.25023412624214 0.25145567502992
4169 0.05397787937247 0.05414173268619 4172 0.10611766127782 0.10659600282361 4173 0.10220245772094 0.10267765768026
4177 0.53025911432173 0.53332881202100 4180 0.47823006785875 0.48087454188358 4183 0.47992535380542 0.48264179290830
4187 0.38552474962003 0.38765470186453 4190 0.32783373598756 0.32950869549679 4193 0.33213053276398 0.33386377555864

4197 0.23030347576364 0.23149155516722 4200 0.17256939115083 0.17334554879948 4203 0.17663522783492 0.17742931051841

4207 0.07972655879042 0.08012570878043 4210 0.02758715940587 0.02767143864301 4213 0.02858789690437 0.02865129316875

4217 0.36568733634698 0.36769596040760 4220 0.42122130501399 0.42368542183959 4221 0.41891504723183 0.42131575514528

4225 0.21098889270671 0.21201477007442 4228 0.26941183801575 0.27088557372561 4229 0.26537711132099 0.26680725795781

4233 0.05900454397587 0.05921492196044 4236 0.11462913222153 0.11520438339244 4237 0.11172336699700 0.11229876077033

4241 0.39212679841850 0.39453250557993 4244 0.33660385233236 0.33854304414794 4247 0.33777936927107 0.33978721594239

4251 0.24034514765921 0.24173265746595 4254 0.18193000997823 0.18286185381476 4257 0.18432173608363 0.18527871875492

4261 0.08557637636181 0.08605146713278 4264 0.02995416385750 0.03006200570078 4267 0.03068451029535 0.03077022156744

4271 0.21776198616361 0.21899319003028 4274 0.27542073940151 0.27713919639802 4275 0.27392515737278 0.27558916071225

4279 0.06256609634920 0.06283004333297 4282 0.12028984428073 0.12097604970072 4283 0.11847141871392 0.11915469567202

4287 0.24438288932734 0.24602149947317 4290 0.18674297133616 0.18787549310543 4293 0.18737670210339 0.18856595003087

4297 0.08929292576658 0.08985835277587 4300 0.03157633687452 0.03171234640812 4303 0.03201551627851 0.03213148499063

4307 0.06455681222077 0.06489808051768 4310 0.12291907904852 0.12376888416887 4311 0.12225441238540 0.12307664651450

4315 0.09072684297266 0.09145262755084 4318 0.03239884060445 0.03258182389965 4321 0.03250936103486 0.03270156461507

4325 0.99222025782418 0.99732465707428 4328 0.98667620482132 0.99174663408490 4331 0.98309568598507 0.98811523439862

4335 0.96981113299080 0.97482011001528 4338 0.95521985947287 0.96014371300095 4341 0.95178445547330 0.95662803121455

4345 0.92349490938073 0.92831225660525 4348 0.90021827612325 0.90489886716863 4351 0.89701313985611 0.90158547103557

4355 0.85441776063313 0.85894627454584 4358 0.82303074280474 0.82737240803946 4361 0.82013464161420 0.82434288436907

4365 0.76428581390754 0.76843018452787 4368 0.72556361416202 0.72947329433276 4371 0.72304754023011 0.72680224005222

4375 0.65532539474762 0.65899279309326 4378 0.61022479835646 0.61361212713653 4381 0.60815017109954 0.61136531246694

4385 0.53022951717416 0.53332881202100 4388 0.47986604166880 0.48264179290830 4391 0.47828294938190 0.48087454188358

4395 0.39209404968082 0.39453250557993 4398 0.33771612581820 0.33978721594239 4401 0.33666132942477 0.33854304414794

4405 0.24434779669722 0.24602149947317 4408 0.18731108619540 0.18856595003087 4411 0.18680357908014 0.18787549310543

4415 0.09069302279800 0.09145262755084 4418 0.03246652968042 0.03270156461507 4421 0.03244574318888 0.03258182389965

4425 0.95688725786318 0.96174642343239 4428 0.96951084671645 0.97445771976688 4429 0.97491837166128 0.97993850223311

4433 0.90709694147439 0.91169622366786 4436 0.92843854627147 0.93317442234844 4437 0.93358441883598 0.93842300923754

4441 0.83495005301818 0.83919703713035 4444 0.86448265974885 0.86891327362344 4445 0.86924014679603 0.87380042730712

4449 0.74222817257939 0.74603403440339 4452 0.77922249405825 0.78325659634400 4453 0.78347480549542 0.78766197887905

4457 0.63122143767916 0.63450119780442 4460 0.67476364108906 0.67831354218297 4461 0.67840666624681 0.68212867843074

4465 0.50467320182060 0.50734483592675 4468 0.55368693225993 0.55666815741981 4469 0.55663199120827 0.55979910606418

4473 0.36571459674770 0.36769596040760 4476 0.41898650333197 0.42131575514528 4477 0.42116308866011 0.42368542183959

4481 0.21779176758241 0.21899319003028 4484 0.27399939874192 0.27558916071225 4485 0.27535877454551 0.27713919639802

4489 0.06457901048541 0.06489808051768 4492 0.12232849455682 0.12307664651450 4493 0.12285561409930 0.12376888416887
4497 0.95295614791021 0.95782627229675 4500 0.93862469698220 0.94340572588151 4503 0.92645515251913 0.93109948672708
4507 0.90749357193984 0.91212918069330 4510 0.88462597450646 0.88912395204085 4513 0.87318773083239 0.87752579051652
4517 0.83966359898850 0.84397244147798 4520 0.80882440132881 0.81294899566772 4523 0.79839839498354 0.80234449700232
4527 0.75114086933072 0.75503429976952 4530 0.71309153634519 0.71675653699822 4533 0.70393342397166 0.70740681914320
4537 0.64411130955468 0.64750470778557 4540 0.59979162353683 0.60291515360926 4543 0.59212523361581 0.59505043764815
4547 0.52121888683708 0.52403140095101 4550 0.47172417703673 0.47422799817777 4553 0.46573536283936 0.46804193951901
4557 0.38550169779038 0.38765470186453 4560 0.33205648248968 0.33386377555864 4563 0.32788702666643 0.32950869549679
4567 0.24031880460600 0.24173265746595 4570 0.18424689356004 0.18527871875492 4573 0.18198718437680 0.18286185381476
4577 0.08926686261120 0.08985835277587 4580 0.03196592229619 0.03213148499063 4583 0.03162151840217 0.03171234640812
4587 0.87828607331940 0.88264418273435 4590 0.89893924942842 0.90343795880678 4591 0.91259504084073 0.91724757291113
4595 0.80847657255829 0.81245524963461 4598 0.83706121893119 0.84122455084753 4599 0.84974527502342 0.85408319411027
4603 0.71874152387089 0.72226097190448 4606 0.75455203889613 0.75829740258220 4607 0.76595300458067 0.76988845253080
4611 0.61129618845500 0.61428223199932 4614 0.65344848720845 0.65669845562050 4615 0.66328704900120 0.66673650213674
4619 0.48879355991057 0.49117782485650 4622 0.53624674936786 0.53892941322420 4623 0.54428292828389 0.54716728628849
4627 0.35425925640499 0.35597898953996 4630 0.40584121873460 0.40789014007015 4631 0.41188119364619 0.41412499590771
4635 0.21101593878857 0.21201477007442 4638 0.26545295758450 0.26680725795781 4639 0.26935587964226 0.27088557372561
4643 0.06258750529141 0.06283004333297 4646 0.11854548479397 0.11915469567202 4647 0.12023193402277 0.12097604970072
4651 0.86911658023817 0.87348645716948 4654 0.84722612084225 0.85145585438060 4657 0.82783396111788 0.83185851269382
4661 0.80419884533596 0.80821720591697 4664 0.77467293430427 0.77850808099961 4667 0.75697090071762 0.76058972529065
4671 0.71945869563900 0.72304696473567 4674 0.68302605006922 0.68639085494422 4677 0.67744844969248 0.67059269459825
4681 0.61698756287884 0.62007290762727 4684 0.57454736535096 0.57737240803946 4687 0.56147552358710 0.56408344619528
4691 0.49931450255424 0.50183059762910 4694 0.45191420534221 0.45413713625960 4697 0.44166645822801 0.44368459126136
4701 0.36934365306399 0.37123155283703 4704 0.31815288443605 0.31971950099029 4707 0.31097576160221 0.31236074918584
4711 0.23028127722975 0.23149155516722 4714 0.17656167696692 0.17742931051841 4717 0.17262274660091 0.17334554879948
4721 0.08555308192753 0.08605146713278 4724 0.03063708611254 0.03077022156744 4727 0.02999736787712 0.03006200570078
4731 0.76206479419881 0.76570811750848 4734 0.78899909235613 0.79282208776554 4735 0.80929546859690 0.81333559813481
4739 0.67752087369728 0.68070344723054 4742 0.71126763094265 0.71466641012403 4743 0.72953418496733 0.73315771736796
4747 0.57627628239052 0.57893759895656 4750 0.61600293992089 0.61891327362344 4751 0.63179174274859 0.63492706038855
4755 0.46082830401802 0.46291638561253 4758 0.50555506304757 0.50792043827084 4759 0.51848005724973 0.52106239198028
4763 0.33402328472204 0.33549663452332 4766 0.38264762922417 0.38442091603673 4767 0.39239483427317 0.39436743817451
4771 0.19898444992190 0.19981584284266 4774 0.25030931920571 0.25145567502992 4775 0.25664546503136 0.25796184921040

4779 0.05902480884370 0.05921492196044 4782 0.11179513825895 0.11229876077033 4783 0.11457482294850
0.11520438339244
4787 0.74890286093072 0.75256098042718 4790 0.72141555782598 0.72489771365710 4793 0.69687450801904
0.70010671028975
4797 0.67002705843164 0.67325581377473 4800 0.63610623849959 0.63912395204085 4803 0.61449419702198
0.61726635234273
4807 0.57463213857907 0.57737285457920 4810 0.53511436483775 0.53761283759456 4813 0.51696126796860
0.51922684821146
4817 0.46507037664576 0.46727305951329 4820 0.42092972803037 0.42286391085199 4823 0.40667939714304
0.40840225586211
4827 0.34404172811010 0.34566745093179 4830 0.29636530729526 0.29770267121937 4833 0.28636404858138
0.28752144456405
4837 0.21452565348082 0.21555036250386 4840 0.16448587377678 0.16521100380282 4843 0.15897379832128
0.15956090107186
4847 0.07970654884309 0.08012570878043 4850 0.02854379079365 0.02865129316875 4853 0.02762731078678
0.02767143864301
4857 0.61958470100313 0.62238474456056 4860 0.65043845998147 0.65343795880678 4861 0.67511955118287
0.67837420599256
4865 0.52702945583474 0.52933760084375 4868 0.56335384630963 0.56588839277437 4869 0.58470236662992
0.58748360721149
4873 0.42147552840622 0.42325640862338 4876 0.46237556455209 0.46440477643596 4877 0.47986830092227
0.48212721227457
4881 0.30552189633557 0.30675323891523 4884 0.34999085208029 0.35148597323061 4885 0.36320057824557
0.36489924528297
4889 0.18202000904966 0.18269678640934 4892 0.22896485030310 0.22991241884926 4893 0.23757117298981
0.23868624784134
4897 0.05399648258722 0.05414173268619 4900 0.10227013497208 0.10267765768026 4901 0.10606824163226
0.10659600282361
4905 0.60406665127214 0.60688687027071 4908 0.57349132716021 0.57611969630745 4911 0.54637833908859
0.54874086390296
4915 0.51809295350119 0.52045596566075 4918 0.48246949206155 0.48461545485338 4921 0.45968394870909
0.46158516200956
4925 0.41933777595134 0.42120970788190 4928 0.37954341072046 0.38117837255433 4931 0.36164409253974
0.36306370151415
4935 0.31023210029099 0.31159186918012 4938 0.26724560879436 0.26835541366451 4941 0.25467015070025
0.25560240774813
4945 0.19345827715838 0.19430160454500 4948 0.14833540079574 0.14892465386972 4951 0.14138944121077
0.14184733440758
4955 0.07188426447786 0.07222698955594 4958 0.02574283049908 0.02582687484406 4961 0.02457296315122
0.02459950892961
4965 0.46477295496443 0.46670355422739 4968 0.49680165733631 0.49892946161930 4969 0.52318285444850
0.52557435787858
4973 0.37171121504260 0.37317445414644 4976 0.40777744026914 0.40945392773414 4977 0.42940526813646
0.43132046051419
4981 0.26946697392493 0.27045656050940 4984 0.30868393771694 0.30989627924849 4985 0.32502817933085
0.32644602194140
4989 0.16055156526690 0.16107912875876 4992 0.20195648366332 0.20270795588095 4993 0.21261782751313
0.21353339889618
4997 0.04763128069916 0.04773539426710 5000 0.09021344349983 0.09052828989049 5001 0.09493424757558
0.09536287489342
5005 0.44876583229496 0.45072372357340 5008 0.41791314814037 0.41968523126722 5011 0.39105707308293
0.39257771720566
5015 0.36324795903846 0.36477477532757 5018 0.32878025044229 0.33010695766590 5021 0.30767384491554
0.30878531389558
5025 0.26875447009083 0.26984386149511 5028 0.23151858612986 0.23240035520468 5031 0.21668001201626
0.21738959135769
5035 0.16760580881077 0.16826849623220 5038 0.12851511866418 0.12897128470575 5041 0.12030752830845
0.12064101560588
5045 0.06228308973104 0.06254980214093 5048 0.02230478035931 0.02236651314621 5051 0.02091058695089
0.02092185766528
5055 0.31276401907906 0.31390370611341 5058 0.34310798749630 0.34442096443183 5059 0.36833766406522
0.36989316754540
5063 0.22675059951287 0.22750034398998 5066 0.25974796627684 0.26067591038448 5067 0.27882332914463
0.27995461412740

```
 5071   0.13511187768744   0.13549516836715  5074   0.16995368844841   0.17051215028973  5075   0.18240648507284
0.18312264899348
 5079   0.04008753369117   0.04015365201645  5082   0.07592477194276   0.07614981515745  5083   0.08145137322014
0.08178159649211
 5087   0.29818361339199   0.29935787718661  5090   0.26989200073845   0.27090721391756  5093   0.24609781036530
0.24690360704919
 5097   0.22063233319767   0.22145140237965  5100   0.19006628979064   0.19072282870714  5103   0.17332967030747
0.17382392175333
 5107   0.13760712216228   0.13809205909104  5110   0.10551510385168   0.10584221448118  5113   0.09624782255263
0.09646411461538
 5117   0.05114059633334   0.05133243100633  5120   0.01831472885771   0.01835541366451  5123   0.01673051109242
0.01672904082956
 5127   0.17842417210075   0.17894231397116  5130   0.20438735908536   0.20503683547307  5131   0.22572487192832
0.22656979459748
 5135   0.10632724956390   0.10657486724767  5138   0.13374427299829   0.13411776966098  5139   0.14768246561211
0.14820281172333
 5143   0.03155099819062   0.03158319359083  5146   0.05975594500328   0.05989627924849  5147   0.06595254776156
0.06618658376761
 5151   0.16704823403916   0.16760607478254  5154   0.14390717803154   0.14434907319406  5157   0.12568254611556
0.12597813030688
 5161   0.10419905860160   0.10451533715378  5164   0.07990020783705   0.08010695766590  5167   0.06979991882288
0.06991194697701
 5171   0.03873037907639   0.03885108505778  5174   0.01387064311684   0.01389234298027  5177   0.01213510692484
0.01212429948811
 5181   0.07490237908617   0.07503033919484  5184   0.09421517806205   0.09442096443183  5185   0.10929788916778
0.10963372937162
 5189   0.02223044301073   0.02223505212040  5192   0.04210250228809   0.04216789816007  5193   0.04881785099964
0.04896183769007
 5197   0.06819774708013   0.06836510074929  5200   0.05229562904974   0.05239920169315  5203   0.04160741578786
0.04163831517961
 5207   0.02535497731832   0.02541309645575  5210   0.00908082127859   0.00908719670580  5213   0.00723589149981
0.00722101765502
 5217   0.01235146661909   0.01233940987285  5220   0.02339232870776   0.02340120347172  5221   0.03046435728973
0.03053148867358
 5225   0.01133575997113   0.01134935307775  5228   0.00406008048682   0.00405829348976  5231   0.00214842152390
0.00213993040208
 5235   0.99221932836816   0.99732465707428  5238   0.98309167568237   0.98811523439862  5241   0.98668693308121
0.99174663408490
 5245   0.95295082955850   0.95782627229675  5248   0.92644687608544   0.93109948672708  5251   0.93865886796192
0.94340572588151
 5255   0.86910723316962   0.87348645716948  5258   0.82782214597079   0.83185851269382  5261   0.84727935655761
0.85145585438060
 5265   0.74889041785614   0.75256098042718  5268   0.69686027711869   0.70010671028975  5271   0.72148262747027
0.72489771365710
 5275   0.60405235055410   0.60688687027071  5278   0.54636304990749   0.54874086390296  5281   0.57356564439102
0.57611969630745
 5285   0.44875109079872   0.45072372357340  5288   0.39104218247676   0.39257771720566  5291   0.41798740668539
0.41968523126722
 5295   0.29816988491876   0.29935787718661  5298   0.24608472812864   0.24690360704919  5301   0.26995888050896
0.27090721391756
 5305   0.16703685993886   0.16760607478254  5308   0.12567248600502   0.12597813030688  5311   0.14396003773267
0.14434907319406
 5315   0.06818979867454   0.06836510074929  5318   0.04160121633675   0.04163831517961  5321   0.05232905300201
0.05239920169315
 5325   0.01133163644087   0.01134935307775  5328   0.00214472612857   0.00213993040208  5331   0.00406700992112
0.00405829348976
 5335   0.95688601997811   0.96174642343239  5338   0.97492942164403   0.97993850223311  5339   0.96948764090914
0.97445771976688
 5343   0.87828891312704   0.88264418273435  5346   0.91260974042798   0.91724757291113  5347   0.89889497049777
0.90343795880678
 5351   0.76207062915697   0.76570811750848  5354   0.80931376299859   0.81333559813481  5355   0.78893814519632
0.79282208776554
 5359   0.61959224685033   0.62238474456056  5362   0.67514031687661   0.67837420599256  5363   0.65036679802796
0.65343795880678
```

5367	0.46478074872185	0.46670355422739	5370	0.52320471360022	0.52557435787858	5371	0.49672627706634	0.49892946161930
5375	0.31277055203343	0.31390370611341	5378	0.36835911472255	0.36989316754540	5379	0.34303623920935	0.34442096443183
5383	0.17842801481912	0.17894231397116	5386	0.22574441922746	0.22656979459748	5387	0.20432621724276	0.20503683547307
5391	0.07490225452255	0.07503033919484	5394	0.10931414416260	0.10963372937162	5395	0.09417052714843	0.09442096443183
5399	0.01234789532233	0.01233940987285	5402	0.03047569521821	0.03053148867358	5403	0.02336843703049	0.02340120347172
5407	0.96981613431525	0.97482011001528	5410	0.95177066914879	0.95662803121455	5413	0.95524069436949	0.96014371300095
5417	0.90749341222041	0.91212918069330	5420	0.87317024487622	0.87752579051652	5423	0.88467019673043	0.88912395204085
5427	0.80419506691252	0.80821720591697	5430	0.75695038198090	0.76058972529065	5433	0.77473380040202	0.77850808099961
5437	0.67002073692228	0.67325581377473	5440	0.61447187313655	0.61726635234273	5443	0.63617773733712	0.63912395204085
5447	0.51808540024646	0.52045596566075	5450	0.45966120714697	0.46158516200956	5453	0.48254455690289	0.48461545485338
5457	0.36324058026463	0.36477477532757	5460	0.30765208186242	0.30878531389558	5463	0.32885142861520	0.33010695766590
5467	0.22062646919018	0.22145140237965	5470	0.17331012099883	0.17382392175333	5473	0.19012643819186	0.19072282870714
5477	0.10419578434016	0.10451533715378	5480	0.06978342798753	0.06991194697701	5483	0.07994307767962	0.08010695766590
5487	0.02535399789740	0.02541309645575	5490	0.00722397039338	0.00722101765502	5493	0.00909487181176	0.00908719670580
5497	0.90709060625559	0.91169622366786	5500	0.93360532957939	0.93842300923754	5501	0.92840445600196	0.93317442234844
5505	0.80847389830887	0.81245524963461	5508	0.84976899668932	0.85408319411027	5509	0.83700792236444	0.84122455084753
5513	0.67752055207339	0.68070344723054	5516	0.72956095339487	0.73315771736796	5517	0.71120044537981	0.71466641012403
5521	0.52703006568930	0.52933760084375	5524	0.58473089188548	0.58748360721149	5525	0.56327929842807	0.56588839277437
5529	0.37171121018362	0.37317445414644	5532	0.42943405561523	0.43132046051419	5533	0.40770275201808	0.40945392773414
5537	0.22674843106898	0.22750034398998	5540	0.27885080902175	0.27995461412740	5541	0.25968032792515	0.26067591038448
5545	0.10632145718634	0.10657486724767	5548	0.14770709495547	0.14820281172333	5549	0.13369008889443	0.13411776966098
5553	0.02222288841314	0.02223505212040	5556	0.04883772993654	0.04896183769007	5557	0.04206696371893	0.04216789816007
5561	0.92350561388796	0.92831225660525	5564	0.89698999047116	0.90158547103557	5567	0.90024871982413	0.90489886716863
5571	0.83966860014066	0.84397244147798	5574	0.79837212224350	0.80234449700232	5577	0.80887758144333	0.81294899566772
5581	0.71946055628079	0.72304696473567	5584	0.66741970023035	0.67059269459825	5587	0.68309302099244	0.68639085494422
5591	0.57463204298218	0.57737285457920	5594	0.51693133611601	0.51922684821146	5597	0.53518847540157	0.53761283759456
5601	0.41933706489562	0.42120970788190	5604	0.36161434812994	0.36306370151415	5607	0.37961727570893	0.38117837255433
5611	0.26875448581558	0.26984386149511	5614	0.21665173262963	0.21738959135769	5617	0.23158477083223	0.23240035520468
5621	0.13760901800209	0.13809205909104	5624	0.09622199040555	0.09646411461538	5627	0.10556676085517	0.10584221448118
5631	0.03873368050789	0.03885108505778	5634	0.01211612502477	0.01212429948811	5637	0.01389195952650	0.01389234298027
5641	0.83493877829310	0.83919703713035	5644	0.86927041784237	0.87380042730712	5645	0.86443853304870	0.86891327362344
5649	0.71873338253418	0.72226097190448	5652	0.76598515353555	0.76988845253080	5653	0.75449102199637	0.75829740258220

```
 5657  0.57626976623477  0.57893759895656  5660  0.63182629134691  0.63492706038855  5661  0.61593112926235
0.61891327362344
 5665  0.42146910381680  0.42325640862338  5668  0.47990381551653  0.48212721227457  5669  0.46229988965442
0.46440477643596
 5673  0.26945903393227  0.27045656050940  5676  0.32506307186800  0.32644602194140  5677  0.30861165280230
0.30989627924849
 5681  0.13510086236184  0.13549516836715  5684  0.18243913333154  0.18312264899348  5685  0.16989162116219
0.17051215028973
 5689  0.03154007706325  0.03158319359083  5692  0.06598087844899  0.06618658376761  5693  0.05971024843177
0.05989627924849
 5697  0.85443390908941  0.85894627454584  5700  0.82010269967239  0.82434288436907  5703  0.82307004644734
0.82737240803946
 5707  0.75115089791521  0.75503429976952  5710  0.70389899332529  0.70740681914320  5713  0.71315235169826
0.71675653699822
 5717  0.61699498333000  0.62007290762727  5720  0.56143920694360  0.56408344619528  5723  0.57461875494643
0.57737240803946
 5727  0.46507644337861  0.46727305951329  5730  0.40664251314571  0.40840225586211  5733  0.42100455448057
0.42286391085199
 5737  0.31023813780240  0.31159186918012  5740  0.25463400366019  0.25560240774813  5743  0.26731633792192
0.26835541366451
 5747  0.16761304056642  0.16826849623220  5750  0.12027323290142  0.12064101560588  5753  0.12857447563835
0.12897128470575
 5757  0.05114835612349  0.05133243100633  5760  0.01670510603268  0.01672904082956  5763  0.01834288866937
0.01835541366451
 5767  0.74221222887519  0.74603403440339  5770  0.78351368950118  0.78766197887905  5771  0.77916941286541
0.78325659634400
 5775  0.61128275274184  0.61428223199932  5778  0.66332681466619  0.66673650213674  5779  0.65338122775616
0.65669845562050
 5783  0.46081569595128  0.46291638561253  5786  0.51852148999831  0.52106239198028  5787  0.50548034195991
0.50792043827084
 5791  0.30550849625028  0.30675323891523  5794  0.36324212533980  0.36489924528297  5795  0.34991581970211
0.35148597323061
 5799  0.16053575324700  0.16107912875876  5802  0.21265783220071  0.21353339889618  5803  0.20188823563127
0.20270795588095
 5807  0.04007364170248  0.04015365201645  5810  0.08148761720184  0.08178159649211  5811  0.07587026072438
0.07614981515745
 5815  0.76430700728982  0.76843018452787  5818  0.72300758630563  0.72680224005222  5821  0.72561080789226
0.72947329433276
 5825  0.64412610147557  0.64750470778557  5828  0.59208346724510  0.59505043764815  5831  0.59985855752405
0.60291515360926
 5835  0.49932725748757  0.50183059762910  5838  0.44162341386815  0.44368459126136  5841  0.45198821027772
0.45413713625960
 5845  0.34405372920199  0.34566745093179  5848  0.28632102582855  0.28752144456405  5851  0.29643892467392
0.29770267121937
 5855  0.19347078477985  0.19430160454500  5858  0.14134764976411  0.14184733440758  5861  0.14840109424652
0.14892465386972
 5865  0.06229522069314  0.06254980214093  5868  0.02087943805906  0.02092185766528  5871  0.02233912898534
0.02236651314621
 5875  0.63120120598912  0.63450119780442  5878  0.67845320050524  0.68212867843074  5879  0.67470290227880
0.67831354218297
 5883  0.48877512614259  0.49117782485650  5886  0.54432930660982  0.54716728628849  5887  0.53617487125100
0.53892941322420
 5891  0.33400482759522  0.33549663452332  5894  0.39244207706031  0.39436743817451  5895  0.38257177333655
0.38442091603673
 5899  0.18199987725370  0.18269678640934  5902  0.23761763359201  0.23868624784134  5903  0.22889220387811
0.22991241884926
 5907  0.04761481562196  0.04773539426710  5910  0.09497759082990  0.09536287489342  5911  0.09015156030067
0.09052828989049
 5915  0.65535110836851  0.65899279309326  5918  0.60810318001220  0.61136531246694  5921  0.61027871692875
0.61361212713653
 5925  0.52123805617800  0.52403140095101  5928  0.46568725833644  0.46804193951901  5931  0.47179555884666
0.47422799817777
 5935  0.36936137889655  0.37123155283703  5938  0.31092698643297  0.31236074918584  5941  0.31822763094392
0.31971950099029
```

5945 0.21454319960882 0.21555036250386 5948 0.15892560947633 0.15956090107186 5951 0.16455634385352 0.16521100380282

5955 0.07190051998691 0.07222698955594 5958 0.02453686189349 0.02459950892961 5961 0.02578254126645 0.02582687484406

5965 0.50464916622740 0.50734483592675 5968 0.55668502293625 0.55979910606418 5969 0.55362001801467 0.55666815741981

5973 0.35423624013678 0.35597898953996 5976 0.41193301464990 0.41412499590771 5977 0.40576645450523 0.40789014007015

5981 0.19896054808087 0.19981584284266 5984 0.25669729099407 0.25796184921040 5985 0.25023412624214 0.25145567502992

5989 0.05397787937247 0.05414173268619 5992 0.10611766127782 0.10659600282361 5993 0.10220245772094 0.10267765768026

5997 0.53025911432173 0.53332881202100 6000 0.47823006785875 0.48087454188358 6003 0.47992535380542 0.48264179290830

6007 0.38552474962003 0.38765470186453 6010 0.32783373598756 0.32950869549679 6013 0.33213053276398 0.33386377555864

6017 0.23030347576364 0.23149155516722 6020 0.17256939115083 0.17334554879948 6023 0.17663522783492 0.17742931051841

6027 0.07972655879042 0.08012570878043 6030 0.02758715940587 0.02767143864301 6033 0.02858789690437 0.02865129316875

6037 0.36568733634698 0.36769596040760 6040 0.42122130501399 0.42368542183959 6041 0.41891504723183 0.42131575514528

6045 0.21098889270670 0.21201477007442 6048 0.26941183801575 0.27088557372561 6049 0.26537711132099 0.26680725795781

6053 0.05900454397587 0.05921492196044 6056 0.11462913222153 0.11520438339244 6057 0.11172336699700 0.11229876077033

6061 0.39212679841850 0.39453250557993 6064 0.33660385233236 0.33854304414794 6067 0.33777936927107 0.33978721594239

6071 0.24034514765921 0.24173265746595 6074 0.18193000997823 0.18286185381476 6077 0.18432173608363 0.18527871875492

6081 0.08557637636181 0.08605146713278 6084 0.02995416385750 0.03006200570078 6087 0.03068451029535 0.03077022156744

6091 0.21776198616361 0.21899319003028 6094 0.27542073940151 0.27713919639802 6095 0.27392515737278 0.27558916071225

6099 0.06256609634920 0.06283004333297 6102 0.12028984428073 0.12097604970072 6103 0.11847141871392 0.11915469567202

6107 0.24438288932734 0.24602149947317 6110 0.18674297133616 0.18787549310543 6113 0.18737670210339 0.18856595003087

6117 0.08929292576658 0.08985835277587 6120 0.03157633687452 0.03171234640812 6123 0.03201551627851 0.03213148499063

6127 0.06455681222077 0.06489808051768 6130 0.12291907904852 0.12376888416887 6131 0.12225441238540 0.12307664651450

6135 0.09072684297266 0.09145262755084 6138 0.03239884060445 0.03258182389965 6141 0.03250936103486 0.03270156461507

6145 0.99222025782418 0.99732465707428 6148 0.98667620482132 0.99174663408490 6151 0.98309568598507 0.98811523439862

6155 0.96981113299080 0.97482011001528 6158 0.95521985947287 0.96014371300095 6161 0.95178445547330 0.95662803121455

6165 0.92349490938073 0.92831225660525 6168 0.90021827612326 0.90489886716863 6171 0.89701313985611 0.90158547103557

6175 0.85441776063313 0.85894627454584 6178 0.82303074280474 0.82737240803946 6181 0.82013464161420 0.82434288436907

6185 0.76428581390755 0.76843018452787 6188 0.72556361416203 0.72947329433276 6191 0.72304754023012 0.72680224005222

6195 0.65532539474763 0.65899279309326 6198 0.61022479835647 0.61361212713653 6201 0.60815017109954 0.61136531246694

6205 0.53022951717417 0.53332881202100 6208 0.47986604166880 0.48264179290830 6211 0.47828294938190 0.48087454188358

6215 0.39209404968082 0.39453250557993 6218 0.33771612581820 0.33978721594239 6221 0.33666132942477 0.33854304414794

6225 0.24434779669722 0.24602149947317 6228 0.18731108619540 0.18856595003087 6231 0.18680357908014 0.18787549310543

6235 0.09069302279800 0.09145262755084 6238 0.03246652968042 0.03270156461507 6241 0.03244574318888 0.03258182389965

