

SUBDIVISION BASED SHAPE OPTIMIZATION OF COMPLIANT MECHANISMS

A Thesis

By

AKSHAT SINGH

Submitted to the College of Graduate Studies  
Texas A&M University-Kingsville  
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2011

Major Subject: Mechanical Engineering

UMI Number: 1504373

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 1504373

Copyright 2011 by ProQuest LLC.

All rights reserved. This edition of the work is protected against unauthorized copying under Title 17, United States Code.



ProQuest LLC  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106-1346

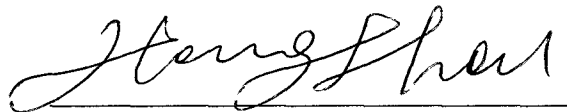
SUBDIVISION BASED SHAPE OPTIMIZATION OF COMPLIANT MECHANISMS

A Thesis

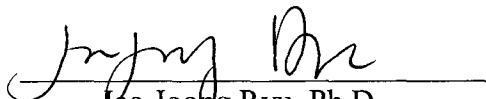
By

AKSHAT SINGH

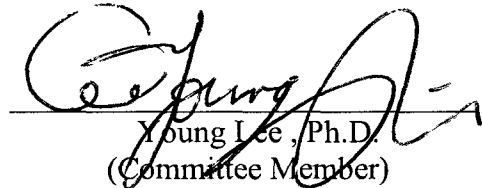
Approved as to style and content by:



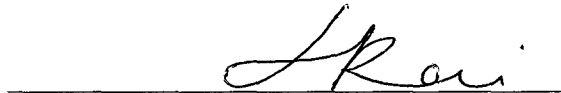
Hong Zhou, Ph.D.  
(Chairman of Committee)



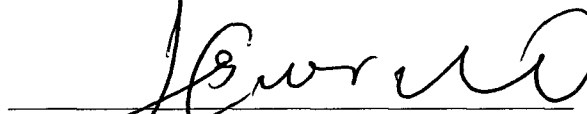
Jae-Joong Ryu, Ph.D,  
(Committee Member)



Young Lee, Ph.D.  
(Committee Member)



Kai Jin, Ph.D.  
(Interim Chair of Mechanical & Industrial Engineering)



Ambrose O. Anoruo, Ph.D.  
(Associate VP for Research and Dean of Graduate Studies)

May 2011

## **ABSTRACT**

### **Subdivision Based Shape Optimization of Compliant Mechanisms**

(May 2011)

Akshat Singh, B.E. V.N.S.G. University, India

Chairman of Advisory Committee: Dr. Hong Zhou

Shape optimization is basically the idea of generating the best profile of a surface so that it improves its mechanical properties and minimizes some other properties such as weight and reduces stress concentration factor at corners. It is an important practice since it largely saves time, cost and material during the designing process.

Using method of subdivision includes refining a coarse grid of vertices to produce a set of refined vertices. This would alter the geometric features of the surface. The final shape and surface smoothness depends on the kind of refinement scheme applied to it and the number of iterative meshes the surface goes through for refinement.

In this thesis, Chaikin's method for subdivision will be used to subdivide the control points on the surfaces of two compliant mechanisms. Compliant displacement inverter and amplifier, and compliant gripper will be used for this analysis. For analysis purposes, the regular quadrilateral discretization and modified quadrilateral discretization will be considered as the topological optimization conditions for both the mechanisms.

## ACKNOWLEDGEMENT

I would like to take this opportunity to express my deepest gratitude to my advisor Dr. Hong Zhou for his constant encouragement, guidance and support throughout my thesis work. It has been a pleasure to work with him and an honor to be his student. His patience and constant encouragement really helped in completing my work efficiently. His kind words instilled a positive self-believe in me not only towards my thesis work but also towards life. I could not have imagined having a better advisor and mentor for my Master's study.

I would like to acknowledge my sincere thanks to all the committee members Dr. Jae-Joong Ryu and Dr. Young Lee, for their advice, support, co-operation and for their time to review my thesis.

My special thanks to the entire faculty of Department of Mechanical Engineering for their support and the knowledge that I gained during my course of study at this university.

Finally, I would also like to thank my parents, my friends for their help and constant support throughout my thesis work and throughout my master's program and making me feel comfortable away from home.

I would like to give a special mention to my friend Mr. Pranjal Killekar for his constant support and helpful inputs on my thesis and also for keeping me positive during difficult times. I would also like to thank Mr. Amit Dhamdhame who helped me in making this report more efficient and presentable.

