www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 4 Issue 10 Oct 2015, Page No. 14726-14798

A New Approach to Automatic Generation of An All Pentagonal Finite Element Mesh For Numerical Computations Over Convex Polygonal Domains

H.T. Rathod [a*] , K. Sugantha Devi [b]

[a] Department of Mathematics, Central College Campus, Bangalore University,
Bangalore -560001, Karnataka state, India.
Email: 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

A new method is presented for subdividing a large class of solid objects into topologically simple subregions suitable for automatic finite element meshing with pentagonal elements.
It is known that one can improve the accuracy of the finite element solution by uniformly refining a triangulation or uniformly refining a quadrangulation. Recently a refinement scheme of pentagonal partition was introduced in [31,32,33]. It is demonstrated that the numerical solution based on the pentagonal refinement scheme outperforms the solutions based on the traditional triangulation refinement scheme as well as quadrangulation refinement scheme. It is natural to ask if one can create a hexagonal refinement or general polygonal refinement schemes with a hope to offer even further improvement. It is shown in literature that one cannot refine a hexagon using hexagons of smaller size. In general, one can only refine an n-gon by n-gons of smaller size if $n \leq 5$. Furthermore, we introduce a refinement scheme of a general polygon based on the pentagon scheme. This paper first presents a pentagonalization (or pentagonal conversion) scheme that can create a pentagonal mesh from any arbitrary mesh structure. We also introduce a pentagonal preservation scheme that can create a pentagonal mesh from any pentagonal mesh. This paper then presents a new numerical integration technique proposed earlier by the first author and co-workers, known as boundary integration method [34-40] is now applied to arbitrary polygonal domains using pentagonal finite element mesh. Numerical results presented for a few benchmark problems in the context of pentagonal domains with composite numerical integration scheme over triangular finite elements show that the proposed method yields accurate results even for low order Gauss Legendre Quadrature rules. Our numerical results suggest that the **refinement scheme for pentagons and polygons** may lead to higher accuracy than the **uniform refinement of triangulations and quadrangulations.**

Keywords: Triangular,Quadrangular and Convex Pentagonal regions, Boundary integration, Green'stheorem, Gauss Legendre Quadrature Rules, Composite Integration.Recursive Refinement of Pentagonal and Polygonal domains

1.0 INTRODUCTION

The finite element method (FEM) is a numerical procedure that can be used to obtain solutions to a large class of engineering problems in stress analysis, heat transfer, electromagnetism and fluid flow etc.

FEM has now become a staple for predicting and simulating the physical behaviour of scientific, engineering, medical and business applications.

The use of symbolic computation in support of computational methods in engineering and sciences is steadily growing. This is due to technical improvements in general purpose computer algebra systems (CAS) such as Mathematica and Maple, as well as the availability of inexpensive personal computers and laptops. Furthermore, Maple's symbolic maths tools box is available in widely used Matlab system[17-21]. In finite element work, CAS tools can be used for a spectrum of tasks, formulation, prototyping, implementation, performance evaluation and automatic code generation. FEM is now used as a general purpose method applicable to all kinds of partial differential equations. The advent of modern computer technologies provided a powerful tool in numerical simulations for a range of problems in partial differential equations over arbitrary complex domains. A mesh is required for finite element method as it uses finite elements of a domain for analysis. Finite Element Analysis (FEA) is widely used in many fields including structures and optimization. The FEA in engineering applications comprises three phases: domain discretization, equation solving and error analysis. The domain discretization or mesh generation is the preprocessing phase which plays an important role in the achievement of accurate solutions.

FEM requires dividing the analysis region into many sub regions. These small regions are the elements which are connected with adjacent elements at their nodes. Mesh generation is a procedure of generating the geometric data of the elements and their nodes, and involves computing the coordinates of nodes, defining their connectivity and thus constructing the elements. Hence mesh designates aggregates of elements, nodes and lines representing their connectivity. Though the FEM is a powerful and versatile tool, its usefulness is often hampered by the need to generate a mesh. Creating a mesh is the first step in a wide range of applications, including scientific and engineering computing and computer graphics. But generating a mesh can be very time consuming and prone to error if done manually. In recognition of this problem a large number of methods have been devised to automate the mesh generation task. An attempt to create a fully automatic mesh generator that is capable of generating valid finite element meshes over arbitrary complex domains, needs only the information of the specified geometric boundary of the domain and the element size, started from the pioneering work [1] in the early 1970's. Since then many methodologies have been proposed and different algorithms have been devised in the development of automatic mesh generators [2-4]. In order to perform a reliable finite element simulation a number of researchers [5-7] have made efforts to develop adaptive FEA method which integrates with error estimation and automatic mesh modification. Traditionally adaptive mesh generation process is started from coarse mesh which gives large discretization error levels and takes a lot of iterations to get a desired final mesh. The research literature on the subject is vast and different techniques have been proposed [8], as several engineering applications to real world problems cannot be defined on a rectangular domain or solved on a structured square mesh. The description and discretization of the design domain geometry, specification of the boundary conditions for the governing state equation, and accurate computation of the design response may require the use of unstructured meshes.

An unstructured simplex mesh requires a choice of mesh points (vertex nodes) and triangulation. Many mesh generators produce a mesh of triangles by first creating all the nodes and then connecting nodes to form triangles. The question arises as to what is the 'best' triangulation on a given set of points. One

particular scheme, namely Delaunay triangulation [8], is considered by many researchers to be most suitable for finite element analysis. If the problem domain is a subset of the Cartesian plane, triangular or quadrilateral meshes are typically employed.

The method used for mesh generation can greatly affect the quality of the resulting mesh. Usually the geometry and physical problem of the domain direct the user which method to apply. The real problems in 2D and 3D involve the complex topology, and distribution of the boundary conditions. Such situation requires automatic mesh generator to reduce the user influence to this process as much as possible. The advancing front is another popular mesh generation method that can be used for adapting FE mesh strategies. Conceptually , the advancing front method is one of the simplest mesh generation processes. This element generating algorithm starts from an initial front formed from the specified boundary of the domain and then generates elements, one by one, as the front advances into the region to be discretized until the whole domain is completely covered by elements [9-10]. In general, good quality meshes of quadrilateral elements cannot be directly obtained from these meshing techniques. An additional step is therefore required to obtain quadrilateral meshes from the triangular meshes. It is generally known that FEA using quadrilateral mesh is more accurate than that of a triangular one [28-30].

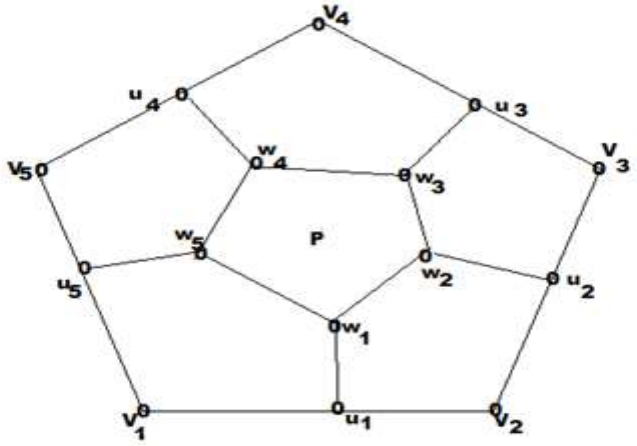
In section 2 of this paper, we begin with a brief description of the refinement scheme for pentagonal partitions. We explain the procedure for pentagonal mesh generation for an irregular Pentagonal domain in section 3.The aim of this section is to present the algorithms to compute the necessary output on nodal coordinates and the element nodal connectivity matrix. In section 4, we consider the mesh generation over some complex domains.By joining several blocks of pentagonal domains,a complex domain can be fully discritised into an all pentagonal finite element mesh We present some examples of these mesh generations

In the next two sections,we propose methods of integrations which are useful in the integration of some arbitrary and smooth functions over two dimensions.In section 5,we present the boundary integration method which is useful in obtaining exact as well as numerical values of some of the complicated integrals[40-44].In section 6,we present some methods for exact integration based on Fubini's theorem and the Boundary Integration theorem. Boundary integration theorem of this paper is found very useful for some integrands over symmetrical and unsymmetrical pentagons. These technique of integration will validate the application of Pentagonal mesh generation for numerical integration in real life applications for Cartesian two space.In section 7,we compute some complicated integrals to demonstrate the utility of the proposed mesh generation scheme and the composite numerical integration scheme which incorporates boundary integration technique. We have also appended the relevant and necessary computer codes developed in this study for furthering research on this interesting topic.

2.0 REFINEMENT SCHEME OF PENTAGONAL PARTITIONS

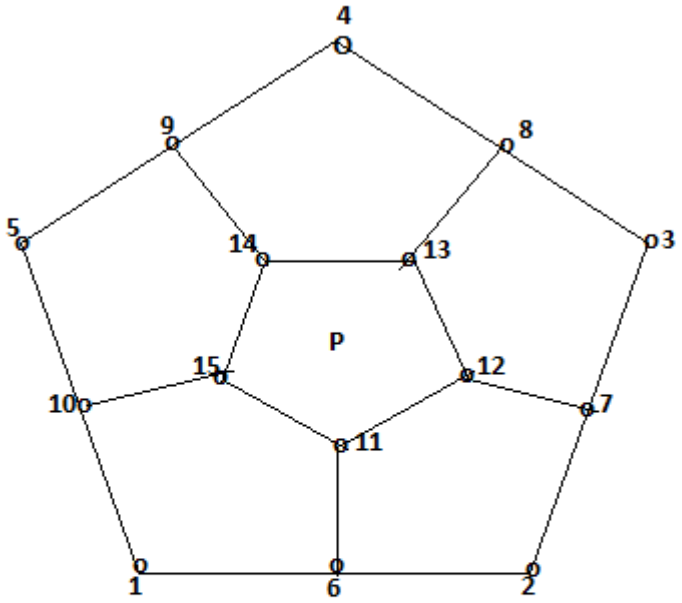
Refinements of a triangulation and a quadriangulation are well-known. Let us first explain how to refine a pentagon partition. We start with a pentagon P =(v_1 , v_2, v_3 , v_4 , v_5) . Let p ==(v_1 + v_2 + v_3 + v_4 + v_5) /5 be the geometric center of P. Write $u_i = (v_i + v_{i+1})/2$ for i = 1,2,3,4,5 with $v_6 = v_1$ and w_i =(u_i+p)/2, i = 1,2,3,4,5 as shown in Fig.1. Connect these u_i,v_i ,w_i,(i = 1, 2,3, 4, 5) as shown to form a uniform refinement of the pentagon P

DOI: 10.18535/ijecs/v4i10.29



(A)

Fig 1. A scheme for the pentagon refinement with vertices v_i , u_i ,w_i (i=1,2,3,4,5)



(B)

Fig.1 A scheme for the pentagon refinement with node numbers nv_i,nu_i,nw_i for the verticces=v_i , midpoints=u_i ,innerpoints=w_i

{nv_i}={1,2,3,4,5},{nu_i}={6,7,8,9,10},{nw_i}={11,12,13,14,15}

Lemma . Let Ω be a convex polygonal region with n corner points. Then Ω can be divided into a collection of convex pentagons if n ≥ 5. So far we still do not know if any polygon Ω can be partitioned into a

collection of strictly convex pentagons although it can be partitioned into a collection of non-strictly convex pentagons. When a pentagon P is not convex, we do not know how to refine it.

3.0 PENTAGONAL ELEMENT MESH GENERATION

We now explain the procedure for pentagonal mesh generation for an irregular Pentagonal domain with reference to Fig.1 above.

3.1 Computing of Nodal Coordinates

We have shown in Fig.1 that the vertices are v_1 , v_2 , v_3 , v_4 , v_5, the midpoint of side joining the vertices v_i , v_{i+1} (i=1,2,3,4,5) with $v_6 = v_1$ are denoted as u_i ,(i=1,2,3,4,5). The interior points are located at $w_i = (u_i + p)/2$,where $p = (v_1+v_2+ v_3 + v_4 + v_5)/5$, is the centre point or centroid of the pentagon P shown in Fig.1. It is possible to refine the given pentagon P recursively for any number of times. But for applications to real problems,we must have the input data of nodal coordinates of all the pentagonal elements as well as the element nodal addresses.Hence,it is important that a systematic procedure is adopted for this purpose.We denote the coordinates of the vertices as (xv_i, yv_i) ,the computed values of midpoint coordinates as (xu_i, yu_i) and the interior point coordinates as (xw_i, yw_i) , where ,(i=1,2,3,4,5), with xv_6 = xv_1 , yv_6 = yv_1. In a similar manner,we denote (nv_i , nu_i , nw_i) ,(i=1,2,3,4,5) as the node numbers for the vertices,midpoint vertices,and interior point vertices.The division of the pentagon P has created the following six new pentagons whose vertices are now a combination of vertices v_i ,midpoint vertices u_i and inner point vertices w_i. Let us denote the given pentagon P as the polygon spanned by the vertices
as P=<v_1 ,v_2 , v_3 , v_4 , v_5>. Following this representation,we can write for the first refinement by introducing yet another notation:
v_i =$v_i^{(0)}$, u_i =$u_i^{(0)}$, w_i =$w_i^{(0)}$, (i=1,2,3,4,5) with v_6 = $v_6^{(0)}$, p=$p^{(0)}$ as the vertices,midpoint vertices and innerpoint vertices and centre point for the given pentagon P=$P^{(0)}$.Then the pentagons ($P_k^{(n)}$,k=1,2,3,4,5,6 ;n=1,2,.....,6^n) so generated by recursive process after nth refinement are spanned by the following equations.

$P_1^{(n)}$=<$v_1^{(n-1)}$, $u_1^{(n-1)}$,$w_1^{(n-1)}$,$w_5^{(n-1)}$, $u_5^{(n-1)}$> ,
$P_2^{(n)}$=<$v_2^{(n-1)}$, $u_2^{(n-1)}$,$w_2^{(n-1)}$,$w_1^{(n-1)}$, $u_1^{(n-1)}$> ,
$P_3^{(n)}$=<$v_3^{(n-1)}$, $u_3^{(n-1)}$,$w_3^{(n-1)}$,$w_2^{(n-1)}$, $u_2^{(n-1)}$> ,
$P_4^{(n)}$=<$v_4^{(n-1)}$, $u_4^{(n-1)}$,$w_4^{(n-1)}$,$w_3^{(n-1)}$, $u_3^{(n-1)}$> ,
$P_5^{(n)}$=<$v_5^{(n-1)}$, $u_5^{(n-1)}$,$w_5^{(n-1)}$,$w_4^{(n-1)}$, $u_4^{(n-1)}$> ,
$P_6^{(n)}$=<$w_1^{(n-1)}$, $w_2^{(n-1)}$,$w_3^{(n-1)}$,$w_4^{(n-1)}$, $w_5^{(n-1)}$> . .....(1a)

Where,
$u_i^{(n-1)} = \frac{v_i^{(n-1)}+v_{i+1}^{(n-1)}}{2}$, $v_6^{(n-1)} = v_1^{(n-1)}$
$p^{(n-1)} = (v_1^{(n-1)} + v_2^{(n-1)} + v_3^{(n-1)} + v_4^{(n-1)} + v_5^{(n-1)})/5$
$w_i^{(n-1)} = (u_i^{(n-1)} + p^{(n-1)})/2$
n=1,2,3,.....
i=1,2,3,4,5 .....(1b)

3.2 A Brief Algorithm to Compute Nodal Coordinates

(i)Input the coordinates vertices for the given pentagon,say ,P.
(ii)Draw the outline of the pentagon P using MATLAB command plot.
(iii)Compute the centre point of the pentagon P.
(iv)Compute midpoints of sides
(v)Compute midpoint of of the line joining the centre point and midpoint of sides**.**
(vi)Use this compted data to draw six new smaller pentagons inside the given pentagon P.
(VII)Now use recursive refinement on each of the six new pentagons.

3.3 Computing Nodal Connections or Nodal Addresses: A Brief Algorithm

We find that the generation of nodal coordinates can be drawn by using the above algorithm.We can draw various refinements of the given pentagon P using the above algorithm which does not depict the element nodal connections.But such refinements will be of no use for practical applications in real world problems.However,it is not easy to generate element nodal connections. We have to pay careful attention to the following guidelines while generating the element nodal connections.
Let us first associate one node number to each nodal point of the given pentagon P.
We associate the node numbers as follows:
(i)Node number **nv** is assigned to vertex nodal coordinate **(xv,yv**).
(ii)Node number **nu** is assigned to midpoint nodal coordinate **(xu,yu**).
(iii) Node number **nw** is assigned to interior poiunt nodal coordinate **(xw,yw**).
Then we have pay careful attention to the following points:
(i)There are five sides to pentagon P.
(ii)We first assign node numbers to the vertex coordinate points of the five sides
(iii)We then find whether these sides are shared by other pentagonal elements.
(iv) We then find whether midpoint node numbers are already determined.
(v) We need not find a midpoint node number for a side which has already occurred in previous element or elements
We have shown the mesh refinements of first four iteration for n=1,2,3,4 by assuming the starting vertex node numbers as 1,2,3,4,5 for a typical Pentagonal element.These are depicted in figures Fig.2, Fig.3,and Fig.4a, Fig.4b.

DOI: 10.18535/ijecs/v4i10.29

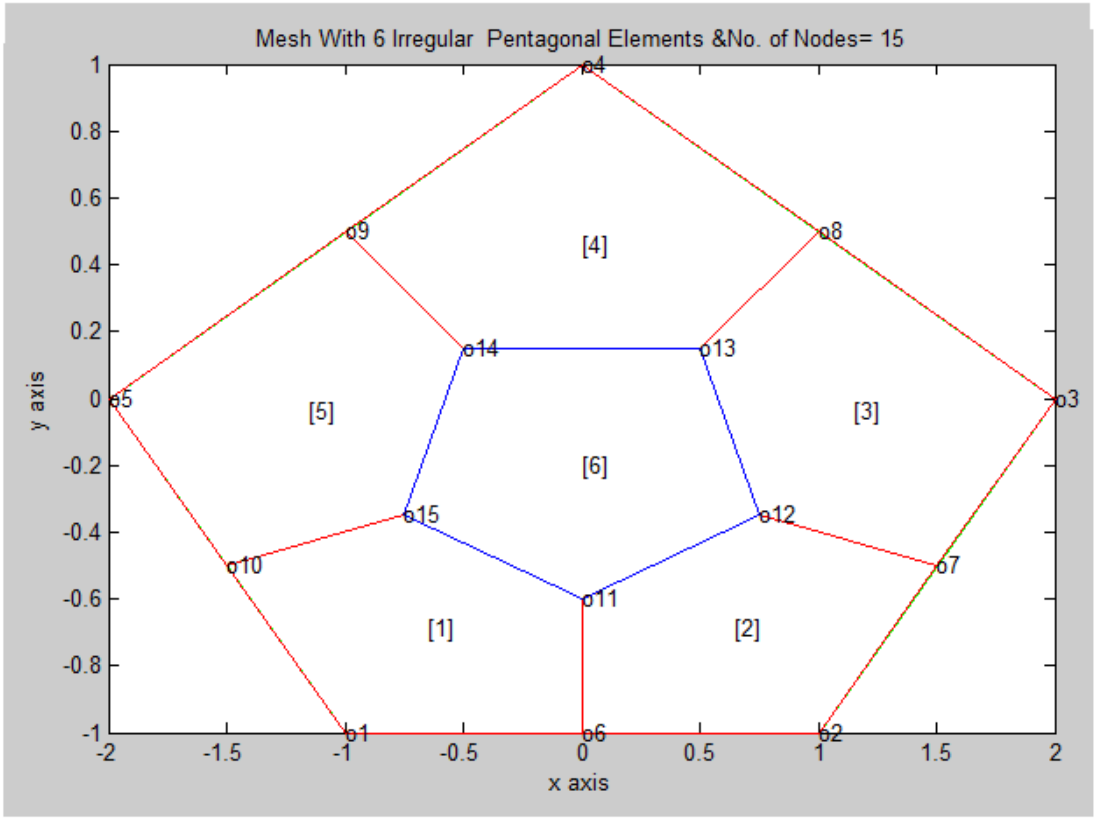


Fig.2:PENTAGONAL DIVISION OF A PENTAGON-FIRST REFINEMENT

DOI: 10.18535/ijecs/v4i10.29

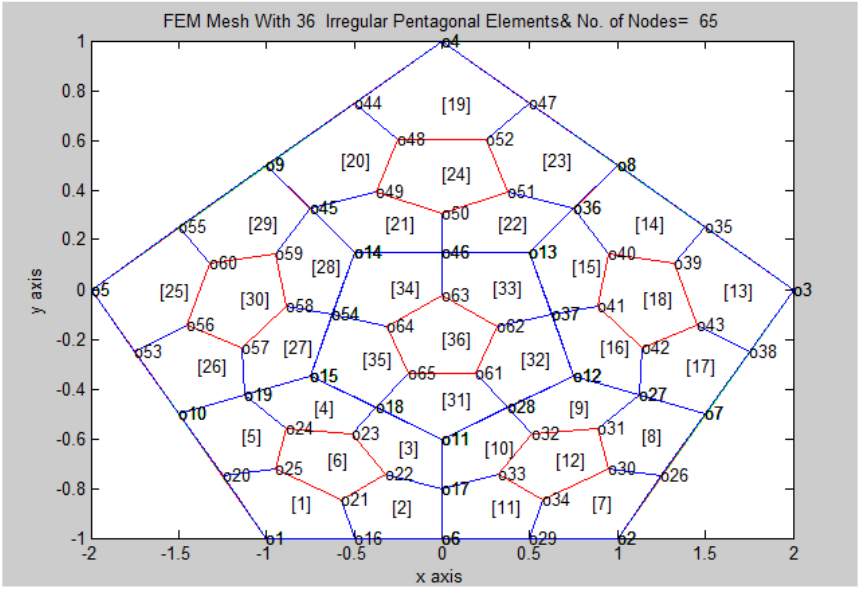


Fig.3 PENTAGONAL DIVISION OF A PENTAGON-SECOND REFINEMENT

We next present the third and fourth refinements for Pentagonal Divisions

In Fig.4a third refinement for Pentagonal Division of a symmetrical Pentagon is shown,this figure clearly depicts the node numbers as well as element noumbers surrounded by nodal connections

In Fig.4b fourth refinement for Pentagonal Division of a symmetrical Pentagon is shown,this figure does not depict the node numbers ,element noumbers surrounded by nodal connections because it is simply impossible!

DOI: 10.18535/ijecs/v4i10.29

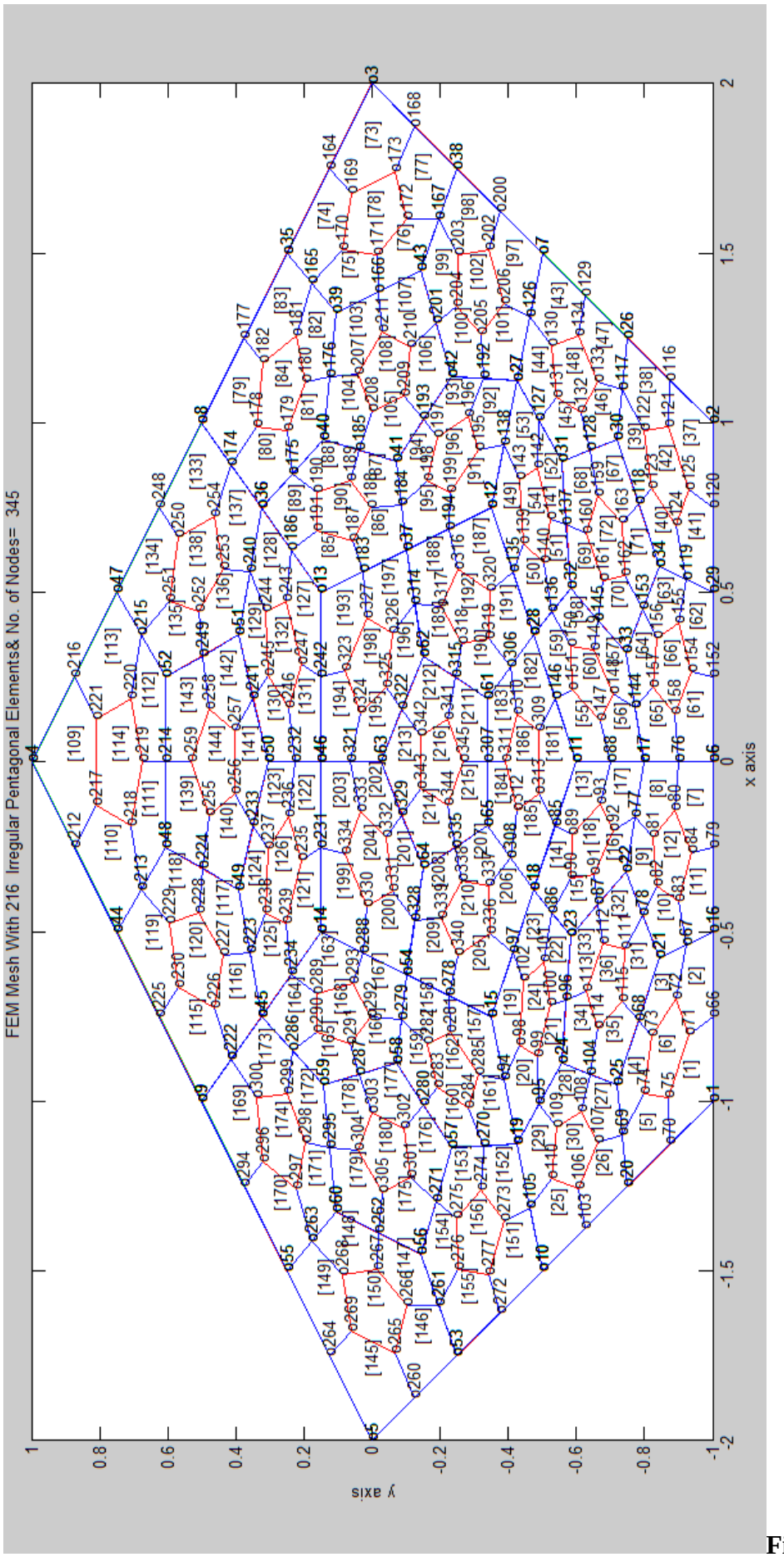
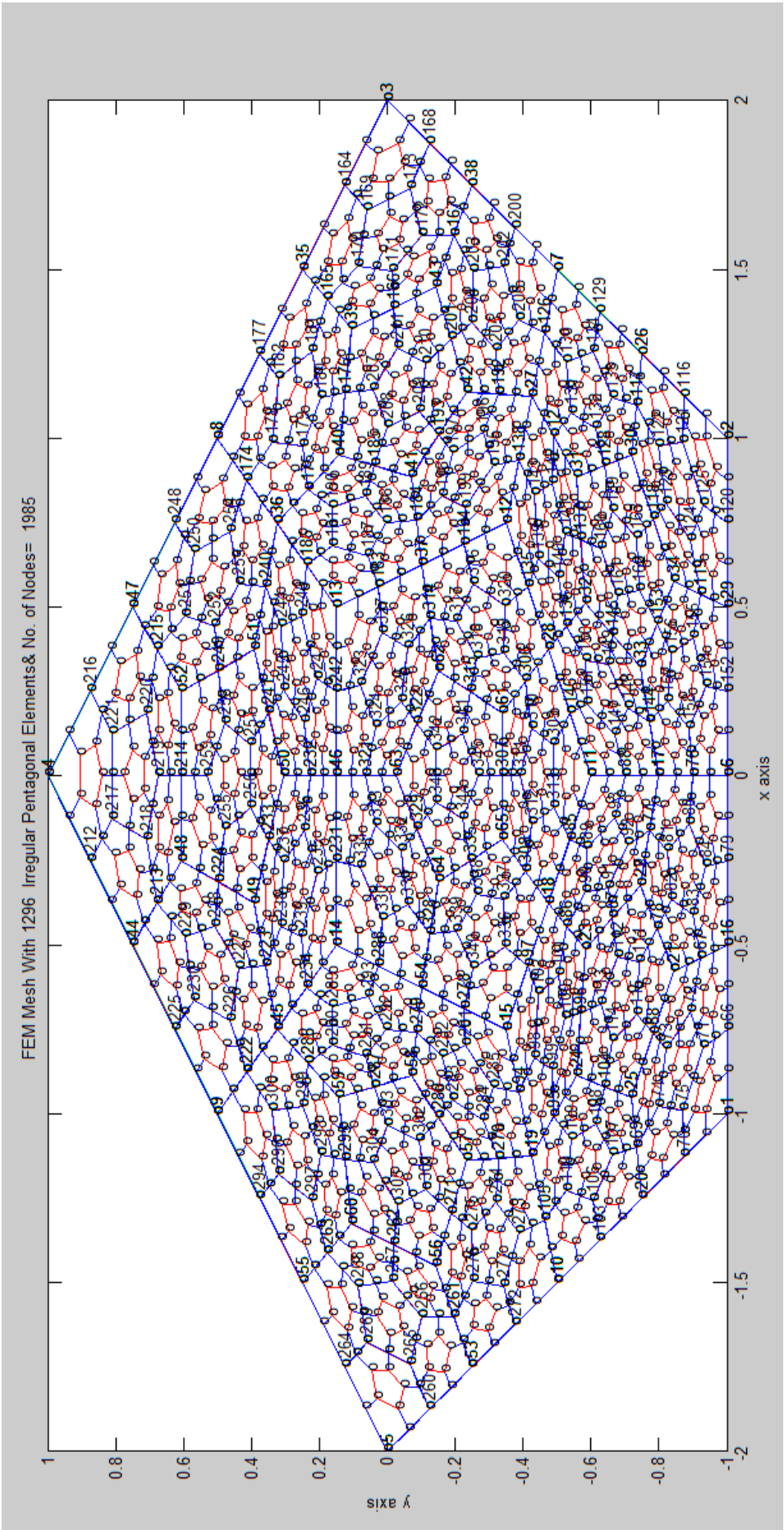


Fig.4-a rotated

DOI: 10.18535/ijecs/v4i10.29



DOI: 10.18535/ijecs/v4i10.29

Fig.4b-rotated

4.0 MESH GENERATION FOR COMPLEX POLYGONAL DOMAINS

We now consider mesh generation over some complex domains.By joining several blocks of pentagonal domains,we can generate an all pentagonal complex domain. We present some examples of these domains.The detais of the mesh generation scheme can be studied from the appended computer programs

