www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 12 Dec. 2016, Page No. 19674-19778

An All Quadrilateral Automesh Generation Technique and Explicit Integration Scheme for Finite Elements to Solve Some Elliptic Boundary Value Problems in Two Space

H.T Rathod [a*], ***K. Sugantha Devi*** [b]
[a] Department of Mathematics, Central College Campus,
Bangalore University, Bangalore- 560001
E-mail: htrathod2010@gmail.com
[b] Department of Mathematics, Dr. T. Thimmaiah Institute of Technology, Oorgam Post,
Kolar Gold Field, Kolar District, Karnataka state, Pin- 563120, India.
Email: suganthadevik@yahoo.co.in

Abstract :
This paper proposes the explicit integration scheme for a unique linear convex 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 quadrilaterals is derived by using the standard transformations in two steps. We first map an arbitrary triangle into a standard right isosceles triangle by using an affine linear transformation from global (x, y) space into a local space (u, v). We discretise the standard right isosceles triangle in (u, v) space into three unique linear convex quadrilaterals. We have shown that any unique linear convex quadrilateral in (x, y) space can be mapped into one of the unique quadrilaterals in (u, v) space. We can always map these linear convex quadrilaterals into a 2-aquare in a natural (ξ, η) space by an appropriate bilinear transformation. Using these two mappings, we have established an integral derivative product relation between the linear convex quadrilaterals in the (x, y) space interior to the arbitrary triangle and the linear convex quadrilaterals of the local space (u, v) interior to the standard right isosceles triangle. Further , we have shown that the product of global derivative integrals $S^{i,j,e}$ in (x, y) space can be expressed as a matrix triple product P ($K^{i,j,e}$) P^T X (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.We have shown that the explicit integration of the product of local derivative integrals in (u, v) space over the unique quadrilateral spanning vertices (1/3 ,1/3), (0, ½), (0, 0) , (1/2, 0) is now possible by use of symbolic processing capabilities in MATLAB which are based on Maple – V software package. In a similar manner we have found explicit integrals of ihe shape function and global derivative products as well as the product of shape functions.We use these integral values in computing stiffness matrix for some elliptic equations with constant coefficients.The proposed explicit integration scheme is a useful technique for boundary value problems governed by either a single equation or a system of second order partial differential equations. The physical applications of such problems are numerous in science, and engineering, the examples of single equations are the well known Laplace and Poisson equations with suitable boundary conditions. These problems are already studied in authors recent paper. We have demonstrated the proposed explicit integration scheme to solve the elliptic equations with constant coefficients subject to Dirichlet boundary conditions over simple polygonal domains like 1-square and 2-square.We have demonstrated the propsed scheme and technique to study five typical elliptic boundary value problems.Three FEM models with 2400,5400 and 7776 elements and 2481,5521 and 7921 nodes respectively are used.Tables of fem computed values,theoretical values at selected nodes and the finite element meshes are appended.The findings confirm with surface plots of exact solutions for the considered elliptic boundary value problems.
Key words: Explicit Integration, Finite Element Method, Matlab Symbolic Mathematics, All Quadrilateral Mesh Generation Technique,Elliptic equations with constant coefficients ,Dirichlet Boundary Conditions ,Polygonal Domain,2-square,1-square

1.INTRODUCTION

DOI: 10.18535/ijecs/v5i12.75

Partial differential equations arise in the mathematical modelling of many physical, chemical and biological phenomena and many diverse subject areas such as fluid dynamics, electromagnetism, material science, astrophysics, economy, financial modelling, etc. Very frequently the equations under consideration are so complicated that finding their solutions in closed form or by purely analytical means (e.g. by Laplace and Fourier transform methods, or in the form of a power series) is either impossible or impracticable, and one has to resort to seeking numerical approximations to the unknown analytical solution.

Solving boundary value problems for partial diBerential equations (PDEs) is a central topic In applied mathematics. Analytical methods (e.g., separation of variables and transform techniques) are valued for their exactness and the insight they provide; however, the range of problems they solve is limited. Numerical methods (e.g., finite element, finite diBerence, and spectral methods) solve a much wider range of problems, albeit only approximately. Some methods (e.g., boundary integral methods) combine specific analytical information about the solution with numerical approximations.

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 fnite element analysis are widely described in references [1, 3]. In this paper the well-known elliptic partial differential 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 a linear polygon in Cartesian two space (x,y). This discrtisation is carried in two steps, We first discretise the linear polygon (for simplicity a square) 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 global derivatives and 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.

2. ELLIPTIC BOUNDARY VALUE PROBLEM IN TWO SPACE

2.1.1 ELLIPTIC EQUATIONS WITH VARTABLE COEFFICIENTS

DOI: 10.18535/ijecs/v5i12.75

We write the linear two dimensional second order equation in the most general form

$-(a_{11}\,u_{xx} + 2a_{12}\,u_{xy} + a_{22}\,u_{yy}\ ) + b_1\,u_x + b_2\,u_y\ + cu = f$ ----------(1)

on a domain $\Omega \subset R^2$, where the coefficients a_{ij} , b_i , c and f are functions of x and y.

Because $\partial^2 u/\partial x\partial y = \partial^2 u/\partial y\partial x$, $i.e\ u_{xy} = u_{yx}$,we may assume without loss of generality that $\exists\ constants\ such\ that\ a_{12} = a_{21}$.

We assume the equation to be elliptic on Ω , which means the coefficients of the principle part of eq(1) satisfy

$a_{11}{\xi_1}^2 + 2a_{12}\,\xi_1\xi_2 + a_{22}\,{\xi_3}^2 \neq 0$, $\forall\ (\xi_1,,\xi_2\ ) \neq (0,0)$ ----------(2)

It is easy to see that eqn(2) is equivalent ${a_{12}}^2 < a_{22}\,a_{11}$

The elliptic eqn(1) is defined on an open connected domain $\Omega \subset R^2$ and to obtain a unique solution we have to provide boundary conditions. It appears that we need one condition along the boundary $\Gamma = \partial\Omega$. The boundary condition can be of the form

u = g on Γ ----------(3)

or more generally

$\frac{\partial u}{\partial n}$ +$\beta_1 \frac{\partial u}{\partial s} + \beta_2 u$ = h on Γ ----------(4)

Where n is the outward unit normal on Γ and s is the vector tangent to the boundary.

Boundary conditions of the type in eqn (3) are called Dirichlet conditions and boundary conditions of the type in eqn (4) are called mixed conditions. The special case when $\beta_1 = \beta_2 = 0$ is called Neumann condition.

2.1.2 ELLIPTIC EQUATIONS WITH CONSTANT COEFFICIENTS

Theorem: If the coefficients in eqn(1) are taken as constant,the principle part

$a_{11}\,u_{xx} + 2a_{12}\,u_{xy} + a_{22}\,u_{yy}$ can be reduced to the form

$b_0\Delta u$,where b_0 is an arbitrary constant

by a linear coordinate transformation.

Hence a transformation of eqn(1) along the line of above theorem reduces to the following form

$-b_0\Delta u + b_1 u_x + b_2 u_y + cu = f$ ----------(5)

This paper is concerned with boundary value problems of elliptic equations with constant coefficients.We assume an elliptic with two independent variables x and y as given in eqn(1),where a_{11} , a_{12}, a_{22} , b_1 , b_2 $and\ c$ are constants and f $is\ a$ function of x and y. We rewrite eqn(1) as

$$\mathcal{L}u = f \quad \text{----------(6)}$$

Where,

$$\mathcal{L}u = -(\,a_{11}\,\frac{\partial^2 u}{\partial x^2} + 2a_{12}\,\frac{\partial^2 u}{\partial x \partial y} + a_{22}\frac{\partial^2 u}{\partial y^3}) + b_1\frac{\partial u}{\partial x}\ + b_2\,\frac{\partial u}{\partial y} + cu \quad \text{----------(7)}$$

and the ellipticity condition can be formulated as

$$(a_{11}a_{22} - a_{12}^2\,) > 0 \quad \text{----------(8)}$$

Hence in view of the result deduced in eqn(5),we obtain some of the following typical elliptic equations :

The Laplace equation

$\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^3} = 0$, in Ω ----------(9)

The Poisson equation

$-\Delta u = f$, in Ω ----------(10)

The reaction diffusion equation

$-\Delta u + cu = f$, in Ω ----------(11)

and the convection diffusion equation

$-\epsilon\Delta u + b_1\frac{\partial u}{\partial x}\ + b_2\,\frac{\partial u}{\partial y} + cu = f$, in Ω ,$\epsilon > 0$ ----------(12)

If we add Dirichlet boundary conditions to the Poisson equation in eqn(10),we obtain the classical Dirichlet problem for Poisson equation. In a recent work[ ],the authors have studied this problem over a polygonal domain.hence the Laplace and Poisson equations will not be considered in this paper.

2.2 Weak Formulation of the Elliptic Boundary Value Problem

A function u $\epsilon\ C^2(\Omega) \cap C(\bar{\Omega})$ satisfying

$\mathcal{L}u = f,$ in Ω ----------(13a)

DOI: 10.18535/ijecs/v5i12.75

and u = 0 , on Γ ----------(13b)

Where,we also have from eqn(1)

$\mathcal{L}u = -\sum_{i,j=1}^{2} \frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial u}{\partial x_j}\right) + \sum_{i=1}^{2} b_i \frac{\partial u}{\partial x_i} + cu$

is called a classical solution of the of the elliptic problem. The theory of partial differential equation tells us that this problem has a unique classical solution provided that a_{ij}, b_i, c, f and $\partial\Omega$ are sufficiently smooth. However in many applications gne has to consider equations where these smoothness requirements are violete and for such problems jhe classical yheory is inappropriate.

In order to overcome the limitations of the classical theory and to be able to deal with partial differential equations with nonsmooth data, we generalise the notion of solution by weakening the differentiability requirements on u.

Hence,let us suppose that u is a classical solution of $\mathcal{L}u = f$, in Ω and u = 0 , on Γ .Then for any $v \in C_0^1(\Omega)$

$\int_\Omega \left(-\sum_{i,j=1}^{2} \frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial u}{\partial x_j}\right) + \sum_{i=1}^{2} b_i \frac{\partial u}{\partial x_i} + cu\right).v \, dx = \int_\Omega fv \, dx$, where $dx = dx_1 dx_3$ ----(14)

Upon integrating by parts in the first integral and noting that $v = 0 \; on \; \Gamma$.we obtain the variational formulation of the given eqn(1)

$\sum_{i,j=1}^{2} \int_\Omega a_{ij}(x) \frac{\partial u}{\partial x_i}\frac{\partial v}{\partial x_j} dx + \sum_{i=1}^{2} \int_\Omega b_i(x) \frac{\partial u}{\partial x_i} v dx + \int_\Omega c(x)uv \, dx = \int_\Omega f(x)v \, dx, \forall v \in C_0^1(\Omega)$-- (15)

A function u satisfying eqn(15) is called weak solution of eqns (13a,b).All derivatives in eqn(15) should be understood as weak de4rivatives. In this variational problem the task is given $a_{ij}, b_i, c \; and \; f$, find u (in ales restrictive function space) s.t. eqn(15) is satisfied for all smooth functions vanishing on the boundary of the given domain.

2.3 FINITE ELEMENTS FOR ELLIPTIC BOUNDARY VALUE PROBLEMS IN TWO SPACE : IMPLEMENTATION FOR ELLIPTIC EQUATION S WITH CONSTANT COEFFICIENTS

2.3.1 Weak Form

Given the elliptic eqn(5)

$-b_0\Delta u + b_1 u_x + b_2 u_y + cu = f$, all x∈ Ω ----------(16a)

u = g on $\Gamma = \partial\Omega$ ----------(16b)

We have already obtained in eqn(6) with 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

$b_0 \int_\Omega \nabla u . \nabla v \, dx + \sum_{i=1}^{2} \int_\Omega b_i \frac{\partial u}{\partial x_i} v dx + \int_\Omega cuv \, dx = \int_\Omega f(x)v \, dx, \forall v \in C_0^1(\Omega)$ --------(17)

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(16a-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

$b_0 \int_\Omega \nabla u_h . \nabla v \, dx + \sum_{k=1}^{2} \int_\Omega b_k \frac{\partial u_h}{\partial x_k} v dx + \int_\Omega c \, u_h v \, dx = \int_\Omega f(x)v \, dx, \forall v \in C_0^1(\Omega)$ -----(18)

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

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

DOI: 10.18535/ijecs/v5i12.75

$\sum_{j=1}^{N} \alpha_j \left(b_0 \int_\Omega \nabla \varphi_i . \nabla \varphi_j \, dx + \sum_{k=1}^{2} \int_\Omega b_k \varphi_I \frac{\partial \varphi_j}{\partial x_k} dx + \int_\Omega c \varphi_I \varphi_j \, dx\right) = \int_\Omega f(x) \varphi_i \, dx$; (i=1,2,3,...., N)

----------(19)

This is really a linear system of the form

$Eu = f$ ----------(20)

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

$E = \sum_{i=1}^{N} \sum_{j=1}^{N} E_{i,j}$; $f = \sum_{i=1}^{N} f_i$

$E_{i,j} = \left(b_0 \int_\Omega \nabla \varphi_i . \nabla \varphi_j \, dx + \sum_{k=1}^{2} \int_\Omega b_k \varphi_i \frac{\partial \varphi_j}{\partial x_k} dx + \int_\Omega c \varphi_I \varphi_j \, dx\right)$ ----------(21a)

$f_i = \int_\Omega f \varphi_i \, dx$ ----------(21b)

and E 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 Ω^e 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 chosen 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 ?
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 methods are 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(21) in section 2.3.2)

$E_{i,j} = \left(b_0 \int_\Omega \nabla \varphi_i . \nabla \varphi_j \, dx + \sum_{k=1}^{2} \int_\Omega b_k \varphi_i \frac{\partial \varphi_j}{\partial x_k} dx + \int_\Omega c \varphi_I \varphi_j \, dx\right)$ ----------(21a)

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 φ_i and φ_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(21a) as

$E_{i,j} = \sum_{\Omega^e \, \varepsilon \Omega_h^e} E_{i,j}^e$ ----------(22a)

Where

$E_{i,j}^e = b_0 K_{i,j}^e + [b_1 \quad b_2] L_{i,j}^e + c M_{i,j}^e$ ----------(22b)

Where.

$K_{i,j}^e = \int_{\Omega^e} \nabla \varphi_i . \nabla \varphi_j \, dx$

$L_{i,j}^e = \int_{\Omega^e} [\varphi_i \ \frac{\partial \varphi_j}{\partial x_1} \quad \varphi_i \ \frac{\partial \varphi_j}{\partial x_2}]^T dx$

$M_{i,j}^e = \int_{\Omega^e} \varphi_i \varphi_j \, dx$

----------(22c)

DOI: 10.18535/ijecs/v5i12.75

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

$$E_{i,j}^e = (b_0 \int_{\Omega^e} \nabla \varphi_i . \nabla \varphi_j \, dx + \sum_{k=1}^{2} \int_{\Omega^e} b_k \varphi_i \frac{\partial \varphi_j}{\partial x_k} \mathrm{dx} + \mathrm{c} \int_{\Omega^e} \varphi_I \varphi_j \, dx)$$

look like the entries

$$E_{i,j} = (b_0 \int_{\Omega} \nabla \varphi_i . \nabla \varphi_j \, dx + \sum_{k=1}^{2} \int_{\Omega} b_k \varphi_i \frac{\partial \varphi_j}{\partial x_k} \mathrm{dx} + \mathrm{c} \int_{\Omega} \varphi_I \varphi_j \, dx) = \sum_{\Omega^e \varepsilon \Omega_h^e} E_{i,j}^e$$

except the domain of integration is an element Ω^e. So,we only need to save all entries of E^e =$[E_{i,j}^e]$ which corresponds to nodes on Ω^e. Then if Ω^e has d nodes , we can think of E^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

$f_i = \int_{\Omega} f \varphi_i \, dx = \sum_{e=1}^{N_e} f_i^e$ ----------(22d)

$f_i^e = \int_{\Omega^e} f \varphi_i \mathrm{d\,x}$ .....(22e)

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

From eqn(20) and eqns(22a-d) it follows that $Eu=f$ is equivalent to

$\sum_{e=1}^{N_e} E^e \mathrm{u^e} = \sum_{e=1}^{N_e} f^e$ .....(22f)

Where

$\mathrm{u^e} = (\mathrm{u_1^e}, \mathrm{u_2^e}, \mathrm{u_3^e}, \ldots\ldots\ldots\ldots \mathrm{u_d^e})^T$, $f^e = (f_1^e, f_2^e, f_3^e, \ldots\ldots\ldots\ldots f_d^e)^T$ ----------(22g)

and 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 $E_{i,j}^e$ and f_i^e

In order to compute the local/element stiffness matrices,we need to compute the integrals.$E_{i,j}^e = (b_0 \int_{\Omega^e} \nabla \varphi_i . \nabla \varphi_j \, dx + \sum_{k=1}^{2} \int_{\Omega^e} b_k \varphi_i \frac{\partial \varphi_j}{\partial x_k} \mathrm{dx} + \mathrm{c} \int_{\Omega^e} \varphi_I \varphi_j \, dx)$ These integrals are computed by making a change of variables to a reference element. We now outline a brief procedure for element oriented computation

(I)For each element Ω^e , compute it's local stiffness matrix K^e. This requires computing the integrals in $K_{i,j}^e$ 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 into 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 quadrilateral will be transformed into a standard square(1-square or a 2-square).Since in two dimensional space $x = (x_1, x_2) = (x, y)$,we rewrite the explicit form of $K_{i,j}^e$ as

$E_{i,j}^e = \int_{\Omega^e} \left\{ \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} + \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_j}{\partial y} \right\} dxdy + \int_{\Omega^e} (b_1 \varphi_i \frac{\partial \varphi_j}{\partial x} + b_2 \varphi_i \frac{\partial \varphi_j}{\partial y}) dxdy + c \int_{\Omega^e} \varphi_I \varphi_j \, dx \, dy$

$= \sum_{e=1}^{N_e} \sum_{n=0}^{2} S_{i,j}^E$ .....(22h)

where

$S_{i,j}^E = [\int_{Q_E} \left\{ \frac{\partial \varphi_i}{\partial x} \frac{\partial \varphi_j}{\partial x} + \frac{\partial \varphi_i}{\partial y} \frac{\partial \varphi_j}{\partial y} \right\} dxdy + \int_{Q_E} (b_1 \varphi_i \frac{\partial \varphi_j}{\partial x} + b_2 \varphi_i \frac{\partial \varphi_j}{\partial y}) dxdy + c \int_{Q_E} \varphi_I \varphi_j \, dx \, dy]$

with E=3e+n-2,e=1,2,... N_e ; n=0,1,2 ----------(22i)

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)

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

(II) For each element Ω^e, we first compute the local stiffness matrices $S^E = [S_{i,j}^E]$ and then add contribution of $E^e = S^{3e-2} + S^{3e-1} + S^{3e}$, to the global stiffness matrix K. We then repeat this procedure for all elements i.e for

DOI: 10.18535/ijecs/v5i12.75

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

2.3.6 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.

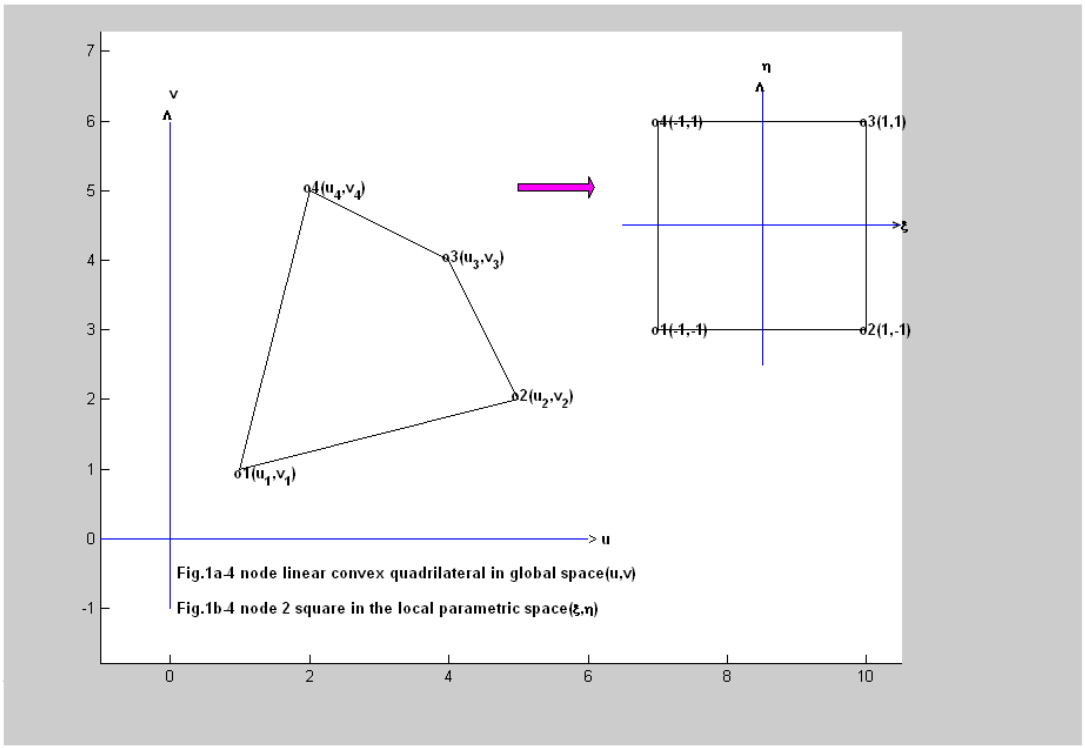


Fig.1a-4 node linear convex quadrilateral in global space(u,v)

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

$$\begin{pmatrix} u \\ v \end{pmatrix} = \sum_{k=1}^{4} \begin{pmatrix} u_k \\ v_k \end{pmatrix} M_k(\xi, \eta) \quad \text{---------- (23)}$$

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

$$M_k(\xi, \eta) = \frac{1}{4}(1 + \xi\xi_k)(1 + \eta\eta_k) \ , \quad k = 1, 2, 3, 4 \quad \text{---------- (24a)}$$

Where $\{(\xi_k, \eta_k), k = 1, 2, 3, 4\}$ = {(-1,-1),(1,-1),(1,1),(-1,1)} ---------- (24b)

describe two transformations over a linear convex quadrilateral element from the original global space into the local parametric space.

2.3.7 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

$$\begin{pmatrix} \phi \\ \psi \end{pmatrix} = \sum_{k=1}^{4} \begin{pmatrix} \phi_k \\ \psi_k \end{pmatrix} N_k{}^e(\xi, \eta) \quad \text{---------- (25)}$$

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.(24a-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 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 a square domain which is a simplest polygonal.

2.3.8 Subparmetric Transformation :

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

$$\begin{pmatrix} \phi \\ \psi \end{pmatrix} = \sum_{k=1}^{nde} \begin{pmatrix} \phi_k{}^e \\ \psi_k{}^e \end{pmatrix} N_k{}^e(\xi, \eta) \qquad \text{---------- (26)}$$

Where ϕ_k , ψ_k refer to unknowns at node k and nde >4 and $N_k{}^e$ are the shape functions choosen from Serendipity or Lagrange higher order 2-square finite elements.

2.3.9 Explicit Form of the Jacobian and Global Derivatives :

Jacobian

Let us consider an arbitrary linear convex quadrilateral in the global Cartesian space (u, v) as in Fig 1a , 1b which is mapped into a 4-node 2- square in the local parametric space (ξ, η).

From the Eq.(23) and Eq.(24), we have

$$\frac{\partial u}{\partial \xi} = \sum_{k=1}^{4} u_k \frac{\partial M_k}{\partial \xi} = \frac{1}{4}[\,(-u_1 + u_2 + u_3 - u_4) + (u_1 - u_2 + u_3 - u_4)\,\eta\,] \qquad \text{---------- (27a)}$$

$$\frac{\partial u}{\partial \eta} = \sum_{k=1}^{4} u_k \frac{\partial M_k}{\partial \eta} = \frac{1}{4}[\,(-u_1 - u_2 + u_3 + u_4) + (u_1 - u_2 + u_3 - u_4)\,\xi\,] \qquad \text{---------- (27b)}$$

$$\frac{\partial v}{\partial \xi} = \frac{1}{4}[\,(-v_1 + v_2 + v_3 - v_4) + (v_1 - v_2 + v_3 - v_4)\,\eta\,] \qquad \text{---------- (27c)}$$

$$\frac{\partial v}{\partial \eta} = \frac{1}{4}[\,(-v_1 - v_2 + v_3 + v_4) + (v_1 - v_2 + v_3 - v_4)\,\xi\,] \qquad \text{---------- (27d)}$$

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

$$J = \frac{\partial(u,v)}{\partial(\xi,\eta)} = \frac{\partial u}{\partial \xi}\frac{\partial v}{\partial \eta} - \frac{\partial u}{\partial \eta}\frac{\partial v}{\partial \xi} = \alpha + \beta\xi + \gamma\,\eta \qquad \text{---------- (28a)}$$

Where

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

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

$$\gamma = \frac{1}{8}[\,(u_4 - u_1)(v_2 - v_3) + (u_3 - u_2)(v_4 - v_1)\,] \qquad \text{---------- (28b)}$$

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]

$$\begin{pmatrix} \frac{\partial N_i^e}{\partial u} \\ \frac{\partial N_i^e}{\partial v} \end{pmatrix} = \frac{1}{J}\begin{bmatrix} \frac{\partial v}{\partial \eta} & -\frac{\partial v}{\partial \xi} \\ -\frac{\partial u}{\partial \eta} & \frac{\partial u}{\partial \xi} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \end{bmatrix} \qquad \text{---------- (29)}$$

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.(27a)–(27d) while J is defined in Eq.(28a-b) , ($i, j = 1, 2, 3, - - - - -$, nde) , nde = the number of nodes per element.

2.3.10 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 and let Z be its centroid. We can obtain three linear convex quadrilaterals ZTRU, ZUPS and ZSQT from triangle Δ PQR as shown in Fig2. If we map each of these quadrilaterals into 2-squares in which the nodes are oriented in counter clockwise from Z then Jacobian J for each element e is given by

$$J = J^e = \frac{1}{48}\Delta\,pqr\,(4 + \xi + \eta), \qquad e = 1, 2, 3 \qquad \text{---------- (30)}$$

DOI: 10.18535/ijecs/v5i12.75

Where Δ pqr is the area of the triangle Δ PQR

$$2\Delta\,\mathbf{pqr} = \begin{vmatrix} 1 & x_p & y_p \\ 1 & x_q & y_q \\ 1 & x_r & y_r \end{vmatrix} = [\,(x_p - x_r)(y_q - y_r) - (x_q - x_r)(y_p - y_r)\,] \qquad \text{---------- (31)}$$

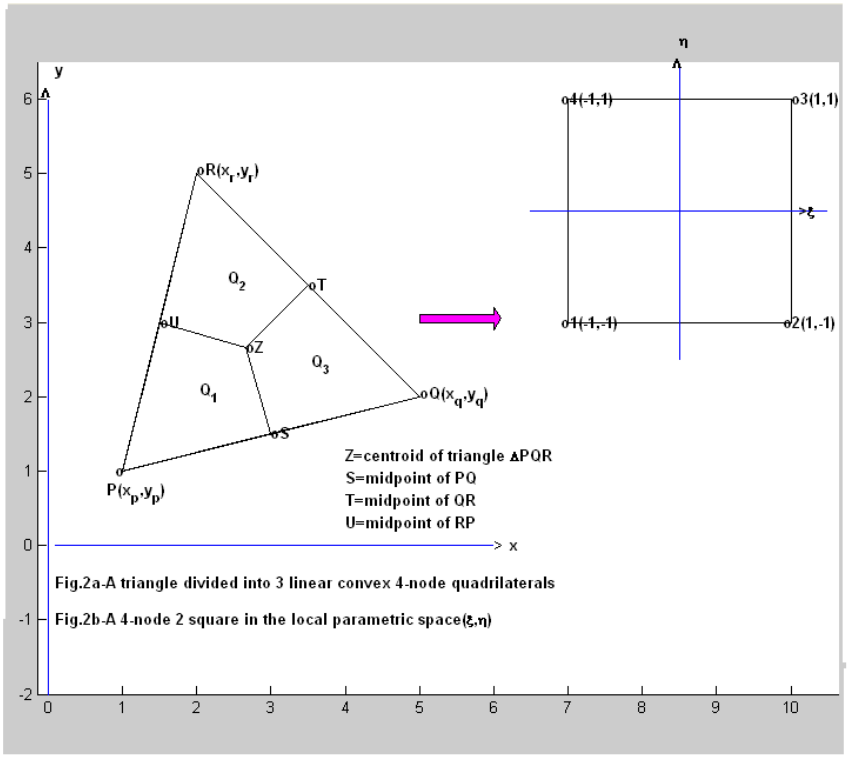


Fig.2a-A triangle divided into 3 linear convex 4-node quadrilaterals

Fig.2b-A 4-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 an arbitrary triangle with the vertices P(x_p, y_p) , Q(x_q, y_q) and R(x_r, y_r) , let S, T, U be the midpoints of sides PQ, QR, and RP and let Z be the centroid of Δ PQR, Then we obtain three quadrilaterals Q_1, Q_2, Q_3 spanning the vertices <ZUPS> , <ZSQT> and <ZTRU> . these quadrilaterals can be mapped into the quadrilateral spanning vertices GECF with G(1/3, 1/3), E(0, ½) , C(0, 0) and F(1/2, 0) of the right isosceles triangle Δ ABC with spanning vertices A(1, 0), B(0, 1) and C(0, 0) in the (u, v) space as shown in Fig 3a and Fig 3b

DOI: 10.18535/ijecs/v5i12.75

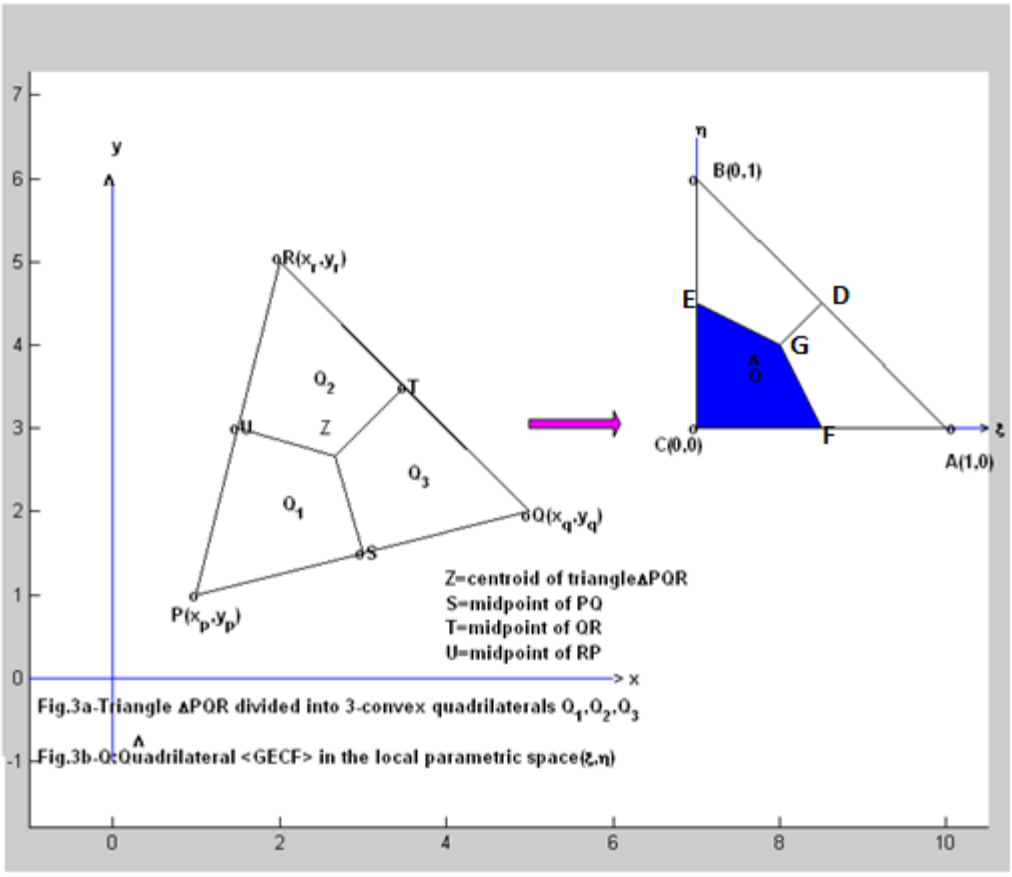


Fig.3a-Triangle △PQR divided into 3-convex quadrilaterals Q_1,Q_2,Q_3

Fig.3b-Quadrilateral <GECF> in the local parametric space(ξ,η)

Proof : The sum of the quadrilaterals Q_1+ Q_2+ Q_3 = Δ PQR as shown in Fig 2a & Fig 3a. The linear transformations

$$\begin{pmatrix} x^{(1)} \\ y^{(1)} \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} w + \begin{pmatrix} x_q \\ y_q \end{pmatrix} u + \begin{pmatrix} x_r \\ y_r \end{pmatrix} v \quad \text{---------- (32a)}$$

$$\begin{pmatrix} x^{(2)} \\ y^{(2)} \end{pmatrix} = \begin{pmatrix} x_q \\ y_q \end{pmatrix} w + \begin{pmatrix} x_r \\ y_r \end{pmatrix} u + \begin{pmatrix} x_p \\ y_p \end{pmatrix} v \quad \text{---------- (32b)}$$

$$\begin{pmatrix} x^{(3)} \\ y^{(3)} \end{pmatrix} = \begin{pmatrix} x_r \\ y_r \end{pmatrix} w + \begin{pmatrix} x_p \\ y_p \end{pmatrix} u + \begin{pmatrix} x_q \\ y_q \end{pmatrix} v \quad \text{---------- (32c)}$$

with $w = 1 - u - v$ ---------- (32d)

map the arbitrary triangle Δ PQR into a right isosceles triangle A(1, 0) , B(0, 1) and C(0, 0) in the uv–plane We can now verify that quadrilateral Q_1 spanned by vertices Z($\frac{x_p+x_q+x_r}{3}$, $\frac{y_p+y_q+y_r}{3}$) , U($\frac{x_r+x_p}{2}$,$\frac{y_r+y_q}{2}$) , P(x_p, y_p) ,S($\frac{x_p+x_q}{2}$, $\frac{y_p+y_q}{2}$) in xy- plane is mapped into the quadrilateral spanning the vertices G(1/3, 1/3) , E(0, ½), C(0, 0) and F(1/2, 0) by use of the transformation given in Eq.(32a),

Similarly, we see that the quadrilateral Q_2 spanned by vertices Z,S,Q,T is mapped into the quadrilateral spanned by vertices G(1/3, 1/3), E(0, ½), C(0, 0) and F(½, 0) by use of the transformation of Eq.(32b), Finally the quadrilateral Q_3 in the xy- plane is mapped into the quadrilateral GECF in uv- plane by use of the linear transformation of Eq.(32c),

This completes the proof

We have shown in the present section that an arbitrary triangle can be discretised into three linear convex quadrilaterals. Further, each of these quadrilaterals can be mapped into a unique quadrilateral in uv-plane spanned by vertices (1/3, 1/3), (0, ½), (0, 0) and (½, 0) .

Algorithms for Computing the Integrals : $K^e_{i,j}$, $L^e_{i,j}$ and $M^e_{i,j}$

From eqn(21),we recall that the entuie in the element stiffness matrix is expressed

$$E^e_{i,j} = b_0 K^e_{i,j} + [b_1 \quad b_2] L^e_{i,j} + cM^e_{i,j}$$

In the following sections,we explain the algorithms to compute the required integrals

3.1 Products of Global Derivative Integrals: $\int_{\Omega^e} \nabla\varphi_i \cdot \nabla\varphi_j \, dx = K^e_{i,j}$

If $N_i^{(e)}$ denotes the basis function for node i of element e , then by chain rule of partial differentiation

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

We note that to transform $Q_e (e = 1, 2, 3)$ of ΔPQR in Cartesian space (x,y) into $\hat{Q}$, the quadrilateral spanned by vertices (1/3,1/3), (0,½), (0,0) and (1/2,0) in uv-plane we must use the earlier transformations.

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_p \\ y_p \end{pmatrix} + \begin{pmatrix} x_q - x_p \\ y_q - y_p \end{pmatrix} u + \begin{pmatrix} x_r - x_p \\ y_r - y_p \end{pmatrix} v \quad \text{for } Q_1 \text{ in } \Delta PQR \quad \text{---------- (34a)}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_q \\ y_q \end{pmatrix} + \begin{pmatrix} x_r - x_q \\ y_r - y_q \end{pmatrix} u + \begin{pmatrix} x_p - x_q \\ y_p - y_q \end{pmatrix} v \quad \text{for } Q_2 \text{ in } \Delta PQR \quad \text{---------- (34b)}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_r \\ y_r \end{pmatrix} + \begin{pmatrix} x_p - x_r \\ y_p - y_r \end{pmatrix} u + \begin{pmatrix} x_q - x_r \\ y_q - y_r \end{pmatrix} v \quad \text{for } Q_3 \text{ in } \Delta PQR \quad \text{---------- (34c)}$$

and the above transformations viz Eqs.(24a)-(24c) are of the form

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

DOI: 10.18535/ijecs/v5i12.75

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

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

This gives

$$u = (\propto_a + \beta_a x + \gamma_a y)/(2\,\Delta_{abc})$$
$$v = (\propto_b + \beta_b x + \gamma_b y)/(2\,\Delta_{abc}) \quad \text{---------- (36)}$$

$\propto_a = (x_b y_c - x_c y_b)$, $\propto_b = (x_c y_a - x_a y_c)$,
$\beta_a = (y_b - y_c)$, $\beta_b = (y_c - y_a)$,
$\gamma_a = (x_c - x_b)$, $\gamma_b = (x_a - x_c)$,

$$\frac{\partial(x,y)}{\partial(u,v)} = 2\Delta_{abc} = \begin{vmatrix} 1 & x_a & y_a \\ 1 & x_b & y_b \\ 1 & x_c & y_c \end{vmatrix} = 2 * \text{area of the triangle } \Delta ABC$$
$$= (\gamma_b \beta_a - \gamma_a \beta_b) \quad \text{---------- (37)}$$

Hence from Eq.(23) and Eq.(22), we obtain

$$\begin{pmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{\beta_a}{2\Delta_{abc}} & \frac{\beta_b}{2\Delta_{abc}} \\ \frac{\gamma_a}{2\Delta_{abc}} & \frac{\gamma_b}{2\Delta_{abc}} \end{pmatrix} \begin{pmatrix} \frac{\partial N_i^e}{\partial u} \\ \frac{\partial N_i^e}{\partial v} \end{pmatrix}$$
$$= \begin{pmatrix} \beta_a^* & \beta_b^* \\ \gamma_a^* & \gamma_b^* \end{pmatrix} \begin{pmatrix} \frac{\partial N_i^e}{\partial u} \\ \frac{\partial N_i^e}{\partial v} \end{pmatrix} \quad \text{---------- (38)}$$

where $\beta_a^* = \frac{\beta_a}{(2\Delta_{abc})}$, $\beta_b^* = \frac{\beta_b}{(2\Delta_{abc})}$
$\gamma_a^* = \frac{\gamma_a}{(2\Delta_{abc})}$, $\gamma_b^* = \frac{\gamma_b}{(2\Delta_{abc})}$ ---------- (39)

Letting,

$$D_{x,y}^{i,e} = \begin{pmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \end{pmatrix}, \quad P = \begin{pmatrix} \beta_a^* & \beta_b^* \\ \gamma_a^* & \gamma_b^* \end{pmatrix}, \quad D_{u,v}^{i,e} = \begin{pmatrix} \frac{\partial N_i^e}{\partial u} \\ \frac{\partial N_i^e}{\partial v} \end{pmatrix} \quad \text{---------- (40)}$$

We obtain from Eq.(24),

$$D_{x,y}^{i,e} = P\, D_{u,v}^{i,e} \quad \text{---------- (41)}$$

So that from Eq.(30) and Eq.(31) , we obtain

$$G_{x,y}^{i,j,e} = \begin{pmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \end{pmatrix} \left(\frac{\partial N_j^e}{\partial x} \quad \frac{\partial N_j^e}{\partial y} \right) = (D_{x,y}^{i,e})\,(D_{x,y}^{j,e})^T$$
$$= \begin{pmatrix} \frac{\partial N_i^e}{\partial x}\frac{\partial N_j^e}{\partial x} & \frac{\partial N_i^e}{\partial x}\frac{\partial N_j^e}{\partial y} \\ \frac{\partial N_i^e}{\partial y}\frac{\partial N_j^e}{\partial x} & \frac{\partial N_i^e}{\partial y}\frac{\partial N_j^e}{\partial y} \end{pmatrix} \quad \text{---------- (42a)}$$

$$G_{u,v}^{i,j,e} = \begin{pmatrix} \frac{\partial N_i^e}{\partial u} \\ \frac{\partial N_i^e}{\partial v} \end{pmatrix} \left(\frac{\partial N_j^e}{\partial u} \quad \frac{\partial N_j^e}{\partial v} \right) = (D_{u,v}^{i,e})\,(D_{u,v}^{j,e})^T$$
$$= \begin{pmatrix} \frac{\partial N_i^e}{\partial u}\frac{\partial N_j^e}{\partial u} & \frac{\partial N_i^e}{\partial u}\frac{\partial N_j^e}{\partial v} \\ \frac{\partial N_i^e}{\partial v}\frac{\partial N_j^e}{\partial u} & \frac{\partial N_i^e}{\partial v}\frac{\partial N_j^e}{\partial v} \end{pmatrix} \quad \text{---------- (42b)}$$

We have now from Eq.(27) and Eq.(28)

$$G_{x,y}^{i,j,e} = (P\, D_{u,v}^{i,e})\,((D_{u,v}^{j,e})^T P^T)$$
$$= P\, G_{u,v}^{i,j,e}\, P^T \quad \text{---------- (43)}$$

We now define the submatrices of global derivative integrals in (x,y) and (u,v) space associated with the nodes i and j as ;

$$S^{i,j,e} = \iint_{Q_e} G_{x,y}^{i,j,e}\, dx\, dy, \quad (e=1,2,3) \quad \text{---------- (44)}$$

$$K^{i,j,e} = \iint_{\hat{Q}} G_{u,v}^{i,j,e}\, du\, dv \quad \text{---------- (45)}$$

where, we have already defined the quadrilaterals Q_e (e=1,2,3) in (x,y) space and $\hat{Q}$ in (u,v) space in Fig 3a-b. From Eqs.(28)-(33) , 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.(32a)- (32b) as

$$S^{i,j,e} = \iint_{Q_e} G_{x,y}^{i,j,e}\, dx\, dy = \begin{pmatrix} \iint_{Q_e} \frac{\partial N_i^e}{\partial x}\frac{\partial N_j^e}{\partial x}\, dxdy & \iint_{Q_e} \frac{\partial N_i^e}{\partial x}\frac{\partial N_j^e}{\partial y}\, dxdy \\ \iint_{Q_e} \frac{\partial N_i^e}{\partial y}\frac{\partial N_j^e}{\partial x}\, dxdy & \iint_{Q_e} \frac{\partial N_i^e}{\partial y}\frac{\partial N_j^e}{\partial y}\, dxdy \end{pmatrix}$$
$$= \begin{pmatrix} S_{2i-1,2j-1}^e & S_{2i-1,2j}^e \\ S_{2i,2j-1}^e & S_{2i,2j}^e \end{pmatrix} \text{ (say)} \quad \text{----------(46)}$$

and in similar manner

$$K^{i,j,e} = \iint_{\hat{Q}} G_{u,v}^{i,j,e}\, du\, dv = \begin{pmatrix} \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial u}\frac{\partial N_j^e}{\partial u}\, dudv & \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial u}\frac{\partial N_j^e}{\partial v}\, dudv \\ \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial v}\frac{\partial N_j^e}{\partial u}\, dudv & \iint_{\hat{Q}} \frac{\partial N_i^e}{\partial v}\frac{\partial N_j^e}{\partial v}\, dudv \end{pmatrix}$$
$$= \begin{pmatrix} K_{2i-1,2j-1}^e & K_{2i-1,2j}^e \\ K_{2i,2j-1}^e & K_{2i,2j}^e \end{pmatrix} \text{ (say)} \quad \text{----------(47)}$$

We now obtain from the above Eq.(23)-(33)

$$S^{i,j,e} = \iint_{Q_e} G_{x,y}^{i,j,e}\, dx\, dy = \iint_{\hat{Q}} (P\, G_{u,v}^{i,j,e} P^T)\, \frac{\partial(x,y)}{\partial(u,v)}\, du\, dv$$
$$= 2\Delta_{abc} \iint_{\hat{Q}} (P\, G_{u,v}^{i,j,e} P^T)\, du\, dv$$
$$= 2\Delta_{abc}\, P \left(\iint_{\hat{Q}} G_{u,v}^{i,j,e}\, du\, dv \right) P^T$$
$$= 2\Delta_{abc}\, P\, (K^{i,j,e})\, P^T \quad \text{----------(48)}$$

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.(33).

DOI: 10.18535/ijecs/v5i12.75

From Eq.(35) and noting the fact that $\widehat{Q}$ is the quadrilateral in (u, v) space spanned by the vertices (1/3, 1/3), (0, 1/2), (0, 0)and (1/2, 0) we obtain

$$\mathbf{K^{i,j,e}} = \iint_{\widehat{Q}} \mathbf{G^{i,j,e}_{u,v}}\ \mathbf{du\ dv}$$
$$= \int_{-1}^{1}\int_{-1}^{1} \mathbf{G^{i,j,e}_{u,v}}\ \frac{\partial(u,v)}{\partial(\xi,\eta)}\ \mathbf{d\xi\ d\eta} \qquad \text{----------(49)}$$

We now refer to section 5 of this paper, in this section we have derived the necessary relations to integrate the integrals of Eq.(39), As in Eq.(22a-d) , we use the transformation

$$u(\xi,\ \eta) = \frac{1}{3}N_1(\xi,\ \eta) + \frac{1}{2}N_4(\xi,\ \eta)$$
$$v(\xi,\ \eta) = \frac{1}{3}N_1(\xi,\ \eta) + \frac{1}{2}N_2(\xi,\ \eta) \qquad \text{----------(50)}$$

to map the quadrilateral $\widehat{Q}$ to the 2-square $-1 \le \xi, \eta \le 1$ Using Eq.(40) in Eq.(39), we obtain

$$\mathbf{K^{i,j,e}} = \iint_{\widehat{Q}} \mathbf{G^{i,j,e}_{u,v}}\ \left(\frac{4+\xi+\eta}{96}\right)\mathbf{d\xi d\eta} \qquad \text{---------- (51)}$$

The submatrices for the quadrilateral Q_e is expressed from Eq.(38) as

$$\mathbf{S^{i,j,e}} = (2\Delta_{abc})\ \mathbf{P}\ (\mathbf{K^{i,j,e}})\ \mathbf{P^T} \qquad \text{---------- (52)}$$

In eqn.(38) , $2\Delta_{abc} = 2$ x area of the triangle spanning vertices A(x_a ,y_a) ,B(x_b ,y_b) , C(x_c , y_c) which is 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 $\mathbf{K^{i,j,e}}$ can be explicity computed by the relations obtained in section 2 and 3 . We find that $\mathbf{K^{i,j,e}}$ 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 Eq.(33) is of order $(2xn_{de})$ x $(2xn_{de})$. We have computed K^e for the four noded isoparametric quadrilateral element.

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

where, (i,j=1,2,3,4)

Table-A
Analytical Values of $\mathbf{K^{i,j,e}}$ for the 4 Noded Quadrilateral spanned by vertices (1/3, 1/3), (0, ½), (0, 0) and (½, 0)

$$\mathbf{K^{1,1,e}} = \begin{pmatrix} -11/2 - 34*\log(2) + 27*\log(3), & -1/2 - 20*\log(2) + 27/2*\log(3) \\ -1/2 - 20*\log(2) + 27/2*\log(3), & -11/2 - 34*\log(2) + 27*\log(3) \end{pmatrix}$$

$$\mathbf{K^{1,2,e}} = \begin{pmatrix} 11/3 + 68/3*\log(2) - 18*\log(3), & 5/6 + 40/3*\log(2) - 9*\log(3) \\ 1/3 + 40/3*\log(2) - 9*\log(3), & 25/6 + 68/3*\log(2) - 18*\log(3) \end{pmatrix}$$

$$\mathbf{K^{1,3,e}} = \begin{pmatrix} [\ -7/3 - 34/3*\log(2) + 9*\log(3), & -2/3 - 20/3*\log(2) + 9/2*\log(3) \\ -2/3 - 20/3*\log(2) + 9/2*\log(3), & -7/3 - 34/3*\log(2) + 9*\log(3) \end{pmatrix}$$

$$\mathbf{K^{1,4,e}} = \begin{pmatrix} 25/6 + 68/3*\log(2) - 18*\log(3), & 1/3 + 40/3*\log(2) - 9*\log(3) \\ 5/6 + 40/3*\log(2) - 9*\log(3), & 11/3 + 68/3*\log(2) - 18*\log(3) \end{pmatrix}$$

$$\mathbf{K^{2,1,e}} = \begin{pmatrix} 11/3 + 68/3*\log(2) - 18*\log(3), & 1/3 + 40/3*\log(2) - 9*\log(3) \\ 5/6 + 40/3*\log(2) - 9*\log(3), & 25/6 + 68/3*\log(2) - 18*\log(3) \end{pmatrix}$$

$$\mathbf{K^{2,2,e}} = \begin{pmatrix} -22/9 - 136/9*\log(2) + 12*\log(3), & -5/9 - 80/9*\log(2) + 6*\log(3) \\ -\frac{5}{9} - \frac{80}{9}*\log(2) + 6*\log(3), & -22/9 - 136/9*\log(2) + 12*\log(3) \end{pmatrix}$$

$$\mathbf{K^{2,3,e}} = \begin{pmatrix} \frac{14}{9} + \frac{68}{9}*\log(2) - 6*\log(3), & \frac{4}{9} + \frac{40}{9}*\log(2) - 3*\log(3) \\ -\frac{1}{18} + \frac{40}{9}*\log(2) - 3*\log(3), & \frac{19}{18} + \frac{68}{9}*\log(2) - 6*\log(3) \end{pmatrix}$$

$$\mathbf{K^{2,4,e}} = \begin{pmatrix} -\frac{25}{9} - \frac{136}{9}*\log(2) + 12*\log(3), & -\frac{2}{9} - \frac{80}{9}*\log(2) + 6*\log(3) \\ -2/9 - 80/9*\log(2) + 6*\log(3), & -25/9 - 136/9*\log(2) + 12*\log(3) \end{pmatrix}$$

$$\mathbf{K^{3,1,e}} = \begin{pmatrix} -\frac{7}{3} - \frac{34}{3}*\log(2) + 9*\log(3), & -\frac{2}{3} - \frac{20}{3}*\log(2) + \frac{9}{2}*\log(3) \\ -2/3 - 20/3*\log(2) + 9/2*\log(3), & -7/3 - 34/3*\log(2) + 9*\log(3) \end{pmatrix}$$

$$\mathbf{K^{3,2,e}} = \begin{pmatrix} \frac{14}{9} + \frac{68}{9}*\log(2) - 6*\log(3), & -\frac{1}{18} + \frac{40}{9}*\log(2) - 3*\log(3\ ) \\ 14/9 + 68/9*\log(2) - 6*\log(3), & -1/18 + 40/9*\log(2) - 3*\log(3 \end{pmatrix}$$

$$\mathbf{K}^{3,3,e}=\begin{pmatrix} -\frac{5}{18}-\frac{34}{9}*\log(2)+3*\log(3), \frac{5}{18}-\frac{20}{9}*\log(2)+\frac{3}{2}*\log(3) \\ 5/18-20/9*\log(2)+3/2*\log(3),\ -5/18-34/9*\log(2)+3*\log(3) \end{pmatrix}$$

$$\mathbf{K}^{3,4,e}=\begin{pmatrix} \frac{19}{18}+\frac{68}{9}*\log(2)-6*\log(3),\ \frac{4}{9}+\frac{40}{9}*\log(2)-3*\log(3) \\ -1/18+40/9*\log(2)-3*\log(3),\ 14/9+68/9*\log(2)-6*\log(3)] \end{pmatrix}$$

$$\mathbf{K}^{4,1,e}=\begin{pmatrix} \frac{25}{6}+\frac{68}{3}*\log(2)-18*\log(3),\ \frac{5}{6}+\frac{40}{3}*\log(2)-9*\log(3) \\ 1/3+40/3*\log(2)-9*\log(3), 11/3+68/3*\log(2)-18*\log(3) \end{pmatrix}$$

$$\mathbf{K}^{4,2,e}=\begin{pmatrix} -25/9-136/9*\log(2)+12*\log(3), & -2/9-80/9*\log(2)+6*\log(3) \\ -2/9-80/9*\log(2)+6*\log(3), & -25/9-136/9*\log(2)+12*\log(3) \end{pmatrix}$$

$$\mathbf{K}^{4,3,e}=\begin{pmatrix} 19/18+68/9*\log(2)-6*\log(3), & -1/18+40/9*\log(2)-3*\log(3) \\ 4/9+40/9*\log(2)-3*\log(3), & 14/9+68/9*\log(2)-6*\log(3) \end{pmatrix}$$

$$\mathbf{K}^{4,4,e}=\begin{pmatrix} -22/9-136/9*\log(2)+12*\log(3), & -5/9-80/9*\log(2)+6*\log(3) \\ -5/9-80/9*\log(2)+6*\log(3), & -22/9-136/9*\log(2)+12*\log(3) \end{pmatrix}$$

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)

$$K_{i,j}^{e}=\int_{\Omega^e}\nabla\varphi_i\,.\nabla\varphi_j\,dx=\int_{\Omega^e}\left\{\frac{\partial\varphi_i}{\partial x}\frac{\partial\varphi_j}{\partial x}+\frac{\partial\varphi_i}{\partial y}\frac{\partial\varphi_j}{\partial y}\right\}dxdy\,, \qquad \text{,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,(53a)}$$

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

$$K_{i,j}^{e}=\int_{Q_e}\nabla N_i\,.\nabla N_j\,dx=\int_{Q_e}\left\{\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}+\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\right\}dxdy=S_{2i-1,2j-1}^{e}+S_{2i,2j}^{e} \qquad \text{....................................(53b)}$$

3.2 Computation of $K_{i,j}^{e}$

The explicit integration scheme explained above compute four derivative product integrals as given in eqn(36) 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 (42).Since, only the sum of two of these integrals is needed viz : $S_{2i-1,2j-1}^{e}+S_{2i,2j}^{e}$,we now present an efficient method to compute this sum by using matrix product.