## TABLE OF CONTENTS

ABSTRACT.....	iii
ACKNOWLEDGEMENT .....	iv
TABLE OF CONTENTS.....	v
LIST OF FIGURES .....	vii
CHAPTER I INTRODUCTION.....	1
1.1    Shape Optimization.....	1
1.2    Subdivision .....	1
1.4    Motivation and Objectives.....	5
1.5    Thesis Organization .....	5
CHAPTER II CHAIKIN’S ALGORITHM FOR SUBDIVISION.....	7
2.1    Background and Definition of Subdivision .....	7
2.2    Chaikin’s Algorithm for Subdivision.....	8
2.3    Topological Rules for Chaikin’s Algorithm .....	10
2.4    Geometric Rules for Chaikin’s Algorithm.....	11
CHAPTER III OTHER SUBDIVISION SCHEMES.....	12
3.1    Background and Definition.....	12
3.2    Classification of Subdivision Refinement Schemes .....	13
3.3    Doo-Sabin Subdivision Scheme .....	16

3.4	Topological Rules of Doo-Sabin Subdivision Scheme.....	17
3.5	Geometric Rules of Doo-Sabin Subdivision Scheme .....	21
3.6	Doo-Sabin and Mid-edge Scheme .....	22
3.7	Loop Subdivision Scheme .....	26
CHAPTER IV SHAPE OPTIMIZATION OF COMPLIANT MECHANISMS.....		36
4.1	Input .....	36
4.2	Output .....	36
CHAPTER V DISCUSSION AND CONCLUSION .....		61
REFERENCES .....		63
VITA .....		66

## LIST OF FIGURES

Figure 1 Subdivision of Closed Loop using Chaikin’s Method for “Corner Cutting” : (a) First Mesh; (b)-(d) New Meshes After 1st, 2nd & 3rd Generations. [20] .....	8
Figure 2 Subdivision of Open Loop using Chaikin’s Corner Cutting Algorithm [16].....	9
Figure 3 Topological Rules for Chaikin’s Scheme [20] .....	10
Figure 4 Face Split Scheme. [19].....	15
Figure 5 Vertex Split Scheme. [19] .....	16
Figure 6 (a)-(f) Doo-Sabin Subdivision Steps and 1st Refinement. ....	21
Figure 7(a)-(e) Doo-Sabin and Mid-edge scheme .....	23
Figure 8 Binary Loop Subdivision [16] .....	27
Figure 9 Ternary Loop Subdivision [16] .....	29
Figure 10 (a)-(c) Subdivision Masks for Ternary Loop Subdivision [16].....	31
Figure 11 (a)-(d) Binary Loop Subdivision for initial mesh with four points .....	32
Figure 12 Initial Mesh.....	37
Figure 13 1st Refinement.....	38
Figure 14 2nd Refinement .....	39
Figure 15 3rd Refinement .....	40
Figure 16 4th Refinement .....	41
Figure 17 5th Refinement .....	42
Figure 18 Initial Mesh.....	43
Figure 19 1st Refinement.....	44
Figure 20 2nd Refinement .....	45



Figure 21 3rd Refinement .....	46
Figure 22 4th Refinement .....	47
Figure 23 5th Refinement .....	48
Figure 24 Initial Mesh.....	49
Figure 25 1st Refinement.....	50
Figure 26 2nd Refinement .....	51
Figure 27 3rd Refinement .....	52
Figure 28 4th Refinement .....	53
Figure 29 5th Refinement .....	54
Figure 30 Initial Mesh.....	55
Figure 31 1st Refinement.....	56
Figure 32 2nd Refinement .....	57
Figure 33 3rd Refinement .....	58
Figure 34 4th Refinement .....	59
Figure 35 5th Refinement .....	60

# CHAPTER I

## INTRODUCTION

### 1.1 Shape Optimization

Shape optimization is the practice of finding the best profile for a component so that it has superior mechanical properties. When optimizing for shape, geometric modeling and structural analysis are integrated into one absolute, automated computer-aided design process. The boundary profile for any element of a 2-D or 3-D structure having least mass can be determined, under constraints of certain geometrical and structural responses, by shape optimization techniques. By making use of results from the analysis programs, the design can be improved efficiently. During the design process, there is a continuous change in the shape of the object, therefore, careful consideration is required for the following: 1) to describe the change in boundary shape, 2) to maintain an appropriate finite element mesh, 3) to improve the correctness of the sensitivity analysis, 4) to impose proper constraints and 5) to utilize existing optimization methods to solve the shape optimization problem. Because of its benefits of cost, material and time saving in engineering design, shape optimization holds an important area in the field of research.

