RAY BASED OFFSETTING TECHNIQUE FOR DISCRETISATION OF PLANER DOMAIN

[1] Shalendra Kumar & [2]Satyajai Kumar

[1]Professor of Mechanical Engineering, NIT Jamshedpur (INDIA)
[2]Executive Engineer, D.V.C.,Patratu, Jharkhand(INDIA)

ABSTRACT

The paper deals with developing an algorithm, based on "Ray Based Offsetting" to generate triangular as well as quadrilateral mesh. The method begins by decomposing the region to be meshed into suitable sub-regions. A series of horizontal lines are introduced in each sub-region. The intersection points between domain boundary and rays (boundary node) are offset along the ray with suitable spacing (element size) for generating nodes on the ray. The entire sub-region gets segmented by the series of horizontal rays. By taking generated nodes on the rays as input, element generation of entire sub region is achieved by meshing one segment at a time and placing them one above other from bottom to top of the sub region. The resulting meshes of various sub regions are merged together to form the final mesh. The input data of the method include region boundary curve, element size and mesh refinement information.

The Proposed technique is capable of meshing any typical geometry (simple as well as multiply connected complex regions). The meshing of the surfaces has been compared with other popular algorithms on the basis of key parameters i.e., element shape, no of element/nodes generated, processing time, etc. and the results are in good agreement.

1. INTRODUCTION

The finite element method has become a powerful tool for the numerical solution of a wide range of engineering problems and applications ranging from deformation and stress analysis of automotive, aircraft, building and bridge structure to field analysis of heat flux, fluid flow, magnetic flux, seepage, and other flow problems.

The stress analysis in the fields of civil, mechanical, aero-space engineering, naval architecture, off-shore engineering and nuclear engineering is invisibly complex and for many of the problems it is extremely difficult and tedious to obtain analytical solution. In these situations engineers usually resort to numerical methods to solve the problems. With the advent of computers**,** one of the most powerful techniques that have been developed in the realm of engineering analysis is the finite clement method, and the method, being general can be used for the analysis of structure / solids of complex shapes and complicated boundary conditions with relative ease. The accuracy of the numerical solution depends upon the accuracy of the geometry meshing.

2. MESH GENERATION BY RAY BASED OFFSETTING TECHNIQUE

Various methods of mesh generation are in existence. The most commonly used techniques for 2D mesh generation are mapping, topology/geometric decomposition and triangulation methods.

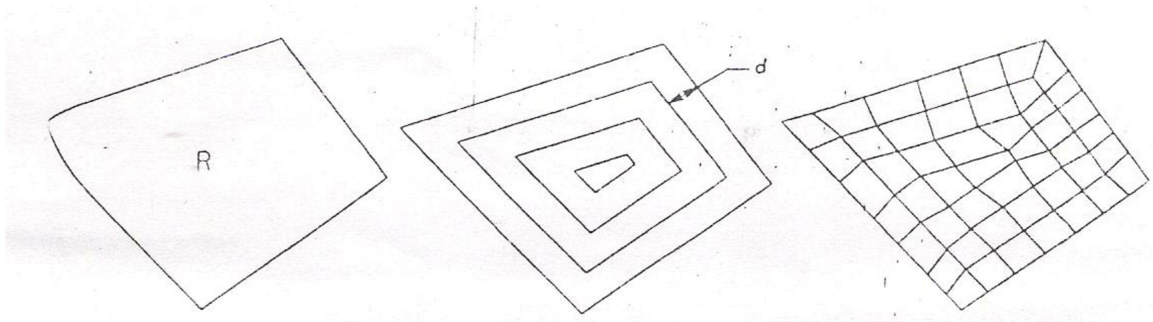
The majority of triangulation methods generate a finite element mesh in two steps. First, nodal points are generated within the region to be meshed. Some methods generate nodal points randomly in the region to be meshed. Others use an imaginary grid of horizontal lines that intersect the region. Second, the triangular elements are generated by connecting the nodal points in some optimum fashion. Offsetting techniques fall under this category of mesh generation.

2.1 OFFSETTING IN GEOMETRIC MODELLING.

Offsetting is broadly classified into two categories – (i) Constrained Offsetting and (ii) Free form Offsetting.

1.1.1 Constrained Offsetting – A typical region has been shown in the figure 1. The constrained offsetting approach utilizes an offsetting distance, "d" and successively shrinks the region by "d" to subdivide it into a nest of concentric layer.

Thus the region is delaminated into lamina separated by the distance, d. Each lamina is divided into finite elements and the resulting elements are connected properly to generate a valid mesh. Mesh algorithm based on constrained offsetting are bound to fail more than they succeed in meshing a given region and most likely they succeed in meshing simple region.



(a) REGION "R" (b) DELAMINATED REGION (C) ELEMENT GENERATION

Fig. 1 Mesh Generation by constrained offsetting.

2.1.2 Free-Form Offsetting

The concept of free form offsetting is based on offsetting points instead of offsetting curve. A set of selected points in space can be offset by an offset distance, d along any offset vector in any desired direction. If the set of points belong to a boundary, the resulting set of offsetting and creating points can be made recursive till the entire given domain gets discretised.