Let $F_{p,q}^{i,j}=\frac{\partial N_i}{\partial p}\frac{\partial N_j}{\partial q}$, $I_{p,q}^{i,j}=\int_{Q_e}F_{p,q}^{i,j}\,dpdq$, then we have from eqns(36-37) :

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

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

$$=\begin{pmatrix}I_{x,x}^{i,j} & I_{x,y}^{i,j}\\ I_{y,x}^{i,j} & I_{y,y}^{i,j}\end{pmatrix} \qquad \text{..(54a)}$$

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

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

$$=\begin{pmatrix}I_{u,u}^{i,j} & I_{u,v}^{i,j}\\ I_{v,u}^{i,j} & I_{v,v}^{i,j}\end{pmatrix} \qquad \text{..(54b)}$$

Let $P=\begin{pmatrix}P_{11} & P_{12}\\ P_{21} & P_{22}\end{pmatrix},\ P^T=\begin{pmatrix}P_{11} & P_{21}\\ P_{12} & P_{22}\end{pmatrix}$ **..(55)**

From eqns(44a-b) and (45)

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

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

DOI: 10.18535/ijecs/v5i12.75

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

.....(56)

From eqn(43b) ,eqn(44a) and eqn(46) , we find

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

$$=(P_{11}^2+P_{21}^2)I^{i,j}_{u,u}+(P_{11}P_{12}+P_{21}P_{22})(I^{i,j}_{u,v}+I^{i,j}_{v,u})+=(P_{12}^2+P_{22}^2)I^{i,j}_{v,v} \quad \text{.....(57)}$$

We 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:

$$(P^T P).*\begin{pmatrix}I^{i,j}_{u,u} & I^{i,j}_{u,v}\\ I^{i,j}_{v,u} & I^{i,j}_{v,v}\end{pmatrix}=\begin{bmatrix}(P_{11}^2+P_{21}^2) & (P_{11}P_{12}+P_{21}P_{22})\\ (P_{11}P_{12}+P_{22}P_{21}) & (P_{12}^2+P_{22}^2)\end{bmatrix} \quad \text{.....(58)}$$

We observe from eqn(48) that sum of all the entries gives us the value of the integral i.e

$$\int_{Q_e}\{\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}+\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\}dxdy=sum(sum\left(\left(P^T P\right).*\begin{pmatrix}I^{i,j}_{u,u} & I^{i,j}_{u,v}\\ I^{i,j}_{v,u} & I^{i,j}_{v,v}\end{pmatrix}\right)) \quad \text{.....(59)}$$

Where,sum is aq 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 .

3.3 Products of Shape Function and Global Derivative Integrals : $L^e_{i,j}=\int_{\Omega^e}[\,\varphi_i\ \frac{\partial\varphi_j}{\partial x_1}\ \ \varphi_i\ \frac{\partial\varphi_j}{\partial x_2}]^T dx$

In section 3.1,we have shown that

$$\begin{pmatrix}\frac{\partial N^e_j}{\partial x}\\ \frac{\partial N^e_j}{\partial y}\end{pmatrix}=\begin{pmatrix}\frac{\beta_a}{2\Delta_{abc}} & \frac{\beta_b}{2\Delta_{abc}}\\ \frac{\gamma_a}{2\Delta_{abc}} & \frac{\gamma_b}{2\Delta_{abc}}\end{pmatrix}\begin{pmatrix}\frac{\partial N^e_j}{\partial u}\\ \frac{\partial N^e_j}{\partial v}\end{pmatrix}$$

$$=\begin{pmatrix}\beta_a^* & \beta_b^*\\ \gamma_a^* & \gamma_b^*\end{pmatrix}\begin{pmatrix}\frac{\partial N^e_j}{\partial u}\\ \frac{\partial N^e_j}{\partial v}\end{pmatrix} \quad \text{---------- (38)}$$

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

$\gamma_a^*=\frac{\gamma_a}{(2\Delta_{abc})}$, $\gamma_b^*=\frac{\gamma_b}{(2\Delta_{abc})}$ ---------- (39)

Letting,

$D^{j,e}_{x,y}=\begin{pmatrix}\frac{\partial N^e_j}{\partial x}\\ \frac{\partial N^e_j}{\partial y}\end{pmatrix}$, $P=\begin{pmatrix}\beta_a^* & \beta_b^*\\ \gamma_a^* & \gamma_b^*\end{pmatrix}$, $D^{j,e}_{u,v}=\begin{pmatrix}\frac{\partial N^e_j}{\partial u}\\ \frac{\partial N^e_j}{\partial v}\end{pmatrix}$ ---------- (40)

We obtain from Eq.(41),

$D^{j,e}_{x,y}=P\,D^{j,e}_{u,v}$

From eqn(38),we have

$$\begin{pmatrix}\frac{\partial N^e_j}{\partial u}\\ \frac{\partial N^e_j}{\partial v}\end{pmatrix}=\frac{1}{J}\begin{bmatrix}\frac{\partial v}{\partial\eta} & -\frac{\partial v}{\partial\xi}\\ -\frac{\partial u}{\partial\eta} & \frac{\partial u}{\partial\xi}\end{bmatrix}\begin{bmatrix}\frac{\partial N^e_j}{\partial\xi}\\ \frac{\partial N^e_j}{\partial\eta}\end{bmatrix} \quad \text{---------- (29)}$$

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.(27a)–(27d) while J is defined in Eq.(28a-b) , ($i,j=1,2,3,-----$, nde) , nde = the number of nodes per element.

and $J=\frac{\partial(u,v)}{\partial(\xi,\eta)}=\frac{1}{96}(4+\xi+\eta)$

Hence from eqn(38) above

DOI: 10.18535/ijecs/v5i12.75

$$\begin{pmatrix} \frac{\partial N_j^e}{\partial x} \\ \frac{\partial N_j^e}{\partial y} \end{pmatrix} = \begin{pmatrix} \frac{\beta_a}{2\Delta_{abc}} & \frac{\beta_b}{2\Delta_{abc}} \\ \frac{\gamma_a}{2\Delta_{abc}} & \frac{\gamma_b}{2\Delta_{abc}} \end{pmatrix} \begin{pmatrix} \frac{\partial N_j^e}{\partial u} \\ \frac{\partial N_j^e}{\partial v} \end{pmatrix}$$

$$= \begin{pmatrix} \beta_a^* & \beta_b^* \\ \gamma_a^* & \gamma_b^* \end{pmatrix} \frac{1}{J} \begin{bmatrix} \frac{\partial v}{\partial \eta} & -\frac{\partial v}{\partial \xi} \\ -\frac{\partial u}{\partial \eta} & \frac{\partial u}{\partial \xi} \end{bmatrix} \begin{bmatrix} \frac{\partial N_j^e}{\partial \xi} \\ \frac{\partial N_j^e}{\partial \eta} \end{bmatrix}$$

$$N_i^e \begin{pmatrix} \frac{\partial N_j^e}{\partial x} \\ \frac{\partial N_j^e}{\partial y} \end{pmatrix} = \begin{pmatrix} \beta_a^* & \beta_b^* \\ \gamma_a^* & \gamma_b^* \end{pmatrix} \frac{1}{J} \begin{bmatrix} \frac{\partial v}{\partial \eta} & -\frac{\partial v}{\partial \xi} \\ -\frac{\partial u}{\partial \eta} & \frac{\partial u}{\partial \xi} \end{bmatrix} N_i^e \begin{bmatrix} \frac{\partial N_j^e}{\partial \xi} \\ \frac{\partial N_j^e}{\partial \eta} \end{bmatrix}$$

$$\iint_{Q_e} N_i^e \begin{pmatrix} \frac{\partial N_j^e}{\partial x} \\ \frac{\partial N_j^e}{\partial y} \end{pmatrix} dxdy = 2\Delta_{abc} \iint_{\bar{Q}} \begin{pmatrix} \beta_a^* & \beta_b^* \\ \gamma_a^* & \gamma_b^* \end{pmatrix} \frac{1}{J} \begin{bmatrix} \frac{\partial v}{\partial \eta} & -\frac{\partial v}{\partial \xi} \\ -\frac{\partial u}{\partial \eta} & \frac{\partial u}{\partial \xi} \end{bmatrix} N_i^e \begin{bmatrix} \frac{\partial N_j^e}{\partial \xi} \\ \frac{\partial N_j^e}{\partial \eta} \end{bmatrix} dudv \quad \text{----------(60)}$$

$$= (2\Delta_{abc}) P \int_{-1}^{1}\int_{-1}^{1} \frac{1}{J} \begin{bmatrix} \frac{\partial v}{\partial \eta} & -\frac{\partial v}{\partial \xi} \\ -\frac{\partial u}{\partial \eta} & \frac{\partial u}{\partial \xi} \end{bmatrix} N_i^e \begin{bmatrix} \frac{\partial N_j^e}{\partial \xi} \\ \frac{\partial N_j^e}{\partial \eta} \end{bmatrix} \frac{\partial(u,v)}{\partial(\xi,\eta)} \, d\xi \, d\eta \text{ ; since } \frac{\partial(u,v)}{\partial(\xi,\eta)} = J$$

$$= (2\Delta_{abc}) P \begin{pmatrix} \int_{-1}^{1}\int_{-1}^{1} [\frac{\partial v}{\partial \eta} N_i^e \frac{\partial N_j^e}{\partial \xi} - \frac{\partial v}{\partial \xi} N_i^e \frac{\partial N_j^e}{\partial \eta}] \, d\xi d\eta \\ \int_{-1}^{1}\int_{-1}^{1} [-\frac{\partial u}{\partial \eta} N_i^e \frac{\partial N_j^e}{\partial \xi} + \frac{\partial N_i^e}{\partial \xi} N_i^e \frac{\partial N_j^e}{\partial \eta}] d\xi d\eta \end{pmatrix} \quad \text{----------(61a)}$$

$$= \begin{pmatrix} \beta_a & \beta_b \\ \gamma_a & \gamma_b \end{pmatrix} \begin{pmatrix} \int_{-1}^{1}\int_{-1}^{1} [\frac{\partial v}{\partial \eta} N_i^e \frac{\partial N_j^e}{\partial \xi} - \frac{\partial v}{\partial \xi} N_i^e \frac{\partial N_j^e}{\partial \eta}] \, d\xi d\eta \\ \int_{-1}^{1}\int_{-1}^{1} [-\frac{\partial u}{\partial \eta} N_i^e \frac{\partial N_j^e}{\partial \xi} + \frac{\partial N_i^e}{\partial \xi} N_i^e \frac{\partial N_j^e}{\partial \eta}] d\xi d\eta \end{pmatrix} \quad \text{----------(61b)}$$

$$= \begin{pmatrix} \beta_a & \beta_b \\ \gamma_a & \gamma_b \end{pmatrix} \begin{pmatrix} \text{IntNiDNj}(2i-1,j) \\ \text{IntNiDNj}(2i,j) \end{pmatrix} \quad \text{----------(61c)}$$

Where

$$\text{IntNiDNj}(2i-1,j) = \int_{-1}^{1}\int_{-1}^{1} [\frac{\partial v}{\partial \eta} N_i^e \frac{\partial N_j^e}{\partial \xi} - \frac{\partial v}{\partial \xi} N_i^e \frac{\partial N_j^e}{\partial \eta}] \, d\xi d\eta \quad \text{----------(61d)}$$

$$\text{IntNiDNj}(2i,j) = \int_{-1}^{1}\int_{-1}^{1} [-\frac{\partial u}{\partial \eta} N_i^e \frac{\partial N_j^e}{\partial \xi} + \frac{\partial N_i^e}{\partial \xi} N_i^e \frac{\partial N_j^e}{\partial \eta}] d\xi d\eta \quad \text{----------(61e)}$$

(i,j=1,2,3,4)

For the 4-node 2-square element,we obtain

```
[IntNiDNj]=[ 1/12 -1/18 -1/24  1/72;...
             1/12  1/72 -1/24 -1/18;...
             1/12 -1/18 -1/18  1/36;...
             1/24  1/18 -5/72 -1/36;...
             1/24 -1/36 -1/12  5/72;...
             1/24  5/72 -1/12 -1/36;...
             1/24 -1/36 -5/72  1/18;...
             1/12  1/36 -1/18 -1/18];          ----------(61f)
```

3.3 Products of Shape Function Integrals : $M_{i,j}^e = \int_{\Omega^e} \varphi_i \varphi_j \, dx$

We can rewrite $M_{i,j}^e$ in the notations of this sections by taking $\varphi_i = N_i$ and $\varphi_j = N_j$ and $\Omega^e = Q_e$ so that

$$M_{i,j}^e = \int_{Q_e} N_i N_j \, dx = \int_{Q_e} N_i N_j \, dxdy$$

$$= 2\Delta_{abc} \iint_{\bar{Q}} N_i N_j \, du \, dv$$

$$= 2\Delta_{abc} \int_{-1}^{1}\int_{-1}^{1} N_i N_j \frac{\partial(u,v)}{\partial(\xi,\eta)} \, d\xi \, d\eta$$

$$= 2\Delta_{abc} \int_{-1}^{1}\int_{-1}^{1} N_i N_j \frac{1}{96}(4+\xi+\eta) \, d\xi \, d\eta \quad \text{----------(62a)}$$

$$= 2\Delta_{abc} \, \text{IntJNiNj} \quad \text{----------(62b)}$$

Where

$$\text{IntJNiNj} = \int_{-1}^{1}\int_{-1}^{1} N_i N_j \frac{1}{96}(4+\xi+\eta) \, d\xi \, d\eta = M_{i,j}^e / 2\Delta_{abc} \quad \text{----------(62c)}$$

For the 4-node 2-square element,we obtain

```
[IntJNiNj] =[  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];      ----------(62d)
```

DOI: 10.18535/ijecs/v5i12.75

3.4 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 $\phi(x, y)$

$$I_{\Delta PQR} = \iint_{\Delta PQR} \phi(x,y)\, dxdy = \sum_{e=1}^{3} \iint_{Q_e} \phi(x,y)\, dxdy \quad \text{---------- (63)}$$

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

$$= (2\,\Delta_{pqr}) \iint_{\hat{Q}} \{ \sum_{e=1}^{3} [\phi\,(x^{(e)}(u,v),\ y^{(e)}(u,v))]\}\, dudv \quad \text{---------- (64)}$$

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

$$I_{\Delta PQR} = (2\,\Delta_{pqr}) \iint_{\hat{Q}} \{ \sum_{e=1}^{3} [\phi\,(x^{(e)}(u,v),\ y^{(e)}(u,v)) \frac{\partial(u,v)}{\partial(\xi,\eta)}\}\, d\xi\, d\eta \quad \text{---------- (65)}$$

In Eq.(14) we have used the transformation

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

$$v(\xi,\ \eta) = \frac{1}{3} N_1(\xi,\ \eta) + \frac{1}{2} N_2(\xi,\ \eta) \quad \text{---------- (66)}$$

to map the quadrilateral $\hat{Q}$ into a 2 – square in $\xi\eta$ – plane.
We can now obtain from Eqs.(14)–(15)

$$I_{\Delta PQR} = (2\,\Delta_{pqr}) \int_{-1}^{1} \int_{-1}^{1} [\sum_{e=1}^{3} \left(\frac{4+\xi+\eta}{96}\right) \phi(x^{(e)}(u,v),\ y^{(e)}(u,v))]\, d\xi\, d\eta \quad \text{---------- (67)}$$

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

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

---------- (68)

Where,

$$u_{i,j}^{(N)} = u(\xi_i^{(N)},\ \eta_j^{(N)}) \text{ and } v_{i,j}^{(N)} = v(\xi_i^{(N)},\ \eta_j^{(N)}) \quad \text{---------- (69)}$$

and $(W_i^{(N)},\ \xi_i^{(N)})$, $(W_j^{(N)},\ \xi_j^{(N)})$ are the weight coefficients and sampling points of N^{th} 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

4.0 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.

4.1 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.

4.2 Division of an Arbitrary Triangle

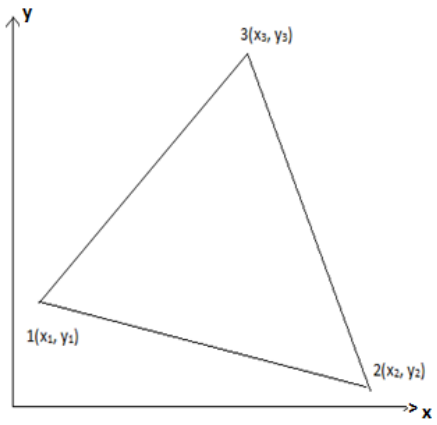
DOI: 10.18535/ijecs/v5i12.75

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.

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

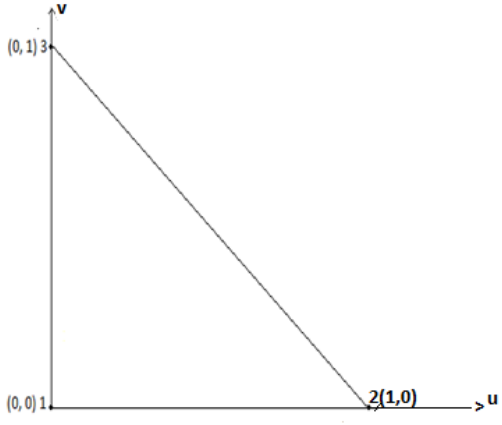
$$y = y_1 + (y_2 - y_1)u + (y_3 - y_1)v \quad \textbf{(70)}$$

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

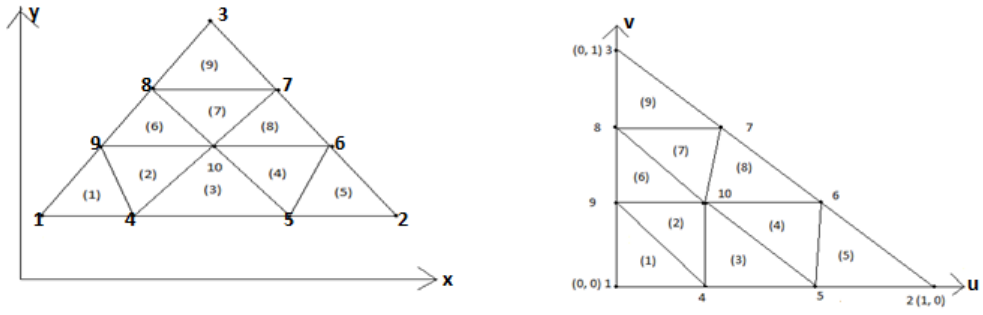
Fig. 4a An Arbitrary Linear Triangle in the (x, y) space



4 b

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

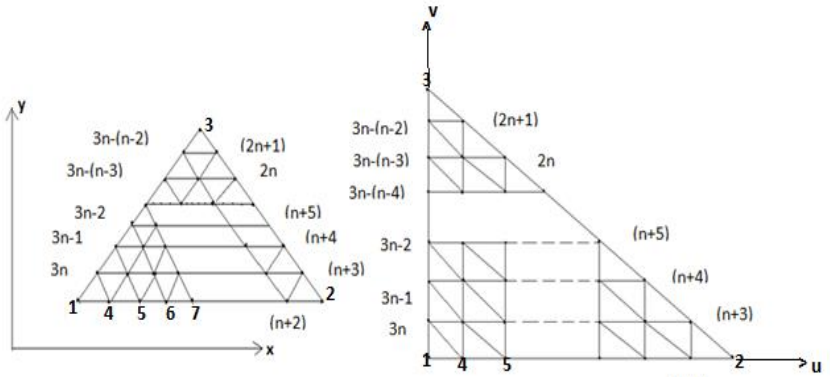
DOI: 10.18535/ijecs/v5i12.75



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



6a 6b

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 $+v = 1$. 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

DOI: 10.18535/ijecs/v5i12.75

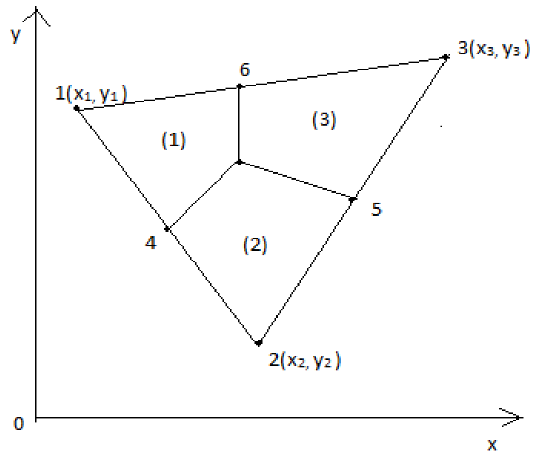
$$\underline{rr} = \begin{bmatrix} 1, & 4, & 5, \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots., & (n+2) & 2 \\ 3n, & (3n+1), \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..,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

....(71)

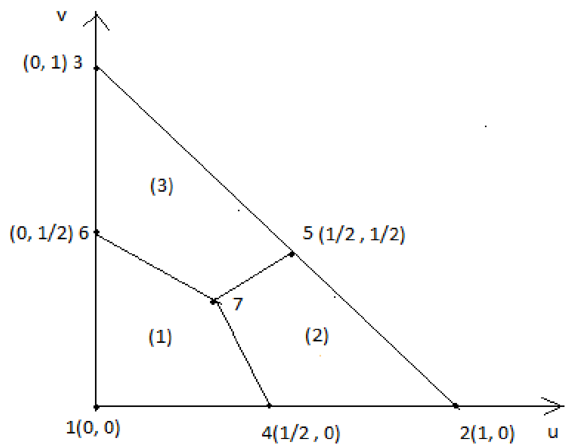
4.3. Quadrangulation of an Arbitrary Triangle

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)



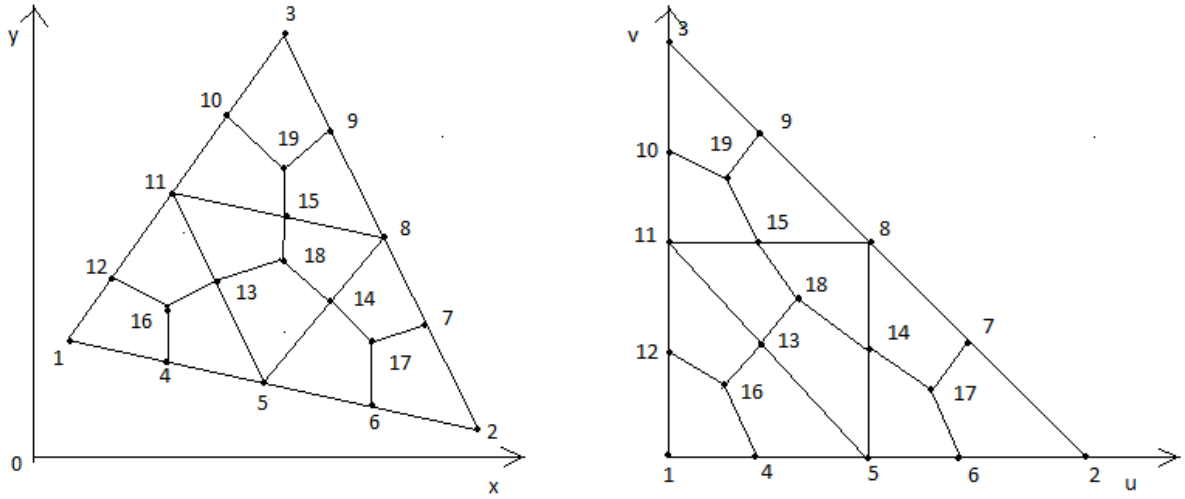
7(b)

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

DOI: 10.18535/ijecs/v5i12.75



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 triangle using sub matrix $\underline{rr}$

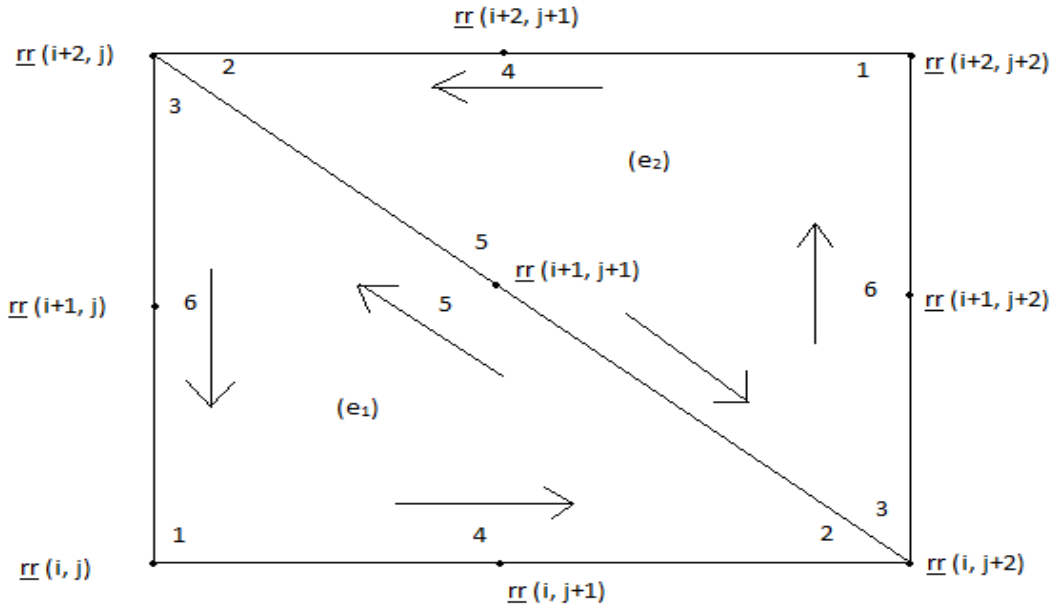


Fig. 9 Six node triangle formation for non zero sub matrix $\underline{rr}$

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) < $\underline{rr}\ \ (i+2, j+2), \underline{rr}\,(i+2, j), \underline{rr}\,(i, j+2), \underline{rr}\,(i+2, j+1), \underline{rr}\,(i+1, j+1), \underline{rr}\ \ i+1, j+2)$ > .....(72)

DOI: 10.18535/ijecs/v5i12.75

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 , $\underline{rr}$ $(i+1,\ j+1)$, $\underline{rr}$ $(i+2,\ j)$, $\underline{rr}$ $(i+1, j)$ > .....(73)

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

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

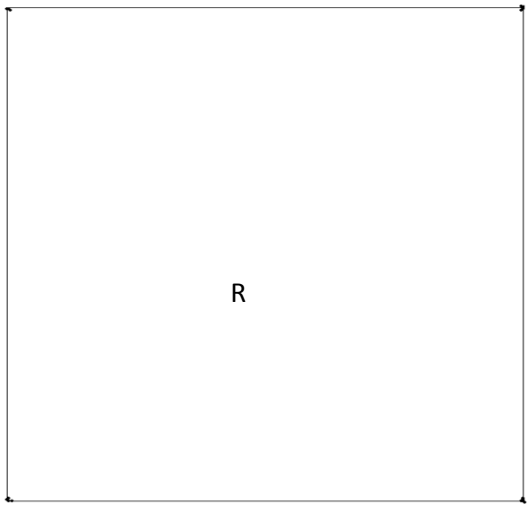
Q_{3n_2-1} < c_2 , $\underline{rr}$ $(i+2,\ j+1)$, $\underline{rr}$ $(i+2,\ j)$, $\underline{rr}$ $(i+1,\ j+1)$ >

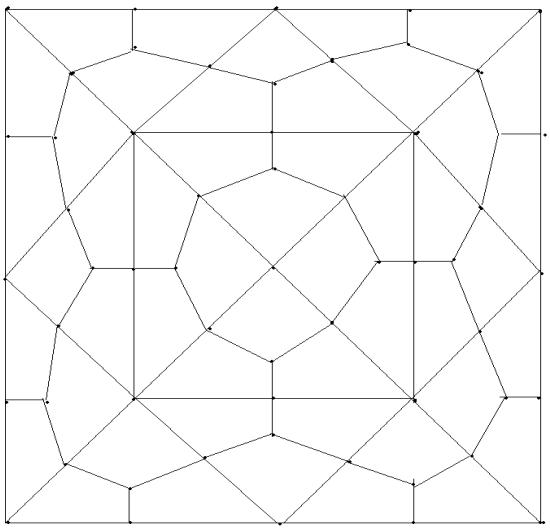
Q_{3n_1} < c_2 , $\underline{rr}$ $(i+1,\ j+1)$, $\underline{rr}$ $(i,\ j+2)$, $\underline{rr}$ $(i+1, j+2)$ > ---------- (74)

4.4 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)

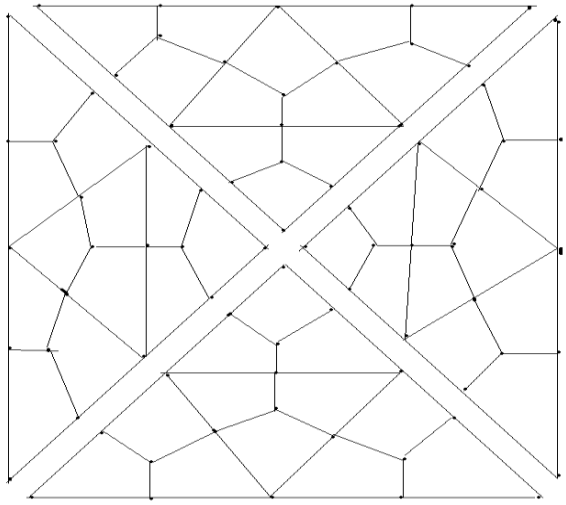


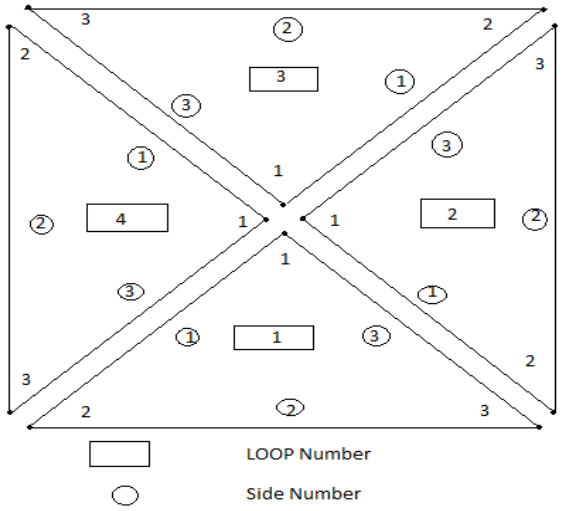


10a **10b**

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

DOI: 10.18535/ijecs/v5i12.75





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

4.5. Application Programs

4.5.1 Mesh Generation Over an Arbitrary Triangle

In applications to boundary value problems due to symmetry considerations, 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 vertices. We use the theory and procedure developed in section 2 and section 3 of this paper for this purpose. The following MATLAB codes were written for this purpose.

1. **polygonal_domain_coordinates. m**
2. **nodaladdresses_special_convex_quadrilaterals. m**
3. **generate_area_coordinate_over_standard_triangle. m**
4. **quadrilateral_mesh4arbitrarytriangle_q4. m**

4.5.2 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 convex quadrilaterals for the complex domains such as those in [21,22]. We use the theory and procedure developed in sections 2, 3 and 4 for this purpose. The following MATLAB codes were written for this purpose.

1. **quadrilateral_mesh4MOINEX_q4. m**
2. **nodaladdresses_special_convex_quadrilaterals_trial. m**
3. **polygonal_domain_coordinates. m**
4. **generate_area_coordinate_over_standard_triangle. m**

(5) quadrilateral_mesh4convexpolygonsixside_q4. m

5.0 Application Examples

In our recent papers [ ],we have used 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 conditions:

$-\Delta u = f, \forall\, x \epsilon \Omega \subset \mathcal{R}^2$, $u = g,\ \ on\ x \epsilon \partial\Omega$,where Ω is a polygonal domain and Δ is the standard Laplace operator

In this paper,we have enhanced the application the explicit integration scheme and the auto mesh generation techniques proposed in this paper to solve boundary value probems in two space for elliptic equations with constant coefficients subject to Dirichlet boundary conditions:

$-b_0 \Delta u + b_1 u_x + b_2 u_y + cu = f$, all x$\in \Omega$ ----------(16a)

u = g on $\Gamma = \partial\Omega$ ----------(16b)

We consider some elliptic boundary value problems of the following types

The reaction diffusion equation

$-\Delta u + cu = f$, in Ω ----------(11)

and the convection diffusion equation

$-\epsilon \Delta u + b_1 \frac{\partial u}{\partial x} + b_2 \frac{\partial u}{\partial y} + cu = f$, in Ω ,$\epsilon > 0$ ----------(12)

We have written the following codes to solve the Elliptic Equations with constant coefficients subject to Dirichlet Boundary Conditions over linear convex polygonal domains

Example 1

Let $\Omega = [-1, 1]^2$. As first example, we consider the boundary value problem: Find u such that

$\frac{-1}{\lambda} \Delta u + \lambda u = f_\lambda(x, y)$ in Ω ----------(75a)

u = 0, on $\partial\Omega$ ----------(75b)

where f_λ(x, y) = $\frac{\lambda^2 + 2\pi^2}{\lambda}$ sin (π x) sin (π y) and λ is a real parameter. The exact solution to this problem is u(x, y) = sin π x sin π y. The variational formulation for (75a-b) is obtained by multiplying the equation by smooth test functions v and integrating over the domain . It reads as

We seek a solution to this problem in the Hilbert space V = $H_0^1(\Omega)$ which requires the weak formulation of the equation. This formulation is obtained by first multiplying the differential equation by smooth test functions v $\in C_0^\infty(\Omega)$ and integrating over the domain. Via integration by parts we have

Find u $\in H_0^1(\Omega)$: a(u, v) = f(v), for all v $\in H_0^1(\Omega)$

where the bilinear form a(u, v) is given by

a(u, v) = $\frac{1}{\lambda}\int_\Omega \nabla u \cdot \nabla v\, dx + \lambda \int_\Omega uv\, dx$ ----------(76a)

and the linear form

f(v) = $\int_\Omega f_\lambda v\, dx$ ----------(76b)

For the discrete approximation of this problem, we use the quadrilateral shape functions and define the stiffness matrix K and the mass matrix M with entries

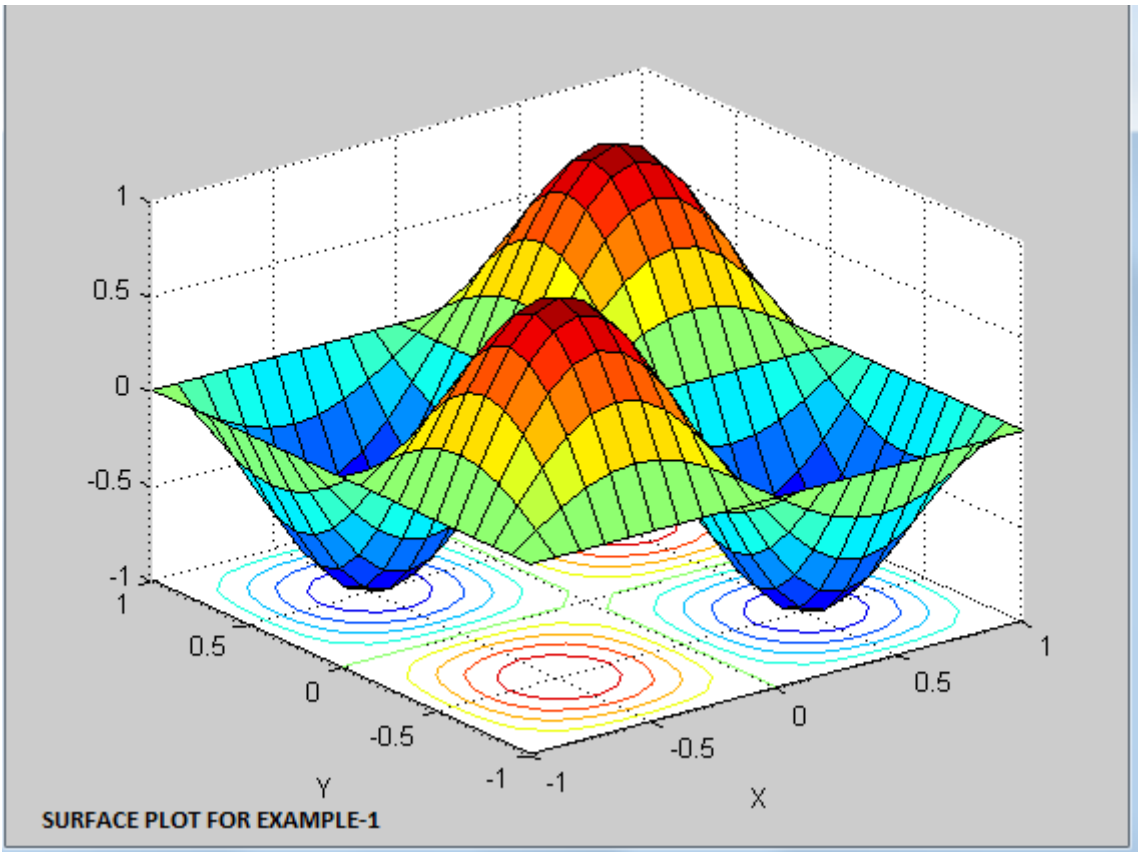
$K_{i,j} = \int_\Omega \nabla \varphi_i \cdot \nabla \varphi_j\, dx$ and $M_{i,j} = \int_\Omega \varphi_i \varphi_j\, dx$

For the right hand side we define the vector f with entries $f_i = \int_\Omega f_\lambda \varphi_i\, dx$

With this notation starting from the variational formulation of (76a-b) we arrive at the linear system

Find u $\in R^N$: ($\frac{1}{\lambda}$K+λM)$u = f$ ----------(77)

DOI: 10.18535/ijecs/v5i12.75



Three dimensional surface plot of the exact solution: u(x, y) = sin (π x) sin(π y)

Example 2

Let $\Omega = [0,1]^2$ as in first example, consider the boundary value problem: Find u such that

$-\epsilon^2 \Delta u + 2u = f$ **in** Ω ----------(78a)

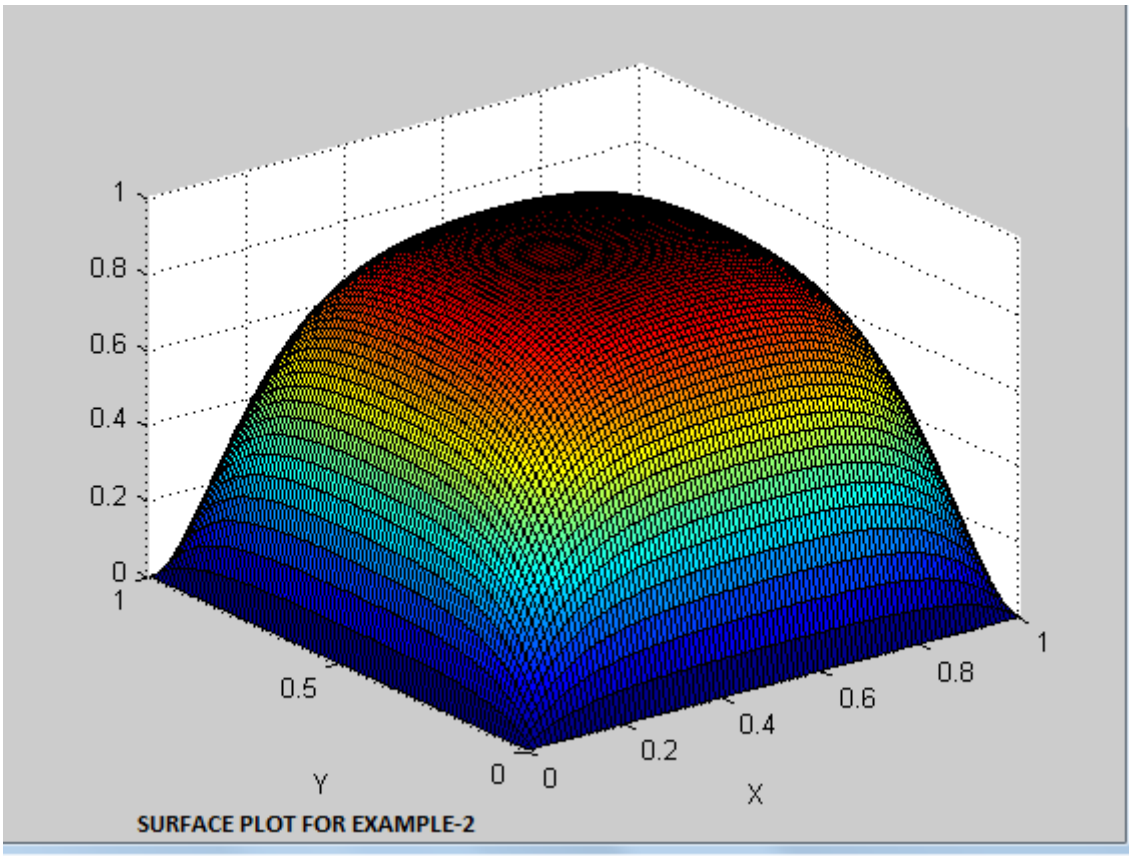
u = 0, on $\partial\Omega$ ----------(78b)

($\epsilon = 10^{-2}$ *to* 10^{-8} **)**

Where f **is chosen such that the solution of eqn(78a-b) is**

$$u(x,y) = \left(1 - \frac{e^{-x/\epsilon} + e^{-(1-x)/\epsilon}}{1+e^{-1/\epsilon}}\right)\left(1 - \frac{e^{-y} + e^{-(1-y)/\epsilon}}{1+e^{-1/\epsilon}}\right) \quad \text{----------(79)}$$

DOI: 10.18535/ijecs/v5i12.75



Three dimensional surface plot of the exact solution: $u(x,y) = \left(1 - \frac{e^{-x/\epsilon} + e^{-(1-x)/\epsilon}}{1+e^{-1/\epsilon}}\right)\left(1 - \frac{e^{-y} + e^{-(1-y)/\epsilon}}{1+e^{-1/\epsilon}}\right)$

Where $\epsilon = 10^{-2}$

Example 3

Consider the boundary value problem:

Find u such that

$-\epsilon \ \Delta u - \frac{\partial u}{\partial x} - \frac{\partial u}{\partial y} + 2u = f$ **in** $\Omega = [0,1]^2$ ----------(80a)

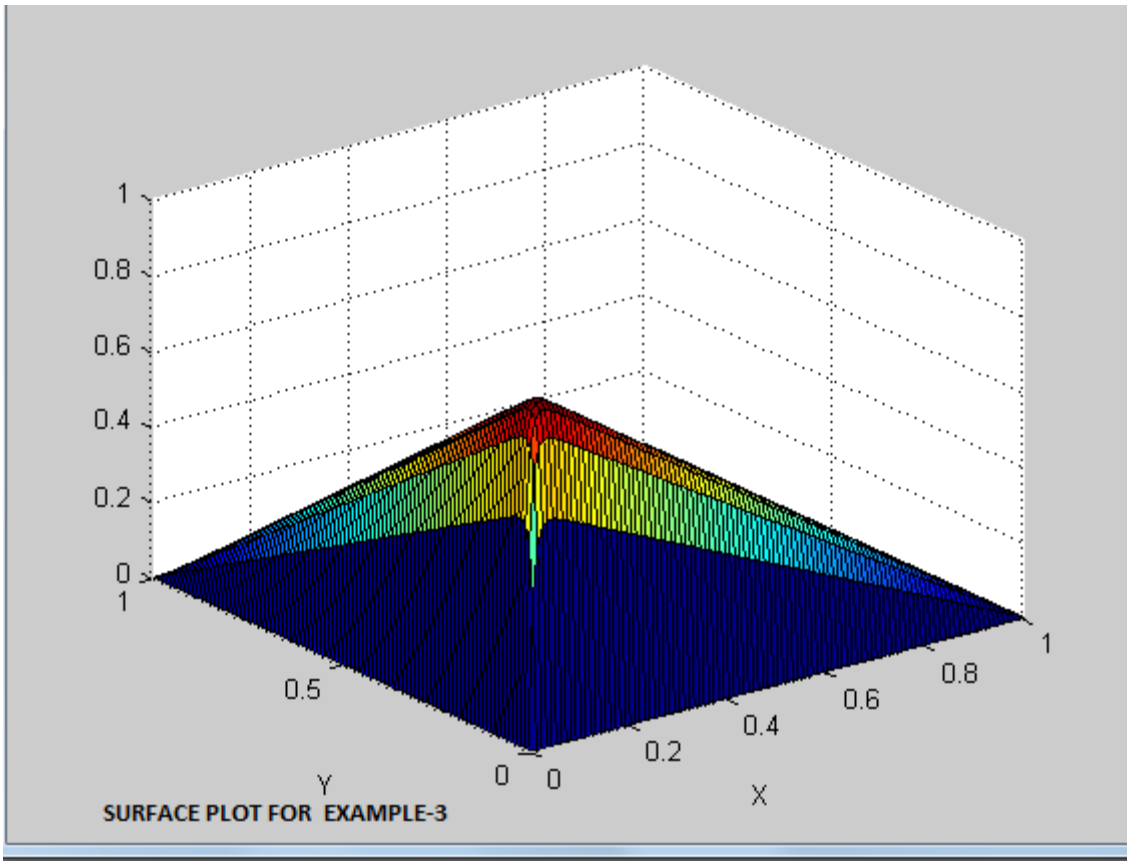
u = 0, on $\partial\Omega$ ----------(80b)

Where $\epsilon = 10^{-2}$

and f **is chosen appropriately such that the exact solution is**

u(x,y)=(1 - $e^{-\frac{x}{\epsilon}}$**)**$(1-x)$ **(1 -** $e^{-\frac{y}{\epsilon}}$**)**$(1-y)$ ----------(81)

DOI: 10.18535/ijecs/v5i12.75



Three dimensional surface plot of the exact solution: u(x,y)=(1 - $e^{-\frac{x}{\epsilon}}$)$(1-x)$ (1 - $e^{-\frac{y}{\epsilon}}$)$(1-y)$

Example 4

Consider the boundary value problem:

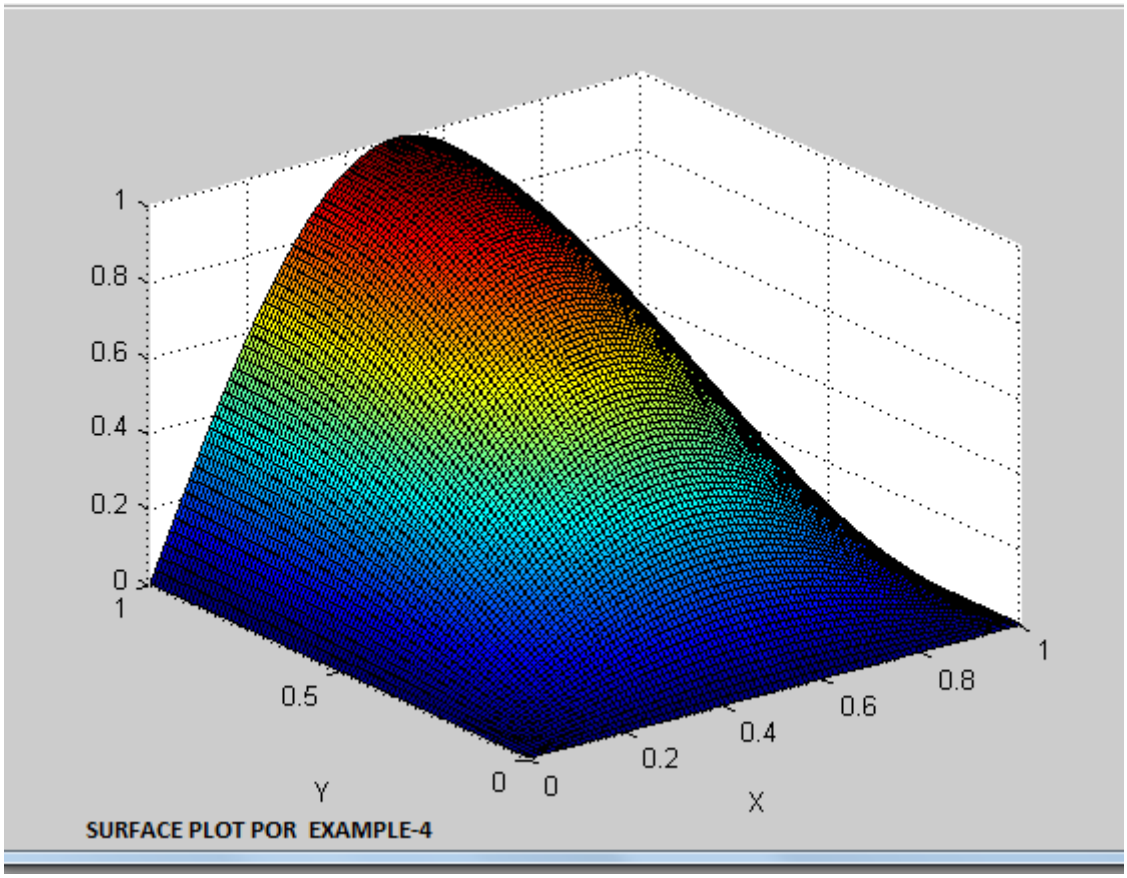
Find u such that

$$-\Delta u + \frac{1}{10}\frac{\partial u}{\partial y} = f \quad \text{in } \Omega = [0,1]^2 \qquad \text{----------(82a)}$$

$$\left.\begin{array}{l} u(x,0) = 0, 0 \leqslant x \leqslant 1 \\ u(0,y) = 0, 0 \leqslant y \leqslant 1 \\ u(1,y) = 0, 0 \leqslant y \leqslant 1 \\ u(x,1) = \sin(\pi x), 0 \leqslant x \leqslant 1 \end{array}\right\} \quad \text{----------(82b)}$$

and f is chosen appropriately such that the exact solution is u(x,y)= $\sin(\pi x)\sin(\frac{\pi y}{2})$

DOI: 10.18535/ijecs/v5i12.75



Three dimensional surface plot of the exact solution: u(x,y)= $\sin(\pi x)\sin(\frac{\pi y}{2})$

Example 5

Consider the boundary value problem:

Find u such that

$$-\epsilon\ \Delta u + 2\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = f \quad \text{in } \Omega = [-1,1]^2 \qquad \text{----------(83a)}$$

$$\mathbf{u}(1,y) = 0, -1 \leqslant y \leqslant 1$$

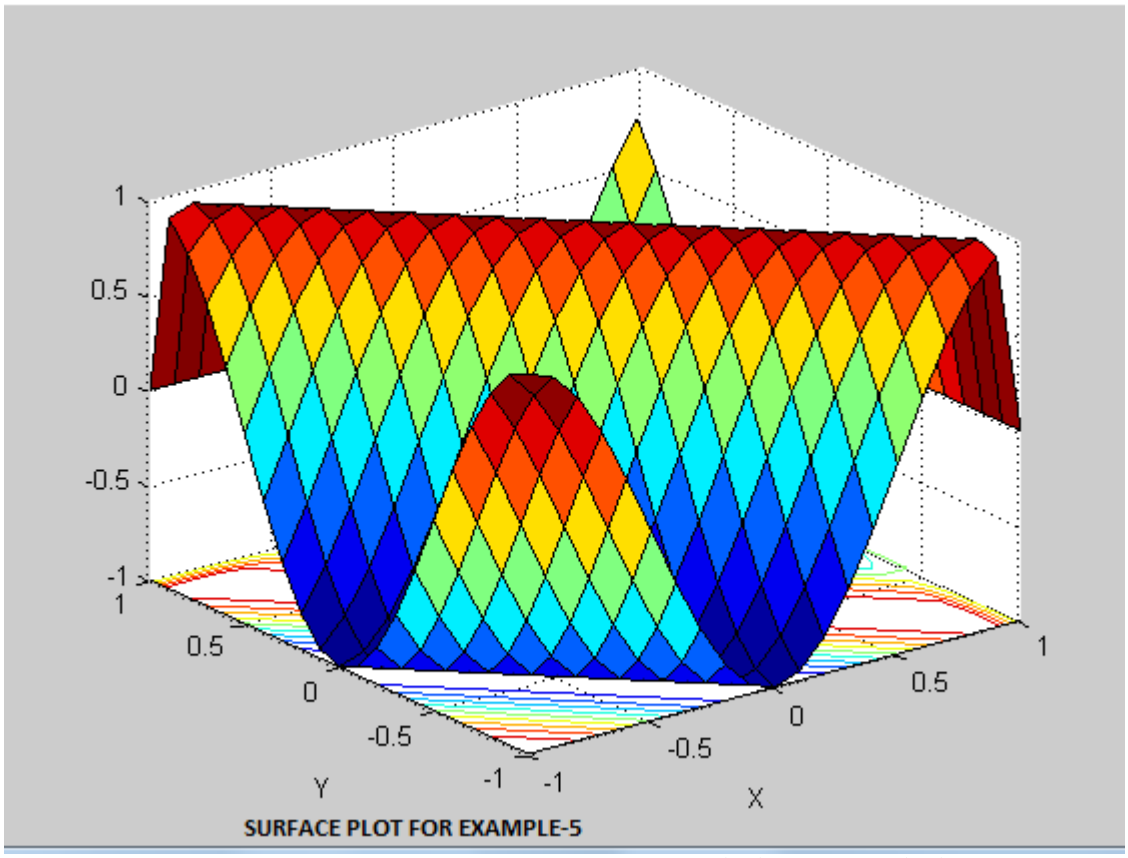
$$\mathbf{u}(x,1) = 0, -1 \leqslant x \leqslant 1$$

$$\left.\begin{aligned} \mathbf{u}(-1,y) &= (1 - e^{\frac{-2}{\epsilon}})(1 - e^{\frac{-(1-y)}{\epsilon}})\cos(\pi(-1+y)),, -1 \leqslant y \leqslant 1 \\ \mathbf{u}(x,-1) &= (1 - e^{\frac{-(1-x)}{\epsilon}})(1 - e^{\frac{-2}{\epsilon}})\cos(\pi(x-1)),, -1 \leqslant x \leqslant 1 \end{aligned}\right\} \qquad \text{----------(83b)}$$

and f is chosen appropriately such that the exact solution is

$$\mathbf{u(x,y)} = (1 - e^{\frac{-(1-x)}{\epsilon}})(1 - e^{\frac{-(1-y)}{\epsilon}})\cos(\pi(x+y)) \qquad \text{----------(84)}$$

DOI: 10.18535/ijecs/v5i12.75



Three dimensional surface plot of the exact solution: u(x,y)= $(1 - e^{\frac{-(1-x)}{\epsilon}})\ (1 - e^{\frac{-(1-y)}{\epsilon}})\ cos(\pi(x+y))$

Where $\epsilon = 0.001$

We have written the following codes to solve the elliptic equations with constant coefficients which are subject to Dirichlet boundary conditions over linear convex polygonal domains.