### 1.2 Subdivision

Subdivision is the method of representing a smooth surface by specifying dense linear mesh for each element of the surface. By the process of recursive subdivision of each polygonal face of the coarse mesh into smaller polygons, a smoother

surface is obtained. It is necessary to define the subdivision surfaces recursively in order to obtain smoother, better surfaces. Initially, the process requires a given polygonal mesh to which a refinement scheme needs to be applied. The refinement scheme when applied to the given polygonal mesh subdivides that mesh. As a result of this subdivision, new vertices and new faces are created. The positions of these new vertices are calculated based upon the locations held by the immediately surrounding old points.

The new surface obtained after the subdivision of the given polygonal mesh, contains a mesh which is denser than the original one since it contains more polygonal faces. By iterative application of this refinement scheme to the new mesh, the limit subdivision surface can be obtained.

Currently, the standard method practiced for modeling of shapes is non-uniform rational B-splines (NURBS). In this representation, a inflexible rectangular mesh of control vertices are used. This representation uses two knot vectors, one along each direction of the edge of the rectangle. Because of this, it has a disadvantage while estimating the shapes of commonly used topology. For example, it becomes difficult to represent surfaces that have holes or that have complex geometry by NURBS. However, the results obtained by using subdivision surfaces compliment the solutions obtained from NURBS. Because of its high efficiency, flexibility and simplicity, subdivision surfaces form an integral part in computer graphic applications.

### **1.3 Current Research Situation**

In his recent work, Z. Wu [1] proposed a highly accurate and efficient approach for shape optimization based on finite element method in association with the popularly

used fully stressed design criteria. In this method, the aim was to decrease the value of the stress concentration factor. The stress value at a fixed point was considered to be the threshold value thus avoiding the restriction on the initial design boundary shape. V. Braibant and C. Fleury [2] showed in their work that B-splines, because of their flexible nature and ability to implement easily with complex shapes, are the most powerful tool for optimum shape design of elastic shapes. J. Stam [4] made an important contribution by providing a convenient and efficient technique for evaluating the Catmull-Clark subdivision surfaces. The surfaces were expressed by certain *eigenbasis* functions. These functions depend on refinement scheme. Because of this contribution many algorithms and investigation techniques which were calculated for parametric shapes can now be extended to Catmull-Clark surfaces. D. Zorin and D. Kristjansson [3] did an extension of the contributions of J. Stam by taking into consideration the subdivision rules for piecewise smooth surfaces with boundaries depending on parameters. They used a varied sets of basis vectors, which were proportional to the coefficients of subdivision rules, for evaluating the surfaces. H. Suzuki, T. Kanai, et al. [5] used Subdivision Limit Position (SLP) for adjusting the control mesh accordingly so that it fits to the data points. However this method is only useful when it is required to create a surface that represents general shape consisting of all the data points. A new algorithm based on quasi-interpolation was introduced by N. Litke, A. Levin and P. Schroder [6] according to which a Catmull-Clark subdivision surface can be fit to a specified shape within a given allowable tolerance. This technique helps in efficiently creating highly detailed features. K.C. Hui and Y.H. Lai [7] proposed an approach by which subdivision surfaces of an object can be created from profile curves lying on the two principle planes. For creating

the surface, it is required to interpolate those surfaces which were generated from sweeping the profile curves about the principle axis. This is a flexible technique by which objects can be generated at various mesh resolutions. J. Lian and Y. Yang [8] introduced a new scheme called the Cross scheme for surface design where the number of vertices were doubled from current level to next subdivision level after each application of the C-scheme. This scheme gives better results than the Catmull-Clark scheme not only at nearby vertices but also in non-vertical and non-horizontal directions. H. Biermann, I. M. Martin, et al. [9] worked on a method where sharp features and trim regions can be created on Catmull-Clark multi resolution subdivision surfaces with a given set of user-defined curves. In their algorithm, the parameterization of a defined surface is changed so that it aligns with a pre-image of the feature curves in the parameter domain which produced a surface similar to the initial surface with curves passing through the mesh edges or the face diagonals. Application of special subdivision rules along the curves then would help in creating sharp profiles and trimming by removing the mesh portion inside the trim curve. A new method of subdivision was introduced by Loop [21] based on the subdivision of triangles called the Binary Loop Subdivision. According to his algorithm, each triangle in the previous mesh is subdivided into four new smaller triangles. Later, Loop [22] extended his own work to subdivide a triangle into nine triangles. This method is called Ternary Loop Subdivision. J. Stam and C. Loop [10] introduced a new hybrid quad/triangle scheme which unifies Catmull-Clark and Loop surfaces in a single framework. Since their algorithm makes use of both the triangular and quadrilateral meshing schemes, the surface produced is better than those in the previous schemes.