6245	0.95688725786318	0.96174642343239	6248	0.96951084671645	0.97445771976688	6249	0.97491837166128	0.97993850223311
6253	0.90709694147439	0.91169622366786	6256	0.92843854627147	0.93317442234844	6257	0.93358441883598	0.93842300923754
6261	0.83495005301818	0.83919703713035	6264	0.86448265974885	0.86891327362344	6265	0.86924014679603	0.87380042730712
6269	0.74222817257940	0.74603403440339	6272	0.77922249405826	0.78325659634400	6273	0.78347480549542	0.78766197887905
6277	0.63122143767917	0.63450119780442	6280	0.67476364108907	0.67831354218297	6281	0.67840666624682	0.68212867843074
6285	0.50467320182060	0.50734483592675	6288	0.55368693225994	0.55666815741981	6289	0.55663199120827	0.55979910606418
6293	0.36571459674770	0.36769596040760	6296	0.41898650333197	0.42131575514528	6297	0.42116308866011	0.42368542183959
6301	0.21779176758241	0.21899319003028	6304	0.27399939874191	0.27558916071225	6305	0.27535877454551	0.27713919639802
6309	0.06457901048541	0.06489808051768	6312	0.12232849455682	0.12307664651450	6313	0.12285561409930	0.12376888416887
6317	0.95295614791021	0.95782627229675	6320	0.93862469698219	0.94340572588151	6323	0.92645515251912	0.93109948672708
6327	0.90749357193983	0.91212918069330	6330	0.88462597450645	0.88912395204085	6333	0.87318773083238	0.87752579051652
6337	0.83966359898849	0.84397244147798	6340	0.80882440132881	0.81294899566772	6343	0.79839839498354	0.80234449700232
6347	0.75114086933073	0.75503429976952	6350	0.71309153634520	0.71675653699822	6353	0.70393342397167	0.70740681914320
6357	0.64411130955469	0.64750470778557	6360	0.59979162353684	0.60291515360926	6363	0.59212523361582	0.59505043764815
6367	0.52121888683708	0.52403140095101	6370	0.47172417703674	0.47422799817777	6373	0.46573536283937	0.46804193951901
6377	0.38550169779038	0.38765470186453	6380	0.33205648248968	0.33386377555864	6383	0.32788702666643	0.32950869549679
6387	0.24031880460600	0.24173265746595	6390	0.18424689356004	0.18527871875492	6393	0.18198718437680	0.18286185381476
6397	0.08926686261119	0.08985835277587	6400	0.03196592229619	0.03213148499063	6403	0.03162151840217	0.03171234640812
6407	0.87828607331939	0.88264418273435	6410	0.89893924942841	0.90343795880678	6411	0.91259504084072	0.91724757291113
6415	0.80847657255829	0.81245524963461	6418	0.83706121893118	0.84122455084753	6419	0.84974527502342	0.85408319411027
6423	0.71874152387089	0.72226097190448	6426	0.75455203889614	0.75829740258220	6427	0.76595300458068	0.76988845253080
6431	0.61129618845501	0.61428223199932	6434	0.65344848720845	0.65669845562050	6435	0.66328704900120	0.66673650213674
6439	0.48879355991057	0.49117782485650	6442	0.53624674936787	0.53892941322420	6443	0.54428292828390	0.54716728628849
6447	0.35425925640499	0.35597898953996	6450	0.40584121873460	0.40789014007015	6451	0.41188119364619	0.41412499590771
6455	0.21101593878857	0.21201477007442	6458	0.26545295758450	0.26680725795781	6459	0.26935587964226	0.27088557372561
6463	0.06258750529141	0.06283004333297	6466	0.11854548479397	0.11915469567202	6467	0.12023193402277	0.12097604970072
6471	0.86911658023816	0.87348645716948	6474	0.84722612084224	0.85145585438060	6477	0.82783396111787	0.83185851269382
6481	0.80419884533595	0.80821720591697	6484	0.77467293430427	0.77850808099961	6487	0.75697090071761	0.76058972529065
6491	0.71945869563900	0.72304696473567	6494	0.68302605006922	0.68639085494422	6497	0.66744844969248	0.67059269459825
6501	0.61698756287884	0.62007290762727	6504	0.57454736535096	0.57737240803946	6507	0.56147552358710	0.56408344619528
6511	0.49931450255424	0.50183059762910	6514	0.45191420534221	0.45413713625960	6517	0.44166645822801	0.44368459126136
6521	0.36934365306399	0.37123155283703	6524	0.31815288443605	0.31971950099029	6527	0.31097576160221	0.31236074918584

```
 6531  0.23028127722975  0.23149155516722  6534  0.17656167696692  0.17742931051841  6537  0.17262274660091
0.17334554879948
 6541  0.08555308192753  0.08605146713278  6544  0.03063708611254  0.03077022156744  6547  0.02999736787712
0.03006200570078
 6551  0.76206479419881  0.76570811750848  6554  0.78899909235612  0.79282208776554  6555  0.80929546859689
0.81333559813481
 6559  0.67752087369728  0.68070344723054  6562  0.71126763094265  0.71466641012403  6563  0.72953418496733
0.73315771736796
 6567  0.57627628239052  0.57893759895656  6570  0.61600293992089  0.61891327362344  6571  0.63179174274859
0.63492706038855
 6575  0.46082830401802  0.46291638561253  6578  0.50555506304757  0.50792043827084  6579  0.51848005724973
0.52106239198028
 6583  0.33402328472204  0.33549663452332  6586  0.38264762922417  0.38442091603673  6587  0.39239483427317
0.39436743817451
 6591  0.19898444992190  0.19981584284266  6594  0.25030931920571  0.25145567502992  6595  0.25664546503136
0.25796184921040
 6599  0.05902480884370  0.05921492196044  6602  0.11179513825895  0.11229876077033  6603  0.11457482294850
0.11520438339244
 6607  0.74890286093071  0.75256098042718  6610  0.72141555782597  0.72489771365710  6613  0.69687450801903
0.70010671028975
 6617  0.67002705843164  0.67325581377473  6620  0.63610623849959  0.63912395204085  6623  0.61449419702198
0.61726635234273
 6627  0.57463213857907  0.57737285457920  6630  0.53511436483775  0.53761283759456  6633  0.51696126796860
0.51922684821146
 6637  0.46507037664576  0.46727305951329  6640  0.42092972803037  0.42286391085199  6643  0.40667939714304
0.40840225586211
 6647  0.34404172811010  0.34566745093179  6650  0.29636530729526  0.29770267121937  6653  0.28636404858138
0.28752144456405
 6657  0.21452565348082  0.21555036250386  6660  0.16448587377678  0.16521100380282  6663  0.15897379832128
0.15956090107186
 6667  0.07970654884309  0.08012570878043  6670  0.02854379079365  0.02865129316875  6673  0.02762731078678
0.02767143864301
 6677  0.61958470100313  0.62238474456056  6680  0.65043845998146  0.65343795880678  6681  0.67511955118286
0.67837420599256
 6685  0.52702945583474  0.52933760084375  6688  0.56335384630963  0.56588839277437  6689  0.58470236662992
0.58748360721149
 6693  0.42147552840622  0.42325640862338  6696  0.46237556455209  0.46440477643596  6697  0.47986830092227
0.48212721227457
 6701  0.30552189633557  0.30675323891523  6704  0.34999085208029  0.35148597323061  6705  0.36320057824557
0.36489924528297
 6709  0.18202000904966  0.18269678640934  6712  0.22896485030310  0.22991241884926  6713  0.23757117298981
0.23868624784134
 6717  0.05399648258722  0.05414173268619  6720  0.10227013497208  0.10267765768026  6721  0.10606824163226
0.10659600282361
 6725  0.60406665127214  0.60688687027071  6728  0.57349132716021  0.57611969630745  6731  0.54637833908859
0.54874086390296
 6735  0.51809295350119  0.52045596566075  6738  0.48246949206155  0.48461545485338  6741  0.45968394870908
0.46158516200956
 6745  0.41933777595133  0.42120970788190  6748  0.37954341072045  0.38117837255433  6751  0.36164409253974
0.36306370151415
 6755  0.31023210029099  0.31159186918012  6758  0.26724560879436  0.26835541366451  6761  0.25467015070025
0.25560240774813
 6765  0.19345827715838  0.19430160454500  6768  0.14833540079574  0.14892465386972  6771  0.14138944121077
0.14184733440758
 6775  0.07188426447786  0.07222698955594  6778  0.02574283049908  0.02582687484406  6781  0.02457296315122
0.02459950892961
 6785  0.46477295496442  0.46670355422739  6788  0.49680165733630  0.49892946161930  6789  0.52318285444849
0.52557435787858
 6793  0.37171121504260  0.37317445414644  6796  0.40777744026914  0.40945392773414  6797  0.42940526813645
0.43132046051419
 6801  0.26946697392493  0.27045656050940  6804  0.30868393771694  0.30989627924849  6805  0.32502817933085
0.32644602194140
 6809  0.16055156526690  0.16107912875876  6812  0.20195648366331  0.20270795588095  6813  0.21261782751313
0.21353339889618
```

6817 0.04763128069916 0.04773539426710 6820 0.09021344349983 0.09052828989049 6821 0.09493424757558 0.09536287489342
6825 0.44876583229496 0.45072372357340 6828 0.41791314814036 0.41968523126722 6831 0.39105707308292 0.39257771720566
6835 0.36324795903845 0.36477477532757 6838 0.32878025044228 0.33010695766590 6841 0.30767384491554 0.30878531389558
6845 0.26875447009083 0.26984386149511 6848 0.23151858612986 0.23240035520468 6851 0.21668001201626 0.21738959135769
6855 0.16760580881077 0.16826849623220 6858 0.12851511866418 0.12897128470575 6861 0.12030752830845 0.12064101560588
6865 0.06228308973104 0.06254980214093 6868 0.02230478035931 0.02236651314621 6871 0.02091058695088 0.02092185766528
6875 0.31276401907905 0.31390370611341 6878 0.34310798749630 0.34442096443183 6879 0.36833766406521 0.36989316754540
6883 0.22675059951286 0.22750034398998 6886 0.25974796627684 0.26067591038448 6887 0.27882332914463 0.27995461412740
6891 0.13511187768744 0.13549516836715 6894 0.16995368844841 0.17051215028973 6895 0.18240648507284 0.18312264899348
6899 0.04008753369117 0.04015365201645 6902 0.07592477194276 0.07614981515745 6903 0.08145137322014 0.08178159649211
6907 0.29818361339199 0.29935787718661 6910 0.26989200073845 0.27090721391756 6913 0.24609781036530 0.24690360704919
6917 0.22063233319767 0.22145140237965 6920 0.19006628979064 0.19072282870714 6923 0.17332967030747 0.17382392175333
6927 0.13760712216228 0.13809205909104 6930 0.10551510385168 0.10584221448118 6933 0.09624782255263 0.09646411461538
6937 0.05114059633334 0.05133243100633 6940 0.01831472885771 0.01835541366451 6943 0.01673051109242 0.01672904082956
6947 0.17842417210075 0.17894231397116 6950 0.20438735908535 0.20503683547307 6951 0.22572487192831 0.22656979459748
6955 0.10632724956390 0.10657486724767 6958 0.13374427299828 0.13411776966098 6959 0.14768246561211 0.14820281172333
6963 0.03155099819062 0.03158319359083 6966 0.05975594500328 0.05989627924849 6967 0.06595254776156 0.06618658376761
6971 0.16704823403915 0.16760607478254 6974 0.14390717803154 0.14434907319406 6977 0.12568254611556 0.12597813030688
6981 0.10419905860160 0.10451533715378 6984 0.07990020783705 0.08010695766590 6987 0.06979991882288 0.06991194697701
6991 0.03873037907639 0.03885108505778 6994 0.01387064311684 0.01389234298027 6997 0.01213510692484 0.01212429948811
7001 0.07490237908617 0.07503033919484 7004 0.09421517806205 0.09442096443183 7005 0.10929788916778 0.10963372937162
7009 0.02223044301073 0.02223505212040 7012 0.04210250228809 0.04216789816007 7013 0.04881785099964 0.04896183769007
7017 0.06819774708013 0.06836510074929 7020 0.05229562904974 0.05239920169315 7023 0.04160741578786 0.04163831517961
7027 0.02535497731832 0.02541309645575 7030 0.00908082127859 0.00908719670580 7033 0.00723589149981 0.00722101765502
7037 0.01235146661909 0.01233940987285 7040 0.02339232870776 0.02340120347172 7041 0.03046435728973 0.03053148867358
7045 0.01133575997113 0.01134935307775 7048 0.00406008048682 0.00405829348976 7051 0.00214842152390 0.00213993040208
7055 0.99221932836816 0.99732465707428 7058 0.98309167568236 0.98811523439862 7061 0.98668693308121 0.99174663408490
7065 0.95295082955849 0.95782627229675 7068 0.92644687608543 0.93109948672708 7071 0.93865886796191 0.94340572588151
7075 0.86910723316960 0.87348645716948 7078 0.82782214597077 0.83185851269382 7081 0.84727935655759 0.85145585438060
7085 0.74889041785613 0.75256098042718 7088 0.69686027711868 0.70010671028975 7091 0.72148262747026 0.72489771365710
7095 0.60405235055409 0.60688687027071 7098 0.54636304990748 0.54874086390296 7101 0.57356564439101 0.57611969630745
7105 0.44875109079871 0.45072372357340 7108 0.39104218247675 0.39257771720566 7111 0.41798740668538 0.41968523126722

7115	0.29816988491876	0.29935787718661	7118	0.24608472812864	0.24690360704919	7121	0.26995888050895	0.27090721391756
7125	0.16703685993886	0.16760607478254	7128	0.12567248600502	0.12597813030688	7131	0.14396003773267	0.14434907319406
7135	0.06818979867454	0.06836510074929	7138	0.04160121633675	0.04163831517961	7141	0.05232905300201	0.05239920169315
7145	0.01133163644087	0.01134935307775	7148	0.00214472612857	0.00213993040208	7151	0.00406700992112	0.00405829348976
7155	0.95688601997810	0.96174642343239	7158	0.97492942164402	0.97993850223311	7159	0.96948764090913	0.97445771976688
7163	0.87828891312702	0.88264418273435	7166	0.91260974042796	0.91724757291113	7167	0.89889497049776	0.90343795880678
7171	0.76207062915695	0.76570811750848	7174	0.80931376299857	0.81333559813481	7175	0.78893814519631	0.79282208776554
7179	0.61959224685031	0.62238474456056	7182	0.67514031687660	0.67837420599256	7183	0.65036679802795	0.65343795880678
7187	0.46478074872184	0.46670355422739	7190	0.52320471360021	0.52557435787858	7191	0.49672627706633	0.49892946161930
7195	0.31277055203343	0.31390370611341	7198	0.36835911472254	0.36989316754540	7199	0.34303623920935	0.34442096443183
7203	0.17842801481911	0.17894231397116	7206	0.22574441922745	0.22656979459748	7207	0.20432621724275	0.20503683547307
7211	0.07490225452255	0.07503033919484	7214	0.10931414416260	0.10963372937162	7215	0.09417052714843	0.09442096443183
7219	0.01234789532233	0.01233940987285	7222	0.03047569521821	0.03053148867358	7223	0.02336843703049	0.02340120347172
7227	0.96981613431523	0.97482011001528	7230	0.95177066914878	0.95662803121455	7233	0.95524069436947	0.96014371300095
7237	0.90749341222039	0.91212918069330	7240	0.87317024487620	0.87752579051652	7243	0.88467019673041	0.88912395204085
7247	0.80419506691250	0.80821720591697	7250	0.75695038198088	0.76058972529065	7253	0.77473380040199	0.77850808099961
7257	0.67002073692226	0.67325581377473	7260	0.61447187313653	0.61726635234273	7263	0.63617773733710	0.63912395204085
7267	0.51808540024645	0.52045596566075	7270	0.45966120714696	0.46158516200956	7273	0.48254455690288	0.48461545485338
7277	0.36324058026462	0.36477477532757	7280	0.30765208186241	0.30878531389558	7283	0.32885142861519	0.33010695766590
7287	0.22062646919018	0.22145140237965	7290	0.17331012099883	0.17382392175333	7293	0.19012643819186	0.19072282870714
7297	0.10419578434015	0.10451533715378	7300	0.06978342798753	0.06991194697701	7303	0.07994307767961	0.08010695766590
7307	0.02535399789740	0.02541309645575	7310	0.00722397039338	0.00722101765502	7313	0.00909487181175	0.00908719670580
7317	0.90709060625556	0.91169622366786	7320	0.93360532957936	0.93842300923754	7321	0.92840445600194	0.93317442234844
7325	0.80847389830885	0.81245524963461	7328	0.84976899668930	0.85408319411027	7329	0.83700792236442	0.84122455084753
7333	0.67752055207337	0.68070344723054	7336	0.72956095339485	0.73315771736796	7337	0.71120044537980	0.71466641012403
7341	0.52703006568929	0.52933760084375	7344	0.58473089188546	0.58748360721149	7345	0.56327929842805	0.56588839277437
7349	0.37171121018361	0.37317445414644	7352	0.42943405561522	0.43132046051419	7353	0.40770275201807	0.40945392773414
7357	0.22674843106897	0.22750034398998	7360	0.27885080902174	0.27995461412740	7361	0.25968032792514	0.26067591038448
7365	0.10632145718634	0.10657486724767	7368	0.14770709495547	0.14820281172333	7369	0.13369008889443	0.13411776966098
7373	0.02222288841314	0.02223505212040	7376	0.04883772993654	0.04896183769007	7377	0.04206696371892	0.04216789816007
7381	0.92350561388792	0.92831225660525	7384	0.89698999047112	0.90158547103557	7387	0.90024871982409	0.90489886716863
7391	0.83966860014063	0.84397244147798	7394	0.79837212224347	0.80234449700232	7397	0.80887758144330	0.81294899566772

7401 0.71946055628077 0.72304696473567 7404 0.66741970023033 0.67059269459825 7407 0.68309302099241 0.68639085494422

7411 0.57463204298216 0.57737285457920 7414 0.51693133611600 0.51922684821146 7417 0.53518847540155 0.53761283759456

7421 0.41933706489561 0.42120970788190 7424 0.36161434812993 0.36306370151415 7427 0.37961727570892 0.38117837255433

7431 0.26875448581557 0.26984386149511 7434 0.21665173262962 0.21738959135769 7437 0.23158477083222 0.23240035520468

7441 0.13760901800209 0.13809205909104 7444 0.09622199040555 0.09646411461538 7447 0.10556676085516 0.10584221448118

7451 0.03873368050789 0.03885108505778 7454 0.01211612502477 0.01212429948811 7457 0.01389195952650 0.01389234298027

7461 0.83493877829306 0.83919703713035 7464 0.86927041784233 0.87380042730712 7465 0.86443853304867 0.86891327362344

7469 0.71873338253415 0.72226097190448 7472 0.76598515353552 0.76988845253080 7473 0.75449102199634 0.75829740258220

7477 0.57626976623475 0.57893759895656 7480 0.63182629134689 0.63492706038855 7481 0.61593112926233 0.61891327362344

7485 0.42146910381679 0.42325640862338 7488 0.47990381551651 0.48212721227457 7489 0.46229988965441 0.46440477643596

7493 0.26945903393226 0.27045656050940 7496 0.32506307186799 0.32644602194140 7497 0.30861165280229 0.30989627924849

7501 0.13510086236184 0.13549516836715 7504 0.18243913333154 0.18312264899348 7505 0.16989162116219 0.17051215028973

7509 0.03154007706325 0.03158319359083 7512 0.06598087844898 0.06618658376761 7513 0.05971024843177 0.05989627924849

7517 0.85443390908937 0.85894627454584 7520 0.82010269967235 0.82434288436907 7523 0.82307004644730 0.82737240803946

7527 0.75115089791518 0.75503429976952 7530 0.70389899332526 0.70740681914320 7533 0.71315235169823 0.71675653699822

7537 0.61699498332997 0.62007290762727 7540 0.56143920694358 0.56408344619528 7543 0.57461875494641 0.57737240803946

7547 0.46507644337859 0.46727305951329 7550 0.40664251314569 0.40840225586211 7553 0.42100455448055 0.42286391085199

7557 0.31023813780239 0.31159186918012 7560 0.25463400366018 0.25560240774812 7563 0.26731633792191 0.26835541366451

7567 0.16761304056642 0.16826849623220 7570 0.12027323290142 0.12064101560588 7573 0.12857447563835 0.12897128470575

7577 0.05114835612348 0.05133243100633 7580 0.01670510603268 0.01672904082956 7583 0.01834288866937 0.01835541366451

7587 0.74221222887515 0.74603403440339 7590 0.78351368950115 0.78766197887905 7591 0.77916941286537 0.78325659634400

7595 0.61128275274181 0.61428223199932 7598 0.66332681466615 0.66673650213674 7599 0.65338122775613 0.65669845562050

7603 0.46081569595127 0.46291638561253 7606 0.51852148999829 0.52106239198028 7607 0.50548034195989 0.50792043827084

7611 0.30550849625026 0.30675323891523 7614 0.36324212533978 0.36489924528297 7615 0.34991581970210 0.35148597323061

7619 0.16053575324699 0.16107912875876 7622 0.21265783220070 0.21353339889618 7623 0.20188823563127 0.20270795588095

7627 0.04007364170248 0.04015365201645 7630 0.08148761720184 0.08178159649211 7631 0.07587026072438 0.07614981515745

7635 0.76430700728979 0.76843018452787 7638 0.72300758630559 0.72680224005222 7641 0.72561080789222 0.72947329433276

7645 0.64412610147554 0.64750470778557 7648 0.59208346724507 0.59505043764815 7651 0.59985855752402 0.60291515360926

7655 0.49932725748755 0.50183059762910 7658 0.44162341386814 0.44368459126136 7661 0.45198821027770 0.45413713625960

7665 0.34405372920197 0.34566745093179 7668 0.28632102582854 0.28752144456405 7671 0.29643892467391 0.29770267121937

7675 0.19347078477984 0.19430160454500 7678 0.14134764976410 0.14184733440758 7681 0.14840109424651 0.14892465386972

7685 0.06229522069314 0.06254980214093 7688 0.02087943805906 0.02092185766528 7691 0.02233912898534 0.02236651314621

7695 0.63120120598909 0.63450119780442 7698 0.67845320050520 0.68212867843074 7699 0.67470290227877
0.67831354218297
7703 0.48877512614257 0.49117782485650 7706 0.54432930660979 0.54716728628849 7707 0.53617487125097
0.53892941322420
7711 0.33400482759521 0.33549663452332 7714 0.39244207706029 0.39436743817451 7715 0.38257177333653
0.38442091603673
7719 0.18199987725370 0.18269678640934 7722 0.23761763359200 0.23868624784134 7723 0.22889220387811
0.22991241884926
7727 0.04761481562196 0.04773539426710 7730 0.09497759082990 0.09536287489342 7731 0.09015156030066
0.09052828989049
7735 0.65535110836847 0.65899279309326 7738 0.60810318001217 0.61136531246694 7741 0.61027871692871
0.61361212713653
7745 0.52123805617798 0.52403140095101 7748 0.46568725833642 0.46804193951901 7751 0.47179555884664
0.47422799817777
7755 0.36936137889653 0.37123155283703 7758 0.31092698643295 0.31236074918584 7761 0.31822763094391
0.31971950099029
7765 0.21454319960881 0.21555036250386 7768 0.15892560947633 0.15956090107186 7771 0.16455634385351
0.16521100380282
7775 0.07190051998690 0.07222698955594 7778 0.02453686189349 0.02459950892961 7781 0.02578254126644
0.02582687484406
7785 0.50464916622738 0.50734483592675 7788 0.55668502293622 0.55979910606418 7789 0.55362001801464
0.55666815741981
7793 0.35423624013677 0.35597898953996 7796 0.41193301464988 0.41412499590771 7797 0.40576645450522
0.40789014007015
7801 0.19896054808086 0.19981584284266 7804 0.25669729099406 0.25796184921040 7805 0.25023412624213
0.25145567502992
7809 0.05397787937247 0.05414173268619 7812 0.10611766127782 0.10659600282361 7813 0.10220245772093
0.10267765768026
7817 0.53025911432170 0.53332881202100 7820 0.47823006785873 0.48087454188358 7823 0.47992535380540
0.48264179290830
7827 0.38552474962002 0.38765470186453 7830 0.32783373598755 0.32950869549679 7833 0.33213053276397
0.33386377555864
7837 0.23030347576363 0.23149155516722 7840 0.17256939115083 0.17334554879948 7843 0.17663522783491
0.17742931051841
7847 0.07972655879042 0.08012570878043 7850 0.02758715940587 0.02767143864301 7853 0.02858789690437
0.02865129316875
7857 0.36568733634696 0.36769596040760 7860 0.42122130501397 0.42368542183959 7861 0.41891504723181
0.42131575514528
7865 0.21098889270670 0.21201477007442 7868 0.26941183801574 0.27088557372561 7869 0.26537711132098
0.26680725795781
7873 0.05900454397587 0.05921492196044 7876 0.11462913222153 0.11520438339244 7877 0.11172336699700
0.11229876077033
7881 0.39212679841848 0.39453250557993 7884 0.33660385233235 0.33854304414794 7887 0.33777936927106
0.33978721594239
7891 0.24034514765920 0.24173265746595 7894 0.18193000997822 0.18286185381476 7897 0.18432173608362
0.18527871875492
7901 0.08557637636180 0.08605146713278 7904 0.02995416385750 0.03006200570078 7907 0.03068451029535
0.03077022156744
7911 0.21776198616360 0.21899319003028 7914 0.27542073940149 0.27713919639802 7915 0.27392515737277
0.27558916071225
7919 0.06256609634920 0.06283004333297 7922 0.12028984428072 0.12097604970072 7923 0.11847141871392
0.11915469567202
7927 0.24438288932733 0.24602149947317 7930 0.18674297133616 0.18787549310543 7933 0.18737670210338
0.18856595003087
7937 0.08929292576657 0.08985835277587 7940 0.03157633687452 0.03171234640812 7943 0.03201551627851
0.03213148499064
7947 0.06455681222077 0.06489808051768 7950 0.12291907904852 0.12376888416887 7951 0.12225441238539
0.12307664651450
7955 0.09072684297266 0.09145262755084 7958 0.03239884060445 0.03258182389965 7961 0.03250936103486
0.03270156461507
7965 0.99222025782418 0.99732465707428 7968 0.98667620482131 0.99174663408490 7971 0.98309568598506
0.98811523439862
7975 0.96981113299077 0.97482011001528 7978 0.95521985947284 0.96014371300095 7981 0.95178445547327
0.95662803121455

7985 0.92349490938070 0.92831225660525 7988 0.90021827612322 0.90489886716863 7991 0.89701313985607 0.90158547103557

7995 0.85441776063309 0.85894627454584 7998 0.82303074280470 0.82737240803946 8001 0.82013464161416 0.82434288436907

8005 0.76428581390751 0.76843018452787 8008 0.72556361416199 0.72947329433276 8011 0.72304754023008 0.72680224005222

8015 0.65532539474759 0.65899279309326 8018 0.61022479835644 0.61361212713653 8021 0.60815017109951 0.61136531246694

8025 0.53022951717414 0.53332881202100 8028 0.47986604166878 0.48264179290830 8031 0.47828294938188 0.48087454188358

8035 0.39209404968080 0.39453250557993 8038 0.33771612581818 0.33978721594239 8041 0.33666132942475 0.33854304414794

8045 0.24434779669721 0.24602149947317 8048 0.18731108619539 0.18856595003087 8051 0.18680357908013 0.18787549310543

8055 0.09069302279800 0.09145262755084 8058 0.03246652968042 0.03270156461507 8061 0.03244574318887 0.03258182389965

8065 0.95688725786315 0.96174642343239 8068 0.96951084671643 0.97445771976688 8069 0.97491837166126 0.97993850223311

8073 0.90709694147436 0.91169622366786 8076 0.92843854627144 0.93317442234844 8077 0.93358441883594 0.93842300923754

8081 0.83495005301814 0.83919703713035 8084 0.86448265974881 0.86891327362344 8085 0.86924014679599 0.87380042730712

8089 0.74222817257935 0.74603403440339 8092 0.77922249405822 0.78325659634400 8093 0.78347480549538 0.78766197887905

8097 0.63122143767914 0.63450119780442 8100 0.67476364108903 0.67831354218297 8101 0.67840666624677 0.68212867843074

8105 0.50467320182057 0.50734483592675 8108 0.55368693225990 0.55666815741981 8109 0.55663199120824 0.55979910606418

8113 0.36571459674768 0.36769596040760 8116 0.41898650333195 0.42131575514528 8117 0.42116308866009 0.42368542183959

8121 0.21779176758240 0.21899319003028 8124 0.27399939874190 0.27558916071225 8125 0.27535877454550 0.27713919639802

8129 0.06457901048541 0.06489808051768 8132 0.12232849455682 0.12307664651450 8133 0.12285561409929 0.12376888416887

8137 0.95295614791018 0.95782627229675 8140 0.93862469698216 0.94340572588151 8143 0.92645515251909 0.93109948672708

8147 0.90749357193980 0.91212918069330 8150 0.88462597450641 0.88912395204085 8153 0.87318773083234 0.87752579051652

8157 0.83966359898845 0.84397244147798 8160 0.80882440132877 0.81294899566772 8163 0.79839839498350 0.80234449700232

8167 0.75114086933068 0.75503429976952 8170 0.71309153634515 0.71675653699822 8173 0.70393342397162 0.70740681914320

8177 0.64411130955464 0.64750470778557 8180 0.59979162353680 0.60291515360926 8183 0.59212523361578 0.59505043764815

8187 0.52121888683705 0.52403140095101 8190 0.47172417703671 0.47422799817777 8193 0.46573536283934 0.46804193951901

8197 0.38550169779036 0.38765470186453 8200 0.33205648248966 0.33386377555864 8203 0.32788702666642 0.32950869549679

8207 0.24031880460599 0.24173265746595 8210 0.18424689356003 0.18527871875492 8213 0.18198718437679 0.18286185381476

8217 0.08926686261119 0.08985835277587 8220 0.03196592229619 0.03213148499064 8223 0.03162151840217 0.03171234640812

8227 0.87828607331935 0.88264418273435 8230 0.89893924942837 0.90343795880678 8231 0.91259504084069 0.91724757291113

8235 0.80847657255825 0.81245524963461 8238 0.83706121893114 0.84122455084753 8239 0.84974527502337 0.85408319411027

8243 0.71874152387085 0.72226097190448 8246 0.75455203889609 0.75829740258220 8247 0.76595300458063 0.76988845253080

8251 0.61129618845497 0.61428223199932 8254 0.65344848720841 0.65669845562050 8255 0.66328704900116 0.66673650213674

8259 0.48879355991054 0.49117782485650 8262 0.53624674936783 0.53892941322420 8263 0.54428292828386 0.54716728628849

8267 0.35425925640497 0.35597898953996 8270 0.40584121873458 0.40789014007015 8271 0.41188119364617 0.41412499590771

```
 8275   0.21101593878856   0.21201477007442  8278   0.26545295758448   0.26680725795781  8279   0.26935587964224
0.27088557372561
 8283   0.06258750529141   0.06283004333297  8286   0.11854548479396   0.11915469567202  8287   0.12023193402276
0.12097604970072
 8291   0.86911658023812   0.87348645716948  8294   0.84722612084220   0.85145585438060  8297   0.82783396111783
0.83185851269382
 8301   0.80419884533591   0.80821720591697  8304   0.77467293430423   0.77850808099961  8307   0.75697090071757
0.76058972529065
 8311   0.71945869563896   0.72304696473567  8314   0.68302605006917   0.68639085494422  8317   0.66744844969244
0.67059269459825
 8321   0.61698756287880   0.62007290762727  8324   0.57454736535092   0.57737240803946  8327   0.56147552358707
0.56408344619528
 8331   0.49931450255421   0.50183059762910  8334   0.45191420534218   0.45413713625960  8337   0.44166645822799
0.44368459126136
 8341   0.36934365306397   0.37123155283703  8344   0.31815288443604   0.31971950099029  8347   0.31097576160219
0.31236074918584
 8351   0.23028127722974   0.23149155516722  8354   0.17656167696691   0.17742931051841  8357   0.17262274660090
0.17334554879948
 8361   0.08555308192753   0.08605146713278  8364   0.03063708611253   0.03077022156744  8367   0.02999736787712
0.03006200570078
 8371   0.76206479419876   0.76570811750848  8374   0.78899909235608   0.79282208776554  8375   0.80929546859685
0.81333559813481
 8379   0.67752087369723   0.68070344723054  8382   0.71126763094261   0.71466641012403  8383   0.72953418496729
0.73315771736796
 8387   0.57627628239049   0.57893759895656  8390   0.61600293992085   0.61891327362344  8391   0.63179174274854
0.63492706038855
 8395   0.46082830401799   0.46291638561253  8398   0.50555506304754   0.50792043827084  8399   0.51848005724970
0.52106239198028
 8403   0.33402328472202   0.33549663452332  8406   0.38264762922415   0.38442091603673  8407   0.39239483427314
0.39436743817451
 8411   0.19898444992189   0.19981584284266  8414   0.25030931920569   0.25145567502992  8415   0.25664546503134
0.25796184921040
 8419   0.05902480884369   0.05921492196044  8422   0.11179513825894   0.11229876077033  8423   0.11457482294849
0.11520438339244
 8427   0.74890286093067   0.75256098042718  8430   0.72141555782593   0.72489771365710  8433   0.69687450801899
0.70010671028975
 8437   0.67002705843159   0.67325581377473  8440   0.63610623849955   0.63912395204085  8443   0.61449419702194
0.61726635234273
 8447   0.57463213857903   0.57737285457920  8450   0.53511436483771   0.53761283759456  8453   0.51696126796857
0.51922684821146
 8457   0.46507037664573   0.46727305951329  8460   0.42092972803034   0.42286391085199  8463   0.40667939714301
0.40840225586211
 8467   0.34404172811008   0.34566745093179  8470   0.29636530729524   0.29770267121937  8473   0.28636404858136
0.28752144456405
 8477   0.21452565348081   0.21555036250386  8480   0.16448587377677   0.16521100380282  8483   0.15897379832127
0.15956090107186
 8487   0.07970654884308   0.08012570878043  8490   0.02854379079365   0.02865129316875  8493   0.02762731078678
0.02767143864301
 8497   0.61958470100308   0.62238474456056  8500   0.65043845998142   0.65343795880678  8501   0.67511955118282
0.67837420599256
 8505   0.52702945583470   0.52933760084375  8508   0.56335384630959   0.56588839277437  8509   0.58470236662988
0.58748360721149
 8513   0.42147552840619   0.42325640862338  8516   0.46237556455206   0.46440477643596  8517   0.47986830092224
0.48212721227457
 8521   0.30552189633555   0.30675323891523  8524   0.34999085208027   0.35148597323061  8525   0.36320057824554
0.36489924528297
 8529   0.18202000904965   0.18269678640934  8532   0.22896485030309   0.22991241884926  8533   0.23757117298979
0.23868624784134
 8537   0.05399648258722   0.05414173268619  8540   0.10227013497207   0.10267765768026  8541   0.10606824163226
0.10659600282361
 8545   0.60406665127209   0.60688687027071  8548   0.57349132716017   0.57611969630745  8551   0.54637833908854
0.54874086390296
 8555   0.51809295350115   0.52045596566075  8558   0.48246949206152   0.48461545485338  8561   0.45968394870905
0.46158516200956
```

8565 0.41933777595131 0.42120970788190 8568 0.37954341072043 0.38117837255433 8571 0.36164409253972 0.36306370151415

8575 0.31023210029097 0.31159186918012 8578 0.26724560879434 0.26835541366451 8581 0.25467015070023 0.25560240774812

8585 0.19345827715837 0.19430160454500 8588 0.14833540079573 0.14892465386972 8591 0.14138944121076 0.14184733440758

8595 0.07188426447786 0.07222698955594 8598 0.02574283049908 0.02582687484406 8601 0.02457296315122 0.02459950892961

8605 0.46477295496439 0.46670355422739 8608 0.49680165733626 0.49892946161930 8609 0.52318285444845 0.52557435787858

8613 0.37171121504258 0.37317445414644 8616 0.40777744026911 0.40945392773414 8617 0.42940526813642 0.43132046051419

8621 0.26946697392491 0.27045656050940 8624 0.30868393771692 0.30989627924849 8625 0.32502817933083 0.32644602194140

8629 0.16055156526689 0.16107912875876 8632 0.20195648366330 0.20270795588095 8633 0.21261782751312 0.21353339889618

8637 0.04763128069916 0.04773539426710 8640 0.09021344349982 0.09052828989049 8641 0.09493424757558 0.09536287489342

8645 0.44876583229492 0.45072372357340 8648 0.41791314814033 0.41968523126722 8651 0.39105707308289 0.39257771720566

8655 0.36324795903843 0.36477477532757 8658 0.32878025044226 0.33010695766590 8661 0.30767384491552 0.30878531389558

8665 0.26875447009081 0.26984386149511 8668 0.23151858612985 0.23240035520468 8671 0.21668001201625 0.21738959135769

8675 0.16760580881076 0.16826849623220 8678 0.12851511866417 0.12897128470575 8681 0.12030752830844 0.12064101560588

8685 0.06228308973103 0.06254980214093 8688 0.02230478035931 0.02236651314621 8691 0.02091058695088 0.02092185766528

8695 0.31276401907903 0.31390370611341 8698 0.34310798749627 0.34442096443183 8699 0.36833766406518 0.36989316754540

8703 0.22675059951285 0.22750034398998 8706 0.25974796627682 0.26067591038448 8707 0.27882332914461 0.27995461412740

8711 0.13511187768743 0.13549516836715 8714 0.16995368844840 0.17051215028973 8715 0.18240648507283 0.18312264899348

8719 0.04008753369117 0.04015365201645 8722 0.07592477194276 0.07614981515745 8723 0.08145137322014 0.08178159649211

8727 0.29818361339197 0.29935787718661 8730 0.26989200073843 0.27090721391756 8733 0.24609781036528 0.24690360704919

8737 0.22063233319765 0.22145140237965 8740 0.19006628979062 0.19072282870714 8743 0.17332967030745 0.17382392175333

8747 0.13760712216227 0.13809205909104 8750 0.10551510385168 0.10584221448118 8753 0.09624782255262 0.09646411461538

8757 0.05114059633334 0.05133243100633 8760 0.01831472885771 0.01835541366451 8763 0.01673051109242 0.01672904082956

8767 0.17842417210074 0.17894231397116 8770 0.20438735908534 0.20503683547307 8771 0.22572487192830 0.22656979459748

8775 0.10632724956389 0.10657486724767 8778 0.13374427299828 0.13411776966098 8779 0.14768246561210 0.14820281172333

8783 0.03155099819061 0.03158319359083 8786 0.05975594500327 0.05989627924849 8787 0.06595254776156 0.06618658376761

8791 0.16704823403914 0.16760607478254 8794 0.14390717803153 0.14434907319406 8797 0.12568254611555 0.12597813030688

8801 0.10419905860159 0.10451533715378 8804 0.07990020783704 0.08010695766590 8807 0.06979991882288 0.06991194697701

8811 0.03873037907638 0.03885108505778 8814 0.01387064311683 0.01389234298027 8817 0.01213510692484 0.01212429948811

8821 0.07490237908616 0.07503033919484 8824 0.09421517806204 0.09442096443183 8825 0.10929788916778 0.10963372937162