2.0 Ray Based Offsetting

In the ray based offsetting approach the region to be meshed is subdivided by grid of horizontal / vertical / inclined lines. The intersection points between rays that bombard the region and the region boundary are offset.

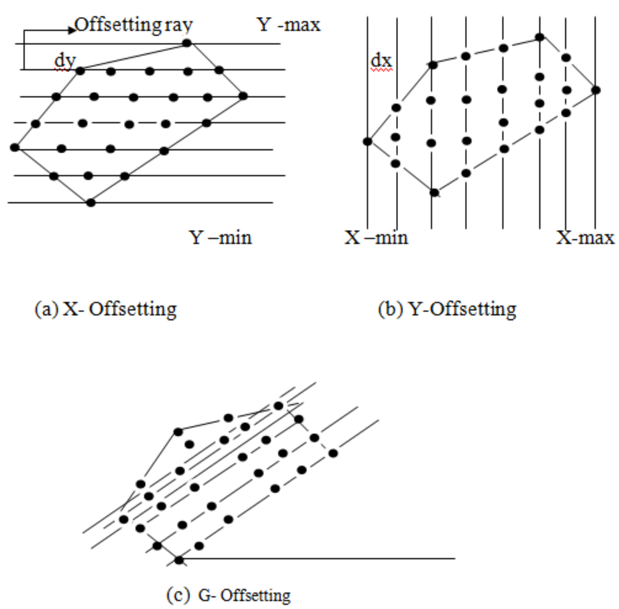


Fig. 2 Ray Based Offsetting

The offsetting direction can be defined by an offsetting vector d. Three types of offsetting exist depending on the orientation of the vector d relative to X-Y co-ordinate system.

(i) Offsetting in X-direction (X – offsetting)
(ii) Offsetting in Y –direction (Y- offsetting)
(iii) Offsetting in any general direction (G- offsetting)

$d_1 = [\, d \;\; 0 \,]^T$ --(1)

$d_2 = [\, 0 \;\; d \,]^T$ --(2)

$d = [\, d_1 \;\; d_2 \,]^T$ -- (3)

Such that $d = \sqrt{d_1^2 + d_2^2}$ ------------------- (4)

In ray-based offsetting, the distance between two consecutive rays may be equal to or different from the offset distance d. In addition, the first and last rays pass through boundary points of minimum and maximum co-ordinates. In the present work, X- offsetting has been used to generate nodal points and subsequently valid mesh.

In the ray – based approach, the intersection points between rays that bombard the region and the region boundary are offset. Y co- ordinate corresponds to any j^{th} intersection point was the same as calculated by the above mentioned ray equation. In the ray based approach any ray intersects the region on two points (left/right). In the extreme case both points may coincide (as in the case of a vertex of a triangle).

By inputting the value of Y co-ordinate of j^{th} ray in the equation of boundary curves, X co-ordinate of intersection point has been calculated. As one ray intersected the boundary on two points there are two value of X co-ordinate corresponding to one ray (i.e. one value of Y co-ordinate). By calculating two values of X co-ordinate corresponding to each of the bombarded rays a set of left and right X co-ordinate was generated. Xlba [j] and Xrba [j] are the value of X co-ordinate on the left and right side of j th ray. Now this left and right X co-ordinate of boundary nodes is the basic input to the mesh generating algorithm.

Offset Node Generation on the Bombarded Rays

Entire convex region got segmented by the series of horizontal rays between two extremities of the given region. Each of the horizontal segments was made of two consecutive horizontal rays. These consecutive horizontal rays were connected by a line segment on the left and right side. These two line segments were drawn between two consecutive intersection points of the rays with the region boundary on left and right side. Following are the basic steps for generating offset nodes.

(i) Offset nodes were generated only on horizontal rays. To ensure valid offsetting, the offset distance (element size) d was modified if necessary in the same fashion as in modifying the distance between the rays. The length of ray segment bounded by two left and right boundary nodes was divided by the offset d to determine the possible generation of offset nodes on the segment. No. of offsetting on any jth ray was calculated by

(ii) (iii) (iv)

$$n[j] = INT\left(\frac{Xrba[j] - Xlba[j]}{d}\right) \qquad (5)$$

(v) Where, INT is the nearest integer function. Similarly offset distance on the jth ray was modified as

(vi) (vii)

$$dr1[j] = \left(\frac{Xrba[j] - Xlba[j]}{n[j]}\right) \qquad (6)$$

(viii) As one horizontal segment lamina was made of two consecutive ray, no. of off-settings and modified offsetting distance correspond to [j + 1]th ray was also needed which were calculated as

(ix) (x) (xi)

$$n[j+1] = INT\left(\frac{Xrba[j+1] - Xlba[j+1]}{d}\right) \qquad (7)$$

(xii) (xiii)

$$dr1[j+1] = \left(\frac{Xrba[j+1] - Xlba[j+1]}{n[j+1]}\right) \qquad (8)$$

(ii) Actual Boundary of the given domain was constructed by drawing a series of line segments, connecting two consecutive intersection points of the horizontal rays and the boundary of the domain, in the left/right side of the domain.