## **1.4 Motivation and Objectives**

In this thesis we focus our attention on optimizing the geometric features such as shape and smoothness of two examples of compliant mechanisms, i.e, compliant displacement inverter and amplifier, and compliant gripper.

Topological optimization techniques of regular quadrilateral discretization and modified quadrilateral discretization will be used for subdivision process for both the examples.

Chaikin's algorithm for subdivision of surfaces will be used to compute the position of new vertices.

According to this criterion, the locations of the new points are calculated based on positions of the neighboring vertices.

This process will be repeated till the desired shape and smoothness is achieved.

The results from this analysis can be used in the design of compliant mechanisms and geometric modeling.

## **1.5 Thesis Organization**

This thesis is divided into five chapters.

The first chapter gives a brief introduction about shape optimization and subdivision surfaces. Current research situation and objectives of this thesis are also reviewed in this chapter.

Chapter two discusses the Chaikin's algorithm for subdivision.

The third chapter deals with the other subdivision schemes commonly used.

Subdivision and shape optimization for compliant mechanisms is discussed in chapter four. Examples of compliant displacement inverter and amplifier, and compliant gripper are also discussed in this chapter.

Chapter five concludes the thesis work and throws some insight on the future work that can be done on this topic.

## CHAPTER II

### CHAIKIN'S ALGORITHM FOR SUBDIVISION

In this chapter, the background and basics of subdivision, in general, are discussed. Chaikin's algorithm [23] for subdivision being one of the most simplest, straightforward and easy to implement is also discussed. Because of its ease of implementation for both closed and open loop surfaces, this method can be implemented on any given initial mesh.

#### 2.1 Background and Definition of Subdivision

The origin of subdivision surfaces can be traced back to the late 40s and 50s when a method called "corner-cutting" was used by G. de Rham for describing smooth curves. Catmull and Clark and Doo and Sabin made important contributions when they used subdivision for modeling surfaces. Since then subdivision surfaces have held an vital role in CAGD applications.

Subdivision is basically a method of subdividing a shape recursively till a surface of desired shape and smoothness is achieved. The process of subdivision involves refining a given polygonal mesh on the surface. The new polygonal mesh formed after each step of subdivision contains the new vertices whose positions are calculated with respect to the vertices of the polygonal mesh in the previous step. The new surface obtained after the subdivision of the given polygonal mesh, contains a mesh which is denser than the original one since it contains more polygonal faces. By iteratively applying this refinement scheme to the new mesh, the limit subdivision surface can be



obtained. The surface shape can be altered by choosing a different set of control points to form different control polygons for each mesh refinement.

## 2.2 Chaikin's Algorithm for Subdivision

Chaikin's algorithm for subdivision of surfaces is the most simplest and easy to implement algorithm. It can be easily implemented on both closed loop and open loop surfaces. Figures 1 and 2 show the implementation of Chaikin's algorithm for closed and open loop conditions.