8829 0.02223044301072 0.02223505212040 8832 0.04210250228809 0.04216789816007 8833 0.04881785099964 0.04896183769007

8837 0.06819774708013 0.06836510074929 8840 0.05229562904974 0.05239920169315 8843 0.04160741578785 0.04163831517961

8847 0.02535497731831 0.02541309645575 8850 0.00908082127859 0.00908719670580 8853 0.00723589149981 0.00722101765502

8857	0.01235146661909	0.01233940987285	8860	0.02339232870776	0.02340120347172	8861	0.03046435728972	0.03053148867358
8865	0.01133575997113	0.01134935307775	8868	0.00406008048682	0.00405829348976	8871	0.00214842152390	0.00213993040208
8874	0.99221932836816	0.99732465707428	8877	0.98309167568236	0.98811523439862	8879	0.98668693308121	0.99174663408490
8883	0.95295082955847	0.95782627229675	8886	0.92644687608539	0.93109948672708	8889	0.93865886796189	0.94340572588151
8893	0.86910723316956	0.87348645716948	8896	0.82782214597073	0.83185851269382	8899	0.84727935655755	0.85145585438060
8903	0.74889041785608	0.75256098042718	8906	0.69686027711863	0.70010671028975	8909	0.72148262747021	0.72489771365710
8913	0.60405235055404	0.60688687027071	8916	0.54636304990744	0.54874086390296	8919	0.57356564439096	0.57611969630745
8923	0.44875109079868	0.45072372357340	8926	0.39104218247673	0.39257771720566	8929	0.41798740668535	0.41968523126722
8933	0.29816988491874	0.29935787718661	8936	0.24608472812862	0.24690360704919	8939	0.26995888050894	0.27090721391756
8943	0.16703685993885	0.16760607478254	8946	0.12567248600501	0.12597813030688	8949	0.14396003773266	0.14434907319406
8953	0.06818979867454	0.06836510074929	8956	0.04160121633674	0.04163831517961	8959	0.05232905300201	0.05239920169315
8963	0.01133163644087	0.01134935307775	8966	0.00214472612857	0.00213993040208	8969	0.00406700992112	0.00405829348976
8973	0.95688601997809	0.96174642343239	8976	0.97492942164402	0.97993850223311	8977	0.96948764090912	0.97445771976688
8981	0.87828891312698	0.88264418273435	8984	0.91260974042793	0.91724757291113	8985	0.89889497049772	0.90343795880678
8989	0.76207062915690	0.76570811750848	8992	0.80931376299852	0.81333559813481	8993	0.78893814519626	0.79282208776554
8997	0.61959224685027	0.62238474456056	9000	0.67514031687655	0.67837420599256	9001	0.65036679802790	0.65343795880678
9005	0.46478074872181	0.46670355422739	9008	0.52320471360017	0.52557435787858	9009	0.49672627706629	0.49892946161930
9013	0.31277055203341	0.31390370611341	9016	0.36835911472251	0.36989316754540	9017	0.34303623920933	0.34442096443183
9021	0.17842801481910	0.17894231397116	9024	0.22574441922744	0.22656979459748	9025	0.20432621724274	0.20503683547307
9029	0.07490225452254	0.07503033919484	9032	0.10931414416259	0.10963372937162	9033	0.09417052714842	0.09442096443183
9037	0.01234789532233	0.01233940987285	9040	0.03047569521821	0.03053148867358	9041	0.02336843703048	0.02340120347172
9044	0.96981613431524	0.97482011001528	9047	0.95177066914877	0.95662803121455	9049	0.95524069436948	0.96014371300095
9053	0.90749341222036	0.91212918069330	9056	0.87317024487616	0.87752579051652	9059	0.88467019673038	0.88912395204085
9063	0.80419506691245	0.80821720591697	9066	0.75695038198083	0.76058972529065	9069	0.77473380040195	0.77850808099961
9073	0.67002073692221	0.67325581377473	9076	0.61447187313649	0.61726635234273	9079	0.63617773733705	0.63912395204085
9083	0.51808540024641	0.52045596566075	9086	0.45966120714693	0.46158516200956	9089	0.48254455690284	0.48461545485338
9093	0.36324058026459	0.36477477532757	9096	0.30765208186239	0.30878531389558	9099	0.32885142861517	0.33010695766590
9103	0.22062646919016	0.22145140237965	9106	0.17331012099882	0.17382392175333	9109	0.19012643819184	0.19072282870714
9113	0.10419578434015	0.10451533715378	9116	0.06978342798753	0.06991194697701	9119	0.07994307767961	0.08010695766590
9123	0.02535399789740	0.02541309645575	9126	0.00722397039338	0.00722101765502	9129	0.00909487181175	0.00908719670580
9133	0.90709060625554	0.91169622366786	9136	0.93360532957936	0.93842300923754	9137	0.92840445600192	0.93317442234844
9141	0.80847389830881	0.81245524963461	9144	0.84976899668926	0.85408319411027	9145	0.83700792236438	0.84122455084753

```
 9149  0.67752055207332  0.68070344723054  9152  0.72956095339480  0.73315771736796  9153  0.71120044537974
0.71466641012403
 9157  0.52703006568925  0.52933760084375  9160  0.58473089188542  0.58748360721149  9161  0.56327929842801
0.56588839277437
 9165  0.37171121018358  0.37317445414644  9168  0.42943405561519  0.43132046051419  9169  0.40770275201804
0.40945392773414
 9173  0.22674843106896  0.22750034398998  9176  0.27885080902172  0.27995461412740  9177  0.25968032792513
0.26067591038448
 9181  0.10632145718633  0.10657486724767  9184  0.14770709495546  0.14820281172333  9185  0.13369008889442
0.13411776966098
 9189  0.02222288841313  0.02223505212040  9192  0.04883772993654  0.04896183769007  9193  0.04206696371892
0.04216789816007
 9196  0.92350561388792  0.92831225660525  9199  0.89698999047111  0.90158547103557  9201  0.90024871982409
0.90489886716863
 9205  0.83966860014060  0.84397244147798  9208  0.79837212224343  0.80234449700232  9211  0.80887758144326
0.81294899566772
 9215  0.71946055628072  0.72304696473567  9218  0.66741970023028  0.67059269459825  9221  0.68309302099237
0.68639085494422
 9225  0.57463204298212  0.57737285457920  9228  0.51693133611596  0.51922684821146  9231  0.53518847540151
0.53761283759456
 9235  0.41933706489558  0.42120970788190  9238  0.36161434812991  0.36306370151415  9241  0.37961727570890
0.38117837255433
 9245  0.26875448581556  0.26984386149511  9248  0.21665173262961  0.21738959135769  9251  0.23158477083221
0.23240035520468
 9255  0.13760901800208  0.13809205909104  9258  0.09622199040554  0.09646411461538  9261  0.10556676085516
0.10584221448118
 9265  0.03873368050788  0.03885108505778  9268  0.01211612502477  0.01212429948811  9271  0.01389195952650
0.01389234298027
 9275  0.83493877829304  0.83919703713035  9278  0.86927041784232  0.87380042730712  9279  0.86443853304865
0.86891327362344
 9283  0.71873338253410  0.72226097190448  9286  0.76598515353549  0.76988845253080  9287  0.75449102199629
0.75829740258220
 9291  0.57626976623470  0.57893759895656  9294  0.63182629134684  0.63492706038855  9295  0.61593112926228
0.61891327362344
 9299  0.42146910381676  0.42325640862338  9302  0.47990381551648  0.48212721227457  9303  0.46229988965437
0.46440477643596
 9307  0.26945903393224  0.27045656050940  9310  0.32506307186797  0.32644602194140  9311  0.30861165280227
0.30989627924849
 9315  0.13510086236183  0.13549516836715  9318  0.18243913333153  0.18312264899348  9319  0.16989162116218
0.17051215028973
 9323  0.03154007706325  0.03158319359083  9326  0.06598087844898  0.06618658376761  9327  0.05971024843177
0.05989627924849
 9330  0.85443390908936  0.85894627454584  9333  0.82010269967233  0.82434288436907  9335  0.82307004644729
0.82737240803946
 9339  0.75115089791514  0.75503429976952  9342  0.70389899332522  0.70740681914320  9345  0.71315235169819
0.71675653699822
 9349  0.61699498332993  0.62007290762727  9352  0.56143920694354  0.56408344619528  9355  0.57461875494636
0.57737240803946
 9359  0.46507644337856  0.46727305951329  9362  0.40664251314567  0.40840225586211  9365  0.42100455448052
0.42286391085199
 9369  0.31023813780237  0.31159186918012  9372  0.25463400366016  0.25560240774812  9375  0.26731633792189
0.26835541366451
 9379  0.16761304056641  0.16826849623220  9382  0.12027323290141  0.12064101560588  9385  0.12857447563834
0.12897128470575
 9389  0.05114835612348  0.05133243100633  9392  0.01670510603268  0.01672904082956  9395  0.01834288866937
0.01835541366451
 9399  0.74221222887513  0.74603403440339  9402  0.78351368950113  0.78766197887905  9403  0.77916941286535
0.78325659634400
 9407  0.61128275274177  0.61428223199932  9410  0.66332681466611  0.66673650213674  9411  0.65338122775608
0.65669845562050
 9415  0.46081569595124  0.46291638561253  9418  0.51852148999825  0.52106239198028  9419  0.50548034195986
0.50792043827084
 9423  0.30550849625025  0.30675323891523  9426  0.36324212533976  0.36489924528297  9427  0.34991581970208
0.35148597323061
```

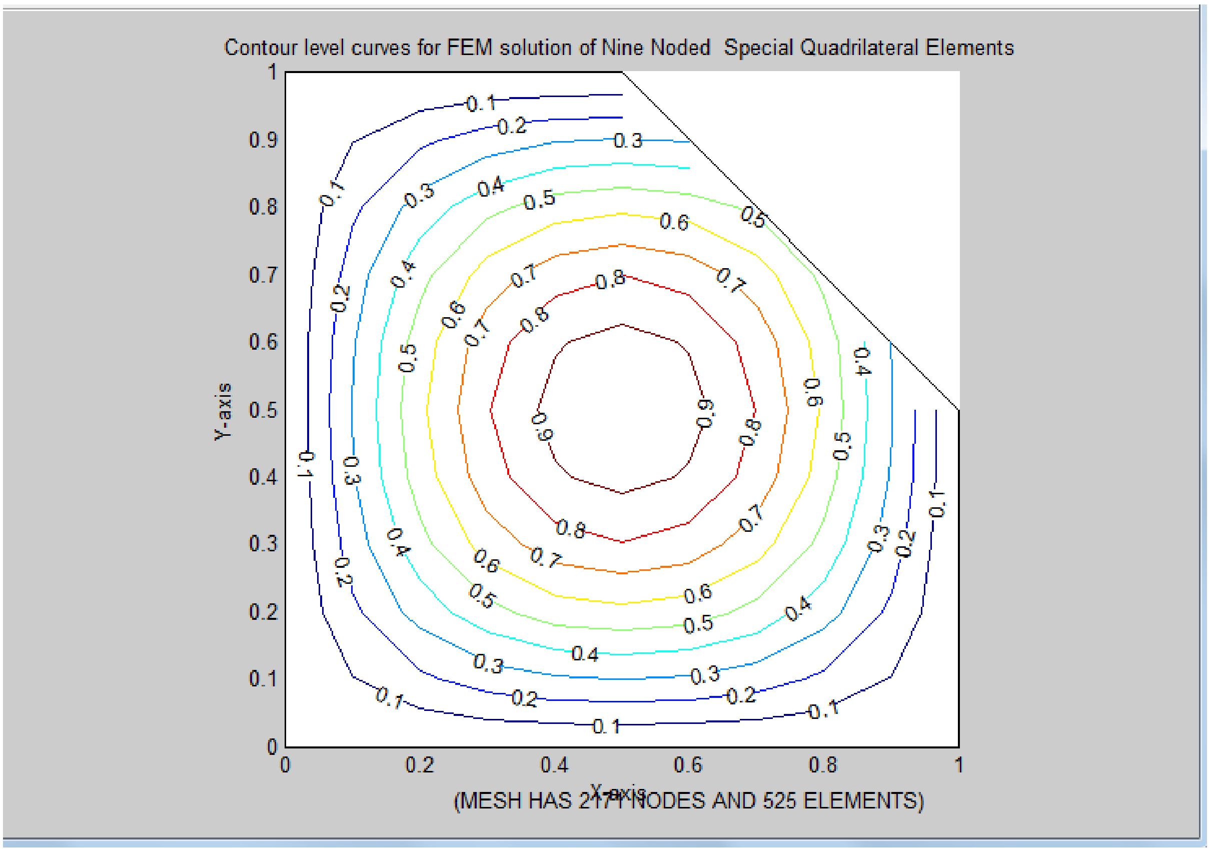
```
 9431  0.16053575324699  0.16107912875876  9434  0.21265783220069  0.21353339889618  9435  0.20188823563126
0.20270795588095
 9439  0.04007364170247  0.04015365201645  9442  0.08148761720184  0.08178159649211  9443  0.07587026072437
0.07614981515745
 9446  0.76430700728977  0.76843018452787  9449  0.72300758630557  0.72680224005222  9451  0.72561080789221
0.72947329433276
 9455  0.64412610147550  0.64750470778557  9458  0.59208346724503  0.59505043764815  9461  0.59985855752398
0.60291515360926
 9465  0.49932725748752  0.50183059762910  9468  0.44162341386811  0.44368459126136  9471  0.45198821027767
0.45413713625960
 9475  0.34405372920195  0.34566745093179  9478  0.28632102582853  0.28752144456405  9481  0.29643892467389
0.29770267121937
 9485  0.19347078477983  0.19430160454500  9488  0.14134764976409  0.14184733440758  9491  0.14840109424650
0.14892465386972
 9495  0.06229522069314  0.06254980214093  9498  0.02087943805906  0.02092185766528  9501  0.02233912898534
0.02236651314621
 9505  0.63120120598906  0.63450119780442  9508  0.67845320050518  0.68212867843074  9509  0.67470290227874
0.67831354218297
 9513  0.48877512614253  0.49117782485650  9516  0.54432930660976  0.54716728628849  9517  0.53617487125093
0.53892941322420
 9521  0.33400482759519  0.33549663452332  9524  0.39244207706026  0.39436743817451  9525  0.38257177333651
0.38442091603673
 9529  0.18199987725368  0.18269678640934  9532  0.23761763359199  0.23868624784134  9533  0.22889220387809
0.22991241884926
 9537  0.04761481562196  0.04773539426710  9540  0.09497759082989  0.09536287489342  9541  0.09015156030066
0.09052828989049
 9544  0.65535110836845  0.65899279309326  9547  0.60810318001215  0.61136531246694  9549  0.61027871692869
0.61361212713653
 9553  0.52123805617794  0.52403140095101  9556  0.46568725833639  0.46804193951901  9559  0.47179555884661
0.47422799817777
 9563  0.36936137889651  0.37123155283703  9566  0.31092698643293  0.31236074918584  9569  0.31822763094389
0.31971950099029
 9573  0.21454319960880  0.21555036250386  9576  0.15892560947632  0.15956090107186  9579  0.16455634385350
0.16521100380282
 9583  0.07190051998690  0.07222698955594  9586  0.02453686189349  0.02459950892961  9589  0.02578254126644
0.02582687484406
 9593  0.50464916622735  0.50734483592675  9596  0.55668502293620  0.55979910606418  9597  0.55362001801462
0.55666815741981
 9601  0.35423624013674  0.35597898953996  9604  0.41193301464985  0.41412499590771  9605  0.40576645450519
0.40789014007015
 9609  0.19896054808085  0.19981584284266  9612  0.25669729099404  0.25796184921040  9613  0.25023412624211
0.25145567502992
 9617  0.05397787937247  0.05414173268619  9620  0.10611766127781  0.10659600282361  9621  0.10220245772093
0.10267765768026
 9624  0.53025911432168  0.53332881202100  9627  0.47823006785871  0.48087454188358  9629  0.47992535380538
0.48264179290830
 9633  0.38552474961999  0.38765470186453  9636  0.32783373598752  0.32950869549679  9639  0.33213053276395
0.33386377555864
 9643  0.23030347576362  0.23149155516722  9646  0.17256939115082  0.17334554879948  9649  0.17663522783490
0.17742931051841
 9653  0.07972655879041  0.08012570878043  9656  0.02758715940587  0.02767143864301  9659  0.02858789690437
0.02865129316875
 9663  0.36568733634695  0.36769596040760  9666  0.42122130501395  0.42368542183959  9667  0.41891504723179
0.42131575514528
 9671  0.21098889270668  0.21201477007442  9674  0.26941183801572  0.27088557372561  9675  0.26537711132096
0.26680725795781
 9679  0.05900454397586  0.05921492196044  9682  0.11462913222152  0.11520438339244  9683  0.11172336699699
0.11229876077033
 9686  0.39212679841847  0.39453250557993  9689  0.33660385233233  0.33854304414794  9691  0.33777936927105
0.33978721594239
 9695  0.24034514765918  0.24173265746595  9698  0.18193000997821  0.18286185381476  9701  0.18432173608361
0.18527871875492
 9705  0.08557637636180  0.08605146713278  9708  0.02995416385750  0.03006200570078  9711  0.03068451029534
0.03077022156744
```

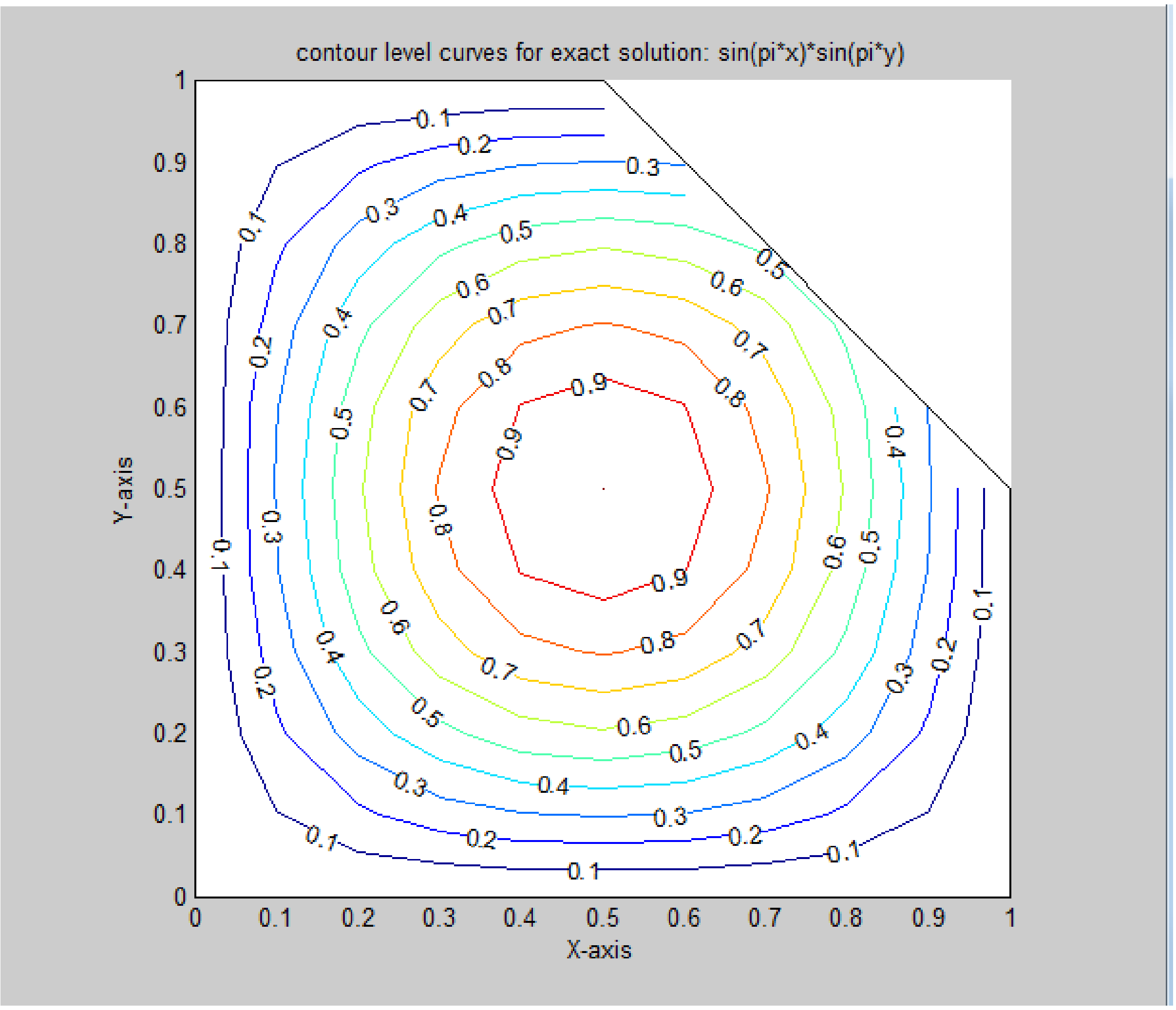
```
 9715   0.21776198616359   0.21899319003028  9718   0.27542073940148   0.27713919639802  9719   0.27392515737276
0.27558916071225
 9723   0.06256609634919   0.06283004333297  9726   0.12028984428072   0.12097604970072  9727   0.11847141871391
0.11915469567202
 9730   0.24438288932732   0.24602149947317  9733   0.18674297133615   0.18787549310543  9735   0.18737670210338
0.18856595003087
 9739   0.08929292576657   0.08985835277587  9742   0.03157633687452   0.03171234640812  9745   0.03201551627851
0.03213148499064
 9749   0.06455681222077   0.06489808051768  9752   0.12291907904851   0.12376888416887  9753   0.12225441238539
0.12307664651450
 9756   0.09072684297265   0.09145262755084  9759   0.03239884060445   0.03258182389965  9761   0.03250936103486
0.03270156461507
--------------------------------------------------------------------------------------------------------------------------------------------------------------
---------------------------------------
```

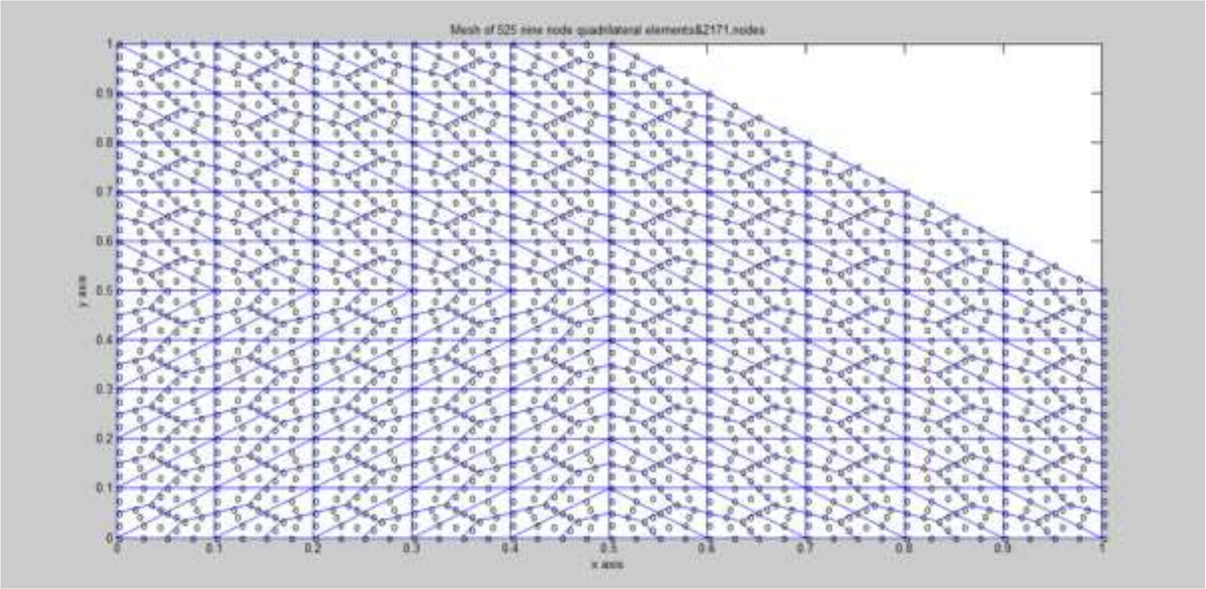
MESHES AND CONTOUR LEVEL CURVES FOR A PENTAGONAL DOMAIN WITH 9-NO0DED QUADRILATERAL ELEMENTS

Mesh No.1

Mesh with 525 nine noded quadrilateral elements & no. of nodes = 2171

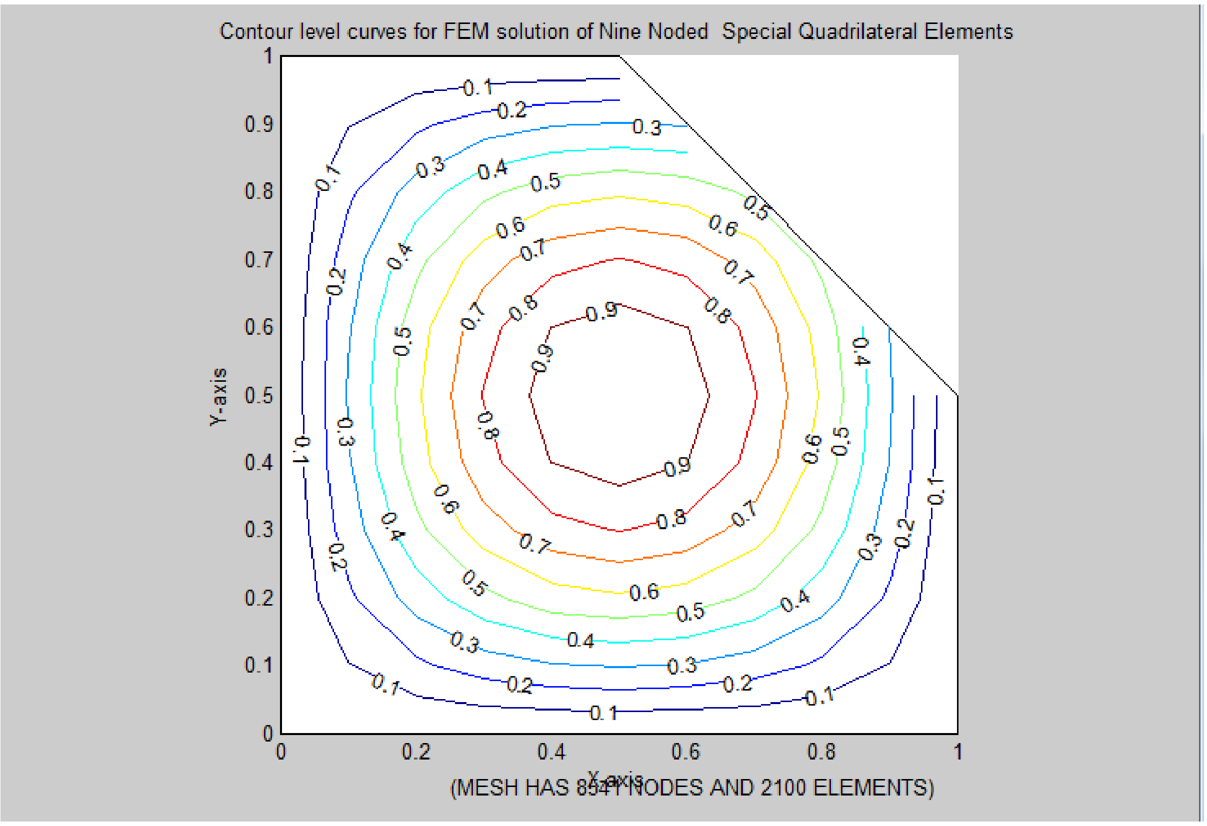

Contour level curves for FEM solution of Nine Noded Special Quadrilateral Elements
Y-axis
X-axis
(MESH HAS 217 NODES AND 525 ELEMENTS)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1

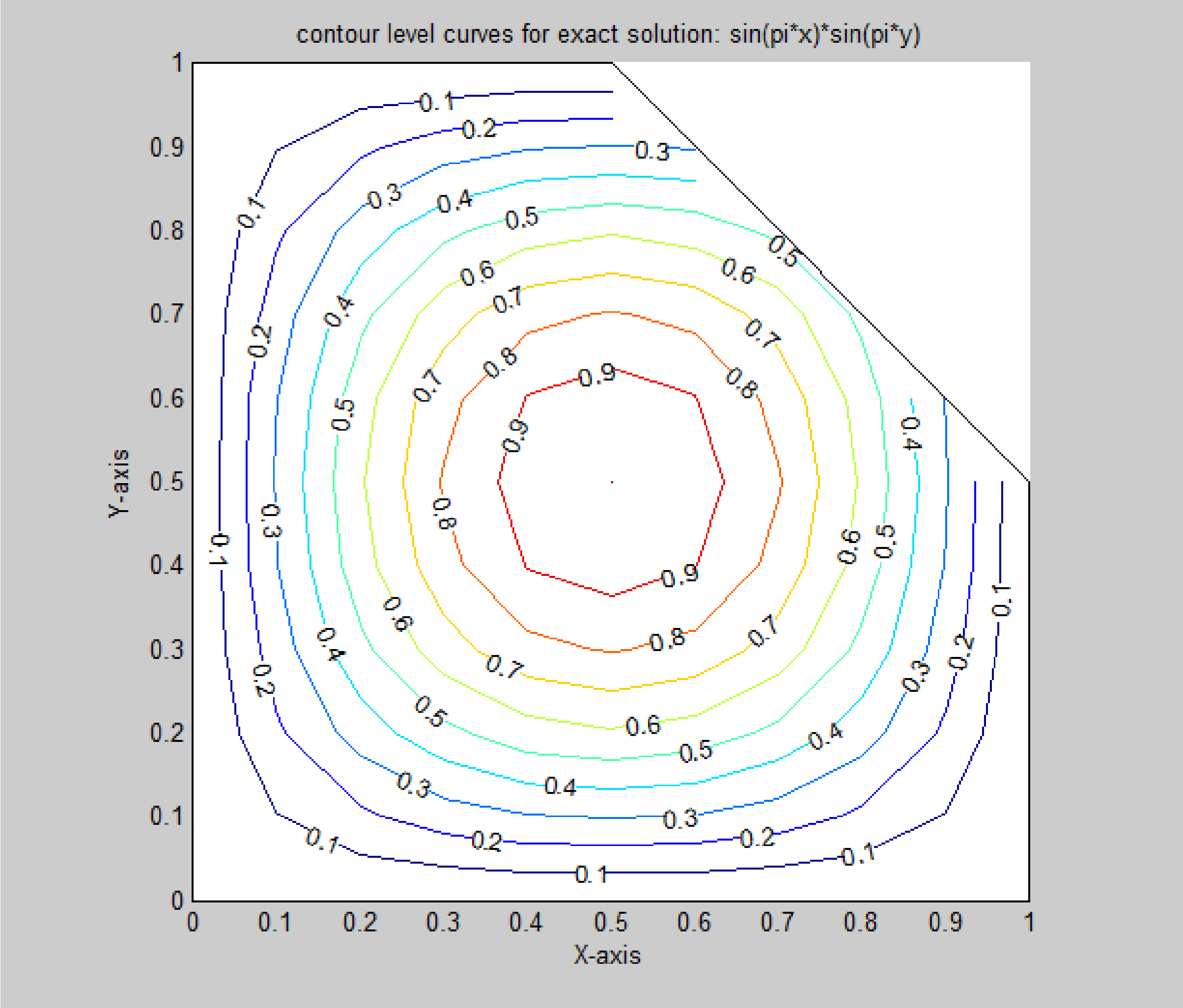

contour level curves for exact solution: sin(pi*x)*sin(pi*y)
Y-axis
X-axis
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

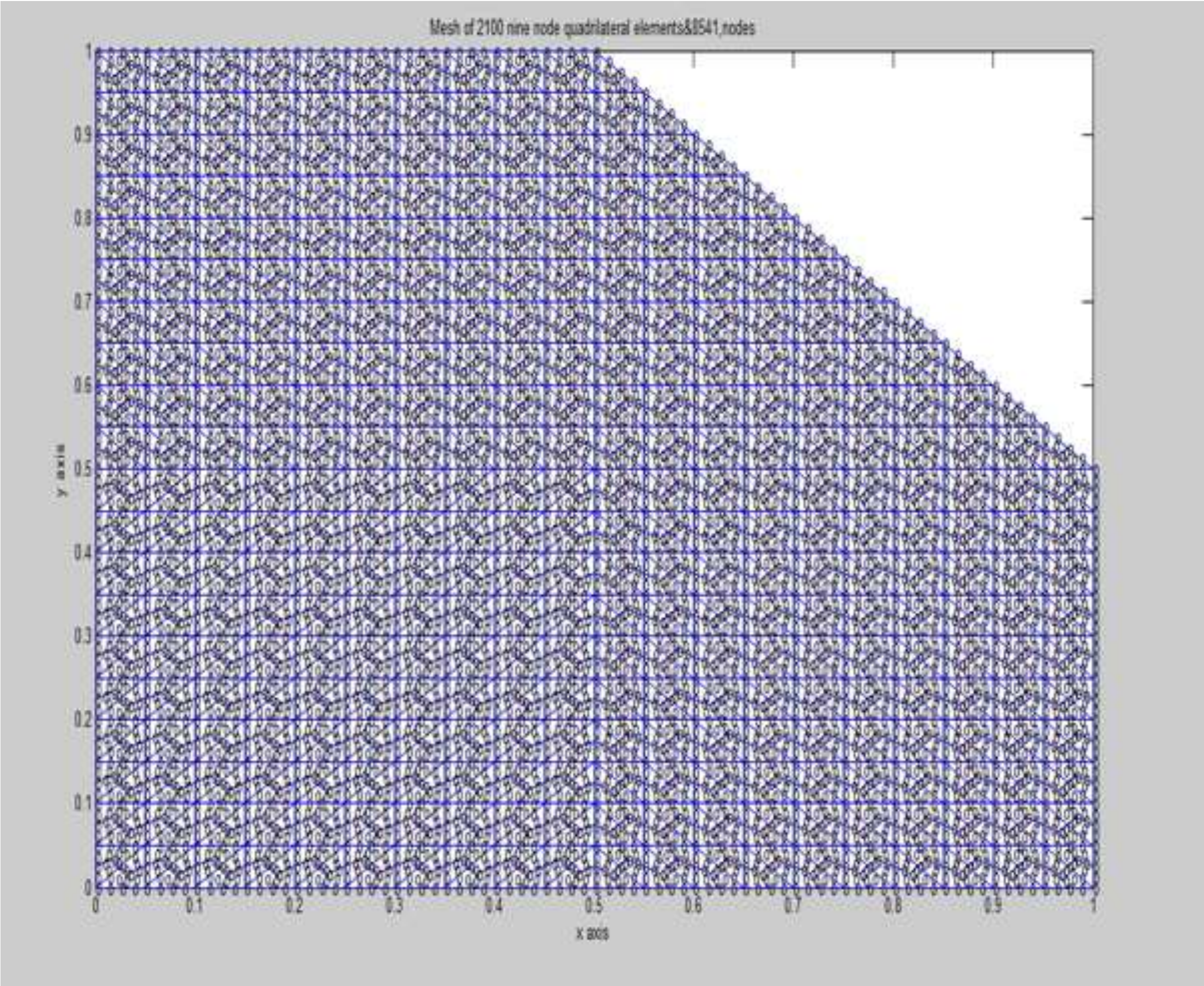


Mesh No.2

Mesh with 2100 nine noded quadrilateral elements & no. of nodes = 8541


Contour level curves for FEM solution of Nine Noded Special Quadrilateral Elements
Y-axis
X-axis
(MESH HAS 8541 NODES AND 2100 ELEMENTS)
0
0.2
0.4
0.6
0.8
1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