As shown in the Fig.2 any arbitrary 2 D planar domain was made of horizontal segments (lamina). These segments were like building block units. In present study mesh generation of entire domain was achieved using “PIECE –WISE APPROACH” meshing one segment at a time.

Depending upon the position of the left/right ends of one ray relative to other ray in a segment, actual boundary of the segment was modified. Modification of actual boundary of the segment and subsequent creation of additional nodes ensured elements of good shape and 100% discretization of the segment.
If the distance between the left/right ends of the two consecutive rays was within permissible limit, no modification has been done. Otherwise, boundary of a segment was modified on left/ right or both side. In the given Fig. 2 modified boundary of a segment has been shown with dotted (…..) line whereas actual boundary has been shown with thick line. Following parameters have been calculated to check the permissible limit and subsequent modification of the boundary.

$$R_1 = INT\left(\frac{(Xlba [j] – Xlba [j + 1])}{dr1 [j + 1])} \right) \quad (9)$$

$$R_2 = INT\left(\frac{(Xrba [j +1] – Xrba [j])}{dr1 [j + 1])} \right) \quad (10)$$

$$R_3 = INT\left(\frac{(Xlba [j + 1] – Xlba [j])}{dr1 [j])} \right) \quad (11)$$

$$R_4 = INT\left(\frac{(Xrba [j] – Xrba [j + 1])}{dr1 [j])} \right) \quad (12)$$

For modification on the left end of a segment, value of R1 or R3 should be more than 0. Zero value of both R1 and R3 ensured that no modification of boundary was needed on the left end of the segment and actual boundary coincided with modified boundary. In case of modification of boundary actual boundary got shifted on either top ray (R1 > 0) or bottom ray (R3 > 0).
Similarly, zero value of R2 and R4 ensured that no modification of boundary was needed on the right end of the segment. Otherwise actual boundary got shifted on top ray (R2 > 0) bottom ray (R4 >0).
On the basis of above discussion it may be concluded that in any segment, depending upon the relative position of top and bottom rays on left and right end, one/two or three distinct region existed.

(a) Region between actual and modified boundary on left hand side of the segment.
(b) Region between modified boundaries.
(c) Region between actual and modified boundary on right hand side of the segment.

Similarly by modifying the boundary of entire segments the Modified Boundary of the entire domain was created.

Additional Boundary Node Generation

If boundary of a segment is modified on the left hand side, additional boundary nodes are generated on the line segment joining two consecutive intersection points on the left hand side. This has been shown as hollow circle. No. of such additional boundary nodes are equal to the offset nodes between Actual and Modified Boundary of the segment on either top or bottom horizontal ray depending on whether ($R_1 > 0$) or ($R_3 > 0$).
Equation of a line segment joining two points (X_1, Y_1) and (X_2, Y_2) have been represented as follows :

(Y-Y1)/(Y1-Y2)=(X-X1)/(X1-X2) (13)

For any left side line segment

Y1=Y [j] = Ymax – j * dr (14)

Y_2 = Y [j + 1] = Ymax – (j + 1)*dr (15)

X_1 = Xlba [j] (16)

X_2 = Xlba [j + 1] (17)

If one offset node lies between actual and modified boundary, only one additional boundary node was generated.
Y co- ordinate of such boundary node = Y [j] – dr/2 (18)

X co-ordinate was calculated by above mentioned equation of line.
Similarly for right side line segment

$$Y_1 = Y[\,j\,] \tag{19}$$
$$Y_2 = Y[\,j+1\,] \tag{20}$$
$$X_1 = Xrba[\,j\,] \tag{21}$$
$$X_2 = Xrba[\,j+1\,] \tag{22}$$

Triangular Element Generation

Depending on the relative position of left and right ends of two consecutive rays actual boundary of a horizontal segment are either modified on left/right or both side of a segment. Actual boundary will not be modified at all if relevant parameters (R1, R2, R3 & R4) have non zero value.

In case of unmodified boundary no additional boundary nodes are created on either side of the segment. Elements are generated only out of offset nods, which are created on two consecutive horizontal rays.

In case of distinct modified boundary, a segment got divided either into two or three regions depending on whether boundary got modified on left/right or both sides of segment. Triangular elements between the region of actual and modified boundary on left/right or both sides of the segment were generated out of horizontal offset nodes and additional boundary nodes. In the interior region between left and right modified boundary elements were generated only out of offset nodes located on two consecutive horizontal rays.