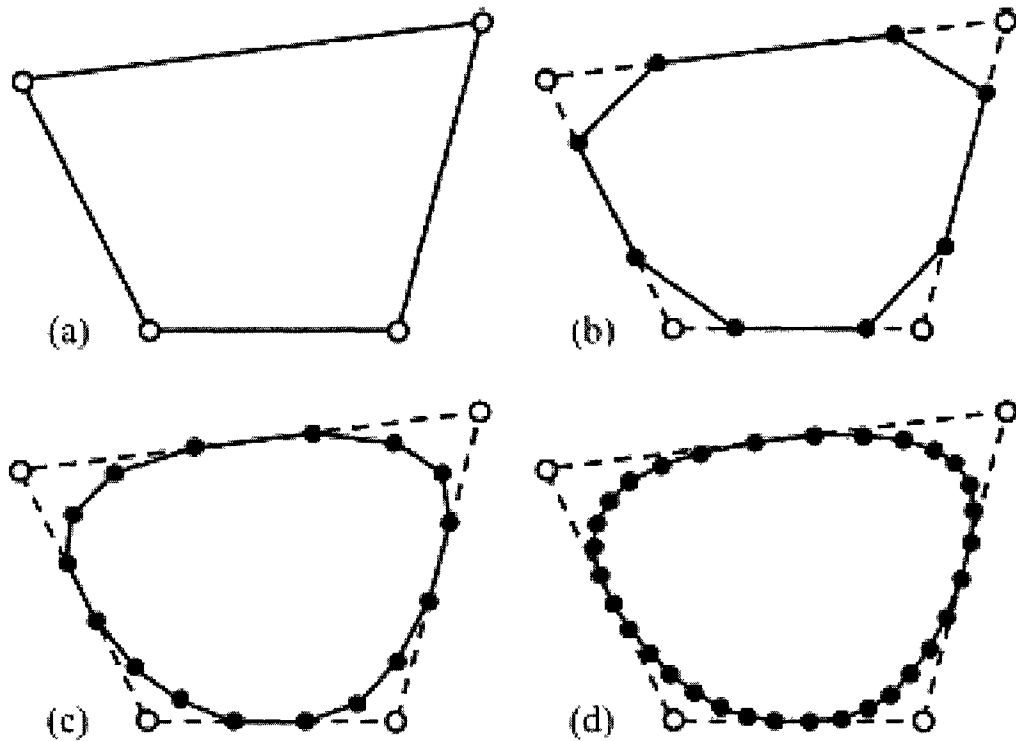


Figure 1 Subdivision of Closed Loop using Chaikin's Method for "Corner Cutting" : (a) First Mesh; (b)-(d) New Meshes After 1st, 2nd & 3rd Generations. [20]

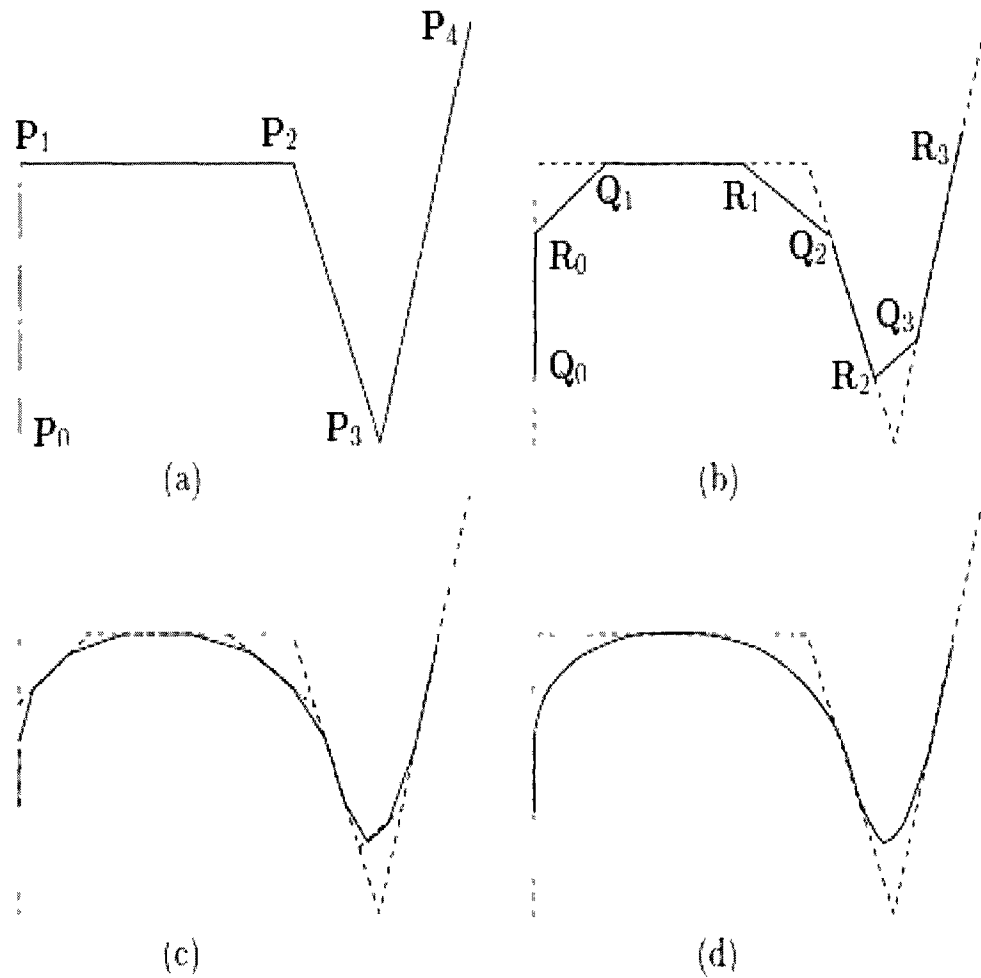


Figure 2 Subdivision of Open Loop using Chaikin's Corner Cutting Algorithm [16]

By applying Chaikin's algorithm, a quadratic B-spline curve is generated from a polygon after its corners are cut successively in during refinement. Because of this refinement scheme, two new points are inserted on each polygon leg.