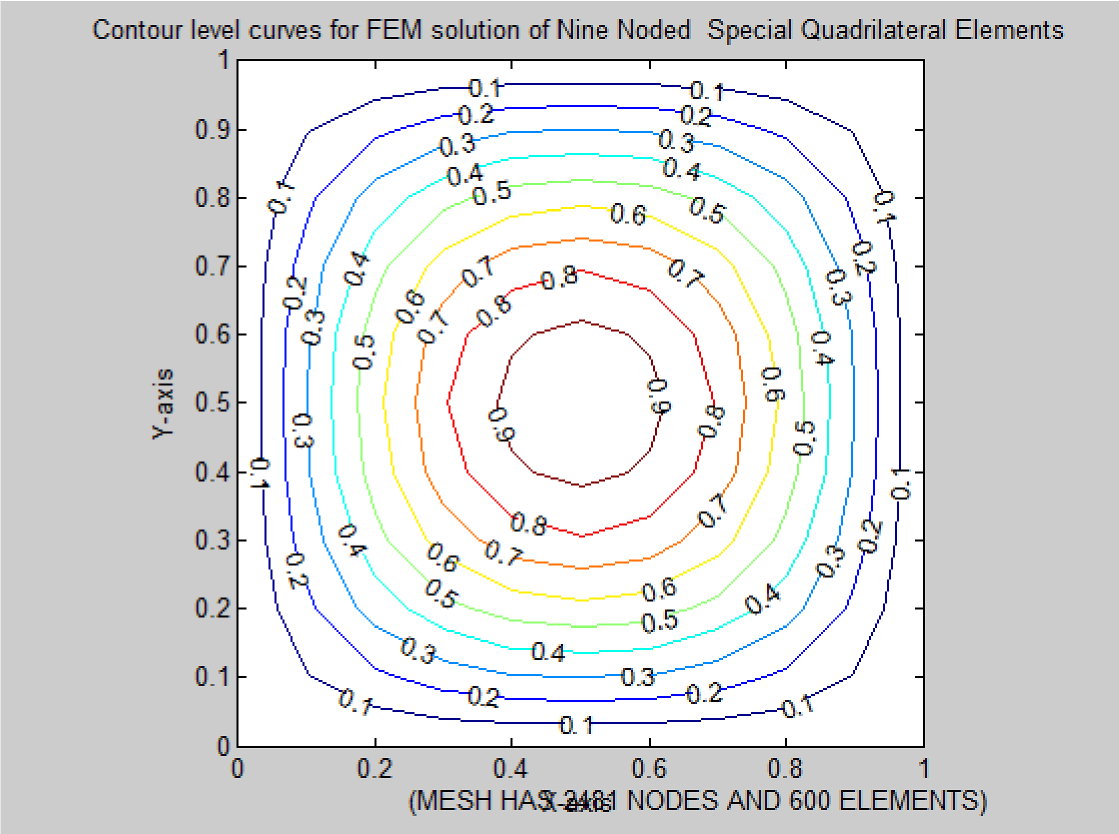

contour level curves for exact solution: sin(pi*x)*sin(pi*y)
Y-axis
X-axis
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

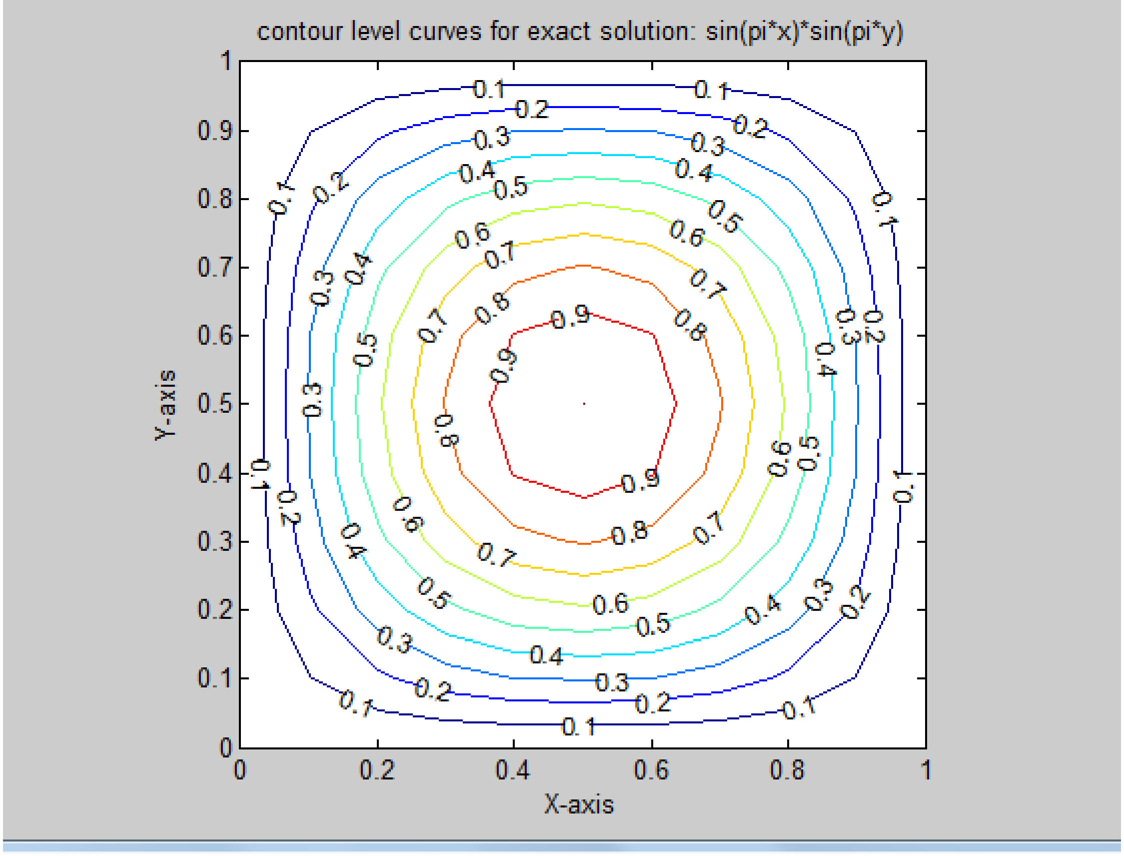


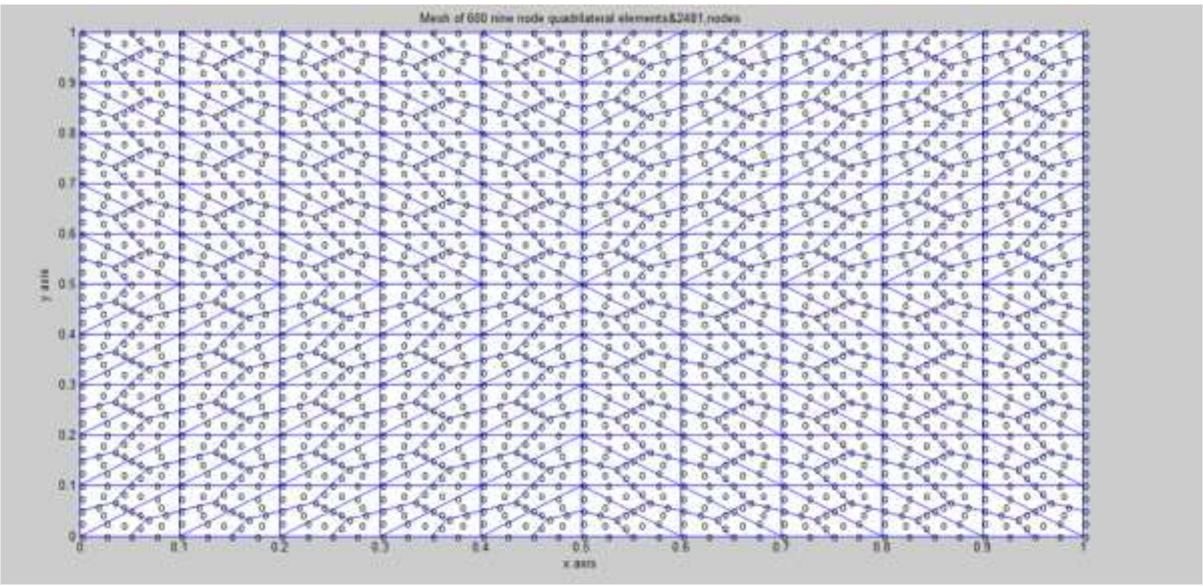
MESHES AND CONTOUR LEVEL CURVES FOR A SQUARE DOMAIN WITH 9-NO0DED QUADRILATERAL ELEMENTS

Mesh No.1

Mesh with 600 nine noded quadrilateral elements & no. of nodes = 2481

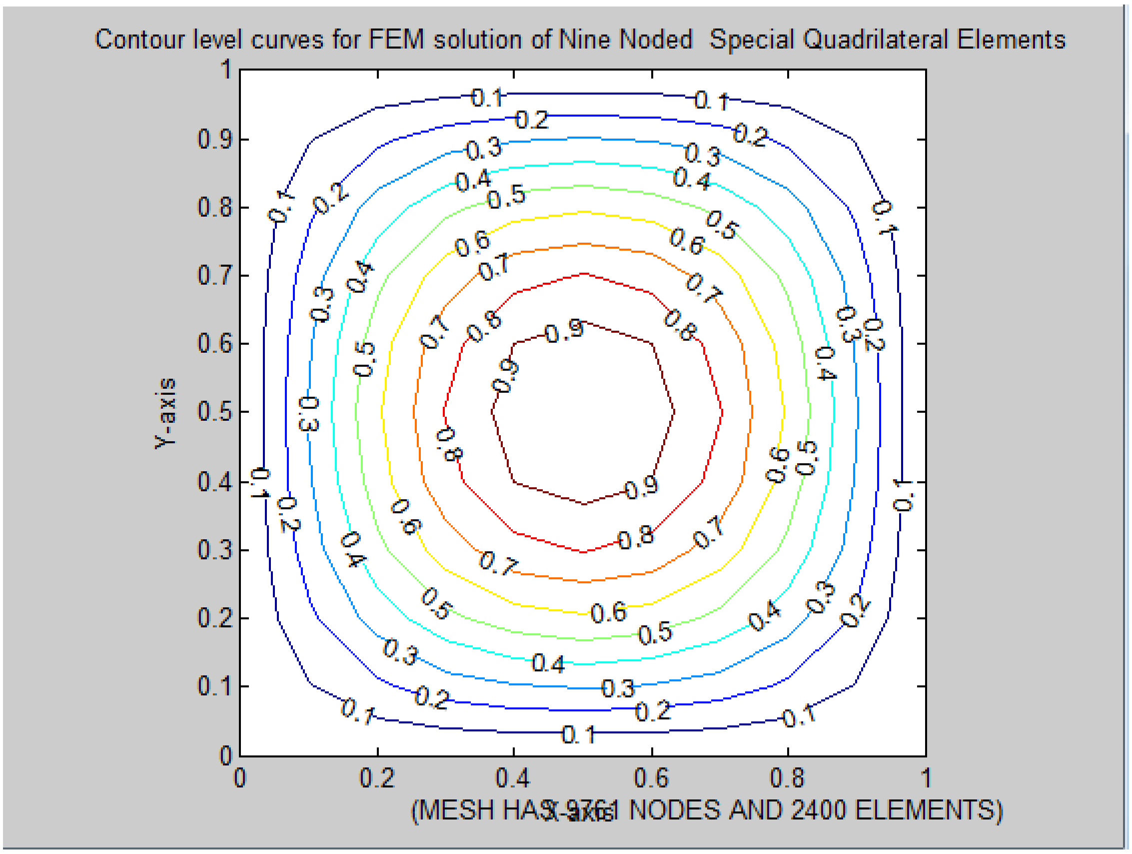

Contour level curves for FEM solution of Nine Noded Special Quadrilateral Elements
Y-axis
X-axis
(MESH HAS 2481 NODES AND 600 ELEMENTS)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

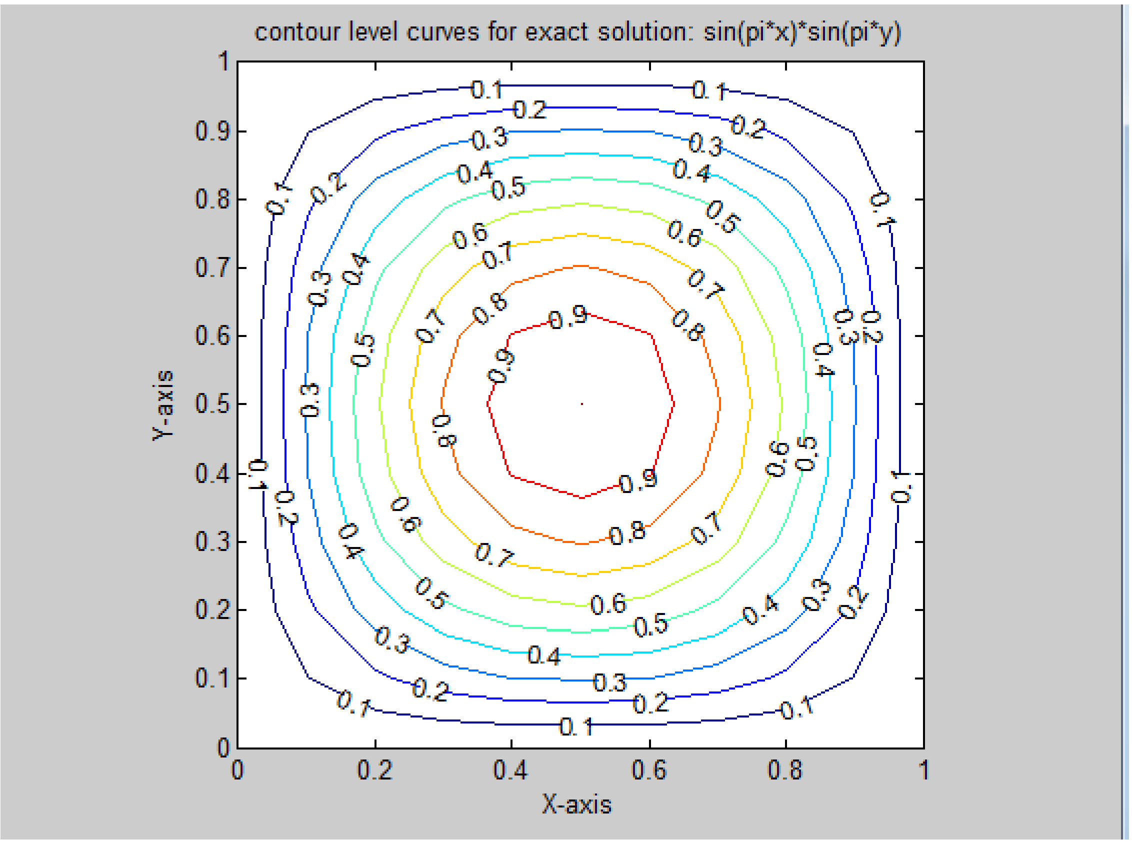

contour level curves for exact solution: sin(pi*x)*sin(pi*y)
Y-axis
X-axis
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

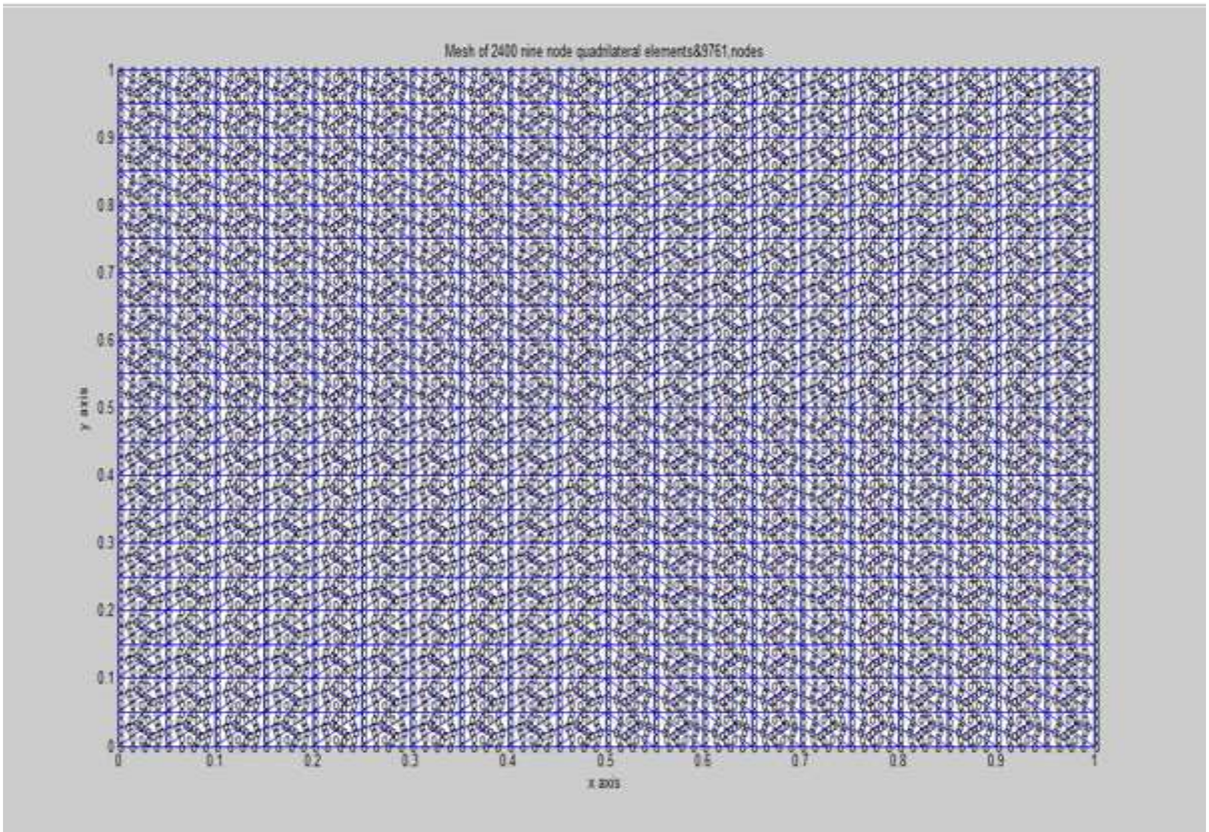


Mesh No.2

Mesh with 2400 nine noded quadrilateral elements & no. of nodes = 9761


Contour level curves for FEM solution of Nine Noded Special Quadrilateral Elements
Y-axis
X-axis
(MESH HAS 9761 NODES AND 2400 ELEMENTS)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1


contour level curves for exact solution: sin(pi*x)*sin(pi*y)
Y-axis
X-axis
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.2
0.4
0.6
0.8
1



MATLAB Codes-I

(1)quadrilateralmesh_over_arbitrarytriangle_q9LGautomeshgen.m
(2)D2LaplaceEquationQ9Ex3automeshgenNew.m
(3)nodaladdresses_special_convex_quadrilaterals_2nd_orderLG.m
(4)coordinate_arbitrarytriangle_2ndorderLAGR.m
(5)coordinate_special_quadrilaterals_in_stdtriangle_2nd_orderLAGR.m
(6)nodaladdresses4Lagrangespecial_convex_quadrilaterals_2nd_order.m
(7)D2LaplaceEquationQ9Ex3automeshgenNewPolygon.m

%(1)quadrilateralmesh_over_arbitrarytriangle_q9LGautomeshgen.m

```
function[]=quadrilateralmesh_over_arbitrarytriangle_q9LGautomeshgen(mmesh,nmesh,tri)
%quadrilateralmesh_over_arbitrarytriangle_q8automeshgen(mmesh,nmesh,tri)
%quadrilateralmesh_over_arbitrarytriangle_q9LGautomeshgen(1,1,1)
%quadrilateralmesh_over_arbitrarytriangle_q9LGautomeshgen(1,1,2)
%quadrilateralmesh_over_arbitrarytriangle_q9LGautomeshgen(1,1,3)
%quadrilateralmesh_over_arbitrarytriangle_q9LGautomeshgen(1,1,4)
clf
switch tri
case 1%standard triangle
xx=sym([0;1;0])
yy=sym([0;0;1])
case 2
 xx=sym([0;1/2;1/2])
 yy=sym([0;0;1/2])

case 3%equilateral triangle

xx=sym([0;1;1/2])
yy=sym([0;0;sqrt(3)/2])
    case 4%equilateral triangle
 xx=sym([-sqrt(3);sqrt(3); 0])
 yy=sym([     -1;      -1; 2])


end
for mesh=mmesh:nmesh
    figure(mesh)
```

```
    ndiv=2*mesh;

[eln,nodetel,nodes,nnode]=nodaladdresses_special_convex_quadrilaterals_2nd_orderLG(ndiv
);

%[coord,gcoord]=coordinate_rtisoscelestriangle00_h0_hh_2ndorder(ndiv);
[coord,gcoord]=coordinate_arbitrarytriangle_2ndorderLAGR(xx,yy,ndiv)

[nel,nnel]=size(nodes)

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

table1=[NN nodes]

[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
switch tri

case 1
axis tight
xmin=0;xmax=1;ymin=0;ymax=1;
axis([xmin,xmax,ymin,ymax]);
%place element number
midx=mean(xvec(1,1:4));
midy=mean(yvec(1,1:4));
case 2
axis tight
xmin=0;xmax=1/2;ymin=0;ymax=1/2;
axis([xmin,xmax,ymin,ymax]);
%place element number
midx=mean(xvec(1,1:4));
midy=mean(yvec(1,1:4));
case 3
axis tight
xmin=0;xmax=1;ymin=0;ymax=1;
axis([xmin,xmax,ymin,ymax]);
end
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)&(tri~=2)
text(midx-.1,midy+.05,['\bf(',num2str(i),'\bf)']);
end
if (mesh<=2)&(tri==2)
text(midx-.005,midy+.02,['\bf(',num2str(i),'\bf)']);
end
end;%i loop
```

```
xlabel('\bfx axis')
ylabel('\bfy axis')

switch tri
case 1
st1='\bfstandard triangle  ';
st2=' using  ';
st3='9-node ';
st4='quadriateral';
st5=' elements'
title([st1,st2,st3,st4,st5])
text(.6,.9,['\bfMESH NO.=',num2str(mesh)])
text(.6,.8,['\bfnumber of elements=',num2str(nel)])
text(.6,.7,['\bfnumber of nodes=',num2str(nnode)])

case 2
st1='\bfone eigth (1/8)square cross section ';
st2=' using  ';
st3='9-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)])
case 3
st1='\bfequilateral triangle  ';
st2=' using  ';
st3='9-node parabolic ';
st4='quadriateral';
st5=' elements'
title([st1,st2,st3,st4,st5])
text(0.6,0.8,['\bfMESH NO.=',num2str(mesh)])
text(0.6,0.75,['\bfnumber of elements=',num2str(nel)])
text(0.6,0.70,['\bfnumber of nodes=',num2str(nnode)])

    case 4
st1='\bfequilateral triangle  ';
st2=' using  ';
st3='9-node parabolic ';
st4='quadriateral';
st5=' elements'
title([st1,st2,st3,st4,st5])
text(1,1.8,['\bfMESH NO.=',num2str(mesh)])
text(1,1.6,['\bfnumber of elements=',num2str(nel)])
text(1,1.4,['\bfnumber of nodes=',num2str(nnode)])




end

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

% (2)D2LaplaceEquationQ9Ex3automeshgenNew.m

```
function[]=D2LaplaceEquationQ9Ex3automeshgenNew(n1,n2,n3,numtri,ndiv,mesh)
%
%D2LaplaceEquationQ9Ex3automeshgenNew(1,2,3,1,2,1)
%*************************************************************************
syms coord
syms x y
ndof=1;


switch mesh
    case 1
    x=sym([0;1/2;1/2])
    y=sym([0;0;1/2])
    case 2  %isoscles triangle(torsion of an equilateral triangle,each side=2*sqrt(3))
 x=sym([-sqrt(3);sqrt(3); 0])
 y=sym([     -1;      -1; 2])


end
syms ui vi wi xi yi
%[ui,vi,wi]=coordinate_special_quadrilaterals_in_stdtriangle_2nd_order(ndiv);
[ui,vi,wi]=coordinate_special_quadrilaterals_in_stdtriangle_2nd_orderLAGR(ndiv)
%disp([ui vi wi])
N=length(ui);
    NN=(1:N)';
    x
    y
    x1=x(n1,1);x2=x(n2,1);x3=x(n3,1);y1=y(n1,1);y2=y(n2,1);y3=y(n3,1);
for i=1:N
    xxi(i,1)=x1+(x2-x1)*ui(i,1)+(x3-x1)*vi(i,1);
    yyi(i,1)=y1+(y2-y1)*ui(i,1)+(y3-y1)*vi(i,1);
end
%disp('_________________________________________________')
%disp('NN   xi    yi')
%disp([NN xi yi])
%disp('_________________________________________________')
coord(:,1)=(xxi(:,1));
coord(:,2)=(yyi(:,1));
gcoord(:,1)=double(xxi(:,1));
gcoord(:,2)=double(yyi(:,1));
%disp(gcoord);
%[eln,nodetel,nodes,nnode]=nodaladdresses_special_convex_quadrilaterals_2nd_order(ndiv)
;
[eln,nodetel,nodes,nnode]=nodaladdresses4Lagrangespecial_convex_quadrilaterals_2nd_orde
r(ndiv);




%*********************************************************************
%syms coord
%ndof=1;

%[eln,nodetel,nodes,nnode]=nodaladdresses4Lagrangespecial_convex_quadrilaterals_2nd_ord
er(ndiv);
%[coord,gcoord]=coordinate_rtisoscelestriangle00_h0_hh_2ndorderLAGR(ndiv);

    %[coord,gcoord]=coordinate_rtisoscelestriangle00_h0_hh(ndiv);
    %[nodetel,nodes]=nodaladdresses4special_convex_quadrilaterals(ndiv)
[nel,nnel]=size(nodes);
%disp([nel nnode nnel ndof])
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
%radius of the hole=1.25cm
%input data for nodal coordinate values
%gcoord(i,j),where i->node no. and j->x or y

%
table1=[N nodes]
[nel,nnel]=size(nodes);
%************************************
switch mesh
    case 1
    nnn=0;
    for nn=1:nnode
        if gcoord(nn,1)==(1/2)
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
        end
    end
    format long g
k1 =double(0.1405770149551555103784039602O329);
 xi=(zeros(nnode,1));
a0=8/pi^3;
for m=1:nnode
    gx=(gcoord(m,1));gy=(gcoord(m,2));rr=(0);
for n=1:2:99
rr=rr+(-1)^((n-1)/2)*(1-(cosh(n*pi*gy)/cosh(n*pi/2)))*cos(n*pi*gx)/n^3;
end
xi(m,1)=(a0*rr);
end
 mm=length(bcdof);

    case 2%torsion of an equilateral triangle

 nnn=0;
 %boundary conditions on side 1
  for nn=1:nnode
      xnn=gcoord(nn,1);ynn=gcoord(nn,2);
        if ((ynn+1)<1.e-5)
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
  end
%boundary conditions on side 2
  for nn=1:nnode
      xnn=gcoord(nn,1);ynn=gcoord(nn,2);
        if ((-(sqrt(3))*xnn-ynn+2)<1.e-5)
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
```

```
  end
%boundary conditions on side 3
  for nn=1:nnode
      xnn=gcoord(nn,1);ynn=gcoord(nn,2);
        if (((sqrt(3))*xnn-ynn+2)<1.e-5)
            nnn=nnn+1
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
  end
  bcdof
  bcval
  mm=length(bcdof);
 for m=1:nnode
    gx=(gcoord(m,1));gy=(gcoord(m,2));
    xi(m,1)=((gy+1)*((sqrt(3))*gx-gy+2)*(-(sqrt(3))*gx-gy+2))/12;
  end
xi=double(xi);
format long g
k1 =9*sqrt(3)/5;

end%switch

%***********************************


 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('  .688003696999291704075192889')) vpa(sym(' .630268637329029196546930317 48'));...
vpa(sym(' .63026863732902919654693031748')) vpa(sym('  .688003696999291704075192889'))];

intJdn1dn2uvrs =[vpa(sym('  .4946976667946543132242312849e-1')) vpa(sym(' -.9249249979013890567437258873e-1'));...
vpa(sym('  .7417416687652776099229407793e-1')) vpa(sym(' -.5804150224125689851696121830e-1'))];

intJdn1dn3uvrs =[vpa(sym(' -.2763918707186419647824461270e-1')) vpa(sym(' -.3640622770513571608466053928e-1'));...
vpa(sym(' -.3640622770513571608466053928e-1')) vpa(sym(' -.2763918707186419647824461270e-1'))];

intJdn1dn4uvrs =[vpa(sym(' -.5804150224125689851696121830e-1')) vpa(sym('  .7417416687652776099229407793e-1'));...
vpa(sym(' -.9249249979013890567437258873e-1')) vpa(sym('  .4946976667946543132242312849e-1'))];

intJdn1dn5uvrs =[vpa(sym(' -.420340201552488982561479599 3')) vpa(sym('  .238355345065568618361178470 3'));...
vpa(sym(' -.4283113216010980483054881964')) vpa(sym('  .1020378129397056233087080320'))];

intJdn1dn6uvrs =[vpa(sym(' .1262687000516458834074677285 3')) vpa(sym(' .1686516911320240413733489215 6'));...
vpa(sym(' .16865169113202404137334892156')) vpa(sym(' .15224655673422927054938212414'))];

intJdn1dn7uvrs =[vpa(sym(' .15224655673422927054938212414')) vpa(sym(' .16865169113202404137334892156'));...
vpa(sym(' .16865169113202404137334892156')) vpa(sym(' .12626870005164588340746772853'))];

intJdn1dn8uvrs =[vpa(sym('  .1020378129397056233087080320')) vpa(sym(' -.4283113216010980483054881964'));...
vpa(sym('  .2383553450655686183611784703')) vpa(sym(' -.4203402015524889825614795993'))];

intJdn1dn9uvrs =[vpa(sym(' -.6120056425393653014388148716')) vpa(sym(' -.7228914824388009885825793847'));...
vpa(sym(' -.7228914824388009885825793847')) vpa(sym(' -.6120056425393653014388148716'))];

intJdn2dn1uvrs =[vpa(sym('  .4946976667946543132242312849e-1')) vpa(sym('  .7417416687652776099229407793e-1'));...
vpa(sym(' -.9249249979013890567437258873e-1')) vpa(sym(' -.5804150224125689851696121830e-1'))];

intJdn2dn2uvrs =[vpa(sym('  .2482224272357035985941591016 5')) vpa(sym(' -.13943577781062161645243966287'));...
vpa(sym(' -.13943577781062161645243966287')) vpa(sym('  .22525137401606776447455758469'))];

intJdn2dn3uvrs =[vpa(sym('  -.9368329064821876846205153869e-1')) vpa(sym(' -.8285228368179732406596827984 6e-1'));...
vpa(sym('  .8381438298486934260069838682 0e-1')) vpa(sym('   .3790480089812643647503258264e-1'))];

intJdn2dn4uvrs =[vpa(sym('   .6292570002530214045175196 0e-3')) vpa(sym(' .2610005204712297985144922346e-1'));...
vpa(sym(' .2610005204712297985144922346e-1')) vpa(sym('   .6292570002530214045175196 0e-3'))];

intJdn2dn5uvrs =[vpa(sym('   -.4686821675279013233513591962')) vpa(sym('  -.3464619445210953430278280635 8'));...
vpa(sym('   .3202047221455713236388386030 9')) vpa(sym(' .8583672920954138128722924288e-1'))];

intJdn2dn6uvrs =[vpa(sym('   .2048671103020189876458458532')) vpa(sym('  .2571877444243557310522163472 2'));...
vpa(sym(' -.4094789222423109356144503194 5')) vpa(sym(' -.2807386267765162146339380350 2'))];
```