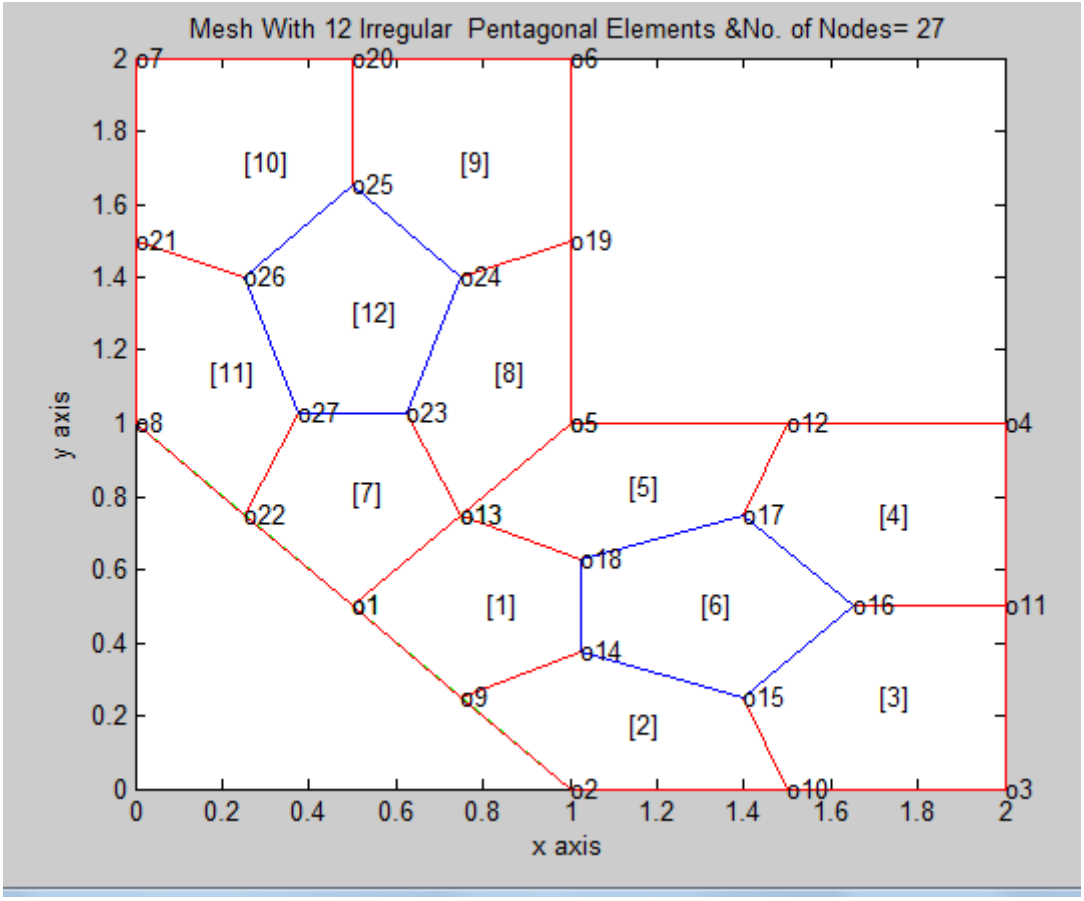


Fig.4c: JOINING OF TWO PENTAGONS

DOI: 10.18535/ijecs/v4i10.29

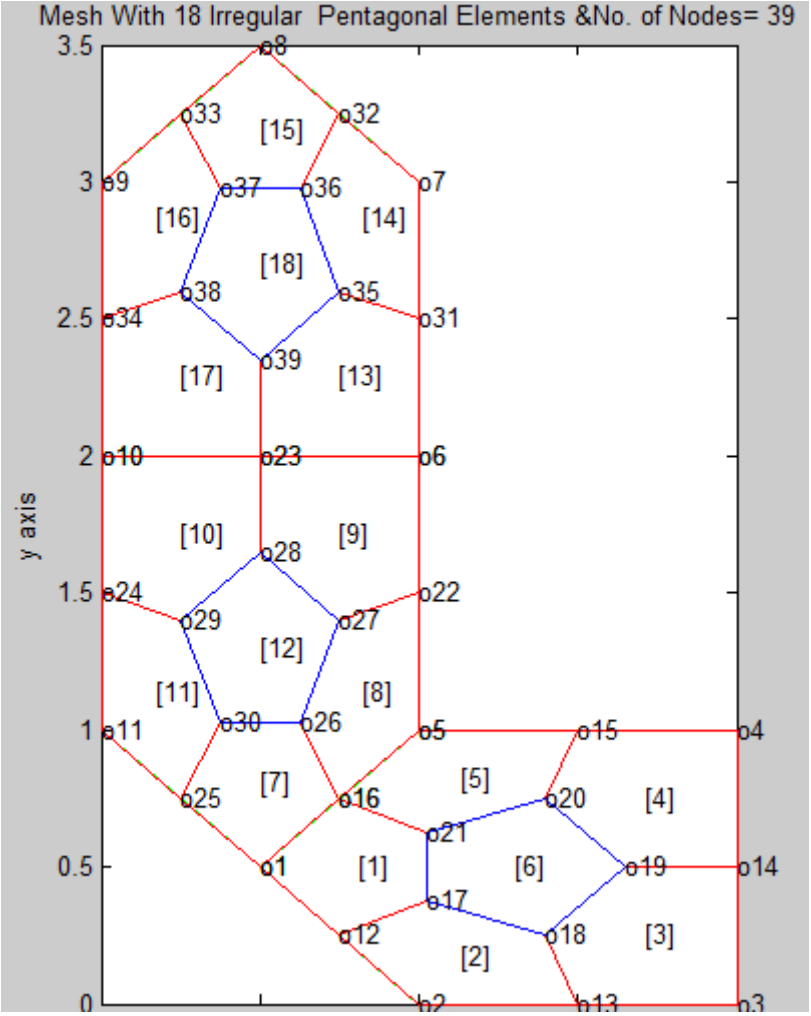


Fig.4d: JOINING OF THREE PENTAGONS

DOI: 10.18535/ijecs/v4i10.29

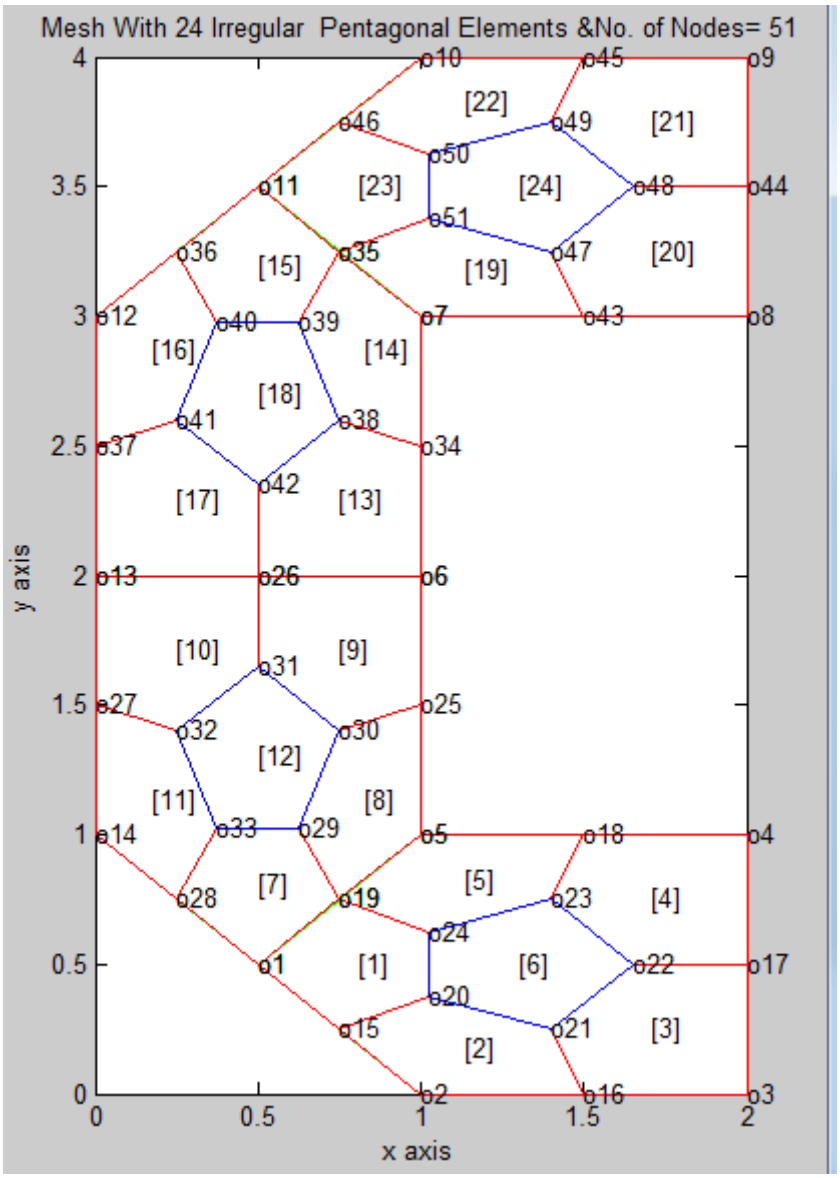


Fig.e: JOINING OF FOUR PENTAGONS

DOI: 10.18535/ijecs/v4i10.29

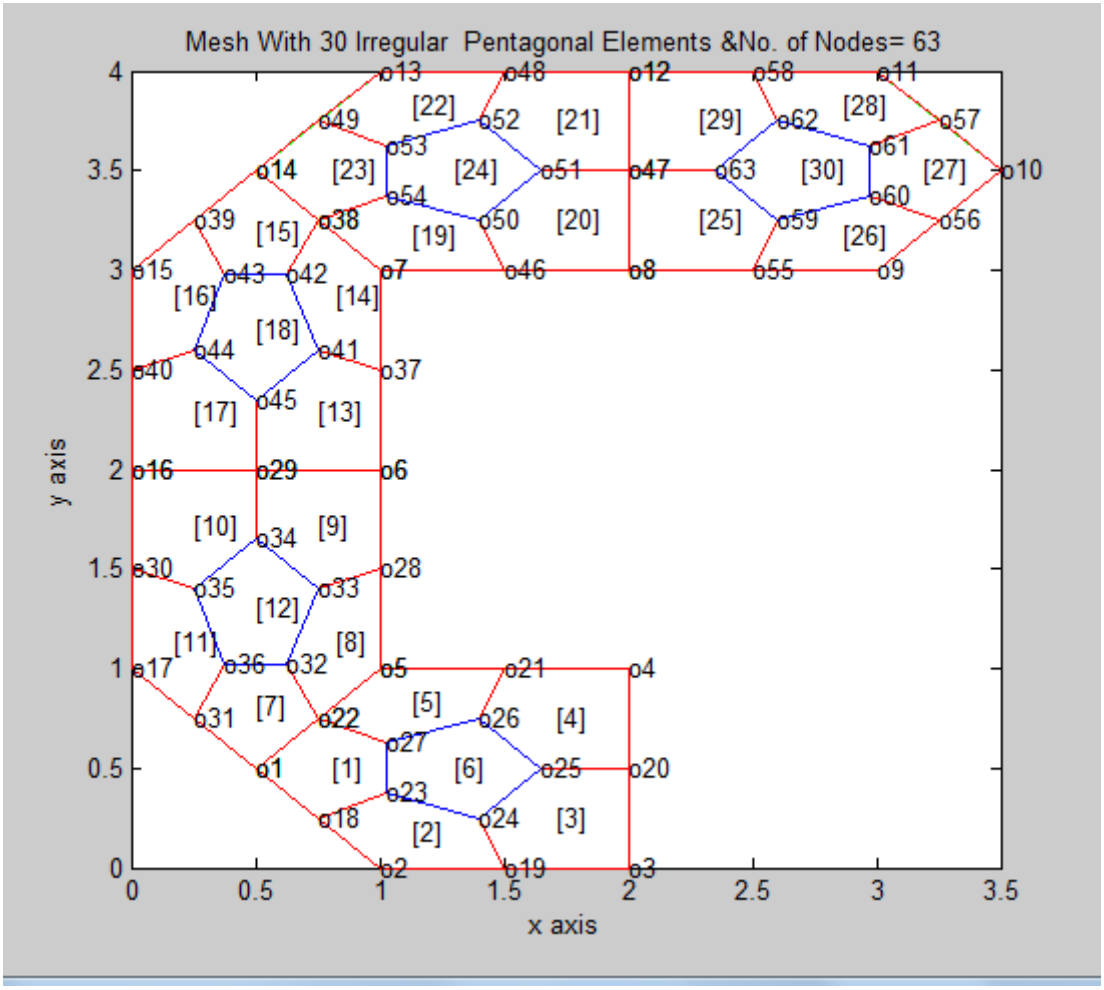


Fig.:4f JOINING OF FIVE PENTAGONS

DOI: 10.18535/ijecs/v4i10.29

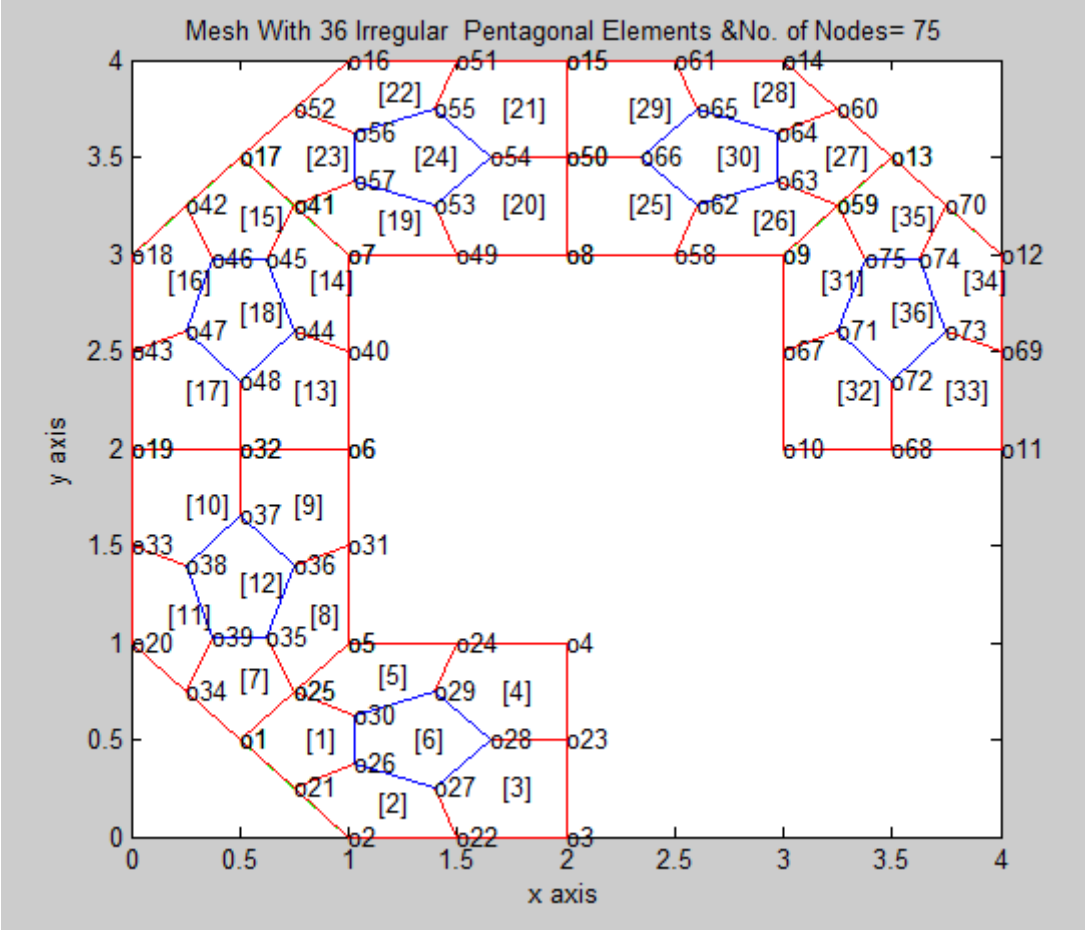


Fig.:4g JOINING OF SIX PENTAGONS

DOI: 10.18535/ijecs/v4i10.29

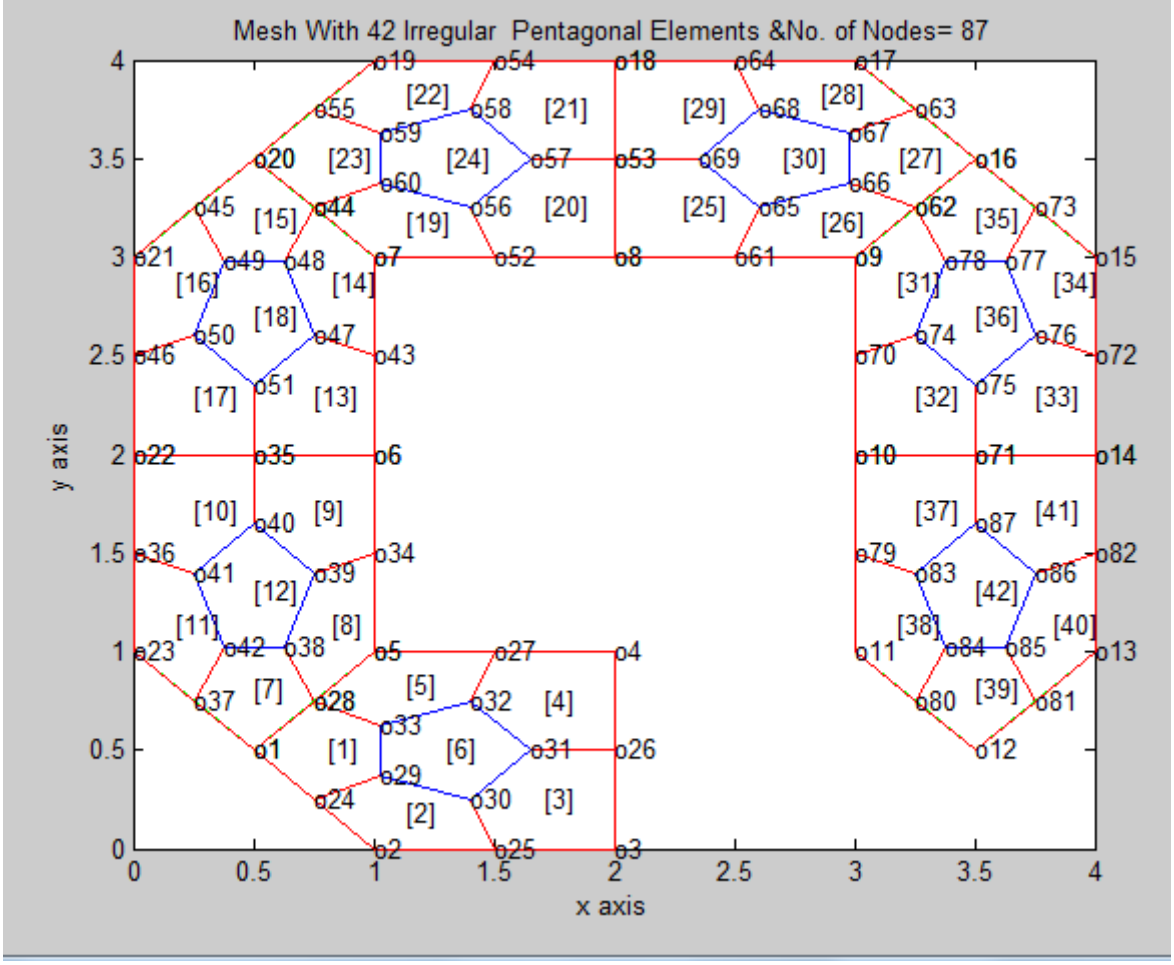


Fig.:4h JOINING OF SEVEN PENTAGONS

DOI: 10.18535/ijecs/v4i10.29

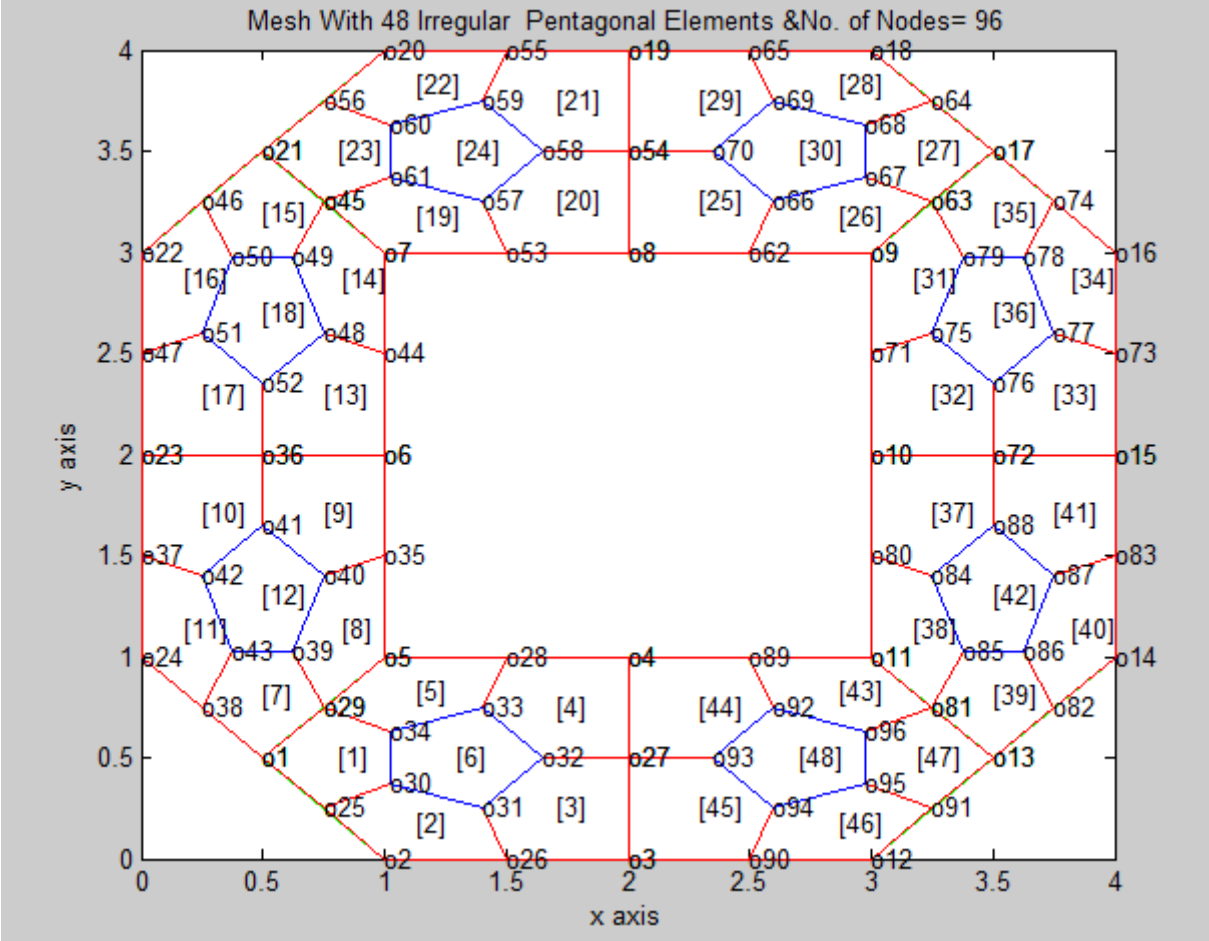


Fig.:4i JOINING OF EIGHT PENTAGONS

DOI: 10.18535/ijecs/v4i10.29

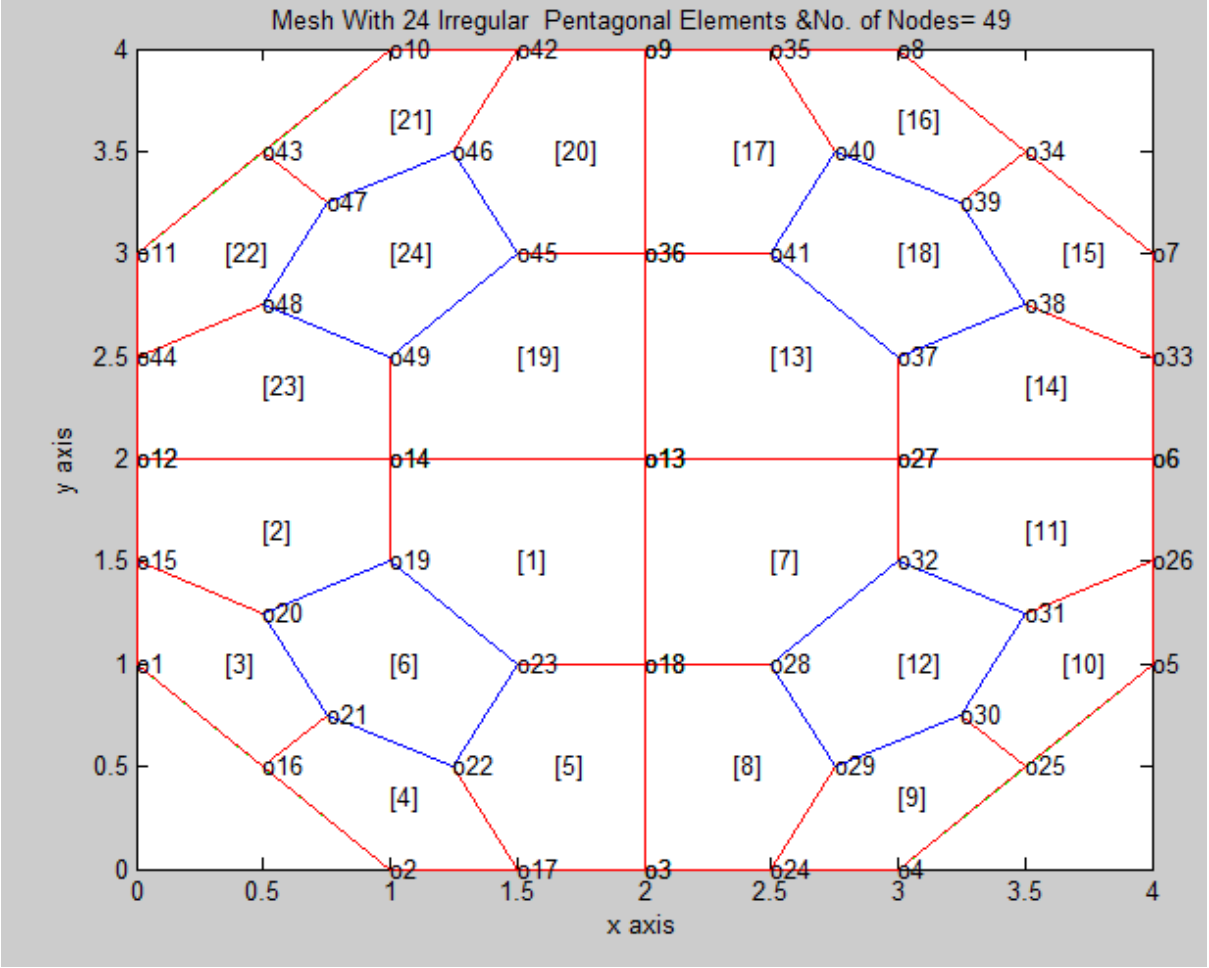


Fig.4j FOUR PENTAGONS TO FORM A SQUARE WITH CORNER CUTS

DOI: 10.18535/ijecs/v4i10.29

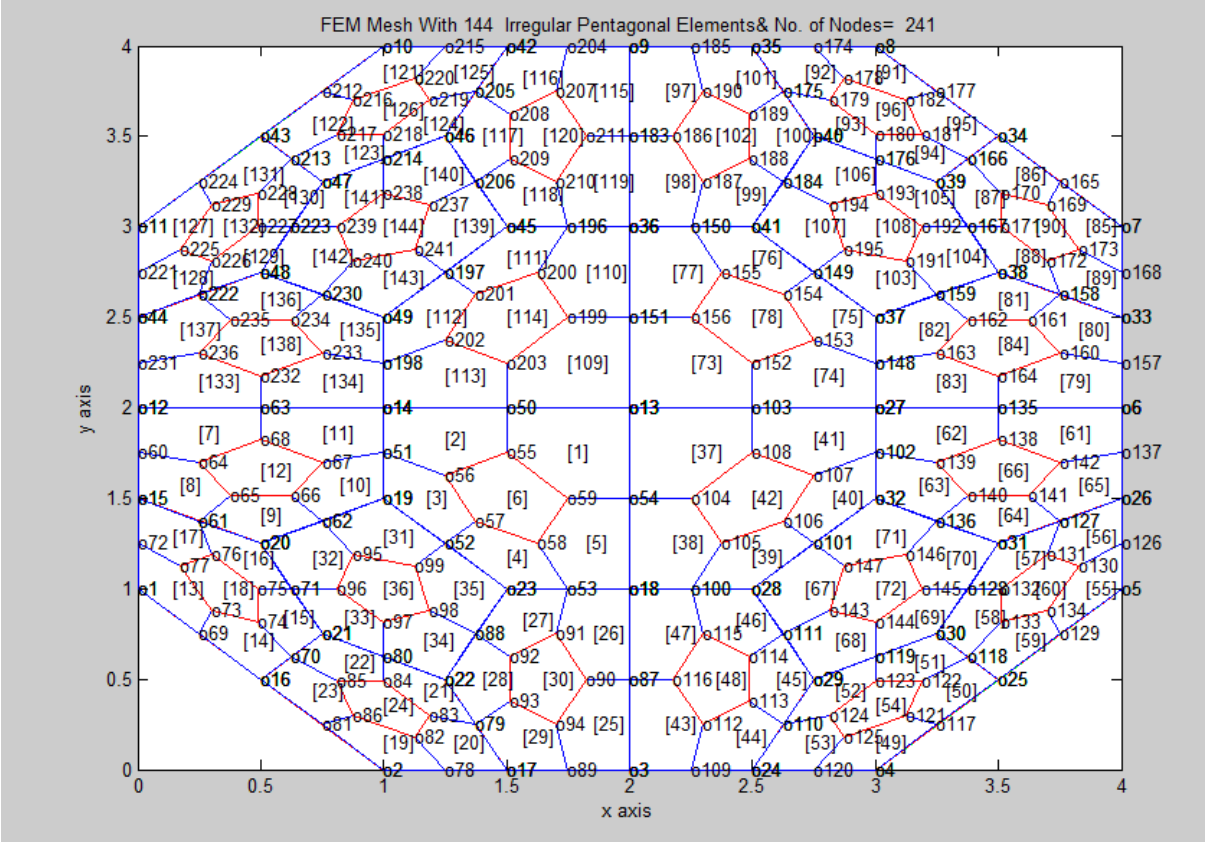


Fig.4k FOUR PENTAGONS TO FORM A SQUARE WITH CORNER CUTS
Second refinement mesh

DOI: 10.18535/ijecs/v4i10.29

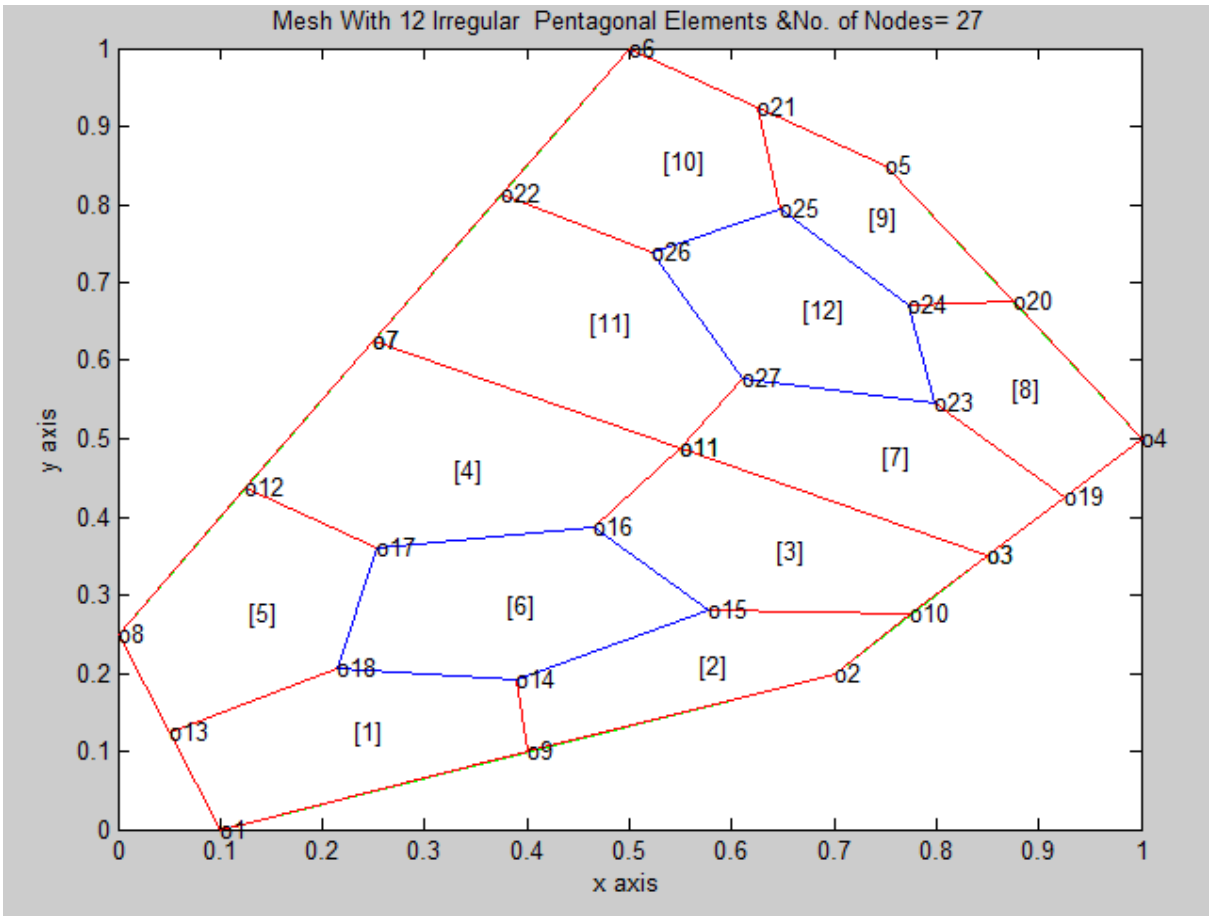
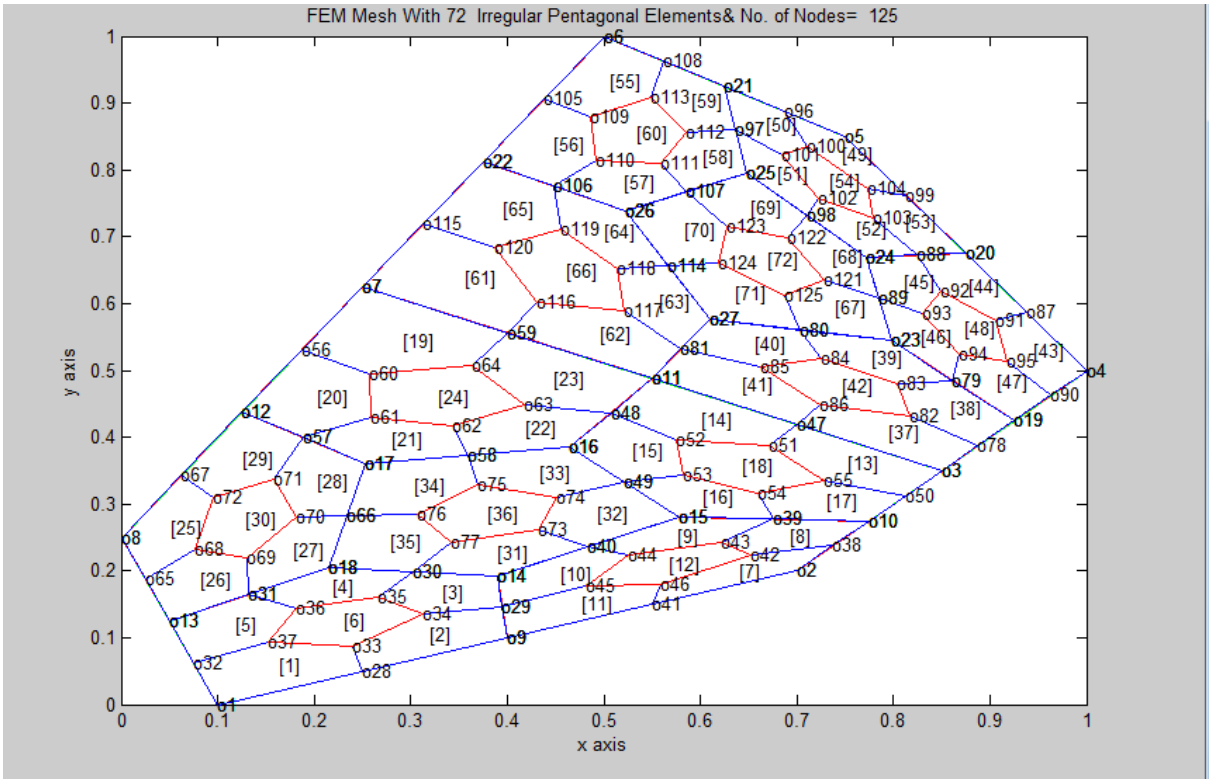


Fig.4l:Two pentagons are joined to form a convex six-gon(first refinement)



DOI: 10.18535/ijecs/v4i10.29

Fig.4m Two pentagons are joined to form a convex six-gon(second refinement)

In the next two sections,we propose methods of integrations which are useful in the integration of some arbitrary and smooth functions over two dimensions.In section 5,we present the boundary integration method which is useful in obtaining exact as well as numerical values of some of the complicated integrals[40-43].In section 6,we present a method for exact integration based on Fubini's theorem and boundary integration theorem of this paper which is found very useful in some special cases [ ].These technique of integration will validate the application of Pentagonal mesh generation for numerical integration for real life applications in Cartesian two space.