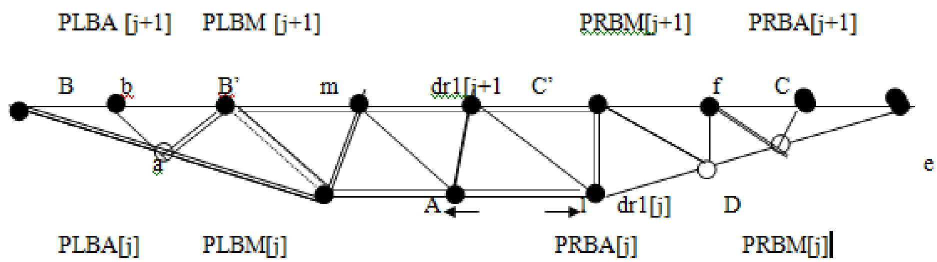


Fig. 3 Boundary Modification & Triangulation of a Segment.

PLBA [j] → Actual left boundary point on the j^{th} ray [A].
PLBM [j] → Modified left boundary point on j^{th} ray [A], coincided with PLBA [j].
PLBA [j + 1] → Actual left boundary point on $[\,j + 1]^{th}$ ray [B]
PLBM [j + 1] → Modified left boundary point of $[\,j+1]^{th}$ ray [B'] generated by shifting B
by 2 times dr1 [j + 1] (i.e. R1 = 2)
PRBA [j] → Actual right boundary point of j^{th} ray [D]
PRBM [j] → Modified right boundary point on j^{th} ray [D], coincided with PRBA [j]
PRBA [j+1] →Actual right boundary point on $(j+1)^{th}$ ray [C]
PRBM [j+1] →Modified right boundary point on $(j+1)^{th}$ ray [C'], generated by shifting C by
3 times dr1[j+1] (R2=3), where
dr→ distance between consecutive rays.
dr1 [j] → Modified offset distance on j^{th} ray.
dr1[j+1] → Modified offset distance on $[\,j + 1\,]^{th}$ ray
ABB' → Region between actual boundary & modified boundary on left side of the segment.
DCC' = Region between actual boundary & modified boundary in right side of the segment.
AB'CD' = Region between left and right modified boundary.
● → Offset nodes on the horizontal rays.
○ → Additional boundary nodes generated on the line segment joining two boundary nodes.
______ Modified Boundary
.... Actual Boundary.

Basic steps in element generation:
* Node generation and element generation is a CONCURRENT process in present study.

In the region ABB' element was generated out of Offset nodes on the upper ray segment BB' and additional boundary nodes on the boundary line segment AB. Two consecutive points on BB' and two consecutive points on AB was generated. One quadrilateral AabB was generated. Two triangle AaB' and abB' were generated by joining shorter diagonal points a and B'.

Generation of such quadrilateral was referred as Basic Quadrization. It is a recursive process until only three points (a,b and B) remained left in the region.
Y co-ordinate of the offset nodes on ray segment BB' was same as the value of $[j + 1]^{th}$ ray calculated by equation.
$Y [j + 1] = Y max - (j+1) * dr$
X co-ordinate was calculated by continuously subtracting modified offset distance from X PLBM.
Co-ordinate of additional boundary nodes were calculated as explained in earlier section.
* In the interior region AB'C'D Recursive Basic Quadrization has been started by selecting two consecutive left end points A an l on j^{th} ray and two consecutive points B' and m on $[j+1)^{th}$ ray. Points on lower rays were referred as P[j][k] and P[j] [k+1] and points on upper ray were referred as P[j+1`] [k] and P[j+1] [k+1].
Y co-ordinate of the points was calculated by earlier mentioned ray equation.
$Y [j] = Y max - j * dr$ and $Y[j+1] = y max - [j+1] * dr$

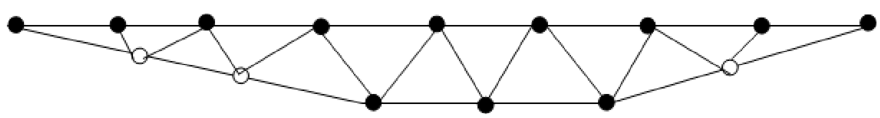
$$X[j] [k] = XLBM [j] + K*dr1 [j] \quad (23)$$

$$X[j] [k+1] = XLBM [j] + (K+1)*dr1 [j] \quad (24)$$

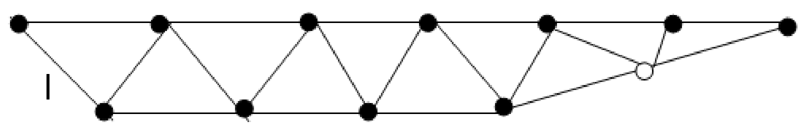
$$X[j+1] [k] = XLBM [j+1] + K*dr1 [j+1] \quad (25)$$

$$X[j+1] [k+1] = XLBM [j+1] + [k+1]*dr1 [j+1] \quad (26)$$

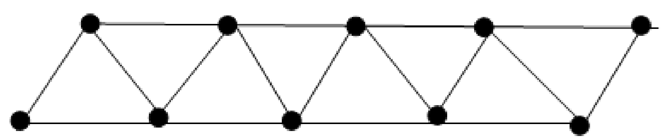
Shorter diagonal Am created two triangles AB'm and Aml.
Above explained process was recursive until P[j] [k+1] reached at PRBM [j] or P [j +1] [k+1] reached at PRBM [j +1]
* In the region DC'C same process as that of region ABB' was repeated till only three points (e, f and C remained).
Two additional boundary nodes were generated to match two offset nodes in the region. Basic quadrization was repeated twice. Above mentioned three basic steps were repeated for the entire segments till the entire domain got meshed and consequently triangulated.