```
intJdn2dn7uvrs =[vpa(sym(' .508859176419664984628888 6644e-1')) vpa(sym('  -.10335794828229434487750587371'));...
vpa(sym('  -.10335794828229434487750587371')) vpa(sym(' -.17609016019421764322513358 0e-2'))];
intJdn2dn8uvrs =[vpa(sym(' -.41736856619169902011341267 58e-1')) vpa(sym(' -.94765689307712237483661393 07e-1'));...
vpa(sym(' -.9476568930771223748366139307e-1')) vpa(sym('  .33881015844374607003400451 61e-1'))];
intJdn2dn9uvrs =[vpa(sym('   .5002783593588245639491753 32e-1')) vpa(sym('    .40941168025551439401144362445'));...
vpa(sym('    .40941168025551439401144362445')) vpa(sym(' -.4296214634864792106158679226e-1'))];
intJdn3dn1uvrs =[vpa(sym(' -.2763918707186419647824461270e-1')) vpa(sym(' -.3640622770513571608466053928e-1'));...
vpa(sym(' -.3640622770513571608466053928e-1')) vpa(sym(' -.2763918707186419647824461270e-1'))];
intJdn3dn2uvrs =[vpa(sym('  -.9368329064821876846205153869e-1')) vpa(sym('  .8381438298486934260069838682 0e-1'));...
vpa(sym(' -.8285228368179732406596827984 6e-1')) vpa(sym('   .3790480089812643647503258264e-1'))];
intJdn3dn3uvrs =[vpa(sym('  .33203856905137126709551224250')) vpa(sym(' .29812794600013651335576030277 1'));...
vpa(sym(' .29812794600013651335576030277 1')) vpa(sym('  .33203856905137126709551224250'))];
intJdn3dn4uvrs =[vpa(sym('   .3790480089812643647503258264e-1')) vpa(sym(' -.8285228368179732406596827984 6e-1'));...
vpa(sym('  .8381438298486934260069838682 0e-1')) vpa(sym('  -.9368329064821876846205153869e-1'))];
intJdn3dn5uvrs =[vpa(sym('   .15710844821544667058002407565')) vpa(sym('   .12945549547911074908308207668'));...
vpa(sym('   .12945549547911074908308207668')) vpa(sym(' .4305964422926219650773122697e-1'))];
intJdn3dn6uvrs =[vpa(sym('  .14159217838663875388665351332')) vpa(sym(' -.35360191630747700837476451283'));...
vpa(sym('  .31306475035918965829190215383')) vpa(sym(' -.34086853405320051524855521401'))];
intJdn3dn7uvrs =[vpa(sym(' -.34086853405320051524855521401')) vpa(sym('  .31306475035918965829190215383'));...
vpa(sym(' -.35360191630747700837476451283')) vpa(sym('  .14159217838663875388665351332'))];
intJdn3dn8uvrs =[vpa(sym(' .4305964422926219650773122697e-1')) vpa(sym('   .12945549547911074908308207668'));...
vpa(sym('   .12945549547911074908308207668')) vpa(sym('   .15710844821544667058002407565'))];
intJdn3dn9uvrs =[vpa(sym('  -.24951262900756184435610227 57')) vpa(sym(' -.48105764260800696388913166479'));...
vpa(sym(' -.48105764260800696388913166479')) vpa(sym('  -.24951262900756184435610227 57'))];
intJdn4dn1uvrs =[vpa(sym(' -.5804150224125689851696121830e-1')) vpa(sym(' -.9249249979013890567437258873e-1'));...
vpa(sym('  .7417416687652776099229407793e-1')) vpa(sym('  .4946976667946543132242312849e-1'))];
intJdn4dn2uvrs =[vpa(sym('   .6292570002530214045175196 0e-3')) vpa(sym(' .2610005204712297985144922346e-1'));...
vpa(sym(' .2610005204712297985144922346e-1')) vpa(sym('   .6292570002530214045175196 0e-3'))];
intJdn4dn3uvrs =[vpa(sym('   .3790480089812643647503258264e-1')) vpa(sym('  .8381438298486934260069838682 0e-1'));...
vpa(sym(' -.8285228368179732406596827984 6e-1')) vpa(sym('  -.9368329064821876846205153869e-1'))];
intJdn4dn4uvrs =[vpa(sym('  .22525137401606776447455758469')) vpa(sym(' -.13943577781062161645243966287'));...
vpa(sym(' -.13943577781062161645243966287')) vpa(sym('  .24822242723570359859415910165'))];
intJdn4dn5uvrs =[vpa(sym('  .33881015844374607003400451 61e-1')) vpa(sym(' -.94765689307712237483661393 07e-1'));...
vpa(sym(' -.9476568930771223748366139307e-1')) vpa(sym(' -.41736856619169902011341267 58e-1'))];
intJdn4dn6uvrs =[vpa(sym(' -.17609016019421764322513358 0e-2')) vpa(sym('  -.10335794828229434487750587371'));...
vpa(sym('  -.10335794828229434487750587371')) vpa(sym(' .508859176419664984628888 6644e-1'))];
intJdn4dn7uvrs =[vpa(sym(' -.28073862677651621463393803502')) vpa(sym(' -.40947892224231093561445031945'));...
vpa(sym('  .25718774442435573105221634722')) vpa(sym('   .20486711030201898764584585 32'))];
intJdn4dn8uvrs =[vpa(sym(' .8583672920954138128722924288e-1')) vpa(sym('   .32020472214557132363883860309'));...
vpa(sym('  -.34646194452109534302782806358')) vpa(sym('   -.46868216752790132335135919 62'))];
intJdn4dn9uvrs =[vpa(sym(' -.4296214634864792106158679226e-1')) vpa(sym('    .40941168025551439401144362445'));...
vpa(sym('    .40941168025551439401144362445')) vpa(sym('   .5002783593588245639491753 32e-1'))];
intJdn5dn1uvrs =[vpa(sym(' -.42034020155248898256147959 93')) vpa(sym(' -.42831132160109804830548819 64'));...
vpa(sym('  .23835534506556861836117847 03')) vpa(sym('  .10203781293970562330870803 20'))];
intJdn5dn2uvrs =[vpa(sym('   -.46868216752790132335135919 62')) vpa(sym('   .32020472214557132363883860309'));...
vpa(sym('  -.34646194452109534302782806358')) vpa(sym(' .8583672920954138128722924288e-1'))];
intJdn5dn3uvrs =[vpa(sym('   .15710844821544667058002407565')) vpa(sym('   .12945549547911074908308207668'));...
vpa(sym('   .12945549547911074908308207668')) vpa(sym(' .4305964422926219650773122697e-1'))];
intJdn5dn4uvrs =[vpa(sym('  .33881015844374607003400451 61e-1')) vpa(sym(' -.94765689307712237483661393 07e-1'));...
vpa(sym(' -.9476568930771223748366139307e-1')) vpa(sym(' -.41736856619169902011341267 58e-1'))];
intJdn5dn5uvrs =[vpa(sym(' 1.06633848333445648567503766 51')) vpa(sym('  .56556859863119073241392390 19'));...
vpa(sym('  .56556859863119073241392390 19')) vpa(sym(' 1.12113332953297566519504242 78'))];
intJdn5dn6uvrs =[vpa(sym('  -.58306188548401601151324941 69')) vpa(sym(' -.54252818687130357695788942932'));...
vpa(sym(' -.54252818687130357695788942932')) vpa(sym(' -.22579554943131879593427777753'))];
intJdn5dn7uvrs =[vpa(sym('   -.16767681050348731002396257 42')) vpa(sym(' -.1709647811328463063306097 92e-2'));...
vpa(sym(' -.1709647811328463063306097 92e-2')) vpa(sym('   .17132457681637017904834081102'))];
intJdn5dn8uvrs =[vpa(sym(' .9027148605382036792067597 12e-1')) vpa(sym('   .41024643803333829631989611 84'));...
vpa(sym('   .41024643803333829631989611 84')) vpa(sym(' .9027148605382036792067597 12e-1'))];
intJdn5dn9uvrs =[vpa(sym('   .29216163161979549627091262 30')) vpa(sym('  -.35816040869776877564539558 36'));...
vpa(sym('  -.35816040869776877564539558 36')) vpa(sym(' -1.34613117273118671532210866 66'))];
intJdn6dn1uvrs =[vpa(sym(' .12626870005164588340746772853')) vpa(sym(' .16865169113202404137334892156'));...
vpa(sym(' .16865169113202404137334892156')) vpa(sym(' .15224655673422927054938212414'))];
intJdn6dn2uvrs =[vpa(sym('   .20486711030201898764584585 32')) vpa(sym(' -.40947892224231093561445031945'));...
vpa(sym('  .25718774442435573105221634722')) vpa(sym(' -.28073862677651621463393803502'))];
intJdn6dn3uvrs =[vpa(sym('  .14159217838663875388665351332')) vpa(sym('  .31306475035918965829190215383'));...
vpa(sym(' -.35360191630747700837476451283')) vpa(sym(' -.34086853405320051524855521401'))];
intJdn6dn4uvrs =[vpa(sym(' -.17609016019421764322513358 0e-2')) vpa(sym('  -.10335794828229434487750587371'));...
vpa(sym('  -.10335794828229434487750587371')) vpa(sym(' .508859176419664984628888 6644e-1'))];
intJdn6dn5uvrs =[vpa(sym('  -.58306188548401601151324941 69')) vpa(sym(' -.54252818687130357695788942932'));...
vpa(sym(' -.54252818687130357695788942932')) vpa(sym(' -.22579554943131879593427777753'))];
intJdn6dn6uvrs =[vpa(sym(' 1.59985108580953085463759282 28')) vpa(sym(' .24469327225090109904911663174'));...
vpa(sym(' .24469327225090109904911663174')) vpa(sym('  .64675514174853057100283264 58'))];
intJdn6dn7uvrs =[vpa(sym(' -.7461268601626634110113636 05e-1')) vpa(sym('   .42496611860422613701397774221'));...
vpa(sym('   .42496611860422613701397774221')) vpa(sym(' -.7461268601626634110113636 05e-1'))];
intJdn6dn8uvrs =[vpa(sym('   .17132457681637017904834081102')) vpa(sym(' -.1709647811328463063306097 92e-2'));...
vpa(sym(' -.1709647811328463063306097 92e-2')) vpa(sym('   -.16767681050348731002396257 42'))];
intJdn6dn9uvrs =[vpa(sym('   -1.58446817826398012957926361 56')) vpa(sym(' -.9430112713910361521519372896e-1'));...
```

```
vpa(sym(' -.943011271391036152151937289 6e-1')) vpa(sym('    .23980459065606283692676632 48'))];
```

```
vpa(sym(' -.9430112713910361521519372896e-1')) vpa(sym('    .2398045906560628369267663248'))];
intJdn7dn1uvrs =[vpa(sym(' .1522465567342292705493821241 4')) vpa(sym(' .1686516911320240413733489215 6'));...
vpa(sym(' .1686516911320240413733489215 6')) vpa(sym(' .1262687000516458834074677285 3'))];
intJdn7dn2uvrs =[vpa(sym(' .5088591764196649846288886644e-1')) vpa(sym('  -.1033579482822943448775058737 1'));...
vpa(sym('  -.1033579482822943448775058737 1')) vpa(sym(' -.1760901601942176432251335 80e-2'))];
intJdn7dn3uvrs =[vpa(sym(' -.3408685340532005152485552140 1')) vpa(sym(' -.3536019163074770083747645128 3'));...
vpa(sym('  .3130647503591896582919021538 3')) vpa(sym('  .1415921783866387538866535133 2'))];
intJdn7dn4uvrs =[vpa(sym(' -.2807386267765162146339380350 2')) vpa(sym('  .2571877444243557310522163472 2'));...
vpa(sym(' -.4094789222423109356144503194 5')) vpa(sym('   .2048671103020189876458458532'))];
intJdn7dn5uvrs =[vpa(sym('   -.1676768105034873100239625742')) vpa(sym(' -.1709647811328463063306097 92e-2'));...
vpa(sym(' -.1709647811328463063306097 92e-2')) vpa(sym('   .1713245768163701790483408110 2'))];
intJdn7dn6uvrs =[vpa(sym(' -.7461268601626634110113636 05e-1')) vpa(sym('   .4249661186042261370139777422 1'));...
vpa(sym('   .4249661186042261370139777422 1')) vpa(sym(' -.7461268601626634110113636 05e-1'))];
intJdn7dn7uvrs =[vpa(sym('  .6467551417485305710028326458')) vpa(sym(' .2446932722509010990491166317 4'));...
vpa(sym(' .2446932722509010990491166317 4')) vpa(sym(' 1.599851085809530854637592822 8'))];
intJdn7dn8uvrs =[vpa(sym(' -.2257955494313187959342777775 3')) vpa(sym(' -.5425281868713035769578894293 2'));...
vpa(sym(' -.5425281868713035769578894293 2')) vpa(sym('  -.5830618854840160115132494169'))];
intJdn7dn9uvrs =[vpa(sym('    .2398045906560628369267663248')) vpa(sym(' -.9430112713910361521519372896e-1'));...
vpa(sym(' -.9430112713910361521519372896e-1')) vpa(sym('   -1.584468178263980129579263615 6'))];
intJdn8dn1uvrs =[vpa(sym('  .1020378129397056233087080320')) vpa(sym('  .2383553450655686183611784703'));...
vpa(sym(' -.4283113216010980483054881964')) vpa(sym(' -.4203402015524889825614795993'))];
intJdn8dn2uvrs =[vpa(sym(' -.4173685661916990201134126758e-1')) vpa(sym(' -.9476568930771223748366139307e-1'));...
vpa(sym(' -.9476568930771223748366139307e-1')) vpa(sym('  .3388101584437460700340045161e-1'))];
intJdn8dn3uvrs =[vpa(sym(' .4305964422926219650773122697e-1')) vpa(sym('   .1294554954791107490830820766 8'));...
vpa(sym('   .1294554954791107490830820766 8')) vpa(sym('   .1571084482154466705800240756 5'))];
intJdn8dn4uvrs =[vpa(sym(' .8583672920954138128722924288e-1')) vpa(sym('  -.3464619445210953430278280635 8'));...
vpa(sym('   .3202047221455713236388386030 9')) vpa(sym('   -.4686821675279013233513591962'))];
intJdn8dn5uvrs =[vpa(sym(' .9027148605382036792067597 12e-1')) vpa(sym('   .4102464380333382963198961184'));...
vpa(sym('   .4102464380333382963198961184')) vpa(sym(' .9027148605382036792067597 12e-1'))];
intJdn8dn6uvrs =[vpa(sym('   .1713245768163701790483408110 2')) vpa(sym(' -.1709647811328463063306097 92e-2'));...
vpa(sym(' -.1709647811328463063306097 92e-2')) vpa(sym('   -.1676768105034873100239625742'))];
intJdn8dn7uvrs =[vpa(sym(' -.2257955494313187959342777775 3')) vpa(sym(' -.5425281868713035769578894293 2'));...
vpa(sym(' -.5425281868713035769578894293 2')) vpa(sym('  -.5830618854840160115132494169'))];
intJdn8dn8uvrs =[vpa(sym(' 1.121133329532975665195042427 8')) vpa(sym('  .5655685986311907324139239019'));...
vpa(sym('  .5655685986311907324139239019')) vpa(sym(' 1.066338483334456485675037665 1'))];
intJdn8dn9uvrs =[vpa(sym(' -1.346131172731186715322108666 6')) vpa(sym('  -.3581604086977687756453955836'));...
vpa(sym('  -.3581604086977687756453955836')) vpa(sym('   .2921616316197954962709126230'))];
intJdn9dn1uvrs =[vpa(sym(' -.6120056425393653014388148716')) vpa(sym(' -.7228914824388009885825793847'));...
vpa(sym(' -.7228914824388009885825793847')) vpa(sym(' -.6120056425393653014388148716'))];
intJdn9dn2uvrs =[vpa(sym('   .5002783593588245639491753 32e-1')) vpa(sym('    .4094116802555143940114436244 5'));...
vpa(sym('    .4094116802555143940114436244 5')) vpa(sym(' -.4296214634864792106158679226e-1'))];
intJdn9dn3uvrs =[vpa(sym('  -.2495126290075618443561022757')) vpa(sym(' -.4810576426080069638891316647 9'));...
vpa(sym(' -.4810576426080069638891316647 9')) vpa(sym('  -.2495126290075618443561022757'))];
intJdn9dn4uvrs =[vpa(sym(' -.4296214634864792106158679226e-1')) vpa(sym('    .4094116802555143940114436244 5'));...
vpa(sym('    .4094116802555143940114436244 5')) vpa(sym('   .5002783593588245639491753 32e-1'))];
intJdn9dn5uvrs =[vpa(sym('   .2921616316197954962709126230')) vpa(sym('  -.3581604086977687756453955836'));...
vpa(sym('  -.3581604086977687756453955836')) vpa(sym(' -1.346131172731186715322108666 6'))];
intJdn9dn6uvrs =[vpa(sym('   -1.584468178263980129579263615 6')) vpa(sym(' -.9430112713910361521519372896e-1'));...
vpa(sym(' -.9430112713910361521519372896e-1')) vpa(sym('    .2398045906560628369267663248'))];
intJdn9dn7uvrs =[vpa(sym('    .2398045906560628369267663248')) vpa(sym(' -.9430112713910361521519372896e-1'));...
vpa(sym(' -.9430112713910361521519372896e-1')) vpa(sym('   -1.584468178263980129579263615 6'))];
intJdn9dn8uvrs =[vpa(sym(' -1.346131172731186715322108666 6')) vpa(sym('  -.3581604086977687756453955836'));...
vpa(sym('  -.3581604086977687756453955836')) vpa(sym('   .2921616316197954962709126230'))];
intJdn9dn9uvrs =[vpa(sym(' 3.253085710679001122165279740 6')) vpa(sym(' 1.290048836209523946170002425 7'));...
vpa(sym(' 1.290048836209523946170002425 7')) vpa(sym(' 3.253085710679001122165279740 6'))];
%integrals of products of gl0bal derivatives
intJdndn=[intJdn1dn1uvrs intJdn1dn2uvrs intJdn1dn3uvrs intJdn1dn4uvrs intJdn1dn5uvrs intJdn1dn6uvrs intJdn1dn7uvrs intJdn1dn8uvrs intJdn1dn9uvrs ;...
          intJdn2dn1uvrs intJdn2dn2uvrs intJdn2dn3uvrs intJdn2dn4uvrs intJdn2dn5uvrs intJdn2dn6uvrs intJdn2dn7uvrs intJdn2dn8uvrs intJdn2dn9uvrs;...
          intJdn3dn1uvrs intJdn3dn2uvrs intJdn3dn3uvrs intJdn3dn4uvrs intJdn3dn5uvrs intJdn3dn6uvrs intJdn3dn7uvrs intJdn3dn8uvrs intJdn3dn9uvrs;...
          intJdn4dn1uvrs intJdn4dn2uvrs intJdn4dn3uvrs intJdn4dn4uvrs intJdn4dn5uvrs intJdn4dn6uvrs intJdn4dn7uvrs intJdn4dn8uvrs intJdn4dn9uvrs;...
          intJdn5dn1uvrs intJdn5dn2uvrs intJdn5dn3uvrs intJdn5dn4uvrs intJdn5dn5uvrs intJdn5dn6uvrs intJdn5dn7uvrs intJdn5dn8uvrs intJdn5dn9uvrs;...
          intJdn6dn1uvrs intJdn6dn2uvrs intJdn6dn3uvrs intJdn6dn4uvrs intJdn6dn5uvrs intJdn6dn6uvrs intJdn6dn7uvrs intJdn6dn8uvrs intJdn6dn9uvrs;...
          intJdn7dn1uvrs intJdn7dn2uvrs intJdn7dn3uvrs intJdn7dn4uvrs intJdn7dn5uvrs intJdn7dn6uvrs intJdn7dn7uvrs intJdn7dn8uvrs intJdn7dn9uvrs;...
          intJdn8dn1uvrs intJdn8dn2uvrs intJdn8dn3uvrs intJdn8dn4uvrs intJdn8dn5uvrs intJdn8dn6uvrs intJdn8dn7uvrs intJdn8dn8uvrs intJdn8dn9uvrs;...
          intJdn9dn1uvrs intJdn9dn2uvrs intJdn9dn3uvrs intJdn9dn4uvrs intJdn9dn5uvrs intJdn9dn6uvrs intJdn9dn7uvrs intJdn9dn8uvrs intJdn9dn9uvrs];


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(1:9,1:9)=(zeros(9,9));
for i=1:9
 for j=i:9
 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 =[5/144;1/24;7/144;1/24]*(2*delabc);
%f=[ -7/432;  -1/72; -5/432;  -1/72; 11/216; 13/216; 13/216; 11/216]*(2*delabc)
f=[ 1/432; 1/216; 1/144; 1/216;  1/72; 5/216; 5/216;  1/72;  2/27]*(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
 %-------------------------------------------------------------------------
%bcdof=[13;37;35;33;31;29;27;25;23;21;19;17;15];
for ii=1:mm
    kk=bcdof(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=bcdof(ii,1);
    ss(kk,kk)=1;
end
phi=ss\ff;
%
phi=double(phi);
if mesh==2
    phi=phi/2;
end
[phi xi]
for I=1:nnode
NN(I,1)=I;
phi_xi(I,1)=phi(I,1)-xi(I,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])
t=0;
 for iii=1:nnode
    t=t+phi(iii,1)*ff(iii,1);
 end
t=0;
 for iii=1:nnode
    t=t+phi(iii,1)*ff(iii,1);
 end
switch mesh
    case 1
T=8*t;
    case 2
        T=2*t;
end

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('---------------------------------------------------------------------------------
------------')

%###################################################################################
if (mesh==2)

[x,y]=meshgrid(-sqrt(3):(1/15)*sqrt(3):sqrt(3),-1:(0.1):2);
 z=(zeros(31,31));
for i=1:31
    for j=1:31
      for iel=1:nel
 %node numbers of quadrilateral
       nd1=nodes(iel,1);nd2=nodes(iel,2);nd3=nodes(iel,3);nd4=nodes(iel,4);

nd5=nodes(iel,5);nd6=nodes(iel,6);nd7=nodes(iel,7);nd8=nodes(iel,8);nd9=nodes(iel,9);
 %coordinates of quadrilateral(u,v)

u(1,1)=gcoord(nd1,1);u(2,1)=gcoord(nd2,1);u(3,1)=gcoord(nd3,1);u(4,1)=gcoord(nd4,1);

v(1,1)=gcoord(nd1,2);v(2,1)=gcoord(nd2,2);v(3,1)=gcoord(nd3,2);v(4,1)=gcoord(nd4,2);
 %coordinates of the grid(x,y)
```

```
          in=inpolygon(x(i,j),y(i,j),u,v);
          if  (in==1)
           X=x(i,j);Y=y(i,j);
           [t]=convexquadrilateral_coordinates(u,v,X,Y);
           r=t(1,1);
           s=t(2,1);
%======================================================================================
========================================
%
%======================================================================================
==================================================
 shn1=(r^2-r)*(s^2-s)/4;
           shn2=(r^2+r)*(s^2-s)/4;
           shn3=(r^2+r)*(s^2+s)/4;
           shn4=(r^2-r)*(s^2+s)/4;
           shn5=(1-r^2)*(s^2-s)/2;
           shn6=(r^2+r)*(1-s^2)/2;
           shn7=(1-r^2)*(s^2+s)/2;
           shn8=(r^2-r)*(1-s^2)/2;
           shn9=(1-r^2)*(1-s^2);

PHI(i,j)=shn1*phi(nd1,1)+shn2*phi(nd2,1)+shn3*phi(nd3,1)+shn4*phi(nd4,1)+shn5*phi(nd5,1
)+shn6*phi(nd6,1)+shn7*phi(nd7,1)+shn8*phi(nd8,1)+shn9*phi(nd9,1);
%======================================================================================
===============================================

%             PHI(i,j)=(1-r)*(1-s)*phi(nd1,1)/4+(1+r)*(1-
%s)*phi(nd2,1)/4+(1+r)*(1+s)*phi(nd3,1)/4+(1-r)*(1+s)*phi(nd4,1)/4;
             z(i,j)=((Y+1)*((sqrt(3))*X-Y+2)*(-(sqrt(3))*X-Y+2))/12;;
              break
           end%if (in==1)
        end%for iel
   %THE PROGRAM EXECUTION JUMPS TO HERE if (in==1)
    end%for j
end%for i
 % z=sin(pi*x).*sin(pi*y);
  %z=(zeros(31,31));

%for ii=1:31
 %   for jj=1:31
  %  xx=(x(ii,jj));yy=(y(ii,jj));
%z(ii,jj)=((yy+1/2)*((sqrt(3))*xx-yy+1)*(-(sqrt(3))*xx-yy+1))/6;;
%end %ii
%end%jj

for i=1:31
    for j=1:31
        if (abs(PHI(i,j))<=1e-5)
            PHI(i,j)=0;
        end
        if (abs(z(i,j))<=1e-5)
            z(i,j)=0;
        end

    end
end


end%(mesh==2)

switch mesh
    case 2
  clf
```

```
figure(1)
 x=[-sqrt(3);sqrt(3);0];
 y=[      -1;      -1;2];
 patch(x,y,'w')
hold on
%[x,y]=meshgrid(0:.1:1,0:0.1:1)
[x,y]=meshgrid(-sqrt(3):(1/15)*sqrt(3):sqrt(3),-1:(0.1):2);
%y((y>1/2)&(y<=1)&(x>1/2)&(x<=1)&(x+y>3/2))=NaN;
%%y((y>-1/2)&(y<=1)&(x>0)&(x<=(sqrt(3)/2))&((-sqrt(3)*x-y+1)<0))=NaN;
%%y((y>-1/2)&(y<=1)&(x>(-sqrt(3)/2))&(x<=0)&((sqrt(3)*x-y+1)<0))=NaN;
%[c,h]=contour(x,y,PHI)
contour(x,y,PHI,20)
xlabel('X-axis');
ylabel('Y-axis');
%clabel(c,h);
axis square
st1='Contour level curves for ';
st2='FEM solution of ';
st3='Nine Noded ';
st4='Special Quadrilateral';
st5=' Elements'
title([st1,st2,st3,st4,st5])
sst1='(MESH HAS '
sst2=num2str(nnode)
sst3=' NODES'
sst4=' AND '
sst5=num2str(nel)
sst6=' ELEMENTS)'
text(0.6,1.8,[sst1 sst2])
text(0.6,1.6,[sst3 sst4])
text(0.6,1.4,[sst5 sst6])
%text(0.25,-.08,[sst1 sst2 sst3 sst4 sst5 sst6])
%
figure(2)
 %x=[0.0 1.0 1.0 0.5 0.0];
 %y=[0.0 0.0 0.5 1.0 1.0];
 x=[-sqrt(3);sqrt(3);0];
 y=[      -1;      -1;2];
 patch(x,y,'w')
hold on
%[x,y]=meshgrid(0:.1:1,0:0.1:1)
%y((y>1/2)&(y<=1)&(x>1/2)&(x<=1)&(x+y>3/2))=NaN;
%[c,h]=contour(x,y,z)
[x,y]=meshgrid(-sqrt(3):(1/15)*sqrt(3):sqrt(3),-1:(0.1):2);

contour(x,y,z,20)
xlabel('X-axis');
ylabel('Y-axis');
%clabel(c,h);
axis square
title('contour level curves for exact solution: ')
hold off

figure(3)
% x=[0.0 1.0 1.0 0.5 0.0];
 %y=[0.0 0.0 0.5 1.0 1.0];
x=[-sqrt(3);sqrt(3);0];
y=[      -1;      -1;2];
 patch(x,y,'w')
hold on
[x,y]=meshgrid(-sqrt(3):(1/15)*sqrt(3):sqrt(3),-1:(0.1):2);
%[x,y]=meshgrid(0:.1:1,0:0.1:1)
%y((y>1/2)&(y<=1)&(x>1/2)&(x<=1)&(x+y>3/2))=NaN;
%%y((y>-1/2)&(y<=1)&(x>0)&(x<=(sqrt(3)/2))&((-sqrt(3)*x-y+1)<0))=NaN;
%%y((y>-1/2)&(y<=1)&(x>(-sqrt(3)/2))&(x<=0)&((sqrt(3)*x-y+1)<0))=NaN;
```

```
contour(x,y,PHI,'r-')

xlabel('X-axis');
ylabel('Y-axis');
%clabel(c,h);
axis square
st1='Contour level curves for ';
st2='FEM solution of ';
st3='Nine Noded ';
st4='Special Quadrilateral';
st5=' Elements'
title([st1,st2,st3,st4,st5])
sst1=' NODES='
sst2=num2str(nnode)
sst3=' ELEMENTS='
sst4=num2str(nel)
text(0.6,1.1,[sst1 sst2])
text(0.6,.9,[sst3 sst4])

hold on
%[x,y]=meshgrid(0:.1:1,0:0.1:1)
%[c,h]=contour(x,y,z,'g-')
contour(x,y,z,'b-')
%xlabel('X-axis');
%ylabel('Y-axis');
%clabel(c,h);
axis square
text(0.6,1.9,'{ SUPERPOSITION OF }')
text(0.6,1.7,'{ FEM/EXACT SOLUTIONS}')
text(0.6,1.5,'--(red)FEM ')
text(0.6,1.3,'--(blue)EXACT')

 mm=0;
 for i=1:31
  for j=1:31
      mm=mm+1;
      femsoln(mm,1)=PHI(i,j);
      exactsoln(mm,1)=z(i,j);
  end
end
end
% [femsoln exactsoln]

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('-----------------------------------------------------------------------------------------
-------------')
```

%(3)nodaladdresses_special_convex_quadrilaterals_2nd_orderLG.m

```
function[eln,nodetel,nodes,nnode]=nodaladdresses_special_convex_quadrilaterals_2nd_orde
rLG(n)
%division of a standard triangle(right isoscles triangle)
%into eight node special_convex_quadrilaterals
for nelm=1:3*(n/2)^2
    spqd(nelm,1:9)=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(elm)
%disp(length(elm))

jj=0;kk=0;
for j=0:n-1
    jj=j+1;
for k=1:(n+1)-j
    kk=kk+1;
    row_nodes(jj,k)=elm(kk,1);
end
end
row_nodes(n+1,1)=3;
%for jj=(n+1):-1:1
 %   disp(row_nodes(jj,:))
 %end
[row_nodes]
rr=row_nodes;
rrr(:,:,1)=rr;
%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


%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%if
%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%if
%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%if
%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);
   nd=nd+1;
   nodes(mmm,9)=nd;
end
nnode=nd;
nel=mm;
spqd=nodes;

%***************************************************************************%(4)coordinate_
arbitrarytriangle_2ndorderLAGR.m

function[coord,gcoord]=coordinate_arbitrarytriangle_2ndorderLAGR(x,y,n)
syms ui vi wi xi yi
x1=x(1,1);x2=x(2,1);x3=x(3,1);
y1=y(1,1);y2=y(2,1);y3=y(3,1);
[ui,vi,wi]=coordinate_special_quadrilaterals_in_stdtriangle_2nd_orderLAGR(n)
%disp([ui vi wi])
N=length(ui);
    NN=(1:N)';
for i=1:N
    xi(i,1)=x1*wi(i,1)+x2*ui(i,1)+x3*vi(i,1);
    yi(i,1)=y1*wi(i,1)+y2*ui(i,1)+y3*vi(i,1);
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);
%(5)coordinate_special_quadrilaterals_in_stdtriangle_2nd_orderLAGR.m
```

```
function[ui,vi,wi]=coordinate_special_quadrilaterals_in_stdtriangle_2nd_orderLAGR(n)
%n must be even:n=2,4,6,......
syms ui vi wi
ui(1:3,1)=[0;1;0];
vi(1:3,1)=[0;0;1];
wi(1:3,1)=[1;0;0];
if (n-1)>0
 kk=3;
 for i=1:n-1
    kk=kk+1;
    ui(kk,1)=sym(i/n);
    vi(kk,1)=sym(0);
    wi(kk,1)=sym(1-ui(kk,1)-vi(kk,1));
 end
 kkk=kk;
 for ii=1:n-1
    kkk=kkk+1;
    ui(kkk,1)=sym((n-ii)/n);
    vi(kkk,1)=sym(1-(n-ii)/n);
    wi(kkk,1)=0;
 end;
 kkkk=kkk;
  for iii=1:n-1
    kkkk=kkkk+1;
    ui(kkkk,1)=0;
    vi(kkkk,1)=sym(1-iii/n);
    wi(kkkk,1)=sym(iii/n);
  end
end%if (n-1)>0
if (n-2)>0
  kkkkk=kkkk;
  for iiii=1:(n-2)
    for jjjj=1:(n-1)-iiii
         kkkkk=kkkkk+1;
         ui(kkkkk,1)=sym(jjjj/n);
         vi(kkkkk,1)=sym(iiii/n);
         wi(kkkkk,1)=sym(1-ui(kkkkk,1)-vi(kkkkk,1));
     end
  end
end%if (n-2)>0
if n==2

    num=(1:6)';
else
  num=(1:kkkkk)';

end
%disp([ui'])
%disp([vi'])
%disp([wi'])
%length(ui)
%length(vi)
%length(wi)

%disp([num ui vi wi])
[eln,nodetel,spqd,nnode]=nodaladdresses4Lagrangespecial_convex_quadrilaterals_2nd_order
(n)
%n=number of divisions along sides;n must be even i.e n=2,4,6,.....
%qq=number of nodes on the triangle
%nc=number of six node triangles,we can insert a centroid in each 6-node triangle
qq=(n+1)*(n+2)/2;
    nc=(n/2)^2;
for pp=1:nc
 qq=qq+1;
 q1=eln(pp,1);
```

```
 q2=eln(pp,2);
 q3=eln(pp,3);
 ui(qq,1)=(ui(q1,1)+ui(q2,1)+ui(q3,1))/3;
 vi(qq,1)=(vi(q1,1)+vi(q2,1)+vi(q3,1))/3;
 wi(qq,1)=1-ui(qq,1)-vi(qq,1);
end
 %disp([ui vi wi])
%length(ui)
%length(vi)
%length(wi)

 num=(1:qq)';
%disp([num ui vi wi])
qqq=qq;
for ppp=1:3*nc
qq1=spqd(ppp,1);
qq2=spqd(ppp,2);
qq3=spqd(ppp,3);
qq4=spqd(ppp,4);
%midside nodes-1,2
 qqq=spqd(ppp,5);
 ui(qqq,1)=(ui(qq1,1)+ui(qq2,1))/2;
 vi(qqq,1)=(vi(qq1,1)+vi(qq2,1))/2;
 wi(qqq,1)=1-ui(qqq,1)-vi(qqq,1);
%midside nodes-2,3
 qqq=spqd(ppp,6);
 ui(qqq,1)=(ui(qq2,1)+ui(qq3,1))/2;
 vi(qqq,1)=(vi(qq2,1)+vi(qq3,1))/2;
 wi(qqq,1)=1-ui(qqq,1)-vi(qqq,1);
 %midside nodes-3,4
 qqq=spqd(ppp,7);
 ui(qqq,1)=(ui(qq3,1)+ui(qq4,1))/2;
 vi(qqq,1)=(vi(qq3,1)+vi(qq4,1))/2;
 wi(qqq,1)=1-ui(qqq,1)-vi(qqq,1);
%midside nodes-4,1
 qqq=spqd(ppp,8);
 ui(qqq,1)=(ui(qq1,1)+ui(qq4,1))/2;
 vi(qqq,1)=(vi(qq1,1)+vi(qq4,1))/2;
 wi(qqq,1)=1-ui(qqq,1)-vi(qqq,1);
 %centre node
 qqq=spqd(ppp,9);
 ui(qqq,1)=(ui(qq1,1)+ui(qq2,1)+ui(qq3,1)+ui(qq4,1))/4;
 vi(qqq,1)=(vi(qq1,1)+vi(qq2,1)+vi(qq3,1)+vi(qq4,1))/4;
 wi(qqq,1)=1-ui(qqq,1)-vi(qqq,1);

end
maxnode=max(max(spqd(1:3*nc,1:9)));
 num=(1:maxnode)';
 %disp(['maximum value of node number=',num2str(maxnode)])
%disp(' node    ui      vi      wi')
%disp([num ui vi wi])
```

% (6)nodaladdresses4Lagrangespecial_convex_quadrilaterals_2nd_order.m

```
function[eln,nodetel,nodes,nnode]=nodaladdresses4Lagrangespecial_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,1:9)=0;
end
syms mst_tri x
%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(elm)
%disp(length(elm))

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


%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
nd=nd+1;
   nodes(mmm,5)=mdpt(mmm1,mmm2);
   nodes(mmm,6)=mdpt(mmm2,mmm3);
   nodes(mmm,7)=mdpt(mmm3,mmm4);
   nodes(mmm,8)=mdpt(mmm4,mmm1);
   nodes(mmm,9)=nd;
end
nnode=nd;
nel=mm;
```

%(7)D2LaplaceEquationQ9Ex3automeshgenNewPolygon.m

```
function[]=D2LaplaceEquationQ9Ex3automeshgenNewPolygon(n1,n2,n3,nmax,numtri,ndiv,mesh)
%ndiv=2,4,6,8,......
%polygonal_domain_coordinates([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,1,2)
%polygonal_domain_coordinates([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,4,4)
%D2LaplaceEquationQ4MoinExautomeshgen(n1,n2,n3,nmax,numtri,ndiv)
%D2LaplaceEquationQ4MoinExautomeshgen([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8
,1,2,1)
%D2LaplaceEquationQ4MoinExautomeshgen([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8
,4,4,1)
%D2LaplaceEquationQ4MoinExautomeshgen([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;
9;2],9,1,2,2)
%D2LaplaceEquationQ4MoinExautomeshgen([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;
9;2],9,4,4,2)
%quadrilateral_mesh4MOINEX_q4(n1,n2,n3,nmax,numtri,ndiv,mesh,xlength,ylength)([1;1;1;1;
1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9,1,2,2,1,1)
%D2POISSONEQUATION_NODALINTERPOLATION_VALUES(n1,n2,n3,nmax,numtri,ndiv,mesh)([1;1;1;1;1
;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9,1,2,2)
%D2LaplaceEquationQ4MoinExautomeshgen([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;
9;2],9,100,20,2)
%D2PoissonEquationQ4MoinEx_MeshgridContour([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6
;7;8;9;2],9,1,2,2)
%D2PoissonEquationQ4MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,1,2,1)
%D2PoissonEquationQ4MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,4,4,1)
%D2PoissonEquationQ4MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,9,6,1)
```