5.0 Boundary Integration Method For Two Dimensions

Let π_{xy} be a simple polygon in the *xy*-plane. We want to evaluate the following integral

$$II_{\pi_{xy}} \overset{def}{=} \iint_{\pi_{xy}} f(x,y)dxdy \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

Theorem 1: The integral $II_{\pi_{xy}}$ over a simple polygon with *n*-oriented edges l_{ki} ,(i=k+1),k=1,2,3,.....N each with the end points (x_k, y_k) and (x_i, y_i) in the *xy*-plane and $(x_{N+1}, y_{N+1}) = (x_1, y_1)$ is expressible as

$$II_{\pi_{xy}} = \sum_{i=1}^{N} \iint_{T_{iok}^{xy}} f(x,y)dxdy \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Where T_{iok}^{xy} refers to the triangle in the *xy*-plane with vertices at $(x_i y_i)$, (0,0), *and* $(x_k y_k)$, $(k = i+1)$

Proof: Let us consider the following expression (see Fig. 5 and 6).

$$\left(\iint_{T_{joi}^{xy}} + \iint_{T_{koj}^{xy}} + \iint_{T_{iok}^{xy}} \right) f(x,y)dxdy \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3)$$

$$= \left(\oint_{\partial T_{joi}^{xy}} + \oint_{\partial T_{koj}^{xy}} + \oint_{\partial T_{iok}^{xy}} \right) \Phi(x,y)dy$$

$$= \left(\int_{l_{jo}} + \int_{l_{oi}} + \int_{l_{ij}} \right) \Phi(x,y)dy + \left(\int_{l_{ko}} + \int_{l_{oj}} + \int_{l_{jk}} \right) \Phi(x,y)dy + \left(\int_{l_{io}} + \int_{l_{ok}} + \int_{l_{ki}} \right) \Phi(x,y)dy$$

DOI: 10.18535/ijecs/v4i10.29

$$= \left(\int_{l_{ij}} + \int_{l_{jk}} + \int_{l_{ki}} \right) \Phi(x,y)dy$$

$$= \oint_{\partial T_{ijk}^{xy}} \Phi(x,y)dy$$

$$= \iint_{T_{ijk}^{xy}} f(x,y)dxdy \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (4)$$

Thus, we have proved that the triangle T_{ijk}^{xy} expands into three new triangles with respect to the origin.

In the above derivations, we have used the fact that

$$\int_{l_{oi}} \Phi(x,y)dy + \int_{l_{io}} \Phi(x,y)dy = 0$$

$$\int_{l_{jo}} \Phi(x,y)dy + \int_{l_{oj}} \Phi(x,y)dy = 0 \ldots\ldots\ldots\ldots\ldots\ldots (5)$$

The general result of eqn (2) can be readily proved on similar lines. This completes the proof of the Theorem-1.

Theorem 2: The integral over the triangle spanned by vertices (x_i, y_i), $(0, 0)$ and (x_k, y_k), $(k=i-1)$ which we have denoted as T_{iok}^{xy} in eqn (2) is expressible as,

$$I\ I_{T_{iok}^{xy}} = (x_k y_i - x_i y_k) \int_0^1 \int_o^1 rf(r(x_i + x_{ki}s), r(y_i + y_{ki}s))\, drds, \text{ Where } x_{ki} = x_k - x_i\ , y_{ki} = y_k - y_i \ldots\ldots (6)$$

Proof: Let us consider the integral

$$I\ I_{T_{ijk}^{xy}} = \iint_{T_{ijk}^{xy}} f(x,y)dxdy \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (7)$$

The parametric equations of the oriented triangle in the *xy*-plane with vertices spanned by

(x_i, y_i), (x_j, y_j) and (x_k, y_k), $(k = i+1)$which map this arbitrary triangle into a unit right isosceles triangle in the *uv*-plane are (see Fig. 7)

DOI: 10.18535/ijecs/v4i10.29

$$x = x_i + x_{ji}u + x_{ki}v$$
$$y = y_i + y_{ji}u + y_{ki}v \quad \text{.....(8)}$$

where,

$$0 \le u, v \le 1, u + v \le 1$$
$$x_{ji} = x_j - x_i, x_{ki} = x_k - x_i$$
$$y_{ji} = y_j - y_i, y_{ki} = y_k - y_i \quad \text{.....(9)}$$

We have then

$$\begin{aligned} dxdy &= \frac{\partial(x,y)}{\partial(u,v)} dudv \\ &= (x_{ji}y_{ki} - x_{ki}y_{ji})dudv \\ &= (2\Delta_{ijk}^{xy})dudv \\ &= (2 \times \text{area of triangle } T_{ijk}^{xy})dudv \quad \text{.....(10)} \end{aligned}$$

and thus we define

$$2\Delta_{ijk}^{xy} = (x_{ji}y_{ki} - x_{ki}y_{ji}) \quad \text{.....(11)}$$

Use of eqns (8)-(11) into eqn (7) gives us

$$II_{T_{ijk}^{xy}} = (2\Delta_{ijk}^{xy}) \int_o^1 \int_o^{1-u} f((x_i + x_{ji}u + x_{ki}v), (y_i + y_{ji}u + y_{ki}v)) \, dudv \quad \text{.....(12)}$$

Let us now map the above integral in eqn (12) into an equivalent integral over the rectangle $\{(r,s)/0 \le r, s \le 1\}$ by the transformation (see Fig.8).

$$u = 1 - r, \; v = rs \quad \text{.....(13)}$$

Use of eqn (13) in eqn (12) gives us

DOI: 10.18535/ijecs/v4i10.29

$x_i + x_{ji}u + x_{ki}v = x_i + (x_j - x_i)(1-r) + x_{ki}rs$

$y_i + y_{ji}u + y_{ki}v = y_i + (y_j - y_i)(1-r) + y_{ki}rs$.....(14)

Letting $x_j = 0, y_j = 0$ in eqn (14) gives

$x_i + x_{ji}u + x_{ki}v = r(x_i + x_{ki}s)$

$y_i + y_{ji}u + y_{ki}v = r(y_i + y_{ki}s)$.....(15)

$2\Delta_{iok}^{xy} = (0 - x_i)y_{ki} - x_{ki}(0 - y_i) = (x_k y_i - x_i y_k)$

and from eqn (13)

$dudv = -rdrds$.....(16)

and the limits : $u = 0, u = 1$, correspond to $r = 1, r = 0$

$v = 0, v = 1 - u$, correspond to $s = 0, s = 1$

Thus from eqns (13)-(16), we obtain

$$II_{T_{iok}^{xy}} = (x_k y_i - x_i y_k) \int_o^1 \int_o^1 rf(r(x_i + x_{ki}s), r(y_i + y_{ki}s))drds \quad \text{.....(17)}$$

This completes the proof of the theorem 2.

DOI: 10.18535/ijecs/v4i10.29

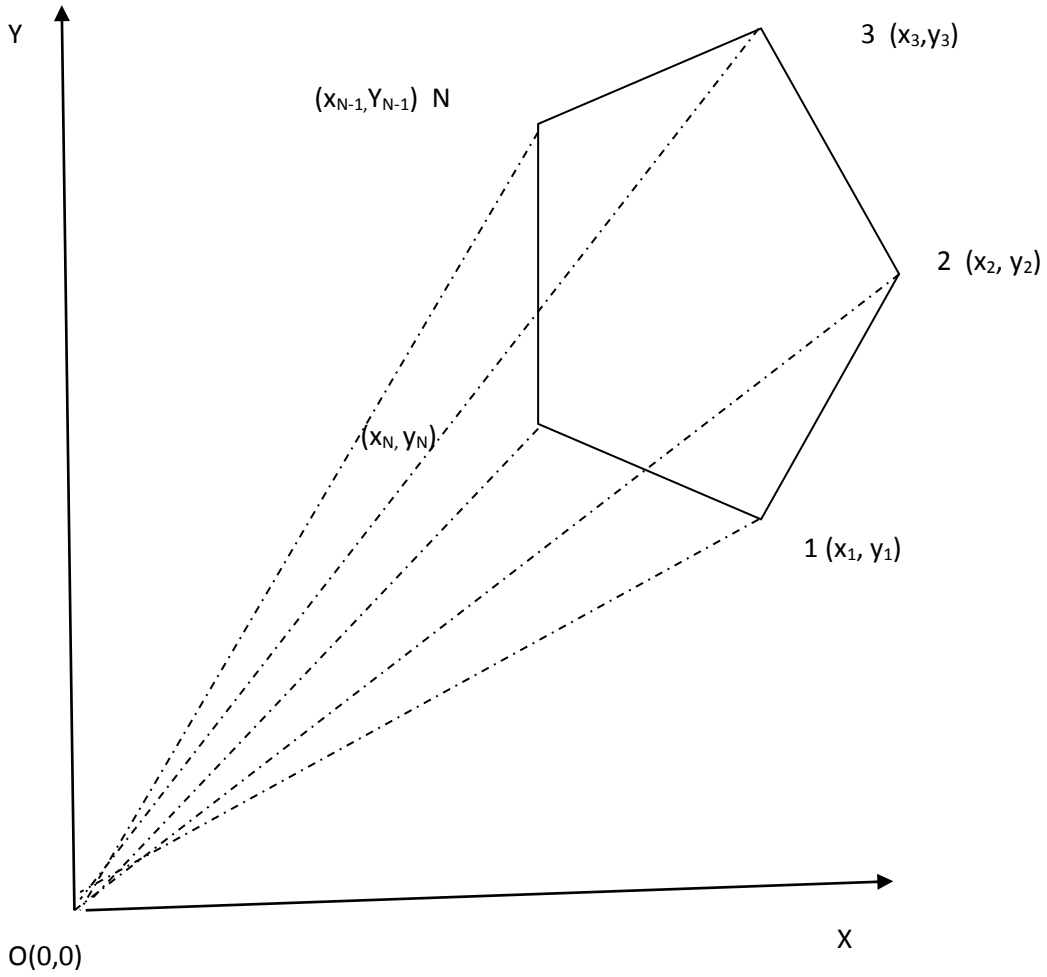


Fig 5. π_{xy} : A simple polygon in the xy-plane with N-oriented edges which expands into N-triangles with respect to the origin.

DOI: 10.18535/ijecs/v4i10.29

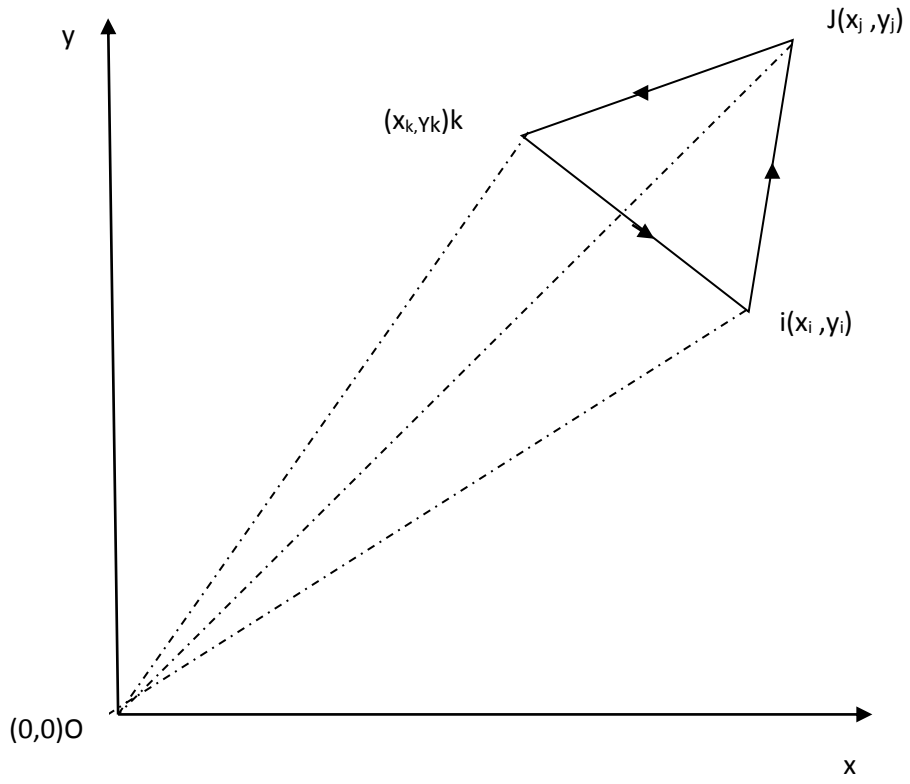
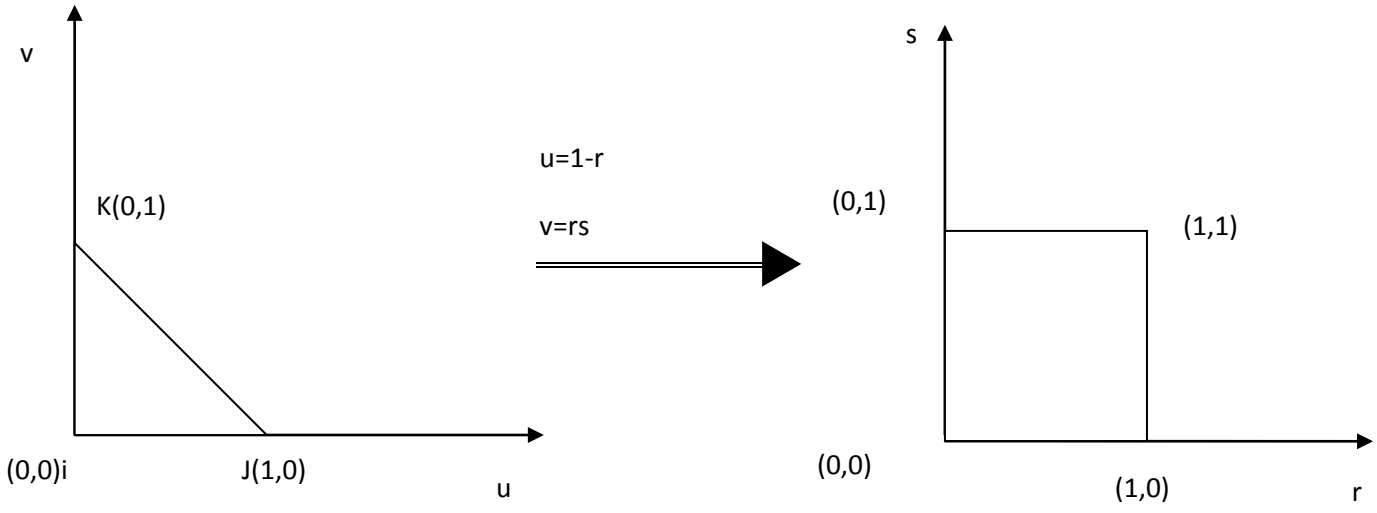


Fig 6: A linear triangle in the *xy*-plane with three oriented edges which expands into 3-triangles with respect to the origin.

7: The mapping between an oriented triangle in the xy-plane and the unit right isoscles triangle in the uv-plane.



DOI: 10.18535/ijecs/v4i10.29

Fig. 8: The mapping of a unit right isosceles triangle(standard triangle) in uv-plane into a unit rectangle in the rs - space

Next we present an exact integration formula over a typical pentagon to validate the boundary integration scheme and the pentagonal finite element mesh generation scheme proposed in the previous sections

6.0 EXACT INTEGRATION IN TWO DIMENSIONS

In mathematical analysis Fubini's theorem, named after Guido Fubini, is a result which gives conditions under which it is possible to compute a double integral using iterated integrals. As a consequence it allows the order of integration to be changed in iterated integrals.In the present context the integration over two dimensional domain can be represented by using Fuboni's theorem which is stated below.

Fubini's Theorm on Non Rectangular Domains

Theorem

If $f : D \subset \mathbb{R}^2 \to \mathbb{R}$ *is continuous in* D, *then hold:*

(a) *(Type I) If* $D = \{(x,y) \in \mathbb{R}^2 : x \in [a,b],\ y \in [g_1(x), g_2(x)]\}$, *with* g_1, g_2 *continuous functions on* $[a,b]$, *then*

$$\iint_D f(x,y)\,dx\,dy = \int_a^b \int_{g_1(x)}^{g_2(x)} f(x,y)\,dy\,dx.$$

(b) *(Type II) If* $D = \{(x,y) \in \mathbb{R}^2 : x \in [h_1(y), h_2(y)],\ y \in [c,d]\}$, *with* h_1, h_2 *continuous functions on* $[c,d]$, *then*

$$\iint_D f(x,y)\,dx\,dy = \int_c^d \int_{h_1(y)}^{h_2(y)} f(x,y)\,dx\,dy.$$

..... (18)

The above stament means that the domain to be integrated can be represented by any one of the following combinations

- Four constant lines
- Three constant lines and one function
- Two costant lines and two functions

We first consider a typical symmetrical pentagonal domain P shown in Fig. 9

DOI: 10.18535/ijecs/v4i10.29

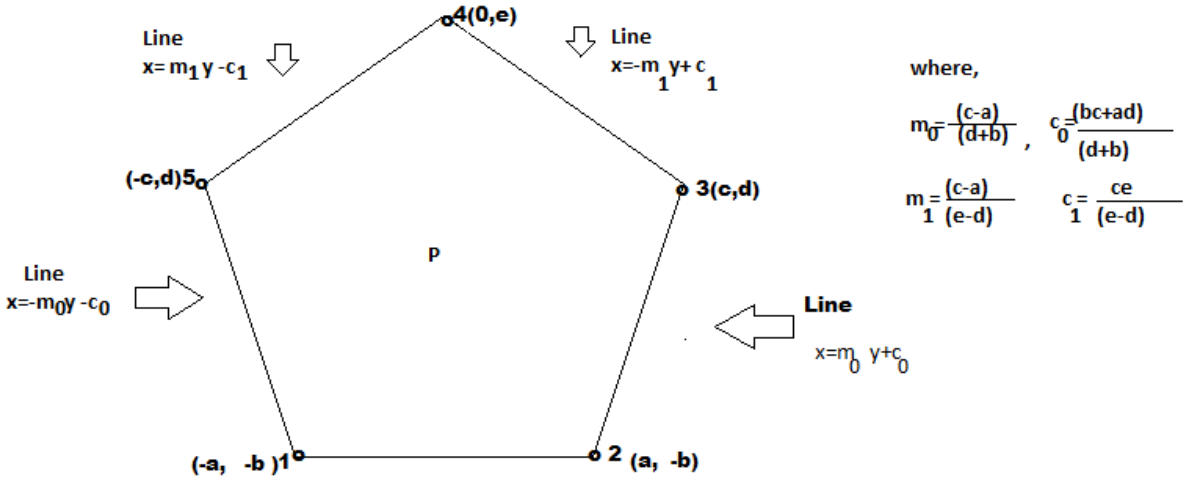


Fig.9- A Typical Pentagon P Symmetrical about y axis

Now using Fubini's Theorem,we can write:

$$\iint_P f(x,y)dxdy = \int_d^e \int_{x=m_1y-c_1}^{x=-m_1y+c_1} f(x,y)dxdy + \int_{-b}^{d} \int_{x=-m_0y-c_0}^{x=m_0\ y+c_0} f(x,y)dxdy \text{ .....(19)}$$

We shall also present, after a while present an exact value of the above integral using eqn(2) andeqn(6) of boundary integration theorem when $f(x,y) = (px+qy)^m$,where ,$p, q, and\ m$ are real and arbitrary.

We next consider a typical **unsymmetrical pentagon P**

DOI: 10.18535/ijecs/v4i10.29

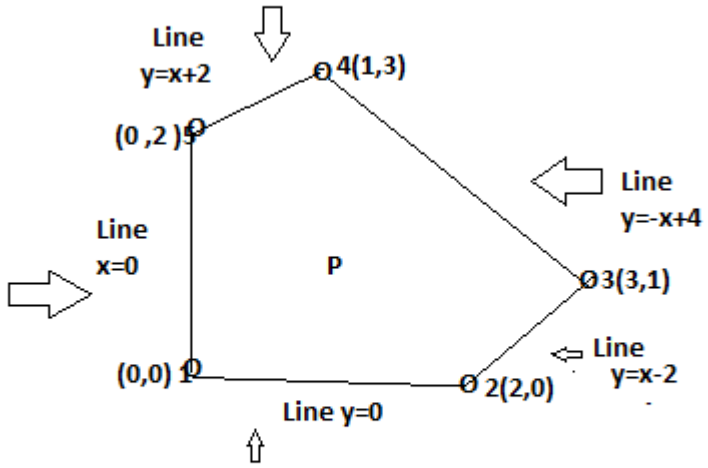


Fig.10: A typical unsymmetrical pentagon P

Now using Fubini's Theorem over the domain shown in Fig 10,we obtain:

$$\iint_P f(x,y)dxdy = \int_0^1 \int_{y=0}^{y=x+2} f(x,y)dydx + \int_1^2 \int_{y=0}^{y=x+2} f(x,y)dydx$$

$$+\int_2^3 \int_{x-2}^{-x+4} f(x,y)dydx \quad \text{.....(20)}$$

We note the following exact value for the above integral, when $f(x,y) = (c_1\, x + c_2\, y)^M$ which is given in literature [41] as

$$\frac{M!}{(M+2)!}\left(\frac{(2c_1)^{M+2}}{c_1(-c_1-c_2)} + 4\frac{(3c_1+c_2)^{M+2}}{(c_1+c_2)(2c_1-2c_2)} + 4\frac{(c_1+3c_2)^{M+2}}{(c_1+c_2)(-2c_1+2c_2)} + \frac{(2c_2)^{M+2}}{(-c_1-c_2)c_2}\right) \quad \text{.....(21)}$$

Where, c_1, c_2, M *are real and arbitrary*

We must now add to the above result that for $c_1 + c_2 \neq 0$ *and* $c_1 - c_2 \neq 0$.
That is, the above result is not valid when $c_1 + c_2 = 0$ *or* $c_1 - c_2 = 0$
These results are presented here for completeness.

(i) When $c_1 + c_2 = 0$, *letting* $c_1 = 1$ *and* $c_2 = -c_1$, so that we obtain:

DOI: 10.18535/ijecs/v4i10.29

$\int_{pentagon}(x-y)^M$ dxdy =0, $if\ M\ is\ odd$

$=2^{M+2}\ \left[\frac{1}{(M+1)(M+2)}+\frac{1}{(M+1)}\right], if\ M\ \ is\ even$

.....(22)

(ii) When $c_1 - c_2 = 0$, $letting\ c_1 = 1\ and\ \ c_2 = 1$,we obtain

$\int_{pentagon}(x+y)^M$ dxdy $= \frac{2*4^{M+1}}{(M+1)} - \frac{2^{M+2}}{(M+1)(M+2)}$ .....(23)

We now display the first four refinements of all Pentagonal mesh generation for the above unsymmetrical pentagon shown in Fig.10.These are displayed in Figs.10a,10b,10c,10d