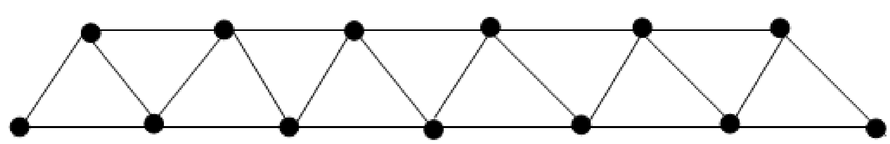
(A) Boundary modified on both side of the segment (Upper ray larger)



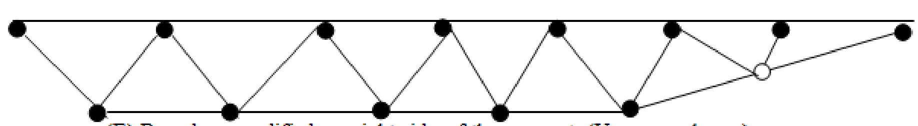
(B) Boundary modified on right side of the segment. (Upper ray larger)



(C) Boundary modification is not needed. (rays are of equal length)

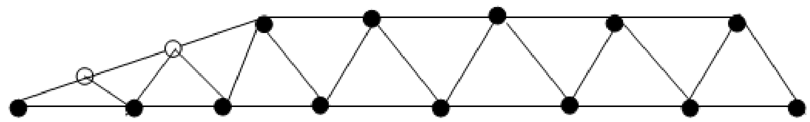


(D) Boundary modification is not needed. (Upper ray shorter)

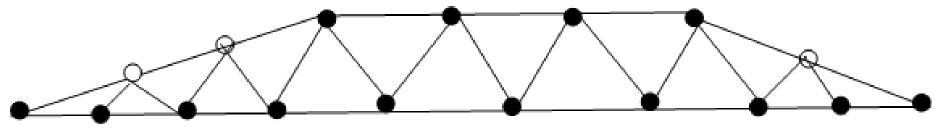


(E) Boundary modified on right side of the segment (Upper ray larger)

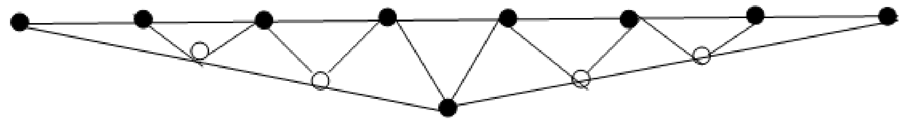
Fig. 4 Triangulation of segments of different shape (Contd.)



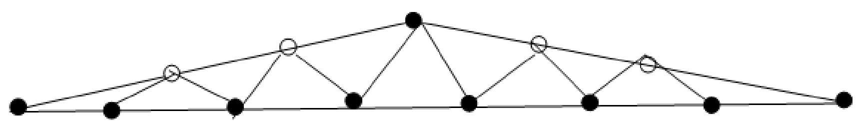
(F) Boundary modified on left side of the segment.(Lower ray larger)



(G) Boundary modified on both side of the segment. .(Lower ray larger)

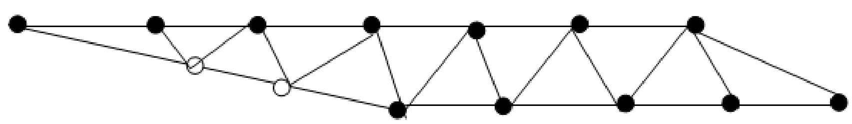


(H) Boundary modified on both side of the segment. .(left and right side coincided as vertex on lower ray.)

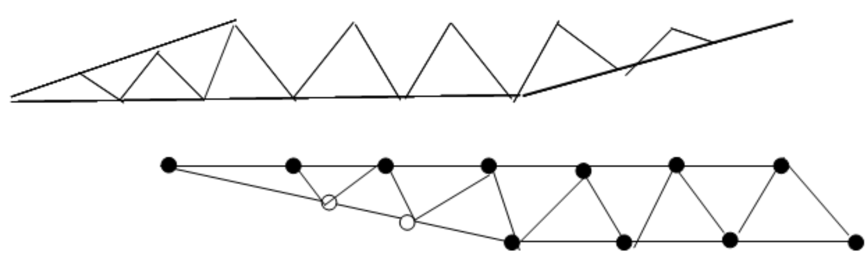


(I) Boundary modified on both side of the segment. .(left and right side coincided as vertex on upper ray.)

Fig. 5 Triangulation of segments of different shape.(Contd.)



(J) Boundary modified on left side of the segment. (Offsetting for boundary modification on upper ray)



(K) Boundary modified on left side of the segment.