```
%D2PoissonEquationQ4MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,16,8,1)
%D2PoissonEquationQ4MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,25,10,1)
%D2PoissonEquationQ8MoinEx_MeshgridContour([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6
;7;8;9;2],9,1,2,2)
%D2PoissonEquationQ8MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,1,2,1)
%D2PoissonEquationQ9MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,1,2,1)
%D2PoissonEquationQ9MoinEx_MeshgridContour([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6
;7;8;9;2],9,1,2,2)
%D2PoissonEquationQ9MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,1,2,1)
%%D2PoissonEquationQ9MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;
8;2],8,1,2,1,1,10)
%D2PoissonEquationQ9MoinEx_MeshgridContour([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6
;7;8;9;2],9,1,2,2,1,10)
%D2LaplaceEquationQ9Ex3automeshgenNewPolygon([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5
;6;7;8;9;2],9,1,2,4)
syms coord
%[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates(n1,n2,n3,nmax,numt
ri,ndiv,mesh)
%nnel=4;
[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates_2nd_orderLG(n1,n2,n
3,nmax,numtri,ndiv,mesh)
nnel=9;
ndof=1;
%nc=(ndiv/2)^2;
%nnode=(ndiv+1)*(ndiv+2)/2+nc;
%nel=3*nc;
sdof=nnode*ndof;
ff=(zeros(sdof,1));ss=(zeros(sdof,sdof));

%nnode=17,nel=12,nnel=4,ndof=1
%>>LaplaceEquationQuad4twodimension(12,17,4,1)
%
%Ex1:nnode=41,nel=36,,nnel=4,nodf=1
%>>LaplaceEquationQuad4twodimensionEx1(36,41,4,1)
%>>improvedLaplaceEquationQuad4twodimensionEx1_explicit(36,41,4,1)
%Ex2:nnode=83,nel=69,,nnel=4,nodf=1
%>>improvedLaplaceEquationQuad4twodimensionEx2_explicit(69,83,4,1)#
%>>improvedLaplaceEquationQuad4twodimensionEx2_explicitfnmesh(69,83,4,1)#
%improvedLaplaceEquationQuad4twodimensionEx2_explicitvfnmesh(72,87,4,1)#new
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(nel=3,nnode=7,nnel=4,ndof=1)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(nel,nnode,nnel=4,ndof=1,quadt
ype=0/3,mesh=1,2,3...)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(nel=12,nnode=19,nnel=4,ndof=1
,quadtype=0/3,mesh=3)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(nel=27,nnode=37,nnel=4,ndof=1
,quadtype=0/3,mesh=4)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(nel=48,nnode=61,nnel=4,ndof=1
,quadtype=0/3,mesh=5)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(nel=75,nnode=91,nnel=4,ndof=1
,quadtype=0/3,mesh=6)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(108,127,4,1,3,7)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(147,169,4,1,3,8)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(192,217,4,1,3,9)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(243,271,4,1,3,10)
disp([nel nnode nnel ndof])
format long g
for i=1:nel
N(i,1)=i;
end
for i=1:nel
NN(i,1)=i;
```

```
end
    %[coord,gcoord]=coordinate_rtisoscelestriangle00_h0_hh(ndiv);
    %[nodetel,nodes]=nodaladdresses4special_convex_quadrilaterals(ndiv)
    %
    %bcdof=[2;5;3]
   %boundary conditions-1
 switch mesh


case 4
    %boundary conditions-2
    nnn=0;
    for nn=1:nnode
        xnn=coord(nn,1);ynn=gcoord(nn,2);
        if (xnn==-1/2)&((ynn>=-1/2)&(ynn<=1/2))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
    end
    %boundary conditions-2
     for nn=1:nnode
        xnn=gcoord(nn,1);ynn=coord(nn,2);
        if (ynn==-1/2)&((xnn>=-1/2)&(xnn<=1/2))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
    end
    %boundary conditions-3
    for nn=1:nnode
        xnn=gcoord(nn,1);ynn=coord(nn,2);
        if (ynn==1/2)&((xnn>=-1/2)&(xnn<=1/2))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
    end
   %boundary conditions-4
   for nn=1:nnode
      xnn=coord(nn,1);ynn=gcoord(nn,2);
        if (xnn==1/2)&((ynn>=-1/2)&(ynn<=1/2))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        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
    gx=(gcoord(m,1));gy=(gcoord(m,2));rr=(0);
for n=1:2:99
rr=rr+(-1)^((n-1)/2)*(1-(cosh(n*pi*gy)/cosh(n*pi/2)))*cos(n*pi*gx)/n^3;
end
xi(m,1)=(a0*rr);
end
```

```
   end %switch
 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
%__________________________________________________________________
%ng=10

table2=[N xx yy];
%disp([xx yy])
%_______________________________________________________________________
```

```
intJdn1dn1uvrs =[vpa(sym('  .68800369699992917040751928889')) vpa(sym(' .63026863732902919654693031748'));...
vpa(sym(' .63026863732902919654693031748')) vpa(sym('  .68800369699992917040751928889'))];
intJdn1dn2uvrs =[vpa(sym('  .49469766679465431322423128849e-1')) vpa(sym(' -.92492499790138905674372588873e-1'));...
vpa(sym('  .74174166876527760992294077793e-1')) vpa(sym(' -.58041502241256898516961218830e-1'))];
intJdn1dn3uvrs =[vpa(sym(' -.27639187071864196478244612770e-1')) vpa(sym(' -.36406227705135716084660539928e-1'));...
vpa(sym(' -.36406227705135716084660539928e-1')) vpa(sym(' -.27639187071864196478244612770e-1'))];
intJdn1dn4uvrs =[vpa(sym(' -.58041502241256898516961218830e-1')) vpa(sym('  .74174166876527760992294077793e-1'));...
vpa(sym(' -.92492499790138905674372588873e-1')) vpa(sym('  .49469766679465431322423128849e-1'))];
intJdn1dn5uvrs =[vpa(sym(' -.42034020155248898256147959993')) vpa(sym('  .23835534506556861836117847703'));...
vpa(sym(' -.42831132160109804830548819964')) vpa(sym('  .10203781293970562330870800320'))];
intJdn1dn6uvrs =[vpa(sym(' .12626870005164588340746772853')) vpa(sym(' .16865169113202404137334892156'));...
vpa(sym(' .16865169113202404137334892156')) vpa(sym(' .15224655673422927054938212414'))];
intJdn1dn7uvrs =[vpa(sym(' .15224655673422927054938212414')) vpa(sym(' .16865169113202404137334892156'));...
vpa(sym(' .16865169113202404137334892156')) vpa(sym(' .12626870005164588340746772853'))];
intJdn1dn8uvrs =[vpa(sym('  .10203781293970562330870800320')) vpa(sym(' -.42831132160109804830548819964'));...
vpa(sym('  .23835534506556861836117847703')) vpa(sym(' -.42034020155248898256147959993'))];
intJdn1dn9uvrs =[vpa(sym(' -.61200564253936530143881487716')) vpa(sym(' -.72289148243880098858257938847'));...
vpa(sym(' -.72289148243880098858257938847')) vpa(sym(' -.61200564253936530143881487716'))];
intJdn2dn1uvrs =[vpa(sym('  .49469766679465431322423128849e-1')) vpa(sym('  .74174166876527760992294077793e-1'));...
vpa(sym(' -.92492499790138905674372588873e-1')) vpa(sym(' -.58041502241256898516961218830e-1'))];
intJdn2dn2uvrs =[vpa(sym('  .24822242723570359859415910165')) vpa(sym(' -.13943577781062161645243966287'));...
vpa(sym(' -.13943577781062161645243966287')) vpa(sym('  .22525137401606776447455758469'))];
intJdn2dn3uvrs =[vpa(sym('  -.93683290648218768462051538869e-1')) vpa(sym(' -.82852283681797324065968279846e-1'));...
vpa(sym('  .83814382984869342600698386820e-1')) vpa(sym('   .37904800898126436475032582264e-1'))];
intJdn2dn4uvrs =[vpa(sym('   .62925700025302140451751960e-3')) vpa(sym(' .26100052047122979851449223346e-1'));...
vpa(sym(' .26100052047122979851449223346e-1')) vpa(sym('   .62925700025302140451751960e-3'))];
intJdn2dn5uvrs =[vpa(sym('   -.46868216752790132335135919662')) vpa(sym('  -.34646194452109534302782806358'));...
vpa(sym('   .32020472214557132363883860309')) vpa(sym(' .85836729209541381287229242888e-1'))];
intJdn2dn6uvrs =[vpa(sym('   .20486711030201898764584585532')) vpa(sym('  .25718774442435573105221634722'));...
vpa(sym(' -.40947892224231093561445031945')) vpa(sym(' -.28073862677651621463393803502'))];
intJdn2dn7uvrs =[vpa(sym(' .50885917641966498462888866644e-1')) vpa(sym('  -.10335794828229434487750587371'));...
vpa(sym('  -.10335794828229434487750587371')) vpa(sym(' -.17609016019421764322513358580e-2'))];
intJdn2dn8uvrs =[vpa(sym(' -.41736856619169902011341267758e-1')) vpa(sym(' -.94765689307712237483661393307e-1'));...
vpa(sym(' -.94765689307712237483661393307e-1')) vpa(sym('  .33881015844374607003400451161e-1'))];
intJdn2dn9uvrs =[vpa(sym('   .50027835935882456394917533332e-1')) vpa(sym('    .40941168025551439401144362445'));...
vpa(sym('    .40941168025551439401144362445')) vpa(sym(' -.42962146348647921061586792226e-1'))];
intJdn3dn1uvrs =[vpa(sym(' -.27639187071864196478244612770e-1')) vpa(sym(' -.36406227705135716084660539928e-1'));...
vpa(sym(' -.36406227705135716084660539928e-1')) vpa(sym(' -.27639187071864196478244612770e-1'))];
intJdn3dn2uvrs =[vpa(sym('  -.93683290648218768462051538869e-1')) vpa(sym('  .83814382984869342600698386820e-1'));...
vpa(sym(' -.82852283681797324065968279846e-1')) vpa(sym('   .37904800898126436475032582264e-1'))];
intJdn3dn3uvrs =[vpa(sym('  .33203856905137126709551224250')) vpa(sym(' .29812794600013651335576030271'));...
vpa(sym(' .29812794600013651335576030271')) vpa(sym('  .33203856905137126709551224250'))];
intJdn3dn4uvrs =[vpa(sym('   .37904800898126436475032582264e-1')) vpa(sym(' -.82852283681797324065968279846e-1'));...
vpa(sym('  .83814382984869342600698386820e-1')) vpa(sym('  -.93683290648218768462051538869e-1'))];
intJdn3dn5uvrs =[vpa(sym('   .15710844821544667058002407565')) vpa(sym('   .12945549547911074908308207668'));...
vpa(sym('   .12945549547911074908308207668')) vpa(sym(' .43059644229262196507731226697e-1'))];
intJdn3dn6uvrs =[vpa(sym('  .14159217838663875388665351332')) vpa(sym(' -.35360191630747700837476451283'));...
vpa(sym('  .31306475035918965829190215383')) vpa(sym(' -.34086853405320051524855521401'))];
intJdn3dn7uvrs =[vpa(sym(' -.34086853405320051524855521401')) vpa(sym('  .31306475035918965829190215383'));...
vpa(sym(' -.35360191630747700837476451283')) vpa(sym('  .14159217838663875388665351332'))];
```

```
intJdn3dn8uvrs =[vpa(sym(' .4305964422926219650773122697e-1')) vpa(sym('   .1294554954791107490830820766 8'));...
vpa(sym('   .12945549547911074908308207668')) vpa(sym('   .15710844821544667058002407565'))];
intJdn3dn9uvrs =[vpa(sym('  -.24951262900756184435610227 57')) vpa(sym(' -.48105764260800696388913166479'));...
vpa(sym(' -.48105764260800696388913166479')) vpa(sym('  -.24951262900756184435610227 57'))];
intJdn4dn1uvrs =[vpa(sym(' -.5804150224125689851696121830e-1')) vpa(sym(' -.9249249979013890567437258873e-1'));...
vpa(sym('  .7417416687652776099229407793e-1')) vpa(sym('  .4946976667946543132242312849e-1'))];
intJdn4dn2uvrs =[vpa(sym('   .6292570002530214045175196 0e-3')) vpa(sym(' .2610005204712297985144922346e-1'));...
vpa(sym(' .2610005204712297985144922346e-1')) vpa(sym('   .62925700025302140451751960e-3'))];
intJdn4dn3uvrs =[vpa(sym('   .3790480089812643647503258264e-1')) vpa(sym('  .8381438298486934260069838682 0e-1'));...
vpa(sym(' -.8285228368179732406596827984 6e-1')) vpa(sym('  -.9368329064821876846205153869e-1'))];
intJdn4dn4uvrs =[vpa(sym('  .22525137401606776447455758469')) vpa(sym(' -.13943577781062161645243966287'));...
vpa(sym(' -.13943577781062161645243966287')) vpa(sym('  .24822242723570359859415910165'))];
intJdn4dn5uvrs =[vpa(sym('  .3388101584437460700340045161e-1')) vpa(sym(' -.9476568930771223748366139307e-1'));...
vpa(sym(' -.9476568930771223748366139307e-1')) vpa(sym(' -.4173685661916990201134126758e-1'))];
intJdn4dn6uvrs =[vpa(sym(' -.1760901601942176432251335 80e-2')) vpa(sym('  -.10335794828229434487750587371'));...
vpa(sym('  -.10335794828229434487750587371')) vpa(sym(' .5088591764196649846288886644e-1'))];
intJdn4dn7uvrs =[vpa(sym(' -.28073862677651621463393803502')) vpa(sym(' -.40947892224231093561445031945'));...
vpa(sym('  .25718774442435573105221634722')) vpa(sym('   .20486711030201898764584585 32'))];
intJdn4dn8uvrs =[vpa(sym(' .8583672920954138128722924288e-1')) vpa(sym('   .32020472214557132363883860309'));...
vpa(sym('  -.34646194452109534302782806358')) vpa(sym('   -.46868216752790132335135919 62'))];
intJdn4dn9uvrs =[vpa(sym(' -.4296214634864792106158679226e-1')) vpa(sym('    .40941168025551439401144362445'));...
vpa(sym('    .40941168025551439401144362445')) vpa(sym('   .5002783593588245639491753 32e-1'))];
intJdn5dn1uvrs =[vpa(sym(' -.42034020155248898256147959 93')) vpa(sym(' -.42831132160109804830548819 64'));...
vpa(sym('  .23835534506556861836117847 03')) vpa(sym('  .10203781293970562330870803 20'))];
intJdn5dn2uvrs =[vpa(sym('   -.46868216752790132335135919 62')) vpa(sym('   .32020472214557132363883860309'));...
vpa(sym('  -.34646194452109534302782806358')) vpa(sym(' .8583672920954138128722924288e-1'))];
intJdn5dn3uvrs =[vpa(sym('   .15710844821544667058002407565')) vpa(sym('   .12945549547911074908308207668'));...
vpa(sym('   .12945549547911074908308207668')) vpa(sym(' .4305964422926219650773122697e-1'))];
intJdn5dn4uvrs =[vpa(sym('  .3388101584437460700340045161e-1')) vpa(sym(' -.9476568930771223748366139307e-1'));...
vpa(sym(' -.9476568930771223748366139307e-1')) vpa(sym(' -.4173685661916990201134126758e-1'))];
intJdn5dn5uvrs =[vpa(sym(' 1.0663384833344564856750376651')) vpa(sym('  .56556859863119073241392390 19'));...
vpa(sym('  .56556859863119073241392390 19')) vpa(sym(' 1.1211333295329756651950424278'))];
intJdn5dn6uvrs =[vpa(sym('  -.58306188548401601151324941 69')) vpa(sym(' -.54252818687130357695788942932'));...
vpa(sym(' -.54252818687130357695788942932')) vpa(sym(' -.22579554943131879593427777753'))];
intJdn5dn7uvrs =[vpa(sym('   -.16767681050348731002396257 42')) vpa(sym(' -.1709647811328463063306097 92e-2'));...
vpa(sym(' -.1709647811328463063306097 92e-2')) vpa(sym('   .17132457681637017904834081102'))];
intJdn5dn8uvrs =[vpa(sym(' .9027148605382036792067597 12e-1')) vpa(sym('   .41024643803333829631989611 84'));...
vpa(sym('   .41024643803333829631989611 84')) vpa(sym(' .9027148605382036792067597 12e-1'))];
intJdn5dn9uvrs =[vpa(sym('   .29216163161979549627091262 30')) vpa(sym('  -.35816040869776877564539558 36'));...
vpa(sym('  -.35816040869776877564539558 36')) vpa(sym(' -1.3461311727311867153221086666'))];
intJdn6dn1uvrs =[vpa(sym(' .12626870005164588340746772853')) vpa(sym(' .16865169113202404137334892156'));...
vpa(sym(' .16865169113202404137334892156')) vpa(sym(' .15224655673422927054938212414'))];
intJdn6dn2uvrs =[vpa(sym('   .20486711030201898764584585 32')) vpa(sym(' -.40947892224231093561445031945'));...
vpa(sym('  .25718774442435573105221634722')) vpa(sym(' -.28073862677651621463393803502'))];
intJdn6dn3uvrs =[vpa(sym('  .14159217838663875388665351332')) vpa(sym('  .31306475035918965829190215383'));...
vpa(sym(' -.35360191630747700837476451283')) vpa(sym(' -.34086853405320051524855521401'))];
intJdn6dn4uvrs =[vpa(sym(' -.1760901601942176432251335 80e-2')) vpa(sym('  -.10335794828229434487750587371'));...
vpa(sym('  -.10335794828229434487750587371')) vpa(sym(' .5088591764196649846288886644e-1'))];
intJdn6dn5uvrs =[vpa(sym('  -.58306188548401601151324941 69')) vpa(sym(' -.54252818687130357695788942932'));...
vpa(sym(' -.54252818687130357695788942932')) vpa(sym(' -.22579554943131879593427777753'))];
intJdn6dn6uvrs =[vpa(sym(' 1.5998510858095308546375928228')) vpa(sym(' .24469327225090109904911663174'));...
vpa(sym(' .24469327225090109904911663174')) vpa(sym('  .64675514174853057100283264 58'))];
intJdn6dn7uvrs =[vpa(sym(' -.7461268601626634110113636 05e-1')) vpa(sym('   .42496611860422613701397774221'));...
vpa(sym('   .42496611860422613701397774221')) vpa(sym(' -.7461268601626634110113636 05e-1'))];
intJdn6dn8uvrs =[vpa(sym('   .17132457681637017904834081102')) vpa(sym(' -.1709647811328463063306097 92e-2'));...
vpa(sym(' -.1709647811328463063306097 92e-2')) vpa(sym('   -.16767681050348731002396257 42'))];
intJdn6dn9uvrs =[vpa(sym('   -1.5844681782639801295792636156')) vpa(sym(' -.9430112713910361521519372896e-1'));...
vpa(sym(' -.9430112713910361521519372896e-1')) vpa(sym('    .23980459065606283692676632 48'))];
intJdn7dn1uvrs =[vpa(sym(' .15224655673422927054938212414')) vpa(sym(' .16865169113202404137334892156'));...
vpa(sym(' .16865169113202404137334892156')) vpa(sym(' .12626870005164588340746772853'))];
intJdn7dn2uvrs =[vpa(sym(' .5088591764196649846288886644e-1')) vpa(sym('  -.10335794828229434487750587371'));...
vpa(sym('  -.10335794828229434487750587371')) vpa(sym(' -.1760901601942176432251335 80e-2'))];
intJdn7dn3uvrs =[vpa(sym(' -.34086853405320051524855521401')) vpa(sym(' -.35360191630747700837476451283'));...
vpa(sym('  .31306475035918965829190215383')) vpa(sym('  .14159217838663875388665351332'))];
intJdn7dn4uvrs =[vpa(sym(' -.28073862677651621463393803502')) vpa(sym('  .25718774442435573105221634722'));...
vpa(sym(' -.40947892224231093561445031945')) vpa(sym('   .20486711030201898764584585 32'))];
intJdn7dn5uvrs =[vpa(sym('   -.16767681050348731002396257 42')) vpa(sym(' -.1709647811328463063306097 92e-2'));...
vpa(sym(' -.1709647811328463063306097 92e-2')) vpa(sym('   .17132457681637017904834081102'))];
```

```
intJdn7dn6uvrs =[vpa(sym(' -.746126860162663411011363605e-1')) vpa(sym('   .42496611860422613701397774221'));...
vpa(sym('   .42496611860422613701397774221')) vpa(sym(' -.746126860162663411011363605e-1'))];
intJdn7dn7uvrs =[vpa(sym('  .6467551417485305710028326458')) vpa(sym(' .24469327225090109904911663174'));...
vpa(sym(' .24469327225090109904911663174')) vpa(sym(' 1.5998510858095308546375928228'))];
intJdn7dn8uvrs =[vpa(sym(' -.22579554943131879593427777753')) vpa(sym(' -.54252818687130357695788942932'));...
vpa(sym(' -.54252818687130357695788942932')) vpa(sym('  -.5830618854840160115132494169'))];
intJdn7dn9uvrs =[vpa(sym('    .2398045906560628369267663248')) vpa(sym(' -.9430112713910361521519372896e-1'));...
vpa(sym(' -.9430112713910361521519372896e-1')) vpa(sym('   -1.5844681782639801295792636156'))];
intJdn8dn1uvrs =[vpa(sym('  .1020378129397056233087080320')) vpa(sym('  .2383553450655686183611784703'));...
vpa(sym(' -.4283113216010980483054881964')) vpa(sym(' -.4203402015524889825614795993'))];
intJdn8dn2uvrs =[vpa(sym(' -.4173685661916990201134126758e-1')) vpa(sym(' -.9476568930771223748366139307e-1'));...
vpa(sym(' -.9476568930771223748366139307e-1')) vpa(sym('  .3388101584437460700340045161e-1'))];
intJdn8dn3uvrs =[vpa(sym(' .4305964422926219650773122697e-1')) vpa(sym('   .12945549547911074908308207668'));...
vpa(sym('   .12945549547911074908308207668')) vpa(sym('   .15710844821544667058002407565'))];
intJdn8dn4uvrs =[vpa(sym(' .8583672920954138128722924288e-1')) vpa(sym('  -.34646194452109534302782806358'));...
vpa(sym('   .32020472214557132363883860309')) vpa(sym('   -.4686821675279013233513591962'))];
intJdn8dn5uvrs =[vpa(sym(' .902714860538203679206759712e-1')) vpa(sym('   .4102464380333382963198961184'));...
vpa(sym('   .4102464380333382963198961184')) vpa(sym(' .902714860538203679206759712e-1'))];
intJdn8dn6uvrs =[vpa(sym('   .17132457681637017904834081102')) vpa(sym(' -.170964781132846306330609792e-2'));...
vpa(sym(' -.170964781132846306330609792e-2')) vpa(sym('   -.1676768105034873100239625742'))];
intJdn8dn7uvrs =[vpa(sym(' -.22579554943131879593427777753')) vpa(sym(' -.54252818687130357695788942932'));...
vpa(sym(' -.54252818687130357695788942932')) vpa(sym('  -.5830618854840160115132494169'))];
intJdn8dn8uvrs =[vpa(sym(' 1.1211333295329756651950424278')) vpa(sym('  .5655685986311907324139239019'));...
vpa(sym('  .5655685986311907324139239019')) vpa(sym(' 1.0663384833344564856750376651'))];
intJdn8dn9uvrs =[vpa(sym(' -1.3461311727311867153221086666')) vpa(sym('  -.3581604086977687756453955836'));...
vpa(sym('  -.3581604086977687756453955836')) vpa(sym('   .2921616316197954962709126230'))];
intJdn9dn1uvrs =[vpa(sym(' -.6120056425393653014388148716')) vpa(sym(' -.7228914824388009885825793847'));...
vpa(sym(' -.7228914824388009885825793847')) vpa(sym(' -.6120056425393653014388148716'))];
intJdn9dn2uvrs =[vpa(sym('   .500278359358824563949175332e-1')) vpa(sym('    .40941168025551439401144362445'));...
vpa(sym('    .40941168025551439401144362445')) vpa(sym(' -.4296214634864792106158679226e-1'))];
intJdn9dn3uvrs =[vpa(sym('  -.2495126290075618443561022757')) vpa(sym(' -.48105764260800696388913166479'));...
vpa(sym(' -.48105764260800696388913166479')) vpa(sym('  -.2495126290075618443561022757'))];
intJdn9dn4uvrs =[vpa(sym(' -.4296214634864792106158679226e-1')) vpa(sym('    .40941168025551439401144362445'));...
vpa(sym('    .40941168025551439401144362445')) vpa(sym('   .500278359358824563949175332e-1'))];
intJdn9dn5uvrs =[vpa(sym('   .2921616316197954962709126230')) vpa(sym('  -.3581604086977687756453955836'));...
vpa(sym('  -.3581604086977687756453955836')) vpa(sym(' -1.3461311727311867153221086666'))];
intJdn9dn6uvrs =[vpa(sym('   -1.5844681782639801295792636156')) vpa(sym(' -.9430112713910361521519372896e-1'));...
vpa(sym(' -.9430112713910361521519372896e-1')) vpa(sym('    .2398045906560628369267663248'))];
intJdn9dn7uvrs =[vpa(sym('    .2398045906560628369267663248')) vpa(sym(' -.9430112713910361521519372896e-1'));...
vpa(sym(' -.9430112713910361521519372896e-1')) vpa(sym('   -1.5844681782639801295792636156'))];
intJdn9dn8uvrs =[vpa(sym(' -1.3461311727311867153221086666')) vpa(sym('  -.3581604086977687756453955836'));...
vpa(sym('  -.3581604086977687756453955836')) vpa(sym('   .2921616316197954962709126230'))];
intJdn9dn9uvrs =[vpa(sym(' 3.2530857106790011221652797406')) vpa(sym(' 1.2900488362095239461700024257'));...
vpa(sym(' 1.2900488362095239461700024257')) vpa(sym(' 3.2530857106790011221652797406'))];
%integrals of products of glbal derivatives
intJdndn=double([intJdn1dn1uvrs intJdn1dn2uvrs intJdn1dn3uvrs intJdn1dn4uvrs intJdn1dn5uvrs intJdn1dn6uvrs intJdn1dn7uvrs intJdn1dn8uvrs intJdn1dn9uvrs ;...
      intJdn2dn1uvrs intJdn2dn2uvrs intJdn2dn3uvrs intJdn2dn4uvrs intJdn2dn5uvrs intJdn2dn6uvrs intJdn2dn7uvrs intJdn2dn8uvrs intJdn2dn9uvrs;...
      intJdn3dn1uvrs intJdn3dn2uvrs intJdn3dn3uvrs intJdn3dn4uvrs intJdn3dn5uvrs intJdn3dn6uvrs intJdn3dn7uvrs intJdn3dn8uvrs intJdn3dn9uvrs;...
      intJdn4dn1uvrs intJdn4dn2uvrs intJdn4dn3uvrs intJdn4dn4uvrs intJdn4dn5uvrs intJdn4dn6uvrs intJdn4dn7uvrs intJdn4dn8uvrs intJdn4dn9uvrs;...
      intJdn5dn1uvrs intJdn5dn2uvrs intJdn5dn3uvrs intJdn5dn4uvrs intJdn5dn5uvrs intJdn5dn6uvrs intJdn5dn7uvrs intJdn5dn8uvrs intJdn5dn9uvrs;...
      intJdn6dn1uvrs intJdn6dn2uvrs intJdn6dn3uvrs intJdn6dn4uvrs intJdn6dn5uvrs intJdn6dn6uvrs intJdn6dn7uvrs intJdn6dn8uvrs intJdn6dn9uvrs;...
      intJdn7dn1uvrs intJdn7dn2uvrs intJdn7dn3uvrs intJdn7dn4uvrs intJdn7dn5uvrs intJdn7dn6uvrs intJdn7dn7uvrs intJdn7dn8uvrs intJdn7dn9uvrs;...
      intJdn8dn1uvrs intJdn8dn2uvrs intJdn8dn3uvrs intJdn8dn4uvrs intJdn8dn5uvrs intJdn8dn6uvrs intJdn8dn7uvrs intJdn8dn8uvrs intJdn8dn9uvrs;...
      intJdn9dn1uvrs intJdn9dn2uvrs intJdn9dn3uvrs intJdn9dn4uvrs intJdn9dn5uvrs intJdn9dn6uvrs intJdn9dn7uvrs intJdn9dn8uvrs intJdn9dn9uvrs]);
%____________________________________________________________________________________________

%
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(1:9,1:9)=(zeros(9,9));
for i=1:9
 for j=i:9
 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 =[5/144;1/24;7/144;1/24]*(2*delabc);

 xe(1,1)=(xa+xb+xc)/3;
 xe(2,1)=(xa+xc)/2;
 xe(3,1)=xa;
 xe(4,1)=(xa+xb)/2;
 %
 ye(1,1)=(ya+yb+yc)/3;
 ye(2,1)=(ya+yc)/2;
 ye(3,1)=ya;
 ye(4,1)=(ya+yb)/2;
 %


 %for j=1:4
 %    qe(j,1)=(2*pi^2)*sin(pi*xe(j,1))*sin(pi*ye(j,1));
 %end
 %II =([  1/72, 7/864, 1/216, 7/864;...
 %     7/864,  1/54,  1/96, 1/216;...
  %    1/216,  1/96, 5/216,  1/96;...
   %   7/864, 1/216,  1/96,  1/54]);
 %f=(2*delabc)*(II*qe);
 %+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++



 f=[ 1/432; 1/216; 1/144; 1/216;  1/72; 5/216; 5/216;  1/72;  2/27]*(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
 %-------------------------------------------------------------------------
%bcdof=[13;37;35;33;31;29;27;25;23;21;19;17;15];
%apply boundary conditions
```

```
%
mm=length(bcdof);
sdof=size(ss);
%
for i=1:mm
c=bcdof(i,1);
for j=1:sdof
ss(c,j)=0;
end
%
ss(c,c)=1;
ff(c,1)=bcval(i,1);
end
%solve the equations

phi=ss\ff;
for I=1:nnode
NN(I,1)=I;
end

disp('__________________________________________________________________')
disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
disp('_________________________________________________________________________________
____')
disp('                         fem-computed values             anlytical(theoretical)-
values           ')

disp([NN phi xi])
disp('_________________________________________________________________________________
____')

disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
nodes
gcoord
[x,y]=meshgrid(-1/2:0.1:1/2,-1/2:0.1:1/2);

a0=8/pi^3;


for i=1:11
    for j=1:11
      for iel=1:nel
 %node numbers of quadrilateral
       nd1=nodes(iel,1);nd2=nodes(iel,2);nd3=nodes(iel,3);nd4=nodes(iel,4);
       nd5=nodes(iel,5);nd6=nodes(iel,6);nd7=nodes(iel,7);nd8=nodes(iel,8);
       nd9=nodes(iel,9);
 %coordinates of quadrilateral(u,v)

u(1,1)=gcoord(nd1,1);u(2,1)=gcoord(nd2,1);u(3,1)=gcoord(nd3,1);u(4,1)=gcoord(nd4,1);

v(1,1)=gcoord(nd1,2);v(2,1)=gcoord(nd2,2);v(3,1)=gcoord(nd3,2);v(4,1)=gcoord(nd4,2);
 %coordinates of the grid(x,y)

          in=inpolygon(x(i,j),y(i,j),u,v);
          if (in==1)
           X=x(i,j);Y=y(i,j);
           [t]=convexquadrilateral_coordinates(u,v,X,Y);
           r=t(1,1);
           s=t(2,1);
           shn1=(r^2-r)*(s^2-s)/4;
           shn2=(r^2+r)*(s^2-s)/4;
```

```
           shn3=(r^2+r)*(s^2+s)/4;
           shn4=(r^2-r)*(s^2+s)/4;
           shn5=(1-r^2)*(s^2-s)/2;
           shn6=(r^2+r)*(1-s^2)/2;
           shn7=(1-r^2)*(s^2+s)/2;
           shn8=(r^2-r)*(1-s^2)/2;
           shn9=(1-r^2)*(1-s^2);
%


%
PHI(i,j)=shn1*phi(nd1,1)+shn2*phi(nd2,1)+shn3*phi(nd3,1)+shn4*phi(nd4,1)+shn5*phi(nd5,1
)+shn6*phi(nd6,1)+shn7*phi(nd7,1)+shn8*phi(nd8,1)+shn9*phi(nd9,1);
              break
           end%if (in==1)
        end%for iel
   %THE PROGRAM EXECUTION JUMPS TO HERE if (in==1)
    end%for j
end%for i

a0=8/pi^3;
for ii=1:11
    for jj=1:11
    xx=(x(ii,jj));yy=(y(ii,jj));rr=(0);

for n=1:2:99
rr=rr+(-1)^((n-1)/2)*(1-(cosh(n*pi*yy)/cosh(n*pi/2)))*cos(n*pi*xx)/n^3;
end
z(ii,jj)=(a0*rr);
end %ii
end%jj

 % z=sin(pi*x).*sin(pi*y);

for i=1:11
    for j=1:11
        if (abs(PHI(i,j))<=1e-5)
            PHI(i,j)=0;
        end
        if (abs(z(i,j))<=1e-5)
            z(i,j)=0;
        end

    end
end
switch mesh

 case 4

    hold off
 clf
figure(1)
[x,y]=meshgrid(-1/2:.1:1/2,-1/2:0.1:1/2)
[c,h]=contour(x,y,PHI)
xlabel('X-axis');
ylabel('Y-axis');
clabel(c,h);
axis square
st1='Contour level curves for ';
st2='FEM solution of ';
st3='Nine Noded  ';
st4='Special Quadrilateral';
st5=' Elements'
title([st1,st2,st3,st4,st5])
sst1='(MESH HAS '
```

```
sst2=num2str(nnode)
sst3=' NODES'
sst4=' AND '
sst5=num2str(nel)
sst6=' ELEMENTS)'
text(0.25,-.08,[sst1 sst2 sst3 sst4 sst5 sst6])

figure(2)
%[x,y]=meshgrid(0:.1:1,0:0.1:1)
[x,y]=meshgrid(-1/2:.1:1/2,-1/2:0.1:1/2)
[c,h]=contour(x,y,z)
xlabel('X-axis');
ylabel('Y-axis');
clabel(c,h);
axis square
title('contour level curves for exact solution: in aseries')
 mm=0;
 for i=1:11
  for j=1:11
      mm=mm+1;
      femsoln(mm,1)=PHI(i,j);
      exactsoln(mm,1)=z(i,j);
  end
end
%+++++++++
hold off

figure(3)
[x,y]=meshgrid(-1/2:.1:1/2,-1/2:0.1:1/2)
%[x,y]=meshgrid(-sqrt(2)/2:(0.1)*sqrt(2)/2:sqrt(2)/2,-
sqrt(2)/2:(0.1)*sqrt(2)/2:sqrt(2)/2);
%[c,h]=contour(x,y,PHI,'r-')
contour(x,y,PHI,'r-')
xlabel('X-axis');
ylabel('Y-axis');
%clabel(c,h);
axis square
st1='Contour level curves for ';
st2='FEM(red)&exact(green) ';
st3='Nine Noded ';
st4='Special Quadrilateral';
st5=' Elements'
title([st1,st2,st3,st4,st5])
sst1='(MESH HAS '
sst2=num2str(nnode)
sst3=' NODES'
sst4=' AND '
sst5=num2str(nel)
sst6=' ELEMENTS)'
text(-1/2,-1/2,[sst1 sst2 sst3 sst4 sst5 sst6])
hold on
%[x,y]=meshgrid(0:.1:1,0:0.1:1)
%[c,h]=contour(x,y,z,'g-')
contour(x,y,z,'g-')
%xlabel('X-axis');
%ylabel('Y-axis');
%clabel(c,h);
axis square


end%switch mesh
%[femsoln exactsoln]

 disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
```

```
[1 phi(1,1) xi(1,1)]
disp('__________________________________________________________________')
disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
disp('_________________________________________________________________________________
____')
disp('                         fem-computed values             anlytical(theoretical)-
values           ')

disp([NN phi xi])
disp('_________________________________________________________________________________
____')

disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]

if mesh==4
%[phi xi]
for I=1:nnode
NN(I,1)=I;
phi_xi(I,1)=phi(I,1)-xi(I,1);
end
MAXPHI_XI=max(abs(double(phi_xi)));

t=0;
 for iii=1:nnode
    t=t+phi(iii,1)*ff(iii,1);
 end
 T=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('---------------------------------------------------------------------------------
------------')
end
disp('_________________________________________________________________________________
____')

disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
for ncpt=1:nel
    centrpt=nodes(ncpt,9)
    elcentr(ncpt,1)=centrpt;
    phicpt(ncpt,1)=phi(centrpt,1);
    xicpt(ncpt,1)=xi(centrpt,1);
end
for I=1:nel
elnumm(I,1)=I;
end
disp('_________________________________________________________________________________
____')
disp(' serial no        center point                fem-computed values
anlytical(theoretical)-values           ')

disp([elnumm  elcentr phicpt xicpt])
disp('_________________________________________________________________________________
____')
```