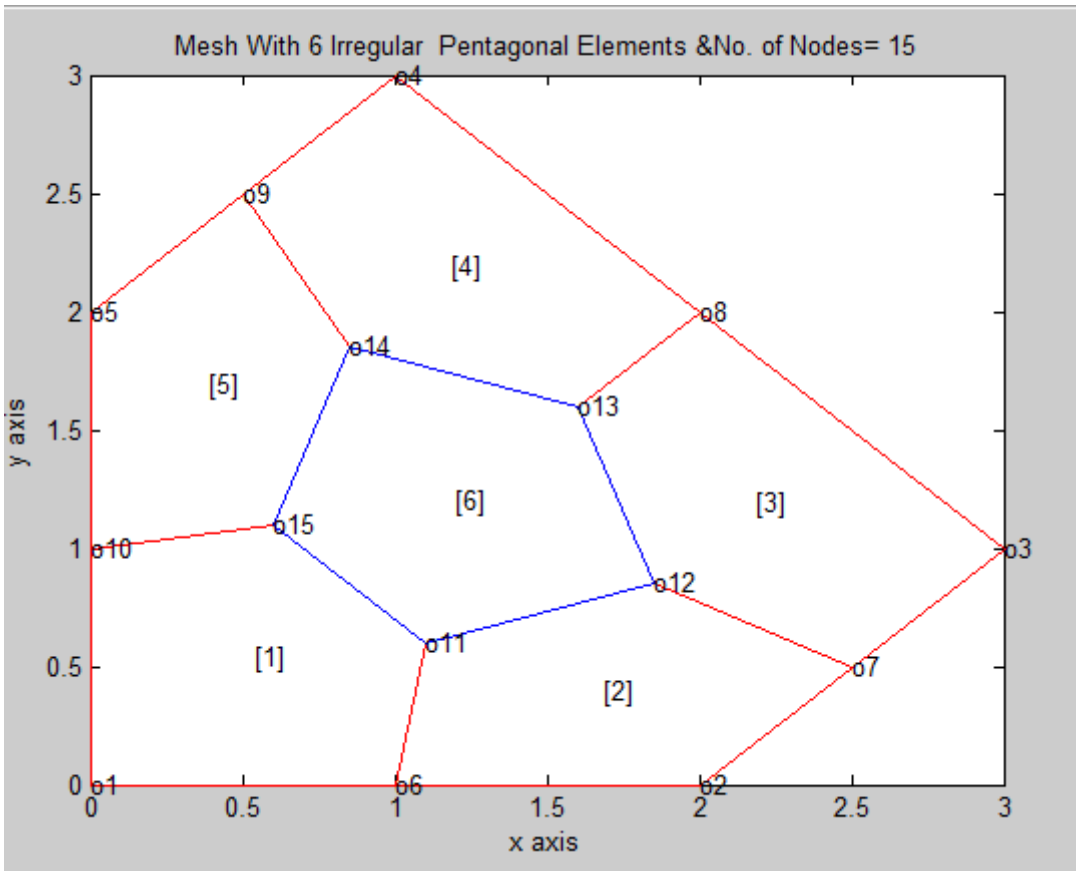


Fig.10a Pentagonal Division of a Unsymmetrical Pentagon-First Refinement

DOI: 10.18535/ijecs/v4i10.29

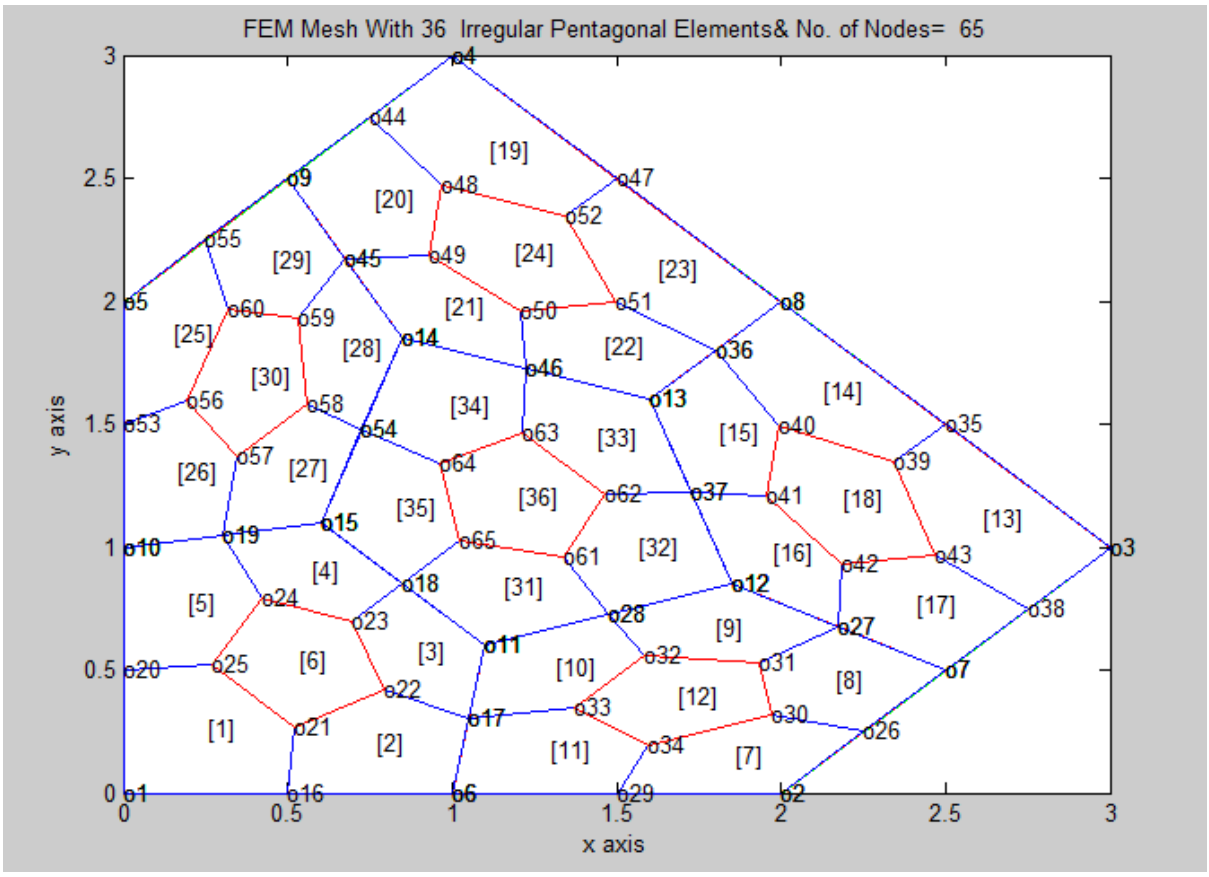


Fig.10b Pentagonal Division of a Unsymmetrical Pentagon-Second Refinement

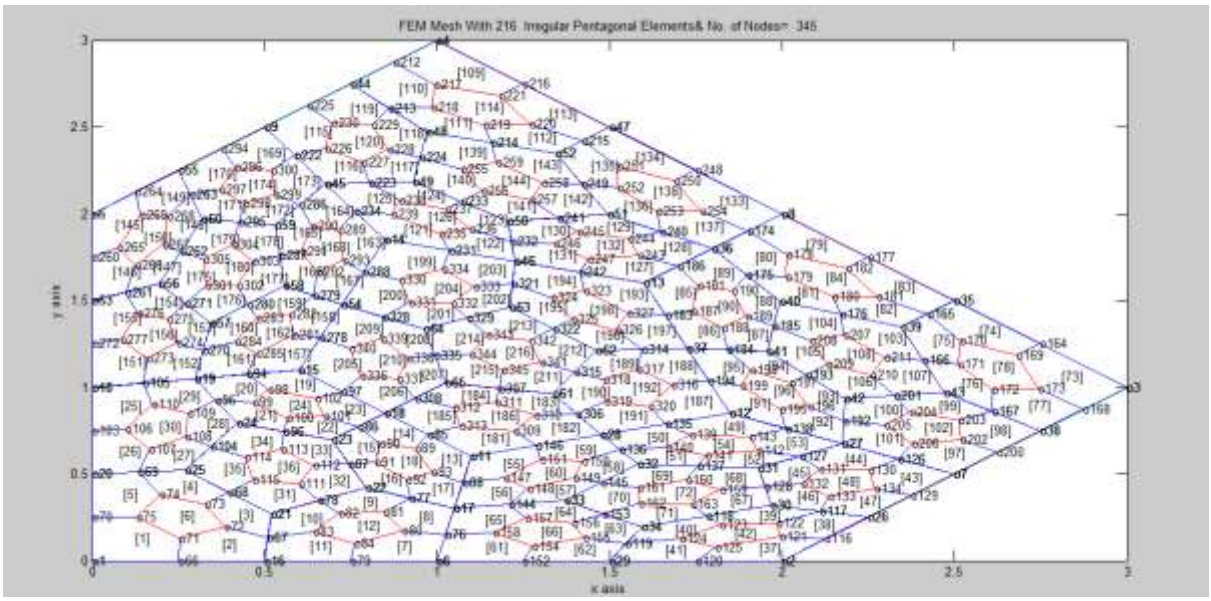
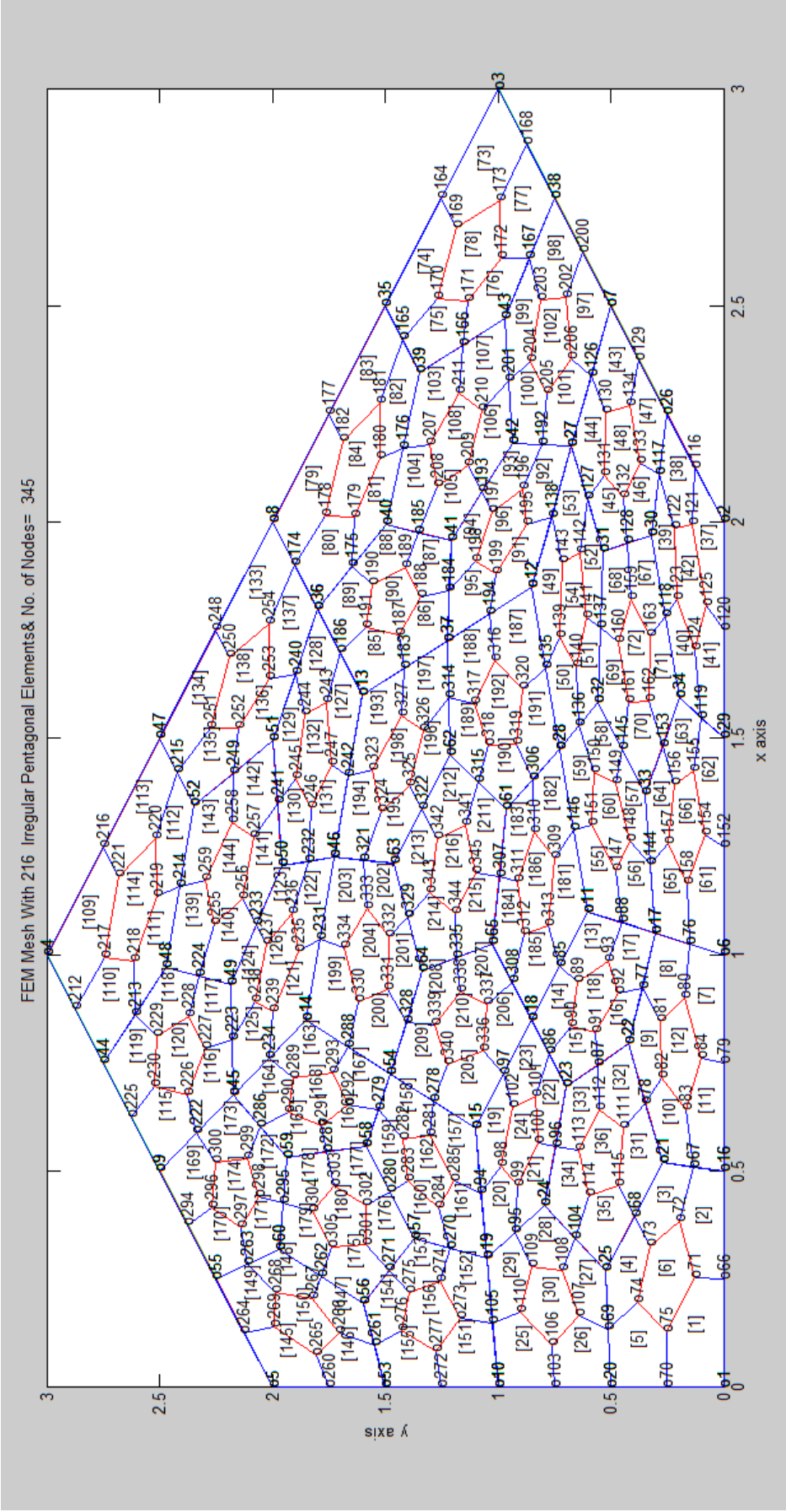


Fig.10c Pentagonal Division of a Unsymmetrical Pentagon-Third Refinement

DOI: 10.18535/ijecs/v4i10.29



DOI: 10.18535/ijecs/v4i10.29

Fig.10c Pentagonal Division of a Unsymmetrical Pentagon-Third Refinement(Rotated View)

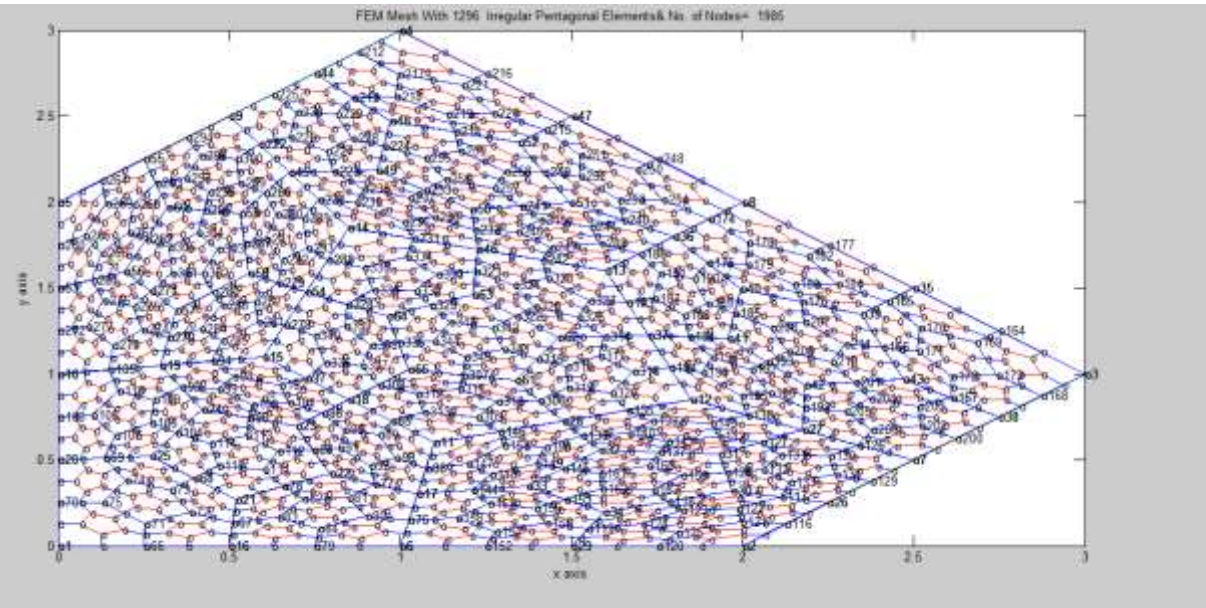


Fig.10d Pentagonal Division of a Unsymmetrical Pentagon-Fourth Refinement

Next we compute the exact value of the following integral

$$\iint_P f(x,y)dxdy = \int_d^e \int_{x=m_1y-c_1}^{x=-m_1y+c_1} f(x,y)dxdy + \int_{-b}^{d} \int_{x=-m_0y-c_0}^{x=m_0\ y+c_0} f(x,y)dxdy \quad \text{.....(19)}$$

Where P is the symmetrical pentagon $depicted\ in\ Fig.11$ and $f(x,y) = (px+qy)^m$ in $which$ $p,q,m\ are$ $real$ and arbitrary

DOI: 10.18535/ijecs/v4i10.29

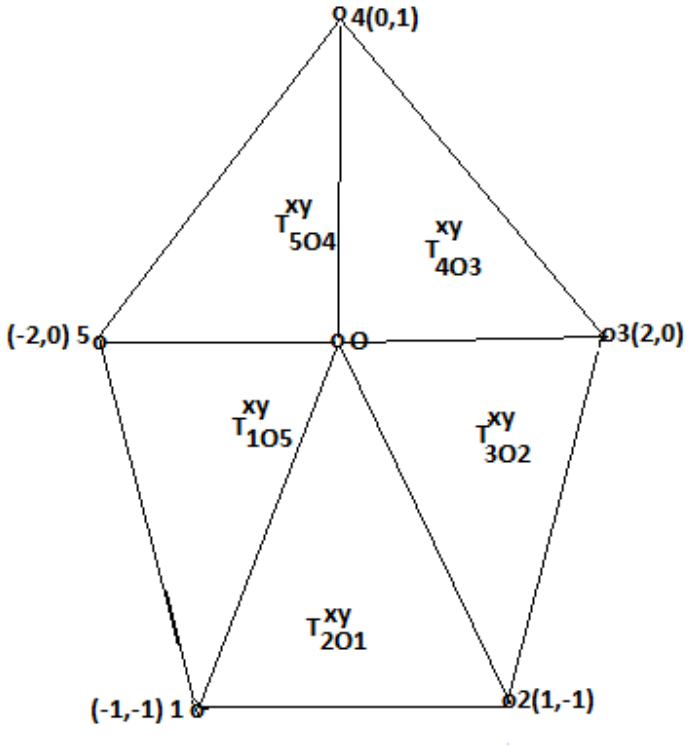


Fig.11 Division of a Pentagon into five triangles, T^{xy}_{ijk}, i=2,3,4,5,1;j=O;k=1,2,3,4,5

The exact value of the above integral can be computed by using the Fubini Type formula given in eqn(19) ,instead of this we show here the direct application of Theorem 2 given in this paper to obtain the exact value of the integral over the pentagon P of Fig.11

Using formulas og eqn(2) and eqn(6),wecan write

$$\int_P (px+qy)^m \, dxdy$$

$$=\int_{T^{xy}_{2o1}} (px+qy)^m \text{dxdy}+\int_{T^{xy}_{3o2}} (px+qy)^m \, dxdy+\int_{T^{xy}_{4o3}} (px+qy)^m \, dxdy+\int_{T^{xy}_{5o4}} (px+qy)^m \, dxdy$$

$$+\int_{T^{xy}_{1o5}} (px+qy)^m \text{dxdy} \qquad \text{.....(24)}$$

On any triangle T^{xy}_{iok} ,we have from Theorem 2

$(px+qy)^m = r^m[(px_i+qy_i)+s(x_{ki}\,p+y_{ki}q)]^m$

and hence from eqn(),we obtain

$$\int_{T^{xy}_{iok}} (px+qy)^m \text{dxdy}=(2\Delta_{iok}{}^{xy})\int_0^1\int_0^1 r^{m+1}\left[\left(px_i+qy_i\right)+s\left(x_{ki}\,p+y_{ki}q\right)\right]^m \text{drds}$$

$$=\frac{=(2\Delta_{iok}{}^{xy})}{(m+2)}\int_0^1\left[\left(px_i+qy_i\right)+s\left(x_{ki}\,p+y_{ki}q\right)\right]^m ds \quad \text{.....(25)}$$

Using the above results , we thus obtain

$$\int_P (px+qy)^m \, dxdy$$

$$=\frac{2}{(m+2)}\int_0^1\{[(p-q)-2ps]^m+[2p-(p+q)s]^m+[q+(2p-q)s]^m+[-(p+q)-(p-q)s]^m\}ds$$

.....(26)

DOI: 10.18535/ijecs/v4i10.29

The above equn(26) suggests that one can straight way proceed to the application Gauss Legendre quadrature which will not put any conditions on p,q. Thus numerical integration is simple and efficient. But we have to impose several conditions on p,q to obtain the exact integration formulas for the evaluation of the above integral expression in eqn(19) and eqn(24).The explicit formulas are now listed below.

Case(i): When none of the quantities $p, (p+q), (2p-q), (2p+q), (p-q)$ are zero, the exact value of the integral is given as

$$\int_P (px+qy)^m \, dxdy$$

$$=\frac{2}{(m+1)(m+2)}\left[\frac{\{(-1)^m(p+q)^{m+1}+(p-q)^{m+1}\}}{2p}+\frac{\{-(p-q)^{m+1}+(2p)^{m+1}\}}{p+q}+\frac{\{(2p)^{m+1}-q^{m+1}\}}{2p-q}+\frac{\{q^{m+1}+(-1)^m(2p)^{m+1}\}}{2p+q}+\frac{(-1)^m\{(2p)^{m+1}-(p+q)^{m+1}\}}{p-q}\right] \quad \text{.....(27)}$$

Case(ii) Substituting $p=0, on\ either\ side\ of\ eqn(\ ), we\ obtain$

$$\int_P y^m \, dxdy = \frac{2}{(m+2)}(-1)^m + \frac{4}{(m+1)(m+2)}[1+(-1)^m] \quad \text{..... (28)}$$

Case(iii) $When, p+q=0, substituting\ \ p=-q\ \ on\ either\ s\ ide\ of\ eqn(\ ), we\ obtain$

$$\int_P (x-y)^m \, dxdy$$

$$=\frac{2}{(m+2)}2^m+\frac{2(-1)^m}{(m+1)(m+2)}\left[2^m(1+(-1)^m)-\frac{2}{3}+(-1)^m\frac{2^{m+1}}{3}+2^{m+1}\right] \quad \text{.....(29)}$$

Case(iv) When $2p-q=0$,substituting$q=2p\ on\ either\ side\ of\ \ equation\ (\ ), we\ obtain$

$$\int_P (x+2y)^m \, dxdy$$

$$=\frac{2}{(m+1)(m+2)}\left\{\frac{(-1)^m}{2}[3^{m+1}-1]+\frac{1}{3}[2^m-(-1)^m]+2^{(m-1)}[1-(-1)^{m+1}]+[(-2)^{m+1}-(-3)^{m+1}]\right\}+\frac{2}{(m+2)}2^m$$

.....(30)

Case(v) When$2p+q=0$,substituting$q=-2p\ on\ either\ side\ of\ \ equation\ (\ ), we\ obtain$

$$\int_P (x-2y)^m \, dxdy$$

$$=\frac{2}{(m+1)(m+2)}\left\{\frac{(3^{m+1}-1)}{2}+(3^{m+1}-2^{m+1})+2^{m-1}(1+(-1)^m)+\frac{1}{3}(1-(-2)^{m+1})\right\}+\frac{2}{(m+2)}(-2)^m$$

.....(31)

Case(vi) $When, p-q=0, substituting\ \ p=q\ \ on\ either\ s\ ide\ of\ eqn(\ ), we\ obtain$

$$\int_P (x+y)^m \, dxdy$$

$$=\frac{2}{(m+1)(m+2)}\left\{2^m[1+(-1)^m]+[2^{m+1}-1]+\frac{[1-(-2)^{m+1}]}{3}\right\}+\frac{2}{(m+2)}(-2)^m \quad \text{.....(32)}$$

7.0 NUMERICAL EXAMPLES

We now choose some typical integrands $f(x,y)$ to demonstrate the computational scheme which uses the all pentagonal mesh generation and the boundary integration method.When $f(x,y)$ *is a simple integrand this method can also be applied to obtain exact values of the integal* as demonstrated in previous section 6 of this paper**.** When integrand is complicated Gaussian Quadrature rules such as **Gauss Legendre, Gauss Jacobi, Gauss Lobatto, Gauss Chebeshev etc** can be easily adopted.We have adopted **Gauss Legendre Rules** for the examples presented in this paper because it is more widely used.Exact values of integrals when $f(x,y)$ is a polynomial function is demonstrated for a symmetrical pentagon and unsymmetrical pentagon as given in the previous section.They are selected as typical test integrals.We have also noted some typical integrals tested in recent works[42,43]. We have summarised our findings in Tables 1 to 11 ,where in the results are displayed for the four Pentagonal element refinements along with the integrals,their domains and the exact values.

Table-1

NUMERICAL VALUES FOR THE INTEGRAL

$$\int\int_P \frac{(x^4+y^3)}{(1+x^2)}dxdy$$

Where P=pentagon

OGLR=Order of Gauss Legendre Rule;NPMR=Number of Pentagons used in Mesh Refinement

P=SP, SYMMETRICAL PENTAGON , SPANNED BY VERTICES <1(-1,-1),2(1,-1),3(2,0),4(0,1),5(-2,0)>

[I] **EXACT VALUE OF INTEGRAL=1.9240305426326500537470565177 5**

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
1	1.826052715428815e+000	1.907987585902751e+000	1.921263256819497e+000	1.923545370574599e+000
2	1.924378632830950e+000	1.924026589969111e+000	1.924030154135693e+000	1.924030525047537e+000
3	1.924019990232980e+000	1.924030496413962e+000	1.924030543202332e+000	1.924030542639478e+000
4	1.924030040438066e+000	1.924030544498806e+000	1.924030542631851e+000	1.924030542632648e+000
5	1.924030615634664e+000	1.924030542598669e+000	1.924030542632650e+000	1.924030542632652e+000
6	1.924030538232491e+000	1.924030542633047e+000	1.924030542632649e+000	1.924030542632651e+000
7	1.924030542813883e+000	1.924030542632648e+000	1.924030542632649e+000	1.924030542632651e+000
8	1.924030542627057e+000	1.924030542632650e+000	1.924030542632650e+000	1.924030542632652e+000
9	1.924030542632803e+000	1.924030542632650e+000	1.924030542632649e+000	1.924030542632651e+000
10	1.924030542632643e+000	1.924030542632650e+000	1.924030542632649e+000	1.924030542632651e+000

[II]

P=USP,UNSYMMETRICAL PENTAGON,SPANNED BY VERTICES {1(0,0),2(2,0),3(3,1),4(1,3),5((0,2)}

EXACT VALUE OF INTEGRAL= 20.482563614686924324814754072577

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
1	2.008018530492386e+001	2.041150706961537e+001	2.046999790811742e+001	2.048033643043846e+001
2	2.048066894719810e+001	2.048251625034292e+001	2.048256248750700e+001	2.048256358646244e+001
3	2.048257124319406e+001	2.048256385922152e+001	2.048256361692538e+001	2.048256361470399e+001
4	2.048256443505482e+001	2.048256361607802e+001	2.048256361468829e+001	2.048256361468691e+001
5	2.048256361769028e+001	2.048256361467609e+001	2.048256361468692e+001	2.048256361468692e+001
6	2.048256361411323e+001	2.048256361468701e+001	2.048256361468692e+001	2.048256361468691e+001
7	2.048256361470748e+001	2.048256361468692e+001	2.048256361468692e+001	2.048256361468692e+001
8	2.048256361468646e+001	2.048256361468692e+001	2.048256361468692e+001	2.048256361468692e+001
9	2.048256361468691e+001	2.048256361468692e+001	2.048256361468692e+001	2.048256361468691e+001
10	2.048256361468692e+001	2.048256361468692e+001	2.048256361468692e+001	2.048256361468691e+001

Table-2

NUMERICAL VALUES FOR THE INTEGRAL

DOI: 10.18535/ijecs/v4i10.29

$$\int \int_{P} (1-x)\sin(10xy)\, \boldsymbol{dxdy}$$

Where P= pentagon

OGLR=Order of Gauss Legendre Rule;NPMR=Number of Pentagons used in Mesh Refinement

P=SP, SYMMETRICAL PENTAGON , SPANNED BY VERTICES {1(-1,-1),2(1,-1),3(2,0),4(0,1),5(-2,0)}

[I] EXACT VALUE OF INTEGRAL= -0.013103719669957

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
1	-6.716976464703817e-001	9.184031604782990e-003	-1.082196017681281e-002	-1.271967264137169e-002
2	1.201575404545178e-001	-1.593479145306170e-002	-1.313401359413872e-002	-1.310445596537816e-002
3	-2.877801360407369e-002	-1.294348766465626e-002	-1.310354565605636e-002	-1.310372052884517e-002
4	-1.141818425287232e-002	-1.310872183608549e-002	-1.310372318795237e-002	-1.310371966954461e-002
5	-1.323027324598598e-002	-1.310366489213734e-002	-1.310371962874142e-002	-1.310371966994762e-002
6	-1.310043004387755e-002	-1.310371871870112e-002	-1.310371967012742e-002	-1.310371966995748e-002
7	-1.310355662854792e-002	-1.310371970068666e-002	-1.310371966995725e-002	-1.310371966995724e-002
8	-1.310372868709502e-002	-1.310371966980632e-002	-1.310371966995699e-002	-1.310371966995721e-002
9	-1.310371993158009e-002	-1.310371966995343e-002	-1.310371966995705e-002	-1.310371966995727e-002
10	-1.310371964498538e-002	-1.310371966995745e-002	-1.310371966995699e-002	-1.310371966995737e-002

[II]

P=USP,UNSYMMETRICAL PENTAGON,SPANNED BY VERTICES {1(0,0),2(2,0),3(3,1),4(1,3),5((0,2)}

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
1	4.914920374262107e-001	1.370572225696587e-001	2.393305295183613e-001	2.347246652259218e-001
2	-1.547821786391834e-001	2.484570553817125e-001	2.330252603696399e-001	2.330029998683565e-001
3	2.795800687728918e-001	2.330422610901967e-001	2.330070061172488e-001	2.330294758805691e-001
4	2.721309325859074e-001	2.327752473746906e-001	2.330301814492648e-001	2.330293241962427e-001
5	2.156324032062001e-001	2.330553932397373e-001	2.330293125747657e-001	2.330293242691677e-001
6	2.359134446275979e-001	2.330282045628970e-001	2.330293242979270e-001	2.330293242715241e-001
7	2.328077982547262e-001	2.330293433635078e-001	2.330293242724600e-001	2.330293242715162e-001
8	2.330348818736523e-001	2.330293243420203e-001	2.330293242715077e-001	2.330293242715164e-001
9	2.330295625354210e-001	2.330293242710551e-001	2.330293242715160e-001	2.330293242715164e-001
10	2.330292947832513e-001	2.330293242710863e-001	2.330293242715159e-001	2.330293242715164e-001

Table-3

NUMERICAL VALUES FOR THE INTEGRAL

$$\int \int_{P} (0{,}2x + 0.3y)^{19}\, \boldsymbol{dxdy}$$

Where P=pentagon

OGLR=Order of Gauss Legendre Rule;NPMR=Number of Pentagons used in Mesh Refinement

P=SP, SYMMETRICAL PENTAGON , SPANNED BY VERTICES {1(-1,-1),2(1,-1),3(2,0),4(0,1),5(-2,0)}

[I] **EXACT VALUE OF INTEGRAL**= -0.0000000556906789198571428571428571428557

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
1	-3.247407437070757e-008	-4.873897934031697e-008	-5.419997586673978e-008	-5.541514597384191e-008
2	-5.213643423241444e-008	-5.541893748890154e-008	-5.567830283127593e-008	-5.569022617441491e-008
3	-5.542527132029883e-008	-5.568552436275478e-008	-5.569062114917165e-008	-5.569067843431846e-008
4	-5.567890472402252e-008	-5.569060420557144e-008	-5.569067869434349e-008	-5.569067891939248e-008
5	-5.569016997954512e-008	-5.569067794096829e-008	-5.569067891917016e-008	-5.569067891985687e-008
6	-5.569065771408572e-008	-5.569067891212251e-008	-5.569067891985601e-008	-5.569067891985717e-008
7	-5.569067842603220e-008	-5.569067891983030e-008	-5.569067891985718e-008	-5.569067891985722e-008
8	-5.569067891565097e-008	-5.569067891985717e-008	-5.569067891985718e-008	-5.569067891985720e-008

9	-5.569067891985232e-008	-5.569067891985719e-008	-5.569067891985719e-008	-5.569067891985720e-008
10	-5.569067891985718e-008	-5.569067891985717e-008	-5.569067891985716e-008	-5.569067891985718e-008

~~~~~~ **[II]**

P=USP,UNSYMMETRICAL PENTAGON,SPANNED BY VERTICES {1(0,0),2(2,0),3(3,1),4(1,3),5((0,2)}

EXACT VALUE OF INTEGRAL= 0.69436450455892343345238095238095

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
1	5.536331888606911e-001	6.635652051383600e-001	6.887425376626309e-001	6.933806223804730e-001
2	6.877568652191953e-001	6.940474450322103e-001	6.943527230903406e-001	6.943641000561289e-001
3	6.942331152371809e-001	6.943631067001767e-001	6.943644930868269e-001	6.943645044728243e-001
4	6.943632947045650e-001	6.943645016603108e-001	6.943645045533783e-001	6.943645045589146e-001
5	6.943644991379966e-001	6.943645045559432e-001	6.943645045589226e-001	6.943645045589252e-001
6	6.943645045465742e-001	6.943645045589210e-001	6.943645045589230e-001	6.943645045589240e-001
7	6.943645045589088e-001	6.943645045589236e-001	6.943645045589235e-001	6.943645045589251e-001
8	6.943645045589234e-001	6.943645045589236e-001	6.943645045589235e-001	6.943645045589250e-001
9	6.943645045589235e-001	6.943645045589234e-001	6.943645045589234e-001	6.943645045589248e-001
10	6.943645045589236e-001	6.943645045589230e-001	6.943645045589232e-001	6.943645045589246e-001

Table-4

NUMERICAL VALUES FOR THE INTEGRAL

$$\int\int_P (0,17x + 0.25y)^{25}\, dxdy$$

Where P=pentagon

OGLR=Order of Gauss Legendre Rule;NPMR=Number of Pentagons used in Mesh Refinement

P=SP, SYMMETRICAL PENTAGON , SPANNED BY VERTICES {1(-1,-1),2(1,-1),3(2,0),4(0,1),5(-2,0)}

[I] **EXACT VALUE OF INTEGRAL=-0.0000000000069990091267998353953694168811778**

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
1	-3.172138336344716e-012	-5.686719074858495e-012	-6.686311072436234e-012	-6.939384076330868e-012
2	-6.148491023561045e-012	-6.911765402105541e-012	-6.994402031261994e-012	-6.998833417910701e-012
3	-6.894156730007480e-012	-6.996232312580676e-012	-6.998974191896908e-012	-6.999008818621111e-012
4	-6.991670008481077e-012	-6.998956251031735e-012	-6.999008938999576e-012	-6.999009126369590e-012
5	-6.998664979913602e-012	-6.999008214439382e-012	-6.999009125822582e-012	-6.999009126799267e-012
6	-6.998992995861380e-012	-6.999009110835939e-012	-6.999009126795658e-012	-6.999009126799841e-012
7	-6.999008254005350e-012	-6.999009126598904e-012	-6.999009126799830e-012	-6.999009126799848e-012
8	-6.999009090268436e-012	-6.999009126798383e-012	-6.999009126799843e-012	-6.999009126799846e-012
9	-6.999009125902003e-012	-6.999009126799838e-012	-6.999009126799842e-012	-6.999009126799846e-012
10	-6.999009126788874e-012	-6.999009126799842e-012	-6.999009126799838e-012	-6.999009126799844e-012

~~~~~~ **[II]**

P=USP,UNSYMMETRICAL PENTAGON,SPANNED BY VERTICES {1(0,0),2(2,0),3(3,1),4(1,3),5((0,2)}

EXACT VALUE OF INTEGRAL= 0.0088740194927385799673192916212099

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
1	6.177840394968492e-003	8.206337758125843e-003	8.747360243525626e-003	8.851787167926313e-003
2	8.656358137589306e-003	8.861336027046940e-003	8.873518313085025e-003	8.874002143505429e-003
3	8.865760790468582e-003	8.873908643046652e-003	8.874018510692692e-003	8.874019485234939e-003
4	8.873859540907999e-003	8.874019004349511e-003	8.874019491725432e-003	8.874019492736758e-003
5	8.874017835458105e-003	8.874019491587636e-003	8.874019492738016e-003	8.874019492738612e-003
6	8.874019483198976e-003	8.874019492737060e-003	8.874019492738578e-003	8.874019492738597e-003
7	8.874019492707407e-003	8.874019492738585e-003	8.874019492738588e-003	8.874019492738612e-003

8	8.874019492738526e-003	8.874019492738588e-003	8.874019492738586e-003	8.874019492738611e-003
9	8.874019492738588e-003	8.874019492738585e-003	8.874019492738586e-003	8.874019492738609e-003
10	8.874019492738581e-003	8.874019492738580e-003	8.874019492738581e-003	8.874019492738602e-003

==

Table-5

==

NUMERICAL VALUES FOR THE INTEGRAL

$$\int \int_P (x+y)^{19}/(10^{10})\, dxdy$$

Where P=pentagon

OGLR=Order of Gauss Legendre Rule;NPMR=Number of Pentagons used in Mesh Refinement

P=SP, SYMMETRICAL PENTAGON , SPANNED BY VERTICES {1(-1,-1),2(1,-1),3(2,0),4(0,1),5(-2,0)}

[I] **EXACT VALUE OF INTEGRAL= -0.000004660338095238095238095238095238 1-**

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
1	-3.152896039677961e-006	-4.118910591384930e-006	-4.511502146288937e-006	-4.629542913011446e-006
2	-4.337319454879283e-006	-4.617986759280833e-006	-4.657836737745992e-006	-4.660239898042306e-006
3	-4.611085420110539e-006	-4.658539877665016e-006	-4.660314032650732e-006	-4.660337886616668e-006
4	-4.655152417298368e-006	-4.660294605998913e-006	-4.660337966072463e-006	-4.660338094976856e-006
5	-4.660004324978146e-006	-4.660337549827477e-006	-4.660338094886030e-006	-4.660338095237922e-006
6	-4.660326632860938e-006	-4.660338091936372e-006	-4.660338095237632e-006	-4.660338095238087e-006
7	-4.660337910296154e-006	-4.660338095229199e-006	-4.660338095238100e-006	-4.660338095238092e-006
8	-4.660338094095138e-006	-4.660338095238087e-006	-4.660338095238100e-006	-4.660338095238092e-006
9	-4.660338095236724e-006	-4.660338095238098e-006	-4.660338095238100e-006	-4.660338095238092e-006
10	-4.660338095238094e-006	-4.660338095238093e-006	-4.660338095238098e-006	-4.660338095238087e-006

~~~~~~ **[II]**

P=USP,UNSYMMETRICAL PENTAGON,SPANNED BY VERTICES {1(0,0),2(2,0),3(3,1),4(1,3),5((0,2)}

EXACT VALUE OF INTEGRAL= 10.9951157784380952380952380952380952 38

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
1	9.535010164240415e+000	1.062003864043387e+001	1.092063769691975e+001	1.098153211145822e+001
2	1.090905677107142e+001	1.099066387244695e+001	1.099494876495080e+001	1.099511009318607e+001
3	1.099316009387640e+001	1.099509426479678e+001	1.099511560354962e+001	1.099511577714817e+001
4	1.099509549344272e+001	1.099511572913958e+001	1.099511577834482e+001	1.099511577843790e+001
5	1.099511567673852e+001	1.099511577838219e+001	1.099511577843808e+001	1.099511577843808e+001
6	1.099511577818981e+001	1.099511577843806e+001	1.099511577843809e+001	1.099511577843807e+001
7	1.099511577843782e+001	1.099511577843810e+001	1.099511577843810e+001	1.099511577843808e+001
8	1.099511577843809e+001	1.099511577843811e+001	1.099511577843810e+001	1.099511577843808e+001
9	1.099511577843810e+001	1.099511577843810e+001	1.099511577843810e+001	1.099511577843808e+001
10	1.099511577843810e+001	1.099511577843810e+001	1.099511577843810e+001	1.099511577843807e+001

==

Table-6

==

NUMERICAL VALUES FOR THE INTEGRAL

$$\int \int_P (x-y)^{20}/(10^5)\, dxdy$$
~~~~~~