Fig. 6 Triangulation of segments of different shape.

Quadrilateral Element Generation

Discretizing a two dimensional region into all quadrilateral elements is not always a trivial task. In fact, very limited region topologies, such as rectangles or squares, are guaranteed to produce quadrilateral elements only. Meshing other region topologies almost always produce at least one triangular element. A triangular element can always be decomposed into three quadrilateral elements by generating three mid-side nodes and a centroid node. The mid side nodes are linear interpolation of the triangle corner nodes i.e.

$P_m = (P_i + P_j)/2$ (27)

here i and j are any two corner nodes. Similarly,

$P_{cg} = (P_i + P_j + P_k) / 3$ (28)

Where i, j and k are three corner nodes,

As shown In figure, quadrizing a triangular element introduces mid-side nodes for the already existing neighbouring quadrilateral elements.

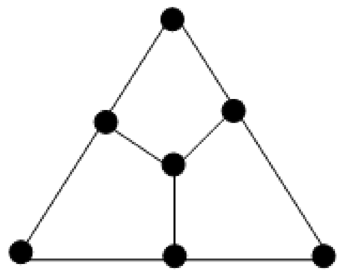


Fig.7 Decomposition of a Triangle into Three Quadrilaterals.

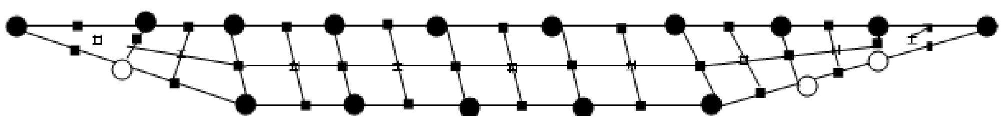
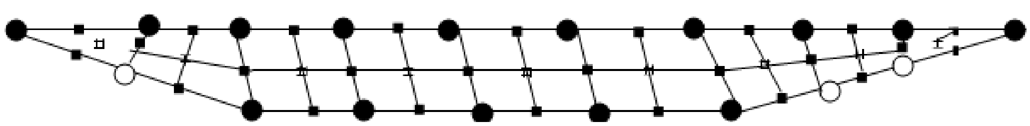


Fig.7 Decomposition of a Triangle into Three Quadrilaterals.



Offset Nodes
Additional Boundary Nodes.
Mid Side Nodes
Centroid Nodes

Fig. 8 Quadrization of a segment.

The existence of these nodes were eliminated by decomposing each existing quadrilateral element into four quadrilateral ones by creating four mid-side nodes and centroid node as done with triangular elements. It was concluded that with the input element size, d element of d/2 size got generated. So, basic input element size was doubled to get the elements of desired size. Basic steps in QUADRIZATION is mentioned below

(i) Modified boundary of entire segments was determined.

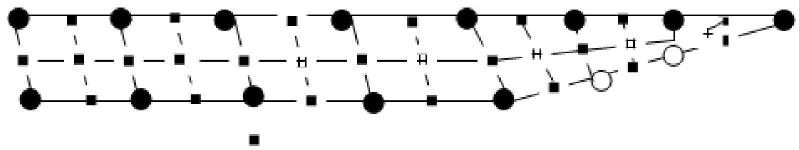
(ii) Additional boundary nodes were calculated if required in the left/right side of the segment.

(iii) Basic quadrization in the boundary region terminated when only three nodes remained in the region(i.e. a triangular element). Basic quadrization in interior region was started with two left most offset nodes on lower ray and upper ray of a segment. It was a recursive process until the entire segment got quadrized.

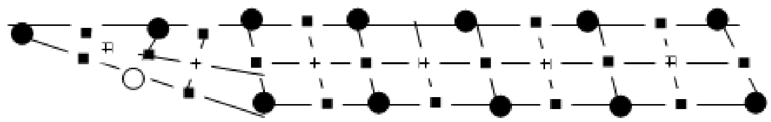
(iv) In this work generation of nodes and elements are concurrent process. In addition to generating offset nodes and additional boundary nodes (if required on the left/right side of a segment) mid-side nodes and centroid nodes were computed simultaneously.

(v) One Basic Quadrilateral was decomposed into four quadrilateral elements. Remaining triangular region was decomposed into three quadrilateral elements.

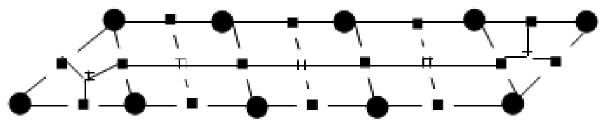
(vi) Process has been repeated recursively for all the domain segments till the entire domain got discretized and quadrized



(A) Boundary modified on right hand side (Upper ray larger)

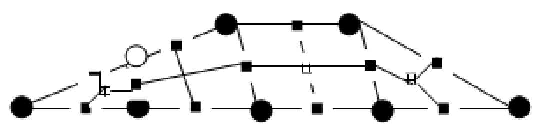


(B) Boundary modified on left hand side (Upper ray larger)