```
format compact
%======================================================================================
===========
disp('__________________________________________________________________')
disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
disp('_________________________________________________________________________________
_____________________________________________________')
disp('        NODE FEM SOLUTION EXACT SOLUTION       NODE FEM SOLUTION EXACT SOLUTION
NODE FEM SOLUTION EXACT SOLUTION          ')

disp('---------------------------------------------------------------------------------
------------------------------------------------------')
for I=1:3:nel
   % A=[elcentr(I) phicpt(I) xicpt(I)];B=[elcentr(I+1) phicpt(I+1)
xicpt(I+1)];C=[elcentr(I+2) phicpt(I+2) xicpt(I+2)];
    %disp([elcentr(I) phicpt(I) xicpt(I) elcentr(I+1) phicpt(I+1) xicpt(I+1)])
    %disp([elcentr(I+2) phicpt(I+2) xicpt(I+2)])
    %disp([A B C])
    fprintf('\n%5d %18.14f %18.14f %5d %18.14f %18.14f %5d %18.14f %18.14f',elcentr(I),
phicpt(I), xicpt(I), elcentr(I+1), phicpt(I+1), xicpt(I+1), elcentr(I+2), phicpt(I+2),
xicpt(I+2));
end
 fprintf('\n')
disp('---------------------------------------------------------------------------------
------------------------------------------------------')
```

MATLAB Codes-II

(1)quadrilateral_mesh4MOINEX_q9LG.m
(2)polygonal_domain_coordinates_2nd_orderLG.m
(3)nodaladdresses_special_convex_quadrilaterals_trial_2nd_orderLG.m
(4)generate_area_coordinate_over_the_standard_triangle.m
(5)glsampleptsweights.m
(6)D2PoissonEquationQ9MoinEx_MeshgridContour.m

%(1)quadrilateral_mesh4MOINEX_q9LG.m

```
function[]=quadrilateral_mesh4MOINEX_q9LG(n1,n2,n3,nmax,numtri,ndiv,mesh,xlength,ylengt
h)
clf
%(1)=generate 2-D quadrilateral mesh
%for a rectangular shape of domain
%quadrilateral_mesh_q4(xlength,ylength)
%xnode=number of nodes along x-axis
%ynode=number of nodes along y-axis
%xzero=x-coord of bottom left corner
%yzero=y-coord of bottom left corner
%xlength=size of domain alog x-axis
%ylength=size of domain alog y-axis
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,1,2,1,1
,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,4,4,1,1
,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9,1
,2,2,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9,4
,4,2,1,1)
%[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates(n1,n2,n3,nmax,numt
ri,ndiv,mesh)
%quadrilateral_mesh4MOINEX_q8([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9,1
,2,2,1,1)
%quadrilateral_mesh4MOINEX_q8([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,1,2,3,1
,1)
%[nel,nnel]=size(nodes);
```

```
%[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates_2nd_orderLG([1;1;1
;1],[2;3;4;5],[3;4;5;2],5,1,2,3)
%quadrilateral_mesh4MOINEX_q9LG([1;1;1;1],[2;3;4;5],[3;4;5;2],5,1,2,3,1,1)
%quadrilateral_mesh4MOINEX_q9LG([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,1,2,1
,1,1)
%quadrilateral_mesh4MOINEX_q9LG([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9
,1,2,2,1,1)
%quadrilateral_mesh4MOINEX_q9LG([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9
,1,2,4,1,1)
[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates_2nd_orderLG(n1,n2,n
3,nmax,numtri,ndiv,mesh)
[nel,nnel]=size(nodes);
disp([xlength,ylength,nnode,nel,nnel])
%gcoord(i,j),where i->node no. and j->x or y
%___________________________________________________________

%plot the mesh for the generated data
%x and y coordinates
xcoord(:,1)=gcoord(:,1);
ycoord(:,1)=gcoord(:,2);
%extract coordinates for each element
clf
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 tight
switch mesh
    case 1
axis([0 xlength 0 ylength])
case 2
axis([0 xlength 0 ylength])
    case 3
axis([0 xlength 0 ylength])
    case 4
axis([-xlength/2 xlength/2 -ylength/2 ylength/2])
end
figure(1)
plot(xvec,yvec);%plot element
hold on;
%place element number
midx=mean(xvec(1,1:4))
midy=mean(yvec(1,1:4))
if ndiv<=2
text(midx+.01,midy-.03,['[',num2str(i),']']);
end
end;%i loop
xlabel('x axis')
ylabel('y axis')
st1='Mesh of ';
st2=num2str(nel);
st3=' nine node ';
st4='quadrilateral';
st5=' elements&';
st6=num2str(nnode);
st7=',nodes'
title([st1,st2,st3,st4,st5,st6,st7])
%put node numbers
disp(nnode)
if ndiv<=2
for jj=1:nnode
text(gcoord(jj,1),gcoord(jj,2),['o',num2str(jj)]);
end
```

```
end
%axis off
if ndiv>=4
for jj=1:nnode
text(gcoord(jj,1),gcoord(jj,2),['o']);
end
end
hold on
figure(1),scatter(gcoord(:,1),gcoord(:,2),'MarkerFaceColor','g')
```

% (2)polygonal_domain_coordinates_2nd_orderLG.m

```
function[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates_2nd_orderLG
(n1,n2,n3,nmax,numtri,n,mesh)
%n1=node number at(0,0)for  a choosen triangle
%n2=node number at(1,0)for  a choosen triangle
%n3=node number at(0,1)for  a choosen triangle
%eln=6-node triangles with centroid
%spqd=4-node special convex quadrilateral
%n must be even,i.e.n=2,4,6,.......i.e number of divisions
%nmax=one plus the number of segments of the polygon
%nmax=the number of segments of the polygon plus a node interior to the polygon
%numtri=number of T6 triangles in each segment i.e a triangle formed by
%joining the end poits of the segment to the interior point(e.g:the centroid) of the
polygon
%[eln,spqd]=nodaladdresses_special_convex_quadrilaterals_trial(n1=1,n2=2,n3=3,nmax=3,n=
2,4,6,...)
%[eln,spqd]=nodaladdresses_special_convex_quadrilaterals_trial([1;1;1;1],[2;3;4;5],[3;4
;5;2],5,1,2)
%[eln,spqd]=nodaladdresses_special_convex_quadrilaterals_trial([1;1;1;1],[2;3;4;5],[3;4
;5;2],5,4,4)
%[eln,spqd]=nodaladdresses_special_convex_quadrilaterals_trial([1;1;1;1],[2;3;4;5],[3;4
;5;2],5,9,6)
%[eln,spqd]=nodaladdresses_special_convex_quadrilaterals_trial([1;1;1;1],[2;3;4;5],[3;4
;5;2],5,16,8)
%[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates_2nd_order([1;1;1;1
],[2;3;4;5],[3;4;5;2],5,1,2,1)
%[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates_2nd_order([1;1;1;1
;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9,1,2,2)
%PARVIZ MOIN EXAMPLE
%[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates_2nd_order([1;1;1;1
;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,1,2,1)
%[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates_2nd_order([1;1;1;1
;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,4,4,1)
%[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates_2nd_orderLG([1;1;1
;1],[2;3;4;5],[3;4;5;2],5,1,2,3)
syms U V W xi yi
syms x y
switch mesh
case 1%domain with seven triangles(8-nodes)
x=sym([1/2;1/2;1;  1;1/2;0;  0;0])%for MOIN EXAMPLE
y=sym([1/2;  0;0;1/2;  1;1;1/2;0])%for MOIN EXAMPLE
case 2%square domain with eight triangles(9-nodes)
x=sym([1/2;1/2;1;  1;  1;1/2;0;  0;0])%FOR UNIT SQUARE
y=sym([1/2;  0;0;1/2;  1;  1;1;1/2;0])%FOR UNIT SQUARE

case 3%square domain with four triangles(5-nodes)
 x=sym([1/2;0;1;1;0])
 y=sym([1/2;0;0;1;1])
 case 4%square domain with eight triangles(9-nodes)
%      1    2    3   4   5    6    7   8    9
x=sym([0;   0; 1/2;1/2;1/2;  0;-1/2;-1/2;-1/2])%FOR UNIT SQUARE
y=sym([0;-1/2;-1/2;  0;1/2;1/2; 1/2;   0;-1/2])%FOR UNIT SQUARE

end
[eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals_trial_2nd_ord
erLG(n1,n2,n3,nmax,numtri,n);
```

```
[U,V,W]=generate_area_coordinate_over_the_standard_triangle(n);

ss1='number of 6-node triangles with centroid=';
[p1,q1]=size(eln);
disp([ss1 num2str(p1)])
%
eln
%
ss2='number of special convex quadrilaterals elements&nodes per element =';
[nel,nnel]=size(spqd);
disp([ss2 num2str(nel) ','  num2str(nnel)])
%
spqd
%
nnode=max(max(spqd));
ss3='number of nodes of the triangular domain& number of special quadrilaterals=';
disp([ss3 num2str(nnode) ',' num2str(nel)])

xi(1:nnode,1)=zeros(nnode,1);yi(1:nnode,1)=zeros(nnode,1);

nitri=nmax-1;
for itri=1:nitri
disp('vertex nodes of the itri triangle')
[n1(itri,1) n2(itri,1) n3(itri,1)]
x1=x(n1(itri,1),1)
x2=x(n2(itri,1),1)
x3=x(n3(itri,1),1)
%
y1=y(n1(itri,1),1)
y2=y(n2(itri,1),1)
y3=y(n3(itri,1),1)
rrr(:,:,itri)
U'
V'
W'
kk=0;
for ii=1:n+1
    for jj=1:(n+1)-(ii-1)
        kk=kk+1;
        mm=rrr(ii,jj,itri);
        uu=U(kk,1);vv=V(kk,1);ww=W(kk,1);
        xi(mm,1)=x1*ww+x2*uu+x3*vv;
        yi(mm,1)=y1*ww+y2*uu+y3*vv;
    end%for jj
end%for ii
[xi yi]
%add coordinates of centroid
 ne=(n/2)^2;
% stdnode=kk;
 for iii=1+(itri-1)*ne:ne*itri
     %kk=kk+1;
     node1=eln(iii,1)
     node2=eln(iii,2)
     node3=eln(iii,3)
     mm=eln(iii,7)
     xi(mm,1)=(xi(node1,1)+xi(node2,1)+xi(node3,1))/3;
     yi(mm,1)=(yi(node1,1)+yi(node2,1)+yi(node3,1))/3;

 end %for iii
 [xi yi]

end%for itri=1:nitri
for mmm=1:nel
    mmm1=nodes(mmm,1)
    mmm2=nodes(mmm,2)
```

```
    mmm3=nodes(mmm,3)
    mmm4=nodes(mmm,4)
   mmm5=nodes(mmm,5)
   mmm6=nodes(mmm,6)
   mmm7=nodes(mmm,7)
   mmm8=nodes(mmm,8)
   mmm9=nodes(mmm,9)
   xi1=xi(mmm1,1)
   xi2=xi(mmm2,1)
   xi3=xi(mmm3,1)
   xi4=xi(mmm4,1)
   %(xi1+xi2)/2
   %
   yi1=yi(mmm1,1)
   yi2=yi(mmm2,1)
   yi3=yi(mmm3,1)
   yi4=yi(mmm4,1)
   %(yi1+yi2)/2
 xi(mmm5,1)=(xi1+xi2)/2;
 xi(mmm6,1)=(xi2+xi3)/2;
 xi(mmm7,1)=(xi3+xi4)/2;
 xi(mmm8,1)=(xi4+xi1)/2;
 xi(mmm9,1)=(xi1+xi2+xi3+xi4)/4;
 yi(mmm5,1)=(yi1+yi2)/2;
 yi(mmm6,1)=(yi2+yi3)/2;
 yi(mmm7,1)=(yi3+yi4)/2;
 yi(mmm8,1)=(yi4+yi1)/2;
 yi(mmm9,1)=(yi1+yi2+yi3+yi4)/4;
end%for nel
%[xi(18,1) yi(18,1)]

N=(1:nnode)'
[N xi yi]
%
coord(:,1)=(xi(:,1));
coord(:,2)=(yi(:,1));
gcoord(:,1)=double(xi(:,1));
gcoord(:,2)=double(yi(:,1));
%disp(gcoord)
```

(3) % **nodaladdresses_special_convex_quadrilaterals_trial_2nd_orderLG.m**

```
function[eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals_trial_2nd_orderLG(n1,n2,n3,nmax,numtri,n)
%n1=node number at(0,0)for  a choosen triangle
%n2=node number at(1,0)for  a choosen triangle
%n3=node number at(0,1)for  a choosen triangle
%eln=6-node triangles with centroid
%spqd=4-node special convex quadrilateral
%n must be even,i.e.n=2,4,6,.......i.e number of divisions
%nmax=one plus the number of segments of the polygon
%nmax=the number of segments of the polygon plus a node interior to the polygon
%numtri=number of T6 triangles in each segment i.e a triangle formed by
%joining the end poits of the segment to the interior point(e.g:the centroid) of the polygon
%[eln,spqd]=nodaladdresses_special_convex_quadrilaterals_trial(n1=1,n2=2,n3=3,nmax=3,n=2,4,6,...)
%[eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals_trial([1;1;1;1],[2;3;4;5],[3;4;5;2],5,1,2)
%[eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals_trial([1;1;1;1],[2;3;4;5],[3;4;5;2],5,4,4)
%[eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals_trial([1;1;1;1],[2;3;4;5],[3;4;5;2],5,9,6)
%[eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals_trial([1;1;1;1],[2;3;4;5],[3;4;5;2],5,16,8)
%PARVIZ MOIN EXAMPLE
```

```
%[eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals_trial([1;1;1
;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,1,2)
%[eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals_trial([1;1;1
;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,4,4)
%[eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals_trial_2nd_or
der([1;1;1;1],[2;3;4;5],[3;4;5;2],5,1,2)
%[eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals_trial_2nd_or
derLG([1;1;1;1],[2;3;4;5],[3;4;5;2],5,1,2)
%syms mst_tri x
ne=0;
nitri=nmax-1;
for itri=1:nitri
    elm(1:(n+1)*(n+2)/2,1)=zeros((n+1)*(n+2)/2,1)
elm(1,1)=n1(itri,1)
elm(n+1,1)=n2(itri,1)
elm((n+1)*(n+2)/2,1)=n3(itri,1)
disp('vertex nodes of the itri triangle')
[n1(itri,1) n2(itri,1) n3(itri,1)]
if itri==1
kk=nmax;
for k=2:n
    kk=kk+1
    elm(k,1)=kk
end
disp('base nodes=')
%elm(2:n)
edgen1n2(1:n+1,itri)=elm(1:n+1,1)
end%itri==1
if itri>1
    elm(1:n+1,1)=edgen1n3(1:n+1,itri-1);
end%if itri>1
if itri==1
    lmax=nmax+3*(n-1);
end%if itri==1
if (itri>1)&(itri<nitri)
    lmax=nmax+2*(n-1);
end% if (itri>1)&(itri<nitri)
mmax=nmax;
if itri==1
    mmax=max(max(edgen1n2(1:n+1,1)))
end%f itri==1
disp('right edge nodes')
nni=n+1;hh=1;qq(1,1)=n2(itri,1);
for i=0:(n-2)
    hh=hh+1;
    nni=nni+(n-i);
    elm(nni,1)=(mmax+1)+i;
    qq(hh,1)=(mmax+1)+i;

end
qq(n+1,1)=n3(itri,1);
edgen2n3(1:n+1,itri)=qq;

if itri<nitri
disp('left edge nodes')
nni=1;gg=1;pp(1,1)=n1(itri,1);
for i=0:(n-2)
    gg=gg+1;
    nni=nni+(n-i)+1;
    elm(nni,1)=lmax-i;
    pp(gg,1)=lmax-i;
end
pp(n+1,1)=n3(itri,1);
edgen1n3(1:n+1,itri)=pp
end%if itri<nitri
```

```
%if itri==n
% elm(1:n+1,1)=edgen1n2(1:n+1,1)
%end

if itri==nitri
disp('left edge nodes')
nni=1;gg=1;
for i=0:(n-2)
    gg=gg+1;
    nni=nni+(n-i)+1;
    elm(nni,1)=edgen1n2(gg,1);
end
%pp(n+1,1)=n3(itri,1);
%edgen1n3(1:n+1,itri)=pp
end%if itri==nitri
if itri==nitri
lmax=max(max(edgen2n3(1:n+1,itri)));
end%if itri==nitri




%elm
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)=lmax+jj;
        [nnj lmax+jj];
    end
end
%disp(elm);
%disp(length(elm));

jj=0;kk=0;
for j=0:n-1
    jj=j+1;
for k=1:(n+1)-j
    kk=kk+1;
    row_nodes(jj,k)=elm(kk,1);
end
end
row_nodes(n+1,1)=n3(itri,1);
%for jj=(n+1):-1:1
%    (row_nodes(jj,:));
%end
%[row_nodes]
rr=row_nodes;
rr
rrr(:,:,itri)=rr;
disp('element computations')
if rem(n,2)==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%j
%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%if(N-2)>0
end%k

end% if rem(n,2)==0
ne
%for kk=1:ne
%[eln(kk,1:6)]
%end
%add node numbers for element centroids

nnd=max(max(eln))
if (n>3)
for kkk=1+(itri-1)*numtri:ne
    nnd=nnd+1;
    eln(kkk,7)=nnd;
end
end
if n==2
 for kkk=itri:ne
    nnd=nnd+1;
    eln(kkk,7)=nnd;
 end
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
   %       spqd(mm,1:4)=[eln(iel,7) eln(iel,6) eln(iel,1) eln(iel,4)];
    %      nodes(mm,1:4)=spqd(mm,1:4);
     %     nodetel(mm,1:3)=[eln(iel,2) eln(iel,3) eln(iel,1)];
     %    case 2
     %     spqd(mm,1:4)=[eln(iel,7) eln(iel,4) eln(iel,2) eln(iel,5)];
      %    nodes(mm,1:4)=spqd(mm,1:4);
       %   nodetel(mm,1:3)=[eln(iel,3) eln(iel,1) eln(iel,2)];
       %   case 3
      %    spqd(mm,1:4)=[eln(iel,7) eln(iel,5) eln(iel,3) eln(iel,6)];
      %    nodes(mm,1:4)=spqd(mm,1:4);
      %    nodetel(mm,1:3)=[eln(iel,1) eln(iel,2) eln(iel,3)];
      %  end%switch
```

```
      %end
      %end
nmax=max(max(eln));
%nel=mm;
%
%ne
%spqd


end%itri

%to generate special quadrilaterals
mm=0;

for iel=1:ne
    for jel=1:3
    mm=mm+1;
        switch jel
         case 1
          spqd(mm,1:4)=[eln(iel,7) eln(iel,6) eln(iel,1) eln(iel,4)];
          nodes(mm,1:4)=spqd(mm,1:4);
          nodetel(mm,1:3)=[eln(iel,2) eln(iel,3) eln(iel,1)];
          case 2
          spqd(mm,1:4)=[eln(iel,7) eln(iel,4) eln(iel,2) eln(iel,5)];
          nodes(mm,1:4)=spqd(mm,1:4);
          nodetel(mm,1:3)=[eln(iel,3) eln(iel,1) eln(iel,2)];
          case 3
          spqd(mm,1:4)=[eln(iel,7) eln(iel,5) eln(iel,3) eln(iel,6)];
          nodes(mm,1:4)=spqd(mm,1:4);
          nodetel(mm,1:3)=[eln(iel,1) eln(iel,2) eln(iel,3)];
        end%switch
     end%for jel=1:3
   end


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%if
%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%if
%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%if
%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);
   nd=nd+1;
   nodes(mmm,9)=nd;
end
nnode=nd;
nel=mm;
spqd=nodes;
ss1='number of 6-node triangles with centroid=';
[p1,q1]=size(eln);
disp([ss1 num2str(p1)])
%
eln
%
ss2='number of special convex quadrilaterals elements&nodes per element =';
[nel,nnel]=size(spqd);
disp([ss2 num2str(nel) ','  num2str(nnel)])
%
nnode=max(max(spqd));
ss3='number of nodes of the triangular domain& number of special quadrilaterals=';
disp([ss3 num2str(nnode) ',' num2str(nel)])
```

%(4)generate_area_coordinate_over_the_standard_triangle.m

```
function[U,V,W]=generate_area_coordinate_over_the_standard_triangle(n)
syms ui vi wi U V W
kk=0;
for j=1:n+1
    for i=1:(n+1)-(j-1)
        kk=kk+1;
        ui(i,j)=(i-1)/n;
        vi(i,j)=(j-1)/n;
        wi(i,j)=1-ui(i,j)-vi(i,j);
        U(kk,1)=ui(i,j);
        V(kk,1)=vi(i,j);
        W(kk,1)=wi(i,j);
    end
end
%  ui
 % vi
  %wi
 % U'
  %V'
  %W'
 %kk
```

%(5)**glsampleptsweights.m**

```
function [s,wwww]=glsampleptsweights(n)
switch n
    case 1
        s(1)=0;wwww(1)=2;
    case 2
        table=[  .57735026918962576450914878050195,  1.000000000000000000000000000000
                -.57735026918962576450914878050195,  1.000000000000000000000000000000];
s=table(:,1);wwww=table(:,2);
case 3
            table=[  .77459666924148337703585307995650,  .55555555555555555555555555555571
                    -.77459666924148337703585307995650,  .55555555555555555555555555555571
                                   0,                                               8/9];
s=table(:,1);wwww=table(:,2);
case 4
    table=[   .86113631159405257522394648889280,  .34785484513745385737306394922182
             -.86113631159405257522394648889280,  .34785484513745385737306394922182
              .33998104358485626480266575910326,  .65214515486254614262693605077797
             -.33998104358485626480266575910326,  .65214515486254614262693605077797];
```

```
         s=table(:,1);wwww=table(:,2);
     case 5
     table=[                                 0,  .5688888888888888888888888888889
              .9061798459386639927976268782994,  .2369268850561890875142640407202
             -.9061798459386639927976268782994,  .2369268850561890875142640407202
              .5384693101056830910363144207002,  .4786286704993664680412915148356
             -.5384693101056830910363144207002,  .4786286704993664680412915148356];
         s=table(:,1);wwww=table(:,2);
     case 6
     table=[  .9324695142031520278123015544940,  .1713244923791703450402961421731
             -.9324695142031520278123015544940,  .1713244923791703450402961421731
              .2386191860831969086305017216807,  .4679139345726910473898703439895
             -.2386191860831969086305017216807,  .4679139345726910473898703439895
              .6612093864662645136613995950198,  .3607615730481386075698335138373
             -.6612093864662645136613995950198,  .3607615730481386075698335138373];
             s=table(:,1);wwww=table(:,2);
     case 7
      table=[                                 0,  .4179591836734693877551020408163
               .9491079123427585245261896840478,  .1294849661688696932706114326786
              -.9491079123427585245261896840478,  .1294849661688696932706114326786
               .4058451513773971669066064120769,  .3818300505051189449503697754890
              -.4058451513773971669066064120769,  .3818300505051189449503697754890
               .7415311855993944398638647732807,  .2797053914892766679014677714243
              -.7415311855993944398638647732807,  .2797053914892766679014677714243];
                s=table(:,1);wwww=table(:,2);
     case 8
               table=[  .9602898564975362316835608685694,  .1012285362903762591525313543105
                       -.9602898564975362316835608685694,  .1012285362903762591525313543105
                        .7966664774136267395915539364758,  .2223810344533744705443559944278
                       -.7966664774136267395915539364758,  .2223810344533744705443559944278
                        .5255324099163289858177390491892,  .3137066458778872873379622019867
                       -.5255324099163289858177390491892,  .3137066458778872873379622019867
                        .1834346424956498049394761423601,  .3626837833783619829651504492771
                       -.1834346424956498049394761423601,  .3626837833783619829651504492771];

                         s=table(:,1);wwww=table(:,2);
     case 9
         table=[                                   0,    .3302393550012597631645250692869
                    .9681602395076260898355762029036, .8127438836157441197189215811499e-1
                   -.9681602395076260898355762029036, .8127438836157441197189215811499e-1
                    .8360311073266357942994297880697,    .1806481606948574040584720312462
                   -.8360311073266357942994297880697,    .1806481606948574040584720312462
                    .6133714327005903973087020393414,    .2606106964029354623187428694187
                   -.6133714327005903973087020393414,    .2606106964029354623187428694187
                    .3242534234038089290385380146433,    .3123470770400028400686304065846
                   -.3242534234038089290385380146433,    .3123470770400028400686304065846];

                             s=table(:,1);wwww=table(:,2);
   case 10
       table=[   -.1488743389816312108848260011297,    .2955242247147528701738929946513
                  .1488743389816312108848260011297,    .2955242247147528701738929946513
                 -.4333953941292471907992659431657,    .2692667193099963550912269215693
                  .4333953941292471907992659431657,    .2692667193099963550912269215693
                 -.6794095682990244062343273651148,    .2190863625159820439955349342295
                  .6794095682990244062343273651148,    .2190863625159820439955349342295
                 -.8650633666889845107320966884235,    .1494513491505805931457763396548
                  .8650633666889845107320966884235,    .1494513491505805931457763396548
                 -.9739065285171717200779640120844, .6667134430868813759356880989621e-1
                  .9739065285171717200779640120844, .6667134430868813759356880989621e-1];

   s=table(:,1);wwww=table(:,2);
end
```

%(6)D2PoissonEquationQ9MoinEx_MeshgridContour.m

```
function[]=D2PoissonEquationQ9MoinEx_MeshgridContour(n1,n2,n3,nmax,numtri,ndiv,mesh,fcn
,ng)

%note that input vlues of X and Y must be symbolic constants
%for the example triangle input for X is sym([-1/2 1/2 0])
%for the example triangle input for Y is sym([0 0 sqrt(3/4)])
%LaplaceEquationQ4twoD(3,sym([-1/2 1/2 0]),sym([0 0 sqrt(3/4)]))
%ndiv=2,4,6,8,......
%polygonal_domain_coordinates([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,1,2)
%polygonal_domain_coordinates([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,4,4)
%D2LaplaceEquationQ4MoinExautomeshgen(n1,n2,n3,nmax,numtri,ndiv)
```

```
%D2LaplaceEquationQ4MoinExautomeshgen([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8
,1,2,1)
%D2LaplaceEquationQ4MoinExautomeshgen([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8
,4,4,1)
%D2LaplaceEquationQ4MoinExautomeshgen([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;
9;2],9,1,2,2)
%D2LaplaceEquationQ4MoinExautomeshgen([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;
9;2],9,4,4,2)
%quadrilateral_mesh4MOINEX_q4(n1,n2,n3,nmax,numtri,ndiv,mesh,xlength,ylength)([1;1;1;1;
1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9,1,2,2,1,1)
%D2POISSONEQUATION_NODALINTERPOLATION_VALUES(n1,n2,n3,nmax,numtri,ndiv,mesh)([1;1;1;1;1
;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9,1,2,2)
%D2LaplaceEquationQ4MoinExautomeshgen([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;
9;2],9,100,20,2)
%D2PoissonEquationQ4MoinEx_MeshgridContour([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6
;7;8;9;2],9,1,2,2)
%D2PoissonEquationQ4MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,1,2,1)
%D2PoissonEquationQ4MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,4,4,1)
%D2PoissonEquationQ4MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,9,6,1)
%D2PoissonEquationQ4MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,16,8,1)
%D2PoissonEquationQ4MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,25,10,1)
%D2PoissonEquationQ8MoinEx_MeshgridContour([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6
;7;8;9;2],9,1,2,2)
%D2PoissonEquationQ8MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,1,2,1)
%D2PoissonEquationQ9MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,1,2,1)
%D2PoissonEquationQ9MoinEx_MeshgridContour([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6
;7;8;9;2],9,1,2,2)
%D2PoissonEquationQ9MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8
;2],8,1,2,1)
%%D2PoissonEquationQ9MoinEx_MeshgridContour([1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;
8;2],8,1,2,1,1,10)
%D2PoissonEquationQ9MoinEx_MeshgridContour([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6
;7;8;9;2],9,1,2,2,1,10)

syms coord
%[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates(n1,n2,n3,nmax,numt
ri,ndiv,mesh)
%nnel=4;
[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates_2nd_orderLG(n1,n2,n
3,nmax,numtri,ndiv,mesh)
nnel=9;
ndof=1;
%nc=(ndiv/2)^2;
%nnode=(ndiv+1)*(ndiv+2)/2+nc;
%nel=3*nc;
sdof=nnode*ndof;
ff=(zeros(sdof,1));ss=(zeros(sdof,sdof));
clf
clc
%nnode=17,nel=12,nnel=4,ndof=1
%>>LaplaceEquationQuad4twodimension(12,17,4,1)
%
%Ex1:nnode=41,nel=36,,nnel=4,nodf=1
%>>LaplaceEquationQuad4twodimensionEx1(36,41,4,1)
%>>improvedLaplaceEquationQuad4twodimensionEx1_explicit(36,41,4,1)
%Ex2:nnode=83,nel=69,,nnel=4,nodf=1
%>>improvedLaplaceEquationQuad4twodimensionEx2_explicit(69,83,4,1)#
%>>improvedLaplaceEquationQuad4twodimensionEx2_explicitfnmesh(69,83,4,1)#
%improvedLaplaceEquationQuad4twodimensionEx2_explicitvfnmesh(72,87,4,1)#new
```

```
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(nel=3,nnode=7,nnel=4,ndof=1)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(nel,nnode,nnel=4,ndof=1,quadt
ype=0/3,mesh=1,2,3...)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(nel=12,nnode=19,nnel=4,ndof=1
,quadtype=0/3,mesh=3)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(nel=27,nnode=37,nnel=4,ndof=1
,quadtype=0/3,mesh=4)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(nel=48,nnode=61,nnel=4,ndof=1
,quadtype=0/3,mesh=5)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(nel=75,nnode=91,nnel=4,ndof=1
,quadtype=0/3,mesh=6)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(108,127,4,1,3,7)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(147,169,4,1,3,8)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(192,217,4,1,3,9)
%improvedLaplaceEquationQuad4twodimensionEx3_explicitmesh(243,271,4,1,3,10)
disp([nel nnode nnel ndof])
format long g
for i=1:nel
N(i,1)=i;
end
for i=1:nel
NN(i,1)=i;
end
    %[coord,gcoord]=coordinate_rtisoscelestriangle00_h0_hh(ndiv);
    %[nodetel,nodes]=nodaladdresses4special_convex_quadrilaterals(ndiv)
    %
    %bcdof=[2;5;3]
   %boundary conditions-1
   switch mesh
       case 1
    nnn=0;
    for nn=1:nnode
        xnn=gcoord(nn,1);ynn=gcoord(nn,2);
        if (xnn==0)&((ynn>=0)&(ynn<=1))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
    end
    %boundary conditions-2
     for nn=1:nnode
        xnn=gcoord(nn,1);ynn=gcoord(nn,2);
        if (ynn==0)&((xnn>=0)&(xnn<=1))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
    end
    %boundary conditions-3
    for nn=1:nnode
        xnn=gcoord(nn,1);ynn=gcoord(nn,2);
        if (ynn==1)&((xnn>=0)&(xnn<=1/2))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
    end
   %boundary conditions-4
   for nn=1:nnode
      xnn=gcoord(nn,1);ynn=gcoord(nn,2);
        if (xnn==1)&((ynn>=0)&(ynn<=1/2))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
    end
```

```
  %boundary conditions-5
  for nn=1:nnode
      xnn=coord(nn,1);ynn=coord(nn,2);
        if ((xnn+ynn)==3/2)
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=double((sin(pi*xnn))*(sin(pi*ynn)))
        end
  end
    bcdof
    mm=length(bcdof);

format long g
%analytical solution

 xi=(zeros(nnode,1));
for m=1:nnode
    xm=(gcoord(m,1));ym=(gcoord(m,2));
    xi(m,1)=sin(pi*xm)*sin(pi*ym);
end



case 2
      nnn=0;
    for nn=1:nnode
        xnn=coord(nn,1);ynn=gcoord(nn,2);
        if (xnn==0)&((ynn>=0)&(ynn<=1))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
    end
    %boundary conditions-2
     for nn=1:nnode
        xnn=gcoord(nn,1);ynn=coord(nn,2);
        if (ynn==0)&((xnn>=0)&(xnn<=1))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
    end
    %boundary conditions-3
    for nn=1:nnode
        xnn=gcoord(nn,1);ynn=coord(nn,2);
        if (ynn==1)&((xnn>=0)&(xnn<=1))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
    end
   %boundary conditions-4
   for nn=1:nnode
      xnn=coord(nn,1);ynn=gcoord(nn,2);
        if (xnn==1)&((ynn>=0)&(ynn<=1))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
    end


%end
    bcdof
    mm=length(bcdof);
```

```
format long g
%analytical solution

 xi=(zeros(nnode,1));
for m=1:nnode
    xm=(gcoord(m,1));ym=(gcoord(m,2));
    xi(m,1)=sin(pi*xm)*sin(pi*ym);
end


   end %switch
 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
%_________________________________________________________________
%ng=10
 [sp,wt]=glsampleptsweights(ng)
table2=[N xx yy];
%disp([xx yy])
%______________________________________________________________________
```