DOI: 10.18535/ijecs/v4i10.29

Where P=pentagon

OGLR=Order of Gauss Legendre Rule;NPMR=Number of Pentagons used in Mesh Refinement

P=SP, SYMMETRICAL PENTAGON , SPANNED BY VERTICES {1(-1,-1),2(1,-1),3(2,0),4(0,1),5(-2,0)}

[I] EXACT VALUE OF INTEGRAL= 1.16508441558441558441558441558441558844

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
1	6.403884248404116e-001	9.723152170709110e-001	1.115926413699784e+000	1.155395118584921e+000
2	1.041017981251689e+000	1.150466025636280e+000	1.164248425255834e+000	1.165051813330887e+000
3	1.148082239981732e+000	1.164504765421631e+000	1.165076579928031e+000	1.165084346855592e+000
4	1.163478171193124e+000	1.165070485510601e+000	1.165084372207653e+000	1.165084415494875e+000
5	1.164981130547152e+000	1.165084223916640e+000	1.165084415452191e+000	1.165084415584353e+000
6	1.165080403553624e+000	1.165084414195942e+000	1.165084415584205e+000	1.165084415584416e+000
7	1.165084333549183e+000	1.165084415579519e+000	1.165084415584416e+000	1.165084415584417e+000
8	1.165084414832602e+000	1.165084415584408e+000	1.165084415584415e+000	1.165084415584417e+000
9	1.165084415581997e+000	1.165084415584416e+000	1.165084415584415e+000	1.165084415584417e+000
10	1.165084415584414e+000	1.165084415584416e+000	1.165084415584415e+000	1.165084415584416e+000

[II]

P=USP,UNSYMMETRICAL PENTAGON,SPANNED BY VERTICES {1(0,0),2(2,0),3(3,1),4(1,3),5((0,2)}

EXACT VALUE OF INTEGRAL= 2.08807341991341991341991341991341991341

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
1	1.080560588938075e+000	1.716717319580095e+000	1.982646411801784e+000	2.065618639933176e+000
2	1.822891224660700e+000	2.049103020656937e+000	2.085099504294714e+000	2.087940189107559e+000
3	2.040453929527864e+000	2.085269563287610e+000	2.088016205274084e+000	2.088072829607343e+000
4	2.080448700698647e+000	2.087930487144813e+000	2.088072755392358e+000	2.088073418333937e+000
5	2.087067085807480e+000	2.088069241058631e+000	2.088073415883397e+000	2.088073419911218e+000
6	2.087987528015845e+000	2.088073357189506e+000	2.088073419901264e+000	2.088073419913416e+000
7	2.088069382948703e+000	2.088073419464628e+000	2.088073419913401e+000	2.088073419913420e+000
8	2.088073330470947e+000	2.088073419912056e+000	2.088073419913417e+000	2.088073419913420e+000
9	2.088073419210281e+000	2.088073419913420e+000	2.088073419913418e+000	2.088073419913419e+000
10	2.088073419912595e+000	2.088073419913419e+000	2.088073419913418e+000	2.088073419913418e+000

Table-7

NUMERICAL VALUES FOR THE INTEGRAL

$$\int \int_{P} cos(30(x+y))\, dxdy$$

Where P=pentagon

OGLR=Order of Gauss Legendre Rule;NPMR=Number of Pentagons used in Mesh Refinement

P=SP, SYMMETRICAL PENTAGON , SPANNED BY VERTICES {1(-1,-1),2(1,-1),3(2,0),4(0,1),5(-2,0)}

[I] EXACT VALUE OF INTEGRAL= −0.00743617729902437321985844889949 64

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
5	1.487048233650115e-002	1.163444740424753e-003	-7.419427696575087e-003	-7.436186026933724e-003
10	-3.513867317140852e-002	-7.436844686825669e-003	-7.436177299548956e-003	-7.436177299024340e-003
15	-7.374551976469766e-003	-7.436177298818107e-003	-7.436177299024401e-003	-7.436177299024409e-003
20	-7.436178716955182e-003	-7.436177299024226e-003	-7.436177299024438e-003	-7.436177299024317e-003
25	-7.436177299022614e-003	-7.436177299024217e-003	-7.436177299024511e-003	-7.436177299024257e-003
30	-7.436177299024596e-003	-7.436177299024278e-003	-7.436177299024432e-003	-7.436177299024351e-003
35	-7.436177299024073e-003	-7.436177299024376e-003	-7.436177299024422e-003	-7.436177299024297e-003

DOI: 10.18535/ijecs/v4i10.29

40	-7.436177299024144e-003	-7.436177299024478e-003	-7.436177299024309e-003	-7.436177299024340e-003
45	-7.436177299024125e-003	-7.436177299024322e-003	-7.436177299024476e-003	-7.436177299024295e-003

[II]

P=USP,UNSYMMETRICAL PENTAGON,SPANNED BY VERTICES {1(0,0),2(2,0),3(3,1),4(1,3),5((0,2)}

EXACT VALUE OF INTEGRAL=0.036538064524804112293611778879387

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
5	-2.978460980082388e-001	3.607635606747262e-002	3.653743681343850e-002	3.653806431485410e-002
10	3.820873524496943e-002	3.653806509012749e-002	3.653806452480418e-002	3.653806452480405e-002
15	3.653805414435617e-002	3.653806452480372e-002	3.653806452480376e-002	3.653806452480427e-002
20	3.653806452480379e-002	3.653806452480439e-002	3.653806452480365e-002	3.653806452480424e-002
25	3.653806452480438e-002	3.653806452480417e-002	3.653806452480371e-002	3.653806452480426e-002
30	3.653806452480481e-002	3.653806452480422e-002	3.653806452480374e-002	3.653806452480436e-002
35	3.653806452480388e-002	3.653806452480389e-002	3.653806452480348e-002	3.653806452480405e-002
40	3.653806452480398e-002	3.653806452480406e-002	3.653806452480353e-002	3.653806452480412e-002
45	3.653806452480463e-002	3.653806452480420e-002	3.653806452480363e-002	3.653806452480431e-002

Table-8

NUMERICAL VALUES FOR THE INTEGRAL

$$\int \int_{P} \sqrt{((x-0.5)^2+(y-0.5)^2)}\, dxdy$$

Where P=pentagon

OGLR=Order of Gauss Legendre Rule;NPMR=Number of Pentagons used in Mesh Refinement

P=SP, SYMMETRICAL PENTAGON , SPANNED BY VERTICES {1(-1,-1),2(1,-1),3(2,0),4(0,1),5(-2,0)}

[I] EXACT VALUE OF INTEGRAL=5.809534594898993711537644374406 4

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
5	5.809632828241665e+000	5.809537500950220e+000	5.809534803290252e+000	5.809534590511865e+000
10	5.809536996640648e+000	5.809535680543106e+000	5.809534645275297e+000	5.809534598295638e+000
15	5.809529306809983e+000	5.809534972777953e+000	5.809534587924151e+000	5.809534596797954e+000
20	5.809536528167203e+000	5.809534673971211e+000	5.809534600720899e+000	5.809534595850399e+000
25	5.809533866423590e+000	5.809534554466219e+000	5.809534598424464e+000	5.809534595337206e+000
30	5.809534530846443e+000	5.809534531596558e+000	5.809534588569728e+000	5.809534595054050e+000
35	5.809534883309418e+000	5.809534563006950e+000	5.809534593540599e+000	5.809534594898946e+000
40	5.809534156130308e+000	5.809534593374709e+000	5.809534596095078e+000	5.809534594815993e+000
45	5.809534743227466e+000	5.809534606740333e+000	5.809534595365181e+000	5.809534594799386e+000

[II]

P=USP,UNSYMMETRICAL PENTAGON,SPANNED BY VERTICES {1(0,0),2(2,0),3(3,1),4(1,3),5((0,2)}

EXACT VALUE OF INTEGRAL=7.800618298021123954289758058444

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
5	7.800631021150374e+000	7.800621381477646e+000	7.800618066668203e+000	7.800618298661771e+000
10	7.800623558100750e+000	7.800618726971288e+000	7.800618396555366e+000	7.800618298096128e+000
15	7.800619530297125e+000	7.800617872510332e+000	7.800618278257638e+000	7.800618297973196e+000
20	7.800618080058866e+000	7.800618275919166e+000	7.800618292342051e+000	7.800618297975006e+000
25	7.800617871275819e+000	7.800618348045524e+000	7.800618305984281e+000	7.800618297971114e+000
30	7.800618069810641e+000	7.800618322119879e+000	7.800618295026687e+000	7.800618297992551e+000
35	7.800618236612546e+000	7.800618269499288e+000	7.800618296939613e+000	7.800618297994679e+000
40	7.800618327385326e+000	7.800618291744808e+000	7.800618300094140e+000	7.800618298012231e+000
45	7.800618358888679e+000	7.800618305993477e+000	7.800618297028252e+000	7.800618298017581e+000

DOI: 10.18535/ijecs/v4i10.29

Table-9

NUMERICAL VALUES FOR THE INTEGRAL

$$\int \int_P e^{-\left(\left(x-\frac{1}{2}\right)^2+\left(y-\frac{1}{2}\right)^2\right)} dxdy$$

Where P=pentagon

OGLR=Order of Gauss Legendre Rule;NPMR=Number of Pentagons used in Mesh Refinement

P=SP, SYMMETRICAL PENTAGON , SPANNED BY VERTICES {1(-1,-1),2(1,-1),3(2,0),4(0,1),5(-2,0)}

[I] EXACT VALUE OF INTEGRAL=1.729175291842216445410886742214 9

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
5	1.729175291727280e+000	1.729175291842019e+000	1.729175291842216e+000	1.729175291842218e+000
10	1.729175291842217e+000	1.729175291842216e+000	1.729175291842216e+000	1.729175291842217e+000
15	1.729175291842216e+000	1.729175291842216e+000	1.729175291842216e+000	1.729175291842218e+000
20	1.729175291842216e+000	1.729175291842216e+000	1.729175291842216e+000	1.729175291842217e+000
25	1.729175291842217e+000	1.729175291842217e+000	1.729175291842216e+000	1.729175291842218e+000
30	1.729175291842216e+000	1.729175291842216e+000	1.729175291842216e+000	1.729175291842217e+000
35	1.729175291842216e+000	1.729175291842216e+000	1.729175291842215e+000	1.729175291842216e+000
40	1.729175291842216e+000	1.729175291842215e+000	1.729175291842215e+000	1.729175291842217e+000
45	1.729175291842216e+000	1.729175291842216e+000	1.729175291842216e+000	1.729175291842217e+000

[II]

P=USP,UNSYMMETRICAL PENTAGON,SPANNED BY VERTICES {1(0,0),2(2,0),3(3,1),4(1,3),5((0,2)}

EXACT VALUE OF INTEGRAL=1.798096146581152670802198439191 6

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
5	1.798096147004040e+000	1.798096146581256e+000	1.798096146581153e+000	1.798096146581151e+000
10	1.798096146581153e+000	1.798096146581153e+000	1.798096146581154e+000	1.798096146581151e+000
15	1.798096146581153e+000	1.798096146581153e+000	1.798096146581153e+000	1.798096146581151e+000
20	1.798096146581154e+000	1.798096146581153e+000	1.798096146581154e+000	1.798096146581151e+000
25	1.798096146581153e+000	1.798096146581152e+000	1.798096146581153e+000	1.798096146581151e+000
30	1.798096146581152e+000	1.798096146581152e+000	1.798096146581153e+000	1.798096146581151e+000
35	1.798096146581152e+000	1.798096146581152e+000	1.798096146581153e+000	1.798096146581150e+000
40	1.798096146581150e+000	1.798096146581152e+000	1.798096146581152e+000	1.798096146581150e+000
45	1.798096146581150e+000	1.798096146581152e+000	1.798096146581152e+000	1.798096146581151e+000

Table-10

NUMERICAL VALUES FOR THE INTEGRAL

$$\int \int_P e^{-100\left(\left(x-\frac{1}{2}\right)^2+\left(y-\frac{1}{2}\right)^2\right)} dxdy$$

Where P=pentagon

OGLR=Order of Gauss Legendre Rule;NPMR=Number of Pentagons used in Mesh Refinement

P=SP, SYMMETRICAL PENTAGON , SPANNED BY VERTICES {1(-1,-1),2(1,-1),3(2,0),4(0,1),5(-2,0)}

[I] EXACT VALUE OF INTEGRAL=0.031391337254729663963679685981861

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
5	3.144575600692522e-002	3.139103643197601e-002	3.139133768940916e-002	3.139133725488259e-002
10	3.138436254302145e-002	3.139133748004325e-002	3.139133725472974e-002	3.139133725472965e-002
15	3.139136991267207e-002	3.139133725473267e-002	3.139133725472965e-002	3.139133725472965e-002
20	3.139133734899555e-002	3.139133725472967e-002	3.139133725472966e-002	3.139133725472965e-002
25	3.139133725436420e-002	3.139133725472965e-002	3.139133725472967e-002	3.139133725472965e-002
30	3.139133725472979e-002	3.139133725472967e-002	3.139133725472965e-002	3.139133725472965e-002

DOI: 10.18535/ijecs/v4i10.29

35	3.139133725472965e-002	3.139133725472966e-002	3.139133725472965e-002	3.139133725472964e-002
40	3.139133725472965e-002	3.139133725472966e-002	3.139133725472965e-002	3.139133725472963e-002
45	3.139133725472969e-002	3.139133725472961e-002	3.139133725472967e-002	3.139133725472965e-002

[II]

P=USP,UNSYMMETRICAL PENTAGON,SPANNED BY VERTICES {1(0,0),2(2,0),3(3,1),4(1,3),5((0,2)}

EXACT VALUE OF INTEGRAL=0.031415926535849631660672804517763

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
5	3.178890452349738e-002	3.141774210056893e-002	3.141592585616345e-002	3.141592653589832e-002
10	3.141581212974690e-002	3.141592652861738e-002	3.141592653584965e-002	3.141592653584959e-002
15	3.141592643874352e-002	3.141592653584964e-002	3.141592653584965e-002	3.141592653584958e-002
20	3.141592653584716e-002	3.141592653584965e-002	3.141592653584963e-002	3.141592653584958e-002
25	3.141592653584961e-002	3.141592653584962e-002	3.141592653584966e-002	3.141592653584960e-002
30	3.141592653584958e-002	3.141592653584960e-002	3.141592653584965e-002	3.141592653584959e-002
35	3.141592653584958e-002	3.141592653584960e-002	3.141592653584963e-002	3.141592653584958e-002
40	3.141592653584961e-002	3.141592653584956e-002	3.141592653584964e-002	3.141592653584956e-002
45	3.141592653584961e-002	3.141592653584966e-002	3.141592653584964e-002	3.141592653584958e-002

Table-11

NUMERICAL VALUES FOR THE INTEGRAL

$$\int \int_{P} (f_1(x,y) + f_2(x,y) + f_3(x,y) + f_4(x,y))\, dxdy$$

Where P=pentagon,

$f_1(x,y)$ =0.75*exp(-0.25*(9*x-2)^2-0.25*(9*y-2)^2)

$f_2(x,y)$ = 0.75*exp((-1/49)*(9*x+1)^2-0.1*(9*y+1))

$f_3(x,y)$ = 0.5*exp(-0.25*(9*x-7)^2-0.25*(9*y-3)^2)

$f_4(x,y)$ = -0.2*exp(-(9*y-4)^2-(9*y-7)^2)

OGLR=Order of Gauss Legendre Rule;NPMR=Number of Pentagons used in Mesh Refinement

P=SP, SYMMETRICAL PENTAGON , SPANNED BY VERTICES {1(-1,-1),2(1,-1),3(2,0),4(0,1),5(-2,0)}

[I] EXACT VALUE OF INTEGRAL=2.190018384318485748596900210234 5

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
5	2.189885492132464e+000	2.190018421988602e+000	2.190018384317350e+000	2.190018384318481e+000
10	2.190018374577564e+000	2.190018384318501e+000	2.190018384318486e+000	2.190018384318484e+000
15	2.190018384318295e+000	2.190018384318486e+000	2.190018384318486e+000	2.190018384318484e+000
20	2.190018384318486e+000	2.190018384318486e+000	2.190018384318486e+000	2.190018384318484e+000
25	2.190018384318487e+000	2.190018384318486e+000	2.190018384318486e+000	2.190018384318484e+000
30	2.190018384318485e+000	2.190018384318486e+000	2.190018384318486e+000	2.190018384318483e+000
35	2.190018384318486e+000	2.190018384318485e+000	2.190018384318486e+000	2.190018384318483e+000
40	2.190018384318486e+000	2.190018384318485e+000	2.190018384318485e+000	2.190018384318484e+000
45	2.190018384318487e+000	2.190018384318486e+000	2.190018384318486e+000	2.190018384318484e+000

[II]

P=USP,UNSYMMETRICAL PENTAGON,SPANNED BY VERTICES {1(0,0),2(2,0),3(3,1),4(1,3),5((0,2)}

EXACT VALUE OF INTEGRAL=0.558035127601503928491469366591 77

OGLR	NPMR=6	NPMR=6^2	NPMR=6^3	NPMR=6^4
5	5.581087222701432e-001	5.580348726179162e-001	5.580351261888881e-001	5.580351276023901e-001
10	5.580354432895984e-001	5.580351272766695e-001	5.580351276015048e-001	5.580351276015034e-001

15	5.580350696854708e-001	5.580351276014699e-001	5.580351276015034e-001	5.580351276015035e-001
20	5.580351272209716e-001	5.580351276015041e-001	5.580351276015034e-001	5.580351276015034e-001
25	5.580351276044324e-001	5.580351276015041e-001	5.580351276015033e-001	5.580351276015034e-001
30	5.580351276015080e-001	5.580351276015039e-001	5.580351276015033e-001	5.580351276015033e-001
35	5.580351276015040e-001	5.580351276015036e-001	5.580351276015033e-001	5.580351276015033e-001
40	5.580351276015041e-001	5.580351276015036e-001	5.580351276015032e-001	5.580351276015031e-001
45	5.580351276015032e-001	5.580351276015040e-001	5.580351276015035e-001	5.580351276015032e-001

==========

8.0 Computer Programs

In this section, we consider several examples to show that the present formulations may be applied to integrals which find applications in practical situations.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 pentagons for the complex domains such as those in [40-44]. We use the theory and procedure developed in sections 2, 3 and 4 for this purpose. The following MATLAB codes are written for this purpose.

(1) fnxy.m
(2) newpolygon_boundary_n_pentagonNsixgon.m
(3) nodaladdressesforpentagon.m
(4) pentagonalmeshgeneratorforfemiterations.m
(5) compositeboundaryintegrationconvexpolygonofpentagons_n.m
(6) `pentagonalmeshgeneratorforfemiterationstrial.m`
[7] `nodaladdressesformultiplepentagon.m`

These MATLAB programs [1] to [5] are appended to this paper

9.0 Conclusions

An automatic recursive pentagonal mesh generator technique is presented for the two dimensional convex polygonal domains. This mesh generation is made fully automatic and allows the user to define the problem domain with minimum amount of input such as coordinates of boundary. Once this input is created, by selecting an appropriate interior point of the convex polygonal domain, we form the pentagonal subdomains.It is shown in literature that one cannot refine a hexagon using hexagons of smaller size. In general, one can only refine an n-gon by n-gons of smaller size if $n \leq 5$. Furthermore, we introduce a refinement scheme of a general polygon based on the pentagon scheme. This paper first presents a pentagonalization (or pentagonal conversion) scheme that can create a pentagonal mesh from any arbitrary mesh structure. We also introduce a pentagonal preservation scheme that can create a pentagonal mesh from any pentagonal mesh. This paper then presents a new numerical integration technique proposed earlier by the first author and co-workers, known as boundary integration method [34-40] is now applied to arbitrary polygonal domains using pentagonal finite element mesh. Numerical results presented is tested on examples of complicated integrals over convex polygons in the context of pentagonal domains with composite numerical integration scheme of triangular finite elements which can be easily created by joining the centre point of pentagons,this shows that the proposed method yields accurate results even for low order Gauss Legendre Quadrature rules. Our numerical results suggest that the **refinement scheme for pentagons and polygons** may lead to higher accuracy than the **uniform refinement of triangulations and quadrangulations.**

We have also appended MATLAB programs which provide the nodal coordinates, element nodal connectivity and graphic display of the generated all pentagonal mesh for the pentagon and the complex polygonal domains. We believe that this work will be useful for various applications in science and engineering.

DOI: 10.18535/ijecs/v4i10.29

REFERENCES

[1] O.C. Zienkiewicz, Taylor R.L. Taylor and Zhu J.Z . Zhu ,Finite Element Method, its basis and fundamentals, Elservier, (2005)
[2] Bathe K.J. Bathe, Finite Element Procedures, Prentice Hall, Englewood Cliffs, N J (1996)
[3] J.N. Reddy, Finite Element Method, Third Edition, Tata Mc Graw-Hill (2005)
[4] R.L. Burden and J.D. Faires Numerical Analysis, 9th Edition, Brooks/Cole,Cengage Learning (2011)
[5]A.H. Stroud and D. Secrest ,Gaussian quadrature formulas, Prentice Hall,Englewood Cliffs NJ (1966)
[6] J. Stoer and R. Bulirsch, Introduction to Numerical Analysis, Springer-Verlag, New York (1980)
[7] T.J. Chung Finite Element Analysis in Fluid Dynamics, pp. 191-199, Mc Graw Hill, C.A , (1978)
[8] H.T. Rathod, Some analytical integration formulae for four node isoparametric element, Computer and structures 30(5), pp.1101-1109, (1988)
[9] D.K. Babu and G.F .Pinder , Analytical integration formulae for linear isoparametric finite elements, Int. J. Numer. Methods Eng 20, pp.1153-1166
[10] A. Mizukami, Some integration formulas for four node isoparametric element, Computer Methods in Applied Mechanics and Engineering. 59 pp. 111-121(1986)
[11] M. Okabe, Analytical integration formulas related to convex quadrilateral finite elements, Computer methods in Applied mechanics and Engineering. 29, pp.201-218 (1981)
[12] D.V. Griffiths ,Stiffness matrix of the four node quadrilateral element in closed form, International Journal for Numerical Methods in Engineering. 28, pp.687- 703(1996)
[13]H.T. Rathod 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] H.T. Rathod 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] G. Yagawa, G.W. Ye 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] H.T. Rathod 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] D. Hanselman and B. Littlefield, Mastering MATLAB 7 , Prentice Hall, Happer Saddle River, N J . (2005)
[18] B.H. Hunt, R.L. Lipsman and J.M. Rosenberg, A Guide to MATLAB for beginners and experienced users, Cambridge University Press (2005)
[19] B. Char, K. Geddes, G. Gonnet, B. Leong,M. Monagan and S. Watt, First Leaves; A tutorial Introduction to Maple V , New York : Springer–Verlag (1992)
[20] D. Eugene, Mathematica , Schaums Outlines Theory and Problems, Tata Mc Graw Hill (2001)
[21] H. Ruskeepaa, Mathematica Navigator, Academic Press (2009)
[22] S.P. Timoshenko and J.N. Goodier, Theory of Elasticity, 3rd Edition, Tata Mc Graw Hill Edition (2010)
[23] R.G. Budynas, Applied Strength and Applied Stress Analysis, Second Edition, Tata Mc Graw Hill Edition (2011)
[24] R.J. Roark, Formulas for stress and strain, Mc Graw Hill, New York (1965)
[25] S.H. Nguyen , An accurate finite element formulation for linear elastic torsion calculations, Computers and Structures. 42, pp.707-711 (1992)
[26] H.T,Rathod .Bharath . Rathod,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]. H.T.Rathod, Bharath Rathod, K.T.Shivaram, A.S.Hariprasad, K.V.Vijayakumar K.Sugantha Devi, A New Approach to an All Quadrilateral Mesh Generation Over Arbitrary Linear Polygonal Domains for Finite Element Analysis, International Journal of Engineering and Computer Science , vol.3,issue4(2014),pp 5224-5272

[28]. H.T. Rathod, Bharath Rathod, Shivaram K.T , H. Y. Shrivalli , Tara Rathod ,K. Sugantha Devi ,An explicit finite element integration scheme using automatic mesh generation technique for linear convex quadrilaterals over plane regions, International Journal of Engineering and Computer Science, vol.3,issue4(2014),pp5400-5435

[29].H.T.Rathod , Bharath Rathod , K.T.Shivaram , K. Sugantha Devi , Tara Rathod, An explicit finite element integration scheme for linear eight node convex quadrilaterals using automatic
mesh generation technique over plane regions, *International Journal of Engineering and Computer Science, vol.3,issue5(2014),pp5657-5713*

[30] H.T. Rathod , K.V.Vijayakumar , A. S. Hariprasad , K. Sugantha Devi , C.S.Nagabhushana A New Approach to Automatic Generation of All Quadrilateral Finite Element Mesh for Planar Multiply Connected Regions, International Journal Of Engineering And Computer Science ISSN:2319-7242 Volume 4 Issue 6 June 2015, Page No. 12792-12848

[31] M. Floater and M. J. Lai, Polygonal spine spaces and the numerical solution of the Poisson equation, preprint available on internet, (2015).
http://alpha.math.uga.edu/~mjlai/papers/LaiFloater2015.pdf

[32] M. J. Lai and G. Slavov, On Recursive Refinement of Convex Polygons, , preprint available on internet, (2015).
http://alpha.math.uga.edu/~mjlai/papers/polygondivision.pdf

[33]M.J.Lai and L.L.Schumaker,Spline functions on Triangulations,Cambridge University Press(2007)

[34] H.T.Rathod and H.S.Govinda Rao, Integration of polynomials over linear polyhedra in Euclidean three-dimensional space, Comput. Methods Appl.Mech.Engrg.126 (1993)373-392.

[35] H.T.Rathod and H.S.Govinda Rao, Integration of polynomials over arbitrary tetrahedron in Euclidean three-dimensional space, Comput.Struct.59 (1996), 35-65.

[36] H.T.Rathod and S.V. Hiremath, Boundary Integration of polynomials over an arbitrary linear tetrahedron in Euclidean three dimensional space, Comput. Methods Appl.Mech. Engrg. 153(1998) 81-106.

[37] H.T.Rathod and S.V. Hiremath, Boundary Integration of polynomials over an arbitrary linear hexahedron in Euclidean three dimensional space, Comput. Methods Appl.Mech. Engrg 161(1998)155-193.

[38] F.Bernardini, Integration of polynomials over n-dimensional polyhedra, Comput. Aided Des,(1991) 51-58.

[39] H.T.Rathod and H.S.Govinda Rao, Integration of polynomials over n-dimensional linear polyhedra,, Comput. Struct. 65(1997) 829-847.

[40].H.T.Rathod,ArunkumarGali,S.V.Hiremath,Numerical integration of arbitrary functions over linear polygon in the Cartesian 2-space ,International e-Journal of Numerical Analysis and Related Topics,Vol.6,March 2011,pp28-51

[41] Jesu's A De Loera,How to integrate a Polynomial over a Convex Polytope: Combinatorics and Algorithms ,seminar presentation,UC Davis , http://math.nist.gov/mcsd/Seminars/2012/2012-09-19-DeLoera-presentation.pdf

[42]Sarada, J. and K.V. Nagaraja, 2011. Generalized Gaussian quadrature rules over two-dimensional regions with linear sides. *Appl. Math. Comput.*, **217**, 5612–5621.

[43]M. Alamgir Hossain and Md. Shafiqul Islam, Generalized Composite Numerical Integration Rule Over a Polygon Using Gaussian Quadrature**,** Dhaka Univ. J. Sci. 62(1): 25-29, 2014 (January)

[44]Logah.Perumal,Integration Techniques for two dimensional domains, International Journal of Research in Engineering and Technology,Vol.3 Issue7,July 2014,pp487-494

APPENDIX:COMPUTER PROGRAMS

%********FIRST COMPUTER PROGRAM**********

```
function[fn]=fnxy(n,x,y)
switch n

  case 100%pentagon
        fn=(x^4+y^3)/(1+x^2);
  case 101
```

```
     fn=(1-x)*sin(10*x*y);
  case 102
     fn=(.2*x+.3*y)^19;
  case 103

    fn=(.17*x+.25*y)^25;

   case 104
        fn=(x+y)^19/10^10;
   case 105
        fn=(x-y)^20/10^5;
   case 106
    fn=cos(30*(x+y));
  case 107
       fn=sqrt((x-0.5)^2+(y-0.5)^2);
  case 108
       fn=exp(-((x-1/2)^2+(y-1/2)^2));

   case 109
       fn=exp(-100*((x-1/2)^2+(y-1/2)^2));
   case 110
      f1=0.75*exp(-0.25*(9*x-2)^2-0.25*(9*y-2)^2);
      f2=0.75*exp((-1/49)*(9*x+1)^2-0.1*(9*y+1));
      f3=0.5*exp(-0.25*(9*x-7)^2-0.25*(9*y-3)^2);
      f4=-0.2*exp(-(9*y-4)^2-(9*y-7)^2);
      fn=f1+f2+f3+f4;
   otherwise
    disp('something wrong')
end
```

%********SECOND COMPUTER PROGRAM**********

```
function[II]=newpolygon_boundary_n_pentagonNsixgon(m,n,dataset)
%integration over polygon with n sides
%using symbolic maths
%fn=integrand function listed,fn=1,2,3,4,5,6,7
%n=number of sides of polygon
%m=1,for n-expanding triangles with respect to origin
%m=2,boundary integration along n-sides
%boundaryintegrationconvexpolygon_n(5,10,100,[-1;1;2;0;-2;-1],[-1;-1;0;1;0;-1])
%newpolygon_boundary_n_pentagonNsixgon(m=1,n=5)
%newpolygon_boundary_n_pentagonNsixgon(m=2,n=5)
syms t x y u v
format long e
%
nn=n;
if nn==6%convex polygon is a sixgon
switch dataset
case 1
A=[0.1;0.7;1;0.75;0.5;0;0.1];
B=[0;0.2;0.5;0.85;1;0.25;0];
case 2

end
end
if nn==5%convex polygon is a pentagon
    switch dataset
    case 1
    A=[-1;1;2;0;-2;-1];
    B=[-1;-1;0;1;0;-1];
   case 2
    A=[0;2;3;1;0;0];
```

```
    B=[0;0;1;3;2;0];
    end
end
n1=length(A)-1;n2=length(B)-1;
if (n1~=n2)|(n~=n1)|(n~=n2)
    disp('mismatch of X,Y &number of sides')
    disp(n)
    disp(n1)
    disp(n2)
end

nn=n+1;
switch m
    case 1
    %THIS PROGRAM SEGMENT IS DELETED
   case 2
       disp('direct application of GREENS THEOREM')
       for fn=3:11
       switch fn
        case 1%no success
            f=(x^4+y^3)/(1+x^2)
           ff=int(f,x)
            X=A;Y=B;
        case 2
            f=(1-x)*sin(10*x*y)
            ff=int(f,x)
            X=A;Y=B;
        case 3
          f=(.2*x+.3*y)^19
           ff=int(f,x)
           X=A;Y=B;
        case 4
           f=((.17*x+.25*y)^25)
           ff=int(f,x)
           X=A;Y=B;
       case 5
         f=(x+y)^19/(10^10)
         ff=int(f,x)
         X=A;Y=B;
        case 6
        f=(x-y)^20/10^5
        ff=int(f,x)
         X=A;Y=B;
         %=========================
           case 7
         f=cos(30*(x+y))
         ff=int(f,x)
         X=A;Y=B;

      case 8
         f=sqrt((x-0.5)^2+(y-0.5)^2)
         ff=int(f,x)
         X=A;Y=B;
     case 9
         f=exp(-((x-0.5)^2+(y-0.5)^2))
         ff=int(f,x)
          X=A-0.5*ones(nn,1)
          Y=B-0.5*ones(nn,1)
          ff=int(exp(-(x^2+y^2)),x)
      case 10
         f=exp(-100*((x-0.5)^2+(y-0.5)^2))
         ff=int(f,x)
```

```
          X=A;Y=B;

      case 11
         f1=0.75*exp(-0.25*(9*x-2)^2-0.25*(9*y-2)^2)
         f2=0.75*exp((-1/49)*(9*x+1)^2-0.1*(9*y+1))
         f3=0.5*exp(-0.25*(9*x-7)^2-0.25*(9*y-3)^2)
         f4=-0.2*exp(-(9*y-4)^2-(9*y-7)^2)
         f=f1+f2+f3+f4
         ff=int(f,x)
         X=A;Y=B;


       end
         ii=0;
  for N=1:n
        xi=X(N+1);yi=Y(N+1);xk=X(N);yk=Y(N);
        xx=xk+(xi-xk)*t;yy=yk+(yi-yk)*t;
        d=(yi-yk);
        fff=subs(ff,{x,y},{xx,yy});
        i=d*int(fff,t,0,1);
        ii=ii+i;
  end%end for N-LOOP%case 2

  iii=vpa(ii)
  II(fn)=vpa(iii,16);
 end%end for fn


 case 3
      %THIS PROGRAM SEGMENT IS DELETED

end%end for switch m
```