### 2.3 Topological Rules for Chaikin's Algorithm

Chaikin's subdivision scheme is carried out by cutting off the corners of the initial control mesh. Figure 3 depicts the topological rules that should be followed while applying Chaikin's corner cutting scheme. Every corner in the initial control mesh yields two new vertices. These vertices are located on the edges joining the old vertex. Thus, a new edge of length shorter than the older edge is formed by joining these new vertices. The same procedure is continued for all the vertices thus reducing the length of all the edges.

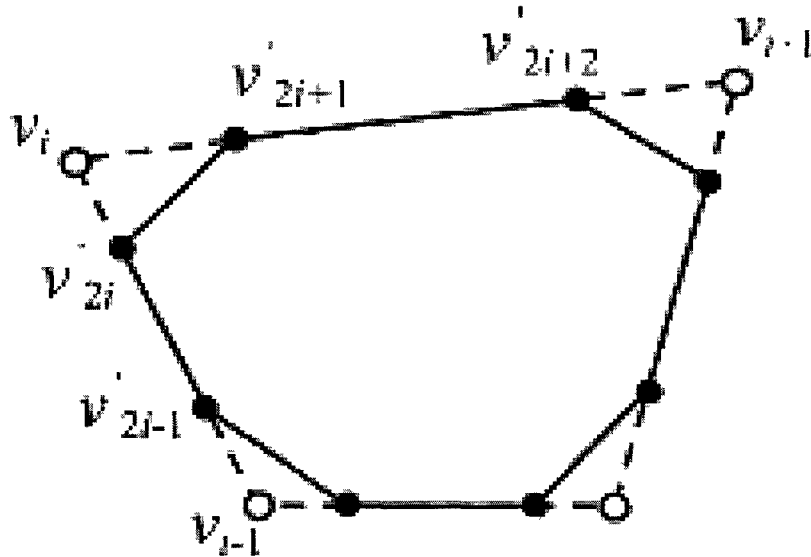


Figure 3 Topological Rules for Chaikin's Scheme [20]

**2.4 Geometric Rules for Chaikin’s Algorithm**

For a polygon with n vertices, the subdivision mask used for calculating the new control vertices by Chaikin’s algorithm is:

$$v'_{2i} = \frac{1}{4}v_{i-1} + \frac{3}{4}v_i \dots\dots\dots (2.1)$$

$$v'_{2i+1} = \frac{3}{4}v_i + \frac{1}{4}v_{i+1} \dots\dots\dots (2.2)$$

A new polygon is formed by joining the newly created points. By repeated corner cutting, quadratic B-spline curve is generated as the limit surface.

The number of vertices generated after each refinement of the mesh is usually double the number of the older vertices. Therefore, the user has a greater control over the control nodes if the aim is to optimize the object for shape.

## **CHAPTER III**

### **OTHER SUBDIVISION SCHEMES**

The smoothness and shape of the limit surface depends on two most important things: the type of refinement scheme applied, and the number of times the initial mesh is subdivided using the selected refinement scheme. In this chapter, an overview is given to describe the general classification of the subdivision schemes. Based on these classifications, several subdivision schemes are briefly discussed.

#### **3.1 Background and Definition**

The origin of subdivision surfaces can be traced back to the late 40s and 50s when a method called “corner-cutting” was used by G. de Rham for describing smooth curves. Catmull and Clark and Doo and Sabin made important contributions when they used subdivision for modeling surfaces. Since then subdivision surfaces have held an vital role in CAGD applications.

Subdivision is basically a method of subdividing a shape recursively till a surface of desired shape and smoothness is achieved. The process of subdivision involves refining a given polygonal mesh on the surface. The new polygonal mesh formed after each step of subdivision contains the new vertices whose positions are calculated with respect to the vertices of the polygonal mesh in the previous step. The new surface obtained after the subdivision of the given polygonal mesh, contains a mesh which is denser than the original one since it contains more polygonal faces. By iteratively applying this refinement scheme to the new mesh, the limit subdivision surface can be

obtained. The surface shape can be altered by choosing a different set of polygons for each mesh refinement.