(1) D2LaplaceEquationQ4MoinExautomeshgenNINJ.m
(2) D2LaplaceEquationQ4MoinExautomeshgenNINJNiDNj.m
(3) polygonal_domain_coordinates.m
(4)nodaladdresses_special_convex_quadrilaterals_trial.m
(5) quadrilateral_mesh4MOINEX_q4.m

Conclusions:

This paper proposes the explicit integration scheme for a unique linear convex 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 quadrilaterals is derived by using the standard transformations in two steps. We first map an arbitrary triangle into a standard right isosceles triangle by using a affine linear transformation from global (x, y) space into a local space (u, v). We discretise the standard right isosceles triangle in (u, v) space into three unique linear convex quadrilaterals. We have shown that any unique linear convex quadrilateral in (x, y) space can be mapped into one of the unique quadrilaterals in (u, v) space. We can always map these linear convex quadrilaterals into a 2-aquare in (ξ, η) space by an approximate transformation. Using these two mappings, we have established an integral derivative product relation between the linear convex quadrilaterals in the (x, y) space interior to the arbitrary triangle and the linear convex quadrilaterals of the local space (u, v) interior to the standard right isosceles triangle. Further , we have shown that the product of global derivative integrals $S^{i,j,e}$ in (x, y) space can be expressed as a matrix triple product P ($K^{i,j,e}$) P^T X (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.We have shown that the explicit integration of the product of local derivative integrals in (u, v) space over the unique quadrilateral spanning vertices (1/3 ,1/3), (0, ½), (0, 0) , (1/2, 0) is now possible by use of symbolic processing capabilities in MATLAB which are based on Maple – V software package. In a similar manner we have found explicit integrals of ihe shape function and global derivative products as well as the prodct of shape functions.We use these integral values in computing stiffness matrix for elliptic equations with constant coefficients.

The proposed explicit integration scheme is a useful technique for boundary value problems governed by either a single equation or a system of second order partial differential equations. The physical applications of such problems are numerous in science, and engineering, the examples of single equations are the well known Laplace and Poisson equations with suitable boundary conditions These problems are already studied in authors recent paper. We have demonstrated the proposed explicit integration scheme to solve the Elliptic Equations with constant coefficients subject to Dirichlet boundary conditions over a simple polygonal domains for square domains. In section 5 of this paper we have studied some boundary value problems on the reaction diffusion and the convection diffusion equations. Monotonic convergence from below is observed with known analytical solutions for the governing unknown function of elliptic boundary value problems.We have shown the solutions in Tables which list both the FEM and exact solutions. 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.

REFERENCES:

[1] Zienkiewicz O.C, Taylor R.L and J.Z Zhu, 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 J.D Faires, Numerical Analysis, 9th Edition, Brooks/Cole, Cengage Learning (2011)
[5] Stroud A.H and D.Secrest, Gaussian quadrature formulas, Prentice Hall,Englewood Cliffs, N J, (1966)
[6] Stoer J and R. Bulirsch, 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 G.F Pinder, 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 Md. Shafiqul Islam, 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 Md. Sajedul Karim, 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 S. Yoshimura, 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 Md. Shafiqul Islam , 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 B. Littlefield, Mastering MATLAB 7 , Prentice Hall, Happer Saddle River, N J . (2005)
[18] Hunt B.H, Lipsman R.L and J.M Rosenberg , 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 S.Watt , 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 J.N Goodier , 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,Rathod .Bharath,Shivaram.K.T,Sugantha Devi.K, A new approach to automatic generation of all quadrilateral mesh for finite analysis, International Journal of Engineering and Computer Science, Vol. 2,issue 12,pp3488-3530(2013)
[27] Rathod H.T, Venkatesh.B, Shivaram. K.T,Mamatha.T.M, Numerical Integration over polygonal domains using convex quadrangulation and Gauss Legendre Quadrature Rules, International Journal of Engineering and Computer Science, Vol. 2,issue 8,pp2576-2610(2013)
[28] H.T. Rathod, Bharath Rathod, Shivaram K.T , H. Y. Shrivalli , Tara Rathod ,K. Sugantha Devi , An explicit finite element integration scheme usingautomatic mesh generation technique for linearconvex quadrilaterals over plane regions International Journal Of Engineering And Computer Science ISSN:2319-7242,Volume 3 Issue 4 April, 2014 Page No. 5400-5435
[29] 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 Issues 8 Aug 2016, Page No. 17397-17481*
[30] William F. Mitchell,A Collection of 2D Elliptic Problems for Testing Adaptive Algorithms, Technical Report NISTIR 7668,(2010)

TABLES

TABLE-1a

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 1: (Output for 1/10th of the FEM MODEL : $\lambda = 1$)

Let $\Omega = [-1, 1]^2$. As first example, we consider the boundary value problem: Find u such that

$\frac{-1}{\lambda}\Delta u + \lambda u = f_\lambda(x, y)$ **in** Ω **, u = 0, on** $\partial\Omega$ **where** $f_\lambda(x, y) = \frac{\lambda^2+2\pi^2}{\lambda} \sin(\pi x) \sin(\pi y)$ **and** λ **is a real parameter**

ALL QUADRILATERAL FEM MODEL : number of nodes=2481,elements=2400 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
238	0.0333333333333333	-0.0666666666666667	-0.0211746845623806	-0.0217326895365599
248	0.0666666666666667	-0.133333333333333	-0.0835210345578409	-0.0845653031794291
258	0.133333333333333	-0.266666666666667	-0.298141739739898	-0.302264231633827
268	0.166666666666667	-0.433333333333333	-0.481089072672213	-0.489073800366903
278	0.233333333333333	-0.666666666666667	-0.568587232571589	-0.579484103556457
288	0.266666666666667	-0.933333333333333	-0.151287975886428	-0.154508497187474
298	0.366666666666667	-0.633333333333333	-0.819711267378697	-0.834565303179429
308	0.466666666666667	-0.533333333333333	-0.972107259711699	-0.989073800366903
318	0.566666666666667	-0.633333333333333	-0.878578486528463	-0.893582297554377
328	0.666666666666667	-0.933333333333333	-0.177392356445941	-0.180056805991955
547	0.0666666666666667	-0.0333333333333333	-0.0211735978095821	-0.0217326895365599
557	0.133333333333333	-0.0666666666666667	-0.0835234212397429	-0.0845653031794291
567	0.266666666666667	-0.133333333333333	-0.298140540706632	-0.302264231633827

577	0.433333333333333	-0.266666666666667	-0.71487894690841	-0.726905328038456
587	0.666666666666667	-0.433333333333333	-0.832285700072775	-0.847100670886274
597	0.933333333333333	-0.666666666666667	-0.177371545739125	-0.180056805991955
607	0.633333333333333	-0.266666666666667	-0.666451450926116	-0.678896579685477
617	0.533333333333333	-0.0666666666666667	-0.20310870425417	-0.2067727288213
627	0.633333333333333	-0.0666666666666667	-0.186297984445455	-0.189936780737357
637	0.933333333333333	-0.266666666666667	-0.151315651745599	-0.154508497187474
856	0.0666666666666667	0.0333333333333333	0.0211746845623794	0.0217326895365599
866	0.133333333333333	0.0666666666666667	0.0835210345578401	0.0845653031794291
876	0.266666666666667	0.133333333333333	0.298141739739898	0.302264231633827
886	0.433333333333333	0.166666666666667	0.481089072672212	0.489073800366903
896	0.666666666666667	0.233333333333333	0.568587232571588	0.579484103556457
906	0.933333333333333	0.266666666666667	0.151287975886428	0.154508497187474
916	0.633333333333333	0.366666666666667	0.819711267378697	0.834565303179429
926	0.533333333333333	0.466666666666667	0.972107259711702	0.989073800366903
936	0.633333333333333	0.566666666666667	0.878578486528463	0.893582297554377
946	0.933333333333333	0.666666666666667	0.177392356445941	0.180056805991955
1165	0.0333333333333333	0.0666666666666667	0.0211735978095807	0.0217326895365599
1175	0.0666666666666667	0.133333333333333	0.083523421239742	0.0845653031794291
1185	0.133333333333333	0.266666666666667	0.298140540706632	0.302264231633827
1195	0.266666666666667	0.433333333333333	0.714878946908411	0.726905328038456
1205	0.433333333333333	0.666666666666667	0.832285700072774	0.847100670886274
1215	0.666666666666667	0.933333333333333	0.177371545739125	0.180056805991955
1225	0.266666666666667	0.633333333333333	0.666451450926115	0.678896579685477
1235	0.0666666666666667	0.533333333333333	0.203108704254168	0.2067727288213
1245	0.0666666666666667	0.633333333333333	0.186297984445454	0.189936780737357
1255	0.266666666666667	0.933333333333333	0.151315651745599	0.154508497187474
1474	-0.0333333333333333	0.0666666666666667	-0.0211746845623811	-0.0217326895365599
1484	-0.0666666666666667	0.133333333333333	-0.0835210345578419	-0.0845653031794291
1494	-0.133333333333333	0.266666666666667	-0.2981417397399	-0.302264231633827
1504	-0.166666666666667	0.433333333333333	-0.481089072672215	-0.489073800366903
1514	-0.233333333333333	0.666666666666667	-0.568587232571591	-0.579484103556457
1524	-0.266666666666667	0.933333333333333	-0.151287975886428	-0.154508497187474
1534	-0.366666666666667	0.633333333333333	-0.819711267378698	-0.834565303179429
1544	-0.466666666666667	0.533333333333333	-0.972107259711702	-0.989073800366903
1554	-0.566666666666667	0.633333333333333	-0.878578486528464	-0.893582297554377
1564	-0.666666666666667	0.933333333333333	-0.177392356445941	-0.180056805991955
1783	-0.0666666666666667	0.0333333333333333	-0.0211735978095827	-0.0217326895365599
1793	-0.133333333333333	0.0666666666666667	-0.0835234212397441	-0.0845653031794291
1803	-0.266666666666667	0.133333333333333	-0.298140540706634	-0.302264231633827
1813	-0.433333333333333	0.266666666666667	-0.714878946908412	-0.726905328038456
1823	-0.666666666666667	0.433333333333333	-0.832285700072777	-0.847100670886274
1833	-0.933333333333333	0.666666666666667	-0.177371545739125	-0.180056805991955
1843	-0.633333333333333	0.266666666666667	-0.666451450926117	-0.678896579685477
1853	-0.533333333333333	0.0666666666666667	-0.20310870425417	-0.2067727288213
1863	-0.633333333333333	0.0666666666666667	-0.186297984445455	-0.189936780737357
1873	-0.933333333333333	0.266666666666667	-0.151315651745599	-0.154508497187474
2092	-0.0666666666666667	-0.0333333333333333	0.021174684562379	0.0217326895365599
2102	-0.133333333333333	-0.0666666666666667	0.0835210345578395	0.0845653031794291
2112	-0.266666666666667	-0.133333333333333	0.298141739739897	0.302264231633827
2122	-0.433333333333333	-0.166666666666667	0.481089072672212	0.489073800366903
2132	-0.666666666666667	-0.233333333333333	0.568587232571589	0.579484103556457
2142	-0.933333333333333	-0.266666666666667	0.151287975886428	0.154508497187474
2152	-0.633333333333333	-0.366666666666667	0.819711267378696	0.834565303179429
2162	-0.533333333333333	-0.466666666666667	0.972107259711699	0.989073800366903
2172	-0.633333333333333	-0.566666666666667	0.878578486528461	0.893582297554377
2182	-0.933333333333333	-0.666666666666667	0.177392356445941	0.180056805991955
2382	-0.0333333333333333	-0.0666666666666667	0.0211735978095808	0.0217326895365599
2392	-0.0666666666666667	-0.133333333333333	0.0835234212397419	0.0845653031794291
2402	-0.133333333333333	-0.266666666666667	0.298140540706631	0.302264231633827
2412	-0.266666666666667	-0.433333333333333	0.714878946908409	0.726905328038456
2422	-0.433333333333333	-0.666666666666667	0.832285700072774	0.847100670886274
2432	-0.666666666666667	-0.933333333333333	0.177371545739125	0.180056805991955
2442	-0.266666666666667	-0.633333333333333	0.666451450926115	0.678896579685477
2452	-0.0666666666666667	-0.533333333333333	0.203108704254169	0.2067727288213
2462	-0.0666666666666667	-0.633333333333333	0.186297984445454	0.189936780737357
2472	-0.266666666666667	-0.933333333333333	0.151315651745599	0.154508497187474

TABLE-1b

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 1: (Output for 1/10th of the FEM MODEL : $\lambda = 1$)

Let $\Omega = [-1, 1]^2$. As first example, we consider the boundary value problem: Find u such that

DOI: 10.18535/ijecs/v5i12.75

$\frac{-1}{\lambda}\Delta u + \lambda u = f_\lambda(x, y)$ in Ω , u = 0, on $\partial\Omega$ where $f_\lambda(x, y) = \frac{\lambda^2+2\pi^2}{\lambda} \sin(\pi x) \sin(\pi y)$ and λ is a real parameter

ALL QUADRILATERAL FEM MODEL : number of nodes=5521,elements=5400 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
503	0.0222222222222222	-0.0444444444444444	-0.00950395458244156	-0.0097082247630093
513	0.0222222222222222	-0.711111111111111	-0.0545044045702902	-0.0549688514440557
523	0.0444444444444444	-0.422222222222222	-0.134053955619073	-0.135039065074129
533	0.0888888888888889	-0.177777777777778	-0.145309135086175	-0.14606554478973
543	0.0888888888888889	-0.844444444444444	-0.128067150149248	-0.129403900197577
553	0.111111111111111	-0.622222222222222	-0.314402212256475	-0.317115554828179
563	0.155555555555556	-0.444444444444444	-0.45895421347216	-0.462339234850303
573	0.177777777777778	-0.288888888888889	-0.414903411297698	-0.417582078759127
583	0.177777777777778	-0.955555555555556	-0.0730343386082784	-0.0737505072618114
593	0.222222222222222	-0.844444444444444	-0.298898593155464	-0.301770503658947
603	0.244444444444444	-0.755555555555556	-0.478251496939971	-0.48255025164875
613	0.288888888888889	-0.711111111111111	-0.615710056806464	-0.620960947799834
623	0.311111111111111	-0.688888888888889	-0.681591205146007	-0.687303296707956
633	0.355555555555556	-0.711111111111111	-0.70245673452397	-0.708259373761442
643	0.377777777777778	-0.755555555555556	-0.638771409586167	-0.644076025529257
653	0.422222222222222	-0.844444444444444	-0.451819928127124	-0.455526250979263
663	0.444444444444444	-0.955555555555556	-0.135950849082093	-0.137058748836223
673	0.511111111111111	-0.622222222222222	-0.919529614788381	-0.926619039214255
683	0.555555555555556	-0.844444444444444	-0.458954588975539	-0.462339234850303
693	0.622222222222222	-0.711111111111111	-0.725354549813088	-0.73063084796916
703	0.688888888888889	-0.711111111111111	-0.648739933393996	-0.653290522317386
713	0.755555555555556	-0.844444444444444	-0.324172148608135	-0.326122350781686
723	0.844444444444444	-0.955555555555556	-0.0649613590795543	-0.0653378132054804
1197	0.377777777777778	-0.355555555555556	-0.827204684117778	-0.833347328309341
1207	0.0888888888888889	-0.0444444444444444	-0.0381067590877037	-0.0383613055494847
1217	0.755555555555556	-0.711111111111111	-0.543733734642629	-0.547398266004612
1227	0.511111111111111	-0.422222222222222	-0.962256137077382	-0.969704648336062
1237	0.288888888888889	-0.177777777777778	-0.41490420194587	-0.417582078759127
1247	0.955555555555556	-0.844444444444444	-0.0649552466445859	-0.0653378132054804
1257	0.777777777777778	-0.622222222222222	-0.591772776839746	-0.595982293616937
1267	0.622222222222222	-0.444444444444444	-0.905987015686626	-0.913097848445116
1277	0.511111111111111	-0.288888888888889	-0.781464396895612	-0.787530718746963
1287	0.422222222222222	-0.177777777777778	-0.510456086177079	-0.514178397356799
1297	0.377777777777778	-0.0888888888888889	-0.253830920045226	-0.255566506029002
1307	0.355555555555556	-0.0444444444444444	-0.124227630248599	-0.1250879545479
1317	0.377777777777778	-0.0222222222222222	-0.0642052162522177	-0.064677076207065
1327	0.422222222222222	-0.0444444444444444	-0.134055644553932	-0.135039065074129
1337	0.511111111111111	-0.0888888888888889	-0.273340905826953	-0.275469444987308
1347	0.622222222222222	-0.177777777777778	-0.487206689866131	-0.491332586020939
1357	0.777777777777778	-0.288888888888889	-0.502101727995882	-0.506523548718153
1367	0.955555555555556	-0.444444444444444	-0.135952152637475	-0.137058748836223
1377	0.755555555555556	-0.177777777777778	-0.364702228505233	-0.368112852567069
1387	0.688888888888889	-0.0444444444444444	-0.114355158438811	-0.11537972978489
1397	0.755555555555556	-0.0444444444444444	-0.0957794571387613	-0.0966777595246446
1407	0.955555555555556	-0.177777777777778	-0.0730421335472235	-0.0737505072618114
1881	0.0444444444444444	0.0222222222222222	0.00950395458243355	0.0097082247630093
1891	0.711111111111111	0.0222222222222222	0.0545044045702877	0.0549688514440557
1901	0.422222222222222	0.0444444444444444	0.134053955619068	0.135039065074129
1911	0.177777777777778	0.0888888888888889	0.145309135086167	0.14606554478973
1921	0.844444444444444	0.0888888888888889	0.128067150149247	0.129403900197577
1931	0.622222222222222	0.111111111111111	0.314402212256472	0.317115554828179
1941	0.444444444444444	0.155555555555556	0.458954213472155	0.462339234850303
1951	0.288888888888889	0.177777777777778	0.41490341129769	0.417582078759127
1961	0.955555555555556	0.177777777777778	0.073034338608278	0.0737505072618114
1971	0.844444444444444	0.222222222222222	0.298898593155463	0.301770503658947
1981	0.755555555555556	0.244444444444444	0.478251496939969	0.48255025164875
1991	0.711111111111111	0.288888888888889	0.615710056806463	0.620960947799834
2001	0.688888888888889	0.311111111111111	0.681591205146007	0.687303296707956
2011	0.711111111111111	0.355555555555556	0.702456734523971	0.708259373761442
2021	0.755555555555556	0.377777777777778	0.638771409586167	0.644076025529257
2031	0.844444444444444	0.422222222222222	0.451819928127124	0.455526250979263
2041	0.955555555555556	0.444444444444444	0.135950849082093	0.137058748836223
2051	0.622222222222222	0.511111111111111	0.91952961478838	0.926619039214255
2061	0.844444444444444	0.555555555555556	0.458954588975539	0.462339234850303
2071	0.711111111111111	0.622222222222222	0.725354549813088	0.73063084796916

2081	0.711111111111111	0.688888888888889	0.648739933393997	0.653290522317386
2091	0.844444444444444	0.755555555555556	0.324172148608136	0.326122350781686
2101	0.955555555555556	0.844444444444444	0.0649613590795543	0.0653378132054804
2575	0.355555555555556	0.377777777777778	0.827204684117773	0.833347328309341
2585	0.0444444444444444	0.0888888888888889	0.038106759087696	0.0383613055494847
2595	0.711111111111111	0.755555555555556	0.54373373464263	0.547398266004612
2605	0.422222222222222	0.511111111111111	0.962256137077378	0.969704648336062
2615	0.177777777777778	0.288888888888889	0.414904201945863	0.417582078759127
2625	0.844444444444444	0.955555555555556	0.0649552466445862	0.0653378132054804
2635	0.622222222222222	0.777777777777778	0.591772776839746	0.595982293616937
2645	0.444444444444444	0.622222222222222	0.905987015686624	0.913097848445116
2655	0.288888888888889	0.511111111111111	0.781464396895606	0.787530718746963
2665	0.177777777777778	0.422222222222222	0.510456086177073	0.514178397356799
2675	0.0888888888888889	0.377777777777778	0.25383092004522	0.255566506029002
2685	0.0444444444444444	0.355555555555556	0.124227630248593	0.1250879545479
2695	0.0222222222222222	0.377777777777778	0.0642052162522118	0.064677076207065
2705	0.0444444444444444	0.422222222222222	0.134055644553927	0.135039065074129
2715	0.0888888888888889	0.511111111111111	0.273340905826948	0.275469444987308
2725	0.177777777777778	0.622222222222222	0.487206689866127	0.491332586020939
2735	0.288888888888889	0.777777777777778	0.502101727995881	0.506523548718153
2745	0.444444444444444	0.955555555555556	0.135952152637475	0.137058748836223
2755	0.177777777777778	0.755555555555556	0.364702228505232	0.368112852567069
2765	0.0444444444444444	0.688888888888889	0.114355158438808	0.11537972978489
2775	0.0444444444444444	0.755555555555556	0.0957794571387591	0.0966777595246446
2785	0.177777777777778	0.955555555555556	0.0730421335472234	0.0737505072618114
3259	-0.0222222222222222	0.0444444444444444	-0.00950395458244144	-0.0097082247630093
3269	-0.0222222222222222	0.711111111111111	-0.0545044045702904	-0.0549688514440557
3279	-0.0444444444444444	0.422222222222222	-0.134053955619073	-0.135039065074129
3289	-0.0888888888888889	0.177777777777778	-0.145309135086175	-0.14606554478973
3299	-0.0888888888888889	0.844444444444444	-0.128067150149248	-0.129403900197577
3309	-0.111111111111111	0.622222222222222	-0.314402212256476	-0.317115554828179
3319	-0.155555555555556	0.444444444444444	-0.458954213472161	-0.462339234850303
3329	-0.177777777777778	0.288888888888889	-0.414903411297697	-0.417582078759127
3339	-0.177777777777778	0.955555555555556	-0.0730343386082785	-0.0737505072618114
3349	-0.222222222222222	0.844444444444444	-0.298898593155465	-0.301770503658947
3359	-0.244444444444444	0.755555555555556	-0.478251496939972	-0.48255025164875
3369	-0.288888888888889	0.711111111111111	-0.615710056806465	-0.620960947799834
3379	-0.311111111111111	0.688888888888889	-0.681591205146008	-0.687303296707956
3389	-0.355555555555556	0.711111111111111	-0.702456734523973	-0.708259373761442
3399	-0.377777777777778	0.755555555555556	-0.638771409586169	-0.644076025529257
3409	-0.422222222222222	0.844444444444444	-0.451819928127126	-0.455526250979263
3419	-0.444444444444444	0.955555555555556	-0.135950849082093	-0.137058748836223
3429	-0.511111111111111	0.622222222222222	-0.919529614788384	-0.926619039214255
3439	-0.555555555555556	0.844444444444444	-0.458954588975542	-0.462339234850303
3449	-0.622222222222222	0.711111111111111	-0.72535454981309	-0.73063084796916
3459	-0.688888888888889	0.711111111111111	-0.648739933393998	-0.653290522317386
3469	-0.755555555555556	0.844444444444444	-0.324172148608137	-0.326122350781686
3479	-0.844444444444444	0.955555555555556	-0.0649613590795545	-0.0653378132054804
3953	-0.377777777777778	0.355555555555556	-0.827204684117781	-0.833347328309341
3963	-0.0888888888888889	0.0444444444444444	-0.0381067590877039	-0.0383613055494847
3973	-0.755555555555556	0.711111111111111	-0.543733734642632	-0.547398266004612
3983	-0.511111111111111	0.422222222222222	-0.962256137077385	-0.969704648336062
3993	-0.288888888888889	0.177777777777778	-0.41490420194587	-0.417582078759127
4003	-0.955555555555556	0.844444444444444	-0.0649552466445863	-0.0653378132054804
4013	-0.777777777777778	0.622222222222222	-0.591772776839748	-0.595982293616937
4023	-0.622222222222222	0.444444444444444	-0.90598701568663	-0.913097848445116
4033	-0.511111111111111	0.288888888888889	-0.781464396895613	-0.787530718746963
4043	-0.422222222222222	0.177777777777778	-0.51045608617708	-0.514178397356799
4053	-0.377777777777778	0.0888888888888889	-0.253830920045229	-0.255566506029002
4063	-0.355555555555556	0.0444444444444444	-0.124227630248601	-0.1250879545479
4073	-0.377777777777778	0.0222222222222222	-0.0642052162522202	-0.064677076207065
4083	-0.422222222222222	0.0444444444444444	-0.134055644553935	-0.135039065074129
4093	-0.511111111111111	0.0888888888888889	-0.273340905826956	-0.275469444987308
4103	-0.622222222222222	0.177777777777778	-0.487206689866134	-0.491332586020939
4113	-0.777777777777778	0.288888888888889	-0.502101727995884	-0.506523548718153
4123	-0.955555555555556	0.444444444444444	-0.135952152637476	-0.137058748836223
4133	-0.755555555555556	0.177777777777778	-0.364702228505236	-0.368112852567069
4143	-0.688888888888889	0.0444444444444444	-0.114355158438814	-0.11537972978489
4153	-0.755555555555556	0.0444444444444444	-0.0957794571387635	-0.0966777595246446
4163	-0.955555555555556	0.177777777777778	-0.0730421335472241	-0.0737505072618114
4637	-0.0444444444444444	-0.0222222222222222	0.00950395458243316	0.0097082247630093
4647	-0.711111111111111	-0.0222222222222222	0.0545044045702849	0.0549688514440557
4657	-0.422222222222222	-0.0444444444444444	0.134053955619065	0.135039065074129
4667	-0.177777777777778	-0.0888888888888889	0.145309135086166	0.14606554478973

4677	-0.844444444444444	-0.0888888888888889	0.128067150149246	0.129403900197577
4687	-0.622222222222222	-0.111111111111111	0.314402212256468	0.317115554828179
4697	-0.444444444444444	-0.155555555555556	0.458954213472151	0.462339234850303
4707	-0.288888888888889	-0.177777777777778	0.414903411297688	0.417582078759127
4717	-0.955555555555556	-0.177777777777778	0.0730343386082776	0.0737505072618114
4727	-0.844444444444444	-0.222222222222222	0.298898593155462	0.301770503658947
4737	-0.755555555555556	-0.244444444444444	0.478251496939966	0.48255025164875
4747	-0.711111111111111	-0.288888888888889	0.61571005680646	0.620960947799834
4757	-0.688888888888889	-0.311111111111111	0.681591205146002	0.687303296707956
4767	-0.711111111111111	-0.355555555555556	0.702456734523966	0.708259373761442
4777	-0.755555555555556	-0.377777777777778	0.638771409586163	0.644076025529257
4787	-0.844444444444444	-0.422222222222222	0.451819928127122	0.455526250979263
4797	-0.955555555555556	-0.444444444444444	0.135950849082092	0.137058748836223
4807	-0.622222222222222	-0.511111111111111	0.919529614788376	0.926619039214255
4817	-0.844444444444444	-0.555555555555556	0.458954588975538	0.462339234850303
4827	-0.711111111111111	-0.622222222222222	0.725354549813085	0.73063084796916
4837	-0.711111111111111	-0.688888888888889	0.648739933393995	0.653290522317386
4847	-0.844444444444444	-0.755555555555556	0.324172148608135	0.326122350781686
4857	-0.955555555555556	-0.844444444444444	0.0649613590795542	0.0653378132054804
5302	-0.355555555555556	-0.377777777777778	0.827204684117769	0.833347328309341
5312	-0.0444444444444444	-0.0888888888888889	0.0381067590876951	0.0383613055494847
5322	-0.711111111111111	-0.755555555555556	0.543733734642629	0.547398266004612
5332	-0.422222222222222	-0.511111111111111	0.962256137077373	0.969704648336062
5342	-0.177777777777778	-0.288888888888889	0.414904201945861	0.417582078759127
5352	-0.844444444444444	-0.955555555555556	0.0649552466445858	0.0653378132054804
5362	-0.622222222222222	-0.777777777777778	0.591772776839745	0.595982293616937
5372	-0.444444444444444	-0.622222222222222	0.90598701568662	0.913097848445116
5382	-0.288888888888889	-0.511111111111111	0.781464396895602	0.787530718746963
5392	-0.177777777777778	-0.422222222222222	0.510456086177071	0.514178397356799
5402	-0.0888888888888889	-0.377777777777778	0.253830920045219	0.255566506029002
5412	-0.0444444444444444	-0.355555555555556	0.124227630248592	0.1250879545479
5422	-0.0222222222222222	-0.377777777777778	0.0642052162522111	0.064677076207065
5432	-0.0444444444444444	-0.422222222222222	0.134055644553926	0.135039065074129
5442	-0.0888888888888889	-0.511111111111111	0.273340905826947	0.275469444987308
5452	-0.177777777777778	-0.622222222222222	0.487206689866127	0.491332586020939
5462	-0.288888888888889	-0.777777777777778	0.502101727995879	0.506523548718153
5472	-0.444444444444444	-0.955555555555556	0.135952152637475	0.137058748836223
5482	-0.177777777777778	-0.755555555555556	0.364702228505231	0.368112852567069
5492	-0.0444444444444444	-0.688888888888889	0.114355158438808	0.11537972978489
5502	-0.0444444444444444	-0.755555555555556	0.0957794571387589	0.0966777595246446
5512	-0.177777777777778	-0.955555555555556	0.0730421335472232	0.0737505072618114

TABLE-1c

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 1: (Output for 1/10th of the FEM MODEL : $\lambda = 1$)

Let $\Omega = [-1, 1]^2$. As first example, we consider the boundary value problem: Find u such that

$\frac{-1}{\lambda}\Delta u + \lambda u = f_\lambda(x, y)$ in Ω , u = 0, on $\partial\Omega$ where f_λ(x, y) = $\frac{\lambda^2+2\pi^2}{\lambda}$ sin (π x) sin (π y) and λ is a real parameter

ALL QUADRILATERAL FEM MODEL : number of nodes=7921,elements=7776 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
710	0.0185185185185185	-0.037037037037037	-0.00661285797786529	-0.00675020262953007
720	0.0185185185185185	-0.592592592592593	-0.0553931910632839	-0.0557021362916143
730	0.037037037037037	-0.185185185185185	-0.0635215104026984	-0.0637940986022171
740	0.037037037037037	-0.740740740740741	-0.0839090498629578	-0.0844429257080959
750	0.0740740740740741	-0.37037037037037	-0.210763261110288	-0.211755207017928
760	0.0740740740740741	-0.925925925925926	-0.0528096307676033	-0.0531836798382939
770	0.0925925925925926	-0.574074074074074	-0.277473559342118	-0.279072414455238
780	0.12962962962963	-0.259259259259259	-0.286903766469633	-0.288097981777303
790	0.12962962962963	-0.814814814814815	-0.216135668705172	-0.217649387470701
800	0.148148148148148	-0.518518518518518	-0.445605515750134	-0.448039883019578
810	0.185185185185185	-0.259259259259259	-0.39796673668236	-0.399698346456447
820	0.185185185185185	-0.814814814814815	-0.29993772857594	-0.301960116980422
830	0.203703703703704	-0.574074074074074	-0.577796748009873	-0.581062104579067
840	0.240740740740741	-0.37037037037037	-0.627017934452964	-0.630118125102971
850	0.240740740740741	-0.925925925925926	-0.157197026072192	-0.158258212856816
860	0.259259259259259	-0.740740740740741	-0.525848049142376	-0.529072414455238
870	0.296296296296296	-0.592592592592593	-0.764153561985769	-0.768425606244347

880	0.314814814814815	-0.462962962962963	-0.825473746172326	-0.829838541721195
890	0.351851851851852	-0.37037037037037	-0.816355393204573	-0.820547883978904
900	0.351851851851852	-0.925925925925926	-0.204834342599097	-0.20608586947205
910	0.37037037037037	-0.851851851851852	-0.40964936682964	-0.412094636014727
920	0.407407407407407	-0.814814814814815	-0.523425568697003	-0.526423837915034
930	0.425925925925926	-0.796296296296296	-0.577796946889551	-0.581062104579067
940	0.462962962962963	-0.814814814814815	-0.542804486105908	-0.545793394943501
950	0.481481481481481	-0.851851851851852	-0.445605890347433	-0.448039883019579
960	0.518518518518518	-0.925925925925926	-0.229019958740593	-0.23022570518901
970	0.574074074074074	-0.592592592592593	-0.92726563973975	-0.932166781027853
980	0.592592592592593	-0.740740740740741	-0.693283080863914	-0.696816320161706
990	0.62962962962963	-0.925925925925926	-0.210763967653023	-0.211755207017928
1000	0.685185185185185	-0.814814814814815	-0.457003767499685	-0.459108053440137
1010	0.740740740740741	-0.814814814814815	-0.397966865765491	-0.399698346456447
1020	0.796296296296296	-0.925925925925926	-0.137200075512053	-0.137714248596867
1030	0.907407407407407	-0.925925925925926	-0.0659146508617871	-0.0661413772434148
1705	0.37037037037037	-0.351851851851852	-0.816355302623672	-0.820547883978904
1715	0.925925925925926	-0.907407407407407	-0.0659140817552305	-0.0661413772434148
1725	0.518518518518518	-0.481481481481481	-0.991295500637659	-0.996619178870971
1735	0.148148148148148	-0.0740740740740741	-0.103143837860206	-0.103500213730423
1745	0.703703703703704	-0.62962962962963	-0.732827312600074	-0.736522435289912
1755	0.351851851851852	-0.259259259259259	-0.646819717415639	-0.650004827820579
1765	0.907407407407407	-0.814814814814815	-0.157027304711389	-0.157600951314475
1775	0.592592592592593	-0.462962962962963	-0.946397864181577	-0.951511929946241
1785	0.296296296296296	-0.148148148148148	-0.358405403549867	-0.359992231328241
1795	0.851851851851852	-0.703703703703704	-0.358404668031708	-0.359992231328241
1805	0.592592592592593	-0.407407407407407	-0.912757050276902	-0.917743905706468
1815	0.351851851851852	-0.148148148148148	-0.399183691115845	-0.401061596377522
1825	0.907407407407407	-0.703703703703704	-0.229059384523691	-0.230051524714688
1835	0.703703703703704	-0.462962962962963	-0.792379015330166	-0.796699522678744
1845	0.518518518518518	-0.259259259259259	-0.722224359985892	-0.726143040493856
1855	0.37037037037037	-0.0740740740740741	-0.210763537654683	-0.211755207017928
1865	0.925925925925926	-0.62962962962963	-0.210762831111944	-0.211755207017928
1875	0.796296296296296	-0.481481481481481	-0.592921042952059	-0.596148293878673
1885	0.703703703703704	-0.351851851851852	-0.712718702909338	-0.716803466606335
1895	0.62962962962963	-0.259259259259259	-0.664060642494032	-0.667886193412533
1905	0.592592592592593	-0.185185185185185	-0.523426608229475	-0.526423837915034
1915	0.574074074074074	-0.148148148148148	-0.434222774684511	-0.43670174021449
1925	0.592592592592593	-0.12962962962963	-0.377256040009581	-0.379440261905885
1935	0.62962962962963	-0.148148148148148	-0.40965073612006	-0.412094636014727
1945	0.703703703703704	-0.185185185185185	-0.438057925555281	-0.440773895937716
1955	0.796296296296296	-0.259259259259259	-0.431625707842849	-0.434357419443489
1965	0.925925925925926	-0.351851851851852	-0.204836365747667	-0.20608586947205
1975	0.703703703703704	-0.0740740740740741	-0.183804576912817	-0.184982338539911
1985	0.907407407407407	-0.259259259259259	-0.207230296720318	-0.208613111791988
1995	0.851851851851852	-0.148148148148148	-0.19999014552998	-0.201420704148607
2005	0.907407407407407	-0.148148148148148	-0.127778053816246	-0.12871705571958
2015	0.925925925925926	-0.0740740740740741	-0.0528137303083621	-0.0531836798382939
2690	0.148148148148148	0.0185185185185185	0.0259507947194373	0.0260953515479177
2700	0.703703703703704	0.0185185185185185	0.0463733855460795	0.0466393158078666
2710	0.296296296296296	0.037037037037037	0.0926902637171081	0.0931208189343668
2720	0.851851851851852	0.037037037037037	0.0517612810293085	0.052102404686486
2730	0.481481481481481	0.0740740740740741	0.229019682262423	0.23022570518901
2740	0.12962962962963	0.0925925925925926	0.113217975543809	0.113596957311482
2750	0.685185185185185	0.0925925925925926	0.238107249853917	0.239620605203944
2760	0.37037037037037	0.12962962962963	0.361968193337813	0.363686820786524
2770	0.925925925925926	0.12962962962963	0.0906751977898313	0.0913422801285822
2780	0.62962962962963	0.148148148148148	0.409649269957337	0.412094636014727
2790	0.37037037037037	0.185185185185185	0.502136110928806	0.504567994539933
2800	0.925925925925926	0.185185185185185	0.125833319735291	0.126725491458587
2810	0.685185185185185	0.203703703703704	0.495897901586782	0.498918724848192
2820	0.481481481481481	0.240740740740741	0.681466384595477	0.685080625629794
2830	0.296296296296296	0.259259259259259	0.580688672382597	0.583443267704437
2840	0.851851851851852	0.259259259259259	0.324324913194941	0.326444694037409
2850	0.703703703703704	0.296296296296296	0.63963923763844	0.643401616355545
2860	0.574074074074074	0.314814814814815	0.808491159792875	0.812967129327321
2870	0.481481481481481	0.351851851851852	0.887389409067044	0.892120755332356
2880	0.407407407407407	0.37037037037037	0.875069172899317	0.879641400430461
2890	0.962962962962963	0.37037037037037	0.105970440372521	0.106598383644455
2900	0.925925925925926	0.407407407407407	0.21964955316535	0.220927585544762
2910	0.907407407407407	0.425925925925926	0.277473797185519	0.279072414455238
2920	0.925925925925926	0.462962962962963	0.227794742668004	0.22905652872545
2930	0.962962962962963	0.481481481481481	0.11526328109495	0.115896503288703
2940	0.574074074074074	0.537037037037037	0.961336996221784	0.966465489263926

2950	0.703703703703704	0.574074074074074	0.776478128423968	0.780501858283407
2960	0.851851851851852	0.592592592592593	0.427820832322802	0.429944907767832
2970	0.685185185185185	0.648148148148148	0.742880677996847	0.746619178870972
2980	0.925925925925926	0.685185185185185	0.191832259519508	0.19267674912369
2990	0.925925925925926	0.740740740740741	0.167061987386575	0.167743905706468
3000	0.851851851851852	0.814814814814815	0.245681934654687	0.246619178870972
3010	0.962962962962963	0.925925925925926	0.0266221304889076	0.0267728684780174
3685	0.462962962962963	0.481481481481481	0.986273799679888	0.991557955641738
3695	0.037037037037037	0.0740740740740741	0.0266211366699706	0.0267728684780173
3705	0.592592592592593	0.62962962962963	0.875069237337254	0.879641400430461
3715	0.185185185185185	0.259259259259259	0.397966675650215	0.399698346456447
3725	0.740740740740741	0.814814814814815	0.397966546567078	0.399698346456447
3735	0.37037037037037	0.462962962962963	0.907189193494822	0.912007458049964
3745	0.0185185185185185	0.148148148148148	0.025950764349749	0.0260953515479177
3755	0.574074074074074	0.703703703703704	0.776478337341408	0.780501858283407
3765	0.259259259259259	0.407407407407407	0.693283230201495	0.696816320161706
3775	0.814814814814815	0.962962962962963	0.0635199987102746	0.0637940986022172
3785	0.518518518518518	0.703703703703704	0.796530502141585	0.800766127265957
3795	0.259259259259259	0.462962962962963	0.718673614854413	0.722455401220792
3805	0.0185185185185185	0.259259259259259	0.0420787897600189	0.0422930159432547
3815	0.574074074074074	0.814814814814815	0.532000198875226	0.534696892449359
3825	0.37037037037037	0.62962962962963	0.838449754464152	0.843120818934367
3835	0.185185185185185	0.481481481481481	0.545693839041997	0.548579295851393
3845	0.037037037037037	0.351851851851852	0.1032430999904	0.103744417372569
3855	0.592592592592593	0.907407407407407	0.273406513508705	0.274754489035403
3865	0.462962962962963	0.814814814814815	0.542805182956557	0.545793394943501
3875	0.37037037037037	0.740740740740741	0.664060557686927	0.667886193412533
3885	0.296296296296296	0.703703703703704	0.639640480740417	0.643401616355545
3895	0.259259259259259	0.685185185185185	0.604123765152262	0.607711811877324
3905	0.240740740740741	0.703703703703704	0.547147075170999	0.550450333568719
3915	0.259259259259259	0.740740740740741	0.525849636789819	0.529072414455238
3925	0.296296296296296	0.814814814814815	0.438058041220996	0.440773895937716
3935	0.37037037037037	0.907407407407407	0.2617651424206	0.263347347780654
3945	0.037037037037037	0.62962962962963	0.105972109989218	0.106598383644455
3955	0.185185185185185	0.814814814814815	0.299940161450923	0.301960116980422
3965	0.0185185185185185	0.703703703703704	0.0463752030218858	0.0466393158078666
3975	0.259259259259259	0.962962962962963	0.0839117569373681	0.084442925708096
3985	0.0185185185185185	0.814814814814815	0.0317754123028054	0.0319511055146974
3995	0.037037037037037	0.907407407407407	0.0331035270873516	0.033295823065967
4670	-0.0185185185185185	0.259259259259259	-0.0420787208228456	-0.0422930159432547
4680	-0.0185185185185185	0.814814814814815	-0.0317726074593422	-0.0319511055146974
4690	-0.037037037037037	0.407407407407407	-0.110654107179343	-0.111215794186113
4700	-0.037037037037037	0.962962962962963	-0.0134829871180013	-0.0134775647100881
4710	-0.0740740740740741	0.592592592592593	-0.21964968889235	-0.220927585544762
4720	-0.0925925925925926	0.240740740740741	-0.196033029451238	-0.196816320161706
4730	-0.0925925925925926	0.796296296296296	-0.170070488395585	-0.171267014541561
4740	-0.12962962962963	0.481481481481481	-0.393335139629977	-0.395409661763065
4750	-0.148148148148148	0.185185185185185	-0.245681520193092	-0.246619178870971
4760	-0.148148148148148	0.740740740740741	-0.324324979941891	-0.326444694037409
4770	-0.185185185185185	0.481481481481481	-0.545693317415214	-0.548579295851393
4780	-0.203703703703704	0.240740740740741	-0.408024486229487	-0.409795090037506
4790	-0.203703703703704	0.796296296296296	-0.354263491278497	-0.356598383644455
4800	-0.240740740740741	0.592592592592593	-0.653713332887978	-0.65741229199245
4810	-0.259259259259259	0.407407407407407	-0.693282755561922	-0.696816320161706
4820	-0.259259259259259	0.962962962962963	-0.0839086842606447	-0.084442925708096
4830	-0.296296296296296	0.814814814814815	-0.438056200926837	-0.440773895937716
4840	-0.314814814814815	0.685185185185185	-0.694015240829378	-0.698039883019578
4850	-0.351851851851852	0.592592592592593	-0.851386032309851	-0.856090697292628
4860	-0.37037037037037	0.518518518518518	-0.91174189216956	-0.91666263055466
4870	-0.407407407407407	0.481481481481481	-0.951284218002977	-0.956368745682866
4880	-0.425925925925926	0.462962962962963	-0.961336926866615	-0.966465489263926
4890	-0.462962962962963	0.481481481481481	-0.986273784152498	-0.991557955641738
4900	-0.481481481481481	0.518518518518518	-0.991295412550683	-0.996619178870971
4910	-0.518518518518518	0.592592592592593	-0.951284178002149	-0.956368745682866
4920	-0.537037037037037	0.685185185185185	-0.825473944898939	-0.829838541721195
4930	-0.574074074074074	0.814814814814815	-0.532000202137057	-0.534696892449359
4940	-0.592592592592593	0.962962962962963	-0.11065516119308	-0.111215794186113
4950	-0.648148148148148	0.796296296296296	-0.531084702022003	-0.533640408995171
4960	-0.703703703703704	0.740740740740741	-0.580688930760885	-0.583443267704437
4970	-0.759259259259259	0.796296296296296	-0.408024809937127	-0.409795090037506
4980	-0.814814814814815	0.962962962962963	-0.0635232387081344	-0.0637940986022172
5655	-0.037037037037037	0.0185185185185185	-0.00661282509594777	-0.00675020262953007
5665	-0.592592592592593	0.574074074074074	-0.927265755919057	-0.932166781027853
5675	-0.185185185185185	0.148148148148148	-0.245681711370232	-0.246619178870971