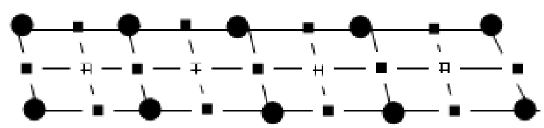


(C) Boundary modified on Both side (rays are of equal length)

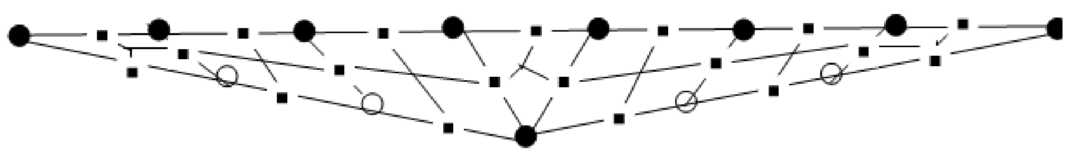
Fig. 9 Quadrization of segments of different shape.



(D) Boundary modified on both side (Upper ray smaller)



(E) Boundary not modified (rays are of equal lenght)



(F) Left and right hand side coincided at lower vertex. (Both side modified)

Fig. 10 Quadriszation of segments of different shape.

3. RESULTS AND DISCUSSIONS

A Computer program (source code) in C has been developed using equations developed above to obtain requisite outputs. Outputs in the form of disretised domain graphics having triangular/quadrilateral elements, selective / overall refined mesh graphics, nodal coordinates node-element connectivity list etc. have been obtained by running computer program.

A galaxy of planar domains has been discretised in this study. Domains included simple convex regions as well as regions having interior holes and complex arbitrary boundary.

Quality of generated mesh graphics has been evaluated on the basis of key parameters such as element shape, no. of nodes /elements generated and processing time.

Key parameters of the mesh generated in previous pages have been summarized in tabulated form as follows:

Table 1 Key parameters of simple convex domains

Sl. No.	Description of domain	Input element size	No. of nodes	No. of elements	Value of γ/γ^*		
					1.0-1.1	1.1-1.15	>1.15
01.	Triangular	10	1236	2318	70	25	5
02.	Circle	10	1546	3060	59	31	10
03.	Semi-Circle	10	1448	2865	61	30	9
04.	Quadrant of a circle	10	736	1474	58	32	10
05.	Quadrant of a circle having hole at the center	10	455	834	56	34	10
06.	Quadrant of a circle having hole at the center (Curved pipe shaped)	10	362	664	57	30	13
07.	Wing of an Aeroplane	10	284	556	78	20	2
08.	River shaped domain	10	195	189	72	25	3
09.	Double Convex shaped	20	206	323	60	35	5
10.	Truncated Circle	20	151	173	55	40	5
11.	Ellipse	10	324	592	65	32	3
12.	Ellipse having right side truncation	10	234	368	60	32	8
13.	Octagonal	10	541	1008	--	100	--
14.	Octagonal domain having right side truncation	10	310	608	35	60	5

4. CONCLUSIONS

The following salient conclusions have emerged out from the present study;

1. An algorithm based on "Ray Based Offsetting" has been successfully developed for discretization (triangular and quadrilateral elements) of an arbitrary planar domain.
2. Basic algorithm is extended to handle input geometry of realistic complexity including regions with arbitrary complex boundaries, interior holes etc.
3. To ensure 100% quadrization of an arbitrary domain, existing triangular mesh has been converted to quadrilateral mesh by transforming one triangle into three quadrilaterals.
4. Element size being basic input, global mesh density control is much easier. For selective refinement of the mesh separate algorithms have been generated for triangular and quadrilateral meshes.
5. Mesh generated by present method exhibit good transition between regions of different elements size.

5. REFERENCES

[1]. Buell W.R. and Bush B.A. " **MESH GENERATION – A SURVEY"**, Transactions of ASME Journal, PP 332-338, February 1973

[2]. Howard D. Cohen, **"A METHOD FOR AUTOMATIC GENERATION OF TRIANGULAR ELEMENT ON A SURFACE"**, International Journal for Numerical Methods in Engineering, August 1980.

[3]. LO S.H., **"A NEW MESH GENERATION SCHEME FOR ARBITRARY PLANAR DOMAINS"** International Journal for Numerical Methods in Engineering, Vol. 21, PP 1403 –1426, 1985.

[4]. Joe B. Simpson R.B., **"TRIANGULAR MESHES FOR REGIONS OF COMPLICATED SHAPE"** International Journal for Numerical Methods in Engineering, Vol. 23, PP 2581-2605, 1986.

[5]. Ho-Le K., **"FINITE ELEMENT MESH GENERATION METHODS: A REVIEW AND CLASSIFICATION",** Computer Aided Design, Vol. 20, PP 27-38, 1988.

[6]. William H. Frey, David A. Field, **"MESH RELAXATION: A NEW TECHNIQUE FOR IMPROVING TRIANGULATION"** International Journal for Numerical Methods in Engineering, Vol. 31, PP 1121 – 1133, 1991.