Every Subdivision scheme is governed by two basic combination of rules: Topological rules and Geometric Rules.

Subdividing any given mesh into a refined mesh is governed by topological rules. After a subdivision scheme is chosen, new control nodes are inserted into the old edges or faces based on these topological rules. These topological rules also define the connection between the old vertices and the new vertices. [20]

Geometric rules help in calculating the exact locations of the new control vertices in the new refined mesh. Affine invariance, subdivision masks, proportionality of the mesh, and how the final surface behaves need to be considered while deciding on setting the geometric rules for mesh refinement. [20]

### **3.2 Classification of Subdivision Refinement Schemes**

The refinement schemes can be classified as follows:

1. On the basis of the refinement rule.
  - (a) Vertex split.
  - (b) Face split.
2. On the basis of the mesh generated.
  - (a) Triangular mesh.
  - (b) Quadrilateral mesh.
3. The refinement scheme can be approximating scheme or an interpolating one.
4. On the basis of smoothness of the limit surfaces for regular meshes ( $C_1$ ,  $C_2$  etc.)

The following table gives a general classification of the refinement schemes:

Table 1 Classification of Subdivision Schemes [19]

Face Split			Vertex Split
	<i>Triangular Meshes</i>	Quad. Meshes	Doo-Sabin, Midedge ( $C^1$ )  Biquartic ( $C^2$ )
<i>Approximating</i>	Loop ( $C^2$ )	Catmull-Clark ( $C^2$ )	
<i>Interpolating</i>	Mod. Butterfly ( $C^1$ )	Kobbelt ( $C^1$ )	Chaikin's Algorithm

### Type of Mesh

A subdivision scheme depends mainly on the control mesh. A control mesh is formed by joining a set of regularly spaced vertices on the plane of the surface. But the face of the mesh can be formed in various ways depending on the order in which the vertices are joined. For a regular polygon, the face polygon can be chosen by selecting so that they form (1) quadrilaterals, (2) triangles and (3) hexagons. Meshes with hexagonal polygons are not commonly used. Quadrilateral and triangular meshes are the most convenient and widely used.

### Face Split and Vertex Split

After deciding the type of mesh to be used, it is required to define how the refined mesh is related to the older mesh. A refined mesh can be generated by two approaches: (a) *face*

*split* or *primal*, (b) *vertex split* or *dual*. Figure 4 [19] depicts the face split scheme. For a *face split scheme*, every face of triangular or quadrilateral mesh can be subdivided into four faces. In case of a quadrilateral mesh, an extra vertex is inserted for each face. For a *vertex split scheme* (figure 4), new vertices are generated for each old vertex, one on each face adjacent to the old vertex. As a result of this scheme, new faces are obtained for every edge and also for all vertices. The new vertices that are created all have a valence of 4. Vertex split scheme for triangular meshing result in non-nesting hexagonal meshes. Quadrilateral meshing overcomes this drawback as both the face split and vertex split schemes can be applied to it.