5685	-0.740740740740741	0.703703703703704	-0.580688653367904	-0.583443267704437
5695	-0.37037037037037	0.296296296296296	-0.732827256201443	-0.736522435289912
5705	-0.925925925925926	0.851851851851852	-0.103143033829387	-0.103500213730423
5715	-0.574074074074074	0.481481481481481	-0.966197282015256	-0.971398632663848
5725	-0.259259259259259	0.12962962962963	-0.286903771417049	-0.288097981777303
5735	-0.814814814814815	0.685185185185185	-0.457003394023787	-0.459108053440137
5745	-0.518518518518518	0.37037037037037	-0.911742330024164	-0.91666263055466
5755	-0.259259259259259	0.0740740740740741	-0.167061052512915	-0.167743905706468
5765	-0.814814814814815	0.62962962962963	-0.502136073667055	-0.504567994539934
5775	-0.574074074074074	0.37037037037037	-0.88859362907809	-0.893465472883626
5785	-0.37037037037037	0.12962962962963	-0.361968428298398	-0.363686820786524
5795	-0.925925925925926	0.685185185185185	-0.191830651294623	-0.19267674912369
5805	-0.740740740740741	0.481481481481481	-0.722224132976992	-0.726143040493856
5815	-0.592592592592593	0.296296296296296	-0.764154315972569	-0.768425606244347
5825	-0.462962962962963	0.148148148148148	-0.443452028010206	-0.445764560698237
5835	-0.37037037037037	0.0185185185185185	-0.0531139806007522	-0.0533895184373967
5845	-0.925925925925926	0.574074074074074	-0.223285177178361	-0.224399590100231
5855	-0.851851851851852	0.481481481481481	-0.445606151935908	-0.448039883019579
5865	-0.814814814814815	0.407407407407407	-0.523426657683986	-0.526423837915034
5875	-0.796296296296296	0.37037037037037	-0.545124341085947	-0.548320637263439
5885	-0.814814814814815	0.351851851851852	-0.488151733548659	-0.491059158954835
5895	-0.851851851851852	0.37037037037037	-0.409650639247764	-0.412094636014727
5905	-0.925925925925926	0.407407407407407	-0.219650944784467	-0.220927585544762
5915	-0.592592592592593	0.0185185185185185	-0.0553943072497978	-0.0557021362916143
5925	-0.740740740740741	0.148148148148148	-0.324327059737392	-0.326444694037409
5935	-0.925925925925926	0.296296296296296	-0.183804973962501	-0.184982338539911
5945	-0.814814814814815	0.12962962962963	-0.216138306316093	-0.217649387470701
5955	-0.814814814814815	0.0740740740740741	-0.125836472349967	-0.126725491458587
5965	-0.925925925925926	0.12962962962963	-0.0906790631577021	-0.0913422801285822
5975	-0.925925925925926	0.0185185185185185	-0.0133830357919691	-0.0134091203483596
6650	-0.37037037037037	-0.0185185185185185	0.0531137009264257	0.0533895184373967
6660	-0.925925925925926	-0.0185185185185185	0.0133788173201248	0.0134091203483596
6670	-0.518518518518518	-0.037037037037037	0.115262792598426	0.115896503288703
6680	-0.148148148148148	-0.0740740740740741	0.103143896706241	0.103500213730423
6690	-0.703703703703704	-0.0740740740740741	0.183802767239071	0.184982338539911
6700	-0.351851851851852	-0.0925925925925926	0.255113404773942	0.256296730100902
6710	-0.907407407407407	-0.0925925925925926	0.081642934336429	0.0822560942935318
6720	-0.592592592592593	-0.12962962962963	0.377254968966207	0.379440261905885
6730	-0.296296296296296	-0.148148148148148	0.358405011678586	0.359992231328241
6740	-0.851851851851852	-0.148148148148148	0.199987208663681	0.201420704148607
6750	-0.592592592592593	-0.185185185185185	0.523425618151502	0.526423837915034
6760	-0.351851851851852	-0.203703703703704	0.531084208638269	0.533640408995171
6770	-0.907407407407407	-0.203703703703704	0.17007027692599	0.171267014541561
6780	-0.703703703703704	-0.240740740740741	0.5471456326669	0.550450333568719
6790	-0.518518518518518	-0.259259259259259	0.722223593590563	0.726143040493856
6800	-0.37037037037037	-0.296296296296296	0.732827245993748	0.736522435289912
6810	-0.925925925925926	-0.296296296296296	0.183802370189393	0.184982338539911
6820	-0.796296296296296	-0.314814814814815	0.495897932070261	0.498918724848192
6830	-0.703703703703704	-0.351851851851852	0.712717681445877	0.716803466606335
6840	-0.62962962962963	-0.37037037037037	0.838449065266442	0.843120818934367
6850	-0.592592592592593	-0.407407407407407	0.912756583401118	0.917743905706468
6860	-0.574074074074074	-0.425925925925926	0.9417003511692	0.946816320161706
6870	-0.592592592592593	-0.462962962962963	0.946397536664788	0.951511929946241
6880	-0.62962962962963	-0.481481481481481	0.91174198665541	0.91666263055466
6890	-0.703703703703704	-0.518518518518518	0.796530137511587	0.800766127265957
6900	-0.796296296296296	-0.537037037037037	0.590030368871199	0.593120818934367
6910	-0.925925925925926	-0.574074074074074	0.223285740461664	0.224399590100231
6920	-0.703703703703704	-0.62962962962963	0.732827189595107	0.736522435289912
6930	-0.907407407407407	-0.648148148148148	0.255114408231585	0.256296730100902
6940	-0.851851851851852	-0.703703703703704	0.358405747196742	0.359992231328241
6950	-0.907407407407407	-0.759259259259259	0.196034142253911	0.196816320161706
6960	-0.925925925925926	-0.851851851851852	0.10314470073706	0.103500213730423
7600	-0.12962962962963	-0.148148148148148	0.177136369359046	0.177760274292364
7610	-0.685185185185185	-0.703703703703704	0.66689857726473	0.670164150798469
7620	-0.259259259259259	-0.296296296296296	0.580688911746194	0.583443267704437
7630	-0.814814814814815	-0.851851851851852	0.245681296908639	0.246619178870972
7640	-0.407407407407407	-0.481481481481481	0.951284344245172	0.956368745682866
7650	-0.037037037037037	-0.12962962962963	0.0457815586824645	0.0459820542655251
7660	-0.592592592592593	-0.685185185185185	0.796278119755284	0.800388561001014
7670	-0.240740740740741	-0.37037037037037	0.627018034040673	0.630118125102971
7680	-0.796296296296296	-0.925925925925926	0.137198078526689	0.137714248596867
7690	-0.481481481481481	-0.62962962962963	0.9117422355383	0.91666263055466
7700	-0.185185185185185	-0.37037037037037	0.502136286145843	0.504567994539933
7710	-0.740740740740741	-0.925925925925926	0.167060063803539	0.167743905706468

7720	-0.481481481481481	-0.685185185185185	0.829585525236141	0.834074298269741
7730	-0.240740740740741	-0.481481481481481	0.681466825075917	0.685080625629794
7740	-0.037037037037037	-0.296296296296296	0.092690699467428	0.0931208189343668
7750	-0.592592592592593	-0.851851851851852	0.427820214988356	0.429944907767832
7760	-0.407407407407407	-0.703703703703704	0.764154334226295	0.768425606244347
7770	-0.259259259259259	-0.574074074074074	0.703832165671378	0.707767190927622
7780	-0.12962962962963	-0.481481481481481	0.393335723519551	0.395409661763065
7790	-0.037037037037037	-0.407407407407407	0.110654840566686	0.111215794186113
7800	-0.592592592592593	-0.962962962962963	0.110653786552957	0.111215794186113
7810	-0.518518518518518	-0.925925925925926	0.229020029548648	0.23022570518901
7820	-0.481481481481481	-0.907407407407407	0.284756941074796	0.286318007034054
7830	-0.462962962962963	-0.925925925925926	0.227795474847902	0.22905652872545
7840	-0.481481481481481	-0.962962962962963	0.115263427892189	0.115896503288703
7850	-0.037037037037037	-0.574074074074074	0.112323651054849	0.112963614600219
7860	-0.12962962962963	-0.703703703703704	0.31569954411053	0.317704766520999
7870	-0.259259259259259	-0.851851851851852	0.324327126484346	0.326444694037409
7880	-0.037037037037037	-0.685185185185185	0.0964037683944825	0.0969942147430386
7890	-0.240740740740741	-0.925925925925926	0.157200139814664	0.158258212856816
7900	-0.185185185185185	-0.925925925925926	0.125836857131405	0.126725491458587
7910	-0.037037037037037	-0.851851851851852	0.0517646170014324	0.052102404686486
7920	-0.037037037037037	-0.962962962962963	0.0134877395743289	0.0134775647100881

TABLE-2a

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 2: (Output for 1/10th of the FEM MODEL)

Let $\Omega = [0,1]^2$ as in first example, consider the boundary value problem: Find u such that

$-\epsilon^2 \Delta u + 2u = f$ in Ω ; u = 0, on $\partial\Omega$, ($\epsilon = 10^{-2}$), Where f is chosen such that the solution is

$$u(x,y) = \left(1 - \frac{e^{-x/\epsilon}+e^{-(1-x)/\epsilon}}{1+e^{-1/\epsilon}}\right)\left(1 - \frac{e^{-y}+e^{-(1-y)/\epsilon}}{1+e^{-1/\epsilon}}\right)$$

ALL QUADRILATERAL FEM MODEL : number of nodes=2481,elements=2400 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
238	0.516666666666667	0.466666666666667	0.971892341120335	0.972300695496585
248	0.533333333333333	0.433333333333333	0.9689578575488	0.969422761553851
258	0.566666666666667	0.366666666666667	0.955913632212716	0.956534552851856
268	0.583333333333333	0.283333333333333	0.922089908645775	0.923081267729785
278	0.616666666666667	0.166666666666667	0.789774258031817	0.791646473021132
288	0.633333333333333	0.0333333333333333	0.273725982067214	0.27568969108262
298	0.683333333333333	0.183333333333333	0.801679038635212	0.803546243342903
308	0.733333333333333	0.233333333333333	0.837562529460848	0.839264144340908
318	0.783333333333333	0.183333333333333	0.740833309357672	0.743303605425222
328	0.833333333333333	0.0333333333333333	0.227123041786856	0.229837667439691
547	0.533333333333333	0.483333333333333	0.971891993590695	0.972300695496585
557	0.566666666666667	0.466666666666667	0.968957181461474	0.969422761553851
567	0.633333333333333	0.433333333333333	0.955912579938364	0.956534552851856
577	0.716666666666667	0.366666666666667	0.913677523644596	0.914706803562037
587	0.833333333333333	0.283333333333333	0.760440090966617	0.762575043326194
597	0.966666666666667	0.166666666666667	0.227116897002209	0.229837667439691
607	0.816666666666667	0.366666666666667	0.81512084608795	0.816885266105045
617	0.766666666666667	0.466666666666667	0.888222689739451	0.889720008155483
627	0.816666666666667	0.466666666666667	0.825820831383372	0.827891860444745
637	0.966666666666667	0.366666666666667	0.273706310432168	0.27568969108262
856	0.533333333333333	0.516666666666667	0.971892341120335	0.972300695496585
866	0.566666666666667	0.533333333333333	0.9689578575488	0.969422761553851
876	0.633333333333333	0.566666666666667	0.955913632212717	0.956534552851856
886	0.716666666666667	0.583333333333333	0.922089908645777	0.923081267729785
896	0.833333333333333	0.616666666666667	0.789774258031817	0.791646473021132
906	0.966666666666667	0.633333333333333	0.273725982067214	0.27568969108262
916	0.816666666666667	0.683333333333333	0.801679038635211	0.803546243342903
926	0.766666666666667	0.733333333333333	0.837562529460849	0.839264144340908
936	0.816666666666667	0.783333333333333	0.740833309357671	0.743303605425222
946	0.966666666666667	0.833333333333333	0.227123041786856	0.229837667439691
1165	0.516666666666667	0.533333333333333	0.971891993590695	0.972300695496585
1175	0.533333333333333	0.566666666666667	0.968957181461473	0.969422761553851
1185	0.566666666666667	0.633333333333333	0.955912579938363	0.956534552851856
1195	0.633333333333333	0.716666666666667	0.913677523644595	0.914706803562037
1205	0.716666666666667	0.833333333333333	0.760440090966616	0.762575043326194
1215	0.833333333333333	0.966666666666667	0.227116897002209	0.229837667439691
1225	0.633333333333333	0.816666666666667	0.81512084608795	0.816885266105045

1235	0.533333333333333	0.766666666666667	0.888222689739452	0.889720008155483
1245	0.533333333333333	0.816666666666667	0.825820831383372	0.827891860444745
1255	0.633333333333333	0.966666666666667	0.273706310432168	0.27568969108262
1474	0.483333333333333	0.533333333333333	0.971892341120334	0.972300695496585
1484	0.466666666666667	0.566666666666667	0.9689578575488	0.969422761553851
1494	0.433333333333333	0.633333333333333	0.955913632212719	0.956534552851856
1504	0.416666666666667	0.716666666666667	0.922089908645779	0.923081267729785
1514	0.383333333333333	0.833333333333333	0.789774258031819	0.791646473021132
1524	0.366666666666667	0.966666666666667	0.273725982067215	0.27568969108262
1534	0.316666666666667	0.816666666666667	0.801679038635212	0.803546243342903
1544	0.266666666666667	0.766666666666667	0.837562529460849	0.839264144340908
1554	0.216666666666667	0.816666666666667	0.740833309357671	0.743303605425222
1564	0.166666666666667	0.966666666666667	0.227123041786856	0.229837667439691
1783	0.466666666666667	0.516666666666667	0.971891993590695	0.972300695496585
1793	0.433333333333333	0.533333333333333	0.968957181461472	0.969422761553851
1803	0.366666666666667	0.566666666666667	0.955912579938363	0.956534552851856
1813	0.283333333333333	0.633333333333333	0.913677523644595	0.914706803562037
1823	0.166666666666667	0.716666666666667	0.760440090966617	0.762575043326194
1833	0.0333333333333333	0.833333333333333	0.227116897002209	0.229837667439691
1843	0.183333333333333	0.633333333333333	0.815120846087953	0.816885266105045
1853	0.233333333333333	0.533333333333333	0.888222689739451	0.889720008155483
1863	0.183333333333333	0.533333333333333	0.825820831383372	0.827891860444745
1873	0.0333333333333333	0.633333333333333	0.273706310432169	0.27568969108262
2092	0.466666666666667	0.483333333333333	0.971892341120334	0.972300695496585
2102	0.433333333333333	0.466666666666667	0.968957857548799	0.969422761553851
2112	0.366666666666667	0.433333333333333	0.955913632212718	0.956534552851856
2122	0.283333333333333	0.416666666666667	0.922089908645778	0.923081267729785
2132	0.166666666666667	0.383333333333333	0.789774258031818	0.791646473021132
2142	0.0333333333333333	0.366666666666667	0.273725982067215	0.27568969108262
2152	0.183333333333333	0.316666666666667	0.801679038635212	0.803546243342903
2162	0.233333333333333	0.266666666666667	0.837562529460848	0.839264144340908
2172	0.183333333333333	0.216666666666667	0.74083330935767	0.743303605425222
2182	0.0333333333333333	0.166666666666667	0.227123041786856	0.229837667439691
2382	0.483333333333333	0.466666666666667	0.971891993590695	0.972300695496585
2392	0.466666666666667	0.433333333333333	0.968957181461472	0.969422761553851
2402	0.433333333333333	0.366666666666667	0.955912579938362	0.956534552851856
2412	0.366666666666667	0.283333333333333	0.913677523644593	0.914706803562037
2422	0.283333333333333	0.166666666666667	0.760440090966616	0.762575043326194
2432	0.166666666666667	0.0333333333333333	0.227116897002209	0.229837667439691
2442	0.366666666666667	0.183333333333333	0.81512084608795	0.816885266105045
2452	0.466666666666667	0.233333333333333	0.888222689739448	0.889720008155483
2462	0.466666666666667	0.183333333333333	0.82582083138337	0.827891860444745
2472	0.366666666666667	0.0333333333333333	0.273706310432168	0.27568969108262

TABLE-2b

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 2: (Output for 1/10th of the FEM MODEL)

Let $\Omega = [0,1]^2$ as in first example, consider the boundary value problem: Find u such that

$-\epsilon^2 \Delta u + 2u = f$ in Ω ; u = 0, on $\partial\Omega$, ($\epsilon = 10^{-2}$), Where f is chosen such that the solution is

$$u(x,y) = \left(1 - \frac{e^{-x/\epsilon}+e^{-(1-x)/\epsilon}}{1+e^{-1/\epsilon}}\right)\left(1 - \frac{e^{-y}+e^{-(1-y)/\epsilon}}{1+e^{-1/\epsilon}}\right)$$

ALL QUADRILATERAL FEM MODEL : number of nodes=5521,elements=5400 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
503	0.511111111111111	0.477777777777778	0.972648109210587	0.972819309791905
513	0.511111111111111	0.144444444444444	0.752313177607539	0.753583227885767
523	0.522222222222222	0.288888888888889	0.930049488864604	0.930518759022195
533	0.544444444444444	0.411111111111111	0.966071628836979	0.966297126545375
543	0.544444444444444	0.0777777777777778	0.531175246180661	0.532481618077092
553	0.555555555555556	0.188888888888889	0.834393755988751	0.835223350158546
563	0.577777777777778	0.277777777777778	0.919996928603406	0.920452578663593
573	0.588888888888889	0.355555555555556	0.95096626282234	0.951265506274738
583	0.588888888888889	0.0222222222222222	0.194602174926996	0.195425023876328
593	0.611111111111111	0.0777777777777778	0.527139741218382	0.52823499118834
603	0.622222222222222	0.122222222222222	0.686719854211691	0.687753059384665
613	0.644444444444444	0.144444444444444	0.739967267412621	0.740905692676139
623	0.655555555555556	0.155555555555556	0.761507502498297	0.762423906196243
633	0.677777777777778	0.144444444444444	0.731645235787836	0.732617045717805
643	0.688888888888889	0.122222222222222	0.672062580485487	0.673145746406093

653	0.711111111111111	0.0777777777777778	0.508809414087934	0.509984437773636
663	0.722222222222222	0.0222222222222222	0.185846273844594	0.186709149691752
673	0.755555555555556	0.188888888888889	0.773410186552295	0.774401233990219
683	0.777777777777778	0.0777777777777778	0.480327614904491	0.48169994450667
693	0.811111111111111	0.144444444444444	0.646792055548604	0.648178815406523
703	0.844444444444444	0.144444444444444	0.600993651713134	0.60253751542408
713	0.877777777777778	0.0777777777777778	0.379361960613293	0.381203568266568
723	0.922222222222222	0.0222222222222222	0.10639626709421	0.107689684743707
1197	0.688888888888889	0.322222222222222	0.914843576361475	0.91529674334271
1207	0.544444444444444	0.477777777777778	0.971377080103094	0.97156716510466
1217	0.877777777777778	0.144444444444444	0.537108031576924	0.538795807650862
1227	0.755555555555556	0.288888888888889	0.860509417109009	0.861182860827226
1237	0.644444444444444	0.411111111111111	0.950966058112998	0.951265506274738
1247	0.977777777777778	0.0777777777777778	0.106398444401483	0.107689684743707
1257	0.888888888888889	0.188888888888889	0.567555251771343	0.569053240347773
1267	0.811111111111111	0.277777777777778	0.794198263850161	0.795095101191394
1277	0.755555555555556	0.355555555555556	0.884613375272887	0.885185181991689
1287	0.711111111111111	0.411111111111111	0.925042662968716	0.925471377928726
1297	0.688888888888889	0.455555555555556	0.9398952717804	0.940279068944549
1307	0.677777777777778	0.477777777777778	0.945377755192921	0.945753544378267
1317	0.688888888888889	0.488888888888889	0.941076130082734	0.941490889890124
1327	0.711111111111111	0.477777777777778	0.9300485205595	0.930518759022195
1337	0.755555555555556	0.455555555555556	0.898578533383568	0.899172620238032
1347	0.811111111111111	0.411111111111111	0.83143133109012	0.832211380057201
1357	0.888888888888889	0.355555555555556	0.649410826186713	0.650460606221821
1367	0.977777777777778	0.277777777777778	0.185843011514053	0.186709149691753
1377	0.877777777777778	0.411111111111111	0.690702118100778	0.691772072761953
1387	0.844444444444444	0.477777777777778	0.776688090627414	0.777830647217426
1397	0.877777777777778	0.477777777777778	0.694189892440155	0.695544893147672
1407	0.977777777777778	0.411111111111111	0.194597516493808	0.195425023876329
1881	0.522222222222222	0.511111111111111	0.972648109210585	0.972819309791905
1891	0.855555555555556	0.511111111111111	0.75231317760754	0.753583227885767
1901	0.711111111111111	0.522222222222222	0.930049488864605	0.930518759022195
1911	0.588888888888889	0.544444444444444	0.966071628836976	0.966297126545375
1921	0.922222222222222	0.544444444444444	0.531175246180661	0.532481618077091
1931	0.811111111111111	0.555555555555556	0.834393755988749	0.835223350158546
1941	0.722222222222222	0.577777777777778	0.919996928603404	0.920452578663593
1951	0.644444444444444	0.588888888888889	0.950966262822336	0.951265506274738
1961	0.977777777777778	0.588888888888889	0.194602174926996	0.195425023876329
1971	0.922222222222222	0.611111111111111	0.52713974121838	0.52823499118834
1981	0.877777777777778	0.622222222222222	0.686719854211688	0.687753059384665
1991	0.855555555555556	0.644444444444444	0.739967267412618	0.740905692676139
2001	0.844444444444444	0.655555555555556	0.761507502498294	0.762423906196243
2011	0.855555555555556	0.677777777777778	0.731645235787834	0.732617045717805
2021	0.877777777777778	0.688888888888889	0.672062580485485	0.673145746406093
2031	0.922222222222222	0.711111111111111	0.508809414087933	0.509984437773636
2041	0.977777777777778	0.722222222222222	0.185846273844594	0.186709149691753
2051	0.811111111111111	0.755555555555556	0.773410186552295	0.774401233990219
2061	0.922222222222222	0.777777777777778	0.480327614904489	0.48169994450667
2071	0.855555555555556	0.811111111111111	0.646792055548601	0.648178815406523
2081	0.855555555555556	0.844444444444444	0.600993651713132	0.602537515424081
2091	0.922222222222222	0.877777777777778	0.379361960613292	0.381203568266568
2101	0.977777777777778	0.922222222222222	0.10639626709421	0.107689684743707
2575	0.677777777777778	0.688888888888889	0.914843576361473	0.91529674334271
2585	0.522222222222222	0.544444444444444	0.971377080103093	0.97156716510466
2595	0.855555555555556	0.877777777777778	0.537108031576922	0.538795807650862
2605	0.711111111111111	0.755555555555556	0.860509417109009	0.861182860827226
2615	0.588888888888889	0.644444444444444	0.950966058112995	0.951265506274738
2625	0.922222222222222	0.977777777777778	0.106398444401483	0.107689684743707
2635	0.811111111111111	0.888888888888889	0.56755525177134	0.569053240347773
2645	0.722222222222222	0.811111111111111	0.794198263850162	0.795095101191394
2655	0.644444444444444	0.755555555555556	0.884613375272884	0.885185181991689
2665	0.588888888888889	0.711111111111111	0.925042662968712	0.925471377928726
2675	0.544444444444444	0.688888888888889	0.939895271780399	0.940279068944549
2685	0.522222222222222	0.677777777777778	0.94537775519292	0.945753544378267
2695	0.511111111111111	0.688888888888889	0.941076130082733	0.941490889890124
2705	0.522222222222222	0.711111111111111	0.9300485205595	0.930518759022195
2715	0.544444444444444	0.755555555555556	0.898578533383566	0.899172620238032
2725	0.588888888888889	0.811111111111111	0.831431331090118	0.832211380057201
2735	0.644444444444444	0.888888888888889	0.64941082618671	0.650460606221821
2745	0.722222222222222	0.977777777777778	0.185843011514053	0.186709149691753
2755	0.588888888888889	0.877777777777778	0.690702118100776	0.691772072761953
2765	0.522222222222222	0.844444444444444	0.776688090627412	0.777830647217426
2775	0.522222222222222	0.877777777777778	0.694189892440153	0.695544893147672

2785	0.588888888888889	0.977777777777778	0.194597516493807	0.195425023876329
3259	0.488888888888889	0.522222222222222	0.972648109210586	0.972819309791905
3269	0.488888888888889	0.855555555555556	0.752313177607537	0.753583227885767
3279	0.477777777777778	0.711111111111111	0.930049488864605	0.930518759022195
3289	0.455555555555556	0.588888888888889	0.966071628836979	0.966297126545375
3299	0.455555555555556	0.922222222222222	0.53117524618066	0.532481618077091
3309	0.444444444444444	0.811111111111111	0.834393755988748	0.835223350158546
3319	0.422222222222222	0.722222222222222	0.919996928603407	0.920452578663593
3329	0.411111111111111	0.644444444444444	0.950966262822338	0.951265506274738
3339	0.411111111111111	0.977777777777778	0.194602174926996	0.195425023876329
3349	0.388888888888889	0.922222222222222	0.527139741218381	0.52823499118834
3359	0.377777777777778	0.877777777777778	0.686719854211689	0.687753059384665
3369	0.355555555555556	0.855555555555556	0.739967267412619	0.740905692676139
3379	0.344444444444444	0.844444444444444	0.761507502498294	0.762423906196243
3389	0.322222222222222	0.855555555555556	0.731645235787834	0.732617045717805
3399	0.311111111111111	0.877777777777778	0.672062580485485	0.673145746406093
3409	0.288888888888889	0.922222222222222	0.508809414087933	0.509984437773636
3419	0.277777777777778	0.977777777777778	0.185846273844594	0.186709149691753
3429	0.244444444444444	0.811111111111111	0.773410186552294	0.774401233990219
3439	0.222222222222222	0.922222222222222	0.480327614904491	0.48169994450667
3449	0.188888888888889	0.855555555555556	0.646792055548602	0.648178815406523
3459	0.155555555555556	0.855555555555556	0.600993651713133	0.602537515424081
3469	0.122222222222222	0.922222222222222	0.379361960613293	0.381203568266567
3479	0.0777777777777778	0.977777777777778	0.10639626709421	0.107689684743707
3953	0.311111111111111	0.677777777777778	0.914843576361474	0.91529674334271
3963	0.455555555555556	0.522222222222222	0.971377080103097	0.97156716510466
3973	0.122222222222222	0.855555555555556	0.537108031576923	0.538795807650862
3983	0.244444444444444	0.711111111111111	0.860509417109006	0.861182860827226
3993	0.355555555555556	0.588888888888889	0.950966058112997	0.951265506274738
4003	0.0222222222222222	0.922222222222222	0.106398444401483	0.107689684743707
4013	0.111111111111111	0.811111111111111	0.567555251771341	0.569053240347773
4023	0.188888888888889	0.722222222222222	0.794198263850161	0.795095101191394
4033	0.244444444444444	0.644444444444444	0.884613375272882	0.885185181991689
4043	0.288888888888889	0.588888888888889	0.925042662968713	0.925471377928726
4053	0.311111111111111	0.544444444444444	0.939895271780399	0.940279068944549
4063	0.322222222222222	0.522222222222222	0.945377755192921	0.945753544378267
4073	0.311111111111111	0.511111111111111	0.941076130082735	0.941490889890124
4083	0.288888888888889	0.522222222222222	0.930048520559499	0.930518759022195
4093	0.244444444444444	0.544444444444444	0.898578533383565	0.899172620238032
4103	0.188888888888889	0.588888888888889	0.831431331090118	0.832211380057201
4113	0.111111111111111	0.644444444444444	0.649410826186712	0.650460606221821
4123	0.0222222222222222	0.722222222222222	0.185843011514054	0.186709149691752
4133	0.122222222222222	0.588888888888889	0.690702118100778	0.691772072761953
4143	0.155555555555556	0.522222222222222	0.776688090627414	0.777830647217426
4153	0.122222222222222	0.522222222222222	0.694189892440155	0.695544893147671
4163	0.0222222222222222	0.588888888888889	0.194597516493807	0.195425023876328
4637	0.477777777777778	0.488888888888889	0.972648109210588	0.972819309791905
4647	0.144444444444444	0.488888888888889	0.752313177607539	0.753583227885767
4657	0.288888888888889	0.477777777777778	0.930049488864606	0.930518759022195
4667	0.411111111111111	0.455555555555556	0.966071628836981	0.966297126545375
4677	0.0777777777777778	0.455555555555556	0.531175246180661	0.532481618077092
4687	0.188888888888889	0.444444444444444	0.834393755988749	0.835223350158546
4697	0.277777777777778	0.422222222222222	0.919996928603407	0.920452578663593
4707	0.355555555555556	0.411111111111111	0.95096626282234	0.951265506274738
4717	0.0222222222222222	0.411111111111111	0.194602174926996	0.195425023876328
4727	0.0777777777777778	0.388888888888889	0.527139741218383	0.52823499118834
4737	0.122222222222222	0.377777777777778	0.686719854211691	0.687753059384665
4747	0.144444444444444	0.355555555555556	0.739967267412621	0.740905692676139
4757	0.155555555555556	0.344444444444444	0.761507502498297	0.762423906196243
4767	0.144444444444444	0.322222222222222	0.731645235787836	0.732617045717805
4777	0.122222222222222	0.311111111111111	0.672062580485487	0.673145746406093
4787	0.0777777777777778	0.288888888888889	0.508809414087934	0.509984437773636
4797	0.0222222222222222	0.277777777777778	0.185846273844595	0.186709149691752
4807	0.188888888888889	0.244444444444444	0.773410186552295	0.774401233990219
4817	0.0777777777777778	0.222222222222222	0.480327614904491	0.48169994450667
4827	0.144444444444444	0.188888888888889	0.646792055548604	0.648178815406523
4837	0.144444444444444	0.155555555555556	0.600993651713135	0.60253751542408
4847	0.0777777777777778	0.122222222222222	0.379361960613294	0.381203568266568
4857	0.0222222222222222	0.0777777777777778	0.10639626709421	0.107689684743707
5302	0.322222222222222	0.311111111111111	0.914843576361474	0.91529674334271
5312	0.477777777777778	0.455555555555556	0.971377080103096	0.97156716510466
5322	0.144444444444444	0.122222222222222	0.537108031576924	0.538795807650861
5332	0.288888888888889	0.244444444444444	0.860509417109007	0.861182860827226
5342	0.411111111111111	0.355555555555556	0.950966058113	0.951265506274738

5352	0.0777777777777778	0.0222222222222222	0.106398444401483	0.107689684743707
5362	0.188888888888889	0.111111111111111	0.567555251771342	0.569053240347773
5372	0.277777777777778	0.188888888888889	0.794198263850162	0.795095101191394
5382	0.355555555555556	0.244444444444444	0.884613375272886	0.885185181991689
5392	0.411111111111111	0.288888888888889	0.925042662968716	0.925471377928726
5402	0.455555555555556	0.311111111111111	0.9398952717804	0.940279068944549
5412	0.477777777777778	0.322222222222222	0.94537775519292	0.945753544378267
5422	0.488888888888889	0.311111111111111	0.941076130082732	0.941490889890124
5432	0.477777777777778	0.288888888888889	0.930048520559497	0.930518759022195
5442	0.455555555555556	0.244444444444444	0.898578533383565	0.899172620238032
5452	0.411111111111111	0.188888888888889	0.831431331090121	0.832211380057201
5462	0.355555555555556	0.111111111111111	0.649410826186712	0.650460606221821
5472	0.277777777777778	0.0222222222222222	0.185843011514054	0.186709149691752
5482	0.411111111111111	0.122222222222222	0.69070211810078	0.691772072761953
5492	0.477777777777778	0.155555555555556	0.776688090627413	0.777830647217426
5502	0.477777777777778	0.122222222222222	0.694189892440155	0.695544893147671
5512	0.411111111111111	0.0222222222222222	0.194597516493808	0.195425023876328

TABLE-2c

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 2: (Output for 1/10th of the FEM MODEL)

Let $\Omega = [0,1]^2$ as in first example, consider the boundary value problem: Find u such that $-\epsilon^2 \Delta u + 2u = f$ in Ω ; u = 0, on $\partial\Omega$, ($\epsilon = 10^{-2}$), Where f is chosen such that the solution is $u(x,y) = \left(1 - \frac{e^{-x/\epsilon}+e^{-(1-x)/\epsilon}}{1+e^{-1/\epsilon}}\right)\left(1 - \frac{e^{-y}+e^{-(1-y)/\epsilon}}{1+e^{-1/\epsilon}}\right)$

ALL QUADRILATERAL FEM MODEL : number of nodes=7921,elements=7776 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
710	0.509259259259259	0.481481481481481	0.972829471675747	0.972945409189695
720	0.509259259259259	0.203703703703704	0.856865114606724	0.857479718249528
730	0.518518518518518	0.407407407407407	0.966725682375948	0.966886306327947
740	0.518518518518518	0.12962962962963	0.715421218674884	0.716347972681752
750	0.537037037037037	0.314814814814815	0.941973144329956	0.942236677985965
760	0.537037037037037	0.037037037037037	0.304230598310567	0.3050272999229
770	0.546296296296296	0.212962962962963	0.867069976462932	0.867579186469399
780	0.564814814814815	0.37037037037037	0.957364956426787	0.957554070227517
790	0.564814814814815	0.0925925925925926	0.592998353120211	0.593833883949959
800	0.574074074074074	0.240740740740741	0.893273931544173	0.893677626529958
810	0.592592592592593	0.37037037037037	0.954174837078291	0.954370472864517
820	0.592592592592593	0.0925925925925926	0.591078421110322	0.591859553678925
830	0.601851851851852	0.212962962962963	0.861700093709687	0.862167095574095
840	0.62037037037037	0.314814814814815	0.932349893301292	0.932614913545918
850	0.62037037037037	0.037037037037037	0.301262707071444	0.301912476549738
860	0.62962962962963	0.12962962962963	0.706374144070727	0.707075225855912
870	0.648148148148148	0.203703703703704	0.841649895371943	0.842145958023461
880	0.657407407407407	0.268518518518519	0.899189394088432	0.899550918334965
890	0.675925925925926	0.314814814814815	0.917183509798347	0.917492168041739
900	0.675925925925926	0.037037037037037	0.296352154748762	0.297016837973641
910	0.685185185185185	0.0740740740740741	0.49934542520333	0.500153271334746
920	0.703703703703704	0.0925925925925926	0.571206532326082	0.572017340239922
930	0.712962962962963	0.101851851851852	0.601219771894147	0.602047482520267
940	0.731481481481482	0.0925925925925926	0.561302044562778	0.562158831431857
950	0.740740740740741	0.0740740740740741	0.482815058379688	0.483703862061459
960	0.759259259259259	0.037037037037037	0.280705468409245	0.281463455588537
970	0.787037037037037	0.203703703703704	0.764853805650222	0.765579229311851
980	0.796296296296296	0.12962962962963	0.63036888352719	0.631334350295504
990	0.814814814814815	0.037037037037037	0.259955246757999	0.260826897734778
1000	0.842592592592593	0.0925925925925926	0.477343531430973	0.478518060321853
1010	0.87037037037037	0.0925925925925926	0.437226810186912	0.438497668873201
1020	0.898148148148148	0.037037037037037	0.196586859282413	0.197688297007127
1030	0.953703703703704	0.037037037037037	0.113572489336139	0.114677486322676
1705	0.685185185185185	0.324074074074074	0.917183504843525	0.917492168041739
1715	0.962962962962963	0.0462962962962963	0.113573098418368	0.114677486322676
1725	0.759259259259259	0.259259259259259	0.840323552182546	0.840852368658795
1735	0.574074074074074	0.462962962962963	0.96834784957294	0.968495440560636
1745	0.851851851851852	0.185185185185185	0.650081064139712	0.651049410068577
1755	0.675925925925926	0.37037037037037	0.934040274768866	0.934299524500891
1765	0.953703703703704	0.0925925925925926	0.222481095316337	0.223709738612429
1775	0.796296296296296	0.268518518518519	0.808779307596363	0.809377576662876
1785	0.648148148148148	0.425925925925926	0.951813890851143	0.952022872541603

1795	0.925925925925926	0.148148148148148	0.402947235050094	0.404150752843076
1805	0.796296296296296	0.296296296296296	0.823019207229537	0.823571529550279
1815	0.675925925925926	0.425925925925926	0.942818604537623	0.943061337198075
1825	0.953703703703704	0.148148148148148	0.285170567518373	0.286246205690198
1835	0.851851851851852	0.268518518518519	0.718557634297157	0.719306336374506
1845	0.759259259259259	0.37037037037037	0.884973014970196	0.885374628976708
1855	0.685185185185185	0.462962962962963	0.941972847927786	0.942236677985965
1865	0.962962962962963	0.185185185185185	0.259954860801068	0.260826897734779
1875	0.898148148148148	0.259259259259259	0.589700985486928	0.590579947393127
1885	0.851851851851852	0.324074074074074	0.740715712923831	0.741383035348618
1895	0.814814814814815	0.37037037037037	0.819915535272791	0.820460039212565
1905	0.796296296296296	0.407407407407407	0.851648209102915	0.852139688104298
1915	0.787037037037037	0.425925925925926	0.864985867994097	0.865466288953381
1925	0.796296296296296	0.435185185185185	0.854470367806726	0.854982263122166
1935	0.814814814814815	0.425925925925926	0.827587644515727	0.828154271094943
1945	0.851851851851852	0.407407407407407	0.756652045534653	0.757309684383465
1955	0.898148148148148	0.37037037037037	0.621088377993061	0.621850542727617
1965	0.962962962962963	0.324074074074074	0.296350851070162	0.297016837973642
1975	0.851851851851852	0.462962962962963	0.760612607255966	0.761377930705403
1985	0.953703703703704	0.37037037037037	0.36000745315177	0.360730797867231
1995	0.925925925925926	0.425925925925926	0.513253874085413	0.514091813857838
2005	0.953703703703704	0.425925925925926	0.363345683463419	0.364113712662898
2015	0.962962962962963	0.462962962962963	0.304227200333987	0.305027299922901
2690	0.574074074074074	0.509259259259259	0.969208671565448	0.969357267272353
2700	0.851851851851852	0.509259259259259	0.761180269836351	0.762055451539176
2710	0.648148148148148	0.518518518518518	0.955329096036218	0.955546850016747
2720	0.925925925925926	0.518518518518518	0.514896611538708	0.515994759705509
2730	0.740740740740741	0.537037037037037	0.910867015855143	0.911247703931811
2740	0.564814814814815	0.546296296296296	0.968749041208452	0.968893600930157
2750	0.842592592592593	0.546296296296296	0.78003475090132	0.780745422425951
2760	0.685185185185185	0.564814814814815	0.940073160823368	0.940328381714164
2770	0.962962962962963	0.564814814814815	0.303706990830652	0.304409533205854
2780	0.814814814814815	0.574074074074074	0.827588501083369	0.828154271094943
2790	0.685185185185185	0.592592592592593	0.936946164892905	0.937202054909801
2800	0.962962962962963	0.592592592592593	0.302734532694205	0.303397457316546
2810	0.842592592592593	0.601851851851852	0.775244443682828	0.775875013755295
2820	0.740740740740741	0.62037037037037	0.901586024534221	0.901942387169459
2830	0.648148148148148	0.62962962962963	0.942947588677599	0.943177799836758
2840	0.925925925925926	0.62962962962963	0.508534961924247	0.509315479589315
2850	0.851851851851852	0.648148148148148	0.747781227741177	0.748428102315426
2860	0.787037037037037	0.657407407407407	0.850390517284424	0.850872965402772
2870	0.740740740740741	0.675925925925926	0.886922727872183	0.887317009661034
2880	0.703703703703704	0.685185185185185	0.905438303634261	0.90578216298882
2890	0.981481481481482	0.685185185185185	0.161048120370943	0.161597449189587
2900	0.962962962962963	0.703703703703704	0.292539430031756	0.293225995070975
2910	0.953703703703704	0.712962962962963	0.348470538157244	0.349243192376889
2920	0.962962962962963	0.731481481481482	0.287454914139983	0.288172352721692
2930	0.981481481481482	0.740740740740741	0.155699432711709	0.15628271322446
2940	0.787037037037037	0.768518518518518	0.792682988961814	0.7933377402101
2950	0.851851851851852	0.787037037037037	0.679498756135927	0.680382069529582
2960	0.925925925925926	0.796296296296296	0.453735639471373	0.454758342031755
2970	0.842592592592593	0.824074074074074	0.654953833239205	0.655925941595335
2980	0.962962962962963	0.842592592592593	0.244351815441343	0.245296644920687
2990	0.962962962962963	0.87037037037037	0.223756516317942	0.22478149917224
3000	0.925925925925926	0.907407407407407	0.314497785414746	0.315855572864409
3010	0.981481481481482	0.962962962962963	0.0515095405992999	0.0523134311711246
3685	0.731481481481482	0.740740740740741	0.860421412966182	0.860894729162419
3695	0.518518518518518	0.537037037037037	0.971952326428513	0.972080392367733
3705	0.796296296296296	0.814814814814815	0.731769404801563	0.732573546432305
3715	0.592592592592593	0.62962962962963	0.954174783239811	0.954370472864517
3725	0.87037037037037	0.907407407407407	0.437226869533282	0.4384976688732
3735	0.685185185185185	0.731481481481482	0.889782511806374	0.89017134002274
3745	0.509259259259259	0.574074074074074	0.969208559633658	0.969357267272353
3755	0.787037037037037	0.851851851851852	0.67949860955723	0.680382069529582
3765	0.62962962962963	0.703703703703704	0.922078000295411	0.922375006195524
3775	0.907407407407407	0.981481481481482	0.101232335456841	0.102051626596366
3785	0.759259259259259	0.851851851851852	0.701750670118949	0.702560273914366
3795	0.62962962962963	0.731481481481482	0.90613392428027	0.906478211669848
3805	0.509259259259259	0.62962962962963	0.96015285906719	0.960351143834264
3815	0.787037037037037	0.907407407407407	0.530747336717756	0.531738384318638
3825	0.685185185185185	0.814814814814815	0.805118993808477	0.805700570778818
3835	0.592592592592593	0.740740740740741	0.906025501451709	0.906378663248611
3845	0.518518518518518	0.675925925925926	0.946292036202491	0.946552142940057
3855	0.796296296296296	0.953703703703704	0.321168633646763	0.322089836519737

3865	0.731481481481482	0.907407407407407	0.561301455405981	0.562158831431857
3875	0.685185185185185	0.87037037037037	0.693621647572493	0.694355466236271
3885	0.648148148148148	0.851851851851852	0.747780626185652	0.748428102315426
3895	0.62962962962963	0.842592592592593	0.770983113178717	0.771607900328534
3905	0.62037037037037	0.851851851851852	0.752959808269207	0.753603027360782
3915	0.62962962962963	0.87037037037037	0.706373247561672	0.707075225855912
3925	0.648148148148148	0.907407407407407	0.584156682290106	0.584918338866609
3935	0.685185185185185	0.953703703703704	0.353495141890177	0.354241517987968
3945	0.518518518518518	0.814814814814815	0.830546166608593	0.831219740509025
3955	0.592592592592593	0.907407407407407	0.591076992667968	0.591859553678925
3965	0.509259259259259	0.851851851851852	0.761178278389422	0.762055451539176
3975	0.62962962962963	0.981481481481482	0.164019661251167	0.164557720705818
3985	0.509259259259259	0.907407407407407	0.594439822262641	0.595568508798031
3995	0.518518518518518	0.953703703703704	0.364429117544163	0.36546150435869
4670	0.490740740740741	0.62962962962963	0.96015305904683	0.960351143834264
4680	0.490740740740741	0.907407407407407	0.594443252582	0.595568508798031
4690	0.481481481481481	0.703703703703704	0.934158096237829	0.934471243764278
4700	0.481481481481481	0.981481481481482	0.166018636906811	0.166715768430507
4710	0.462962962962963	0.796296296296296	0.856166766136472	0.856717358538725
4720	0.453703703703704	0.62037037037037	0.960765072831729	0.960945813652305
4730	0.453703703703704	0.898148148148148	0.628317675607239	0.629214610764044
4740	0.435185185185185	0.740740740740741	0.909041692874167	0.909402169113725
4750	0.425925925925926	0.592592592592593	0.96315547351646	0.96332050987907
4760	0.425925925925926	0.87037037037037	0.712960067005559	0.713706140813381
4770	0.407407407407407	0.740740740740741	0.906025823079947	0.906378663248611
4780	0.398148148148148	0.62037037037037	0.954758479259975	0.954951287538657
4790	0.398148148148148	0.898148148148148	0.624515437510423	0.625289474339361
4800	0.37962962962963	0.796296296296296	0.847482291090446	0.847968885030797
4810	0.37037037037037	0.703703703703704	0.922078149774951	0.922375006195524
4820	0.37037037037037	0.981481481481482	0.164021673944724	0.164557720705818
4830	0.351851851851852	0.907407407407407	0.584157629414835	0.584918338866609
4840	0.342592592592593	0.842592592592593	0.765077091552256	0.765711283955105
4850	0.324074074074074	0.796296296296296	0.833700635088016	0.834218710701044
4860	0.314814814814815	0.759259259259259	0.869003795698982	0.869447395151934
4870	0.296296296296296	0.740740740740741	0.875563598845975	0.875992131881589
4880	0.287037037037037	0.731481481481482	0.877185194734227	0.877611078220692
4890	0.268518518518519	0.740740740740741	0.860421411910967	0.860894729162419
4900	0.259259259259259	0.759259259259259	0.840323618979036	0.840852368658795
4910	0.240740740740741	0.796296296296296	0.789878486517137	0.790534579226001
4920	0.231481481481481	0.842592592592593	0.713123123096131	0.713934611118787
4930	0.212962962962963	0.907407407407407	0.530747686953192	0.531738384318638
4940	0.203703703703704	0.981481481481482	0.146290006548815	0.146930535661403
4950	0.175925925925926	0.898148148148148	0.527510282931595	0.528620692707413
4960	0.148148148148148	0.87037037037037	0.55993132347411	0.56107657492912
4970	0.12037037037037	0.898148148148148	0.445700113728918	0.446993285372569
4980	0.0925925925925926	0.981481481481482	0.101231415269261	0.102051626596366
5655	0.481481481481481	0.509259259259259	0.972829439455836	0.972945409189695
5665	0.203703703703704	0.787037037037037	0.764853817735103	0.765579229311851
5675	0.407407407407407	0.574074074074074	0.963155440398035	0.96332050987907
5685	0.12962962962963	0.851851851851852	0.559931199283641	0.56107657492912
5695	0.314814814814815	0.648148148148148	0.92592814967693	0.926210729779986
5705	0.037037037037037	0.925925925925926	0.160765658422568	0.161913036784921
5715	0.212962962962963	0.740740740740741	0.813713971145056	0.814308602405627
5725	0.37037037037037	0.564814814814815	0.95736485485431	0.957554070227517
5735	0.0925925925925926	0.842592592592593	0.477343463314636	0.478518060321853
5745	0.240740740740741	0.685185185185185	0.869003615287148	0.869447395151934
5755	0.37037037037037	0.537037037037037	0.959307449390743	0.959497324198999
5765	0.0925925925925926	0.814814814814815	0.507735939776758	0.508814057461601
5775	0.212962962962963	0.685185185185185	0.841492425497487	0.842000949989211
5785	0.314814814814815	0.564814814814815	0.940072937744772	0.940328381714164
5795	0.037037037037037	0.842592592592593	0.244351753344221	0.245296644920687
5805	0.12962962962963	0.740740740740741	0.670696001027618	0.671518992099487
5815	0.203703703703704	0.648148148148148	0.841649544927753	0.842145958023461
5825	0.268518518518519	0.574074074074074	0.914639762393506	0.914979117531887
5835	0.314814814814815	0.509259259259259	0.94279258724165	0.943075138038371
5845	0.037037037037037	0.787037037037037	0.271768482443109	0.272578304695916
5855	0.0740740740740741	0.740740740740741	0.482814224084151	0.483703862061459
5865	0.0925925925925926	0.703703703703704	0.571205835458069	0.572017340239922
5875	0.101851851851852	0.685185185185185	0.609872268410294	0.610663912036004
5885	0.0925925925925926	0.675925925925926	0.578633044619119	0.579412414042727
5895	0.0740740740740741	0.685185185185185	0.499344363441794	0.500153271334746
5905	0.037037037037037	0.703703703703704	0.292538251600983	0.293225995070975
5915	0.203703703703704	0.509259259259259	0.85686397689627	0.857479718249528
5925	0.12962962962963	0.574074074074074	0.712958670220246	0.713706140813381

5935	0.037037037037037	0.648148148148148	0.299185954035742	0.299839270392546
5945	0.0925925925925926	0.564814814814815	0.592996461383652	0.593833883949959
5955	0.0925925925925926	0.537037037037037	0.594095610214449	0.595039006552709
5965	0.037037037037037	0.564814814814815	0.303704670001206	0.304409533205853
5975	0.037037037037037	0.509259259259259	0.304258758033672	0.30529873194402
6650	0.314814814814815	0.490740740740741	0.942792946874049	0.943075138038371
6660	0.037037037037037	0.490740740740741	0.30426426296009	0.30529873194402
6670	0.240740740740741	0.481481481481481	0.896524327630125	0.896985635104925
6680	0.425925925925926	0.462962962962963	0.968347912374099	0.968495440560636
6690	0.148148148148148	0.462962962962963	0.760614218328154	0.761377930705403
6700	0.324074074074074	0.453703703703704	0.945117065542375	0.945363670613312
6710	0.0462962962962963	0.453703703703704	0.364155584654503	0.365002638053584
6720	0.203703703703704	0.435185185185185	0.854471102371556	0.854982263122166
6730	0.351851851851852	0.425925925925926	0.951814021440671	0.952022872541603
6740	0.0740740740740741	0.425925925925926	0.513255917653493	0.514091813857838
6750	0.203703703703704	0.407407407407407	0.851648782080736	0.852139688104298
6760	0.324074074074074	0.398148148148148	0.939219527926808	0.93946634827747
6770	0.0462962962962963	0.398148148148148	0.361994424343352	0.362725696092573
6780	0.148148148148148	0.37962962962963	0.752960555580984	0.753603027360782
6790	0.240740740740741	0.37037037037037	0.884973324434193	0.885374628976708
6800	0.314814814814815	0.351851851851852	0.925928205256199	0.926210729779986
6810	0.037037037037037	0.351851851851852	0.299187384616566	0.299839270392546
6820	0.101851851851852	0.342592592592593	0.616330352072537	0.617098369907053
6830	0.148148148148148	0.324074074074074	0.740716203755213	0.741383035348618
6840	0.185185185185185	0.314814814814815	0.805119361214492	0.805700570778818
6850	0.203703703703704	0.296296296296296	0.823019406550714	0.823571529550279
6860	0.212962962962963	0.287037037037037	0.829574765352078	0.830120369370689
6870	0.203703703703704	0.268518518518519	0.808779439473309	0.809377576662876
6880	0.185185185185185	0.259259259259259	0.778533271460601	0.779202096810843
6890	0.148148148148148	0.240740740740741	0.701750903274474	0.702560273914366
6900	0.101851851851852	0.231481481481481	0.574427958413281	0.575370761765424
6910	0.037037037037037	0.212962962962963	0.271769133979786	0.272578304695916
6920	0.148148148148148	0.185185185185185	0.650081119490984	0.651049410068577
6930	0.0462962962962963	0.175925925925926	0.305655723440578	0.306648866804961
6940	0.0740740740740741	0.148148148148148	0.402947460763337	0.404150752843076
6950	0.0462962962962963	0.12037037037037	0.258137398707544	0.259297425015467
6960	0.037037037037037	0.0740740740740741	0.160764710029738	0.161913036784921
7600	0.435185185185185	0.425925925925926	0.966380758296121	0.966533962853751
7610	0.157407407407407	0.148148148148148	0.611225164455337	0.612284382302494
7620	0.37037037037037	0.351851851851852	0.942947550474272	0.943177799836758
7630	0.0925925925925926	0.0740740740740741	0.314497745188564	0.315855572864409
7640	0.296296296296296	0.259259259259259	0.875563540071131	0.875992131881589
7650	0.481481481481481	0.435185185185185	0.969969381696321	0.970111654117582
7660	0.203703703703704	0.157407407407407	0.688066903927041	0.688954378011347
7670	0.37962962962963	0.314814814814815	0.932349786923696	0.932614913545918
7680	0.101851851851852	0.037037037037037	0.196587558594016	0.197688297007127
7690	0.259259259259259	0.185185185185185	0.778533040355818	0.779202096810843
7700	0.407407407407407	0.314814814814815	0.936946003953803	0.937202054909801
7710	0.12962962962963	0.037037037037037	0.223756830079504	0.22478149917224
7720	0.259259259259259	0.157407407407407	0.732058601845163	0.732806553782733
7730	0.37962962962963	0.259259259259259	0.901585781368879	0.901942387169459
7740	0.481481481481481	0.351851851851852	0.955328849230457	0.955546850016747
7750	0.203703703703704	0.0740740740740741	0.453735079112602	0.454758342031754
7760	0.296296296296296	0.148148148148148	0.731219485618036	0.731920721235779
7770	0.37037037037037	0.212962962962963	0.856956556115877	0.857425397846482
7780	0.435185185185185	0.259259259259259	0.909041275368996	0.909402169113725
7790	0.481481481481481	0.296296296296296	0.934157651451739	0.934471243764278
7800	0.203703703703704	0.0185185185185185	0.146288987105194	0.146930535661403
7810	0.240740740740741	0.037037037037037	0.280704603373111	0.281463455588537
7820	0.259259259259259	0.0462962962962963	0.341779254556394	0.342590961958565
7830	0.268518518518519	0.037037037037037	0.287453877401954	0.288172352721692
7840	0.259259259259259	0.0185185185185185	0.155697937274411	0.156282713224461
7850	0.481481481481481	0.212962962962963	0.868109913832428	0.868669871341719
7860	0.435185185185185	0.148148148148148	0.759140061332267	0.759835924646264
7870	0.37037037037037	0.0740740740740741	0.508533605953991	0.509315479589315
7880	0.481481481481481	0.157407407407407	0.780927322243709	0.781726943461326
7890	0.37962962962963	0.037037037037037	0.301261118140868	0.301912476549738
7900	0.407407407407407	0.037037037037037	0.302732694109106	0.303397457316545
7910	0.481481481481481	0.0740740740740741	0.514892654202472	0.515994759705509
7920	0.481481481481481	0.0185185185185185	0.166013763072047	0.166715768430508