%********THIRD COMPUTER PROGRAM**********

```
function[nvv,nuu,nww,elm]=nodaladdressesforpentagon(iter)
%
nvv(1:6,1)=[1;2;3;4;5;1]
maxnvv=max(max(nvv))
mdpt(1:maxnvv,1:maxnvv)=zeros(maxnvv,maxnvv)
for i=1:5
    maxnvv=maxnvv+1;
    mdpt(nvv(i,1),nvv(i+1,1))=maxnvv;
     mdpt(nvv(i+1,1),nvv(i,1))=maxnvv;
nuu(i,1)=maxnvv;
end
for i=1:5
    maxnvv=maxnvv+1;
nww(i,1)=maxnvv;
end
nvv;
nuu;
nww;

[p,q]=size(nvv)
nel=p*q
for j=1:q
    k=(j-1)*p+1;
    elm(1:5,k)=[nvv(1,j) nuu(1,j) nww(1,j) nww(5,j) nuu(5,j)];
```

```
    elm(1:5,k+1)=[nvv(2,j) nuu(2,j) nww(2,j) nww(1,j) nuu(1,j)];
    elm(1:5,k+2)=[nvv(3,j) nuu(3,j) nww(3,j) nww(2,j) nuu(2,j)];
    elm(1:5,k+3)=[nvv(4,j) nuu(4,j) nww(4,j) nww(3,j) nuu(3,j)];
    elm(1:5,k+4)=[nvv(5,j) nuu(5,j) nww(5,j) nww(4,j) nuu(4,j)];
    elm(1:5,k+5)=[nww(1,j) nww(2,j) nww(3,j) nww(4,j) nww(5,j)];
end
elm
[pp,qq]=size(elm)
if(iter==1)
    return
end
%iter=2
 for j=1:qq
   nvv(1:5,j)=elm(1:5,j);
   nvv(6,j)=elm(1,j);
 end
nvv
maxnvv=max(max(nvv))
mdpt(1:maxnvv,1:maxnvv)=zeros(maxnvv,maxnvv)
%compute mdpt matrix and nuu
for j=1:qq
for i=1:5

    if (mdpt(nvv(i,j),nvv(i+1,j))==0)
          maxnvv=maxnvv+1;
    mdpt(nvv(i,j),nvv(i+1,j))=maxnvv;
    mdpt(nvv(i+1,j),nvv(i,j))=maxnvv;
    nuu(i,j)=maxnvv;
    end

    if (mdpt(nvv(i,j),nvv(i+1,j))~=0)
    nuu(i,j)=mdpt(nvv(i,j),nvv(i+1,j));
    end

end

for i=1:5
    maxnvv=maxnvv+1;
   nww(i,j)=maxnvv;
end


end

mdpt
nuu

[p,q]=size(nvv)
 nel=p*q
for j=1:q
    k=(j-1)*p+1;
    elm(1:5,k)=[nvv(1,j) nuu(1,j) nww(1,j) nww(5,j) nuu(5,j)];
    elm(1:5,k+1)=[nvv(2,j) nuu(2,j) nww(2,j) nww(1,j) nuu(1,j)];
    elm(1:5,k+2)=[nvv(3,j) nuu(3,j) nww(3,j) nww(2,j) nuu(2,j)];
    elm(1:5,k+3)=[nvv(4,j) nuu(4,j) nww(4,j) nww(3,j) nuu(3,j)];
    elm(1:5,k+4)=[nvv(5,j) nuu(5,j) nww(5,j) nww(4,j) nuu(4,j)];
    elm(1:5,k+5)=[nww(1,j) nww(2,j) nww(3,j) nww(4,j) nww(5,j)];
end
elm
[pp,qq]=size(elm)
```

```
if(iter==2)
    return
end

%iter=3
 for j=1:qq
   nvv(1:5,j)=elm(1:5,j);
   nvv(6,j)=elm(1,j);
 end
nvv
maxnvv=max(max(nvv))
mdpt(1:maxnvv,1:maxnvv)=zeros(maxnvv,maxnvv)
%-----------------------
%compute mdpt matrix and nuu
for j=1:qq
for i=1:5

    if (mdpt(nvv(i,j),nvv(i+1,j))==0)
          maxnvv=maxnvv+1;
    mdpt(nvv(i,j),nvv(i+1,j))=maxnvv;
    mdpt(nvv(i+1,j),nvv(i,j))=maxnvv;
    nuu(i,j)=maxnvv;
    end

    if (mdpt(nvv(i,j),nvv(i+1,j))~=0)
    nuu(i,j)=mdpt(nvv(i,j),nvv(i+1,j));
    end

end

for i=1:5
    maxnvv=maxnvv+1;
   nww(i,j)=maxnvv;
end



end

mdpt
nuu

[p,q]=size(nvv)
 nel=p*q
for j=1:q
    k=(j-1)*p+1;
    elm(1:5,k)=[nvv(1,j) nuu(1,j) nww(1,j) nww(5,j) nuu(5,j)];
    elm(1:5,k+1)=[nvv(2,j) nuu(2,j) nww(2,j) nww(1,j) nuu(1,j)];
    elm(1:5,k+2)=[nvv(3,j) nuu(3,j) nww(3,j) nww(2,j) nuu(2,j)];
    elm(1:5,k+3)=[nvv(4,j) nuu(4,j) nww(4,j) nww(3,j) nuu(3,j)];
    elm(1:5,k+4)=[nvv(5,j) nuu(5,j) nww(5,j) nww(4,j) nuu(4,j)];
    elm(1:5,k+5)=[nww(1,j) nww(2,j) nww(3,j) nww(4,j) nww(5,j)];
end
elm
[pp,qq]=size(elm)

if(iter==3)
    nnode=max(max(elm))
    iter
```

```
    return
end


%----------------------------------------
%iter=4
 for j=1:qq
   nvv(1:5,j)=elm(1:5,j);
   nvv(6,j)=elm(1,j);
 end
nvv
maxnvv=max(max(nvv))
mdpt(1:maxnvv,1:maxnvv)=zeros(maxnvv,maxnvv)
%-----------------------
%compute mdpt matrix and nuu
for j=1:qq
for i=1:5

    if (mdpt(nvv(i,j),nvv(i+1,j))==0)
          maxnvv=maxnvv+1;
    mdpt(nvv(i,j),nvv(i+1,j))=maxnvv;
    mdpt(nvv(i+1,j),nvv(i,j))=maxnvv;
    nuu(i,j)=maxnvv;
    end

    if (mdpt(nvv(i,j),nvv(i+1,j))~=0)
    nuu(i,j)=mdpt(nvv(i,j),nvv(i+1,j));
    end

end

for i=1:5
    maxnvv=maxnvv+1;
   nww(i,j)=maxnvv;
end



end

mdpt
nuu

[p,q]=size(nvv)
 nel=p*q
for j=1:q
    k=(j-1)*p+1;
    elm(1:5,k)=[nvv(1,j) nuu(1,j) nww(1,j) nww(5,j) nuu(5,j)];
    elm(1:5,k+1)=[nvv(2,j) nuu(2,j) nww(2,j) nww(1,j) nuu(1,j)];
    elm(1:5,k+2)=[nvv(3,j) nuu(3,j) nww(3,j) nww(2,j) nuu(2,j)];
    elm(1:5,k+3)=[nvv(4,j) nuu(4,j) nww(4,j) nww(3,j) nuu(3,j)];
    elm(1:5,k+4)=[nvv(5,j) nuu(5,j) nww(5,j) nww(4,j) nuu(4,j)];
    elm(1:5,k+5)=[nww(1,j) nww(2,j) nww(3,j) nww(4,j) nww(5,j)];
end
elm
[pp,qq]=size(elm)

if(iter==4)
    nnode=max(max(elm))
    iter
    return
```

```
end
```