```
intJdn1dn1uvrs =[vpa(sym('  .68800369699992917040751928889')) vpa(sym(' .63026863732902919654693031748'));...
vpa(sym(' .63026863732902919654693031748')) vpa(sym('  .68800369699992917040751928889'))];

intJdn1dn2uvrs =[vpa(sym('  .49469766679465431322423128849e-1')) vpa(sym(' -.92492499790138905674372588873e-1'));...
vpa(sym('  .74174166876527760992294077793e-1')) vpa(sym(' -.58041502241256898516961218830e-1'))];

intJdn1dn3uvrs =[vpa(sym(' -.27639187071864196478244612770e-1')) vpa(sym(' -.36406227705135716084660539928e-1'));...
vpa(sym(' -.36406227705135716084660539928e-1')) vpa(sym(' -.27639187071864196478244612770e-1'))];

intJdn1dn4uvrs =[vpa(sym(' -.58041502241256898516961218830e-1')) vpa(sym('  .74174166876527760992294077793e-1'));...
vpa(sym(' -.92492499790138905674372588873e-1')) vpa(sym('  .49469766679465431322423128849e-1'))];

intJdn1dn5uvrs =[vpa(sym(' -.42034020155248898256147959993')) vpa(sym('  .23835534506556861836117847003'));...
vpa(sym(' -.42831132160109804830548819664')) vpa(sym('  .10203781293970562330870803320'))];

intJdn1dn6uvrs =[vpa(sym(' .12626870005164588340746772853')) vpa(sym(' .16865169113202404137334892156'));...
vpa(sym(' .16865169113202404137334892156')) vpa(sym(' .15224655673422927054938212414'))];

intJdn1dn7uvrs =[vpa(sym(' .15224655673422927054938212414')) vpa(sym(' .16865169113202404137334892156'));...
vpa(sym(' .16865169113202404137334892156')) vpa(sym(' .12626870005164588340746772853'))];

intJdn1dn8uvrs =[vpa(sym('  .10203781293970562330870803320')) vpa(sym(' -.42831132160109804830548819664'));...
vpa(sym('  .23835534506556861836117847003')) vpa(sym(' -.42034020155248898256147959993'))];

intJdn1dn9uvrs =[vpa(sym(' -.61200564253936530143881487716')) vpa(sym(' -.72289148243880098858257938847'));...
vpa(sym(' -.72289148243880098858257938847')) vpa(sym(' -.61200564253936530143881487716'))];

intJdn2dn1uvrs =[vpa(sym('  .49469766679465431322423128849e-1')) vpa(sym('  .74174166876527760992294077793e-1'));...
vpa(sym(' -.92492499790138905674372588873e-1')) vpa(sym(' -.58041502241256898516961218830e-1'))];

intJdn2dn2uvrs =[vpa(sym('  .24822242723570359859415910165')) vpa(sym(' -.13943577781062161645243966287'));...
vpa(sym(' -.13943577781062161645243966287')) vpa(sym('  .22525137401606776447455758469'))];

intJdn2dn3uvrs =[vpa(sym('  -.93683290648218768462051538669e-1')) vpa(sym(' -.82852283681797324065968279846e-1'));...
vpa(sym('  .83814382984869342600698386820e-1')) vpa(sym('   .37904800898126436475032582264e-1'))];

intJdn2dn4uvrs =[vpa(sym('   .62925700025302140451751960e-3')) vpa(sym(' .26100052047122979851449223466e-1'));...
vpa(sym(' .26100052047122979851449223466e-1')) vpa(sym('   .62925700025302140451751960e-3'))];

intJdn2dn5uvrs =[vpa(sym('   -.46868216752790132335135919662')) vpa(sym('  -.34646194452109534302782806358'));...
vpa(sym('   .32020472214557132363883860309')) vpa(sym(' .85836729209541381287229242888e-1'))];

intJdn2dn6uvrs =[vpa(sym('   .20486711030201898764584585532')) vpa(sym('  .25718774442435573105221634722'));...
vpa(sym(' -.40947892224231093561445031945')) vpa(sym(' -.28073862677651621463393803502'))];

intJdn2dn7uvrs =[vpa(sym(' .50885917641966498462888866644e-1')) vpa(sym('  -.10335794828229434487750587371'));...
vpa(sym('  -.10335794828229434487750587371')) vpa(sym(' -.17609016019421764322513358 0e-2'))];
```

```
intJdn2dn8uvrs =[vpa(sym(' -.41736856619169902011341267580e-1')) vpa(sym(' -.94765689307712237483661393070e-1'));...
vpa(sym(' -.94765689307712237483661393070e-1')) vpa(sym('  .33881015844374607003400451610e-1'))];

intJdn2dn9uvrs =[vpa(sym('   .50027835935882456394917533200e-1')) vpa(sym('    .40941168025551439401144362445'));...
vpa(sym('    .40941168025551439401144362445')) vpa(sym(' -.42962146348647921061586792260e-1'))];

intJdn3dn1uvrs =[vpa(sym(' -.27639187071864196478244612700e-1')) vpa(sym(' -.36406227705135716084660539280e-1'));...
vpa(sym(' -.36406227705135716084660539280e-1')) vpa(sym(' -.27639187071864196478244612700e-1'))];

intJdn3dn2uvrs =[vpa(sym('  -.93683290648218768462051538690e-1')) vpa(sym('  .83814382984869342600698386820e-1'));...
vpa(sym(' -.82852283681797324065968279846e-1')) vpa(sym('   .37904800898126436475032582640e-1'))];

intJdn3dn3uvrs =[vpa(sym('  .33203856905137126709551224250')) vpa(sym(' .29812794600013651335576030277 1'));...
vpa(sym(' .29812794600013651335576030277 1')) vpa(sym('  .33203856905137126709551224250'))];

intJdn3dn4uvrs =[vpa(sym('   .37904800898126436475032582640e-1')) vpa(sym(' -.82852283681797324065968279846e-1'));...
vpa(sym('  .83814382984869342600698386820e-1')) vpa(sym('  -.93683290648218768462051538690e-1'))];

intJdn3dn5uvrs =[vpa(sym('   .15710844821544667058002407565')) vpa(sym('   .12945549547911074908308207668'));...
vpa(sym('   .12945549547911074908308207668')) vpa(sym(' .43059644229262196507731226970e-1'))];

intJdn3dn6uvrs =[vpa(sym('  .14159217838663875388665351332')) vpa(sym(' -.35360191630747700837476451283'));...
vpa(sym('  .31306475035918965829190215383')) vpa(sym(' -.34086853405320051524855521401'))];

intJdn3dn7uvrs =[vpa(sym(' -.34086853405320051524855521401')) vpa(sym('  .31306475035918965829190215383'));...
vpa(sym(' -.35360191630747700837476451283')) vpa(sym('  .14159217838663875388665351332'))];

intJdn3dn8uvrs =[vpa(sym(' .43059644229262196507731226970e-1')) vpa(sym('   .12945549547911074908308207668'));...
vpa(sym('   .12945549547911074908308207668')) vpa(sym('   .15710844821544667058002407565'))];

intJdn3dn9uvrs =[vpa(sym('  -.24951262900756184435610227570')) vpa(sym(' -.48105764260800696388913166479'));...
vpa(sym(' -.48105764260800696388913166479')) vpa(sym('  -.24951262900756184435610227570'))];

intJdn4dn1uvrs =[vpa(sym(' -.58041502241256898516961218300e-1')) vpa(sym(' -.92492499790138905674372588730e-1'));...
vpa(sym('  .74174166876527760992294077930e-1')) vpa(sym('  .49469766679465431322423128490e-1'))];

intJdn4dn2uvrs =[vpa(sym('   .62925700025302140451751960e-3')) vpa(sym(' .26100052047122979851449223460e-1'));...
vpa(sym(' .26100052047122979851449223460e-1')) vpa(sym('   .62925700025302140451751960e-3'))];

intJdn4dn3uvrs =[vpa(sym('   .37904800898126436475032582640e-1')) vpa(sym('  .83814382984869342600698386820e-1'));...
vpa(sym(' -.82852283681797324065968279846e-1')) vpa(sym('  -.93683290648218768462051538690e-1'))];

intJdn4dn4uvrs =[vpa(sym('  .22525137401606776447455758469')) vpa(sym(' -.13943577781062161645243966287'));...
vpa(sym(' -.13943577781062161645243966287')) vpa(sym('  .24822242723570359859415910165'))];

intJdn4dn5uvrs =[vpa(sym('  .33881015844374607003400451610e-1')) vpa(sym(' -.94765689307712237483661393070e-1'));...
vpa(sym(' -.94765689307712237483661393070e-1')) vpa(sym(' -.41736856619169902011341267580e-1'))];

intJdn4dn6uvrs =[vpa(sym(' -.17609016019421764322513358000e-2')) vpa(sym('  -.10335794828229434487750587371'));...
vpa(sym('  -.10335794828229434487750587371')) vpa(sym(' .50885917641966498462888866440e-1'))];

intJdn4dn7uvrs =[vpa(sym(' -.28073862677651621463393803502')) vpa(sym(' -.40947892224231093561445031945'));...
vpa(sym('  .25718774442435573105221634722')) vpa(sym('   .20486711030201898764584585320'))];

intJdn4dn8uvrs =[vpa(sym(' .85836729209541381287229242880e-1')) vpa(sym('   .32020472214557132363883860309'));...
vpa(sym('  -.34646194452109534302782806358')) vpa(sym('   -.46868216752790132335135919620'))];

intJdn4dn9uvrs =[vpa(sym(' -.42962146348647921061586792260e-1')) vpa(sym('    .40941168025551439401144362445'));...
vpa(sym('    .40941168025551439401144362445')) vpa(sym('   .50027835935882456394917533200e-1'))];

intJdn5dn1uvrs =[vpa(sym(' -.42034020155248898256147959930')) vpa(sym(' -.42831132160109804830548819640'));...
vpa(sym('  .23835534506556861836117847030')) vpa(sym('  .10203781293970562330870803200'))];

intJdn5dn2uvrs =[vpa(sym('   -.46868216752790132335135919620')) vpa(sym('   .32020472214557132363883860309'));...
vpa(sym('  -.34646194452109534302782806358')) vpa(sym(' .85836729209541381287229242880e-1'))];

intJdn5dn3uvrs =[vpa(sym('   .15710844821544667058002407565')) vpa(sym('   .12945549547911074908308207668'));...
vpa(sym('   .12945549547911074908308207668')) vpa(sym(' .43059644229262196507731226970e-1'))];

intJdn5dn4uvrs =[vpa(sym('  .33881015844374607003400451610e-1')) vpa(sym(' -.94765689307712237483661393070e-1'));...
vpa(sym(' -.94765689307712237483661393070e-1')) vpa(sym(' -.41736856619169902011341267580e-1'))];

intJdn5dn5uvrs =[vpa(sym(' 1.0663384833344564856750376651')) vpa(sym('  .56556859863119073241392390190'));...
vpa(sym('  .56556859863119073241392390190')) vpa(sym(' 1.1211333295329756651950424278'))];

intJdn5dn6uvrs =[vpa(sym('  -.58306188548401601151324941690')) vpa(sym(' -.54252818687130357695788942932'));...
vpa(sym(' -.54252818687130357695788942932')) vpa(sym(' -.22579554943131879593427777753'))];

intJdn5dn7uvrs =[vpa(sym('   -.16767681050348731002396257420')) vpa(sym(' -.17096478113284630633060979200e-2'));...
vpa(sym(' -.17096478113284630633060979200e-2')) vpa(sym('   .17132457681637017904834081102'))];

intJdn5dn8uvrs =[vpa(sym(' .90271486053820367920675971200e-1')) vpa(sym('   .41024643803333829631989611840'));...
vpa(sym('   .41024643803333829631989611840')) vpa(sym(' .90271486053820367920675971200e-1'))];
```

```
intJdn5dn9uvrs =[vpa(sym('   .29216163161979549627091262 30')) vpa(sym('  -.358160408697768775645395583 6'));...
vpa(sym('  -.35816040869776877564539558 36')) vpa(sym(' -1.3461311727311867153221086666'))];

intJdn6dn1uvrs =[vpa(sym(' .12626870005164588340746772853')) vpa(sym(' .16865169113202404137334892156'));...
vpa(sym(' .16865169113202404137334892156')) vpa(sym(' .15224655673422927054938212414'))];

intJdn6dn2uvrs =[vpa(sym('   .20486711030201898764584585 32')) vpa(sym(' -.40947892224231093561445031945'));...
vpa(sym('  .25718774442435573105221634722')) vpa(sym(' -.28073862677651621463393803502'))];

intJdn6dn3uvrs =[vpa(sym('  .14159217838663875388665351332')) vpa(sym('  .31306475035918965829190215383'));...
vpa(sym(' -.35360191630747700837476451283')) vpa(sym(' -.34086853405320051524855521401'))];

intJdn6dn4uvrs =[vpa(sym(' -.17609016019421764322513358 0e-2')) vpa(sym('  -.10335794828229434487750587371'));...
vpa(sym('  -.10335794828229434487750587371')) vpa(sym(' .50885917641966498462888866 44e-1'))];

intJdn6dn5uvrs =[vpa(sym('  -.58306188548401601151324941 69')) vpa(sym(' -.54252818687130357695788942932'));...
vpa(sym(' -.54252818687130357695788942932')) vpa(sym(' -.22579554943131879593427777753'))];

intJdn6dn6uvrs =[vpa(sym(' 1.5998510858095308546375928228')) vpa(sym(' .24469327225090109904911663174'));...
vpa(sym(' .24469327225090109904911663174')) vpa(sym('  .64675514174853057100283264 58'))];

intJdn6dn7uvrs =[vpa(sym(' -.74612686016266341101136360 5e-1')) vpa(sym('   .42496611860422613701397774221'));...
vpa(sym('   .42496611860422613701397774221')) vpa(sym(' -.74612686016266341101136360 5e-1'))];

intJdn6dn8uvrs =[vpa(sym('   .17132457681637017904834081102')) vpa(sym(' -.17096478113284630633060979 2e-2'));...
vpa(sym(' -.17096478113284630633060979 2e-2')) vpa(sym('   -.16767681050348731002396257 42'))];

intJdn6dn9uvrs =[vpa(sym('   -1.5844681782639801295792636156')) vpa(sym(' -.94301127139103615215193728 96e-1'));...
vpa(sym(' -.94301127139103615215193728 96e-1')) vpa(sym('    .23980459065606283692676632 48'))];

intJdn7dn1uvrs =[vpa(sym(' .15224655673422927054938212414')) vpa(sym(' .16865169113202404137334892156'));...
vpa(sym(' .16865169113202404137334892156')) vpa(sym(' .12626870005164588340746772853'))];

intJdn7dn2uvrs =[vpa(sym(' .50885917641966498462888866 44e-1')) vpa(sym('  -.10335794828229434487750587371'));...
vpa(sym('  -.10335794828229434487750587371')) vpa(sym(' -.17609016019421764322513358 0e-2'))];

intJdn7dn3uvrs =[vpa(sym(' -.34086853405320051524855521401')) vpa(sym(' -.35360191630747700837476451283'));...
vpa(sym('  .31306475035918965829190215383')) vpa(sym('  .14159217838663875388665351332'))];

intJdn7dn4uvrs =[vpa(sym(' -.28073862677651621463393803502')) vpa(sym('  .25718774442435573105221634722'));...
vpa(sym(' -.40947892224231093561445031945')) vpa(sym('   .20486711030201898764584585 32'))];

intJdn7dn5uvrs =[vpa(sym('   -.16767681050348731002396257 42')) vpa(sym(' -.17096478113284630633060979 2e-2'));...
vpa(sym(' -.17096478113284630633060979 2e-2')) vpa(sym('   .17132457681637017904834081102'))];

intJdn7dn6uvrs =[vpa(sym(' -.74612686016266341101136360 5e-1')) vpa(sym('   .42496611860422613701397774221'));...
vpa(sym('   .42496611860422613701397774221')) vpa(sym(' -.74612686016266341101136360 5e-1'))];

intJdn7dn7uvrs =[vpa(sym('  .64675514174853057100283264 58')) vpa(sym(' .24469327225090109904911663174'));...
vpa(sym(' .24469327225090109904911663174')) vpa(sym(' 1.5998510858095308546375928228'))];

intJdn7dn8uvrs =[vpa(sym(' -.22579554943131879593427777753')) vpa(sym(' -.54252818687130357695788942932'));...
vpa(sym(' -.54252818687130357695788942932')) vpa(sym('  -.58306188548401601151324941 69'))];

intJdn7dn9uvrs =[vpa(sym('    .23980459065606283692676632 48')) vpa(sym(' -.94301127139103615215193728 96e-1'));...
vpa(sym(' -.94301127139103615215193728 96e-1')) vpa(sym('   -1.5844681782639801295792636156'))];

intJdn8dn1uvrs =[vpa(sym('  .10203781293970562330870803 20')) vpa(sym('  .23835534506556861836117847 03'));...
vpa(sym(' -.42831132160109804830548819 64')) vpa(sym(' -.42034020155248898256147959 93'))];

intJdn8dn2uvrs =[vpa(sym(' -.41736856619169902011341267 58e-1')) vpa(sym(' -.94765689307712237483661393 07e-1'));...
vpa(sym(' -.94765689307712237483661393 07e-1')) vpa(sym('  .33881015844374607003400451 61e-1'))];

intJdn8dn3uvrs =[vpa(sym(' .43059644229262196507731226 97e-1')) vpa(sym('   .12945549547911074908308207668'));...
vpa(sym('   .12945549547911074908308207668')) vpa(sym('   .15710844821544667058002407565'))];

intJdn8dn4uvrs =[vpa(sym(' .85836729209541381287229242 88e-1')) vpa(sym('  -.34646194452109534302782806358'));...
vpa(sym('   .32020472214557132363883860309')) vpa(sym('   -.46868216752790132335135919 62'))];

intJdn8dn5uvrs =[vpa(sym(' .90271486053820367920675971 2e-1')) vpa(sym('   .41024643803333829631989611 84'));...
vpa(sym('   .41024643803333829631989611 84')) vpa(sym(' .90271486053820367920675971 2e-1'))];

intJdn8dn6uvrs =[vpa(sym('   .17132457681637017904834081102')) vpa(sym(' -.17096478113284630633060979 2e-2'));...
vpa(sym(' -.17096478113284630633060979 2e-2')) vpa(sym('   -.16767681050348731002396257 42'))];

intJdn8dn7uvrs =[vpa(sym(' -.22579554943131879593427777753')) vpa(sym(' -.54252818687130357695788942932'));...
vpa(sym(' -.54252818687130357695788942932')) vpa(sym('  -.58306188548401601151324941 69'))];

intJdn8dn8uvrs =[vpa(sym(' 1.1211333295329756651950424278')) vpa(sym('  .56556859863119073241392390 19'));...
vpa(sym('  .56556859863119073241392390 19')) vpa(sym(' 1.0663384833344564856750376651'))];

intJdn8dn9uvrs =[vpa(sym(' -1.3461311727311867153221086666')) vpa(sym('  -.35816040869776877564539558 36'));...
vpa(sym('  -.35816040869776877564539558 36')) vpa(sym('   .29216163161979549627091262 30'))];
```

```
intJdn9dn1uvrs =[vpa(sym(' -.6120056425393653014388148716')) vpa(sym(' -.7228914824388009885825793847'));...
vpa(sym(' -.7228914824388009885825793847')) vpa(sym(' -.6120056425393653014388148716'))];

intJdn9dn2uvrs =[vpa(sym('   .500278359358824563949175332e-1')) vpa(sym('    .40941168025551439401144362445'));...
vpa(sym('    .40941168025551439401144362445')) vpa(sym(' -.4296214634864792106158679226e-1'))];

intJdn9dn3uvrs =[vpa(sym('  -.2495126290075618443561022757')) vpa(sym(' -.4810576426080069638891316479'));...
vpa(sym(' -.4810576426080069638891316479')) vpa(sym('  -.2495126290075618443561022757'))];

intJdn9dn4uvrs =[vpa(sym(' -.4296214634864792106158679226e-1')) vpa(sym('    .40941168025551439401144362445'));...
vpa(sym('    .40941168025551439401144362445')) vpa(sym('   .500278359358824563949175332e-1'))];

intJdn9dn5uvrs =[vpa(sym('   .2921616316197954962709126230')) vpa(sym('  -.3581604086977687756453955836'));...
vpa(sym('  -.3581604086977687756453955836')) vpa(sym(' -1.3461311727311867153221086666'))];

intJdn9dn6uvrs =[vpa(sym('   -1.5844681782639801295792636156')) vpa(sym(' -.9430112713910361521519372896e-1'));...
vpa(sym(' -.9430112713910361521519372896e-1')) vpa(sym('     .2398045906560628369267663248'))];

intJdn9dn7uvrs =[vpa(sym('     .2398045906560628369267663248')) vpa(sym(' -.9430112713910361521519372896e-1'));...
vpa(sym(' -.9430112713910361521519372896e-1')) vpa(sym('   -1.5844681782639801295792636156'))];

intJdn9dn8uvrs =[vpa(sym(' -1.3461311727311867153221086666')) vpa(sym('  -.3581604086977687756453955836'));...
vpa(sym('  -.3581604086977687756453955836')) vpa(sym('   .2921616316197954962709126230'))];

intJdn9dn9uvrs =[vpa(sym(' 3.2530857106790011221652797406')) vpa(sym(' 1.2900488362095239461700024257'));...
vpa(sym(' 1.2900488362095239461700024257')) vpa(sym(' 3.2530857106790011221652797406'))];


%integrals of products of glbal derivatives
intJdndn=double([intJdn1dn1uvrs intJdn1dn2uvrs intJdn1dn3uvrs intJdn1dn4uvrs intJdn1dn5uvrs intJdn1dn6uvrs intJdn1dn7uvrs intJdn1dn8uvrs intJdn1dn9uvrs ;...
          intJdn2dn1uvrs intJdn2dn2uvrs intJdn2dn3uvrs intJdn2dn4uvrs intJdn2dn5uvrs intJdn2dn6uvrs intJdn2dn7uvrs intJdn2dn8uvrs intJdn2dn9uvrs;...
          intJdn3dn1uvrs intJdn3dn2uvrs intJdn3dn3uvrs intJdn3dn4uvrs intJdn3dn5uvrs intJdn3dn6uvrs intJdn3dn7uvrs intJdn3dn8uvrs intJdn3dn9uvrs;...
          intJdn4dn1uvrs intJdn4dn2uvrs intJdn4dn3uvrs intJdn4dn4uvrs intJdn4dn5uvrs intJdn4dn6uvrs intJdn4dn7uvrs intJdn4dn8uvrs intJdn4dn9uvrs;...
          intJdn5dn1uvrs intJdn5dn2uvrs intJdn5dn3uvrs intJdn5dn4uvrs intJdn5dn5uvrs intJdn5dn6uvrs intJdn5dn7uvrs intJdn5dn8uvrs intJdn5dn9uvrs;...
          intJdn6dn1uvrs intJdn6dn2uvrs intJdn6dn3uvrs intJdn6dn4uvrs intJdn6dn5uvrs intJdn6dn6uvrs intJdn6dn7uvrs intJdn6dn8uvrs intJdn6dn9uvrs;...
          intJdn7dn1uvrs intJdn7dn2uvrs intJdn7dn3uvrs intJdn7dn4uvrs intJdn7dn5uvrs intJdn7dn6uvrs intJdn7dn7uvrs intJdn7dn8uvrs intJdn7dn9uvrs;...
          intJdn8dn1uvrs intJdn8dn2uvrs intJdn8dn3uvrs intJdn8dn4uvrs intJdn8dn5uvrs intJdn8dn6uvrs intJdn8dn7uvrs intJdn8dn8uvrs intJdn8dn9uvrs;...
          intJdn9dn1uvrs intJdn9dn2uvrs intJdn9dn3uvrs intJdn9dn4uvrs intJdn9dn5uvrs intJdn9dn6uvrs intJdn9dn7uvrs intJdn9dn8uvrs intJdn9dn9uvrs]);


%_________________________________________________________________________


%
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(1:9,1:9)=(zeros(9,9));
for i=1:9
 for j=i:9
 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 =[5/144;1/24;7/144;1/24]*(2*delabc);

 xe(1,1)=(xa+xb+xc)/3;
 xe(2,1)=(xa+xc)/2;
 xe(3,1)=xa;
 xe(4,1)=(xa+xb)/2;
 %
 ye(1,1)=(ya+yb+yc)/3;
```

```
 ye(2,1)=(ya+yc)/2;
 ye(3,1)=ya;
 ye(4,1)=(ya+yb)/2;
 %
 [sp,wt]=glsampleptsweights(ng);
 %for j=1:4
 %    qe(j,1)=(2*pi^2)*sin(pi*xe(j,1))*sin(pi*ye(j,1));
 %end
 %II =([  1/72, 7/864, 1/216, 7/864;...
 %     7/864,  1/54,  1/96, 1/216;...
  %    1/216,  1/96, 5/216,  1/96;...
   %   7/864, 1/216,  1/96,  1/54]);
 %f=(2*delabc)*(II*qe);
 %+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  xe1=xe(1,1);xe2=xe(2,1);xe3=xe(3,1);xe4=xe(4,1);
 ye1=ye(1,1);ye2=ye(2,1);ye3=ye(3,1);ye4=ye(4,1);
 f(1:9,1)=zeros(9,1)
 for i=1:ng
     si=sp(i,1);wi=wt(i,1);
     for j=1:ng
         sj=sp(j,1);wj=wt(j,1);
         n1ij=(si^2-si)*(sj^2-sj)/4;
         n2ij=(si^2+si)*(sj^2-sj)/4;
         n3ij=(si^2+si)*(sj^2+sj)/4;
         n4ij=(si^2-si)*(sj^2+sj)/4;
         n5ij=(1-si^2)*(sj^2-sj)/2;
         n6ij=(si^2+si)*(1-sj^2)/2;
         n7ij=(1-si^2)*(sj^2+sj)/2;
         n8ij=(si^2-si)*(1-sj^2)/2;
         n9ij=(1-si^2)*(1-sj^2);

        %___________________________________
         N1ij=(((1-si)*(1-sj)))/4;
         N2ij=(((1+si)*(1-sj)))/4;
         N3ij=(((1+si)*(1+sj)))/4;
         N4ij=(((1-si)*(1+sj)))/4;
         xeij=xe1*N1ij+xe2*N2ij+xe3*N3ij+xe4*N4ij;
         yeij=ye1*N1ij+ye2*N2ij+ye3*N3ij+ye4*N4ij;
         %
         %
        fcnxyij=fcnxy(fcn,xeij,yeij);
        f1i=n1ij*fcnxyij*(4+si+sj)/96;
        f2i=n2ij*fcnxyij*(4+si+sj)/96;
        f3i=n3ij*fcnxyij*(4+si+sj)/96;
        f4i=n4ij*fcnxyij*(4+si+sj)/96;
        f5i=n5ij*fcnxyij*(4+si+sj)/96;
        f6i=n6ij*fcnxyij*(4+si+sj)/96;
        f7i=n7ij*fcnxyij*(4+si+sj)/96;
        f8i=n8ij*fcnxyij*(4+si+sj)/96;
        f9i=n9ij*fcnxyij*(4+si+sj)/96;
       % f1i=n1ij*(2*pi^2)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;
        %f2i=n2ij*(2*pi^2)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;
        %f3i=n3ij*(2*pi^2)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;
        %f4i=n4ij*(2*pi^2)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;
        %f5i=n5ij*(2*pi^2)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;
        %f6i=n6ij*(2*pi^2)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;
        %f7i=n7ij*(2*pi^2)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;
        %f8i=n8ij*(2*pi^2)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;
        %f9i=n9ij*(2*pi^2)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;
%---------------------------------------------------------------
          f(1,1)=f(1,1)+f1i*wi*wj;
          f(2,1)=f(2,1)+f2i*wi*wj;
          f(3,1)=f(3,1)+f3i*wi*wj;
          f(4,1)=f(4,1)+f4i*wi*wj;
          f(5,1)=f(5,1)+f5i*wi*wj;
          f(6,1)=f(6,1)+f6i*wi*wj;
```

```
          f(7,1)=f(7,1)+f7i*wi*wj;
          f(8,1)=f(8,1)+f8i*wi*wj;
          f(9,1)=f(9,1)+f9i*wi*wj;
      end
  end
 f=(delabc)*f;

 %+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
%______________________________________
 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
 %-------------------------------------------------------------------------
%bcdof=[13;37;35;33;31;29;27;25;23;21;19;17;15];
%apply boundary conditions

%
mm=length(bcdof);
sdof=size(ss);
%
for i=1:mm
c=bcdof(i,1);
for j=1:sdof
ss(c,j)=0;
end
%
ss(c,c)=1;
ff(c,1)=bcval(i,1);
end
%solve the equations

phi=ss\ff;
for I=1:nnode
NN(I,1)=I;
end

disp('__________________________________________________________________')
disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
disp('_________________________________________________________________________________
____')
disp('                         fem-computed values             anlytical(theoretical)-
values           ')

disp([NN phi xi])
disp('_________________________________________________________________________________
____')
```

```
disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
nodes
gcoord


[x,y]=meshgrid(0:0.1:1,0:0.1:1);

for i=1:11
    for j=1:11
      for iel=1:nel
 %node numbers of quadrilateral
       nd1=nodes(iel,1);nd2=nodes(iel,2);nd3=nodes(iel,3);nd4=nodes(iel,4);
       nd5=nodes(iel,5);nd6=nodes(iel,6);nd7=nodes(iel,7);nd8=nodes(iel,8);
       nd9=nodes(iel,9);
 %coordinates of quadrilateral(u,v)

u(1,1)=gcoord(nd1,1);u(2,1)=gcoord(nd2,1);u(3,1)=gcoord(nd3,1);u(4,1)=gcoord(nd4,1);

v(1,1)=gcoord(nd1,2);v(2,1)=gcoord(nd2,2);v(3,1)=gcoord(nd3,2);v(4,1)=gcoord(nd4,2);
 %coordinates of the grid(x,y)

          in=inpolygon(x(i,j),y(i,j),u,v);
          if (in==1)
           X=x(i,j);Y=y(i,j);
           [t]=convexquadrilateral_coordinates(u,v,X,Y);
           r=t(1,1);
           s=t(2,1);
           shn1=(r^2-r)*(s^2-s)/4;
           shn2=(r^2+r)*(s^2-s)/4;
           shn3=(r^2+r)*(s^2+s)/4;
           shn4=(r^2-r)*(s^2+s)/4;
           shn5=(1-r^2)*(s^2-s)/2;
           shn6=(r^2+r)*(1-s^2)/2;
           shn7=(1-r^2)*(s^2+s)/2;
           shn8=(r^2-r)*(1-s^2)/2;
           shn9=(1-r^2)*(1-s^2);

PHI(i,j)=shn1*phi(nd1,1)+shn2*phi(nd2,1)+shn3*phi(nd3,1)+shn4*phi(nd4,1)+shn5*phi(nd5,1
)+shn6*phi(nd6,1)+shn7*phi(nd7,1)+shn8*phi(nd8,1)+shn9*phi(nd9,1);
              break
           end%if (in==1)
        end%for iel
   %THE PROGRAM EXECUTION JUMPS TO HERE if (in==1)
    end%for j
end%for i
  z=sin(pi*x).*sin(pi*y);

for i=1:11
    for j=1:11
        if (abs(PHI(i,j))<=1e-5)
            PHI(i,j)=0;
        end
        if (abs(z(i,j))<=1e-5)
            z(i,j)=0;
        end

    end
end
switch mesh
case 1
    hold off
 clf
figure(1)
```

```
 x=[0.0 1.0 1.0 0.5 0.0];
 y=[0.0 0.0 0.5 1.0 1.0];
 patch(x,y,'w')
hold on
[x,y]=meshgrid(0:.1:1,0:0.1:1)
y((y>1/2)&(y<=1)&(x>1/2)&(x<=1)&(x+y>3/2))=NaN;
[c,h]=contour(x,y,PHI)
xlabel('X-axis');
ylabel('Y-axis');
clabel(c,h);
axis square
st1='Contour level curves for ';
st2='FEM solution of ';
st3='Nine Noded  ';
st4='Special Quadrilateral';
st5=' Elements'
title([st1,st2,st3,st4,st5])
sst1='(MESH HAS '
sst2=num2str(nnode)
sst3=' NODES'
sst4=' AND '
sst5=num2str(nel)
sst6=' ELEMENTS)'
text(0.25,-.08,[sst1 sst2 sst3 sst4 sst5 sst6])
figure(2)
 x=[0.0 1.0 1.0 0.5 0.0];
 y=[0.0 0.0 0.5 1.0 1.0];
 patch(x,y,'w')
hold on
[x,y]=meshgrid(0:.1:1,0:0.1:1)
y((y>1/2)&(y<=1)&(x>1/2)&(x<=1)&(x+y>3/2))=NaN;
[c,h]=contour(x,y,z)
xlabel('X-axis');
ylabel('Y-axis');
clabel(c,h);
axis square
title('contour level curves for exact solution: sin(pi*x)*sin(pi*y)')
 mm=0;
 for i=1:11
  for j=1:11
      mm=mm+1;
      femsoln(mm,1)=PHI(i,j);
      exactsoln(mm,1)=z(i,j);
  end
end


case 2
    hold off
 clf
figure(1)
[x,y]=meshgrid(0:.1:1,0:0.1:1)
[c,h]=contour(x,y,PHI)
xlabel('X-axis');
ylabel('Y-axis');
clabel(c,h);
axis square
st1='Contour level curves for ';
st2='FEM solution of ';
st3='Nine Noded  ';
st4='Special Quadrilateral';
st5=' Elements'
title([st1,st2,st3,st4,st5])
sst1='(MESH HAS '
sst2=num2str(nnode)
```

```
sst3=' NODES'
sst4=' AND '
sst5=num2str(nel)
sst6=' ELEMENTS)'
text(0.25,-.08,[sst1 sst2 sst3 sst4 sst5 sst6])

figure(2)
[x,y]=meshgrid(0:.1:1,0:0.1:1)
[c,h]=contour(x,y,z)
xlabel('X-axis');
ylabel('Y-axis');
clabel(c,h);
axis square
title('contour level curves for exact solution: sin(pi*x)*sin(pi*y)')
 mm=0;
 for i=1:11
  for j=1:11
      mm=mm+1;
      femsoln(mm,1)=PHI(i,j);
      exactsoln(mm,1)=z(i,j);
  end
end

end%switch mesh
[femsoln exactsoln]

 disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
[1 phi(1,1) xi(1,1)]
disp('____________________________________________________________________')
disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
disp('___________________________________________________________________________________
____')
disp('                          fem-computed values             anlytical(theoretical)-
values           ')

disp([NN phi xi])
disp('___________________________________________________________________________________
____')

disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
for ncpt=1:nel
    centrpt=nodes(ncpt,9)
    elcentr(ncpt,1)=centrpt;
    phicpt(ncpt,1)=phi(centrpt,1);
    xicpt(ncpt,1)=xi(centrpt,1);
end
for I=1:nel
elnumm(I,1)=I;
end
disp('___________________________________________________________________________________
____')
disp(' serial no        center point                fem-computed values
anlytical(theoretical)-values           ')

disp([elnumm  elcentr phicpt xicpt])
disp('___________________________________________________________________________________
____')

format compact
%=========================================================================================
===========
```

```
disp('__________________________________________________________________')
disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
disp('_________________________________________________________________________________
_____________________________________________________')
disp('        NODE FEM SOLUTION EXACT SOLUTION       NODE FEM SOLUTION EXACT SOLUTION
NODE FEM SOLUTION EXACT SOLUTION          ')

disp('---------------------------------------------------------------------------------
------------------------------------------------------')
for I=1:3:nel
   % A=[elcentr(I) phicpt(I) xicpt(I)];B=[elcentr(I+1) phicpt(I+1)
xicpt(I+1)];C=[elcentr(I+2) phicpt(I+2) xicpt(I+2)];
    %disp([elcentr(I) phicpt(I) xicpt(I) elcentr(I+1) phicpt(I+1) xicpt(I+1)])
    %disp([elcentr(I+2) phicpt(I+2) xicpt(I+2)])
    %disp([A B C])
    fprintf('\n%5d %18.14f %18.14f %5d %18.14f %18.14f %5d %18.14f %18.14f',elcentr(I),
phicpt(I), xicpt(I), elcentr(I+1), phicpt(I+1), xicpt(I+1), elcentr(I+2), phicpt(I+2),
xicpt(I+2));
end
 fprintf('\n')
disp('---------------------------------------------------------------------------------
------------------------------------------------------')
```