TABLE-3a

DOI: 10.18535/ijecs/v5i12.75

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 3: (Output for 1/10th of the FEM MODEL)

Consider the boundary value problem: Find u such that

$-\epsilon\ \ \Delta u - \frac{\partial u}{\partial x} - \frac{\partial u}{\partial y} + 2u = f$ **in** $\Omega = [0,1]^2$ **; u = 0, on** $\partial\Omega$

Where, $(\epsilon = 10^{-2})$**, and** f **is chosen appropriately such that the exact solution is**

u(x,y)=(1 - $e^{-\frac{x}{\epsilon}}$$)(1-x)$ **(1 -** $e^{-\frac{y}{\epsilon}}$$)(1-y)$

ALL QUADRILATERAL FEM MODEL : number of nodes=2481,elements=2400 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
238	0.516666666666667	0.466666666666667	0.257592315251024	0.257777777777778
248	0.533333333333333	0.433333333333333	0.264167697662717	0.264444444444444
258	0.566666666666667	0.366666666666667	0.27423285436389	0.274444444444444
268	0.583333333333333	0.283333333333333	0.29835384671141	0.298611111110963
278	0.616666666666667	0.166666666666667	0.319298344553341	0.319444425987748
288	0.633333333333333	0.0333333333333333	0.360769182487386	0.341799995691363
298	0.683333333333333	0.183333333333333	0.258447033681316	0.258611108288951
308	0.733333333333333	0.233333333333333	0.204283608445266	0.204444444429412
318	0.783333333333333	0.183333333333333	0.176849018406885	0.176944442513493
328	0.833333333333333	0.0333333333333333	0.161380192668087	0.155363634405165
547	0.533333333333333	0.483333333333333	0.240834099803124	0.241111111111111
557	0.566666666666667	0.466666666666667	0.230905931411066	0.231111111111111
567	0.633333333333333	0.433333333333333	0.207502335757033	0.207777777777778
577	0.716666666666667	0.366666666666667	0.17922538871668	0.179444444444444
587	0.833333333333333	0.283333333333333	0.119339642561103	0.119444444444385
597	0.966666666666667	0.166666666666667	0.0278069520357592	0.0277777761728476
607	0.816666666666667	0.366666666666667	0.115951500830114	0.116111111111111
617	0.766666666666667	0.466666666666667	0.124206367615493	0.124444444444444
627	0.816666666666667	0.466666666666667	0.0975337591233789	0.0977777777777778
637	0.966666666666667	0.366666666666667	0.0211385366648937	0.0211111111111111
856	0.533333333333333	0.516666666666667	0.225588346196218	0.225555555555556
866	0.566666666666667	0.533333333333333	0.202327732893651	0.202222222222222
876	0.633333333333333	0.566666666666667	0.158957631935969	0.158888888888889
886	0.716666666666667	0.583333333333333	0.118116355699735	0.118055555555556
896	0.833333333333333	0.616666666666667	0.0639206898776725	0.0638888888888889
906	0.966666666666667	0.633333333333333	0.0122202827097577	0.0122222222222222
916	0.816666666666667	0.683333333333333	0.0580931246980537	0.0580555555555556
926	0.766666666666667	0.733333333333333	0.0622661871845632	0.0622222222222222
936	0.816666666666667	0.783333333333333	0.03975457285716	0.0397222222222222
946	0.966666666666667	0.833333333333333	0.0055533110762653	0.00555555555555555
1165	0.516666666666667	0.533333333333333	0.225590582049829	0.225555555555556
1175	0.533333333333333	0.566666666666667	0.202325036729695	0.202222222222222
1185	0.566666666666667	0.633333333333333	0.158959558519239	0.158888888888889
1195	0.633333333333333	0.716666666666667	0.103943377852349	0.103888888888889
1205	0.716666666666667	0.833333333333333	0.0472536889087306	0.0472222222222222
1215	0.833333333333333	0.966666666666667	0.00554861685089725	0.00555555555555555
1225	0.633333333333333	0.816666666666667	0.067256367607494	0.0672222222222222
1235	0.533333333333333	0.766666666666667	0.108953126060091	0.108888888888889
1245	0.533333333333333	0.816666666666667	0.085608210794932	0.0855555555555556
1255	0.633333333333333	0.966666666666667	0.012215543439818	0.0122222222222222
1474	0.483333333333333	0.533333333333333	0.240833713727367	0.241111111111111
1484	0.466666666666667	0.566666666666667	0.230905349954989	0.231111111111111
1494	0.433333333333333	0.633333333333333	0.207501184505229	0.207777777777778
1504	0.416666666666667	0.716666666666667	0.165042986225651	0.165277777777778
1514	0.383333333333333	0.833333333333333	0.102561652518672	0.102777777777778
1524	0.366666666666667	0.966666666666667	0.0211354486232361	0.0211111111111111
1534	0.316666666666667	0.816666666666667	0.125176133238708	0.125277777777776
1544	0.266666666666667	0.766666666666667	0.171003629214828	0.171111111110662
1554	0.216666666666667	0.816666666666667	0.143574939662476	0.143611111055203
1564	0.166666666666667	0.966666666666667	0.0278038455577196	0.0277777761728476
1783	0.466666666666667	0.516666666666667	0.257592032578476	0.257777777777778
1793	0.433333333333333	0.533333333333333	0.264167117690576	0.264444444444444
1803	0.366666666666667	0.566666666666667	0.274231866889076	0.274444444444444
1813	0.283333333333333	0.633333333333333	0.262525512236987	0.262777777777648
1823	0.166666666666667	0.716666666666667	0.236028087322496	0.236111097469205
1833	0.0333333333333333	0.833333333333333	0.161927426031497	0.155363634405165
1843	0.183333333333333	0.633333333333333	0.299247182588478	0.29944444117668
1853	0.233333333333333	0.533333333333333	0.357514112138943	0.357777777751471
1863	0.183333333333333	0.533333333333333	0.380862674747614	0.381111106952138
1873	0.0333333333333333	0.633333333333333	0.361419460230421	0.341799995691363
2092	0.466666666666667	0.483333333333333	0.275485408350312	0.275555555555556

2102	0.433333333333333	0.466666666666667	0.302237180300408	0.302222222222222
2112	0.366666666666667	0.433333333333333	0.358947737555334	0.358888888888889
2122	0.283333333333333	0.416666666666667	0.418145458265905	0.418055555555348
2132	0.166666666666667	0.383333333333333	0.513988505909784	0.513888859197681
2142	0.0333333333333333	0.366666666666667	0.632236868450819	0.590381810739626
2152	0.183333333333333	0.316666666666667	0.558164181341213	0.558055549465621
2162	0.233333333333333	0.266666666666667	0.562305027963074	0.562222222179408
2172	0.183333333333333	0.216666666666667	0.639835069970244	0.639722214992044
2182	0.0333333333333333	0.166666666666667	0.832376664584111	0.776818127143224
2382	0.483333333333333	0.466666666666667	0.275482680492854	0.275555555555556
2392	0.466666666666667	0.433333333333333	0.30223987316429	0.302222222222222
2402	0.433333333333333	0.366666666666667	0.358945508769378	0.358888888888889
2412	0.366666666666667	0.283333333333333	0.453968835010081	0.453888888888664
2422	0.283333333333333	0.166666666666667	0.597310441923865	0.597222187715928
2432	0.166666666666667	0.0333333333333333	0.831176073682469	0.776818127143224
2442	0.366666666666667	0.183333333333333	0.517338118618267	0.517222216577901
2452	0.466666666666667	0.233333333333333	0.409000219679818	0.408888888858823
2462	0.466666666666667	0.183333333333333	0.435681250690365	0.435555550802443
2472	0.366666666666667	0.0333333333333333	0.631160013634438	0.590381810739626

TABLE-3b

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 3: (Output for 1/10th of the FEM MODEL)

Consider the boundary value problem: Find u such that

$-\epsilon \ \ \Delta u - \frac{\partial u}{\partial x} - \frac{\partial u}{\partial y} + 2u = f$ **in** $\Omega = [0,1]^2$ **; u = 0, on** $\partial\Omega$

Where, $(\epsilon = 10^{-2})$**, and** f **is chosen appropriately such that the exact solution is**

u(x,y)=(1 - $e^{-\frac{x}{\epsilon}}$$)(1-x)$ **(1 -** $e^{-\frac{y}{\epsilon}}$$)(1-y)$

ALL QUADRILATERAL FEM MODEL : number of nodes=5521,elements=5400 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
503	0.511111111111111	0.477777777777778	0.25523123896356	0.255308641975309
513	0.511111111111111	0.144444444444444	0.418232505101849	0.418271381932622
523	0.522222222222222	0.288888888888889	0.33965125035358	0.339753086419657
533	0.544444444444444	0.411111111111111	0.268174846150564	0.268271604938272
543	0.544444444444444	0.0777777777777778	0.419979864479358	0.419947449377025
553	0.555555555555556	0.188888888888889	0.360393307405693	0.360493824903358
563	0.577777777777778	0.277777777777778	0.304843973298188	0.304938271604675
573	0.588888888888889	0.355555555555556	0.264822802996052	0.264938271604938
583	0.588888888888889	0.0222222222222222	0.371883100785093	0.358414039060433
593	0.611111111111111	0.0777777777777778	0.358518366925895	0.358491725077948
603	0.622222222222222	0.122222222222222	0.331530968052642	0.331603306808256
613	0.644444444444444	0.144444444444444	0.304141402672138	0.30419736867827
623	0.655555555555556	0.155555555555556	0.290791685679022	0.290864146480561
633	0.677777777777778	0.144444444444444	0.27563224108761	0.275678865364682
643	0.688888888888889	0.122222222222222	0.273030033862353	0.273085076195034
653	0.711111111111111	0.0777777777777778	0.266327052534477	0.266308138629333
663	0.722222222222222	0.0222222222222222	0.25026224736025	0.242171648013806
673	0.755555555555556	0.188888888888889	0.198221630051141	0.198271603696847
683	0.777777777777778	0.0777777777777778	0.204862653729696	0.204852414330256
693	0.811111111111111	0.144444444444444	0.161590580143041	0.161604852110331
703	0.844444444444444	0.144444444444444	0.133077400319627	0.133086348796743
713	0.877777777777778	0.0777777777777778	0.11266440790785	0.112668827881641
723	0.922222222222222	0.0222222222222222	0.0681227589916811	0.0678080614438656
1197	0.688888888888889	0.322222222222222	0.210752803297259	0.210864197530862
1207	0.544444444444444	0.477777777777778	0.237804631400252	0.237901234567901
1217	0.877777777777778	0.144444444444444	0.104563573373902	0.104567845483155
1227	0.755555555555556	0.288888888888889	0.173744360327236	0.173827160493778
1237	0.644444444444444	0.411111111111111	0.209279715241248	0.209382716049383
1247	0.977777777777778	0.0777777777777778	0.0204884766892738	0.0204852414330257
1257	0.888888888888889	0.188888888888889	0.0901035597569351	0.0901234562258396

1267	0.811111111111111	0.277777777777778	0.136382862471838	0.136419753086302
1277	0.755555555555556	0.355555555555556	0.157421173829398	0.157530864197531
1287	0.711111111111111	0.411111111111111	0.170016737615249	0.170123456790123
1297	0.688888888888889	0.455555555555556	0.169259351677047	0.169382716049383
1307	0.677777777777778	0.477777777777778	0.168167463350567	0.168271604938272
1317	0.688888888888889	0.488888888888889	0.158887965266879	0.159012345679012
1327	0.711111111111111	0.477777777777778	0.150757404495778	0.150864197530864
1337	0.755555555555556	0.455555555555556	0.132958099272582	0.133086419753086
1347	0.811111111111111	0.411111111111111	0.111133288268765	0.111234567901235
1357	0.888888888888889	0.355555555555556	0.0715617355249924	0.0716049382716049
1367	0.977777777777778	0.277777777777778	0.0160596082488435	0.0160493827160355
1377	0.877777777777778	0.411111111111111	0.0719046908114308	0.0719753086419753
1387	0.844444444444444	0.477777777777778	0.0811183756553351	0.0812345679012346
1397	0.877777777777778	0.477777777777778	0.0637125711569517	0.0638271604938272
1407	0.977777777777778	0.411111111111111	0.013093976725036	0.0130864197530864
1881	0.522222222222222	0.511111111111111	0.233557023586119	0.23358024691358
1891	0.855555555555556	0.511111111111111	0.0705903822175622	0.0706172839506173
1901	0.711111111111111	0.522222222222222	0.138045247099975	0.138024691358025
1911	0.588888888888889	0.544444444444444	0.187319219233695	0.187283950617284
1921	0.922222222222222	0.544444444444444	0.0354373493885597	0.0354320987654321
1931	0.811111111111111	0.555555555555556	0.0839696492949595	0.0839506172839506
1941	0.722222222222222	0.577777777777778	0.117310965299342	0.117283950617284
1951	0.644444444444444	0.588888888888889	0.146205235322841	0.146172839506173
1961	0.977777777777778	0.588888888888889	0.00913459932492401	0.00913580246913582
1971	0.922222222222222	0.611111111111111	0.0302522922862042	0.0302469135802469
1981	0.877777777777778	0.622222222222222	0.046184273329443	0.0461728395061728
1991	0.855555555555556	0.644444444444444	0.0513717173411088	0.051358024691358
2001	0.844444444444444	0.655555555555556	0.0535955367919671	0.0535802469135802
2011	0.855555555555556	0.677777777777778	0.0465568256556887	0.0465432098765432
2021	0.877777777777778	0.688888888888889	0.0380360513593615	0.0380246913580247
2031	0.922222222222222	0.711111111111111	0.022474489047505	0.0224691358024691
2041	0.977777777777778	0.722222222222222	0.00617162856548692	0.00617283950617285
2051	0.811111111111111	0.755555555555556	0.0461901544567427	0.0461728395061728
2061	0.922222222222222	0.777777777777778	0.0172891889247812	0.0172839506172839
2071	0.855555555555556	0.811111111111111	0.0272957413553482	0.027283950617284
2081	0.855555555555556	0.844444444444444	0.0224795014759494	0.0224691358024691
2091	0.922222222222222	0.877777777777778	0.00951005598298989	0.00950617283950617
2101	0.977777777777778	0.922222222222222	0.00172668958490112	0.0017283950617284
2575	0.677777777777778	0.688888888888889	0.100274209260213	0.100246913580247
2585	0.522222222222222	0.544444444444444	0.21768605862417	0.217654320987654
2595	0.855555555555556	0.877777777777778	0.0176625517418633	0.0176543209876543
2605	0.711111111111111	0.755555555555556	0.0706396596778806	0.0706172839506173
2615	0.588888888888889	0.644444444444444	0.146204761332343	0.146172839506173
2625	0.922222222222222	0.977777777777778	0.00172588864752676	0.0017283950617284
2635	0.811111111111111	0.888888888888889	0.0209964637680142	0.0209876543209877
2645	0.722222222222222	0.811111111111111	0.052486576817852	0.0524691358024691
2655	0.644444444444444	0.755555555555556	0.0869372727515968	0.0869135802469136
2665	0.588888888888889	0.711111111111111	0.118792962389117	0.118765432098765
2675	0.544444444444444	0.688888888888889	0.14175780017414	0.141728395061728
2685	0.522222222222222	0.677777777777778	0.153973178184317	0.153950617283951
2695	0.511111111111111	0.688888888888889	0.152087311325211	0.152098765432099
2705	0.522222222222222	0.711111111111111	0.138044596383131	0.138024691358025
2715	0.544444444444444	0.755555555555556	0.111381749235068	0.111358024691358
2725	0.588888888888889	0.811111111111111	0.0776726267154725	0.0776543209876543
2735	0.644444444444444	0.888888888888889	0.0395157637242186	0.0395061728395062
2745	0.722222222222222	0.977777777777778	0.00617075908936981	0.00617283950617285
2755	0.588888888888889	0.877777777777778	0.0502575503093606	0.0502469135802469
2765	0.522222222222222	0.844444444444444	0.0743279179798492	0.074320987654321
2775	0.522222222222222	0.877777777777778	0.0583985413936234	0.0583950617283951
2785	0.588888888888889	0.977777777777778	0.0091337297018018	0.00913580246913582
3259	0.488888888888889	0.522222222222222	0.244077936981311	0.244197530864197
3269	0.488888888888889	0.855555555555556	0.0736891530408501	0.0738271604938272
3279	0.477777777777778	0.711111111111111	0.150757099326712	0.150864197530864
3289	0.455555555555556	0.588888888888889	0.223710438380586	0.223827160493827
3299	0.455555555555556	0.922222222222222	0.0422530379561816	0.0423456790123457
3309	0.444444444444444	0.811111111111111	0.104826215266227	0.104938271604938
3319	0.422222222222222	0.722222222222222	0.160368908392855	0.160493827160494
3329	0.411111111111111	0.644444444444444	0.209279447323853	0.209382716049383
3339	0.411111111111111	0.977777777777778	0.0130933621372738	0.0130864197530864
3349	0.388888888888889	0.922222222222222	0.047494167221633	0.0475308641975308
3359	0.377777777777778	0.877777777777778	0.0760026932782244	0.0760493827160494
3369	0.355555555555556	0.855555555555556	0.0930213069707305	0.0930864197530864
3379	0.344444444444444	0.844444444444444	0.101926187343592	0.101975308641975
3389	0.322222222222222	0.855555555555556	0.0978523821000392	0.0979012345679003

3399	0.311111111111111	0.877777777777778	0.0841799894293416	0.0841975308641949
3409	0.288888888888889	0.922222222222222	0.0552912435108406	0.0553086419752929
3419	0.277777777777778	0.977777777777778	0.0160589929567578	0.0160493827160355
3429	0.244444444444444	0.811111111111111	0.14268856331	0.142716049379262
3439	0.222222222222222	0.922222222222222	0.0604780024721823	0.0604938271469817
3449	0.188888888888889	0.855555555555556	0.117134749535188	0.117160493093592
3459	0.155555555555556	0.855555555555556	0.121950569377239	0.121975287233784
3469	0.122222222222222	0.922222222222222	0.0682557250170963	0.0682712690487585
3479	0.0777777777777778	0.977777777777778	0.0204871050301192	0.0204852414330257
3953	0.311111111111111	0.677777777777778	0.221881403474212	0.221975308641968
3963	0.455555555555556	0.522222222222222	0.260013370118646	0.260123456790124
3973	0.122222222222222	0.855555555555556	0.126765386319761	0.12678949966198
3983	0.244444444444444	0.711111111111111	0.218205828558616	0.218271604932988
3993	0.355555555555556	0.588888888888889	0.264822566382529	0.264938271604938
4003	0.0222222222222222	0.922222222222222	0.0682138848004449	0.0678080614438656
4013	0.111111111111111	0.811111111111111	0.167885892634916	0.167898725227112
4023	0.188888888888889	0.722222222222222	0.225247499473402	0.225308640564599
4033	0.244444444444444	0.644444444444444	0.268558829056273	0.268641975302139
4043	0.288888888888889	0.588888888888889	0.29223356672699	0.292345679012263
4053	0.311111111111111	0.544444444444444	0.313731805863536	0.313827160493818
4063	0.322222222222222	0.522222222222222	0.323724200370016	0.323827160493824
4073	0.311111111111111	0.511111111111111	0.336737827196247	0.33679012345678
4083	0.288888888888889	0.522222222222222	0.339651047848635	0.339753086419657
4093	0.244444444444444	0.544444444444444	0.344106047399352	0.344197530855866
4103	0.188888888888889	0.588888888888889	0.333360304567475	0.333456788035607
4113	0.111111111111111	0.644444444444444	0.315998393766844	0.316044659251034
4123	0.0222222222222222	0.722222222222222	0.250390383455072	0.242171648013806
4133	0.122222222222222	0.588888888888889	0.360781998031134	0.360862422114867
4143	0.155555555555556	0.522222222222222	0.403366902383957	0.403456719311746
4153	0.122222222222222	0.522222222222222	0.419297583631841	0.419380652728088
4163	0.0222222222222222	0.588888888888889	0.372025699154446	0.358414039060433
4637	0.477777777777778	0.488888888888889	0.266876116545443	0.266913580246914
4647	0.144444444444444	0.488888888888889	0.437336146127171	0.437283717475013
4657	0.288888888888889	0.477777777777778	0.371403838987405	0.371358024691252
4667	0.411111111111111	0.455555555555556	0.320633500231802	0.320617283950617
4677	0.0777777777777778	0.455555555555556	0.502060198270059	0.501888415109128
4687	0.188888888888889	0.444444444444444	0.450665605785786	0.450617281129198
4697	0.277777777777778	0.422222222222222	0.417326976536479	0.417283950616924
4707	0.355555555555556	0.411111111111111	0.379532374881984	0.379506172839506
4717	0.0222222222222222	0.411111111111111	0.538345091083872	0.513403893789269
4727	0.0777777777777778	0.388888888888889	0.563535618873025	0.563344139408205
4737	0.122222222222222	0.377777777777778	0.546224236202576	0.546170152390069
4747	0.144444444444444	0.355555555555556	0.551408360048367	0.551357730729365
4757	0.155555555555556	0.344444444444444	0.553629386537301	0.553580149753325
4767	0.144444444444444	0.322222222222222	0.579926071758255	0.579876234042947
4777	0.122222222222222	0.311111111111111	0.604741478331444	0.604688383003272
4787	0.0777777777777778	0.288888888888889	0.655743563321757	0.655527725856634
4797	0.0222222222222222	0.277777777777778	0.66027142087399	0.629646284835352
4807	0.188888888888889	0.244444444444444	0.612881825402989	0.612839502320875
4817	0.0777777777777778	0.222222222222222	0.717216434236001	0.716983449995749
4827	0.144444444444444	0.188888888888889	0.69399629492528	0.693950242952321
4837	0.144444444444444	0.155555555555556	0.722515050316214	0.722468623808593
4847	0.0777777777777778	0.122222222222222	0.809425825861814	0.809163055583089
4857	0.0222222222222222	0.0777777777777778	0.843457991154862	0.803673037803036
5302	0.322222222222222	0.311111111111111	0.466943645408889	0.466913580246895
5312	0.477777777777778	0.455555555555556	0.284316935956153	0.284320987654321
5322	0.144444444444444	0.122222222222222	0.751035342217563	0.750983559141806
5332	0.288888888888889	0.244444444444444	0.537321450189411	0.537283950604126
5342	0.411111111111111	0.355555555555556	0.379532791262581	0.379506172839506
5352	0.0777777777777778	0.0222222222222222	0.843068605962914	0.803673037803036
5362	0.188888888888889	0.111111111111111	0.721031718773201	0.720976874402218
5372	0.277777777777778	0.188888888888889	0.585846213959429	0.585802465467452
5382	0.355555555555556	0.244444444444444	0.486955054338288	0.486913580235127
5392	0.411111111111111	0.288888888888889	0.418806346869762	0.418765432098646
5402	0.455555555555556	0.311111111111111	0.375103592434207	0.37506172839505
5412	0.477777777777778	0.322222222222222	0.353995428994546	0.353950617283947
5422	0.488888888888889	0.311111111111111	0.352139989937697	0.352098765432088
5432	0.477777777777778	0.288888888888889	0.371404377025781	0.371358024691252
5442	0.455555555555556	0.244444444444444	0.411403658955349	0.411358024681401
5452	0.411111111111111	0.188888888888889	0.477703062087332	0.47765431799695
5462	0.355555555555556	0.111111111111111	0.572888432565665	0.572830944892499
5472	0.277777777777778	0.0222222222222222	0.659923770903207	0.629646284835352
5482	0.411111111111111	0.122222222222222	0.516966063326502	0.516911037083458
5492	0.477777777777778	0.155555555555556	0.441042121807768	0.440987576922141

DOI: 10.18535/ijecs/v5i12.75

5502	0.477777777777778	0.122222222222222	0.458453000147723	0.458392806470236
5512	0.411111111111111	0.0222222222222222	0.538018360235135	0.513403893789269

TABLE-3c

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 3: (Output for 1/10th of the FEM MODEL)

Consider the boundary value problem: Find u such that

$-\epsilon \ \ \Delta u - \frac{\partial u}{\partial x} - \frac{\partial u}{\partial y} + 2u = f$ in $\Omega = [0,1]^2$; u = 0, on $\partial\Omega$

Where, ($\epsilon = 10^{-2}$), and f is chosen appropriately such that the exact solution is

u(x,y)=(1 - $e^{-\frac{x}{\epsilon}}$)$(1-x)$ (1 - $e^{-\frac{y}{\epsilon}}$)$(1-y)$

ALL QUADRILATERAL FEM MODEL : number of nodes=7921,elements=7776 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
710	0.509259259259259	0.481481481481481	0.25440503836116	0.254458161865569
720	0.509259259259259	0.203703703703704	0.390750932321388	0.390775033737409
730	0.518518518518518	0.407407407407407	0.285256225491632	0.285322359396433
740	0.518518518518518	0.12962962962963	0.419014219094171	0.419066232393197
750	0.537037037037037	0.314814814814815	0.317149479639952	0.317215363511653
760	0.537037037037037	0.037037037037037	0.43829070623402	0.434834785534642
770	0.546296296296296	0.212962962962963	0.357009948152644	0.357081618454365
780	0.564814814814815	0.37037037037037	0.273936725469395	0.27400548696845
790	0.564814814814815	0.0925925925925926	0.3948260385222	0.394852656875535
800	0.574074074074074	0.240740740740741	0.323314682755929	0.323388203006496
810	0.592592592592593	0.37037037037037	0.256445947974914	0.256515775034294
820	0.592592592592593	0.0925925925925926	0.369625223508314	0.369649295798373
830	0.601851851851852	0.212962962962963	0.313289028220197	0.313357338643626
840	0.62037037037037	0.314814814814815	0.260048078742194	0.260116598079555
850	0.62037037037037	0.037037037037037	0.359219765820389	0.356564524138407
860	0.62962962962963	0.12962962962963	0.32230886767496	0.32235864030246
870	0.648148148148148	0.203703703703704	0.280127880048722	0.280178326075878
880	0.657407407407407	0.268518518518519	0.250529089542101	0.250600137173665
890	0.675925925925926	0.314814814814815	0.221984070094686	0.222050754458157
900	0.675925925925926	0.037037037037037	0.306510015061868	0.30438434987425
910	0.685185185185185	0.0740740740740741	0.291348355800096	0.29131833661265
920	0.703703703703704	0.0925925925925926	0.268823515932338	0.268835851489726
930	0.712962962962963	0.101851851851852	0.257763888030487	0.257792057669441
940	0.731481481481482	0.0925925925925926	0.243622470424573	0.243632490412564
950	0.740740740740741	0.0740740740740741	0.239928295449222	0.239909218386888
960	0.759259259259259	0.037037037037037	0.227443814745586	0.226114088478014
970	0.787037037037037	0.203703703703704	0.169562175962803	0.169581618414347
980	0.796296296296296	0.12962962962963	0.177275326905302	0.177297252166353
990	0.814814814814815	0.037037037037037	0.174735019267653	0.173933914213857
1000	0.842592592592593	0.0925925925925926	0.142814506932847	0.142819046103917
1010	0.87037037037037	0.0925925925925926	0.117612213216827	0.117615685026755
1020	0.898148148148148	0.037037037037037	0.0957121529606258	0.0956636528176213
1030	0.953703703703704	0.037037037037037	0.0432121077563307	0.0434834785534642
1705	0.685185185185185	0.324074074074074	0.212714278184006	0.212791495198901
1715	0.962962962962963	0.0462962962962963	0.0348783038882523	0.0349776706507348
1725	0.759259259259259	0.259259259259259	0.178288652385665	0.17832647462179
1735	0.574074074074074	0.462962962962963	0.228658443821153	0.228737997256516
1745	0.851851851851852	0.185185185185185	0.120696149356168	0.120713304803868
1755	0.675925925925926	0.37037037037037	0.203974473540595	0.204046639231824
1765	0.953703703703704	0.0925925925925926	0.0420075586840828	0.0420056017952696
1775	0.796296296296296	0.268518518518519	0.148966449860654	0.149005486968125
1785	0.648148148148148	0.425925925925926	0.201916329727499	0.2019890260631
1795	0.925925925925926	0.148148148148148	0.0631007255166177	0.063100113944812
1805	0.796296296296296	0.296296296296296	0.143300330894759	0.143347050754439
1815	0.675925925925926	0.425925925925926	0.185968528085427	0.186042524005487
1825	0.953703703703704	0.148148148148148	0.0394408888525698	0.0394375712155075
1835	0.851851851851852	0.268518518518519	0.108344369008795	0.108367626885909
1845	0.759259259259259	0.37037037037037	0.151508900416054	0.151577503429355
1855	0.685185185185185	0.462962962962963	0.16898255611911	0.169067215363512
1865	0.962962962962963	0.185185185185185	0.0301726475832784	0.03017832620096 7
1875	0.898148148148148	0.259259259259259	0.0754422358474167	0.0754458161861418
1885	0.851851851851852	0.324074074074074	0.100102085437546	0.100137174211247
1895	0.814814814814815	0.37037037037037	0.116541282145373	0.116598079561043
1905	0.796296296296296	0.407407407407407	0.120630979276914	0.120713305898491
1915	0.787037037037037	0.425925925925926	0.122179933864627	0.122256515775034
1925	0.796296296296296	0.435185185185185	0.114967400272981	0.115054869684499
1935	0.814814814814815	0.425925925925926	0.106235276105324	0.106310013717421

1945	0.851851851851852	0.407407407407407	0.0877218169211871	0.0877914951989026
1955	0.898148148148148	0.37037037037037	0.0641086068685998	0.0641289437585734
1965	0.962962962962963	0.324074074074074	0.0250281801651893	0.0250342935528119
1975	0.851851851851852	0.462962962962963	0.0794711530614926	0.0795610425240055
1985	0.953703703703704	0.37037037037037	0.0291500466494157	0.0291495198902606
1995	0.925925925925926	0.425925925925926	0.0424942274667237	0.0425240054869684
2005	0.953703703703704	0.425925925925926	0.0265673585237363	0.0265775034293553
2015	0.962962962962963	0.462962962962963	0.0198570115585754	0.0198902606310014
2690	0.574074074074074	0.509259259259259	0.209007159843263	0.209019204389575
2700	0.851851851851852	0.509259259259259	0.0726749319657384	0.0727023319615912
2710	0.648148148148148	0.518518518518518	0.169421277985341	0.169410150891632
2720	0.925925925925926	0.518518518518518	0.0356597462793296	0.0356652949245542
2730	0.740740740740741	0.537037037037037	0.120044205167199	0.12002743484225
2740	0.564814814814815	0.546296296296296	0.197470657812084	0.197445130315501
2750	0.842592592592593	0.546296296296296	0.0714269473674001	0.0714163237311386
2760	0.685185185185185	0.564814814814815	0.137023807053669	0.137002743484225
2770	0.962962962962963	0.564814814814815	0.0161180807899671	0.0161179698216736
2780	0.814814814814815	0.574074074074074	0.078888329807211	0.0788751714677641
2790	0.685185185185185	0.592592592592593	0.128278722915195	0.128257887517147
2800	0.962962962962963	0.592592592592593	0.0150892689695738	0.0150891632373114
2810	0.842592592592593	0.601851851851852	0.0626824997732493	0.0626714677640604
2820	0.740740740740741	0.62037037037037	0.0984402420982626	0.0984224965706447
2830	0.648148148148148	0.62962962962963	0.130337255043972	0.130315500685871
2840	0.925925925925926	0.62962962962963	0.0274386977309823	0.0274348422496571
2850	0.851851851851852	0.648148148148148	0.0521363111509286	0.0521262002743484
2860	0.787037037037037	0.657407407407407	0.0729744033577412	0.0729595336076817
2870	0.740740740740741	0.675925925925926	0.0840362842240259	0.0840192043895748
2880	0.703703703703704	0.685185185185185	0.0932969962753676	0.093278463648834
2890	0.981481481481482	0.685185185185185	0.00582884386779596	0.00582990397805212
2900	0.962962962962963	0.703703703703704	0.0109740410616473	0.0109739368998628
2910	0.953703703703704	0.712962962962963	0.013289937554968	0.0132887517146776
2920	0.962962962962963	0.731481481481482	0.00994523283960039	0.0099451303155007
2930	0.981481481481482	0.740740740740741	0.0048000365774583	0.00480109739368998
2940	0.787037037037037	0.768518518518518	0.0493100670096416	0.0492969821673525
2950	0.851851851851852	0.787037037037037	0.0315593181983628	0.0315500685871056
2960	0.925925925925926	0.796296296296296	0.0150928989003414	0.0150891632373114
2970	0.842592592592593	0.824074074074074	0.0277010561073429	0.0276920438957476
2980	0.962962962962963	0.842592592592593	0.00582994336467349	0.00582990397805214
2990	0.962962962962963	0.87037037037037	0.00480105565318836	0.00480109739368999
3000	0.925925925925926	0.907407407407407	0.00686092581642264	0.00685871056241426
3010	0.981481481481482	0.962962962962963	0.000683706680115812	0.000685871056241427
3685	0.731481481481482	0.740740740740741	0.0696318313142142	0.0696159122085048
3695	0.518518518518518	0.537037037037037	0.222922514682865	0.222908093278464
3705	0.796296296296296	0.814814814814815	0.0377338153742582	0.0377229080932785
3715	0.592592592592593	0.62962962962963	0.150914757869533	0.150891632373114
3725	0.87037037037037	0.907407407407407	0.0120069665924376	0.012002743484225
3735	0.685185185185185	0.731481481481482	0.0845508654861793	0.0845336076817558
3745	0.509259259259259	0.574074074074074	0.209007248954238	0.209019204389575
3755	0.787037037037037	0.851851851851852	0.031559335876748	0.0315500685871056
3765	0.62962962962963	0.703703703703704	0.109758793265658	0.109739368998628
3775	0.907407407407407	0.981481481481482	0.00171309592464661	0.00171467764060356
3785	0.759259259259259	0.851851851851852	0.0356748987625423	0.0356652949245542
3795	0.62962962962963	0.731481481481482	0.0994693478654622	0.0994513031550068
3805	0.509259259259259	0.62962962962963	0.181741733767115	0.181755829903978
3815	0.787037037037037	0.907407407407407	0.0197239399289049	0.019718792866941
3825	0.685185185185185	0.814814814814815	0.0583116003119246	0.0582990397805213
3835	0.592592592592593	0.740740740740741	0.105642075028909	0.10562414266118
3845	0.518518518518518	0.675925925925926	0.156045165907815	0.156035665294925
3855	0.796296296296296	0.953703703703704	0.00943147961711781	0.00943072702331961
3865	0.731481481481482	0.907407407407407	0.0248681013112203	0.0248628257887517
3875	0.685185185185185	0.87037037037037	0.0408176839217593	0.0408093278463649
3885	0.648148148148148	0.851851851851852	0.0521362522587556	0.0521262002743484
3895	0.62962962962963	0.842592592592593	0.0583096960063849	0.0582990397805213
3905	0.62037037037037	0.851851851851852	0.0562515024003878	0.056241426611797
3915	0.62962962962963	0.87037037037037	0.0480193757066703	0.0480109739368999
3925	0.648148148148148	0.907407407407407	0.0325841923940091	0.0325788751714678
3935	0.685185185185185	0.953703703703704	0.0145755393990745	0.0145747599451303
3945	0.518518518518518	0.814814814814815	0.0891639972467507	0.0891632373113855
3955	0.592592592592593	0.907407407407407	0.0377282304064451	0.0377229080932785
3965	0.509259259259259	0.851851851851852	0.072674860169077	0.0727023319615912
3975	0.62962962962963	0.981481481481482	0.0068572453657892	0.00685871056241426
3985	0.509259259259259	0.907407407407407	0.0454090746893979	0.0454389574759945
3995	0.518518518518518	0.953703703703704	0.0222852383780761	0.0222908093278464
4670	0.490740740740741	0.62962962962963	0.188531793901513	0.188614540466392

4680	0.490740740740741	0.907407407407407	0.0470600962154168	0.0471536351165981
4690	0.481481481481481	0.703703703703704	0.153559965117578	0.15363511659808
4700	0.481481481481481	0.981481481481482	0.0095883989096942	0.00960219478737996
4710	0.462962962962963	0.796296296296296	0.109305984397712	0.109396433470508
4720	0.453703703703704	0.62037037037037	0.207318682481143	0.207390260631001
4730	0.453703703703704	0.898148148148148	0.0555750089659295	0.0556412894375857
4740	0.435185185185185	0.740740740740741	0.146346428579925	0.146433470507545
4750	0.425925925925926	0.592592592592593	0.233811719860229	0.233882030178326
4760	0.425925925925926	0.87037037037037	0.0743549107819851	0.0744170096021948
4770	0.407407407407407	0.740740740740741	0.153549875392024	0.15363511659808
4780	0.398148148148148	0.62037037037037	0.228409437110144	0.228480795610425
4790	0.398148148148148	0.898148148148148	0.0612672433929435	0.0612997256515775
4800	0.37962962962963	0.796296296296296	0.126296796167951	0.126371742112483
4810	0.37037037037037	0.703703703703704	0.186484664959141	0.186556927297668
4820	0.37037037037037	0.981481481481482	0.0116655899650039	0.0116598079561042
4830	0.351851851851852	0.907407407407407	0.0599923650721504	0.0600137174211248
4840	0.342592592592593	0.842592592592593	0.10344535579208	0.103480795610425
4850	0.324074074074074	0.796296296296296	0.137632632975668	0.137688614540465
4860	0.314814814814815	0.759259259259259	0.16489800848549	0.16495198902606
4870	0.296296296296296	0.740740740740741	0.182378599229436	0.182441700960195
4880	0.287037037037037	0.731481481481482	0.191390868468169	0.191443758573323
4890	0.268518518518519	0.740740740740741	0.189587864199661	0.189643347050341
4900	0.259259259259259	0.759259259259259	0.178288469301504	0.17832647462179
4910	0.240740740740741	0.796296296296296	0.154630714102118	0.15466392317702
4920	0.231481481481481	0.842592592592593	0.120959060882362	0.120970507533877
4930	0.212962962962963	0.907407407407407	0.072862215397997	0.0728737996845643
4940	0.203703703703704	0.981481481481482	0.0147524072169204	0.0147462276882041
4950	0.175925925925926	0.898148148148148	0.0839311115260748	0.0839334685863591
4960	0.148148148148148	0.87037037037037	0.110420344824592	0.110425199403421
4970	0.12037037037037	0.898148148148148	0.0895891703237756	0.0895913762647428
4980	0.0925925925925926	0.981481481481482	0.0168075043641424	0.0168022407181078
5655	0.481481481481481	0.509259259259259	0.254405023688797	0.254458161865569
5665	0.203703703703704	0.787037037037037	0.169561980864016	0.169581618414347
5675	0.407407407407407	0.574074074074074	0.252321193688788	0.252400548696845
5685	0.12962962962963	0.851851851851852	0.12892714005685	0.128943456120984
5695	0.314814814814815	0.648148148148148	0.241015487170059	0.241083676268856
5705	0.037037037037037	0.925925925925926	0.0694056240066608	0.0695735656855428
5715	0.212962962962963	0.740740740740741	0.204004577120012	0.20404663911678
5725	0.37037037037037	0.564814814814815	0.27393662099496	0.27400548696845
5735	0.0925925925925926	0.842592592592593	0.142814230468646	0.142819046103917
5745	0.240740740740741	0.685185185185185	0.238964718267752	0.239026063091758
5755	0.37037037037037	0.537037037037037	0.291428510410428	0.291495198902606
5765	0.0925925925925926	0.814814814814815	0.168016584075253	0.168022407181079
5775	0.212962962962963	0.685185185185185	0.247715715089583	0.247770918927519
5785	0.314814814814815	0.564814814814815	0.298114475488243	0.298182441700954
5795	0.037037037037037	0.842592592592593	0.148372297623629	0.147843827081778
5805	0.12962962962963	0.740740740740741	0.225621197751634	0.225651048211722
5815	0.203703703703704	0.648148148148148	0.280127734398817	0.280178326075878
5825	0.268518518518519	0.574074074074074	0.311481091663831	0.311556927296989
5835	0.314814814814815	0.509259259259259	0.336220127130211	0.336248285322352
5845	0.037037037037037	0.787037037037037	0.201076502788862	0.200024001345935
5855	0.0740740740740741	0.740740740740741	0.239928153833219	0.239909218386888
5865	0.0925925925925926	0.703703703703704	0.268823291043289	0.268835851489726
5875	0.101851851851852	0.685185185185185	0.282707534231484	0.282739676153581
5885	0.0925925925925926	0.675925925925926	0.294023935951352	0.294039212566888
5895	0.0740740740740741	0.685185185185185	0.291348236161618	0.29131833661265
5905	0.037037037037037	0.703703703703704	0.280141732981071	0.278294262742171
5915	0.203703703703704	0.509259259259259	0.390750819824872	0.390775033737409
5925	0.12962962962963	0.574074074074074	0.370653708058584	0.370712436347828
5935	0.037037037037037	0.648148148148148	0.332851478782493	0.330474437006328
5945	0.0925925925925926	0.564814814814815	0.394825849111048	0.394852656875535
5955	0.0925925925925926	0.537037037037037	0.420028442583196	0.420056017952697
5965	0.037037037037037	0.564814814814815	0.411914815342104	0.408744698402564
5975	0.037037037037037	0.509259259259259	0.464810258375779	0.460924872666721
6650	0.314814814814815	0.490740740740741	0.348966267083588	0.348936899862818
6660	0.037037037037037	0.490740740740741	0.483042161305295	0.478318264088107
6670	0.240740740740741	0.481481481481481	0.393723064301413	0.393689986268778
6680	0.425925925925926	0.462962962962963	0.308306878389034	0.308299039780521
6690	0.148148148148148	0.462962962962963	0.457512231790851	0.457475826099887
6700	0.324074074074074	0.453703703703704	0.369284082112924	0.369255829903975
6710	0.0462962962962963	0.453703703703704	0.51805042586108	0.515920642098338
6720	0.203703703703704	0.435185185185185	0.449793353141823	0.449759944490225
6730	0.351851851851852	0.425925925925926	0.372106378317874	0.372085048010974
6740	0.0740740740740741	0.425925925925926	0.5314308269491	0.531227554999538

6750	0.203703703703704	0.407407407407407	0.471912618767213	0.471879286022531
6760	0.324074074074074	0.398148148148148	0.406830033435262	0.406807270233193
6770	0.0462962962962963	0.398148148148148	0.57062179125851	0.56838714807444
6780	0.148148148148148	0.37962962962963	0.528498877901823	0.5284634542878
6790	0.240740740740741	0.37037037037037	0.478082272038398	0.478052126183516
6800	0.314814814814815	0.351851851851852	0.444122624528257	0.444101508916314
6810	0.037037037037037	0.351851851851852	0.614247032793074	0.608768699748499
6820	0.101851851851852	0.342592592592593	0.590476263843104	0.5904269707913
6830	0.148148148148148	0.324074074074074	0.575823190411641	0.575788539746404
6840	0.185185185185185	0.314814814814815	0.558330971703063	0.558299034717876
6850	0.203703703703704	0.296296296296296	0.560386685675301	0.56035665215168
6860	0.212962962962963	0.287037037037037	0.561157111852062	0.561128257570953
6870	0.203703703703704	0.268518518518519	0.582505250575304	0.582475993682792
6880	0.185185185185185	0.259259259259259	0.603596748150166	0.603566524016018
6890	0.148148148148148	0.240740740740741	0.646808716252184	0.646776167911649
6900	0.101851851851852	0.231481481481481	0.69026716469553	0.690217444666782
6910	0.037037037037037	0.212962962962963	0.746006528725213	0.73921913499211
6920	0.148148148148148	0.185185185185185	0.694133324655256	0.694101247098849
6930	0.0462962962962963	0.175925925925926	0.781352884604459	0.778253154165112
6940	0.0740740740740741	0.148148148148148	0.788554552033054	0.788272855937034
6950	0.0462962962962963	0.12037037037037	0.834043661009659	0.83071475942031
6960	0.037037037037037	0.0740740740740741	0.877473636573145	0.869141906282956
7600	0.435185185185185	0.425925925925926	0.324249803687483	0.324245541838134
7610	0.157407407407407	0.148148148148148	0.717796041583832	0.717763691441997
7620	0.37037037037037	0.351851851851852	0.408110340611597	0.408093278463649
7630	0.0925925925925926	0.0740740740740741	0.839952950325368	0.839602304907937
7640	0.296296296296296	0.259259259259259	0.521287356374608	0.521262002740546
7650	0.481481481481481	0.435185185185185	0.29287889747532	0.292866941015089
7660	0.203703703703704	0.157407407407407	0.670984771434162	0.670953261959984
7670	0.37962962962963	0.314814814814815	0.425091101247466	0.425068587105615
7680	0.101851851851852	0.037037037037037	0.851423357520274	0.843547659825811
7690	0.259259259259259	0.185185185185185	0.60359692169701	0.603566524016018
7700	0.407407407407407	0.314814814814815	0.406060787215019	0.406035665294916
7710	0.12962962962963	0.037037037037037	0.825082702745926	0.817487479289799
7720	0.259259259259259	0.157407407407407	0.62417489543992	0.624142570149936
7730	0.37962962962963	0.259259259259259	0.459562741544519	0.459533607679227
7740	0.481481481481481	0.351851851851852	0.336106351381213	0.336076817558299
7750	0.203703703703704	0.0740740740740741	0.737126870750469	0.736864026853893
7760	0.296296296296296	0.148148148148148	0.599484961918102	0.599451082475633
7770	0.37037037037037	0.212962962962963	0.495574091565606	0.495541837855037
7780	0.435185185185185	0.259259259259259	0.418412686168457	0.418381344304968
7790	0.481481481481481	0.296296296296296	0.364915218846466	0.36488340192039
7800	0.203703703703704	0.0185185185185185	0.684140722840747	0.658888710648219
7810	0.240740740740741	0.037037037037037	0.719663419369393	0.713129048251814
7820	0.259259259259259	0.0462962962962963	0.7022973398611	0.699553413010846
7830	0.268518518518519	0.037037037037037	0.693309024756193	0.687038961143237
7840	0.259259259259259	0.0185185185185185	0.636322236220477	0.612919731704477
7850	0.481481481481481	0.212962962962963	0.40812764259868	0.40809327823356
7860	0.435185185185185	0.148148148148148	0.48117420128645	0.481138368829191
7870	0.37037037037037	0.0740740740740741	0.582850732810473	0.582636673225299
7880	0.481481481481481	0.157407407407407	0.436937440057534	0.436899799107359
7890	0.37962962962963	0.037037037037037	0.587922422025993	0.582678612616421
7900	0.407407407407407	0.037037037037037	0.561614698713659	0.556588525484342
7910	0.481481481481481	0.0740740740740741	0.480008967500769	0.479818436773776
7920	0.481481481481481	0.0185185185185185	0.448318631933271	0.429043812195494

TABLE-4a

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 4: (Output for 1/10th of the FEM MODEL)

Consider the boundary value problem: Find u such that

$$-\Delta u + \frac{1}{10}\frac{\partial u}{\partial y} = f \quad \text{in } \Omega = [0,1]^2$$

$$\left.\begin{aligned} \mathbf{u}(\mathbf{x},\mathbf{0}) &= \mathbf{0}, \mathbf{0} \leqslant x \leqslant \mathbf{1} \\ \mathbf{u}(\mathbf{0},\mathbf{y}) &= \mathbf{0}, \mathbf{0} \leqslant y \leqslant \mathbf{1} \\ \mathbf{u}(\mathbf{1},\mathbf{y}) &= \mathbf{0}, \mathbf{0} \leqslant y \leqslant \mathbf{1} \\ \mathbf{u}(\mathbf{x},\mathbf{1}) &= \sin(\pi x), \mathbf{0} \leqslant x \leqslant \mathbf{1} \end{aligned}\right\}$$

and f is chosen appropriately such that the exact solution is u(x,y)= $\sin(\pi x)\sin(\frac{\pi y}{2})$

ALL QUADRILATERAL FEM MODEL : number of nodes=2481,elements=2400 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values