%********FOURTH COMPUTERPROGRAM**********

```
 function[ELM,xcord,ycord]=pentagonalmeshgeneratorforfemiterations(xv,yv,iteration)
%(xv,yv,iteration)
%pentagonalmeshelementnodalconnectivity
% coordinates of vertices for the original pentagon
%clear
%----------------------example-1----------
%xv(1,1)=0;yv(1,1)=1;
%xv(2,1)=0;yv(2,1)=0.1;
%xv(3,1)=0.45;yv(3,1)=-0.2;
%xv(4,1)=1.1;yv(4,1)=0;
%xv(5,1)=1.1;yv(5,1)=1.1;
%xv(6,1)=0;yv(6,1)=1;
%xv=[0;0.0; 0.45;1.1;1.1;0];
%yv=[1;0.1;-0.20;0.0;1.1;1];
%convert the column vectors (xv,yv) into row vectors (xvv,yvv)
%xvv(1,:)=xv(:,1)';yvv(1,:)=yv(:,1)';
% draw outline of the given pentagon
% with vertex nodes
%pentagonalmeshpreprocessor_fem_3iterations([0;0.0;0.45;1.1;1.1;0],[1;0.1;-
0.20;0.0;1.1;1],iteration)
%pentagonalmeshgeneratorforfemiterations([0;0.0;0.45;1.1;1.1;0],[1;0.1;-
0.20;0.0;1.1;1],iteration)
%iteration=1,2,3
%----------------------example-2-------------
%pentagonalmeshgeneratorforfemiterations([-1;1;2;0;-2;-1],[-1;-1;0;1;0;-1],1)
switch iteration
 case 1
figure(1)
plot(xv(:,1)',yv(:,1)','g')
hold on
%centroid of the given pentagonal region '(pxv,pyv)'
pxv(1,1)=(xv(1,1)+xv(2,1)+xv(3,1)+xv(4,1)+xv(5,1))/5;
pyv(1,1)=(yv(1,1)+yv(2,1)+yv(3,1)+yv(4,1)+yv(5,1))/5;
%find mid-points of five sides,say '(xu,yu)' and  say '(xw,yw)' mid-point of centroid c
and the midpoint of sides 'u'
for i=1:5
xu(i,1)=(xv(i,1)+xv(i+1,1))/2;yu(i,1)=(yv(i,1)+yv(i+1,1))/2;
xw(i,1)=(xu(i,1)+pxv(1,1))/2;yw(i,1)=(yu(i,1)+pyv(1,1))/2;
end
%we are assuming that the given pentagon has nodal vertices as 1,2,3,4,5 stored in a
vector nv()
%using this compute node numbers of midpoint vertices nu() and iiterior vertices nw()
[nv,nu,nw,elm]=nodaladdressesforpentagon(iteration)
%for i=1:5
 %   nv(i,1)=i;nu(i,1)=i+5;nw(i,1)=i+10;
%end
%nvv(1,:)=nv(:,1)'
%
% six new pentagons are created and their outlines are drawn in figure(1)
plot([xv(1,1),xu(1,1),xw(1,1),xw(5,1),xu(5,1),xv(1,1)],[yv(1,1),yu(1,1),yw(1,1),yw(5,1)
,yu(5,1),yv(1,1)],'r')
plot([xv(2,1),xu(2,1),xw(2,1),xw(1,1),xu(1,1),xv(2,1)],[yv(2,1),yu(2,1),yw(2,1),yw(1,1)
,yu(1,1),yv(2,1)],'r')
plot([xv(3,1),xu(3,1),xw(3,1),xw(2,1),xu(2,1),xv(3,1)],[yv(3,1),yu(3,1),yw(3,1),yw(2,1)
,yu(2,1),yv(3,1)],'r')
plot([xv(4,1),xu(4,1),xw(4,1),xw(3,1),xu(3,1),xv(4,1)],[yv(4,1),yu(4,1),yw(4,1),yw(3,1)
,yu(3,1),yv(4,1)],'r')
```

```
plot([xv(5,1),xu(5,1),xw(5,1),xw(4,1),xu(4,1),xv(5,1)],[yv(5,1),yu(5,1),yw(5,1),yw(4,1)
,yu(4,1),yv(5,1)],'r')
plot([xw(1,1),xw(2,1),xw(3,1),xw(4,1),xw(5,1),xw(1,1)],[yw(1,1),yw(2,1),yw(3,1),yw(4,1)
,yw(5,1),yw(1,1)],'b')
hold on
%
% outline of given pentagon with vertex nodes
%
for i=1:5
    text(xv(i,1),yv(i,1),['o',num2str(nv(i,1))])
    text(xu(i,1),yu(i,1),['o',num2str(nu(i,1))])
    text(xw(i,1),yw(i,1),['o',num2str(nw(i,1))])
end


%centroids of six new pentagons
pxv(2,1)=sum([xv(1,1),xu(1,1),xw(1,1),xw(5,1),xu(5,1)])/5;pyv(2,1)=sum([yv(1,1),yu(1,1)
,yw(1,1),yw(5,1),yu(5,1)])/5;
pxv(3,1)=sum([xv(2,1),xu(2,1),xw(2,1),xw(1,1),xu(1,1)])/5;pyv(3,1)=sum([yv(2,1),yu(2,1)
,yw(2,1),yw(1,1),yu(1,1)])/5;
pxv(4,1)=sum([xv(3,1),xu(3,1),xw(3,1),xw(2,1),xu(2,1)])/5;pyv(4,1)=sum([yv(3,1),yu(3,1)
,yw(3,1),yw(2,1),yu(2,1)])/5;
pxv(5,1)=sum([xv(4,1),xu(4,1),xw(4,1),xw(3,1),xu(3,1)])/5;pyv(5,1)=sum([yv(4,1),yu(4,1)
,yw(4,1),yw(3,1),yu(3,1)])/5;
pxv(6,1)=sum([xv(5,1),xu(5,1),xw(5,1),xw(4,1),xu(4,1)])/5;pyv(6,1)=sum([yv(5,1),yu(5,1)
,yw(5,1),yw(4,1),yu(4,1)])/5;
pxv(7,1)=sum([xw(1,1),xw(2,1),xw(3,1),xw(4,1),xw(5,1)])/5,pyv(7,1)=sum([yw(1,1),yw(2,1)
,yw(3,1),yw(4,1),yw(5,1)])/5;
for kk=2:7
text(pxv(kk,1),pyv(kk,1),['[',num2str(kk-1),']'])
end
elm(1:5,1)=[nv(1,1),nu(1,1),nw(1,1),nw(5,1),nu(5,1)]';
elm(1:5,2)=[nv(2,1),nu(2,1),nw(2,1),nw(1,1),nu(1,1)]';
elm(1:5,3)=[nv(3,1),nu(3,1),nw(3,1),nw(2,1),nu(2,1)]';
elm(1:5,4)=[nv(4,1),nu(4,1),nw(4,1),nw(3,1),nu(3,1)]';
elm(1:5,5)=[nv(5,1),nu(5,1),nw(5,1),nw(4,1),nu(4,1)]';
elm(1:5,6)=[nw(1,1),nw(2,1),nw(3,1),nw(4,1),nw(5,1)]';
[(1:5)' elm]
nnode=max(max(elm))
[nnel,nel]=size(elm)
hold on
xlabel('x axis')
ylabel('y axis')
st1='Mesh With ';
st2=num2str(nel);
st3=' Irregular ';
st4=' Pentagonal';
st5=' Elements'
st6=' &No. of Nodes= '
st7=num2str(nnode);
title([st1,st2,st3,st4,st5,st6,st7])
[(1:nel)' elm']
ELM=elm';
for pp=1:5
xcord(nv(pp,1),1)=xv(pp,1);ycord(nv(pp,1),1)=yv(pp,1);
xcord(nu(pp,1),1)=xu(pp,1);ycord(nu(pp,1),1)=yu(pp,1);
xcord(nw(pp,1),1)=xw(pp,1);ycord(nw(pp,1),1)=yw(pp,1);
end
[sn,sm]=size(xcord)
disp('----------------------------------------')
disp('xcoord                        ycoord')
[(1:sn)' xcord ycord]
```

```
if iteration==1
     return
 end

%-------------------------------------------------------------------------
case 2
figure(2)
plot(xv(:,1)',yv(:,1)','g')
hold on
%centroid of the given pentagonal region '(pxv,pyv)'
pxv(1,1)=(xv(1,1)+xv(2,1)+xv(3,1)+xv(4,1)+xv(5,1))/5;
pyv(1,1)=(yv(1,1)+yv(2,1)+yv(3,1)+yv(4,1)+yv(5,1))/5;
%find mid-points of five sides,say '(xu,yu)' and  say '(xw,yw)' mid-point of centroid c
and the midpoint of sides 'u'
for i=1:5
xu(i,1)=(xv(i,1)+xv(i+1,1))/2;yu(i,1)=(yv(i,1)+yv(i+1,1))/2;
xw(i,1)=(xu(i,1)+pxv(1,1))/2;yw(i,1)=(yu(i,1)+pyv(1,1))/2;
end
[nv,nu,nw,elm]=nodaladdressesforpentagon(1)
%compute node numbers of vertices
%for i=1:5
 %   nv(i,1)=i;nu(i,1)=i+5;nw(i,1)=i+10;
%end
%nvv(1,:)=nv(:,1)'
%evv(1,:)=[nv(1,1),nv(2,1),nv(3,1),nv(4,1),nv(5,1),nv(1,1)];
% outline of given pentagon with vertex nodes
%
% six new pentagons are created and their outlines are drawn in figure(1)
plot([xv(1,1),xu(1,1),xw(1,1),xw(5,1),xu(5,1),xv(1,1)],[yv(1,1),yu(1,1),yw(1,1),yw(5,1)
,yu(5,1),yv(1,1)],'r')
plot([xv(2,1),xu(2,1),xw(2,1),xw(1,1),xu(1,1),xv(2,1)],[yv(2,1),yu(2,1),yw(2,1),yw(1,1)
,yu(1,1),yv(2,1)],'r')
plot([xv(3,1),xu(3,1),xw(3,1),xw(2,1),xu(2,1),xv(3,1)],[yv(3,1),yu(3,1),yw(3,1),yw(2,1)
,yu(2,1),yv(3,1)],'r')
plot([xv(4,1),xu(4,1),xw(4,1),xw(3,1),xu(3,1),xv(4,1)],[yv(4,1),yu(4,1),yw(4,1),yw(3,1)
,yu(3,1),yv(4,1)],'r')
plot([xv(5,1),xu(5,1),xw(5,1),xw(4,1),xu(4,1),xv(5,1)],[yv(5,1),yu(5,1),yw(5,1),yw(4,1)
,yu(4,1),yv(5,1)],'r')
plot([xw(1,1),xw(2,1),xw(3,1),xw(4,1),xw(5,1),xw(1,1)],[yw(1,1),yw(2,1),yw(3,1),yw(4,1)
,yw(5,1),yw(1,1)],'b')
hold on

% outline of given pentagon with vertex nodes

for i=1:5
    text(xv(i,1),yv(i,1),['o',num2str(nv(i,1))])
    text(xu(i,1),yu(i,1),['o',num2str(nu(i,1))])
    text(xw(i,1),yw(i,1),['o',num2str(nw(i,1))])
end

%
elm(1:5,1)=[nv(1,1),nu(1,1),nw(1,1),nw(5,1),nu(5,1)]';
elm(1:5,2)=[nv(2,1),nu(2,1),nw(2,1),nw(1,1),nu(1,1)]';
elm(1:5,3)=[nv(3,1),nu(3,1),nw(3,1),nw(2,1),nu(2,1)]';
elm(1:5,4)=[nv(4,1),nu(4,1),nw(4,1),nw(3,1),nu(3,1)]';
elm(1:5,5)=[nv(5,1),nu(5,1),nw(5,1),nw(4,1),nu(4,1)]';
elm(1:5,6)=[nw(1,1),nw(2,1),nw(3,1),nw(4,1),nw(5,1)]';


%
```

```
%vertices of pentagons ((xv(1:5,kk),yv(1:5,kk)),kk=2,3,4,5,6,7) are now defined
xv(1:6,2)=[xv(1,1),xu(1,1),xw(1,1),xw(5,1),xu(5,1),xv(1,1)]';yv(1:6,2)=[yv(1,1),yu(1,1)
,yw(1,1),yw(5,1),yu(5,1),yv(1,1)]';
xv(1:6,3)=[xv(2,1),xu(2,1),xw(2,1),xw(1,1),xu(1,1),xv(2,1)]';yv(1:6,3)=[yv(2,1),yu(2,1)
,yw(2,1),yw(1,1),yu(1,1),yv(2,1)]';
xv(1:6,4)=[xv(3,1),xu(3,1),xw(3,1),xw(2,1),xu(2,1),xv(3,1)]';yv(1:6,4)=[yv(3,1),yu(3,1)
,yw(3,1),yw(2,1),yu(2,1),yv(3,1)]';
xv(1:6,5)=[xv(4,1),xu(4,1),xw(4,1),xw(3,1),xu(3,1),xv(4,1)]';yv(1:6,5)=[yv(4,1),yu(4,1)
,yw(4,1),yw(3,1),yu(3,1),yv(4,1)]';
xv(1:6,6)=[xv(5,1),xu(5,1),xw(5,1),xw(4,1),xu(4,1),xv(5,1)]';yv(1:6,6)=[yv(5,1),yu(5,1)
,yw(5,1),yw(4,1),yu(4,1),yv(5,1)]';
xv(1:6,7)=[xw(1,1),xw(2,1),xw(3,1),xw(4,1),xw(5,1),xw(1,1)]';yv(1:6,7)=[yw(1,1),yw(2,1)
,yw(3,1),yw(4,1),yw(5,1),yw(1,1)]';
for kk=2:7
pxv(kk,1)=(xv(1,kk)+xv(2,kk)+xv(3,kk)+xv(4,kk)+xv(5,kk))/5;
pyv(kk,1)=(yv(1,kk)+yv(2,kk)+yv(3,kk)+yv(4,kk)+yv(5,kk))/5;
%text(pxv(kk,1),pyv(kk,1),['[',num2str(kk-1),']'])
end
[(1:5)' elm]
[nnel,nel]=size(elm)


%-------------------------------------------------------------------------------------
--------------------
%now divide each of these six pentagons into six new pentagons
% note that xv(p,q) and yv(p,q) with p=1,2,3,4,5,6 andq=2,3,4,5,6,7
%refer to the vertices where (xv(6,q),yv(6,q))=(xv(1,q),yv(1,q))
%compute the centroids,midpoint nodes and interior nodes for each pentagon
for kk=2:7
pxv(kk,1)=(xv(1,kk)+xv(2,kk)+xv(3,kk)+xv(4,kk)+xv(5,kk))/5;
pyv(kk,1)=(yv(1,kk)+yv(2,kk)+yv(3,kk)+yv(4,kk)+yv(5,kk))/5;

for i=1:5
xu(i,kk)=(xv(i,kk)+xv(i+1,kk))/2;yu(i,kk)=(yv(i,kk)+yv(i+1,kk))/2;
xw(i,kk)=(xu(i,kk)+pxv(kk,1))/2;yw(i,kk)=(yu(i,kk)+pyv(kk,1))/2;
end
%divide each pentagon into six new pentagons
% 1st to 5th-pentagons adjacent to the perimeter are colored with blue lines
%6th pentagon is colored with red lines
plot([xv(1,kk),xu(1,kk),xw(1,kk),xw(5,kk),xu(5,kk),xv(1,kk)],[yv(1,kk),yu(1,kk),yw(1,kk
),yw(5,kk),yu(5,kk),yv(1,kk)],'b')
plot([xv(2,kk),xu(2,kk),xw(2,kk),xw(1,kk),xu(1,kk),xv(2,kk)],[yv(2,kk),yu(2,kk),yw(2,kk
),yw(1,kk),yu(1,kk),yv(2,kk)],'b')
plot([xv(3,kk),xu(3,kk),xw(3,kk),xw(2,kk),xu(2,kk),xv(3,kk)],[yv(3,kk),yu(3,kk),yw(3,kk
),yw(2,kk),yu(2,kk),yv(3,kk)],'b')
plot([xv(4,kk),xu(4,kk),xw(4,kk),xw(3,kk),xu(3,kk),xv(4,kk)],[yv(4,kk),yu(4,kk),yw(4,kk
),yw(3,kk),yu(3,kk),yv(4,kk)],'b')
plot([xv(5,kk),xu(5,kk),xw(5,kk),xw(4,kk),xu(4,kk),xv(5,kk)],[yv(5,kk),yu(5,kk),yw(5,kk
),yw(4,kk),yu(4,kk),yv(5,kk)],'b')
plot([xw(1,kk),xw(2,kk),xw(3,kk),xw(4,kk),xw(5,kk),xw(1,kk)],[yw(1,kk),yw(2,kk),yw(3,kk
),yw(4,kk),yw(5,kk),yw(1,kk)],'r')


end
%--------------------------WE HAVE DELETED THE FOLLOWING-----------------------------
-----------------------------------
nv(6,1)=nv(1,1)
%initial pentagon as closed domain has vertex nodes
[nv(1:6,1)]

%nodal vector for first pentagons of 2nd iteration
```

```
%nv(1:6,2)=[nv(1,1),nu(1,1),nw(1,1),nw(5,1),nu(5,1),nv(1,1)]';
%nv(1:6,3)=[nv(2,1),nu(2,1),nw(2,1),nw(1,1),nu(1,1),nv(2,1)]';
%nv(1:6,4)=[nv(3,1),nu(3,1),nw(3,1),nw(2,1),nu(2,1),nv(3,1)]';
%nv(1:6,5)=[nv(4,1),nu(4,1),nw(4,1),nw(3,1),nu(3,1),nv(4,1)]';
%nv(1:6,6)=[nv(5,1),nu(5,1),nw(5,1),nw(4,1),nu(4,1),nv(5,1)]';
%nv(1:6,7)=[nw(1,1),nw(2,1),nw(3,1),nw(4,1),nw(5,1),nw(1,1)]';
%


[nvv,nuu,nww,elm]=nodaladdressesforpentagon(2)

for iel=1:nel
    nv(1:6,iel+1)=nvv(1:6,iel);
    nu(1:5,iel+1)=nuu(1:5,iel);
    nw(1:5,iel+1)=nww(1:5,iel);
end


kk=1
%kkk=max(max(elm))

for j=2:7

elm(1:5,kk)=[nv(1,j),nu(1,j),nw(1,j),nw(5,j),nu(5,j)]';
elm(1:5,kk+1)=[nv(2,j),nu(2,j),nw(2,j),nw(1,j),nu(1,j)]';
elm(1:5,kk+2)=[nv(3,j),nu(3,j),nw(3,j),nw(2,j),nu(2,j)]';
elm(1:5,kk+3)=[nv(4,j),nu(4,j),nw(4,j),nw(3,j),nu(3,j)]';
elm(1:5,kk+4)=[nv(5,j),nu(5,j),nw(5,j),nw(4,j),nu(4,j)]';
elm(1:5,kk+5)=[nw(1,j),nw(2,j),nw(3,j),nw(4,j),nw(5,j)]';
%elnum(:,1)=[kk;kk+1;kk+2;kk+3;kk+4;kk+5];
%[elnum elm(kk:kk+5,:)]
%[mdpt(1,6),mdpt(6,11),mdpt(11,15),mdpt(15,10),mdpt(10,1)]
%[mdpt(6,1),mdpt(11,6),mdpt(15,11),mdpt(10,15),mdpt(1,10)]
%-----------------------------------------------------------WE HAVE DELETED THE ABOVE-
------------------------------

for i=1:5
    text(xv(i,j),yv(i,j),['o',num2str(nv(i,j))])
    text(xu(i,j),yu(i,j),['o',num2str(nu(i,j))])
    text(xw(i,j),yw(i,j),['o',num2str(nw(i,j))])
end
pxvv(kk,1)=(xv(1,j)+xu(1,j)+xw(1,j)+xw(5,j)+xu(5,j))/5;pyvv(kk,1)=(yv(1,j)+yu(1,j)+yw(1
,j)+yw(5,j)+yu(5,j))/5;
pxvv(kk+1,1)=(xv(2,j)+xu(2,j)+xw(2,j)+xw(1,j)+xu(1,j))/5;pyvv(kk+1,1)=(yv(2,j)+yu(2,j)+
yw(2,j)+yw(1,j)+yu(1,j))/5;
pxvv(kk+2,1)=(xv(3,j)+xu(3,j)+xw(3,j)+xw(2,j)+xu(2,j))/5;pyvv(kk+2,1)=(yv(3,j)+yu(3,j)+
yw(3,j)+yw(2,j)+yu(2,j))/5;
pxvv(kk+3,1)=(xv(4,j)+xu(4,j)+xw(4,j)+xw(3,j)+xu(3,j))/5;pyvv(kk+3,1)=(yv(4,j)+yu(4,j)+
yw(4,j)+yw(3,j)+yu(3,j))/5;
pxvv(kk+4,1)=(xv(5,j)+xu(5,j)+xw(5,j)+xw(4,j)+xu(4,j))/5;pyvv(kk+4,1)=(yv(5,j)+yu(5,j)+
yw(5,j)+yw(4,j)+yu(4,j))/5;
pxvv(kk+5,1)=(xw(1,j)+xw(2,j)+xw(3,j)+xw(4,j)+xw(5,j))/5;pyvv(kk+5,1)=(yw(1,j)+yw(2,j)+
yw(3,j)+yw(4,j)+yw(5,j))/5;
%
text(pxvv(kk,1),pyvv(kk,1),['[',num2str(kk),']'])
text(pxvv(kk+1,1),pyvv(kk+1,1),['[',num2str(kk+1),']'])
text(pxvv(kk+2,1),pyvv(kk+2,1),['[',num2str(kk+2),']'])
text(pxvv(kk+3,1),pyvv(kk+3,1),['[',num2str(kk+3),']'])
text(pxvv(kk+4,1),pyvv(kk+4,1),['[',num2str(kk+4),']'])
text(pxvv(kk+5,1),pyvv(kk+5,1),['[',num2str(kk+5),']'])
```

```
%=========================================================================================
==========================
kk=kk+6
end%(for j==2:7)
hold on
nnode=max(max(elm))
[nnel,nel]=size(elm)
%[nel,nnel]=size(elm)
hold on
xlabel('x axis')
ylabel('y axis')
st1='FEM Mesh With ';
st2=num2str(nel);
st3='  Irregular';
st4=' Pentagonal';
st5=' Elements'
st6='& No. of Nodes=  '
st7=num2str(nnode);
title([st1,st2,st3,st4,st5,st6,st7])
[(1:nel)' elm']
ELM=elm';
hold on

for qq=1:7
for pp=1:5
xcord(nv(pp,qq),1)=xv(pp,qq);ycord(nv(pp,qq),1)=yv(pp,qq);
xcord(nu(pp,qq),1)=xu(pp,qq);ycord(nu(pp,qq),1)=yu(pp,qq);
xcord(nw(pp,qq),1)=xw(pp,qq);ycord(nw(pp,qq),1)=yw(pp,qq);
end
end
[sn,sm]=size(xcord)
disp('----------------------------------------')
disp('xcoord                        ycoord')
[(1:sn)' xcord ycord]
 if iteration==2
     return
 end

 case 3
figure(3)
plot(xv(:,1)',yv(:,1)','g')
hold on
%centroid of the given pentagonal region '(pxv,pyv)'
pxv(1,1)=(xv(1,1)+xv(2,1)+xv(3,1)+xv(4,1)+xv(5,1))/5;
pyv(1,1)=(yv(1,1)+yv(2,1)+yv(3,1)+yv(4,1)+yv(5,1))/5;
%find mid-points of five sides,say '(xu,yu)' and  say '(xw,yw)' mid-point of centroid c
and the midpoint of sides 'u'
for i=1:5
xu(i,1)=(xv(i,1)+xv(i+1,1))/2;yu(i,1)=(yv(i,1)+yv(i+1,1))/2;
xw(i,1)=(xu(i,1)+pxv(1,1))/2;yw(i,1)=(yu(i,1)+pyv(1,1))/2;
end

[nv,nu,nw,elm]=nodaladdressesforpentagon(1)
%compute node numbers of vertices
%for i=1:5
 %   nv(i,1)=i;nu(i,1)=i+5;nw(i,1)=i+10;
%end
%nvv(1,:)=nv(:,1)'
```

```
%evv(1,:)=[nv(1,1),nv(2,1),nv(3,1),nv(4,1),nv(5,1),nv(1,1)];
% outline of given pentagon with vertex nodes
%
% six new pentagons are created and their outlines are drawn in figure(1)
plot([xv(1,1),xu(1,1),xw(1,1),xw(5,1),xu(5,1),xv(1,1)],[yv(1,1),yu(1,1),yw(1,1),yw(5,1)
,yu(5,1),yv(1,1)],'r')
plot([xv(2,1),xu(2,1),xw(2,1),xw(1,1),xu(1,1),xv(2,1)],[yv(2,1),yu(2,1),yw(2,1),yw(1,1)
,yu(1,1),yv(2,1)],'r')
plot([xv(3,1),xu(3,1),xw(3,1),xw(2,1),xu(2,1),xv(3,1)],[yv(3,1),yu(3,1),yw(3,1),yw(2,1)
,yu(2,1),yv(3,1)],'r')
plot([xv(4,1),xu(4,1),xw(4,1),xw(3,1),xu(3,1),xv(4,1)],[yv(4,1),yu(4,1),yw(4,1),yw(3,1)
,yu(3,1),yv(4,1)],'r')
plot([xv(5,1),xu(5,1),xw(5,1),xw(4,1),xu(4,1),xv(5,1)],[yv(5,1),yu(5,1),yw(5,1),yw(4,1)
,yu(4,1),yv(5,1)],'r')
plot([xw(1,1),xw(2,1),xw(3,1),xw(4,1),xw(5,1),xw(1,1)],[yw(1,1),yw(2,1),yw(3,1),yw(4,1)
,yw(5,1),yw(1,1)],'b')
hold on

% outline of given pentagon with vertex nodes

for i=1:5
    text(xv(i,1),yv(i,1),['o',num2str(nv(i,1))])
    text(xu(i,1),yu(i,1),['o',num2str(nu(i,1))])
    text(xw(i,1),yw(i,1),['o',num2str(nw(i,1))])
end
 %
 %
elm(1:5,1)=[nv(1,1),nu(1,1),nw(1,1),nw(5,1),nu(5,1)]';
elm(1:5,2)=[nv(2,1),nu(2,1),nw(2,1),nw(1,1),nu(1,1)]';
elm(1:5,3)=[nv(3,1),nu(3,1),nw(3,1),nw(2,1),nu(2,1)]';
elm(1:5,4)=[nv(4,1),nu(4,1),nw(4,1),nw(3,1),nu(3,1)]';
elm(1:5,5)=[nv(5,1),nu(5,1),nw(5,1),nw(4,1),nu(4,1)]';
elm(1:5,6)=[nw(1,1),nw(2,1),nw(3,1),nw(4,1),nw(5,1)]';


%

%vertices of pentagons ((xv(1:5,kk),yv(1:5,kk)),kk=2,3,4,5,6,7) are now defined
xv(1:6,2)=[xv(1,1),xu(1,1),xw(1,1),xw(5,1),xu(5,1),xv(1,1)]';yv(1:6,2)=[yv(1,1),yu(1,1)
,yw(1,1),yw(5,1),yu(5,1),yv(1,1)]';
xv(1:6,3)=[xv(2,1),xu(2,1),xw(2,1),xw(1,1),xu(1,1),xv(2,1)]';yv(1:6,3)=[yv(2,1),yu(2,1)
,yw(2,1),yw(1,1),yu(1,1),yv(2,1)]';
xv(1:6,4)=[xv(3,1),xu(3,1),xw(3,1),xw(2,1),xu(2,1),xv(3,1)]';yv(1:6,4)=[yv(3,1),yu(3,1)
,yw(3,1),yw(2,1),yu(2,1),yv(3,1)]';
xv(1:6,5)=[xv(4,1),xu(4,1),xw(4,1),xw(3,1),xu(3,1),xv(4,1)]';yv(1:6,5)=[yv(4,1),yu(4,1)
,yw(4,1),yw(3,1),yu(3,1),yv(4,1)]';
xv(1:6,6)=[xv(5,1),xu(5,1),xw(5,1),xw(4,1),xu(4,1),xv(5,1)]';yv(1:6,6)=[yv(5,1),yu(5,1)
,yw(5,1),yw(4,1),yu(4,1),yv(5,1)]';
xv(1:6,7)=[xw(1,1),xw(2,1),xw(3,1),xw(4,1),xw(5,1),xw(1,1)]';yv(1:6,7)=[yw(1,1),yw(2,1)
,yw(3,1),yw(4,1),yw(5,1),yw(1,1)]';
for kk=2:7
pxv(kk,1)=(xv(1,kk)+xv(2,kk)+xv(3,kk)+xv(4,kk)+xv(5,kk))/5;
pyv(kk,1)=(yv(1,kk)+yv(2,kk)+yv(3,kk)+yv(4,kk)+yv(5,kk))/5;
%text(pxv(kk,1),pyv(kk,1),['[',num2str(kk-1),']'])
end
[(1:5)' elm]
%-------------------------------------------------------------------------------
--------------------
%now divide each of these six pentagons into six new pentagons
% note that xv(p,q) and yv(p,q) with p=1,2,3,4,5,6 andq=2,3,4,5,6,7
%refer to the vertices where (xv(6,q),yv(6,q))=(xv(1,q),yv(1,q))
```

```
%compute the centroids,midpoint nodes and interior nodes for each pentagon
for kk=2:7
pxv(kk,1)=(xv(1,kk)+xv(2,kk)+xv(3,kk)+xv(4,kk)+xv(5,kk))/5;
pyv(kk,1)=(yv(1,kk)+yv(2,kk)+yv(3,kk)+yv(4,kk)+yv(5,kk))/5;

for i=1:5
xu(i,kk)=(xv(i,kk)+xv(i+1,kk))/2;yu(i,kk)=(yv(i,kk)+yv(i+1,kk))/2;
xw(i,kk)=(xu(i,kk)+pxv(kk,1))/2;yw(i,kk)=(yu(i,kk)+pyv(kk,1))/2;
end
%divide each pentagon into six new pentagons
% 1st to 5th-pentagons adjacent to the perimeter are colored with blue lines
%6th pentagon is colored with red lines
plot([xv(1,kk),xu(1,kk),xw(1,kk),xw(5,kk),xu(5,kk),xv(1,kk)],[yv(1,kk),yu(1,kk),yw(1,kk
),yw(5,kk),yu(5,kk),yv(1,kk)],'b')
plot([xv(2,kk),xu(2,kk),xw(2,kk),xw(1,kk),xu(1,kk),xv(2,kk)],[yv(2,kk),yu(2,kk),yw(2,kk
),yw(1,kk),yu(1,kk),yv(2,kk)],'b')
plot([xv(3,kk),xu(3,kk),xw(3,kk),xw(2,kk),xu(2,kk),xv(3,kk)],[yv(3,kk),yu(3,kk),yw(3,kk
),yw(2,kk),yu(2,kk),yv(3,kk)],'b')
plot([xv(4,kk),xu(4,kk),xw(4,kk),xw(3,kk),xu(3,kk),xv(4,kk)],[yv(4,kk),yu(4,kk),yw(4,kk
),yw(3,kk),yu(3,kk),yv(4,kk)],'b')
plot([xv(5,kk),xu(5,kk),xw(5,kk),xw(4,kk),xu(4,kk),xv(5,kk)],[yv(5,kk),yu(5,kk),yw(5,kk
),yw(4,kk),yu(4,kk),yv(5,kk)],'b')
plot([xw(1,kk),xw(2,kk),xw(3,kk),xw(4,kk),xw(5,kk),xw(1,kk)],[yw(1,kk),yw(2,kk),yw(3,kk
),yw(4,kk),yw(5,kk),yw(1,kk)],'r')

end%for kk=2:7
nv(6,1)=nv(1,1)
%initial pentagon as closed domain has vertex nodes
[nv(1:6,1)]

[nvv,nuu,nww,elm]=nodaladdressesforpentagon(2)


%

kk=1
%kkk=max(max(elm))

for j=2:7




%-------------------------------------------------------------------------

for i=1:5
    text(xv(i,j),yv(i,j),['o',num2str(nvv(i,j-1))])
    text(xu(i,j),yu(i,j),['o',num2str(nuu(i,j-1))])
    text(xw(i,j),yw(i,j),['o',num2str(nww(i,j-1))])
end


%=========================================================================================
==============
pxvv(kk,1)=(xv(1,j)+xu(1,j)+xw(1,j)+xw(5,j)+xu(5,j))/5;pyvv(kk,1)=(yv(1,j)+yu(1,j)+yw(1
,j)+yw(5,j)+yu(5,j))/5;
pxvv(kk+1,1)=(xv(2,j)+xu(2,j)+xw(2,j)+xw(1,j)+xu(1,j))/5;pyvv(kk+1,1)=(yv(2,j)+yu(2,j)+
yw(2,j)+yw(1,j)+yu(1,j))/5;
```

```
pxvv(kk+2,1)=(xv(3,j)+xu(3,j)+xw(3,j)+xw(2,j)+xu(2,j))/5;pyvv(kk+2,1)=(yv(3,j)+yu(3,j)+
yw(3,j)+yw(2,j)+yu(2,j))/5;
pxvv(kk+3,1)=(xv(4,j)+xu(4,j)+xw(4,j)+xw(3,j)+xu(3,j))/5;pyvv(kk+3,1)=(yv(4,j)+yu(4,j)+
yw(4,j)+yw(3,j)+yu(3,j))/5;
pxvv(kk+4,1)=(xv(5,j)+xu(5,j)+xw(5,j)+xw(4,j)+xu(4,j))/5;pyvv(kk+4,1)=(yv(5,j)+yu(5,j)+
yw(5,j)+yw(4,j)+yu(4,j))/5;
pxvv(kk+5,1)=(xw(1,j)+xw(2,j)+xw(3,j)+xw(4,j)+xw(5,j))/5;pyvv(kk+5,1)=(yw(1,j)+yw(2,j)+
yw(3,j)+yw(4,j)+yw(5,j))/5;
%

kk=kk+6
end%(for j==2:7)

%hold on
[nel,nnel]=size(elm)
%hold on
%xlabel('x axis')
%ylabel('y axis')
%st1='FEM Mesh With ';
%st2=num2str(nel);
%st3='  Irregular';
%st4=' Pentagonal';
%st5=' Elements'
%st6='& No. of Nodes=  '
%st7=num2str(nnode);
%title([st1,st2,st3,st4,st5,st6,st7])


hold on


%make further refinement of 36-element mesh
for kk=2:7
pxv(kk,1)=(xv(1,kk)+xv(2,kk)+xv(3,kk)+xv(4,kk)+xv(5,kk))/5;
pyv(kk,1)=(yv(1,kk)+yv(2,kk)+yv(3,kk)+yv(4,kk)+yv(5,kk))/5;

for i=1:5
xu(i,kk)=(xv(i,kk)+xv(i+1,kk))/2;yu(i,kk)=(yv(i,kk)+yv(i+1,kk))/2;
xw(i,kk)=(xu(i,kk)+pxv(kk,1))/2;yw(i,kk)=(yu(i,kk)+pyv(kk,1))/2;
end
%divide each pentagon into six new pentagons
% 1st to 5th-pentagons adjacent to the perimeter are colored with blue lines
%6th pentagon is colored with red lines
plot([xv(1,kk),xu(1,kk),xw(1,kk),xw(5,kk),xu(5,kk),xv(1,kk)],[yv(1,kk),yu(1,kk),yw(1,kk
),yw(5,kk),yu(5,kk),yv(1,kk)],'b')
plot([xv(2,kk),xu(2,kk),xw(2,kk),xw(1,kk),xu(1,kk),xv(2,kk)],[yv(2,kk),yu(2,kk),yw(2,kk
),yw(1,kk),yu(1,kk),yv(2,kk)],'b')
plot([xv(3,kk),xu(3,kk),xw(3,kk),xw(2,kk),xu(2,kk),xv(3,kk)],[yv(3,kk),yu(3,kk),yw(3,kk
),yw(2,kk),yu(2,kk),yv(3,kk)],'b')
plot([xv(4,kk),xu(4,kk),xw(4,kk),xw(3,kk),xu(3,kk),xv(4,kk)],[yv(4,kk),yu(4,kk),yw(4,kk
),yw(3,kk),yu(3,kk),yv(4,kk)],'b')
plot([xv(5,kk),xu(5,kk),xw(5,kk),xw(4,kk),xu(4,kk),xv(5,kk)],[yv(5,kk),yu(5,kk),yw(5,kk
),yw(4,kk),yu(4,kk),yv(5,kk)],'b')
plot([xw(1,kk),xw(2,kk),xw(3,kk),xw(4,kk),xw(5,kk),xw(1,kk)],[yw(1,kk),yw(2,kk),yw(3,kk
),yw(4,kk),yw(5,kk),yw(1,kk)],'r')


hold on

end%for kk=2:7
```

```
hold on
%end%switch iteration
%return
hold on
%if iteration==3
    mm=8;
for kk=2:7

xv(1:6,mm)=[xv(1,kk),xu(1,kk),xw(1,kk),xw(5,kk),xu(5,kk),xv(1,kk)]';yv(1:6,mm)=[yv(1,kk
),yu(1,kk),yw(1,kk),yw(5,kk),yu(5,kk),yv(1,kk)]';
xv(1:6,mm+1)=[xv(2,kk),xu(2,kk),xw(2,kk),xw(1,kk),xu(1,kk),xv(2,kk)]';yv(1:6,mm+1)=[yv(
2,kk),yu(2,kk),yw(2,kk),yw(1,kk),yu(1,kk),yv(2,kk)]';
xv(1:6,mm+2)=[xv(3,kk),xu(3,kk),xw(3,kk),xw(2,kk),xu(2,kk),xv(3,kk)]';yv(1:6,mm+2)=[yv(
3,kk),yu(3,kk),yw(3,kk),yw(2,kk),yu(2,kk),yv(3,kk)]';
xv(1:6,mm+3)=[xv(4,kk),xu(4,kk),xw(4,kk),xw(3,kk),xu(3,kk),xv(4,kk)]';yv(1:6,mm+3)=[yv(
4,kk),yu(4,kk),yw(4,kk),yw(3,kk),yu(3,kk),yv(4,kk)]';
xv(1:6,mm+4)=[xv(5,kk),xu(5,kk),xw(5,kk),xw(4,kk),xu(4,kk),xv(5,kk)]';yv(1:6,mm+4)=[yv(
5,kk),yu(5,kk),yw(5,kk),yw(4,kk),yu(4,kk),yv(5,kk)]';
xv(1:6,mm+5)=[xw(1,kk),xw(2,kk),xw(3,kk),xw(4,kk),xw(5,kk),xw(1,kk)]';yv(1:6,mm+5)=[yw(
1,kk),yw(2,kk),yw(3,kk),yw(4,kk),yw(5,kk),yw(1,kk)]';
for nn=mm:mm+5
pppp=0;qqqq=0;
for  i=1:5
pppp=pppp+xv(i,nn);qqqq=qqqq+yv(i,nn);
end
pxvv(nn,1)=pppp/5;pyvv(nn,1)=qqqq/5;
for i=1:5
xu(i,nn)=(xv(i,nn)+xv(i+1,nn))/2;yu(i,nn)=(yv(i,nn)+yv(i+1,nn))/2;
xw(i,nn)=(xu(i,nn)+pxvv(nn,1))/2;yw(i,nn)=(yu(i,nn)+pyvv(nn,1))/2;
end

plot([xv(1,nn),xu(1,nn),xw(1,nn),xw(5,nn),xu(5,nn),xv(1,nn)],[yv(1,nn),yu(1,nn),yw(1,nn
),yw(5,nn),yu(5,nn),yv(1,nn)],'b')
plot([xv(2,nn),xu(2,nn),xw(2,nn),xw(1,nn),xu(1,nn),xv(2,nn)],[yv(2,nn),yu(2,nn),yw(2,nn
),yw(1,nn),yu(1,nn),yv(2,nn)],'b')
plot([xv(3,nn),xu(3,nn),xw(3,nn),xw(2,nn),xu(2,nn),xv(3,nn)],[yv(3,nn),yu(3,nn),yw(3,nn
),yw(2,nn),yu(2,nn),yv(3,nn)],'b')
plot([xv(4,nn),xu(4,nn),xw(4,nn),xw(3,nn),xu(3,nn),xv(4,nn)],[yv(4,nn),yu(4,nn),yw(4,nn
),yw(3,nn),yu(3,nn),yv(4,nn)],'b')
plot([xv(5,nn),xu(5,nn),xw(5,nn),xw(4,nn),xu(4,nn),xv(5,nn)],[yv(5,nn),yu(5,nn),yw(5,nn
),yw(4,nn),yu(4,nn),yv(5,nn)],'b')
plot([xw(1,nn),xw(2,nn),xw(3,nn),xw(4,nn),xw(5,nn),xw(1,nn)],[yw(1,nn),yw(2,nn),yw(3,nn
),yw(4,nn),yw(5,nn),yw(1,nn)],'r')
end%for nn=mm:(mm+5)
mm=mm+6;
end%for kk=2:7


%-----------------------------------
[nvv,nuu,nww,elm]=nodaladdressesforpentagon(3)



kk=1


for j=8:43

%-------------------------------------------------------------------------
```

```
for i=1:5
    text(xv(i,j),yv(i,j),['o',num2str(nvv(i,j-7))])
    text(xu(i,j),yu(i,j),['o',num2str(nuu(i,j-7))])
    text(xw(i,j),yw(i,j),['o',num2str(nww(i,j-7))])
end

pxvv(kk,1)=(xv(1,j)+xu(1,j)+xw(1,j)+xw(5,j)+xu(5,j))/5;pyvv(kk,1)=(yv(1,j)+yu(1,j)+yw(1
,j)+yw(5,j)+yu(5,j))/5;
pxvv(kk+1,1)=(xv(2,j)+xu(2,j)+xw(2,j)+xw(1,j)+xu(1,j))/5;pyvv(kk+1,1)=(yv(2,j)+yu(2,j)+
yw(2,j)+yw(1,j)+yu(1,j))/5;
pxvv(kk+2,1)=(xv(3,j)+xu(3,j)+xw(3,j)+xw(2,j)+xu(2,j))/5;pyvv(kk+2,1)=(yv(3,j)+yu(3,j)+
yw(3,j)+yw(2,j)+yu(2,j))/5;
pxvv(kk+3,1)=(xv(4,j)+xu(4,j)+xw(4,j)+xw(3,j)+xu(3,j))/5;pyvv(kk+3,1)=(yv(4,j)+yu(4,j)+
yw(4,j)+yw(3,j)+yu(3,j))/5;
pxvv(kk+4,1)=(xv(5,j)+xu(5,j)+xw(5,j)+xw(4,j)+xu(4,j))/5;pyvv(kk+4,1)=(yv(5,j)+yu(5,j)+
yw(5,j)+yw(4,j)+yu(4,j))/5;
pxvv(kk+5,1)=(xw(1,j)+xw(2,j)+xw(3,j)+xw(4,j)+xw(5,j))/5;pyvv(kk+5,1)=(yw(1,j)+yw(2,j)+
yw(3,j)+yw(4,j)+yw(5,j))/5;
%

%
text(pxvv(kk,1),pyvv(kk,1),['[',num2str(kk),']'])
text(pxvv(kk+1,1),pyvv(kk+1,1),['[',num2str(kk+1),']'])
text(pxvv(kk+2,1),pyvv(kk+2,1),['[',num2str(kk+2),']'])
text(pxvv(kk+3,1),pyvv(kk+3,1),['[',num2str(kk+3),']'])
text(pxvv(kk+4,1),pyvv(kk+4,1),['[',num2str(kk+4),']'])
text(pxvv(kk+5,1),pyvv(kk+5,1),['[',num2str(kk+5),']'])



%=============================================================================================
============================
kk=kk+6
end%(for j==8:43)

hold on
hold on
nnode=max(max(elm))
[nnel,nel]=size(elm)
hold on
xlabel('x axis')
ylabel('y axis')
st1='FEM Mesh With ';
st2=num2str(nel);
st3='  Irregular';
st4=' Pentagonal';
st5=' Elements'
st6='& No. of Nodes=  '
st7=num2str(nnode);
title([st1,st2,st3,st4,st5,st6,st7])
[(1:nel)' elm']
ELM=elm';
hold on

for qq=8:43
for pp=1:5
xcord(nvv(pp,qq-7),1)=xv(pp,qq);ycord(nvv(pp,qq-7),1)=yv(pp,qq);
xcord(nuu(pp,qq-7),1)=xu(pp,qq);ycord(nuu(pp,qq-7),1)=yu(pp,qq);
xcord(nww(pp,qq-7),1)=xw(pp,qq);ycord(nww(pp,qq-7),1)=yw(pp,qq);
end
end
```

```
[sn,sm]=size(xcord)
disp('----------------------------------------')
disp('xcoord                          ycoord')
[(1:sn)' xcord ycord]
 if iteration==3
     return
 end
```

```
%*************************************************************************************
***************
```

```
 case 4
figure(4)
plot(xv(:,1)',yv(:,1)','g')
hold on
%centroid of the given pentagonal region '(pxv,pyv)'
pxv(1,1)=(xv(1,1)+xv(2,1)+xv(3,1)+xv(4,1)+xv(5,1))/5;
pyv(1,1)=(yv(1,1)+yv(2,1)+yv(3,1)+yv(4,1)+yv(5,1))/5;
%find mid-points of five sides,say '(xu,yu)' and  say '(xw,yw)' mid-point of centroid c
and the midpoint of sides 'u'
for i=1:5
xu(i,1)=(xv(i,1)+xv(i+1,1))/2;yu(i,1)=(yv(i,1)+yv(i+1,1))/2;
xw(i,1)=(xu(i,1)+pxv(1,1))/2;yw(i,1)=(yu(i,1)+pyv(1,1))/2;
end

[nv,nu,nw,elm]=nodaladdressesforpentagon(1)
%compute node numbers of vertices
%for i=1:5
 %   nv(i,1)=i;nu(i,1)=i+5;nw(i,1)=i+10;
%end
%nvv(1,:)=nv(:,1)'
%evv(1,:)=[nv(1,1),nv(2,1),nv(3,1),nv(4,1),nv(5,1),nv(1,1)];
% outline of given pentagon with vertex nodes
%
% six new pentagons are created and their outlines are drawn in figure(1)
plot([xv(1,1),xu(1,1),xw(1,1),xw(5,1),xu(5,1),xv(1,1)],[yv(1,1),yu(1,1),yw(1,1),yw(5,1)
,yu(5,1),yv(1,1)],'r')
plot([xv(2,1),xu(2,1),xw(2,1),xw(1,1),xu(1,1),xv(2,1)],[yv(2,1),yu(2,1),yw(2,1),yw(1,1)
,yu(1,1),yv(2,1)],'r')
plot([xv(3,1),xu(3,1),xw(3,1),xw(2,1),xu(2,1),xv(3,1)],[yv(3,1),yu(3,1),yw(3,1),yw(2,1)
,yu(2,1),yv(3,1)],'r')
plot([xv(4,1),xu(4,1),xw(4,1),xw(3,1),xu(3,1),xv(4,1)],[yv(4,1),yu(4,1),yw(4,1),yw(3,1)
,yu(3,1),yv(4,1)],'r')
plot([xv(5,1),xu(5,1),xw(5,1),xw(4,1),xu(4,1),xv(5,1)],[yv(5,1),yu(5,1),yw(5,1),yw(4,1)
,yu(4,1),yv(5,1)],'r')
plot([xw(1,1),xw(2,1),xw(3,1),xw(4,1),xw(5,1),xw(1,1)],[yw(1,1),yw(2,1),yw(3,1),yw(4,1)
,yw(5,1),yw(1,1)],'b')
hold on

% outline of given pentagon with vertex nodes

for i=1:5
    text(xv(i,1),yv(i,1),['o',num2str(nv(i,1))])
    text(xu(i,1),yu(i,1),['o',num2str(nu(i,1))])
    text(xw(i,1),yw(i,1),['o',num2str(nw(i,1))])
```

```
end
 %
 %
elm(1:5,1)=[nv(1,1),nu(1,1),nw(1,1),nw(5,1),nu(5,1)]';
elm(1:5,2)=[nv(2,1),nu(2,1),nw(2,1),nw(1,1),nu(1,1)]';
elm(1:5,3)=[nv(3,1),nu(3,1),nw(3,1),nw(2,1),nu(2,1)]';
elm(1:5,4)=[nv(4,1),nu(4,1),nw(4,1),nw(3,1),nu(3,1)]';
elm(1:5,5)=[nv(5,1),nu(5,1),nw(5,1),nw(4,1),nu(4,1)]';
elm(1:5,6)=[nw(1,1),nw(2,1),nw(3,1),nw(4,1),nw(5,1)]';


%

%vertices of pentagons ((xv(1:5,kk),yv(1:5,kk)),kk=2,3,4,5,6,7) are now defined
xv(1:6,2)=[xv(1,1),xu(1,1),xw(1,1),xw(5,1),xu(5,1),xv(1,1)]';yv(1:6,2)=[yv(1,1),yu(1,1)
,yw(1,1),yw(5,1),yu(5,1),yv(1,1)]';
xv(1:6,3)=[xv(2,1),xu(2,1),xw(2,1),xw(1,1),xu(1,1),xv(2,1)]';yv(1:6,3)=[yv(2,1),yu(2,1)
,yw(2,1),yw(1,1),yu(1,1),yv(2,1)]';
xv(1:6,4)=[xv(3,1),xu(3,1),xw(3,1),xw(2,1),xu(2,1),xv(3,1)]';yv(1:6,4)=[yv(3,1),yu(3,1)
,yw(3,1),yw(2,1),yu(2,1),yv(3,1)]';
xv(1:6,5)=[xv(4,1),xu(4,1),xw(4,1),xw(3,1),xu(3,1),xv(4,1)]';yv(1:6,5)=[yv(4,1),yu(4,1)
,yw(4,1),yw(3,1),yu(3,1),yv(4,1)]';
xv(1:6,6)=[xv(5,1),xu(5,1),xw(5,1),xw(4,1),xu(4,1),xv(5,1)]';yv(1:6,6)=[yv(5,1),yu(5,1)
,yw(5,1),yw(4,1),yu(4,1),yv(5,1)]';
xv(1:6,7)=[xw(1,1),xw(2,1),xw(3,1),xw(4,1),xw(5,1),xw(1,1)]';yv(1:6,7)=[yw(1,1),yw(2,1)
,yw(3,1),yw(4,1),yw(5,1),yw(1,1)]';
for kk=2:7
pxv(kk,1)=(xv(1,kk)+xv(2,kk)+xv(3,kk)+xv(4,kk)+xv(5,kk))/5;
pyv(kk,1)=(yv(1,kk)+yv(2,kk)+yv(3,kk)+yv(4,kk)+yv(5,kk))/5;
%text(pxv(kk,1),pyv(kk,1),['[',num2str(kk-1),']'])
end
[(1:5)' elm]
%-----------------------------------------------------------------------------------------
--------------------
%now divide each of these six pentagons into six new pentagons
% note that xv(p,q) and yv(p,q) with p=1,2,3,4,5,6 andq=2,3,4,5,6,7
%refer to the vertices where (xv(6,q),yv(6,q))=(xv(1,q),yv(1,q))
%compute the centroids,midpoint nodes and interior nodes for each pentagon
for kk=2:7
pxv(kk,1)=(xv(1,kk)+xv(2,kk)+xv(3,kk)+xv(4,kk)+xv(5,kk))/5;
pyv(kk,1)=(yv(1,kk)+yv(2,kk)+yv(3,kk)+yv(4,kk)+yv(5,kk))/5;

for i=1:5
xu(i,kk)=(xv(i,kk)+xv(i+1,kk))/2;yu(i,kk)=(yv(i,kk)+yv(i+1,kk))/2;
xw(i,kk)=(xu(i,kk)+pxv(kk,1))/2;yw(i,kk)=(yu(i,kk)+pyv(kk,1))/2;
end
%divide each pentagon into six new pentagons
% 1st to 5th-pentagons adjacent to the perimeter are colored with blue lines
%6th pentagon is colored with red lines
plot([xv(1,kk),xu(1,kk),xw(1,kk),xw(5,kk),xu(5,kk),xv(1,kk)],[yv(1,kk),yu(1,kk),yw(1,kk
),yw(5,kk),yu(5,kk),yv(1,kk)],'b')
plot([xv(2,kk),xu(2,kk),xw(2,kk),xw(1,kk),xu(1,kk),xv(2,kk)],[yv(2,kk),yu(2,kk),yw(2,kk
),yw(1,kk),yu(1,kk),yv(2,kk)],'b')
plot([xv(3,kk),xu(3,kk),xw(3,kk),xw(2,kk),xu(2,kk),xv(3,kk)],[yv(3,kk),yu(3,kk),yw(3,kk
),yw(2,kk),yu(2,kk),yv(3,kk)],'b')
plot([xv(4,kk),xu(4,kk),xw(4,kk),xw(3,kk),xu(3,kk),xv(4,kk)],[yv(4,kk),yu(4,kk),yw(4,kk
),yw(3,kk),yu(3,kk),yv(4,kk)],'b')
plot([xv(5,kk),xu(5,kk),xw(5,kk),xw(4,kk),xu(4,kk),xv(5,kk)],[yv(5,kk),yu(5,kk),yw(5,kk
),yw(4,kk),yu(4,kk),yv(5,kk)],'b')
plot([xw(1,kk),xw(2,kk),xw(3,kk),xw(4,kk),xw(5,kk),xw(1,kk)],[yw(1,kk),yw(2,kk),yw(3,kk
),yw(4,kk),yw(5,kk),yw(1,kk)],'r')
```

```
end%for kk=2:7
nv(6,1)=nv(1,1)
%initial pentagon as closed domain has vertex nodes
[nv(1:6,1)]

[nvv,nuu,nww,elm]=nodaladdressesforpentagon(2)


%

kk=1
%kkk=max(max(elm))

for j=2:7




%-------------------------------------------------------------------------

for i=1:5
    text(xv(i,j),yv(i,j),['o',num2str(nvv(i,j-1))])
    text(xu(i,j),yu(i,j),['o',num2str(nuu(i,j-1))])
    text(xw(i,j),yw(i,j),['o',num2str(nww(i,j-1))])
end



%=========================================================================================
==============
pxvv(kk,1)=(xv(1,j)+xu(1,j)+xw(1,j)+xw(5,j)+xu(5,j))/5;pyvv(kk,1)=(yv(1,j)+yu(1,j)+yw(1
,j)+yw(5,j)+yu(5,j))/5;
pxvv(kk+1,1)=(xv(2,j)+xu(2,j)+xw(2,j)+xw(1,j)+xu(1,j))/5;pyvv(kk+1,1)=(yv(2,j)+yu(2,j)+
yw(2,j)+yw(1,j)+yu(1,j))/5;
pxvv(kk+2,1)=(xv(3,j)+xu(3,j)+xw(3,j)+xw(2,j)+xu(2,j))/5;pyvv(kk+2,1)=(yv(3,j)+yu(3,j)+
yw(3,j)+yw(2,j)+yu(2,j))/5;
pxvv(kk+3,1)=(xv(4,j)+xu(4,j)+xw(4,j)+xw(3,j)+xu(3,j))/5;pyvv(kk+3,1)=(yv(4,j)+yu(4,j)+
yw(4,j)+yw(3,j)+yu(3,j))/5;
pxvv(kk+4,1)=(xv(5,j)+xu(5,j)+xw(5,j)+xw(4,j)+xu(4,j))/5;pyvv(kk+4,1)=(yv(5,j)+yu(5,j)+
yw(5,j)+yw(4,j)+yu(4,j))/5;
pxvv(kk+5,1)=(xw(1,j)+xw(2,j)+xw(3,j)+xw(4,j)+xw(5,j))/5;pyvv(kk+5,1)=(yw(1,j)+yw(2,j)+
yw(3,j)+yw(4,j)+yw(5,j))/5;
%

kk=kk+6
end%(for j==2:7)

%hold on
[nel,nnel]=size(elm)
%hold on
%xlabel('x axis')
%ylabel('y axis')
%st1='FEM Mesh With ';
%st2=num2str(nel);
%st3='  Irregular';
%st4=' Pentagonal';
%st5=' Elements'
%st6='& No. of Nodes=  '
%st7=num2str(nnode);
```

```
%title([st1,st2,st3,st4,st5,st6,st7])


hold on


%make further refinement of 36-element mesh
for kk=2:7
pxv(kk,1)=(xv(1,kk)+xv(2,kk)+xv(3,kk)+xv(4,kk)+xv(5,kk))/5;
pyv(kk,1)=(yv(1,kk)+yv(2,kk)+yv(3,kk)+yv(4,kk)+yv(5,kk))/5;

for i=1:5
xu(i,kk)=(xv(i,kk)+xv(i+1,kk))/2;yu(i,kk)=(yv(i,kk)+yv(i+1,kk))/2;
xw(i,kk)=(xu(i,kk)+pxv(kk,1))/2;yw(i,kk)=(yu(i,kk)+pyv(kk,1))/2;
end
%divide each pentagon into six new pentagons
% 1st to 5th-pentagons adjacent to the perimeter are colored with blue lines
%6th pentagon is colored with red lines
plot([xv(1,kk),xu(1,kk),xw(1,kk),xw(5,kk),xu(5,kk),xv(1,kk)],[yv(1,kk),yu(1,kk),yw(1,kk
),yw(5,kk),yu(5,kk),yv(1,kk)],'b')
plot([xv(2,kk),xu(2,kk),xw(2,kk),xw(1,kk),xu(1,kk),xv(2,kk)],[yv(2,kk),yu(2,kk),yw(2,kk
),yw(1,kk),yu(1,kk),yv(2,kk)],'b')
plot([xv(3,kk),xu(3,kk),xw(3,kk),xw(2,kk),xu(2,kk),xv(3,kk)],[yv(3,kk),yu(3,kk),yw(3,kk
),yw(2,kk),yu(2,kk),yv(3,kk)],'b')
plot([xv(4,kk),xu(4,kk),xw(4,kk),xw(3,kk),xu(3,kk),xv(4,kk)],[yv(4,kk),yu(4,kk),yw(4,kk
),yw(3,kk),yu(3,kk),yv(4,kk)],'b')
plot([xv(5,kk),xu(5,kk),xw(5,kk),xw(4,kk),xu(4,kk),xv(5,kk)],[yv(5,kk),yu(5,kk),yw(5,kk
),yw(4,kk),yu(4,kk),yv(5,kk)],'b')
plot([xw(1,kk),xw(2,kk),xw(3,kk),xw(4,kk),xw(5,kk),xw(1,kk)],[yw(1,kk),yw(2,kk),yw(3,kk
),yw(4,kk),yw(5,kk),yw(1,kk)],'r')


hold on

end%for kk=2:7
hold on
%end%switch iteration
%return
hold on
%if iteration==3
    mm=8;
for kk=2:7

xv(1:6,mm)=[xv(1,kk),xu(1,kk),xw(1,kk),xw(5,kk),xu(5,kk),xv(1,kk)]';yv(1:6,mm)=[yv(1,kk
),yu(1,kk),yw(1,kk),yw(5,kk),yu(5,kk),yv(1,kk)]';
xv(1:6,mm+1)=[xv(2,kk),xu(2,kk),xw(2,kk),xw(1,kk),xu(1,kk),xv(2,kk)]';yv(1:6,mm+1)=[yv(
2,kk),yu(2,kk),yw(2,kk),yw(1,kk),yu(1,kk),yv(2,kk)]';
xv(1:6,mm+2)=[xv(3,kk),xu(3,kk),xw(3,kk),xw(2,kk),xu(2,kk),xv(3,kk)]';yv(1:6,mm+2)=[yv(
3,kk),yu(3,kk),yw(3,kk),yw(2,kk),yu(2,kk),yv(3,kk)]';
xv(1:6,mm+3)=[xv(4,kk),xu(4,kk),xw(4,kk),xw(3,kk),xu(3,kk),xv(4,kk)]';yv(1:6,mm+3)=[yv(
4,kk),yu(4,kk),yw(4,kk),yw(3,kk),yu(3,kk),yv(4,kk)]';
xv(1:6,mm+4)=[xv(5,kk),xu(5,kk),xw(5,kk),xw(4,kk),xu(4,kk),xv(5,kk)]';yv(1:6,mm+4)=[yv(
5,kk),yu(5,kk),yw(5,kk),yw(4,kk),yu(4,kk),yv(5,kk)]';
xv(1:6,mm+5)=[xw(1,kk),xw(2,kk),xw(3,kk),xw(4,kk),xw(5,kk),xw(1,kk)]';yv(1:6,mm+5)=[yw(
1,kk),yw(2,kk),yw(3,kk),yw(4,kk),yw(5,kk),yw(1,kk)]';
for nn=mm:mm+5
pppp=0;qqqq=0;
for i=1:5
pppp=pppp+xv(i,nn);qqqq=qqqq+yv(i,nn);
end
```

```
pxvv(nn,1)=pppp/5;pyvv(nn,1)=qqqq/5;
for i=1:5
xu(i,nn)=(xv(i,nn)+xv(i+1,nn))/2;yu(i,nn)=(yv(i,nn)+yv(i+1,nn))/2;
xw(i,nn)=(xu(i,nn)+pxvv(nn,1))/2;yw(i,nn)=(yu(i,nn)+pyvv(nn,1))/2;
end

plot([xv(1,nn),xu(1,nn),xw(1,nn),xw(5,nn),xu(5,nn),xv(1,nn)],[yv(1,nn),yu(1,nn),yw(1,nn
),yw(5,nn),yu(5,nn),yv(1,nn)],'b')
plot([xv(2,nn),xu(2,nn),xw(2,nn),xw(1,nn),xu(1,nn),xv(2,nn)],[yv(2,nn),yu(2,nn),yw(2,nn
),yw(1,nn),yu(1,nn),yv(2,nn)],'b')
plot([xv(3,nn),xu(3,nn),xw(3,nn),xw(2,nn),xu(2,nn),xv(3,nn)],[yv(3,nn),yu(3,nn),yw(3,nn
),yw(2,nn),yu(2,nn),yv(3,nn)],'b')
plot([xv(4,nn),xu(4,nn),xw(4,nn),xw(3,nn),xu(3,nn),xv(4,nn)],[yv(4,nn),yu(4,nn),yw(4,nn
),yw(3,nn),yu(3,nn),yv(4,nn)],'b')
plot([xv(5,nn),xu(5,nn),xw(5,nn),xw(4,nn),xu(4,nn),xv(5,nn)],[yv(5,nn),yu(5,nn),yw(5,nn
),yw(4,nn),yu(4,nn),yv(5,nn)],'b')
plot([xw(1,nn),xw(2,nn),xw(3,nn),xw(4,nn),xw(5,nn),xw(1,nn)],[yw(1,nn),yw(2,nn),yw(3,nn
),yw(4,nn),yw(5,nn),yw(1,nn)],'r')
end%for nn=mm:(mm+5)
mm=mm+6;
end%for kk=2:7


%-----------------------------------
[nvv,nuu,nww,elm]=nodaladdressesforpentagon(3)



kk=1


for j=8:43

%------------------------------------------------------------------------

for i=1:5
    text(xv(i,j),yv(i,j),['o',num2str(nvv(i,j-7))])
    text(xu(i,j),yu(i,j),['o',num2str(nuu(i,j-7))])
    text(xw(i,j),yw(i,j),['o',num2str(nww(i,j-7))])
end

pxvv(kk,1)=(xv(1,j)+xu(1,j)+xw(1,j)+xw(5,j)+xu(5,j))/5;pyvv(kk,1)=(yv(1,j)+yu(1,j)+yw(1
,j)+yw(5,j)+yu(5,j))/5;
pxvv(kk+1,1)=(xv(2,j)+xu(2,j)+xw(2,j)+xw(1,j)+xu(1,j))/5;pyvv(kk+1,1)=(yv(2,j)+yu(2,j)+
yw(2,j)+yw(1,j)+yu(1,j))/5;
pxvv(kk+2,1)=(xv(3,j)+xu(3,j)+xw(3,j)+xw(2,j)+xu(2,j))/5;pyvv(kk+2,1)=(yv(3,j)+yu(3,j)+
yw(3,j)+yw(2,j)+yu(2,j))/5;
pxvv(kk+3,1)=(xv(4,j)+xu(4,j)+xw(4,j)+xw(3,j)+xu(3,j))/5;pyvv(kk+3,1)=(yv(4,j)+yu(4,j)+
yw(4,j)+yw(3,j)+yu(3,j))/5;
pxvv(kk+4,1)=(xv(5,j)+xu(5,j)+xw(5,j)+xw(4,j)+xu(4,j))/5;pyvv(kk+4,1)=(yv(5,j)+yu(5,j)+
yw(5,j)+yw(4,j)+yu(4,j))/5;
pxvv(kk+5,1)=(xw(1,j)+xw(2,j)+xw(3,j)+xw(4,j)+xw(5,j))/5;pyvv(kk+5,1)=(yw(1,j)+yw(2,j)+
yw(3,j)+yw(4,j)+yw(5,j))/5;
%


%
%text(pxvv(kk,1),pyvv(kk,1),['[',num2str(kk),']'])
%text(pxvv(kk+1,1),pyvv(kk+1,1),['[',num2str(kk+1),']'])
%text(pxvv(kk+2,1),pyvv(kk+2,1),['[',num2str(kk+2),']'])
%text(pxvv(kk+3,1),pyvv(kk+3,1),['[',num2str(kk+3),']'])
```

```
%text(pxvv(kk+4,1),pyvv(kk+4,1),['[',num2str(kk+4),']'])
%text(pxvv(kk+5,1),pyvv(kk+5,1),['[',num2str(kk+5),']'])



%=========================================================================================
===========================
kk=kk+6
end%(for j==8:43)

hold on
mm=44;
for kk=8:43

xv(1:6,mm)=[xv(1,kk),xu(1,kk),xw(1,kk),xw(5,kk),xu(5,kk),xv(1,kk)]';yv(1:6,mm)=[yv(1,kk
),yu(1,kk),yw(1,kk),yw(5,kk),yu(5,kk),yv(1,kk)]';
xv(1:6,mm+1)=[xv(2,kk),xu(2,kk),xw(2,kk),xw(1,kk),xu(1,kk),xv(2,kk)]';yv(1:6,mm+1)=[yv(
2,kk),yu(2,kk),yw(2,kk),yw(1,kk),yu(1,kk),yv(2,kk)]';
xv(1:6,mm+2)=[xv(3,kk),xu(3,kk),xw(3,kk),xw(2,kk),xu(2,kk),xv(3,kk)]';yv(1:6,mm+2)=[yv(
3,kk),yu(3,kk),yw(3,kk),yw(2,kk),yu(2,kk),yv(3,kk)]';
xv(1:6,mm+3)=[xv(4,kk),xu(4,kk),xw(4,kk),xw(3,kk),xu(3,kk),xv(4,kk)]';yv(1:6,mm+3)=[yv(
4,kk),yu(4,kk),yw(4,kk),yw(3,kk),yu(3,kk),yv(4,kk)]';
xv(1:6,mm+4)=[xv(5,kk),xu(5,kk),xw(5,kk),xw(4,kk),xu(4,kk),xv(5,kk)]';yv(1:6,mm+4)=[yv(
5,kk),yu(5,kk),yw(5,kk),yw(4,kk),yu(4,kk),yv(5,kk)]';
xv(1:6,mm+5)=[xw(1,kk),xw(2,kk),xw(3,kk),xw(4,kk),xw(5,kk),xw(1,kk)]';yv(1:6,mm+5)=[yw(
1,kk),yw(2,kk),yw(3,kk),yw(4,kk),yw(5,kk),yw(1,kk)]';
for nn=mm:mm+5
pppp=0;qqqq=0;
for i=1:5
pppp=pppp+xv(i,nn);qqqq=qqqq+yv(i,nn);
end
pxvv(nn,1)=pppp/5;pyvv(nn,1)=qqqq/5;
for i=1:5
xu(i,nn)=(xv(i,nn)+xv(i+1,nn))/2;yu(i,nn)=(yv(i,nn)+yv(i+1,nn))/2;
xw(i,nn)=(xu(i,nn)+pxvv(nn,1))/2;yw(i,nn)=(yu(i,nn)+pyvv(nn,1))/2;
end

plot([xv(1,nn),xu(1,nn),xw(1,nn),xw(5,nn),xu(5,nn),xv(1,nn)],[yv(1,nn),yu(1,nn),yw(1,nn
),yw(5,nn),yu(5,nn),yv(1,nn)],'b')
plot([xv(2,nn),xu(2,nn),xw(2,nn),xw(1,nn),xu(1,nn),xv(2,nn)],[yv(2,nn),yu(2,nn),yw(2,nn
),yw(1,nn),yu(1,nn),yv(2,nn)],'b')
plot([xv(3,nn),xu(3,nn),xw(3,nn),xw(2,nn),xu(2,nn),xv(3,nn)],[yv(3,nn),yu(3,nn),yw(3,nn
),yw(2,nn),yu(2,nn),yv(3,nn)],'b')
plot([xv(4,nn),xu(4,nn),xw(4,nn),xw(3,nn),xu(3,nn),xv(4,nn)],[yv(4,nn),yu(4,nn),yw(4,nn
),yw(3,nn),yu(3,nn),yv(4,nn)],'b')
plot([xv(5,nn),xu(5,nn),xw(5,nn),xw(4,nn),xu(4,nn),xv(5,nn)],[yv(5,nn),yu(5,nn),yw(5,nn
),yw(4,nn),yu(4,nn),yv(5,nn)],'b')
plot([xw(1,nn),xw(2,nn),xw(3,nn),xw(4,nn),xw(5,nn),xw(1,nn)],[yw(1,nn),yw(2,nn),yw(3,nn
),yw(4,nn),yw(5,nn),yw(1,nn)],'r')
end%for nn=mm:(mm+5)
mm=mm+6;
end%for kk=8:43

[nvv,nuu,nww,elm]=nodaladdressesforpentagon(4)



kk=1
```

```
for j=44:259

%for i=1:5
 %   text(xv(i,j),yv(i,j),['o',num2str(nvv(i,j-43))])
  %  text(xu(i,j),yu(i,j),['o',num2str(nuu(i,j-43))])
   % text(xw(i,j),yw(i,j),['o',num2str(nww(i,j-43))])
%end
for i=1:5
   text(xv(i,j),yv(i,j),['o'])
  text(xu(i,j),yu(i,j),['o'])
 text(xw(i,j),yw(i,j),['o'])
end

pxvv(kk,1)=(xv(1,j)+xu(1,j)+xw(1,j)+xw(5,j)+xu(5,j))/5;pyvv(kk,1)=(yv(1,j)+yu(1,j)+yw(1
,j)+yw(5,j)+yu(5,j))/5;
pxvv(kk+1,1)=(xv(2,j)+xu(2,j)+xw(2,j)+xw(1,j)+xu(1,j))/5;pyvv(kk+1,1)=(yv(2,j)+yu(2,j)+
yw(2,j)+yw(1,j)+yu(1,j))/5;
pxvv(kk+2,1)=(xv(3,j)+xu(3,j)+xw(3,j)+xw(2,j)+xu(2,j))/5;pyvv(kk+2,1)=(yv(3,j)+yu(3,j)+
yw(3,j)+yw(2,j)+yu(2,j))/5;
pxvv(kk+3,1)=(xv(4,j)+xu(4,j)+xw(4,j)+xw(3,j)+xu(3,j))/5;pyvv(kk+3,1)=(yv(4,j)+yu(4,j)+
yw(4,j)+yw(3,j)+yu(3,j))/5;
pxvv(kk+4,1)=(xv(5,j)+xu(5,j)+xw(5,j)+xw(4,j)+xu(4,j))/5;pyvv(kk+4,1)=(yv(5,j)+yu(5,j)+
yw(5,j)+yw(4,j)+yu(4,j))/5;
pxvv(kk+5,1)=(xw(1,j)+xw(2,j)+xw(3,j)+xw(4,j)+xw(5,j))/5;pyvv(kk+5,1)=(yw(1,j)+yw(2,j)+
yw(3,j)+yw(4,j)+yw(5,j))/5;
%

%
%text(pxvv(kk,1),pyvv(kk,1),['[',num2str(kk),']'])
%text(pxvv(kk+1,1),pyvv(kk+1,1),['[',num2str(kk+1),']'])
%text(pxvv(kk+2,1),pyvv(kk+2,1),['[',num2str(kk+2),']'])
%text(pxvv(kk+3,1),pyvv(kk+3,1),['[',num2str(kk+3),']'])
%text(pxvv(kk+4,1),pyvv(kk+4,1),['[',num2str(kk+4),']'])
%text(pxvv(kk+5,1),pyvv(kk+5,1),['[',num2str(kk+5),']'])



%=============================================================================================
============================
kk=kk+6
end%(for j==44:259)


nnode=max(max(elm))
[nnel,nel]=size(elm)
hold on
xlabel('x axis')
ylabel('y axis')
st1='FEM Mesh With ';
st2=num2str(nel);
st3='  Irregular';
st4=' Pentagonal';
st5=' Elements'
st6='& No. of Nodes=  '
st7=num2str(nnode);
title([st1,st2,st3,st4,st5,st6,st7])
[(1:nel)' elm']
ELM=elm';
hold on
```

```
for qq=44:259
for pp=1:5
xcord(nvv(pp,qq-43),1)=xv(pp,qq);ycord(nvv(pp,qq-43),1)=yv(pp,qq);
xcord(nuu(pp,qq-43),1)=xu(pp,qq);ycord(nuu(pp,qq-43),1)=yu(pp,qq);
xcord(nww(pp,qq-43),1)=xw(pp,qq);ycord(nww(pp,qq-43),1)=yw(pp,qq);
end
end

[sn,sm]=size(xcord)
disp('---------------------------------------')
disp('xcoord                        ycoord')
[(1:sn)' xcord ycord]
 if iteration==4
     return
 end
 end%switch
 %-------------------------------------------------------------
```