[7]. Levent Sezer and Ibrahim Zeid, **"AUTOMATIC QUADRILATERAL/ TRIANGULAR FREE – FROM MESH GENERATIONM FOR PLANAR REGIONS"** International Journal for Numerical Methods in Engineering, Vol. 32, PP 1441 – 1483, 1991.

[8]. Bruce P. Johnston and John M. Sullivan, Jr. **"FULLY AUTOMATIC TWO DIMENSIONAL MESH GENERATION USING NORMAL OFFSETTING"** International Journal for Numerical Methods in Engineering, vol. 33, PP 425- 442, 1992.

[9]. T. Fang and L. Piegl**, "DELAUNAY TRIANGULATION USING A UNIFORM GRID**", IEEE Computer Graphics and Application, PP 36- 47, 1993

[10]. Ernest Rank, Manfred Schweingruber and Markus Sommer, **"ADAPTIVE MESH GENERATION AND TRANSFORMATION OF TRIANGULAR TO QUADRILATERAL MESHES"**, Communication in Numerical Methods in Engineering, Vol. 9, PP 121- 129, 1993.

[11]. R.V. Nambiar, R.S. Valera, K.L. Lawrence, **"AN ALGORITHM FOR ADAPTIVE REFINEMENT OF TRIANGULAR ELEMENT MESHES",** International Journal For Numerical Methods In Engineering, Vol. 36, PP 499-509, 1993.

[12]. Yuan-Yao Qian and G. Dhatt**, "A SIMPLE, ADAPTABLE 2D MESH GENERATION PACKAGE"**, Computer & Structure, Vol. 53, No.- 4, PP 801-810, 1994.

[13]. Antonio C.O. Miranda, Joquim B. Cavalcante Neto, Luiz F. Martha, **"AN ALGORITHM FOR TWO-DIMENSIONAL MESH GENERATION FOR ARBITRARY REGIONS WITH CRACKS"**, IEEE Transmagnetics Magazine, PP. 29-38, July 1999.

[14]. R. Rodriguez, I. Gutierrez, **"MECHANICAL BEHAVIOUR OF STEELS WITH MIXED MICRO STRUCTURE"**, Material Science & Engineering,, PP. 356 - 363, 2003.

[15]. Mark W. Beall, Joe Walsh, Mark S. Shephard, **"A COMPARISION OF TECHNIQUES FOR GEOMETRY ACCESS RELATED TO MESH GENERATION"**, Engineering with Computers, Vol.- 20, No.- 3, PP. 210 – 221, 2004.

[16]. Jonathan Richard Shewchuk, **"A SIMPLE TWO DIMENSIONAL MESH GENERATOR"**, Computer Science Division, University of California, July' 2005.

NOMENCLATURE

Y max	Y co-ordinate corresponding to the first ray
Y min	Y co-ordinate corresponding to the last ray
diff	Difference between Ymax and Ymin.
d	Input element size.
ray	Total no. of rays.
dr	Distance between consecutive rays.
j	Counter of ray.
Plba[j]	Intersection point between jth ray and the domain boundary on left side.
Prba[j]	Intersection point between jth ray and the right side domain boundary.
Xlba[j]	X co-ordinate of Plba[j]
Xrba[j]	X co-ordinate of Prba[j]
n[j]	No. of offsettings (nodes) on jth ray.
dr1[j]	Modified element size (distance between nodes) on jth ray.
Plb[j]	Modified boundary nodes of jth ray on left side of the domain.
Prb[j]	Modified boundary nodes of jth ray on right side of the domain.
Xlb[j]	X co-ordinate of Plb[j]
Xrb [j]	X co-ordinate of Prb [j]
Y [j]	Y co-ordinate of jth ray.
Y [j+1]	Y co-ordinate of (j+1)th ray.
R1, R3	Boundary modifying parameter on left side of the segment.
R2, R4	Boundary modifying parameter on right side of the segment.
k	Node counter on a ray.
Xlbn[j][k]	X co-ordinate of Kth offset node on jth ray between actual and modified boundary of the segment.
Xlbc [j] [k]	X Co-ordinate of Kth additional boundary node on line segment between Plba [j] and Plba [j+1]
Ylbc[j] [k]	Y Co-ordinate of above point.
Xrbc[j][k]	X co-ordinate of Kth additional boundary node on line segment between Prba[j] and Prba[j+1]
Yrbc [j] [k]	Y co-ordinate of above point.
X[j][k]	X co-ordinate corresponding to Kth node between modified boundaries on jth ray.
Xcg[j] k]	Centroid node X coordinate of basic quadrilateral formed by P[j][k], P[j][k+1], P[j+1] [k], P[j+1] [k+1]
Ycg [j] [k]	Y co-ordinate of centroid node.
Srms	Root mean square value of sides of a triangle.
A	Area of a triangle.
A*	Area of an equilateral triangle.
γ	Metric value of a triangle
γ^*	Metric value of an equilateral triangle.
γ/γ^*	Normalized dimensionless parameter.