238	0.516666666666667	0.466666666666667	0.666666473596968	0.668213586118192
248	0.533333333333333	0.433333333333333	0.624275777232731	0.625872908100787
258	0.566666666666667	0.366666666666667	0.531273052017579	0.532737365365924
268	0.583333333333333	0.283333333333333	0.4146333047846	0.415841786911166
278	0.616666666666667	0.166666666666667	0.24091927381249	0.24162839451241
288	0.633333333333333	0.0333333333333333	0.047653214771025	0.0478112750971233
298	0.683333333333333	0.183333333333333	0.237557707471007	0.238195310448054
308	0.733333333333333	0.233333333333333	0.265638054732676	0.266319287321534
318	0.783333333333333	0.183333333333333	0.178310820583599	0.178736647793227
328	0.833333333333333	0.0333333333333333	0.0260847043972887	0.0261679781214719
547	0.533333333333333	0.483333333333333	0.683042107338584	0.684583697304376
557	0.566666666666667	0.466666666666667	0.652906823943825	0.654508497187474
567	0.633333333333333	0.433333333333333	0.573389958478285	0.574912784645444
577	0.716666666666667	0.366666666666667	0.422124708599413	0.42326402651375
587	0.833333333333333	0.283333333333333	0.214716214048079	0.215255548404148
597	0.966666666666667	0.166666666666667	0.0269685559451976	0.0270539570489681
607	0.816666666666667	0.366666666666667	0.295816160310539	0.2966316784621
617	0.766666666666667	0.466666666666667	0.446384173268536	0.447735768366173
627	0.816666666666667	0.466666666666667	0.363262520982004	0.364434647746309
637	0.966666666666667	0.366666666666667	0.0567399007930966	0.0569302813656986
856	0.533333333333333	0.516666666666667	0.719875243168795	0.721400694310709
866	0.566666666666667	0.533333333333333	0.725283470949367	0.726905328038456
876	0.633333333333333	0.566666666666667	0.70831415684914	0.709958163014307
886	0.716666666666667	0.583333333333333	0.615041669906852	0.616551344415731
896	0.833333333333333	0.616666666666667	0.411020966949538	0.412063094311008
906	0.966666666666667	0.633333333333333	0.0874561187739125	0.0876649456551455
916	0.816666666666667	0.683333333333333	0.477598310642895	0.478638104194905
926	0.766666666666667	0.733333333333333	0.610119813394323	0.611281226008774
936	0.816666666666667	0.783333333333333	0.512520587049964	0.513399352073571
946	0.966666666666667	0.833333333333333	0.100824320451644	0.100966742252535
1165	0.516666666666667	0.533333333333333	0.740624504510676	0.742126371321759
1175	0.533333333333333	0.566666666666667	0.771301446760142	0.772888674565986
1185	0.566666666666667	0.633333333333333	0.818713494935605	0.820343603841875
1195	0.633333333333333	0.716666666666667	0.8230465354815	0.824552686652871
1205	0.716666666666667	0.833333333333333	0.749603088939846	0.750665354967537
1215	0.833333333333333	0.966666666666667	0.498993179846775	0.499314767377287
1225	0.633333333333333	0.816666666666667	0.874660879110862	0.875925413486921
1235	0.533333333333333	0.766666666666667	0.927017761170158	0.928466175238718
1245	0.533333333333333	0.816666666666667	0.952252353721051	0.953567220037621
1255	0.633333333333333	0.966666666666667	0.911726127743788	0.912293475342785
1474	0.483333333333333	0.533333333333333	0.740624238075223	0.742126371321759
1484	0.466666666666667	0.566666666666667	0.771301208082964	0.772888674565986
1494	0.433333333333333	0.633333333333333	0.818713325240386	0.820343603841875
1504	0.416666666666667	0.716666666666667	0.870286377794477	0.871830436581993
1514	0.383333333333333	0.833333333333333	0.900542126902601	0.90176944487161
1524	0.366666666666667	0.966666666666667	0.911725216617775	0.912293475342785
1534	0.316666666666667	0.816666666666667	0.802947323340588	0.804133891599188
1544	0.266666666666667	0.766666666666667	0.692585070582643	0.693785463118374
1554	0.216666666666667	0.816666666666667	0.602478313542414	0.603404810493553
1564	0.166666666666667	0.966666666666667	0.498992668066654	0.499314767377287
1783	0.466666666666667	0.516666666666667	0.719874686628579	0.721400694310709
1793	0.433333333333333	0.533333333333333	0.725282953213406	0.726905328038456
1803	0.366666666666667	0.566666666666667	0.708313703952124	0.709958163014307
1813	0.283333333333333	0.633333333333333	0.65030652583573	0.65176944487161
1823	0.166666666666667	0.716666666666667	0.450372615553771	0.45129264217493
1833	0.0333333333333333	0.833333333333333	0.100824464887742	0.100966742252535
1843	0.183333333333333	0.633333333333333	0.455671165359783	0.4567727288213
1853	0.233333333333333	0.533333333333333	0.495853734942804	0.497260947684137
1863	0.183333333333333	0.533333333333333	0.40353084181104	0.404745680624419
1873	0.0333333333333333	0.633333333333333	0.0874565389108431	0.0876649456551453
2092	0.466666666666667	0.483333333333333	0.68304268631448	0.684583697304376
2102	0.433333333333333	0.466666666666667	0.652907504078121	0.654508497187474
2112	0.366666666666667	0.433333333333333	0.573390571525676	0.574912784645444
2122	0.283333333333333	0.416666666666667	0.471791101649147	0.473096486044902
2132	0.166666666666667	0.383333333333333	0.282396337085691	0.283203118462416
2142	0.0333333333333333	0.366666666666667	0.056740713946956	0.0569302813656985
2152	0.183333333333333	0.316666666666667	0.259207263553611	0.25987928673676
2162	0.233333333333333	0.266666666666667	0.271474600492949	0.272159936609677
2172	0.183333333333333	0.216666666666667	0.18137884954701	0.181804245694478
2182	0.0333333333333333	0.166666666666667	0.0269693252927089	0.027053957048968
2382	0.483333333333333	0.466666666666667	0.666666813417368	0.668213586118192
2392	0.466666666666667	0.433333333333333	0.624276019300482	0.625872908100787

DOI: 10.18535/ijecs/v5i12.75

2402	0.433333333333333	0.366666666666667	0.531273324439472	0.532737365365924
2412	0.366666666666667	0.283333333333333	0.392199603089928	0.393291456953952
2422	0.283333333333333	0.166666666666667	0.200630586791677	0.201140175649574
2432	0.166666666666667	0.0333333333333333	0.0260840835308031	0.0261679781214719
2442	0.366666666666667	0.183333333333333	0.258717222121411	0.259460928055066
2452	0.466666666666667	0.233333333333333	0.355252796100648	0.356404772421034
2462	0.466666666666667	0.183333333333333	0.281498932589996	0.282459478928619
2472	0.366666666666667	0.0333333333333333	0.0476528666114226	0.0478112750971233

TABLE-4b

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 4: (Output for 1/10th of the FEM MODEL)

Consider the boundary value problem: Find u such that

$$-\Delta u + \frac{1}{10}\frac{\partial u}{\partial y} = f \quad \text{in } \Omega = [0,1]^2$$

$$\left.\begin{aligned} \mathbf{u}(\mathbf{x},0) &= 0, 0 \leqslant x \leqslant 1 \\ \mathbf{u}(0,\mathbf{y}) &= 0, 0 \leqslant y \leqslant 1 \\ \mathbf{u}(1,\mathbf{y}) &= 0, 0 \leqslant y \leqslant 1 \\ \mathbf{u}(\mathbf{x},1) &= \sin(\pi x), 0 \leqslant x \leqslant 1 \end{aligned}\right\}$$

and f is chosen appropriately such that the exact solution is u(x,y)= $\sin(\pi x)\sin(\frac{\pi y}{2})$

ALL QUADRILATERAL FEM MODEL : number of nodes=5521,elements=5400 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
503	0.511111111111111	0.477777777777778	0.680917856049658	0.681582905088527
513	0.511111111111111	0.144444444444444	0.224440297621105	0.224814020239533
523	0.522222222222222	0.288888888888889	0.436698509411917	0.437303296707956
533	0.544444444444444	0.411111111111111	0.595259704138834	0.595958200716442
543	0.544444444444444	0.0777777777777778	0.120483035061912	0.120683319332619
553	0.555555555555556	0.188888888888889	0.287515097954794	0.287929921572347
563	0.577777777777778	0.277777777777778	0.409531800084388	0.410064693213191
573	0.588888888888889	0.355555555555556	0.508767446795089	0.509391090647197
583	0.588888888888889	0.0222222222222222	0.0334917842947837	0.0335475493876398
593	0.611111111111111	0.0777777777777778	0.114361187385059	0.114519722697841
603	0.622222222222222	0.122222222222222	0.176679913582581	0.176915019819241
613	0.644444444444444	0.144444444444444	0.201930623039542	0.202184668352986
623	0.655555555555556	0.155555555555556	0.213339591750279	0.213604355379595
633	0.677777777777778	0.144444444444444	0.190542201842522	0.190769313364695
643	0.688888888888889	0.122222222222222	0.158000671272071	0.158187826348637
653	0.711111111111111	0.0777777777777778	0.0959237704392919	0.0960343531382467
663	0.722222222222222	0.0222222222222222	0.0266933346324544	0.0267345655166
673	0.755555555555556	0.188888888888889	0.202877029527067	0.203098451971015
683	0.777777777777778	0.0777777777777778	0.0782540207947403	0.0783361039414628
693	0.811111111111111	0.144444444444444	0.125666611287914	0.125791033217352
703	0.844444444444444	0.144444444444444	0.105507338805416	0.105608123033148
713	0.877777777777778	0.0777777777777778	0.0456077824344631	0.0456530595748363
723	0.922222222222222	0.0222222222222222	0.00841747589637169	0.00844295239774339
1197	0.688888888888889	0.322222222222222	0.401446729606276	0.401925390720355
1207	0.544444444444444	0.477777777777778	0.674660399267453	0.675361198904008
1217	0.877777777777778	0.144444444444444	0.0841893090040727	0.084268148153073
1227	0.755555555555556	0.288888888888889	0.304168608172019	0.304518186484742
1237	0.644444444444444	0.411111111111111	0.540266355979046	0.540907759782456
1247	0.977777777777778	0.0777777777777778	0.00847560287802772	0.00850117565345499
1257	0.888888888888889	0.188888888888889	0.099898788631183	0.0999970123536406
1267	0.811111111111111	0.277777777777778	0.236063767908115	0.236325132842542
1277	0.755555555555556	0.355555555555556	0.367671485428379	0.368112852567069
1287	0.711111111111111	0.411111111111111	0.473657667690219	0.474236709925892
1297	0.688888888888889	0.455555555555556	0.54322815017045	0.543897584847108
1307	0.677777777777778	0.477777777777778	0.577658691222846	0.578367410832807
1317	0.688888888888889	0.488888888888889	0.575179661946217	0.575897889200368
1327	0.711111111111111	0.477777777777778	0.536737826768015	0.537422041671398
1337	0.755555555555556	0.455555555555556	0.455138389872424	0.455736896003458
1347	0.811111111111111	0.411111111111111	0.336094074144088	0.336530690148709
1357	0.888888888888889	0.355555555555556	0.181018152251101	0.181243062704074
1367	0.977777777777778	0.277777777777778	0.0294335822117086	0.0294803596789031
1377	0.877777777777778	0.411111111111111	0.225135038728121	0.225443875689507
1387	0.844444444444444	0.477777777777778	0.319689637725239	0.320178835915956
1397	0.877777777777778	0.477777777777778	0.255066218581495	0.255481082378251
1407	0.977777777777778	0.411111111111111	0.0419055459416483	0.0419804938613262
1881	0.522222222222222	0.511111111111111	0.716936090328037	0.717587524738918

1891	0.855555555555556	0.511111111111111	0.314861601027813	0.315337813205481
1901	0.711111111111111	0.522222222222222	0.575618650876002	0.576314581565988
1911	0.588888888888889	0.544444444444444	0.724741788724006	0.725473411025839
1921	0.922222222222222	0.544444444444444	0.182313007693878	0.182580772274722
1931	0.811111111111111	0.555555555555556	0.427826155829235	0.428366616335333
1941	0.722222222222222	0.577777777777778	0.602983716402608	0.603651258918428
1951	0.644444444444444	0.588888888888889	0.717084219623916	0.717808841593605
1961	0.977777777777778	0.588888888888889	0.0556473430641479	0.0557099969877403
1971	0.922222222222222	0.611111111111111	0.197933858895959	0.198170815338736
1981	0.877777777777778	0.622222222222222	0.310207899019269	0.310562940868641
1991	0.855555555555556	0.644444444444444	0.371359857077516	0.371759816444386
2001	0.844444444444444	0.655555555555556	0.401996145516043	0.402415672229584
2011	0.855555555555556	0.677777777777778	0.383024608985373	0.383408044023024
2021	0.877777777777778	0.688888888888889	0.330433104560453	0.330757989925662
2031	0.922222222222222	0.711111111111111	0.217233385690821	0.21743795943439
2041	0.977777777777778	0.722222222222222	0.0631696582073854	0.0632208353505186
2051	0.811111111111111	0.755555555555556	0.518049967792923	0.518474631686401
2061	0.922222222222222	0.777777777777778	0.227151745995883	0.227332220101547
2071	0.855555555555556	0.811111111111111	0.418917689362893	0.419216412551365
2081	0.855555555555556	0.844444444444444	0.425079688735274	0.42534965025215
2091	0.922222222222222	0.877777777777778	0.237345317688555	0.237477108991822
2101	0.977777777777778	0.922222222222222	0.0692092388258741	0.0692365195668007
2575	0.677777777777778	0.688888888888889	0.748122503204667	0.748782025129912
2585	0.522222222222222	0.544444444444444	0.752187061259335	0.75287114561692
2595	0.855555555555556	0.877777777777778	0.430079895853506	0.430317034127285
2605	0.711111111111111	0.755555555555556	0.730059087094991	0.730630847969159
2615	0.588888888888889	0.644444444444444	0.814473611530688	0.815196151148589
2625	0.922222222222222	0.977777777777778	0.241713631828955	0.241774523317379
2635	0.811111111111111	0.888888888888889	0.550414065339886	0.550697506767399
2645	0.722222222222222	0.811111111111111	0.732074317171605	0.732571944233734
2655	0.644444444444444	0.755555555555556	0.832717037628227	0.833347328309341
2665	0.588888888888889	0.711111111111111	0.863285923025795	0.863976289244801
2675	0.544444444444444	0.688888888888889	0.873650971120876	0.874354807580428
2685	0.522222222222222	0.677777777777778	0.87178747058699	0.872489177491037
2695	0.511111111111111	0.688888888888889	0.88171578621855	0.882409725041803
2705	0.522222222222222	0.711111111111111	0.895918532578009	0.896604629175613
2715	0.544444444444444	0.755555555555556	0.917502962800793	0.918160565030217
2725	0.588888888888889	0.811111111111111	0.918670183476691	0.919259131550908
2735	0.644444444444444	0.888888888888889	0.88470519916037	0.885139345156633
2745	0.722222222222222	0.977777777777778	0.765389649512626	0.765577789542058
2755	0.588888888888889	0.877777777777778	0.943122425406957	0.943600611140057
2765	0.522222222222222	0.844444444444444	0.967389097290964	0.967932134653681
2775	0.522222222222222	0.877777777777778	0.97852082360345	0.979235988965195
2785	0.588888888888889	0.977777777777778	0.960444515280542	0.960676121285575
3259	0.488888888888889	0.522222222222222	0.730265389895804	0.73090818070466
3269	0.488888888888889	0.855555555555556	0.97325316692673	0.973776504868366
3279	0.477777777777778	0.711111111111111	0.895918523502814	0.896604629175613
3289	0.455555555555556	0.588888888888889	0.790143948558417	0.790863244162972
3299	0.455555555555556	0.922222222222222	0.982499081262647	0.98288676072273
3309	0.444444444444444	0.811111111111111	0.941182303705109	0.941776337914845
3319	0.422222222222222	0.722222222222222	0.878701872110761	0.879386572452317
3329	0.411111111111111	0.644444444444444	0.814473567091786	0.815196151148589
3339	0.411111111111111	0.977777777777778	0.960444320745805	0.960676121285575
3349	0.388888888888889	0.922222222222222	0.932313067834125	0.932688294486801
3359	0.377777777777778	0.877777777777778	0.90968195839832	0.910148875696544
3369	0.355555555555556	0.855555555555556	0.875261768471574	0.875758013121103
3379	0.344444444444444	0.844444444444444	0.856212193012721	0.856720275876695
3389	0.322222222222222	0.855555555555556	0.82583869773495	0.826312678392932
3399	0.311111111111111	0.877777777777778	0.81337857125874	0.813805817319494
3409	0.288888888888889	0.922222222222222	0.781813902069488	0.78213704094433
3419	0.277777777777778	0.977777777777778	0.765389481299056	0.765577789542058
3429	0.244444444444444	0.811111111111111	0.663847693251203	0.664305103439471
3439	0.222222222222222	0.922222222222222	0.637728606546335	0.637996368317099
3449	0.188888888888889	0.855555555555556	0.544534424741282	0.544860825582235
3459	0.155555555555556	0.855555555555556	0.457162707257377	0.457439037046514
3469	0.122222222222222	0.922222222222222	0.371655643268152	0.371814332674429
3479	0.0777777777777778	0.977777777777778	0.241713584292825	0.241774523317379
3953	0.311111111111111	0.677777777777778	0.724436286654847	0.725092598915646
3963	0.455555555555556	0.522222222222222	0.723538143409465	0.724236217669415
3973	0.122222222222222	0.855555555555556	0.364783628683	0.365005450695638
3983	0.244444444444444	0.711111111111111	0.623803094523995	0.624354807580428
3993	0.355555555555556	0.588888888888889	0.717084147838937	0.717808841593605
4003	0.0222222222222222	0.922222222222222	0.0692092457025235	0.0692365195668005
4013	0.111111111111111	0.811111111111111	0.326840721980116	0.327075489697496

DOI: 10.18535/ijecs/v5i12.75

4023	0.188888888888889	0.722222222222222	0.506351552742036	0.506800882871172
4033	0.244444444444444	0.644444444444444	0.58851010649541 9	0.589103708546878
4043	0.288888888888889	0.588888888888889	0.628657359792575	0.629333370129456
4053	0.311111111111111	0.544444444444444	0.62497399441211	0.625682598371921
4063	0.322222222222222	0.522222222222222	0.619501975557737	0.620223114275092
4073	0.311111111111111	0.511111111111111	0.5956413673067	0.596359721914984
4083	0.288888888888889	0.522222222222222	0.57561856421226	0.576314581565988
4093	0.244444444444444	0.544444444444444	0.523627646184083	0.524265327167345
4103	0.188888888888889	0.588888888888889	0.446067036639344	0.446591309678186
4113	0.111111111111111	0.644444444444444	0.289732501215208	0.290049531394481
4123	0.0222222222222222	0.722222222222222	0.063169726504653	0.0632208353505184
4133	0.122222222222222	0.588888888888889	0.298803394910813	0.299174127799796
4143	0.155555555555556	0.522222222222222	0.342860208315425	0.343349765248398
4153	0.122222222222222	0.522222222222222	0.273559839113626	0.273969918745675
4163	0.0222222222222222	0.588888888888889	0.0556474444705618	0.0557099969877401
4637	0.477777777777778	0.488888888888889	0.692305681058676	0.692966217581967
4647	0.144444444444444	0.488888888888889	0.304032707647951	0.304518186484742
4657	0.288888888888889	0.477777777777778	0.536737996433175	0.537422041671398
4667	0.411111111111111	0.455555555555556	0.629933376604389	0.630644414843062
4677	0.0777777777777778	0.455555555555556	0.158451058645174	0.158715043918661
4687	0.188888888888889	0.444444444444444	0.358950856281565	0.359442269775637
4697	0.277777777777778	0.422222222222222	0.471039192622554	0.471624052015652
4707	0.355555555555556	0.411111111111111	0.540266472678919	0.540907759782456
4717	0.0222222222222222	0.411111111111111	0.0419057041498323	0.041980493861326
4727	0.0777777777777778	0.388888888888889	0.138570453225717	0.138760698753347
4737	0.122222222222222	0.377777777777778	0.209209640867905	0.209477348631529
4747	0.144444444444444	0.355555555555556	0.232017368770798	0.232301315567534
4757	0.155555555555556	0.344444444444444	0.2415072986782	0.24179572992226
4767	0.144444444444444	0.322222222222222	0.212279354826558	0.212526549201764
4777	0.122222222222222	0.311111111111111	0.175666498539011	0.175867142840867
4787	0.0777777777777778	0.288888888888889	0.105930563408526	0.106051578807414
4797	0.0222222222222222	0.277777777777778	0.0294337516351229	0.029480359678903
4807	0.188888888888889	0.244444444444444	0.209253025973566	0.209477348631529
4817	0.0777777777777778	0.222222222222222	0.0826554671681429	0.0827421614066158
4827	0.144444444444444	0.188888888888889	0.12804025033006	0.128167319487984
4837	0.144444444444444	0.155555555555556	0.105949412948942	0.106051578807414
4847	0.0777777777777778	0.122222222222222	0.0461143350614113	0.046160873858962
4857	0.0222222222222222	0.0777777777777778	0.00847573435130611	0.00850117565345496
5302	0.322222222222222	0.311111111111111	0.397660543158081	0.398134465020157
5312	0.477777777777778	0.455555555555556	0.653756855642538	0.654460902169298
5322	0.144444444444444	0.122222222222222	0.0835656611538364	0.0836451581208877
5332	0.288888888888889	0.244444444444444	0.294853360980392	0.29519402398372
5342	0.411111111111111	0.355555555555556	0.50876749902044	0.509391090647197
5352	0.0777777777777778	0.0222222222222222	0.00841735406381464	0.00844295239774339
5362	0.188888888888889	0.111111111111111	0.0970076007511774	0.0971028286519749
5372	0.277777777777778	0.188888888888889	0.223716091961727	0.223969719728075
5382	0.355555555555556	0.244444444444444	0.336204601531936	0.336694175866634
5392	0.411111111111111	0.288888888888889	0.42085014223658	0.421389392012894
5402	0.455555555555556	0.311111111111111	0.464292341586745	0.464902697809071
5412	0.477777777777778	0.322222222222222	0.482982588382475	0.483628648377863
5422	0.488888888888889	0.311111111111111	0.468543711246523	0.469185573394539
5432	0.477777777777778	0.288888888888889	0.436698512510651	0.437303296707956
5442	0.455555555555556	0.244444444444444	0.370444357042519	0.370960947799834
5452	0.411111111111111	0.188888888888889	0.280663499110999	0.281045720726155
5462	0.355555555555556	0.111111111111111	0.155875795322843	0.156073948237737
5472	0.277777777777778	0.0222222222222222	0.0266932236402432	0.0267345655166
5482	0.411111111111111	0.122222222222222	0.183156508949882	0.183417378495944
5492	0.477777777777778	0.155555555555556	0.240946539390853	0.241332586020939
5502	0.477777777777778	0.122222222222222	0.190025015941765	0.190344194253834
5512	0.411111111111111	0.0222222222222222	0.0334917341173787	0.0335475493876398

TABLE-4c

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 4: (Output for 1/10th of the FEM MODEL)

Consider the boundary value problem: Find u such that

$$-\Delta u + \frac{1}{10}\frac{\partial u}{\partial y} = f \quad \text{in } \Omega = [0,1]^2$$

$$\left.\begin{aligned} \mathbf{u}(\mathbf{x},\mathbf{0}) &= \mathbf{0}, \mathbf{0} \leqslant x \leqslant \mathbf{1} \\ \mathbf{u}(\mathbf{0},\mathbf{y}) &= \mathbf{0}, \mathbf{0} \leqslant y \leqslant \mathbf{1} \\ \mathbf{u}(\mathbf{1},\mathbf{y}) &= \mathbf{0}, \mathbf{0} \leqslant y \leqslant \mathbf{1} \\ \mathbf{u}(\mathbf{x},\mathbf{1}) &= \sin(\pi x), \mathbf{0} \leqslant x \leqslant \mathbf{1} \end{aligned}\right\}$$

DOI: 10.18535/ijecs/v5i12.75

and f is chosen appropriately such that the exact solution is u(x,y)= $\sin(\pi x)\sin(\frac{\pi y}{2})$

ALL QUADRILATERAL FEM MODEL : number of nodes=7921,elements= 7776 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
710	0.509259259259259	0.481481481481481	0.685497568057507	0.685951323400243
720	0.509259259259259	0.203703703703704	0.314074592074697	0.314411688018379
730	0.518518518518518	0.407407407407407	0.595652066898395	0.596148293878673
740	0.518518518518518	0.12962962962963	0.201640700268635	0.201875452204884
750	0.537037037037037	0.314814814814815	0.470959230764794	0.471391291831865
760	0.537037037037037	0.037037037037037	0.0576784989973409	0.0577516743782273
770	0.546296296296296	0.212962962962963	0.324527398496381	0.324850683817573
780	0.564814814814815	0.37037037037037	0.537700233270462	0.538156464249616
790	0.564814814814815	0.0925925925925926	0.14178630116427	0.141937657207596
800	0.574074074074074	0.240740740740741	0.358918932494459	0.359254147829146
810	0.592592592592593	0.37037037037037	0.525982746837986	0.526423837915034
820	0.592592592592593	0.0925925925925926	0.138705279241202	0.138843201216726
830	0.601851851851852	0.212962962962963	0.31136810310399	0.311652663330152
840	0.62037037037037	0.314814814814815	0.440692736318707	0.441068678622104
850	0.62037037037037	0.037037037037037	0.0539826950847859	0.0540367527945429
860	0.62962962962963	0.12962962962963	0.185511139925422	0.185679432009504
870	0.648148148148148	0.203703703703704	0.280846605258124	0.28108746093965
880	0.657407407407407	0.268518518518519	0.360044043812897	0.360347684784312
890	0.675925925925926	0.314814814814815	0.403605888971135	0.403940287436981
900	0.675925925925926	0.037037037037037	0.0494422885611486	0.0494880333017028
910	0.685185185185185	0.0740740740740741	0.0969120196504791	0.0969942147430386
920	0.703703703703704	0.0925925925925926	0.1161600787882	0.116253205719454
930	0.712962962962963	0.101851851851852	0.124863585898389	0.124962802841441
940	0.731481481481482	0.0925925925925926	0.108184740932187	0.108267732524086
950	0.740740740740741	0.0740740740740741	0.0843771227083811	0.0844429257080959
960	0.759259259259259	0.037037037037037	0.0398666151327712	0.0399014026251222
970	0.787037037037037	0.203703703703704	0.194948679493083	0.195091821357551
980	0.796296296296296	0.12962962962963	0.120671194675646	0.120755960711356
990	0.814814814814815	0.037037037037037	0.0319229965350115	0.0319511055146974
1000	0.842592592592593	0.0925925925925926	0.0687389533392226	0.0687847140154323
1010	0.87037037037037	0.0925925925925926	0.0573667782966136	0.0574045769260343
1020	0.898148148148148	0.037037037037037	0.0182692310934675	0.0182891510311229
1030	0.953703703703704	0.037037037037037	0.00840945912656236	0.00842703816309703
1705	0.685185185185185	0.324074074074074	0.406839296096647	0.407176431227806
1715	0.962962962962963	0.0462962962962963	0.00841722736344253	0.00843507555561353
1725	0.759259259259259	0.259259259259259	0.271594900251083	0.271806427373376
1735	0.574074074074074	0.462962962962963	0.646380165180086	0.646876207018316
1745	0.851851851851852	0.185185185185185	0.128628678446946	0.12871705571958
1755	0.675925925925926	0.37037037037037	0.467308024463693	0.467696252889881
1765	0.953703703703704	0.0925925925925926	0.0209844812838532	0.0210052438422556
1775	0.796296296296296	0.268518518518519	0.244283812845454	0.244472137996346
1785	0.648148148148148	0.425925925925926	0.553806200511463	0.554262679684344
1795	0.925925925925926	0.148148148148148	0.0531470243253328	0.0531836798382939
1805	0.796296296296296	0.296296296296296	0.267792745771697	0.268004286405873
1815	0.675925925925926	0.425925925925926	0.527451172290627	0.527892767455569
1825	0.953703703703704	0.148148148148148	0.0333958269092098	0.0334235869324616
1835	0.851851851851852	0.268518518518519	0.183596586111611	0.18373493513968
1845	0.759259259259259	0.37037037037037	0.376779187708608	0.377095941134884
1855	0.685185185185185	0.462962962962963	0.554954254062863	0.555428842798157
1865	0.962962962962963	0.185185185185185	0.0332655728939349	0.0332958230659671
1875	0.898148148148148	0.259259259259259	0.124492811414744	0.124584813425375
1885	0.851851851851852	0.324074074074074	0.218546044855435	0.218723057399182
1895	0.814814814814815	0.37037037037037	0.301702586843213	0.301960116980422
1905	0.796296296296296	0.407407407407407	0.35628095028859	0.356598383644455
1915	0.787037037037037	0.425925925925926	0.384344497566328	0.38469206462878
1925	0.796296296296296	0.435185185185185	0.376803593681673	0.377152438813767
1935	0.814814814814815	0.425925925925926	0.340509748167987	0.340824970970066
1945	0.851851851851852	0.407407407407407	0.267756044106941	0.268004286405873
1955	0.898148148148148	0.37037037037037	0.172691810998887	0.172845167510404
1965	0.962962962962963	0.324074074074074	0.0565239947563493	0.0565780826705373
1975	0.851851851851852	0.462962962962963	0.298049216333953	0.298359839488192
1985	0.953703703703704	0.37037037037037	0.0795638068627347	0.0796413578978271
1995	0.925925925925926	0.425925925925926	0.142885755688292	0.143036147884195
2005	0.953703703703704	0.425925925925926	0.0897910250236474	0.0898918829578527
2015	0.962962962962963	0.462962962962963	0.0770734687919917	0.0771780893375271
2690	0.574074074074074	0.509259259259259	0.697477170031456	0.697980732733718
2700	0.851851851851852	0.509259259259259	0.321593410015669	0.321930868881666
2710	0.648148148148148	0.518518518518518	0.649494553882817	0.650004827820579

2720	0.925925925925926	0.518518518518518	0.167552848376771	0.167743905706468
2730	0.740740740740741	0.537037037037037	0.54290817790018	0.54336634641493
2740	0.564814814814815	0.546296296296296	0.740482472727196	0.740983591890458
2750	0.842592592592593	0.546296296296296	0.358756536028223	0.359089655705443
2760	0.685185185185185	0.564814814814815	0.647273074856626	0.647763773900153
2770	0.962962962962963	0.564814814814815	0.0899244921650491	0.0900082360862329
2780	0.814814814814815	0.574074074074074	0.430679283169221	0.431043450412688
2790	0.685185185185185	0.592592592592593	0.66967900664294	0.670164150798468
2800	0.962962962962963	0.592592592592593	0.0930419827646353	0.0931208189343669
2810	0.842592592592593	0.601851851851852	0.384456504627974	0.384769609822035
2820	0.740740740740741	0.62037037037037	0.601398807120312	0.601834363527739
2830	0.648148148148148	0.62962962962963	0.746125407264527	0.746619178870972
2840	0.925925925925926	0.62962962962963	0.192524474456146	0.19267674912369
2850	0.851851851851852	0.648148148148148	0.381697811136205	0.381980464844671
2860	0.787037037037037	0.657407407407407	0.532201218196637	0.53257255878452
2870	0.740740740740741	0.675925925925926	0.634732169597453	0.635146849691195
2880	0.703703703703704	0.685185185185185	0.705587643190075	0.706029950160822
2890	0.981481481481482	0.685185185185185	0.0511491896389203	0.0511791592969303
2900	0.962962962962963	0.703703703703704	0.10367797166545	0.103744417372569
2910	0.953703703703704	0.712962962962963	0.130365515992802	0.13044815363961
2920	0.962962962962963	0.731481481481482	0.105854806884464	0.105918350647796
2930	0.981481481481482	0.740740740740741	0.0533618525532986	0.0533895184373966
2940	0.787037037037037	0.768518518518518	0.579367695628774	0.579683828312443
2950	0.851851851851852	0.787037037037037	0.423695734873539	0.423921136369671
2960	0.925925925925926	0.796296296296296	0.218796680002704	0.218910418793451
2970	0.842592592592593	0.824074074074074	0.456378376936325	0.456593758574379
2980	0.962962962962963	0.842592592592593	0.11251330349886	0.112562284797413
2990	0.962962962962963	0.87037037037037	0.113650252823004	0.113694506701978
3000	0.925925925925926	0.907407407407407	0.228106545476555	0.228180945030171
3010	0.981481481481482	0.962962962962963	0.058035338073568	0.058046457062615
3685	0.731481481481482	0.740740740740741	0.685540671471742	0.685930452656867
3695	0.518518518518518	0.537037037037037	0.745293452984286	0.745761223053062
3705	0.796296296296296	0.814814814814815	0.571797003699558	0.572071668040356
3715	0.592592592592593	0.62962962962963	0.799883603168018	0.800388561001014
3725	0.87037037037037	0.907407407407407	0.391769926834435	0.391897812718541
3735	0.685185185185185	0.731481481481482	0.761833180737262	0.762264360732099
3745	0.509259259259259	0.574074074074074	0.783601864724181	0.784083817937011
3755	0.787037037037037	0.851851851851852	0.603263772168209	0.603516963330138
3765	0.62962962962963	0.703703703703704	0.820074139482258	0.820547883978904
3775	0.907407407407407	0.981481481481482	0.286634441959801	0.286681900627067
3785	0.759259259259259	0.851851851851852	0.667466219964932	0.667743905706468
3795	0.62962962962963	0.731481481481482	0.837283159132923	0.837742219771384
3805	0.509259259259259	0.62962962962963	0.834644098047469	0.835134358362996
3815	0.787037037037037	0.907407407407407	0.613489709941545	0.613686820786524
3825	0.685185185185185	0.814814814814815	0.800020721460893	0.800388561001014
3835	0.592592592592593	0.740740740740741	0.879177979785215	0.879641400430461
3845	0.518518518518518	0.675925925925926	0.871241434574897	0.871728428838503
3855	0.796296296296296	0.953703703703704	0.595444808066547	0.595580251801947
3865	0.731481481481482	0.907407407407407	0.738903607654208	0.739137710549738
3875	0.685185185185185	0.87037037037037	0.817919173523954	0.818227152706685
3885	0.648148148148148	0.851851851851852	0.86919699513552	0.869544656849401
3895	0.62962962962963	0.842592592592593	0.889925768826079	0.890291226704305
3905	0.62037037037037	0.851851851851852	0.903939091222964	0.904296841434968
3915	0.62962962962963	0.87037037037037	0.89891445749225	0.899246333027272
3925	0.648148148148148	0.907407407407407	0.883923770328634	0.884197343931013
3935	0.685185185185185	0.953703703703704	0.833094152941246	0.833279547531715
3945	0.518518518518518	0.814814814814815	0.955961215479278	0.956368745682866
3955	0.592592592592593	0.907407407407407	0.947586330478428	0.947874712808797
3965	0.509259259259259	0.851851851851852	0.972265433078615	0.97263322402731
3975	0.62962962962963	0.981481481481482	0.917680484720801	0.917827655631736
3985	0.509259259259259	0.907407407407407	0.988730172731036	0.989023055377075
3995	0.518518518518518	0.953703703703704	0.995456620159017	0.995669546650459
4670	0.490740740740741	0.62962962962963	0.834644093676367	0.835134358362996
4680	0.490740740740741	0.907407407407407	0.988730163606475	0.989023055377075
4690	0.481481481481481	0.703703703703704	0.891642063629652	0.892120755332356
4700	0.481481481481481	0.981481481481482	0.997729296458352	0.997885824086973
4710	0.462962962962963	0.796296296296296	0.942398110964824	0.942824204401101
4720	0.453703703703704	0.62037037037037	0.818170562064962	0.818671374337122
4730	0.453703703703704	0.898148148148148	0.976497944074903	0.976805591337667
4740	0.435185185185185	0.740740740740741	0.898780123591947	0.899246333027272
4750	0.425925925925926	0.592592592592593	0.779995376881979	0.780501858283407
4760	0.425925925925926	0.87037037037037	0.952597838339478	0.952942368559425
4770	0.407407407407407	0.740740740740741	0.879177963020621	0.879641400430461
4780	0.398148148148148	0.62037037037037	0.784904884736923	0.785410426741167

4790	0.398148148148148	0.898148148148148	0.936819178067429	0.937119973147798
4800	0.37962962962963	0.796296296296296	0.881761180873851	0.882176300695123
4810	0.37037037037037	0.703703703703704	0.82007411754848	0.820547883978904
4820	0.37037037037037	0.981481481481482	0.917680387992586	0.917827655631736
4830	0.351851851851852	0.907407407407407	0.88392371218802	0.884197343931013
4840	0.342592592592593	0.842592592592593	0.853078358756202	0.85343261834412
4850	0.324074074074074	0.796296296296296	0.807526262209164	0.807916240133182
4860	0.314814814814815	0.759259259259259	0.77604530002457	0.776458529058299
4870	0.296296296296296	0.740740740740741	0.736109857177237	0.736522435289912
4880	0.287037037037037	0.731481481481482	0.71525715769085	0.715668256556575
4890	0.268518518518519	0.740740740740741	0.685540649465006	0.685930452656867
4900	0.259259259259259	0.759259259259259	0.675613115241199	0.675982892983761
4910	0.240740740740741	0.796296296296296	0.651084032525996	0.651409826460405
4920	0.231481481481481	0.842592592592593	0.644299200159948	0.644578027188272
4930	0.212962962962963	0.907407407407407	0.613489664509919	0.613686820786524
4940	0.203703703703704	0.981481481481482	0.596808251684126	0.596905963809653
4950	0.175925925925926	0.898148148148148	0.518094586702348	0.51827216381108
4960	0.148148148148148	0.87037037037037	0.439353267005159	0.43952726818539
4970	0.12037037037037	0.898148148148148	0.364364789824019	0.364491057368984
4980	0.0925925925925926	0.981481481481482	0.286634413132415	0.286681900627067
5655	0.481481481481481	0.509259259259259	0.715658452448112	0.716102495241577
5665	0.203703703703704	0.787037037037037	0.563762486666852	0.564056620322892
5675	0.407407407407407	0.574074074074074	0.750952383386316	0.751461980288934
5685	0.12962962962963	0.851851851851852	0.38523828392821	0.385403384684858
5695	0.314814814814815	0.648148148148148	0.710627948062087	0.71109760591145
5705	0.037037037037037	0.925925925925926	0.115274850972449	0.11530793537122
5715	0.212962962962963	0.740740740740741	0.569177706632766	0.569510218141316
5725	0.37037037037037	0.564814814814815	0.711394955253347	0.711904018854324
5735	0.0925925925925926	0.842592592592593	0.277956471998488	0.278080944082601
5745	0.240740740740741	0.685185185185185	0.603639128972352	0.604030844337777
5755	0.37037037037037	0.537037037037037	0.685419291897091	0.685930452656867
5765	0.0925925925925926	0.814814814814815	0.274618894115635	0.274754489035403
5775	0.212962962962963	0.685185185185185	0.545572292637079	0.545932142273602
5785	0.314814814814815	0.564814814814815	0.64727303593278	0.647763773900153
5795	0.037037037037037	0.842592592592593	0.11251331591428	0.112562284797412
5805	0.12962962962963	0.740740740740741	0.363467545941671	0.363686820786524
5815	0.203703703703704	0.648148148148148	0.507888277134495	0.50825163348724
5825	0.268518518518519	0.574074074074074	0.5855208313181	0.585978167969099
5835	0.314814814814815	0.509259259259259	0.598808007647454	0.599308842204367
5845	0.037037037037037	0.787037037037037	0.109600613191366	0.109657597989488
5855	0.0740740740740741	0.740740740740741	0.21162709362763	0.211755207017928
5865	0.0925925925925926	0.703703703703704	0.256126217489292	0.256296730100902
5875	0.101851851851852	0.685185185185185	0.276671147395932	0.276862731953059
5885	0.0925925925925926	0.675925925925926	0.250260511674084	0.250438233290593
5895	0.0740740740740741	0.685185185185185	0.202848003011369	0.202988410255708
5905	0.037037037037037	0.703703703703704	0.103678003494777	0.103744417372569
5915	0.203703703703704	0.509259259259259	0.427938970095542	0.428351460448661
5925	0.12962962962963	0.574074074074074	0.310414898883909	0.310691173038795
5935	0.037037037037037	0.648148148148148	0.0987365833682937	0.0988086147637803
5945	0.0925925925925926	0.564814814814815	0.222152652042353	0.22236200438821
5955	0.0925925925925926	0.537037037037037	0.214028232437235	0.21424920534815
5965	0.037037037037037	0.564814814814815	0.0899245387025144	0.0900082360862328
5975	0.037037037037037	0.509259259259259	0.0831660561338572	0.0832753141363726
6650	0.314814814814815	0.490740740740741	0.581622094018971	0.582124380760023
6660	0.037037037037037	0.490740740740741	0.0807615599600617	0.0808874944943688
6670	0.240740740740741	0.481481481481481	0.470483899563358	0.470927585544762
6680	0.425925925925926	0.462962962962963	0.646380226469934	0.646876207018316
6690	0.148148148148148	0.462962962962963	0.298049290310985	0.298359839488192
6700	0.324074074074074	0.453703703703704	0.556043115089632	0.556511909407875
6710	0.0462962962962963	0.453703703703704	0.0946474588165401	0.094765275277908
6720	0.203703703703704	0.435185185185185	0.376803663393012	0.377152438813767
6730	0.351851851851852	0.425925925925926	0.553806262079972	0.554262679684344
6740	0.0740740740740741	0.425925925925926	0.142885829905059	0.143036147884195
6750	0.203703703703704	0.407407407407407	0.356281014948249	0.356598383644455
6760	0.324074074074074	0.398148148148148	0.497853967557797	0.498268751616949
6770	0.0462962962962963	0.398148148148148	0.0847595024555891	0.0848473763294661
6780	0.148148148148148	0.37962962962963	0.251824488323472	0.252046565123284
6790	0.240740740740741	0.37037037037037	0.376779244607785	0.377095941134884
6800	0.314814814814815	0.351851851851852	0.438247293339999	0.438611534146769
6810	0.037037037037037	0.351851851851852	0.0608856460156492	0.0609460610585371
6820	0.101851851851852	0.342592592592593	0.161081949205832	0.161217555931432
6830	0.148148148148148	0.324074074074074	0.218546101788128	0.218723057399182
6840	0.185185185185185	0.314814814814815	0.260589195011822	0.260797164172053
6850	0.203703703703704	0.296296296296296	0.267792789206233	0.268004286405873

6860	0.212962962962963	0.287037037037037	0.270059223002625	0.270270598601427
6870	0.203703703703704	0.268518518518519	0.24428388020232	0.244472137996346
6880	0.185185185185185	0.259259259259259	0.217485553585889	0.217649387470701
6890	0.148148148148148	0.240740740740741	0.165578717475862	0.165699416238904
6900	0.101851851851852	0.231481481481481	0.111788548796585	0.111868082150196
6910	0.037037037037037	0.212962962962963	0.0380811343249684	0.0381152976279061
6920	0.148148148148148	0.185185185185185	0.128628705470619	0.12871705571958
6930	0.0462962962962963	0.175925925925926	0.0395111277348257	0.0395432080103981
6940	0.0740740740740741	0.148148148148148	0.0531470702961679	0.0531836798382939
6950	0.0462962962962963	0.12037037037037	0.0272164820806916	0.0272403421571265
6960	0.037037037037037	0.0740740740740741	0.0134588829383089	0.0134775647100881
7600	0.435185185185185	0.425925925925926	0.606937984806132	0.60742181165972
7610	0.157407407407407	0.148148148148148	0.109377609737595	0.109450377524036
7620	0.37037037037037	0.351851851851852	0.481637598206002	0.482041951798119
7630	0.0925925925925926	0.0740740740740741	0.0332705654126815	0.033295823065967
7640	0.296296296296296	0.259259259259259	0.317447700272458	0.317704766520999
7650	0.481481481481481	0.435185185185185	0.630013090351845	0.6305098206928
7660	0.203703703703704	0.157407407407407	0.146046845572543	0.146150590604395
7670	0.37962962962963	0.314814814814815	0.440692763069661	0.441068678622104
7680	0.101851851851852	0.037037037037037	0.0182691840568743	0.0182891510311229
7690	0.259259259259259	0.185185185185185	0.208452878414903	0.208613111791988
7700	0.407407407407407	0.314814814814815	0.454266967146888	0.454662176759293
7710	0.12962962962963	0.037037037037037	0.0230076671097035	0.0230299902312481
7720	0.259259259259259	0.157407407407407	0.177884273148764	0.178019857342822
7730	0.37962962962963	0.259259259259259	0.36777509365961	0.368095749964829
7740	0.481481481481481	0.351851851851852	0.523616298576725	0.524088403049343
7750	0.203703703703704	0.0740740740740741	0.0692754242780604	0.069325881105695
7760	0.296296296296296	0.148148148148148	0.184835215811345	0.184982338539911
7770	0.37037037037037	0.212962962962963	0.301201041106279	0.301466118446525
7780	0.435185185185185	0.259259259259259	0.387536565915286	0.387897004341542
7790	0.481481481481481	0.296296296296296	0.447610695154631	0.448039883019578
7800	0.203703703703704	0.0185185185185185	0.0173459410811431	0.017368189684708
7810	0.240740740740741	0.037037037037037	0.0398665652604372	0.0399014026251222
7820	0.259259259259259	0.0462962962962963	0.0528047092155116	0.0528494927538132
7830	0.268518518518519	0.037037037037037	0.0433972810268503	0.0434356449591437
7840	0.259259259259259	0.0185185185185185	0.0211306110534	0.0211554577863048
7850	0.481481481481481	0.212962962962963	0.327419144583609	0.327761714516275
7860	0.435185185185185	0.148148148148148	0.225622675927223	0.225851490241905
7870	0.37037037037037	0.0740740740740741	0.106498429686171	0.106598383644455
7880	0.481481481481481	0.157407407407407	0.244056208971909	0.244329276952193
7890	0.37962962962963	0.037037037037037	0.0539826698211894	0.0540367527945429
7900	0.407407407407407	0.037037037037037	0.055643147884083	0.0557021362916143
7910	0.481481481481481	0.0740740740740741	0.11574733099366	0.115896503288703
7920	0.481481481481481	0.0185185185185185	0.0289917236944398	0.0290355120022734

TABLE-5a

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 5: (Output for 1/10th of the FEM MODEL)

Consider the boundary value problem: Find u such that

$$-\epsilon \ \ \Delta u + 2\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = f \ \text{ in } \Omega = [-1,1]^2$$

$$\mathbf{u}(1,\mathbf{y}) = 0, -1 \leqslant y \leqslant 1$$

$$\mathbf{u}(\mathbf{x},1) = 0, -1 \leqslant x \leqslant 1$$

$$\left.\begin{aligned} \mathbf{u}(-1,\mathbf{y}) &= (1-e^{\frac{-2}{\epsilon}})(1-e^{\frac{-(1-y)}{\epsilon}})\cos(\pi(-1+y)), -1 \leqslant y \leqslant 1 \\ \mathbf{u}(\mathbf{x},-1) &= (1-e^{\frac{-(1-x)}{\epsilon}})(1-e^{\frac{-2}{\epsilon}})\cos(\pi(x-1)), -1 \leqslant x \leqslant 1 \end{aligned}\right\}$$

Where, $\epsilon = 10^{-3}$ and f is chosen appropriately such that the exact solution is

$$\mathbf{u(x,y)}= (1-e^{\frac{-(1-x)}{\epsilon}})(1-e^{\frac{-(1-y)}{\epsilon}})\cos(\pi(\mathbf{x}+y))$$

ALL QUADRILATERAL FEM MODEL : number of nodes=2481,elements=2400 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
238	0.0333333333333333	-0.0666666666666667	0.962795848807421	0.994521895368273
248	0.0666666666666667	-0.133333333333333	1.03151182911106	0.978147600733806
258	0.133333333333333	-0.266666666666667	0.992464520568502	0.913545457642601
268	0.166666666666667	-0.433333333333333	0.920602353338929	0.669130606358858
278	0.233333333333333	-0.666666666666667	0.500779110656084	0.207911690817759

288	0.266666666666667	-0.933333333333333	-0.578087113439388	-0.5
298	0.366666666666667	-0.633333333333333	1.05615915373495	0.669130606358858
308	0.466666666666667	-0.533333333333333	1.31102848621291	0.978147600733806
318	0.566666666666667	-0.633333333333333	1.42894681579618	0.978147600733806
328	0.666666666666667	-0.933333333333333	0.79677596879466	0.669130606358858
547	0.0666666666666667	-0.0333333333333333	0.981384424567908	0.994521895368273
557	0.133333333333333	-0.0666666666666667	0.919643913426996	0.978147600733806
567	0.266666666666667	-0.133333333333333	0.963665821190691	0.913545457642601
577	0.433333333333333	-0.266666666666667	0.657731843928972	0.866025403784439
587	0.666666666666667	-0.433333333333333	0.837009303963601	0.743144825477394
597	0.933333333333333	-0.666666666666667	0.961764098277272	0.669130606358858
607	0.633333333333333	-0.266666666666667	0.0162714823925281	0.4067366430758
617	0.533333333333333	-0.0666666666666667	0.0681692603126016	0.104528463267653
627	0.633333333333333	-0.0666666666666667	-0.451466382860389	-0.207911690817759
637	0.933333333333333	-0.266666666666667	-1.48793013802885	-0.5
856	0.0666666666666667	0.0333333333333333	0.814744445917097	0.951056516295154
866	0.133333333333333	0.0666666666666667	0.656843584934404	0.809016994374947
876	0.266666666666667	0.133333333333333	0.259463468970787	0.309016994374947
886	0.433333333333333	0.166666666666667	-0.352513361993681	-0.309016994374947
896	0.666666666666667	0.233333333333333	-0.832828324274619	-0.951056516295153
906	0.933333333333333	0.266666666666667	-0.932770560727023	-0.809016994374947
916	0.633333333333333	0.366666666666667	-0.780445960570131	-1
926	0.533333333333333	0.466666666666667	-0.822523657802451	-1
936	0.633333333333333	0.566666666666667	-0.376096549550792	-0.809016994374947
946	0.933333333333333	0.666666666666667	2.74071851151176	0.309016994374947
1165	0.0333333333333333	0.0666666666666667	0.938249091843676	0.951056516295154
1175	0.0666666666666667	0.133333333333333	0.61976954828292	0.809016994374947
1185	0.133333333333333	0.266666666666667	0.0332779110102447	0.309016994374947
1195	0.266666666666667	0.433333333333333	-0.650858318399325	-0.587785252292473
1205	0.433333333333333	0.666666666666667	-0.850279625085646	-0.951056516295154
1215	0.666666666666667	0.933333333333333	1.03538174963923	0.309016994374947
1225	0.266666666666667	0.633333333333333	-1.09766388165467	-0.951056516295153
1235	0.0666666666666667	0.533333333333333	-0.657595455105519	-0.309016994374947
1245	0.0666666666666667	0.633333333333333	-0.85205176841664	-0.587785252292473
1255	0.266666666666667	0.933333333333333	-0.754944187225066	-0.809016994374947
1474	-0.0333333333333333	0.0666666666666667	0.98069545291728	0.994521895368273
1484	-0.0666666666666667	0.133333333333333	0.983486863344345	0.978147600733806
1494	-0.133333333333333	0.266666666666667	0.911124766223779	0.913545457642601
1504	-0.166666666666667	0.433333333333333	0.754201098399797	0.669130606358858
1514	-0.233333333333333	0.666666666666667	0.498901690123068	0.207911690817759
1524	-0.266666666666667	0.933333333333333	-1.62585413187677	-0.5
1534	-0.366666666666667	0.633333333333333	0.975778769507884	0.669130606358858
1544	-0.466666666666667	0.533333333333333	1.15826872527238	0.978147600733806
1554	-0.566666666666667	0.633333333333333	1.23969222583715	0.978147600733806
1564	-0.666666666666667	0.933333333333333	0.352490029927639	0.669130606358858
1783	-0.0666666666666667	0.0333333333333333	0.98367757671354	0.994521895368273
1793	-0.133333333333333	0.0666666666666667	0.967606427837943	0.978147600733806
1803	-0.266666666666667	0.133333333333333	0.887713419021712	0.913545457642601
1813	-0.433333333333333	0.266666666666667	0.833015044376165	0.866025403784439
1823	-0.666666666666667	0.433333333333333	0.758061163474013	0.743144825477394
1833	-0.933333333333333	0.666666666666667	0.561479495272302	0.669130606358858
1843	-0.633333333333333	0.266666666666667	0.418884742310744	0.4067366430758
1853	-0.533333333333333	0.0666666666666667	0.101033542229898	0.104528463267653
1863	-0.633333333333333	0.0666666666666667	-0.215703023375759	-0.207911690817759
1873	-0.933333333333333	0.266666666666667	-0.473079000506166	-0.5
2092	-0.0666666666666667	-0.0333333333333333	0.987604471633239	0.951056516295154
2102	-0.133333333333333	-0.0666666666666667	0.823648358549986	0.809016994374947
2112	-0.266666666666667	-0.133333333333333	0.342518865152743	0.309016994374947
2122	-0.433333333333333	-0.166666666666667	-0.27435585353139	-0.309016994374947
2132	-0.666666666666667	-0.233333333333333	-0.910632003395366	-0.951056516295153
2142	-0.933333333333333	-0.266666666666667	-0.777838401672852	-0.809016994374947
2152	-0.633333333333333	-0.366666666666667	-0.945396345262756	-1
2162	-0.533333333333333	-0.466666666666667	-0.945809479044675	-1
2172	-0.633333333333333	-0.566666666666667	-0.764809397849098	-0.809016994374947
2182	-0.933333333333333	-0.666666666666667	0.255714129126071	0.309016994374947
2382	-0.0333333333333333	-0.0666666666666667	0.906717390300642	0.951056516295154
2392	-0.0666666666666667	-0.133333333333333	0.874027837991597	0.809016994374947
2402	-0.133333333333333	-0.266666666666667	0.378032384158121	0.309016994374947
2412	-0.266666666666667	-0.433333333333333	-0.529124303860477	-0.587785252292473
2422	-0.433333333333333	-0.666666666666667	-0.899843266592665	-0.951056516295154
2432	-0.666666666666667	-0.933333333333333	0.234636077506996	0.309016994374947
2442	-0.266666666666667	-0.633333333333333	-0.937588677996922	-0.951056516295153
2452	-0.0666666666666667	-0.533333333333333	-0.30829579643619	-0.309016994374947
2462	-0.0666666666666667	-0.633333333333333	-0.564544587415486	-0.587785252292473