%********FIFTH COMPUTER PROGRAM*****************

```
function[]=compositeboundaryintegrationconvexpolygonofpentagons_n(ns,ng,fn,xv,yv)
syms t x y rs
  syms c1 c2  m
  syms p q  m
exact=0
%coordinates for the vertices of polygon
%1:convex polygon 6-sides
%X=[0.1;0.7;1;0.75;0.5;0;0.1];Y=[0;0.2;0.5;0.85;1;0.25;0];
%2  nonconvex polygon 9-sides
%X=[0.25;0.75;0.75;1.0;0.75;0.75;0.5;0;0.25;0.25];
%  1  2  3  4   5    6   7   8   9  10
%Y=[0;0.5;0;0.5;0.75;0.85;1;0.75;0.5;0];
%maximum value of ng =47,this is very important
%X=[0.25;0.75;0.75;1.0;0.75;0.75;0.5;0;0.25;0.25];
%  1  2  3  4   5    6   7   8   9  10
%Y=[0;0.5;0;0.5;0.75;0.85;1;0.75;0.5;0];
%---------------------------------------
%X=[-1;1;2;0;-2;-1]
%Y=[-1;-1;0;1;0;-1]
%FOR PENTAGON ns=5
%ng=10
%boundaryintegrationconvexpolygon_n(ns=5,ng=10,fn=100,X=[-1;1;2;0;-2;-1],Y=[-1;-
1;0;1;0;-1])
%boundaryintegrationconvexpolygon_n(5,10,100,[-1;1;2;0;-2;-1],[-1;-1;0;1;0;-1])
%boundaryintegrationconvexpolygon_n(5,10,101,[-1;1;2;0;-2;-1],[-1;-1;0;1;0;-1])
%compositeboundaryintegrationconvexpolygon_n(ns=5,ng=10,fn=100,X=[-1;1;2;0;-2;0;1.5;1;-
1;-1.5;0;0.75;0.5;-0.5;-0.75],Y=[-1;-1;0;1;0;-1;-0.5;0.5;0.5;-0.5;-0.6;-
0.35;0.15;0.15;-0.35])
%---------------------------------------------------------------------
%   1  2 3 4  5 6  7  8  9  10 11  12   13   14   15
%X=[-1;1;2;0;-2;0;1.5;1;-1;-1.5;0;0.75;0.5;-0.5;-0.75]
%    1  2 3 4 5  6    7   8   9   10   11    12   13   14    15
%Y=[-1;-1;0;1;0;-1;-0.5;0.5;0.5;-0.5;-0.6;-0.35;0.15;0.15;-0.35]
%---------------------------------------------------------------------
%compositeboundaryintegrationconvexpolygonofpentagons_n(ns=5,ng=10,fn=100,xv=[-
1;1;2;0;-2;-1],yv=[-1;-1;0;1;0;-1],iteration=1)
%compositeboundaryintegrationconvexpolygonofpentagons_n(5,10,100,[-1;1;2;0;-2;-1],[-1;-
1;0;1;0;-1],1)
%-------------------------------------------------------------
%X=[0;2;3;1;0;0]
%Y=[0;0;1;3;2;0]
```

```
%compositeboundaryintegrationconvexpolygonofpentagons_n(5,10,102,[0;2;3;1;0;0],[0;0;1;3
;2;0],1)
format long e
for iteration=1:4
[elm,X,Y]=pentagonalmeshgeneratorforfemiterations(xv,yv,iteration)
[nel,nnel]=size(elm);
for iel=1:nel
    elm(iel,6)=elm(iel,1);
end
[nel,nnel]=size(elm)
[(1:nel)' elm]
%elm(1:6,1:6)=[ 1  6 11 15 10  1;...
 %             2  7 12 11  6  2;...
  %            3  8 13 12  7  3;...
   %           4  9 14 13  8  4;...
    %          5 10 15 14  9  5;...
     %        11 12 13 14 15 11]
[nel,nnel]=size(elm);
%determine element centroid and element coordinates
for iel=1:nel
    xc(iel,1)=0; yc(iel,1)=0;
end
for iel=1:nel
    for k=1:nnel-1
    ielk=elm(iel,k);
    xc(iel,1)=xc(iel,1)+X(ielk,1);
    yc(iel,1)=yc(iel,1)+Y(ielk,1);
    end
     xc(iel,1)= xc(iel,1)/5;yc(iel,1)= yc(iel,1)/5;

end
switch fn
    case 16
     f=(x+y)^19
    case 17
    f=cos(30*(x+y))
case 18
    f=sqrt((x-0.5)^2+(y-0.5)^2)
case 19
    f=exp(-((x-0.5)^2+(y-0.5)^2))
case 20
    f=exp(-100*((x-0.5)^2+(y-0.5)^2))
case 21
         f1=0.75*exp(-0.25*(9*x-2)^2-0.25*(9*y-2)^2)
         f2=0.75*exp((-1/49)*(9*x+1)^2-0.1*(9*y+1))
         f3=0.5*exp(-0.25*(9*x-7)^2-0.25*(9*y-3)^2)
         f4=-0.2*exp(-(9*y-4)^2-(9*y-7)^2)
         f=f1+f2+f3+f4
case 100
        f=(x^4+y^3)/(1+x^2)
        if (xv==[0;2;3;1;0;0]& yv==[0;0;1;3;2;0])
        exact=vpa(20.48256361468692452827984674976 5,32)

        end

        if( xv==[-1;1;2;0;-2;-1]&yv==[-1;-1;0;1;0;-1])
        exact=vpa(1.92403054263265012157514536962 99,32)
        end
case 101
      f=(1-x)*sin(10*x*y)
```

```
       if( xv==[-1;1;2;0;-2;-1]&yv==[-1;-1;0;1;0;-1])
      exact=vpa(-0.013103719669957,16)
       end
 case 102
    f=(.2*x+.3*y)^19
    if (xv==[0;2;3;1;0;0]& yv==[0;0;1;3;2;0])
    syms c1 c2  m
    c1=sym('0.2');c2=sym('0.3'); m=sym('19');
    exact=( (-2*(c1+3*c2)^(m+2))/((c1+c2)*(c1-c2))  + (2*(3*c1+c2)^(m+2))/((c1+c2)*(c1-
c2))  - (2*c1)^(m+2)/( c1*(c1+c2)) - (2*c2)^(m+2)/( c2*(c1+c2)) )/((m+1)*(m+2));
    exact=vpa(exact,32);
    end
    if( xv==[-1;1;2;0;-2;-1]&yv==[-1;-1;0;1;0;-1])
        syms p q  m
        p=sym('0.2');q=sym('0.3'); m=sym('19');
    MM=2/((m+1)*(m+2))
    exact1=(((-1)^m)*(p+q)^(m+1)+ (p-q)^(m+1))/(2*p);
    exact2=(-(p-q)^(m+1)+(2*p)^(m+1))/(p+q);
    exact3=((2*p)^(m+1)-q^(m+1))/(2*p-q);
    exact4=( q^(m+1)+((-1)^m)*(2*p)^(m+1) )/(2*p+q);
    exact5=((-1)^m)*( (2*p)^(m+1)-(p+q)^(m+1) )/(p-q);
    exact=MM*(exact1+exact2+exact3+exact4+exact5);
    exact=vpa(exact,32);
    end



    case 103

    f=((.17*x+.25*y)^25)
     if (xv==[0;2;3;1;0;0]& yv==[0;0;1;3;2;0])
    c1=sym('0.17');c2=sym('.25'); m=sym('25');
    exact=( (-2*(c1+3*c2)^(m+2))/((c1+c2)*(c1-c2))  + (2*(3*c1+c2)^(m+2))/((c1+c2)*(c1-
c2))  - (2*c1)^(m+2)/( c1*(c1+c2)) - (2*c2)^(m+2)/( c2*(c1+c2)) )/((m+1)*(m+2));
     end
     if( xv==[-1;1;2;0;-2;-1]&yv==[-1;-1;0;1;0;-1])
        p=sym('0.17');q=sym('.25'); m=sym('25');
    MM=2/((m+1)*(m+2));
    exact1=(((-1)^m)*(p+q)^(m+1)+ (p-q)^(m+1))/(2*p);
    exact2=(-(p-q)^(m+1)+(2*p)^(m+1))/(p+q);
    exact3=((2*p)^(m+1)-q^(m+1))/(2*p-q);
    exact4=( q^(m+1)+((-1)^m)*(2*p)^(m+1) )/(2*p+q);
    exact5=((-1)^m)*( (2*p)^(m+1)-(p+q)^(m+1) )/(p-q);
    exact=MM*(exact1+exact2+exact3+exact4+exact5)
   end


  case 104
    f=(x+y)^19/10^10
    if (xv==[0;2;3;1;0;0]& yv==[0;0;1;3;2;0])
    m=sym('19');
    exact=((2^(m+1)-(1/(m+2)))/(m+1))*2^(m+2);
    exact=exact/10^10;
    end
    if( xv==[-1;1;2;0;-2;-1]&yv==[-1;-1;0;1;0;-1])
     m=sym('19');M1=2/(m+2);
     M0=2/((m+1)*(m+2));
    ex1=(2^m)*(1+(-1)^m);
    ex2=(2^(m+1)-1);
    ex3=(1-(-2)^(m+1))/3;
    ex4=(-2)^m;
```

```
 exact=M0*(ex1+ex2+ex3)+M1*ex4;
  exact=exact/10^10;
 end
 case 105
     f=(x-y)^20/10^5
    if (xv==[0;2;3;1;0;0]& yv==[0;0;1;3;2;0])
     m=sym('20');
     exact=0;
     if mod(m,2)==0
         exact=(2^(m+2)/(m+1))*(1/(m+2)+1);
         exact=exact/10^5;
     end
    end
     if( xv==[-1;1;2;0;-2;-1]&yv==[-1;-1;0;1;0;-1])
      m=sym('20');
      M0=2*((-1)^m)/((m+1)*(m+2));M1=2/(m+2);
      exact=M0*( (2^m)*(1+(-1)^m)-2/3+((-1)^m)*(2^(m+1))/3+2^(m+1) )+M1*2^(m);
      exact=exact/10^5;
     end
 case 106
 f=cos(30*(x+y))
 if (xv==[0;2;3;1;0;0]& yv==[0;0;1;3;2;0])
  exact=vpa(sym(' 0.036538064524804112293611778879387 '))
 end

 if( xv==[-1;1;2;0;-2;-1]&yv==[-1;-1;0;1;0;-1])
 exact=vpa(sym('-0.007436177299024373219858448899496 4'))
 end
 case 107
    f=sqrt((x-0.5)^2+(y-0.5)^2)
    if (xv==[0;2;3;1;0;0]& yv==[0;0;1;3;2;0])
  exact=vpa(sym('7.800618298021123954289758058444 '))
 end

 if( xv==[-1;1;2;0;-2;-1]&yv==[-1;-1;0;1;0;-1])
 exact=vpa(sym('5.809534594898993711537644374406 4 '))
 end
 case 108
    f=exp(-((x-1/2)^2+(y-1/2)^2));

if (xv==[0;2;3;1;0;0]& yv==[0;0;1;3;2;0])
  exact=vpa(sym('1.798096146581152670802198439191 6 '))
 end

 if( xv==[-1;1;2;0;-2;-1]&yv==[-1;-1;0;1;0;-1])
 exact=vpa(sym('1.729175291842216445410886742214 9 '))
 end
case 109
    f=exp(-100*((x-1/2)^2+(y-1/2)^2));

if (xv==[0;2;3;1;0;0]& yv==[0;0;1;3;2;0])
  exact=vpa(sym('0.031415926535849631660672804517763 '))
 end

 if( xv==[-1;1;2;0;-2;-1]&yv==[-1;-1;0;1;0;-1])
 exact=vpa(sym('0.031391337254729663963679685981861 '))
 end

 case 110
      f1=0.75*exp(-0.25*(9*x-2)^2-0.25*(9*y-2)^2)
      f2=0.75*exp((-1/49)*(9*x+1)^2-0.1*(9*y+1))
```

```
          f3=0.5*exp(-0.25*(9*x-7)^2-0.25*(9*y-3)^2)
          f4=-0.2*exp(-(9*y-4)^2-(9*y-7)^2)
          f=f1+f2+f3+f4

   if (xv==[0;2;3;1;0;0]& yv==[0;0;1;3;2;0])
     exact=vpa(sym(' 0.558035127601503928491469366591 77 '))
    end

    if( xv==[-1;1;2;0;-2;-1]&yv==[-1;-1;0;1;0;-1])
    exact=vpa(sym('2.190018384318485748596900210234 5 '))
    end


end

     ff=f;
    cc=0;
 for n=5:5:ng
     cc=cc+1;
    iii(cc,iteration)=0;
    np0(cc,1)=n;
    [ss,ww]=glsampleptsweights(n);
    [sss,www]=glsampleptsweights(n+1);
    for iel=1:nel
    ii(cc,iel)=0;
    xcc=xc(iel,1);
    ycc=yc(iel,1);
 for N=1:ns
        II=0;k=0;
        ielN=elm(iel,N);ielN1=elm(iel,N+1);
        xi=X(ielN1,1);yi=Y(ielN1,1);xk=X(ielN,1);yk=Y(ielN,1);
       d=(xcc-xi)*(yk-yi)-(xk-xi)*(ycc-yi);
      if d~=0
        for jj=1:(n+1)
           for kk=1:n
               k=k+1;
               ps=(1+sss(jj))/2;qs=(1-sss(jj))/2;
               t=ss(kk);wt=ww(kk)*www(jj)/(4);
               xs=xcc*qs+ps*((xi+xk)/2+(xk-xi)*t/2);ys=ycc*qs+ps*((yi+yk)/2+(yk-
yi)*t/2);
                fff=ps*wt*fnxy(fn,xs,ys);
                II=II+fff;
            end
        end
      end%if
      ii(cc,iel)=ii(cc,iel)+II*d;
 end%for N=1:ns
  iii(cc,iteration)=iii(cc,iteration)+ii(cc,iel);
 end%for iel=1:nel
iii(cc,iteration)=double(iii(cc,iteration));
end%for n=1:ng
iii(:,iteration)%%
%table8(:,1)=np0;
table8(:,iteration)=iii(:,iteration);
end%for iteration
%table8(:,1)=np0;
%table8(:,iteration+1)=iii(:,iteration);
disp('----------------------NUMERICAL  VALUES   FOR THE   INTEGRAL-------------------
----------------------')
disp(f)
```

```
disp('    iteration=1                    iteration=2                iteration=3
iteration=4     ')
disp(table8)
disp('-------------------------------------------------------------------------------
------------------------')
if exact~=0
disp('exact value=')
disp(exact)
end
%
```

9.0 Conclusions

An automatic recursive pentagonal mesh generator technique is presented for the two dimensional convex polygonal domains. This mesh generation is made fully automatic and allows the user to define the problem domain with minimum amount of input such as coordinates of boundary. Once this input is created, by selecting an appropriate interior point of the convex polygonal domain, we form the pentagonal subdomains.It is shown in literature that one cannot refine a hexagon using hexagons of smaller size. In general, one can only refine an n-gon by n-gons of smaller size if $n \leq 5$. Furthermore, we introduce a refinement scheme of a general polygon based on the pentagon scheme. This paper first presents a pentagonalization (or pentagonal conversion) scheme that can create a pentagonal mesh from any arbitrary mesh structure. We also introduce a pentagonal preservation scheme that can create a pentagonal mesh from any pentagonal mesh. This paper then presents a new numerical integration technique proposed earlier by the first author and co-workers, known as boundary integration method [34-40] is now applied to arbitrary polygonal domains using pentagonal finite element mesh. Numerical results presented is tested on examples of complicated integrals over convex polygons in the context of pentagonal domains with composite numerical integration scheme of triangular finite elements which can be easily created by joining the centre point of pentagons,this shows that the proposed method yields accurate results even for low order Gauss Legendre Quadrature rules. Our numerical results suggest that the **refinement scheme for pentagons and polygons** may lead to higher accuracy than the **uniform refinement of triangulations and quadrangulations.**

We have also appended MATLAB programs which provide the nodal coordinates, element nodal connectivity and graphic display of the generated all pentagonal mesh for the pentagon and the complex polygonal domains. We believe that this work will be useful for various applications in science and engineering.