DOI: 10.18535/ijecs/v5i12.75

2472	-0.266666666666667	-0.933333333333333	-0.776469296303108	-0.809016994374947

TABLE-5b

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 5: (Output for 1/10th of the FEM MODEL)

Consider the boundary value problem: Find u such that

$$-\epsilon \ \ \Delta u + 2\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = f \ \text{ in } \ \Omega = [-1,1]^2$$

$$\mathbf{u}(1,\mathbf{y}) = 0, -1 \leqslant y \leqslant 1$$

$$\mathbf{u}(\mathbf{x},1) = 0, -1 \leqslant x \leqslant 1$$

$$\left.\begin{aligned} \mathbf{u}(-1,\mathbf{y}) &= (1 - e^{\frac{-2}{\epsilon}})(1 - e^{\frac{-(1-y)}{\epsilon}})\cos(\pi(-1+y)), -1 \leqslant y \leqslant 1 \\ \mathbf{u}(\mathbf{x},-1) &= (1 - e^{\frac{-(1-x)}{\epsilon}})(1 - e^{\frac{-2}{\epsilon}})\cos(\pi(x-1)), -1 \leqslant x \leqslant 1 \end{aligned}\right\}$$

Where, $\epsilon = 10^{-3}$ and f is chosen appropriately such that the exact solution is

$$\mathbf{u(x,y)} = (1 - e^{\frac{-(1-x)}{\epsilon}})(1 - e^{\frac{-(1-y)}{\epsilon}})\cos(\pi(\mathrm{x}+y))$$

ALL QUADRILATERAL FEM MODEL : number of nodes=5521,elements=5400 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
503	0.0222222222222222	-0.0444444444444444	0.980529814842816	0.997564050259824
513	0.0222222222222222	-0.711111111111111	-0.545320478822174	-0.559192903470747
523	0.0444444444444444	-0.422222222222222	0.437276701116311	0.374606593415912
533	0.0888888888888889	-0.177777777777778	0.955291879703962	0.961261695938319
543	0.0888888888888889	-0.844444444444444	-0.705278530976618	-0.719339800338651
553	0.111111111111111	-0.622222222222222	0.00135421871270793	-0.0348994967025007
563	0.155555555555556	-0.444444444444444	0.641163825787995	0.615661475325658
573	0.177777777777778	-0.288888888888889	0.971403131450033	0.939692620785908
583	0.177777777777778	-0.955555555555556	-0.772639143206528	-0.766044443118978
593	0.222222222222222	-0.844444444444444	-0.361054065525629	-0.374606593415912
603	0.244444444444444	-0.755555555555556	-0.0269242715003042	-0.0348994967025007
613	0.288888888888889	-0.711111111111111	0.299422043638742	0.241921895599668
623	0.311111111111111	-0.688888888888889	0.432760499375364	0.374606593415912
633	0.355555555555556	-0.711111111111111	0.513121612411265	0.438371146789077
643	0.377777777777778	-0.755555555555556	0.404268342358512	0.374606593415912
653	0.422222222222222	-0.844444444444444	0.262167833592101	0.241921895599668
663	0.444444444444444	-0.955555555555556	-0.0572401270657721	-0.0348994967025012
673	0.511111111111111	-0.622222222222222	1.0746845796784	0.939692620785908
683	0.555555555555556	-0.844444444444444	0.720527863283385	0.615661475325658
693	0.622222222222222	-0.711111111111111	1.06448653150324	0.961261695938319
703	0.688888888888889	-0.711111111111111	1.11392969083966	0.997564050259824
713	0.755555555555556	-0.844444444444444	1.24089039983814	0.961261695938319
723	0.844444444444444	-0.955555555555556	1.08409153331539	0.939692620785908
1197	0.377777777777778	-0.355555555555556	1.04156584374708	0.997564050259824
1207	0.0888888888888889	-0.0444444444444444	0.974887391838504	0.99026806874157
1217	0.755555555555556	-0.711111111111111	1.12922461565003	0.99026806874157
1227	0.511111111111111	-0.422222222222222	1.02922133703998	0.961261695938319
1237	0.288888888888889	-0.177777777777778	0.936644364724831	0.939692620785908
1247	0.955555555555556	-0.844444444444444	2.04235792047483	0.939692620785908
1257	0.777777777777778	-0.622222222222222	1.07353782365702	0.882947592858927
1267	0.622222222222222	-0.444444444444444	0.859639941031916	0.848048096156426
1277	0.511111111111111	-0.288888888888889	0.79192756716057	0.766044443118978
1287	0.422222222222222	-0.177777777777778	0.713882568143066	0.719339800338651
1297	0.377777777777778	-0.0888888888888889	0.603146345554874	0.615661475325658
1307	0.355555555555556	-0.0444444444444444	0.529000660855693	0.559192903470747
1317	0.377777777777778	-0.0222222222222222	0.399101356769028	0.438371146789077
1327	0.422222222222222	-0.0444444444444444	0.337223768558886	0.374606593415912
1337	0.511111111111111	-0.0888888888888889	0.229241364699836	0.241921895599668
1347	0.622222222222222	-0.177777777777778	0.0906478567663862	0.17364817766693
1357	0.777777777777778	-0.288888888888889	-0.0656389038350726	0.0348994967025009
1367	0.955555555555556	-0.444444444444444	-0.513017911492948	-0.0348994967025012
1377	0.755555555555556	-0.177777777777778	-0.482311185968698	-0.241921895599668
1387	0.688888888888889	-0.0444444444444444	-0.496948648439881	-0.438371146789078
1397	0.755555555555556	-0.0444444444444444	-0.774025257738773	-0.615661475325658
1407	0.955555555555556	-0.177777777777778	-2.00039266115931	-0.766044443118978
1881	0.0444444444444444	0.0222222222222222	0.942927144366211	0.978147600733806
1891	0.711111111111111	0.0222222222222222	-0.802351282757724	-0.669130606358858
1901	0.422222222222222	0.0444444444444444	0.0576556268731376	0.104528463267653
1911	0.177777777777778	0.0888888888888889	0.648218658972098	0.669130606358858

1921	0.844444444444444	0.0888888888888889	-1.43035781236903	-0.978147600733806
1931	0.622222222222222	0.111111111111111	-0.688325921357063	-0.669130606358858
1941	0.444444444444444	0.155555555555556	-0.308182125349526	-0.309016994374947
1951	0.288888888888889	0.177777777777778	0.0992352326344036	0.104528463267653
1961	0.955555555555556	0.177777777777778	-2.33026045704268	-0.913545457642601
1971	0.844444444444444	0.222222222222222	-1.13490601319086	-0.978147600733806
1981	0.755555555555556	0.244444444444444	-0.967013830591618	-1
1991	0.711111111111111	0.288888888888889	-0.936506670966081	-1
2001	0.688888888888889	0.311111111111111	-0.96156761282191	-1
2011	0.711111111111111	0.355555555555556	-0.884289733380429	-0.978147600733806
2021	0.755555555555556	0.377777777777778	-0.829690259083642	-0.913545457642601
2031	0.844444444444444	0.422222222222222	-0.399754017783207	-0.669130606358858
2041	0.955555555555556	0.444444444444444	0.168879370986177	-0.309016994374948
2051	0.622222222222222	0.511111111111111	-0.841352246563409	-0.913545457642601
2061	0.844444444444444	0.555555555555556	0.183345535620069	-0.309016994374948
2071	0.711111111111111	0.622222222222222	-0.325354951892248	-0.5
2081	0.711111111111111	0.688888888888889	-0.0968704575528709	-0.309016994374948
2091	0.844444444444444	0.755555555555556	0.903553848370204	0.309016994374947
2101	0.955555555555556	0.844444444444444	2.9568928820796	0.809016994374947
2575	0.355555555555556	0.377777777777778	-0.665422507629421	-0.669130606358858
2585	0.0444444444444444	0.0888888888888889	0.871393007646949	0.913545457642601
2595	0.711111111111111	0.755555555555556	0.150122908459935	-0.104528463267653
2605	0.422222222222222	0.511111111111111	-0.956995107983458	-0.978147600733806
2615	0.177777777777778	0.288888888888889	0.0916468342652035	0.104528463267653
2625	0.844444444444444	0.955555555555556	1.40808804034952	0.809016994374947
2635	0.622222222222222	0.777777777777778	-0.196598532602441	-0.309016994374948
2645	0.444444444444444	0.622222222222222	-0.910726279235113	-0.978147600733806
2655	0.288888888888889	0.511111111111111	-0.831677037608638	-0.809016994374947
2665	0.177777777777778	0.422222222222222	-0.33057592493605	-0.309016994374947
2675	0.0888888888888889	0.377777777777778	0.0276585690004263	0.104528463267653
2685	0.0444444444444444	0.355555555555556	0.246935070160594	0.309016994374947
2695	0.0222222222222222	0.377777777777778	0.275143655695823	0.309016994374947
2705	0.0444444444444444	0.422222222222222	0.0350807073221236	0.104528463267653
2715	0.0888888888888889	0.511111111111111	-0.405319640235736	-0.309016994374947
2725	0.177777777777778	0.622222222222222	-0.859915250680018	-0.809016994374947
2735	0.288888888888889	0.777777777777778	-1.12771735378188	-0.978147600733806
2745	0.444444444444444	0.955555555555556	-0.159114339471001	-0.309016994374948
2755	0.177777777777778	0.755555555555556	-1.09063927280549	-0.978147600733806
2765	0.0444444444444444	0.688888888888889	-0.763819971933775	-0.669130606358858
2775	0.0444444444444444	0.755555555555556	-0.919048973757975	-0.809016994374947
2785	0.177777777777778	0.955555555555556	-1.33396680316765	-0.913545457642601
3259	-0.0222222222222222	0.0444444444444444	0.996529257419397	0.997564050259824
3269	-0.0222222222222222	0.711111111111111	-0.750102638345046	-0.559192903470747
3279	-0.0444444444444444	0.422222222222222	0.341450413303974	0.374606593415912
3289	-0.0888888888888889	0.177777777777778	0.95863837497041	0.961261695938319
3299	-0.0888888888888889	0.844444444444444	-0.903773721043573	-0.719339800338651
3309	-0.111111111111111	0.622222222222222	-0.0881991181534378	-0.0348994967025007
3319	-0.155555555555556	0.444444444444444	0.654980667818601	0.615661475325658
3329	-0.177777777777778	0.288888888888889	0.940388166751458	0.939692620785908
3339	-0.177777777777778	0.955555555555556	-1.67126455627444	-0.766044443118978
3349	-0.222222222222222	0.844444444444444	-0.355811603773999	-0.374606593415912
3359	-0.244444444444444	0.755555555555556	-0.0277736254841229	-0.0348994967025007
3369	-0.288888888888889	0.711111111111111	0.363981754323673	0.241921895599668
3379	-0.311111111111111	0.688888888888889	0.441152792859012	0.374606593415912
3389	-0.355555555555556	0.711111111111111	0.599044313012916	0.438371146789077
3399	-0.377777777777778	0.755555555555556	0.472892684747238	0.374606593415912
3409	-0.422222222222222	0.844444444444444	0.41816123969012	0.241921895599668
3419	-0.444444444444444	0.955555555555556	-0.402847203378901	-0.0348994967025012
3429	-0.511111111111111	0.622222222222222	1.04665640955226	0.939692620785908
3439	-0.555555555555556	0.844444444444444	0.904745875038915	0.615661475325658
3449	-0.622222222222222	0.711111111111111	1.07963339299785	0.961261695938319
3459	-0.688888888888889	0.711111111111111	1.03486571309702	0.997564050259824
3469	-0.755555555555556	0.844444444444444	1.1918571052556	0.961261695938319
3479	-0.844444444444444	0.955555555555556	1.46794415840284	0.939692620785908
3953	-0.377777777777778	0.355555555555556	1.00996388545905	0.997564050259824
3963	-0.0888888888888889	0.0444444444444444	0.978407249452116	0.99026806874157
3973	-0.755555555555556	0.711111111111111	0.96536558993191	0.99026806874157
3983	-0.511111111111111	0.422222222222222	0.972412856941469	0.961261695938319
3993	-0.288888888888889	0.177777777777778	0.929452623214791	0.939692620785908
4003	-0.955555555555556	0.844444444444444	0.77278599582384	0.939692620785908
4013	-0.777777777777778	0.622222222222222	0.877740135474826	0.882947592858927
4023	-0.622222222222222	0.444444444444444	0.84513676828439	0.848048096156426
4033	-0.511111111111111	0.288888888888889	0.768362142870532	0.766044443118978
4043	-0.422222222222222	0.177777777777778	0.707664869857216	0.719339800338651

4053	-0.377777777777778	0.0888888888888889	0.607742067037021	0.615661475325658
4063	-0.355555555555556	0.0444444444444444	0.555264246296504	0.559192903470747
4073	-0.377777777777778	0.0222222222222222	0.43476146562073	0.438371146789077
4083	-0.422222222222222	0.0444444444444444	0.37052224875511	0.374606593415912
4093	-0.511111111111111	0.0888888888888889	0.235273700373895	0.241921895599668
4103	-0.622222222222222	0.177777777777778	0.168411843911232	0.17364817766693
4113	-0.777777777777778	0.288888888888889	0.0375377439884344	0.0348994967025009
4123	-0.955555555555556	0.444444444444444	-0.0358626150039799	-0.0348994967025012
4133	-0.755555555555556	0.177777777777778	-0.237059826031569	-0.241921895599668
4143	-0.688888888888889	0.0444444444444444	-0.442581901791509	-0.438371146789078
4153	-0.755555555555556	0.0444444444444444	-0.620997328184706	-0.615661475325658
4163	-0.955555555555556	0.177777777777778	-0.761714074626367	-0.766044443118978
4637	-0.0444444444444444	-0.0222222222222222	0.971994517951197	0.978147600733806
4647	-0.711111111111111	-0.0222222222222222	-0.662057279811124	-0.669130606358858
4657	-0.422222222222222	-0.0444444444444444	0.101860937481622	0.104528463267653
4667	-0.177777777777778	-0.0888888888888889	0.667585613387358	0.669130606358858
4677	-0.844444444444444	-0.0888888888888889	-0.973127178474949	-0.978147600733806
4687	-0.622222222222222	-0.111111111111111	-0.659899816864426	-0.669130606358858
4697	-0.444444444444444	-0.155555555555556	-0.2989950886684	-0.309016994374947
4707	-0.288888888888889	-0.177777777777778	0.108167141635059	0.104528463267653
4717	-0.955555555555556	-0.177777777777778	-0.904110767798934	-0.913545457642601
4727	-0.844444444444444	-0.222222222222222	-0.973921381187542	-0.978147600733806
4737	-0.755555555555556	-0.244444444444444	-0.988340929386173	-1
4747	-0.711111111111111	-0.288888888888889	-0.989919527799618	-1
4757	-0.688888888888889	-0.311111111111111	-0.987469972267835	-1
4767	-0.711111111111111	-0.355555555555556	-0.968984996394387	-0.978147600733806
4777	-0.755555555555556	-0.377777777777778	-0.902874775787302	-0.913545457642601
4787	-0.844444444444444	-0.422222222222222	-0.668187060310035	-0.669130606358858
4797	-0.955555555555556	-0.444444444444444	-0.320379806247976	-0.309016994374948
4807	-0.622222222222222	-0.511111111111111	-0.898967562189266	-0.913545457642601
4817	-0.844444444444444	-0.555555555555556	-0.31088069554677	-0.309016994374948
4827	-0.711111111111111	-0.622222222222222	-0.498668601248311	-0.5
4837	-0.711111111111111	-0.688888888888889	-0.310072652908308	-0.309016994374948
4847	-0.844444444444444	-0.755555555555556	0.303589565794245	0.309016994374947
4857	-0.955555555555556	-0.844444444444444	0.782757323262625	0.809016994374947
5302	-0.355555555555556	-0.377777777777778	-0.650302563245026	-0.669130606358858
5312	-0.0444444444444444	-0.0888888888888889	0.915980831449001	0.913545457642601
5322	-0.711111111111111	-0.755555555555556	-0.107752766428576	-0.104528463267653
5332	-0.422222222222222	-0.511111111111111	-0.956406437173747	-0.978147600733806
5342	-0.177777777777778	-0.288888888888889	0.117953650247351	0.104528463267653
5352	-0.844444444444444	-0.955555555555556	0.788694220881374	0.809016994374947
5362	-0.622222222222222	-0.777777777777778	-0.303712303692265	-0.309016994374948
5372	-0.444444444444444	-0.622222222222222	-0.962520791986144	-0.978147600733806
5382	-0.288888888888889	-0.511111111111111	-0.785816018234096	-0.809016994374947
5392	-0.177777777777778	-0.422222222222222	-0.291874629941771	-0.309016994374947
5402	-0.0888888888888889	-0.377777777777778	0.123046210820995	0.104528463267653
5412	-0.0444444444444444	-0.355555555555556	0.324912229653107	0.309016994374947
5422	-0.0222222222222222	-0.377777777777778	0.351181359865799	0.309016994374947
5432	-0.0444444444444444	-0.422222222222222	0.123976690018172	0.104528463267653
5442	-0.0888888888888889	-0.511111111111111	-0.277701051467042	-0.309016994374947
5452	-0.177777777777778	-0.622222222222222	-0.800085611862491	-0.809016994374947
5462	-0.288888888888889	-0.777777777777778	-0.968608313669566	-0.978147600733806
5472	-0.444444444444444	-0.955555555555556	-0.322463130466913	-0.309016994374948
5482	-0.177777777777778	-0.755555555555556	-0.97249524853658	-0.978147600733806
5492	-0.0444444444444444	-0.688888888888889	-0.656135866334549	-0.669130606358858
5502	-0.0444444444444444	-0.755555555555556	-0.793643482779375	-0.809016994374947
5512	-0.177777777777778	-0.955555555555556	-0.89908148929605	-0.913545457642601

TABLE-5c

ELLIPTIC BOUNDARY VALUE PROBLEM:Example 5: (Output for 1/10th of the FEM MODEL)

Consider the boundary value problem: Find u such that

$$-\epsilon\ \ \Delta u + 2\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = f \quad \text{in } \Omega = [-1,1]^2$$

$$\mathbf{u}(1,y) = 0, -1 \leqslant y \leqslant 1$$

$$\mathbf{u}(x,1) = 0, -1 \leqslant x \leqslant 1$$

$$\left.\begin{aligned} \mathbf{u}(-1,y) &= (1 - e^{\frac{-2}{\epsilon}})(1 - e^{\frac{-(1-y)}{\epsilon}})\cos(\pi(-1+y)), -1 \leqslant y \leqslant 1 \\ \mathbf{u}(x,-1) &= (1 - e^{\frac{-(1-x)}{\epsilon}})(1 - e^{\frac{-2}{\epsilon}})\cos(\pi(x-1)), -1 \leqslant x \leqslant 1 \end{aligned}\right\}$$

Where, $\epsilon = 10^{-3}$ and f is chosen appropriately such that the exact solution is

DOI: 10.18535/ijecs/v5i12.75

$u(x,y)= (1 - e^{\frac{-(1-x)}{\epsilon}}) (1 - e^{\frac{-(1-y)}{\epsilon}}) \cos(\pi(x+y))$

ALL QUADRILATERAL FEM MODEL : number of nodes=7921,elements=7776 & nodes per element=4

Node no.	x-coordinate	y-coordinate	FEM computed values	Analytical Values
710	0.0185185185185185	-0.037037037037037	0.988738593990048	0.998308158271268
720	0.0185185185185185	-0.592592592592593	-0.218973566508674	-0.23061587074244
730	0.037037037037037	-0.185185185185185	0.892819473611032	0.893632640323412
740	0.037037037037037	-0.740740740740741	-0.596976520742453	-0.597158591702786
750	0.0740740740740741	-0.37037037037037	0.604926347756676	0.597158591702786
760	0.0740740740740741	-0.925925925925926	-0.888814130334214	-0.893632640323412
770	0.0925925925925926	-0.574074074074074	0.0748543179075226	0.0581448289104759
780	0.12962962962963	-0.259259259259259	0.917974523990793	0.918216106880274
790	0.12962962962963	-0.814814814814815	-0.544705949465855	-0.549508978070806
800	0.148148148148148	-0.518518518518518	0.415329103214261	0.396079766039157
810	0.185185185185185	-0.259259259259259	0.972904050983572	0.973044870579824
820	0.185185185185185	-0.814814814814815	-0.390116728326068	-0.396079766039156
830	0.203703703703704	-0.574074074074074	0.417282426610194	0.396079766039157
840	0.240740740740741	-0.37037037037037	0.923674683996025	0.918216106880274
850	0.240740740740741	-0.925925925925926	-0.544442964162362	-0.549508978070806
860	0.259259259259259	-0.740740740740741	0.0623909116892624	0.0581448289104759
870	0.296296296296296	-0.592592592592593	0.610054077655305	0.597158591702786
880	0.314814814814815	-0.462962962962963	0.915353362487409	0.893632640323412
890	0.351851851851852	-0.37037037037037	1.00028349983889	0.998308158271268
900	0.351851851851852	-0.925925925925926	-0.225732738989481	-0.23061587074244
910	0.37037037037037	-0.851851851851852	0.0479486182632554	0.0581448289104759
920	0.407407407407407	-0.814814814814815	0.303566452639882	0.28680323271109
930	0.425925925925926	-0.796296296296296	0.396628746893631	0.396079766039157
940	0.462962962962963	-0.814814814814815	0.478208608739483	0.448799180200462
950	0.481481481481481	-0.851851851851852	0.376465172172974	0.396079766039157
960	0.518518518518518	-0.925925925925926	0.288550598049443	0.28680323271109
970	0.574074074074074	-0.592592592592593	1.02869967777312	0.998308158271268
980	0.592592592592593	-0.740740740740741	0.99593368394477	0.893632640323412
990	0.62962962962963	-0.925925925925926	0.606141567979186	0.597158591702786
1000	0.685185185185185	-0.814814814814815	1.02396907724894	0.918216106880274
1010	0.740740740740741	-0.814814814814815	1.09357732645586	0.973044870579824
1020	0.796296296296296	-0.925925925925926	1.18971170311874	0.918216106880274
1030	0.907407407407407	-0.925925925925926	1.68412697637982	0.998308158271268
1705	0.37037037037037	-0.351851851851852	1.01566641955964	0.998308158271268
1715	0.925925925925926	-0.907407407407407	2.00570299986455	0.998308158271268
1725	0.518518518518518	-0.481481481481481	1.00349606064947	0.993238357741943
1735	0.148148148148148	-0.0740740740740741	0.966319457616082	0.973044870579824
1745	0.703703703703704	-0.62962962962963	1.08400653133775	0.973044870579824
1755	0.351851851851852	-0.259259259259259	0.95754485926884	0.957989512315489
1765	0.907407407407407	-0.814814814814815	1.36578288379851	0.957989512315489
1775	0.592592592592593	-0.462962962962963	0.965194482167668	0.918216106880274
1785	0.296296296296296	-0.148148148148148	0.886693591882759	0.893632640323412
1795	0.851851851851852	-0.703703703703704	1.03297740488251	0.893632640323412
1805	0.592592592592593	-0.407407407407407	0.870509447541328	0.835487811412936
1815	0.351851851851852	-0.148148148148148	0.793519852022072	0.802123192755044
1825	0.907407407407407	-0.703703703703704	1.03686522819998	0.802123192755044
1835	0.703703703703704	-0.462962962962963	0.785132609242851	0.727373641573049
1845	0.518518518518518	-0.259259259259259	0.675403197029711	0.686241637868734
1855	0.37037037037037	-0.0740740740740741	0.583329863502247	0.597158591702786
1865	0.925925925925926	-0.62962962962963	0.869489891627548	0.597158591702786
1875	0.796296296296296	-0.481481481481481	0.537895595668345	0.549508978070806
1885	0.703703703703704	-0.351851851851852	0.465984641519684	0.448799180200462
1895	0.62962962962963	-0.259259259259259	0.37017724959609	0.396079766039157
1905	0.592592592592593	-0.185185185185185	0.274900127169576	0.28680323271109
1915	0.574074074074074	-0.148148148148148	0.208572842660786	0.23061587074244
1925	0.592592592592593	-0.12962962962963	0.0993162254251576	0.116092914125231
1935	0.62962962962963	-0.148148148148148	0.0226558923477	0.0581448289104759
1945	0.703703703703704	-0.185185185185185	-0.107184177857561	-0.058144828910476
1955	0.796296296296296	-0.259259259259259	-0.224603193146767	-0.11609291412523
1965	0.925925925925926	-0.351851851851852	-0.601651708862301	-0.23061587074244
1975	0.703703703703704	-0.0740740740740741	-0.442145637186119	-0.396079766039157
1985	0.907407407407407	-0.259259259259259	-0.879247870027242	-0.448799180200462
1995	0.851851851851852	-0.148148148148148	-0.884253870075065	-0.597158591702786
2005	0.907407407407407	-0.148148148148148	-1.26945258850652	-0.727373641573049
2015	0.925925925925926	-0.0740740740740741	-2.0727143170652	-0.893632640323412
2690	0.148148148148148	0.0185185185185185	0.845174611194463	0.866025403784439

2700	0.703703703703704	0.0185185185185185	-0.711788186109186	-0.642787609686539
2710	0.296296296296296	0.037037037037037	0.474707386695566	0.5
2720	0.851851851851852	0.037037037037037	-1.21842528206631	-0.939692620785908
2730	0.481481481481481	0.0740740740740741	-0.186272513051098	-0.17364817766693
2740	0.12962962962963	0.0925925925925926	0.758780737721252	0.766044443118978
2750	0.685185185185185	0.0925925925925926	-0.778295505044476	-0.766044443118978
2760	0.37037037037037	0.12962962962963	-0.00740967364567281	6.12323399573677e-017
2770	0.925925925925926	0.12962962962963	-2.12099969889357	-0.984807753012208
2780	0.62962962962963	0.148148148148148	-0.764817182688495	-0.766044443118978
2790	0.37037037037037	0.185185185185185	-0.171108366708092	-0.17364817766693
2800	0.925925925925926	0.185185185185185	-1.88663741949832	-0.939692620785908
2810	0.685185185185185	0.203703703703704	-0.925831034979431	-0.939692620785908
2820	0.481481481481481	0.240740740740741	-0.629241024863172	-0.642787609686539
2830	0.296296296296296	0.259259259259259	-0.176142177797188	-0.17364817766693
2840	0.851851851851852	0.259259259259259	-0.934716453405105	-0.939692620785908
2850	0.703703703703704	0.296296296296296	-0.968945338864568	-1
2860	0.574074074074074	0.314814814814815	-0.921382360293863	-0.939692620785908
2870	0.481481481481481	0.351851851851852	-0.840026893772854	-0.866025403784438
2880	0.407407407407407	0.37037037037037	-0.756758663345591	-0.766044443118978
2890	0.962962962962963	0.37037037037037	-0.762077348709821	-0.5
2900	0.925925925925926	0.407407407407407	-0.464904413155998	-0.5
2910	0.907407407407407	0.425925925925926	-0.351835565524831	-0.5
2920	0.925925925925926	0.462962962962963	-0.0384837889565263	-0.342020143325669
2930	0.962962962962963	0.481481481481481	0.312558352367042	-0.17364817766693
2940	0.574074074074074	0.537037037037037	-0.906146528244671	-0.939692620785908
2950	0.703703703703704	0.574074074074074	-0.567525864950109	-0.642787609686539
2960	0.851851851851852	0.592592592592593	-0.0139826521216357	-0.17364817766693
2970	0.685185185185185	0.648148148148148	-0.437737484333029	-0.5
2980	0.925925925925926	0.685185185185185	1.52840793640971	0.342020143325669
2990	0.925925925925926	0.740740740740741	1.82344558308076	0.499999999999999
3000	0.851851851851852	0.814814814814815	0.788544632396141	0.499999999999999
3010	0.962962962962963	0.925925925925926	2.69734220971997	0.939692620785908
3685	0.462962962962963	0.481481481481481	-0.966453845231915	-0.984807753012208
3695	0.037037037037037	0.0740740740740741	0.920590444343819	0.939692620785908
3705	0.592592592592593	0.62962962962963	-0.692842570027991	-0.766044443118978
3715	0.185185185185185	0.259259259259259	0.162262583816335	0.17364817766693
3725	0.740740740740741	0.814814814814815	0.336707702119602	0.17364817766693
3735	0.37037037037037	0.462962962962963	-0.844053719293821	-0.866025403784438
3745	0.0185185185185185	0.148148148148148	0.861151688797072	0.866025403784439
3755	0.574074074074074	0.703703703703704	-0.595217998650458	-0.642787609686539
3765	0.259259259259259	0.407407407407407	-0.494685498413765	-0.5
3775	0.814814814814815	0.962962962962963	1.42771818752607	0.766044443118978
3785	0.518518518518518	0.703703703703704	-0.737515825935198	-0.766044443118978
3795	0.259259259259259	0.462962962962963	-0.637564236988763	-0.642787609686539
3805	0.0185185185185185	0.259259259259259	0.632188919201481	0.642787609686539
3815	0.574074074074074	0.814814814814815	-0.301567612038986	-0.342020143325669
3825	0.37037037037037	0.62962962962963	-0.975685824525192	-1
3835	0.185185185185185	0.481481481481481	-0.523448021571195	-0.5
3845	0.037037037037037	0.351851851851852	0.318015932812359	0.342020143325669
3855	0.592592592592593	0.907407407407407	0.18456841082055	-1.83697019872103e-016
3865	0.462962962962963	0.814814814814815	-0.667205943913616	-0.642787609686539
3875	0.37037037037037	0.740740740740741	-0.925723086239699	-0.939692620785908
3885	0.296296296296296	0.703703703703704	-1.04500223352022	-1
3895	0.259259259259259	0.685185185185185	-0.999550701145863	-0.984807753012208
3905	0.240740740740741	0.703703703703704	-1.04799184485728	-0.984807753012208
3915	0.259259259259259	0.740740740740741	-1.03569783888334	-1
3925	0.296296296296296	0.814814814814815	-1.05374403536758	-0.939692620785908
3935	0.37037037037037	0.907407407407407	-0.703932141108182	-0.642787609686539
3945	0.037037037037037	0.62962962963	-0.537875035957037	-0.5
3955	0.185185185185185	0.814814814814815	-1.14931029782071	-1
3965	0.0185185185185185	0.703703703703704	-0.834806107313225	-0.642787609686539
3975	0.259259259259259	0.962962962962963	-1.09653299471521	-0.766044443118978
3985	0.0185185185185185	0.814814814814815	-1.23272236073055	-0.866025403784438
3995	0.037037037037037	0.907407407407407	-1.18042083673317	-0.984807753012208
4670	-0.0185185185185185	0.259259259259259	0.722330777285489	0.727373641573049
4680	-0.0185185185185185	0.814814814814815	-1.06126421613284	-0.802123192755044
4690	-0.037037037037037	0.407407407407407	0.374667562782388	0.396079766039157
4700	-0.037037037037037	0.962962962962963	-1.93048532258596	-0.973044870579824
4710	-0.0740740740740741	0.592592592592593	-0.0710398014832935	-0.0581448289104758
4720	-0.0925925925925926	0.240740740740741	0.888005427006229	0.893632640323412
4730	-0.0925925925925926	0.796296296296296	-0.746285042066899	-0.597158591702786
4740	-0.12962962962963	0.481481481481481	0.460060980355649	0.448799180200462
4750	-0.148148148148148	0.185185185185185	0.988235651330769	0.993238357741943
4760	-0.148148148148148	0.740740740740741	-0.340819175147122	-0.28680323271109

4770	-0.185185185185185	0.481481481481481	0.616218248101861	0.597158591702786
4780	-0.203703703703704	0.240740740740741	0.987954858961705	0.993238357741943
4790	-0.203703703703704	0.796296296296296	-0.333383309552947	-0.28680323271109
4800	-0.240740740740741	0.592592592592593	0.483845904589272	0.448799180200462
4810	-0.259259259259259	0.407407407407407	0.90239162625299	0.893632640323412
4820	-0.259259259259259	0.962962962962963	-1.17150177709714	-0.597158591702786
4830	-0.296296296296296	0.814814814814815	0.00928160842553703	-0.0581448289104758
4840	-0.314814814814815	0.685185185185185	0.425165663882051	0.396079766039157
4850	-0.351851851851852	0.592592592592593	0.777158443076255	0.727373641573049
4860	-0.37037037037037	0.518518518518518	0.918788151391179	0.893632640323412
4870	-0.407407407407407	0.481481481481481	0.988387812139287	0.973044870579824
4880	-0.425925925925926	0.462962962962963	1.00666638794819	0.993238357741943
4890	-0.462962962962963	0.481481481481481	1.00806836750628	0.998308158271268
4900	-0.481481481481481	0.518518518518518	1.01334359301346	0.993238357741943
4910	-0.518518518518518	0.592592592592593	1.00326572327849	0.973044870579824
4920	-0.537037037037037	0.685185185185185	0.97948440226043	0.893632640323412
4930	-0.574074074074074	0.814814814814815	0.921035687632124	0.727373641573049
4940	-0.592592592592593	0.962962962962963	0.42413975492173	0.396079766039157
4950	-0.648148148148148	0.796296296296296	1.07543595292789	0.893632640323412
4960	-0.703703703703704	0.740740740740741	1.08344781976349	0.993238357741943
4970	-0.759259259259259	0.796296296296296	1.1274408471956	0.993238357741943
4980	-0.814814814814815	0.962962962962963	1.36825677051625	0.893632640323412
5655	-0.037037037037037	0.0185185185185185	1.00046766133971	0.998308158271268
5665	-0.592592592592593	0.574074074074074	1.01851239394872	0.998308158271268
5675	-0.185185185185185	0.148148148148148	0.987862290345009	0.993238357741943
5685	-0.740740740740741	0.703703703703704	0.984424732836975	0.993238357741943
5695	-0.37037037037037	0.296296296296296	0.969990085045449	0.973044870579824
5705	-0.925925925925926	0.851851851851852	0.959618203751462	0.973044870579824
5715	-0.574074074074074	0.481481481481481	0.958839165439914	0.957989512315489
5725	-0.259259259259259	0.12962962962963	0.912952053877508	0.918216106880274
5735	-0.814814814814815	0.685185185185185	0.916507067743758	0.918216106880274
5745	-0.518518518518518	0.37037037037037	0.893736406608055	0.893632640323412
5755	-0.259259259259259	0.0740740740740741	0.830505064594171	0.835487811412936
5765	-0.814814814814815	0.62962962962963	0.832946303526534	0.835487811412937
5775	-0.574074074074074	0.37037037037037	0.801630889829753	0.802123192755044
5785	-0.37037037037037	0.12962962962963	0.721237552412335	0.727373641573049
5795	-0.925925925925926	0.685185185185185	0.71832020511989	0.727373641573049
5805	-0.740740740740741	0.481481481481481	0.683439989524505	0.686241637868734
5815	-0.592592592592593	0.296296296296296	0.598294721530041	0.597158591702786
5825	-0.462962962962963	0.148148148148148	0.542937585850115	0.549508978070806
5835	-0.37037037037037	0.0185185185185185	0.445710758159853	0.448799180200462
5845	-0.925925925925926	0.574074074074074	0.447454010859159	0.448799180200462
5855	-0.851851851851852	0.481481481481481	0.394389532363512	0.396079766039157
5865	-0.814814814814815	0.407407407407407	0.287830023614438	0.28680323271109
5875	-0.796296296296296	0.37037037037037	0.230792320644687	0.23061587074244
5885	-0.814814814814815	0.351851851851852	0.117512646042582	0.116092914125231
5895	-0.851851851851852	0.37037037037037	0.058783229952903	0.0581448289104759
5905	-0.925925925925926	0.407407407407407	-0.0565076392274952	-0.058144828910476
5915	-0.592592592592593	0.0185185185185185	-0.234056322352655	-0.23061587074244
5925	-0.740740740740741	0.148148148148148	-0.285914887991521	-0.28680323271109
5935	-0.925925925925926	0.296296296296296	-0.393904315531269	-0.396079766039157
5945	-0.814814814814815	0.12962962962963	-0.546583771097079	-0.549508978070806
5955	-0.814814814814815	0.0740740740740741	-0.689046848693535	-0.686241637868733
5965	-0.925925925925926	0.12962962962963	-0.799665134689236	-0.802123192755044
5975	-0.925925925925926	0.0185185185185185	-0.962456807252005	-0.957989512315489
6650	-0.37037037037037	-0.0185185185185185	0.338135323057117	0.342020143325669
6660	-0.925925925925926	-0.0185185185185185	-0.9741754602242	-0.984807753012208
6670	-0.518518518518518	-0.037037037037037	-0.173804528450831	-0.17364817766693
6680	-0.148148148148148	-0.0740740740740741	0.761174262679978	0.766044443118978
6690	-0.703703703703704	-0.0740740740740741	-0.760715278456665	-0.766044443118978
6700	-0.351851851851852	-0.0925925925925926	0.172013513831745	0.17364817766693
6710	-0.907407407407407	-0.0925925925925926	-0.996358874086111	-1
6720	-0.592592592592593	-0.12962962962963	-0.63716776054389	-0.642787609686539
6730	-0.296296296296296	-0.148148148148148	0.173607445707762	0.17364817766693
6740	-0.851851851851852	-0.148148148148148	-0.994511373551325	-1
6750	-0.592592592592593	-0.185185185185185	-0.759390474634218	-0.766044443118978
6760	-0.351851851851852	-0.203703703703704	-0.169499937750877	-0.17364817766693
6770	-0.907407407407407	-0.203703703703704	-0.936605236501873	-0.939692620785908
6780	-0.703703703703704	-0.240740740740741	-0.979133786523296	-0.984807753012208
6790	-0.518518518518518	-0.259259259259259	-0.757576196795748	-0.766044443118978
6800	-0.37037037037037	-0.296296296296296	-0.490766164856494	-0.5
6810	-0.925925925925926	-0.296296296296296	-0.768766538262843	-0.766044443118978
6820	-0.796296296296296	-0.314814814814815	-0.934402093438341	-0.939692620785908
6830	-0.703703703703704	-0.351851851851852	-0.979978680928689	-0.984807753012208

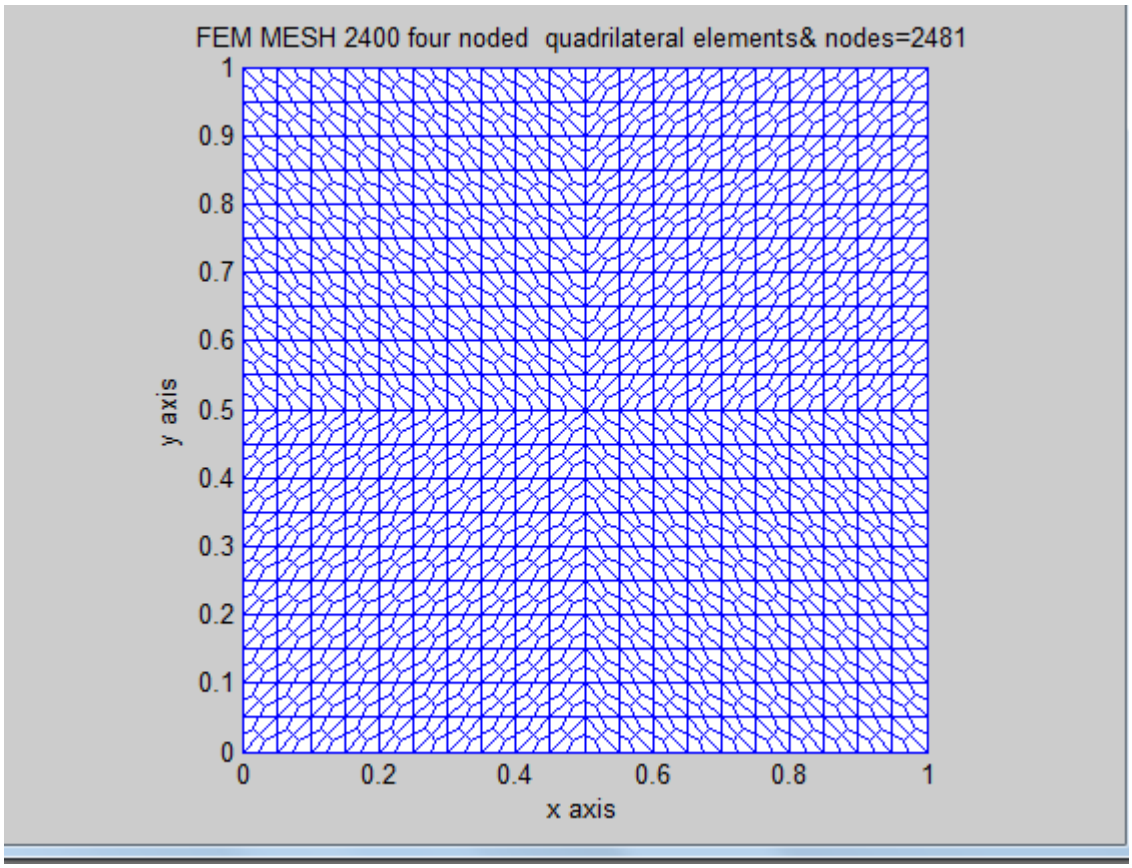
6840	-0.62962962962963	-0.37037037037037	-0.99113950301526	-1
6850	-0.592592592592593	-0.407407407407407	-0.992061144063107	-1
6860	-0.574074074074074	-0.425925925925926	-0.989648024892117	-1
6870	-0.592592592592593	-0.462962962962963	-0.977246823453055	-0.984807753012208
6880	-0.62962962962963	-0.481481481481481	-0.931312422074251	-0.939692620785908
6890	-0.703703703703704	-0.518518518518518	-0.763726995977279	-0.766044443118978
6900	-0.796296296296296	-0.537037037037037	-0.49809694649111	-0.5
6910	-0.925925925925926	-0.574074074074074	-0.0126847661015476	-1.83697019872103e-016
6920	-0.703703703703704	-0.62962962962963	-0.499822849613656	-0.5
6930	-0.907407407407407	-0.648148148148148	0.171429381825621	0.17364817766693
6940	-0.851851851851852	-0.703703703703704	0.167061626534633	0.17364817766693
6950	-0.907407407407407	-0.759259259259259	0.496606734699719	0.5
6960	-0.925925925925926	-0.851851851851852	0.752982276228234	0.766044443118978
7600	-0.12962962962963	-0.148148148148148	0.638526664654525	0.642787609686539
7610	-0.685185185185185	-0.703703703703704	-0.339540986656394	-0.342020143325669
7620	-0.259259259259259	-0.296296296296296	-0.165405029821146	-0.17364817766693
7630	-0.814814814814815	-0.851851851851852	0.49308587602289	0.499999999999999
7640	-0.407407407407407	-0.481481481481481	-0.92633925979159	-0.939692620785908
7650	-0.037037037037037	-0.12962962962963	0.862687622575078	0.866025403784439
7660	-0.592592592592593	-0.685185185185185	-0.638868206352092	-0.642787609686539
7670	-0.240740740740741	-0.37037037037037	-0.332759779006399	-0.342020143325668
7680	-0.796296296296296	-0.925925925925926	0.631506318290724	0.642787609686539
7690	-0.481481481481481	-0.62962962962963	-0.930418739862265	-0.939692620785908
7700	-0.185185185185185	-0.37037037037037	-0.165655781311669	-0.17364817766693
7710	-0.740740740740741	-0.925925925925926	0.488542105007613	0.499999999999999
7720	-0.481481481481481	-0.685185185185185	-0.857693860385852	-0.866025403784439
7730	-0.240740740740741	-0.481481481481481	-0.629450612685588	-0.642787609686539
7740	-0.037037037037037	-0.296296296296296	0.503296587033004	0.5
7750	-0.592592592592593	-0.851851851851852	-0.180794028586172	-0.17364817766693
7760	-0.407407407407407	-0.703703703703704	-0.928817282877505	-0.939692620785908
7770	-0.259259259259259	-0.574074074074074	-0.851958296872033	-0.866025403784438
7780	-0.12962962962963	-0.481481481481481	-0.32937819061505	-0.342020143325668
7790	-0.037037037037037	-0.407407407407407	0.182322679103155	0.17364817766693
7800	-0.592592592592593	-0.962962962962963	0.161965403844305	0.17364817766693
7810	-0.518518518518518	-0.925925925925926	-0.181649709911801	-0.17364817766693
7820	-0.481481481481481	-0.907407407407407	-0.350290958498218	-0.342020143325669
7830	-0.462962962962963	-0.925925925925926	-0.348361170227596	-0.342020143325669
7840	-0.481481481481481	-0.962962962962963	-0.183364424245975	-0.17364817766693
7850	-0.037037037037037	-0.574074074074074	-0.335559227555695	-0.342020143325669
7860	-0.12962962962963	-0.703703703703704	-0.856885077145297	-0.866025403784439
7870	-0.259259259259259	-0.851851851851852	-0.939830205420467	-0.939692620785908
7880	-0.037037037037037	-0.685185185185185	-0.637032148177635	-0.642787609686539
7890	-0.240740740740741	-0.925925925925926	-0.863000054953221	-0.866025403784439
7900	-0.185185185185185	-0.925925925925926	-0.933859825728083	-0.939692620785908
7910	-0.037037037037037	-0.851851851851852	-0.932410436647176	-0.939692620785908
7920	-0.037037037037037	-0.962962962962963	-0.991490692909287	-1

FINITE ELEMENT MESHES

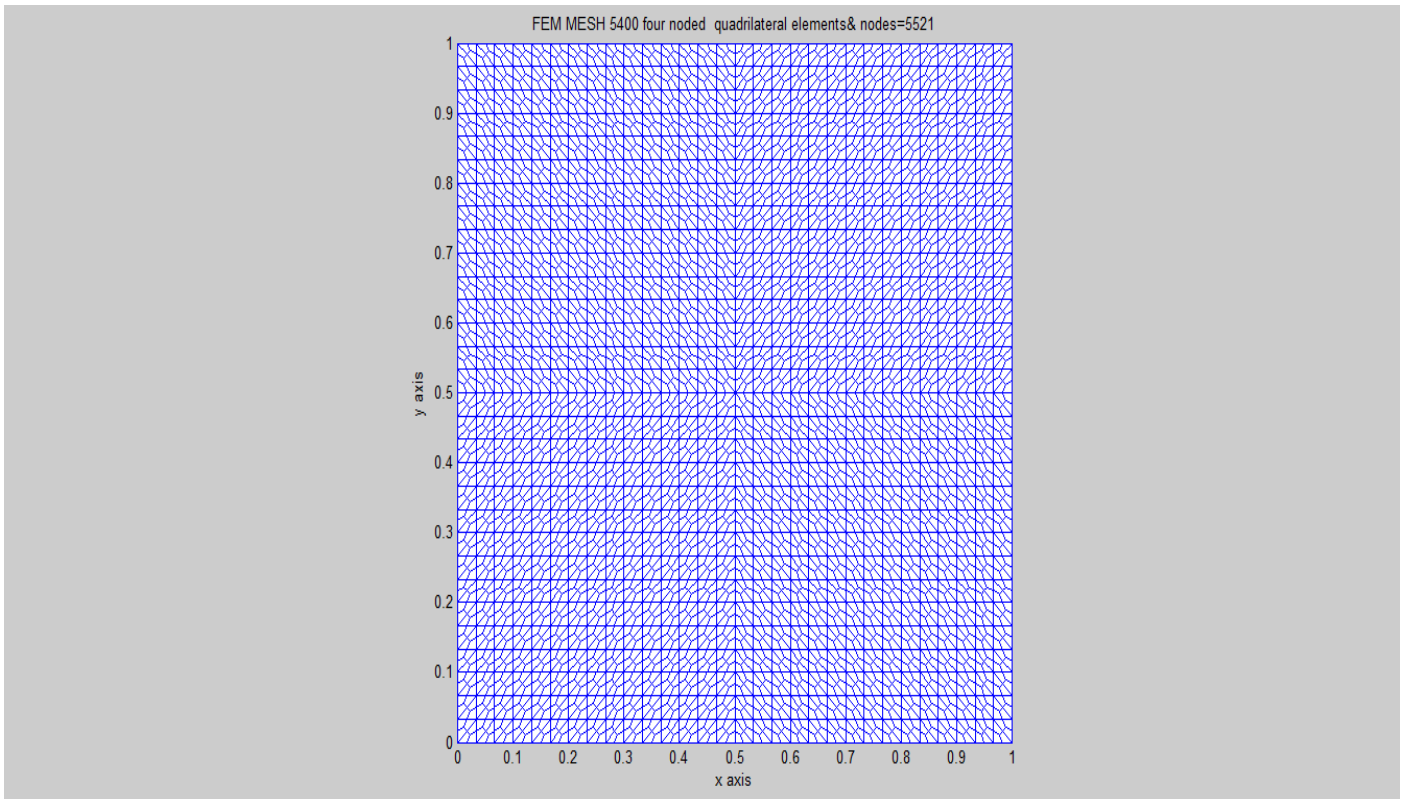
We have presented three FEM meshes over the 1-square $[0,1]^2$,the meshes over the 2-square $[-1,1]^2$,are similar and we feel that it is not necessary to display here.

MESH- 1 OVER THE DOMAIN $\Omega = [0,1]^2$

DOI: 10.18535/ijecs/v5i12.75

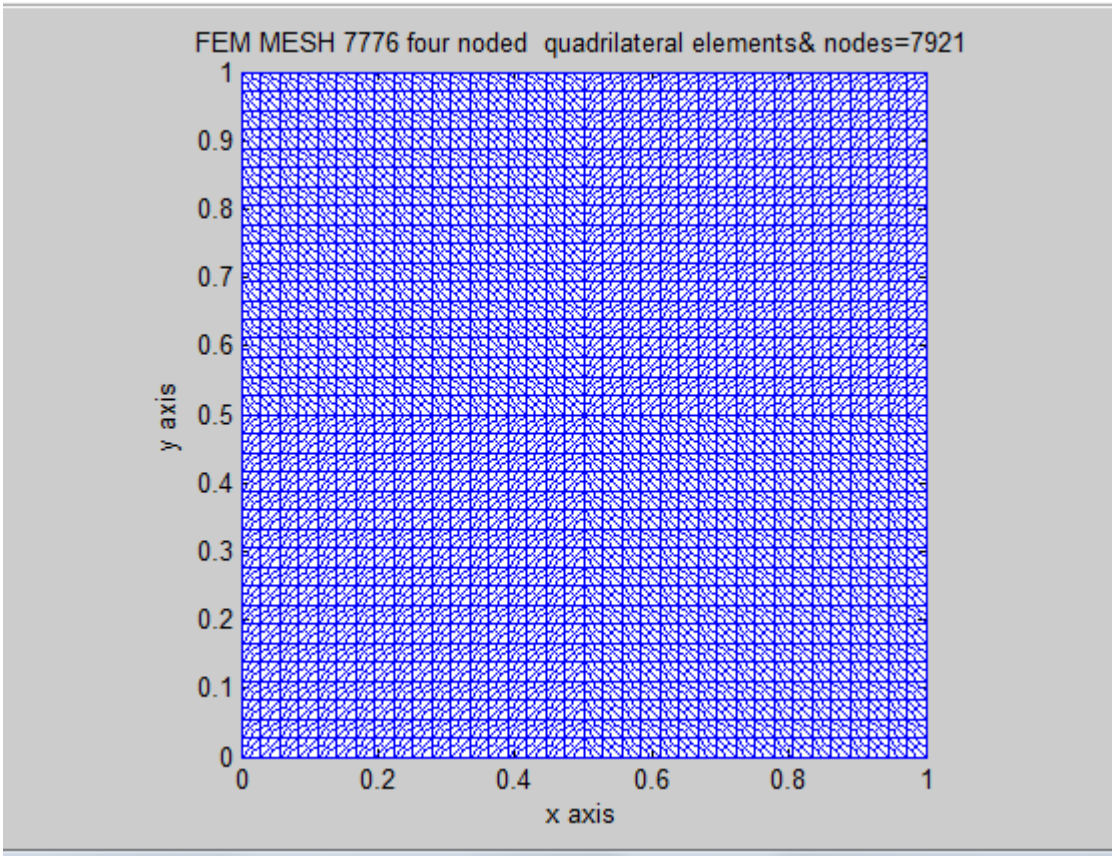


MESH- 2 OVER THE DOMAIN $\Omega = [0, 1]^2$



MESH- 3 OVER THE DOMAIN $\Omega = [0, 1]^2$

DOI: 10.18535/ijecs/v5i12.75



COMPUTER PROGRAMS

(1) D2LaplaceEquationQ4MoinExautomeshgenNINJ.m

```
function[NNC phic
xic]=D2LaplaceEquationQ4MoinExautomeshgenNINJ(n1,n2,n3,nmax,numtri,ndiv,mesh,lambda)
%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)
%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)
%function[]=quadrilateral_mesh4MOINEX_q4(n1,n2,n3,nmax,numtri,ndiv,mesh,xlength,ylength
)
%D2LaplaceEquationQ4MoinExautomeshgenNINJ([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;
7;8;9;2],9,1,2,12,lambda=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)
```

```
%D2LaplaceEquationQ4MoinExautomeshgenNINJ([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;
7;8;9;2],9,9,6,13,0.1);
global EDGEN1N2 EDGEN2N3 EDGEN1N3  ncrack npart  NMAX NITRI part nitri NE N1 N2 N3 rrr
TRINUM trifirst trilast qqfirst
syms coord
[npart,part,numsame,maxpart]=checksamenumbers(n1)
ncrack=0
[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates(n1,n2,n3,nmax,numtr
i,ndiv,mesh)
ntriel=0;
for triel=1:nel/3
    if ( (nodes(3*triel-2,1)==nodes(3*triel-1,1))&(nodes(3*triel-
1,1)==nodes(3*triel,1)) )
    ntriel=ntriel+1;
    centnode(ntriel,1)=nodes(3*triel,1);
    end
end
centnode
%
nnel=4;
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 1
           lambda1=1;lambda2=0;
    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
case 2
    lambda1=1;lambda2=0;
      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
  %=========================boundary conditions mesh-12(2-square) veronika
pillwein========
  case 12
      lambda1=lambda;lambda2=lambda;
      nnn=0;
    %boundary conditions 1
    for nn=1:nnode
        xnn=coord(nn,1);ynn=gcoord(nn,2);
        if (xnn==-1)&((ynn>=(-1))&(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==(-1))&((xnn>=(-1))&(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>=(-1))&(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>=(-1))&(ynn<=1))
            nnn=nnn+1;
```

```
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
    end
 %============================boundary conditions mesh-13 exponential product
solution==================
 case 13
    lambda1=(1/lambda)^2;lambda2=2;p=lambda;
      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
if (mesh ~= 13)
 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
if (mesh == 13)
 xi=(zeros(nnode,1));
for m=1:nnode
    xm=(gcoord(m,1));ym=(gcoord(m,2));
    U1=1-(exp((-xm/p))+exp(-(1-xm)/p))/(1+exp(-1/p));
```

```
    U2=1-(exp((-ym/p))+exp(-(1-ym)/p))/(1+exp(-1/p));
    xi(m,1)=(U1*U2)
end

end
 for L=1:nel
   for M=1:3
      LM=nodetel(L,M);
      xx(L,M)=gcoord(LM,1);
      yy(L,M)=gcoord(LM,2);
    end
  end
%_____________________________________________________________________
ng=10
 [sp,wt]=glsampleptsweights(ng)
table2=[N xx yy];
%disp([xx yy])
intJdn1dn1uvrs =[vpa(sym(' .595527655000821147485729267330')),
vpa(sym('.468322285820574645491168269290'));vpa(sym('
.468322285820574645491168269290')), vpa(sym('.595527655000821147485729267330'))];
intJdn1dn2uvrs =[vpa(sym('  -.397018436667214098323819511552')),
vpa(sym('.187785142786283569672554487 1395'));vpa(sym(' -
.312214857213716430327445512 8604')),   vpa(sym('.102981563332785901676180488448'))];
intJdn1dn3uvrs =[vpa(sym(' -.301490781666392950838090244 2235')),vpa(sym(' -
.343892571393141784836277243 5700'));vpa(sym(' -.343892571393141784836277243 5700')),
vpa(sym('-.301490781666392950838090244 2235'))];
intJdn1dn4uvrs =[vpa(sym('   .102981563332785901676180488448')), vpa(sym('-
.312214857213716430327445512 8604'));vpa(sym('  .187785142786283569672554487 1395')),
vpa(sym('-.397018436667214098323819511552'))];
%
intJdn2dn1uvrs =[vpa(sym('  -.397018436667214098323819511552')), vpa(sym('-
.312214857213716430327445512 8604'));vpa(sym('  .187785142786283569672554487 1395')),
vpa(sym(' .102981563332785901676180488448'))];
intJdn2dn2uvrs =[vpa(sym('   .264678957778142732215879674369')), vpa(sym('-
.125190095190855713115036324 7600'));vpa(sym(' -.125190095190855713115036324 7600')),
vpa(sym(' .264678957778142732215879674369'))];
intJdn2dn3uvrs =[vpa(sym('  .200993854444261967225393496 1491')),
vpa(sym('.229261714262094523224184829 0466'));vpa(sym(' -
.270738285737905476775815170 9534')), vpa(sym('-.299006145555738032774606503 8509'))];
intJdn2dn4uvrs =[vpa(sym(' -.686543755551906011174536589 65e-1')),
vpa(sym('.208143238142477620218297008 5734'));vpa(sym('
.208143238142477620218297008 5734')), vpa(sym('-.686543755551906011174536589 65e-1'))];
%
intJdn3dn1uvrs =[vpa(sym(' -.301490781666392950838090244 2235')), vpa(sym('-
.343892571393141784836277243 5700'));vpa(sym(' -
.343892571393141784836277243 5700')),vpa(sym(' -.301490781666392950838090244 2235'))];
intJdn3dn2uvrs =[vpa(sym('  .200993854444261967225393496 1491')), vpa(sym('-
.270738285737905476775815170 9534'));vpa(sym('  .229261714262094523224184829 0466')),
vpa(sym('-.299006145555738032774606503 8509'))];
intJdn3dn3uvrs =[vpa(sym(' .399503072777869016387303251 9254')),
vpa(sym('.385369142868952738387907585 4768'));vpa(sym('
.385369142868952738387907585 4768')), vpa(sym('.399503072777869016387303251 9254'))];
intJdn3dn4uvrs =[vpa(sym(' -.299006145555738032774606503 8509')), vpa(sym('
.229261714262094523224184829 0466'));vpa(sym(' -.270738285737905476775815170 9534')),
vpa(sym('.200993854444261967225393496 1491'))];
%
intJdn4dn1uvrs =[vpa(sym('   .102981563332785901676180488448')),
vpa(sym('.187785142786283569672554487 1395'));vpa(sym(' -
.312214857213716430327445512 8604')), vpa(sym(' -.397018436667214098323819511552'))];
intJdn4dn2uvrs =[vpa(sym(' -.686543755551906011174536589 65e-1')),  vpa(sym('
.208143238142477620218297008 5734'));vpa(sym('   .208143238142477620218297008 5734')),
vpa(sym('-.686543755551906011174536589 65e-1'))];
```

```
intJdn4dn3uvrs =[vpa(sym(' -.29900614555573803277460650385 09')), vpa(sym('-
.27073828573790547677581517095 34')));vpa(sym('  .22926171426209452322418482904 66')),
vpa(sym('.20099385444426196722539349614 91'))];
intJdn4dn4uvrs =[vpa(sym('   .26467895777814273221587967436 9')), vpa(sym('-
.12519009519085571311503632476 00'));vpa(sym(' -.12519009519085571311503632476 00')),
vpa(sym(' .26467895777814273221587967436 9'))];
%
%
intJdndn=double([intJdn1dn1uvrs intJdn1dn2uvrs intJdn1dn3uvrs intJdn1dn4uvrs;...
          intJdn2dn1uvrs intJdn2dn2uvrs intJdn2dn3uvrs intJdn2dn4uvrs;...
          intJdn3dn1uvrs intJdn3dn2uvrs intJdn3dn3uvrs intJdn3dn4uvrs;...
          intJdn4dn1uvrs intJdn4dn2uvrs intJdn4dn3uvrs intJdn4dn4uvrs]);

 IntJNiNj =[  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];



%
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(1:4,1:4)=(zeros(4,4));
for i=1:4
 for j=i:4
 sk(i,j)=(delabc*sum(sum(Q.*(intJdndn(2*i-1:2*i,2*j-
1:2*j)))))/lambda1+lambda2*delabc*IntJNiNj(i,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(1:4,1)=zeros(4,1)
 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=(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;
        if (mesh~=13)

f1i=n1ij*(((lambda2^2)+(2*pi^2))/lambda1)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;

f2i=n2ij*(((lambda2^2)+(2*pi^2))/lambda1)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;

f3i=n3ij*(((lambda2^2)+(2*pi^2))/lambda1)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;

f4i=n4ij*(((lambda2^2)+(2*pi^2))/lambda1)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;
        end
        if (mesh==13)
        fxy= 2 - (exp(xeij/p) + exp(yeij/p) + exp(1/p - xeij/p) + exp(1/p -
yeij/p))/(exp(1/p) + 1);
        f1i=n1ij*fxy*(4+si+sj)/96;
        f2i=n2ij*fxy*(4+si+sj)/96;
        f3i=n3ij*fxy*(4+si+sj)/96;
        f4i=n4ij*fxy*(4+si+sj)/96;
        end
          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;
      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]
[centnode]


%for I=1:nel/3
%II=centnode(I,1);
%NNC(I,1)=II;
%phic(I,1)=phi(II,1);
%xic(I,1)=xi(II,1);
%end

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

%disp([NNC phic xic])
%disp('_____________________________________________________________________
_____')
```

```
for I=1:nel/3
II=centnode(I,1);
NNC(I,1)=II;
xnnc(I,1)=gcoord(II,1);
ynnc(I,1)=gcoord(II,2);
phic(I,1)=phi(II,1);
xic(I,1)=xi(II,1);
end

disp('_____________________________________________________________________________________
_____________________________________')
disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
disp('_____________________________________________________________________________________
_____________________________________________________________________')
disp('                        node              x-coordinate            y-coordinate
fem-computed values      anlytical(theoretical)-values           ')
if (nel/3)<11
disp([NNC  xnnc ynnc  phic xic])
end
if (nel/3)>10
disp([NNC(1:10:nel/3,1)  xnnc(1:10:nel/3,1) ynnc(1:10:nel/3,1)  phic(1:10:nel/3,1)
xic(1:10:nel/3,1)])
end
disp('_____________________________________________________________________________________
_____________________________________________________________________')
```

(2) D2LaplaceEquationQ4MoinExautomeshgenNINJNiDNj.m

```
function[NNC phic
xic]=D2LaplaceEquationQ4MoinExautomeshgenNINJNiDNj(n1,n2,n3,nmax,numtri,ndiv,mesh,lambd
a)
%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)
%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)
%D2LaplaceEquationQ4MoinExautomeshgenNINJ([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;
7;8;9;2],9,1,2,12,lambda=1)
%D2LaplaceEquationQ4MoinExautomeshgenNINJNiDNj([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4
;5;6;7;8;9;2],9,1,2,14,.01)
global EDGEN1N2 EDGEN2N3 EDGEN1N3  ncrack npart  NMAX NITRI part nitri NE N1 N2 N3 rrr
TRINUM trifirst trilast qqfirst
syms coord
```

```
[npart,part,numsame,maxpart]=checksamenumbers(n1)
ncrack=0
[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates(n1,n2,n3,nmax,numtr
i,ndiv,mesh)
ntriel=0;
for triel=1:nel/3
    if ( (nodes(3*triel-2,1)==nodes(3*triel-1,1))&(nodes(3*triel-
1,1)==nodes(3*triel,1)) )
    ntriel=ntriel+1;
    centnode(ntriel,1)=nodes(3*triel,1);
    end
end
centnode
%
nnel=4;
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 1
           lambda1=1;lambda2=0;
```

```
    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
case 2
    lambda1=1;lambda2=0;
      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
  %=========================boundary conditions mesh-12(2-square)========
  case 12
      lambda1=lambda;lambda2=lambda;
      nnn=0;
    %boundary conditions 1
    for nn=1:nnode
        xnn=coord(nn,1);ynn=gcoord(nn,2);
        if (xnn==-1)&((ynn>=(-1))&(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==(-1))&((xnn>=(-1))&(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>=(-1))&(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>=(-1))&(ynn<=1))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=0;
        end
  end
%=========================boundary conditions mesh-13=================
case 13
```

```
    lambda1=(1/lambda)^2;lambda2=2;p=lambda;
      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
%=========================================================================
 case 14
    lambda1=(lambda);lambda2=2;lambda3=-1;p=lambda;
      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
%=========================================================================
       case 15

    lambda1=lambda;lambda2=0;lambda3=1/10;
      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)=sin(pi*xnn);
        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

 %=========================boundary conditions mesh-16(2-square)========

case 16
     lambda1=lambda;lambda2=0;lambda3=1
     p=lambda;
     nnn=0;
    %boundary conditions 1
    for nn=1:nnode
        xnn=coord(nn,1);ynn=gcoord(nn,2);
```

```
        if (xnn==-1)&((ynn>=(-1))&(ynn<=1))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=(1-exp(-2/p))*(1-exp(-(1-ynn)/p))*cos(pi*(-1+ynn));
        end
    end
    %boundary conditions-2
     for nn=1:nnode
        xnn=gcoord(nn,1);ynn=coord(nn,2);
        if (ynn==(-1))&((xnn>=(-1))&(xnn<=1))
            nnn=nnn+1;
            bcdof(nnn,1)=nn;
            bcval(nnn,1)=(1-exp(-(1-xnn)/p))*(1-exp(-2/p))*cos(pi*(xnn-1))
        end
     end

     %boundary conditions-3
    for nn=1:nnode
        xnn=gcoord(nn,1);ynn=coord(nn,2);
        if (ynn==1)&((xnn>=(-1))&(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>=(-1))&(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
if (mesh == 1)|(mesh == 2)|(mesh==12)
 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
if (mesh == 13)
 xi=(zeros(nnode,1));
for m=1:nnode
    xm=(gcoord(m,1));ym=(gcoord(m,2));
    U1=1-(exp((-xm/p))+exp(-(1-xm)/p))/(1+exp(-1/p));
    U2=1-(exp((-ym/p))+exp(-(1-ym)/p))/(1+exp(-1/p));
    xi(m,1)=(U1*U2)
end
end

if (mesh == 14)
```

```
 xi=(zeros(nnode,1));
for m=1:nnode
    xm=(gcoord(m,1));ym=(gcoord(m,2));
    U1=(1-exp(-xm/p))*(1-xm);U2=(1-exp(-ym/p))*(1-ym);
    xi(m,1)=(U1*U2)
end
end

if (mesh == 15)
 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/2);
end
end

if (mesh == 16)
 xi=(zeros(nnode,1));
for m=1:nnode
    xm=(gcoord(m,1));ym=(gcoord(m,2));
    U1=(1-exp(-(1-xm)/p));U2=(1-exp(-(1-ym)/p));U3=cos(pi*(xm+ym))
     xi(m,1)=(U1*U2*U3)
end
end



 for L=1:nel
   for M=1:3
      LM=nodetel(L,M);
      xx(L,M)=gcoord(LM,1);
      yy(L,M)=gcoord(LM,2);
    end
  end
%_____________________________________________________________
ng=10
 [sp,wt]=glsampleptsweights(ng)
table2=[N xx yy];
%disp([xx yy])
intJdn1dn1uvrs =[vpa(sym(' .595527655000821147485729267330')),
vpa(sym('.468322285820574645491168269290'));vpa(sym('
.468322285820574645491168269290')), vpa(sym('.595527655000821147485729267330'))];
intJdn1dn2uvrs =[vpa(sym('  -.397018436667214098323819511552')),
vpa(sym('.187785142786283569672554487139 5'));vpa(sym(' -
.312214857213716430327445512860 4')),   vpa(sym('.102981563332785901676180488448'))];
intJdn1dn3uvrs =[vpa(sym(' -.301490781666392950838090244223 5')),vpa(sym(' -
.343892571393141784836277243570 0'));vpa(sym(' -.343892571393141784836277243570 0')),
vpa(sym('-.301490781666392950838090244223 5'))];
intJdn1dn4uvrs =[vpa(sym('   .102981563332785901676180488448')), vpa(sym('-
.312214857213716430327445512860 4'));vpa(sym('  .187785142786283569672554487139 5')),
vpa(sym('-.397018436667214098323819511552'))];
%
intJdn2dn1uvrs =[vpa(sym('  -.397018436667214098323819511552')), vpa(sym('-
.312214857213716430327445512860 4'));vpa(sym('  .187785142786283569672554487139 5')),
vpa(sym(' .102981563332785901676180488448'))];
intJdn2dn2uvrs =[vpa(sym('   .264678957778142732215879674369')), vpa(sym('-
.125190095190855713115036324760 0'));vpa(sym(' -.125190095190855713115036324760 0')),
vpa(sym(' .264678957778142732215879674369'))];
intJdn2dn3uvrs =[vpa(sym('  .200993854444261967225393496149 1')),
vpa(sym('.229261714262094523224184829046 6'));vpa(sym(' -
.270738285737905476775815170953 4')), vpa(sym('-.299006145555738032774606503850 9'))];
```

```
intJdn2dn4uvrs =[vpa(sym(' -.68654375555190601117453658965e-1')),
vpa(sym('.20814323814247762021829700857 34')));vpa(sym('
.20814323814247762021829700857 34')), vpa(sym('-.68654375555190601117453658965e-1'))];
%
intJdn3dn1uvrs =[vpa(sym(' -.30149078166639295083809024422 35')), vpa(sym('-
.34389257139314178483627724357 00'));vpa(sym(' -
.34389257139314178483627724357 00')),vpa(sym(' -.30149078166639295083809024422 35'))];
intJdn3dn2uvrs =[vpa(sym('  .20099385444426196722539349614 91')), vpa(sym('-
.27073828573790547677581517095 34'));vpa(sym('  .22926171426209452322418482904 66')),
vpa(sym('-.29900614555573803277460650385 09'))];
intJdn3dn3uvrs =[vpa(sym(' .39950307277786901638730325192 54')),
vpa(sym('.38536914286895273838790758547 68'));vpa(sym('
.38536914286895273838790758547 68')), vpa(sym('.39950307277786901638730325192 54'))];
intJdn3dn4uvrs =[vpa(sym(' -.29900614555573803277460650385 09')), vpa(sym('
.22926171426209452322418482904 66'));vpa(sym(' -.27073828573790547677581517095 34')),
vpa(sym('.20099385444426196722539349614 91'))];
%
intJdn4dn1uvrs =[vpa(sym('   .10298156333278590167618048844 8')),
vpa(sym('.18778514278628356967255448713 95'));vpa(sym(' -
.31221485721371643032744551286 04')), vpa(sym(' -.39701843666721409832381951155 2'))];
intJdn4dn2uvrs =[vpa(sym(' -.68654375555190601117453658965e-1')),  vpa(sym('
.20814323814247762021829700857 34'));vpa(sym('   .20814323814247762021829700857 34')),
vpa(sym('-.68654375555190601117453658965e-1'))];
intJdn4dn3uvrs =[vpa(sym(' -.29900614555573803277460650385 09')), vpa(sym('-
.27073828573790547677581517095 34'));vpa(sym('  .22926171426209452322418482904 66')),
vpa(sym('.20099385444426196722539349614 91'))];
intJdn4dn4uvrs =[vpa(sym('   .26467895777814273221587967436 9')), vpa(sym('-
.12519009519085571311503632476 00'));vpa(sym(' -.12519009519085571311503632476 00')),
vpa(sym(' .26467895777814273221587967436 9'))];
%
%
intJdndn=double([intJdn1dn1uvrs intJdn1dn2uvrs intJdn1dn3uvrs intJdn1dn4uvrs;...
          intJdn2dn1uvrs intJdn2dn2uvrs intJdn2dn3uvrs intJdn2dn4uvrs;...
          intJdn3dn1uvrs intJdn3dn2uvrs intJdn3dn3uvrs intJdn3dn4uvrs;...
          intJdn4dn1uvrs intJdn4dn2uvrs intJdn4dn3uvrs intJdn4dn4uvrs]);

 IntJNiNj =[  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];


IntNiDNj=[ 1/12 -1/18 -1/24  1/72;...
           1/12  1/72 -1/24 -1/18;...
           1/12 -1/18 -1/18  1/36;...
           1/24  1/18 -5/72 -1/36;...
           1/24 -1/36 -1/12  5/72;...
           1/24  5/72 -1/12 -1/36;...
           1/24 -1/36 -5/72  1/18;...
           1/12  1/36 -1/18 -1/18];




%
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(1:4,1:4)=(zeros(4,4));
for i=1:4
 for j=1:4
 if (mesh == 1)|(mesh == 2)|(mesh==12)|(mesh==13)
 sk(i,j)=(delabc*sum(sum(Q.*(intJdndn(2*i-1:2*i,2*j-
1:2*j)))))/lambda1+lambda2*delabc*IntJNiNj(i,j);
 end
 if mesh==14
     sk(i,j)=(delabc*sum(sum(Q.*(intJdndn(2*i-1:2*i,2*j-
1:2*j)))))*lambda1+lambda3*([bta+gma btb+gmb ]*IntNiDNj(2*i-
1:2*i,j))+lambda2*delabc*IntJNiNj(i,j);
 end

 if mesh==15
     sk(i,j)=(delabc*sum(sum(Q.*(intJdndn(2*i-1:2*i,2*j-
1:2*j)))))*lambda1+lambda3*([gma gmb ]*IntNiDNj(2*i-1:2*i,j));
 end

 if mesh==16
     sk(i,j)=(delabc*sum(sum(Q.*(intJdndn(2*i-1:2*i,2*j-
1:2*j)))))*lambda1+lambda3*([2*bta+gma 2*btb+gmb ]*IntNiDNj(2*i-
1:2*i,j))+lambda2*delabc*IntJNiNj(i,j);
 end
 %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(1:4,1)=zeros(4,1)
 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=(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;
        if (mesh == 1)|(mesh == 2)|(mesh==12)

f1i=n1ij*(((lambda2^2)+(2*pi^2))/lambda1)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;

f2i=n2ij*(((lambda2^2)+(2*pi^2))/lambda1)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;

f3i=n3ij*(((lambda2^2)+(2*pi^2))/lambda1)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;

f4i=n4ij*(((lambda2^2)+(2*pi^2))/lambda1)*sin(pi*xeij)*sin(pi*yeij)*(4+si+sj)/96;
        end
        if (mesh==13)
        fxy= 2 - (exp(xeij/p) + exp(yeij/p) + exp(1/p - xeij/p) + exp(1/p -
yeij/p))/(exp(1/p) + 1);
        f1i=n1ij*fxy*(4+si+sj)/96;
        f2i=n2ij*fxy*(4+si+sj)/96;
        f3i=n3ij*fxy*(4+si+sj)/96;
        f4i=n4ij*fxy*(4+si+sj)/96;
        end
        if (mesh==14)
         x=xeij;y=yeij;
         fxy=(x + 3*y - (x + y - 2*x*y)/exp(x/p) - 2*x*y - 2)/exp(y/p) + 2*x*y - 3*y -
3*x + (3*x + y - 2*x*y - 2)/exp(x/p) + 4;
        f1i=n1ij*fxy*(4+si+sj)/96;
        f2i=n2ij*fxy*(4+si+sj)/96;
        f3i=n3ij*fxy*(4+si+sj)/96;
        f4i=n4ij*fxy*(4+si+sj)/96;
        end
        if (mesh==15)
        fxy=(pi*sin(pi*xeij)*(cos((pi*yeij)/2) + 25*pi*sin((pi*yeij)/2)))/20;
        f1i=n1ij*fxy*(4+si+sj)/96;
        f2i=n2ij*fxy*(4+si+sj)/96;
        f3i=n3ij*fxy*(4+si+sj)/96;
        f4i=n4ij*fxy*(4+si+sj)/96;
        end
        if (mesh==16)
            x=xeij;y=yeij;
      % fxy=(2*pi^2*p - exp((x - 1)/p)/p + (exp((x - 1)/p)*exp((y - 1)/p))/p -
2*pi^2*p*exp((x - 1)/p) - 2*pi^2*p*exp((y - 1)/p) + 2*pi^2*p*exp((x - 1)/p)*exp((y -
1)/p))*cos(pi*(x + y)) + (pi*exp((x - 1)/p) - 3*pi + pi*exp((y - 1)/p) + pi*exp((x -
1)/p)*exp((y - 1)/p))*sin(pi*(x + y));
      % fxy=(pi*exp((x - 1)/p) - 3*pi + exp((y - 1)/p)*(pi + pi*exp((x -
1)/p)))*sin(pi*(x + y)) + (exp((y - 1)/p)*(exp((x - 1)/p)*(2*pi^2*p + 1/p) - 2*pi^2*p)
- exp((x - 1)/p)*(2*pi^2*p + 1/p) + 2*pi^2*p)*cos(pi*(x + y));
      %fxy =pi*exp((x - 1)/p)*sin(pi*(x + y)) - 3*pi*sin(pi*(x + y)) + pi*exp((y -
1)/p)*sin(pi*(x + y)) + 2*pi^2*p*cos(pi*(x + y)) - (cos(pi*(x + y))*exp((x - 1)/p))/p -
2*pi^2*p*cos(pi*(x + y))*exp((x - 1)/p) - 2*pi^2*p*cos(pi*(x + y))*exp((y - 1)/p) +
(cos(pi*(x + y))*exp((x - 1)/p)*exp((y - 1)/p))/p + pi*exp((x - 1)/p)*exp((y -
1)/p)*sin(pi*(x + y)) + 2*pi^2*p*cos(pi*(x + y))*exp((x - 1)/p)*exp((y - 1)/p);
      fxy=(2*pi^2*p - exp((x - 1)/p)/p + (exp((x - 1)/p)*exp((y - 1)/p))/p -
2*pi^2*p*exp((x - 1)/p) - 2*pi^2*p*exp((y - 1)/p) + 2*pi^2*p*exp((x - 1)/p)*exp((y -
1)/p))*cos(pi*(x + y)) + (pi*exp((x - 1)/p) - 3*pi + pi*exp((y - 1)/p) + pi*exp((x -
1)/p)*exp((y - 1)/p))*sin(pi*(x + y));
        f1i=n1ij*fxy*(4+si+sj)/96;
        f2i=n2ij*fxy*(4+si+sj)/96;
```

```
        f3i=n3ij*fxy*(4+si+sj)/96;
        f4i=n4ij*fxy*(4+si+sj)/96;
        end



          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;
      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]
[centnode]


%for I=1:nel/3
%II=centnode(I,1);
%NNC(I,1)=II;
%phic(I,1)=phi(II,1);
%xic(I,1)=xi(II,1);
%end

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

%disp([NNC phic xic])
%disp('_____
_____')
ng


for I=1:nel/3
II=centnode(I,1);
NNC(I,1)=II;
xnnc(I,1)=gcoord(II,1);
ynnc(I,1)=gcoord(II,2);
phic(I,1)=phi(II,1);
xic(I,1)=xi(II,1);
end

disp('_____
_____')
disp('number of nodes,elements & nodes per element')
[nnode nel nnel ndof]
disp('_____
_____')
disp('                      node             x-coordinate           y-coordinate
fem-computed values      anlytical(theoretical)-values           ')
if (nel/3)<11
disp([NNC  xnnc ynnc  phic xic])
end
if (nel/3)>10
disp([NNC(1:10:nel/3,1)  xnnc(1:10:nel/3,1) ynnc(1:10:nel/3,1)  phic(1:10:nel/3,1)
xic(1:10:nel/3,1)])
end
disp('_____
_____')
```

(3) polygonal_domain_coordinates.m

```
function[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates(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)
%PARVIZ MOIN EXAMPLE
syms U V W xi yi

switch mesh
case 1%for MOIN POLYGON
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%for a unit square: 0<=x,y<=1
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%for A POLYGON like MOIN OVER(-1/2)<=x,y<=(1/2)
    %  1    2    3   4     5   6    7   8
  x=sym([0;   0; 1/2;1/2;  0;-1/2;-1/2;-1/2])
  y=sym([0;-1/2;-1/2;  0;1/2; 1/2;   0;-1/2])

case 4%for a unit square: -0.5<=x,y<=0.5
  %       1    2   3   4   5   6    7   8
  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])
case 5%for a convexpolygonsixside
  %    1    2   3  4   5  6 7
x=sym([0.5;0.1;0.7;1;.75;.5;0])
y=sym([0.5;0;0.2;.5;.85;1;.25])
case 6%standard triangle
x=sym([0;1;0])
y=sym([0;0;1])
case 7%equilateral triangle

x=sym([0;1;1/2])
y=sym([0;0;sqrt(3)/2])
case 8%for a convexpolygonsixside
  %    1    2   3  4   5  6  7  8
x=sym([0.5;0.1;0.7;1;.75;.5;.25;0])
y=sym([0.5;0;0.2;.5;.85;1;.625;.25])
```

```
case 9%for a convexpolygeightside
%    1  2   3  4   5    6  7     8   9
x=([.2;0.5;.8;1.0;.75;0.5;0.25;0.0;0.5])
y=([.2;.05;.2;0.5;.85;1.0;0.90;0.6;0.5])
case 10
%    1  2   3  4   5    6  7     8   9
x=([0.5;.2;0.5;.8;1.0;.75;0.5;0.25;0.0])
y=([0.5;.2;.05;.2;0.5;.85;1.0;0.90;0.6])
 case 11 %a square
 %  1 2 3 4  5
x=([0;1;1;0;0.5])
y=([0;0;1;1;0.5])
case 12 %a 2-square [-1,1]x[-1,1]
 %       1    2   3   4   5   6    7   8    9
  x=sym([0;   0;  1;  1;  1;  0;  -1;  -1; -1])
  y=sym([0;  -1; -1;  0;  1;  1;   1;   0; -1])
case 13%for a unit square: 0<=x,y<=1
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 14%for a unit square: 0<=x,y<=1
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 15%for a unit square: 0<=x,y<=1
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 SQUAREpi

case 16 %a 2-square [-1,1]x[-1,1]
 %       1    2   3   4   5   6    7   8    9
  x=sym([0;   0;  1;  1;  1;  0;  -1;  -1; -1])
  y=sym([0;  -1; -1;  0;  1;  1;   1;   0; -1])

case 17%isosceles triangle(one triangle in a square-required for torsion analysis)
x=sym([0;sqrt(2)/2; -sqrt(2)/2])
y=sym([0;sqrt(2)/2;sqrt(2)/2])
case 18%isoscles triangle(torsion of an equilateral triangle,each side=2*sqrt(3))
 x=sym([-sqrt(3);sqrt(3); 0])
 y=sym([     -1;      -1; 2])
 case 19%triangle inscribed in a circle of radius=1(torsion of an equilateral triangle
each side=sqrt(3))
 x=sym([-sqrt(3)/2;sqrt(3)/2; 0])
 y=sym([     -1/2;      -1/2; 1])
 case 20%square inscribed in a circle of radius=1,required for torsion analysis
%      1      2          3       4          5
x=sym([0;-sqrt(2)/2; sqrt(2)/2;sqrt(2)/2;-sqrt(2)/2])
y=sym([0;-sqrt(2)/2;-sqrt(2)/2;sqrt(2)/2; sqrt(2)/2])
case 21%regular pentagon inscribed in a circle of radius=1,required for torsion
analysis
 %      1      2    3       4           5            6
x=sym([0;cos(pi/10);0;-cos(pi/10);-cos(3*pi/10); cos(3*pi/10)])
y=sym([0;sin(pi/10);1; sin(pi/10);-sin(3*pi/10);-sin(3*pi/10)])
case 22%regular hexagon inscribed in a circle of radius=1,required for torsion analysis
    %      1 2   3          4        5     6           7
    x=sym([0;1;cos(pi/3);-cos(pi/3);-1; -cos(pi/3); cos(pi/3)])
    y=sym([0;0;sin(pi/3); sin(pi/3); 0; -sin(pi/3);-sin(pi/3)])
    case 23%regular heptagon inscribed in a circle of radius=1,required for torsion
analysis
 %      1     2            3              4            5       6       7         8
x=sym([0;-cos(5*pi/14); cos(5*pi/14); cos(pi/14);cos(3*pi/14);0;-cos(3*pi/14);-
cos(pi/14)])
y=sym([0;-sin(5*pi/14);-sin(5*pi/14);-sin(pi/14);sin(3*pi/14);1; sin(3*pi/14);-
sin(pi/14)])
    case 24 %regular octagon inscribed in a circle of radius=1,required for torsion
analysis
```

```
 %     1 2     3     4     5       6       7     8     9
x=sym([0;1;cos(pi/4);0;-cos(pi/4);-1;-cos(pi/4); 0; cos(pi/4)])
y=sym([0;0;sin(pi/4);1; sin(pi/4); 0;-sin(pi/4);-1;-sin(pi/4)])
    case 25%(9-gon)nonagon
 %       1     2             3      4          5         6            7             8
9              10
  x=sym([0;cos(pi/18); cos(5*pi/18);0;-cos(5*pi/18);-cos(pi/18);-cos(3*pi/18);-
cos(7*pi/18); cos(7*pi/18); cos(3*pi/18)])
  y=sym([0;sin(pi/18); sin(5*pi/18);1; sin(5*pi/18); sin(pi/18);-sin(3*pi/18);-
sin(7*pi/18);-sin(7*pi/18);-sin(3*pi/18)])
    case 26%(10-gon)decagon
 %       1 2     3          4          5              6        7     8           9
10       11
  x=sym([0;1;cos(pi/5); cos(2*pi/5);-cos(2*pi/5);-cos(pi/5);-1;-cos(pi/5);-cos(2*pi/5);
cos(2*pi/5); cos(pi/5)])
  y=sym([0;0;sin(pi/5); sin(2*pi/5); sin(2*pi/5); sin(pi/5); 0;-sin(pi/5);-
sin(2*pi/5);-sin(2*pi/5);-sin(pi/5)])
    case 27%(11-gon)hendecagon
  %      1        2          3       4          5          6              7        8
9              10                11       12
  x=sym([0;cos(3*pi/22);cos(7*pi/22);0;-cos(7*pi/22);-cos(3*pi/22);-cos(pi/22);-
cos(5*pi/22);-cos(9*pi/22); cos(9*pi/22); cos(5*pi/22); cos(pi/22)])
  y=sym([0;sin(3*pi/22);sin(7*pi/22);1; sin(7*pi/22); sin(3*pi/22);-sin(pi/22);-
sin(5*pi/22);-sin(9*pi/22);-sin(9*pi/22);-sin(5*pi/22);-sin(pi/22)])
  case 28%(12-gon)dodecagon
  %      1 2   3           4     5      6          7      8       9         10    11
12     13
  x=sym([0;1;cos(pi/6);cos(pi/3);0;-cos(pi/3);-cos(pi/6);-1;-cos(pi/6);-cos(pi/3); 0;
cos(pi/3); cos(pi/6)])
  y=sym([0;0;sin(pi/6);sin(pi/3);1; sin(pi/3); sin(pi/6); 0;-sin(pi/6);-sin(pi/3);-1;-
sin(pi/3);-sin(pi/6)])
  case 29%(13-gon)tridecagon
  %      1       2         3           4        5      6                 7            8
9                10           11              12          13            14
  x=sym([0;cos(pi/26);cos(5*pi/26);cos(9*pi/26);0;-cos(9*pi/26);-cos(5*pi/26);-
cos(pi/26);-cos(3*pi/26);-cos(7*pi/26);-cos(11*pi/26); cos(11*pi/26); cos(7*pi/26);
cos(3*pi/26)])
  y=sym([0;sin(pi/26);sin(5*pi/26);sin(9*pi/26);1; sin(9*pi/26); sin(5*pi/26);
sin(pi/26);-sin(3*pi/26);-sin(7*pi/26);-sin(11*pi/26);-sin(11*pi/26);-sin(7*pi/26);-
sin(3*pi/26)])
    case 30%(14-gon)tetradecagon
  %      1 2         3          4             5         6                7
8       9       10            11              12          13           14            15
  x=sym([0;1;cos(2*pi/14);cos(4*pi/14);cos(6*pi/14);-cos(6*pi/14);-cos(4*pi/14);-
cos(2*pi/14);-1;-cos(2*pi/14);-cos(4*pi/14);-cos(6*pi/14); cos(6*pi/14); cos(4*pi/14);
cos(2*pi/14)])
  y=sym([0;0;sin(2*pi/14);sin(4*pi/14);sin(6*pi/14); sin(6*pi/14); sin(4*pi/14);
sin(2*pi/14); 0;-sin(2*pi/14);-sin(4*pi/14);-sin(6*pi/14);-sin(6*pi/14);-sin(4*pi/14);-
sin(2*pi/14)])
    case 31%(15-gon)pentadecagon
  %      1       2         3              4             5       6    7
8              9       10            11              12          13           14
15             16
   x=sym([0; cos(pi/30);cos(3*pi/30);cos(7*pi/30);cos(11*pi/30);0;-cos(11*pi/30);-
cos(7*pi/30);-cos(3*pi/30);-cos(pi/30);-cos(5*pi/30);-cos(9*pi/30);-cos(13*pi/30);
cos(13*pi/30); cos(9*pi/30); cos(5*pi/30)])
   y=sym([0;-sin(pi/30);sin(3*pi/30);sin(7*pi/30);sin(11*pi/30);1; sin(11*pi/30);
sin(7*pi/30); sin(3*pi/30);-sin(pi/30);-sin(5*pi/30);-sin(9*pi/30);-sin(13*pi/30);-
sin(13*pi/30);-sin(9*pi/30);-sin(5*pi/30)])
    case 32%(16-gon)hexadecagon
    %  1 2     3       4          5       6      7             8        9     10
11          12          13    14       15           16        17
```

```
   x=sym([0;1;cos(pi/8);cos(pi/4);cos(3*pi/8);0;-cos(3*pi/8);-cos(pi/4);-cos(pi/8);-1;-
cos(pi/8);-cos(pi/4);-cos(3*pi/8); 0; cos(3*pi/8); cos(pi/4); cos(pi/8)]);
   y=sym([0;0;sin(pi/8);sin(pi/4);sin(3*pi/8);1; sin(3*pi/8); sin(pi/4); sin(pi/8); 0;-
sin(pi/8);-sin(pi/4);-sin(3*pi/8);-1;-sin(3*pi/8);-sin(pi/4);-sin(pi/8)]);
    case 33%(17-gon)heptadecogon
   %   1          2          3       4              5       6          7
8          9            10           11          12              13              14
15           16             17       18
   x=sym([0;cos(pi/34);cos(5*pi/34);cos(9*pi/34);cos(13*pi/34);0;-cos(13*pi/34);-
cos(9*pi/34);-cos(5*pi/34);-cos(pi/34);-cos(3*pi/34);-cos(7*pi/34);-cos(11*pi/34);-
cos(15*pi/34); cos(15*pi/34); cos(11*pi/34); cos(7*pi/34); cos(3*pi/34)]);
   y=sym([0;sin(pi/34);sin(5*pi/34);sin(9*pi/34);sin(13*pi/34);1; sin(13*pi/34);
sin(9*pi/34); sin(5*pi/34); sin(pi/34);-sin(3*pi/34);-sin(7*pi/34);-sin(11*pi/34);-
sin(15*pi/34);-sin(15*pi/34);-sin(11*pi/34);-sin(7*pi/34);-sin(3*pi/34)]);
 case 34%(18-gon)octadecogon
  %  1 2           3       4              5              6          7              8
9             10        11         12           13              14            15
16              17       18              19
 x=sym([0;1;cos(4*pi/36);cos(8*pi/36);cos(12*pi/36);cos(16*pi/36);-cos(16*pi/36);-
cos(12*pi/36);-cos(8*pi/36);-cos(4*pi/36);-1;-cos(4*pi/36);-cos(8*pi/36);-
cos(12*pi/36);-cos(16*pi/36); cos(16*pi/36); cos(12*pi/36); cos(8*pi/36);
cos(4*pi/36)]);
 y=sym([0;0;sin(4*pi/36);sin(8*pi/36);sin(12*pi/36);sin(16*pi/36); sin(16*pi/36);
sin(12*pi/36); sin(8*pi/36); sin(4*pi/36); 0;-sin(4*pi/36);-sin(8*pi/36);-
sin(12*pi/36);-sin(16*pi/36);-sin(16*pi/36);-sin(12*pi/36);-sin(8*pi/36);-
sin(4*pi/36)]);
case 35%(19-gon)enneadecogon
  %  1        2           3           4             5        6      7              8
9             10             11          12           13              14            15
16                  17       18              19             20
 x=sym([0;cos(3*pi/38);cos(7*pi/38);cos(11*pi/38);cos(15*pi/38);0;-cos(15*pi/38);-
cos(11*pi/38);-cos(7*pi/38);-cos(3*pi/38);-cos(pi/38);-cos(5*pi/38);-cos(9*pi/38);-
cos(13*pi/38);-cos(17*pi/38); cos(17*pi/38); cos(13*pi/38); cos(9*pi/38); cos(5*pi/38);
cos(pi/38)]);
 y=sym([0;sin(3*pi/38);sin(7*pi/38);sin(11*pi/38);sin(15*pi/38);1; sin(15*pi/38);
sin(11*pi/38); sin(7*pi/38); sin(3*pi/38);-sin(pi/38);-sin(5*pi/38);-sin(9*pi/38);-
sin(13*pi/38);-sin(17*pi/38);-sin(17*pi/38);-sin(13*pi/38);-sin(9*pi/38);-
sin(5*pi/38);-sin(pi/38)]);
  case 36%(20-gon)icosagon
  % 1 2     3           4             5           6      7           8             9
10          11   12      13           14            15            16      17       18
19             20      21
x=sym([0;1;cos(pi/10);cos(2*pi/10);cos(3*pi/10);cos(4*pi/10);0;-cos(4*pi/10);-
cos(3*pi/10);-cos(2*pi/10);-cos(pi/10);-1;-cos(pi/10);-cos(2*pi/10);-cos(3*pi/10);-
cos(4*pi/10); 0; cos(4*pi/10); cos(3*pi/10); cos(2*pi/10); cos(pi/10)]);
y=sym([0;0;sin(pi/10);sin(2*pi/10);sin(3*pi/10);sin(4*pi/10);1; sin(4*pi/10);
sin(3*pi/10); sin(2*pi/10); sin(pi/10); 0;-sin(pi/10);-sin(2*pi/10);-sin(3*pi/10);-
sin(4*pi/10);-1;-sin(4*pi/10);-sin(3*pi/10);-sin(2*pi/10);-sin(pi/10)]);



end
if (nmax>3)
[eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals_trial(n1,n2,n
3,nmax,numtri,n);
end
if (nmax==3)
    [eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals(n);
end
[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)])



nitri=nmax-1;
if (nmax==3)nitri=1
end
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
end
[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

end
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)
```

(4)nodaladdresses_special_convex_quadrilaterals_trial.m

```
function[eln,spqd,rrr,nodes,nodetel]=nodaladdresses_special_convex_quadrilaterals_trial
(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)
%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
```

```
   end
```

```
%for mmm=1:mm
    %spqd(:,1:4)
 %end
%
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)])
```

(5) quadrilateral_mesh4MOINEX_q4.m

```
function[]=quadrilateral_mesh4MOINEX_q4(n1,n2,n3,nmax,numtri,ndiv,mesh,xlength,ylength)
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;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;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],[3;4;5;6;7;8;2],8,9,6,1,1
,1)
%quadrilateral_mesh4MOINEX_q4([1;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;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9,4
,4,2,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9,9
,6,2,1,1)
%quadrilateral_mesh4MOINEX_q4([1;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)
%quadrilateral_mesh4MOINEX_q4([9;9;9;9;9;9;9;9],[1;2;3;4;5;6;7;8],[2;3;4;5;6;7;8;1],9,1
,2,9,1,1)
%quadrilateral_mesh4MOINEX_q4([9;9;9;9;9;9;9;9],[1;2;3;4;5;6;7;8],[2;3;4;5;6;7;8;1],9,4
,4,9,1,1)
%quadrilateral_mesh4MOINEX_q4([9;9;9;9;9;9;9;9],[1;2;3;4;5;6;7;8],[2;3;4;5;6;7;8;1],9,9
,6,9,1,1)
%quadrilateral_mesh4MOINEX_q4([1;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
,2,10,1,1)
```

```
%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
,4,10,1,1)
%quadrilateral_mesh4MOINEX_q4([1],[2],[3],3,1,2,6,1,1)
%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],9,2
5,10,2,1,1)
%%quadrilateral_mesh4MOINEX_q4([1],[2],[3],3,1,2,17,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1],[2;3;4;5;6],[3;4;5;6;2],6,9,6,21,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1],[2;3;4;5;6;7],[3;4;5;6;7;2],7,9,6,22,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8],[3;4;5;6;7;8;2],8,1,2,23,
1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9],[3;4;5;6;7;8;9;2],9,2
5,10,24,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9;10],[3;4;5;6;7;8;9;1
0;2],10,1,2,25,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9;10;11],[3;4;5;6;7;
8;9;10;11;2],11,1,2,26,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9;10;11;12],[3;4;5
;6;7;8;9;10;11;12;2],12,9,6,27,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9;10;11;12;13],[
3;4;5;6;7;8;9;10;11;12;13;2],13,9,6,28,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9;10;11;12;13;
14],[3;4;5;6;7;8;9;10;11;12;13;14;2],14,1,2,29,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9;10;11;12;1
3;14;15],[3;4;5;6;7;8;9;10;11;12;13;14;15;2],15,1,2,30,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9;10;11;12
;13;14;15;16],[3;4;5;6;7;8;9;10;11;12;13;14;15;16;2],16,1,2,31,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9;10;11;
12;13;14;15;16;17],[3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;2],17,1,2,32,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9;10;1
1;12;13;14;15;16;17;18],[3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;2],18,1,2,33,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9;10;1
1;12;13;14;15;16;17;18],[3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;2],18,1,2,33,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9;10
;11;12;13;14;15;16;17;18;19],[3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;2],19,1,2,34,
1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;9;
10;11;12;13;14;15;16;17;18;19;20],[3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;20;2],20
,1,2,35,1,1)
%quadrilateral_mesh4MOINEX_q4([1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1],[2;3;4;5;6;7;8;
9;10;11;12;13;14;15;16;17;18;19;20;21],[3;4;5;6;7;8;9;10;11;12;13;14;15;16;17;18;19;20;
21;2],21,1,2,36,1,1)
[coord,gcoord,nodes,nodetel,nnode,nel]=polygonal_domain_coordinates(n1,n2,n3,nmax,numtr
i,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
figure(ndiv/2)
for i=1:nel
for j=1:nnel
x(1,j)=xcoord(nodes(i,j),1);
y(1,j)=ycoord(nodes(i,j),1);
end;%j loop
xvec(1,1:5)=[x(1,1),x(1,2),x(1,3),x(1,4),x(1,1)];
yvec(1,1:5)=[y(1,1),y(1,2),y(1,3),y(1,4),y(1,1)];
```

```
axis equal
%axis tight
rtext=0;
switch mesh
case 1
axis([0 xlength 0 ylength])
case 2
axis([0 xlength 0 ylength])
case 3
 xl=xlength/2;yl=ylength/2;
axis([-xl xl -yl yl])
case 4
xl=xlength/2;yl=ylength/2;
axis([-xl xl -yl yl])
case 12
    axis([-xlength xlength -ylength ylength])

case 17
    axis([-xlength xlength 0 ylength])
     rpoly=' one isosceles triangle in a  square';
    rtext=1;
case 18
    axis([-2*xlength 2*xlength -ylength 2*ylength])
    rpoly='   equilateral triangle';
    rtext=1;
case 19
    axis([-xlength xlength -ylength/2 ylength])
     rpoly='   equilateral triangle';
    rtext=1;

 case 20
    axis([-xlength xlength -ylength ylength])
     rpoly='    square';
    rtext=1;
 case 21
    axis([-xlength xlength -ylength ylength])
     rpoly='     pentagon';
    rtext=1;
 case 22
    axis([-xlength xlength -ylength ylength])
     rpoly='     hexagon';
    rtext=1;
  case 23
    axis([-xlength xlength -ylength ylength])
     rpoly='     heptagon';
    rtext=1;
    case 24
    axis([-xlength xlength -ylength ylength])
     rpoly='     octagon';
    rtext=1;
     case 25
    axis([-xlength xlength -ylength ylength])
     rpoly='     nonadecagon';
    rtext=1;
     case 26
    axis([-xlength xlength -ylength ylength])
     rpoly='     decagon';
     rtext=1;
     case 27
    axis([-xlength xlength -ylength ylength])
     rpoly='     hendecagon';
    rtext=1;
     case 28
```

```
    axis([-xlength xlength -ylength ylength])
     rpoly='     dodecagon';
    rtext=1;
     case 29
    axis([-xlength xlength -ylength ylength])
     rpoly='     tridecagon';
    rtext=1;
     case 30
    axis([-xlength xlength -ylength ylength])
     rpoly='     tetradecagon';
    rtext=1;
     case 31
    axis([-xlength xlength -ylength ylength])
     rpoly='     pentadecagon';
    rtext=1;
      case 32
    axis([-xlength xlength -ylength ylength])
     rpoly='     hexadecagon';
    rtext=1;
      case 33
    axis([-xlength xlength -ylength ylength])
     rpoly='     heptadecagon';
    rtext=1;
     case 34
    axis([-xlength xlength -ylength ylength])
    rpoly='     octadecagon';
    rtext=1;
     case 35
    axis([-xlength xlength -ylength ylength])
    rpoly='     enneadecagon';
    rtext=1;
  case 36
    axis([-xlength xlength -ylength ylength])
    rpoly='     icosagon';
    rtext=1;

end
plot(xvec,yvec);%plot element
hold on;
%place element number
if ndiv<=4
midx=mean(xvec(1,1:4))
midy=mean(yvec(1,1:4))
text(midx,midy,['[',num2str(i),']']);
end
end;%i loop

xlabel('x axis')
ylabel('y axis')
st1='FEM MESH ';
st2=num2str(nel);
st3=' four noded  ';
st4='quadrilateral';
st5=' elements'
st6='& nodes='
st7=num2str(nnode);
title([st1,st2,st3,st4,st5,st6,st7])
%put node numbers
if ndiv<=4
for jj=1:nnode
text(gcoord(jj,1),gcoord(jj,2),['o',num2str(jj)]);
end
end
```

```
if rtext==1
  text(0.1,-1.2,rpoly)
end
%axis off
```