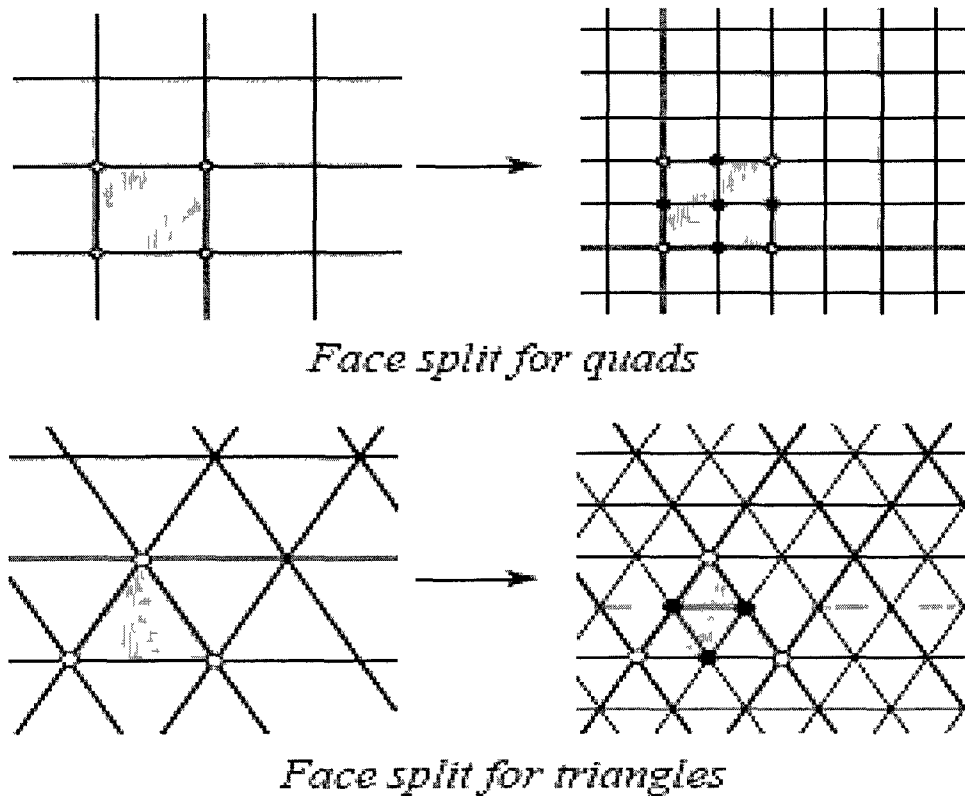


Figure 4 Face Split Scheme. [19]



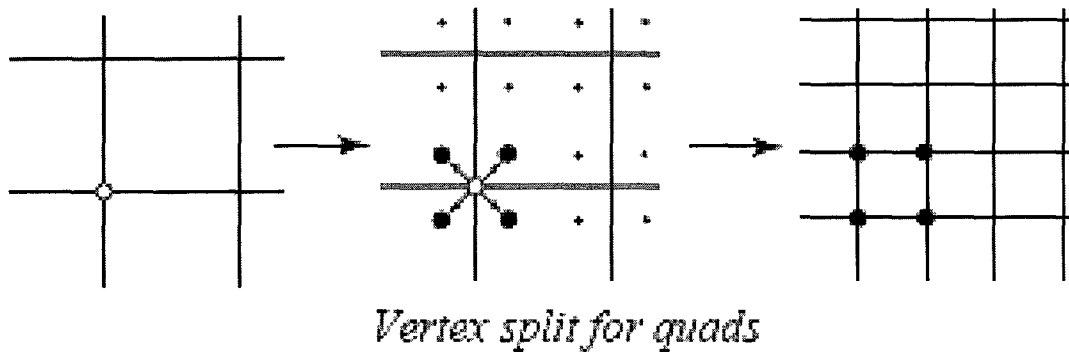


Figure 5 Vertex Split Scheme. [19]

### Approximating and Interpolating Schemes

Face split schemes can be both approximating or interpolating. The vertices in the original mesh also form the vertices of the refined mesh. A certain sequence of control points is defined for each old vertex corresponding to various subdivision levels. The scheme is approximating if all these points are not in the same sequence whereas the scheme is called interpolating if these control points are in the same order. Interpolating scheme is convenient to use since control points in the original mesh remain the same control points in the limit surface. However, the quality of the surface generated from interpolating scheme is not as good as the surface generated from an approximating scheme. Approximating schemes also converge faster than the interpolating schemes.

### 3.3 Doo-Sabin Subdivision Scheme

Doo-Sabin scheme [24] is an approximating refinement scheme that was introduced by Donald Doo and Malcolm Sabin.

The Doo-Sabin subdivision scheme is based on Chaikin's "corner cutting" algorithm. It is a generalized tensor product form of algorithm proposed by Chaikin.

### 3.4 Topological Rules of Doo-Sabin Subdivision Scheme

The surfaces generated as a result of Doo-Sabin subdivision scheme are generalized version of homogeneous bi-quadratic B-spline surfaces. For a polygon with  $n$ -vertices, the basic principle to subdivide it with this scheme is to insert  $n$  new vertices. These new vertices are calculated by a set of geometric rules which are as follows:

1. Consider vertex  $V_1$  on the initial mesh (figure 6 a) which is located on face  $F$  and is at the meeting point of the adjacent edges. A new point 1 is created such that it is a weighted average of (a)  $V_1$ , (b) the center points of edges ( $V_1-V_2$  and  $V_1-V_{10}$ ) which are adjacent to  $V_1$ , and (c) the face point which is mean of all the points of that face  $F$ . The same process can be repeated for all the vertices to obtain all the new points (Figure 6 b).
2. Again, consider the face  $F$  containing all the new points 1, 2, 3, 4. Connecting all these points will form a polygon which now becomes a new face on the refined surface. The polygons formed in the process are called Face-polygons (or F-polygons). The same procedure can be repeated for all other faces (Figure 6 c).
3. A vertex is usually common to several faces. Consider a vertex such as  $V_{11}$  as the same. By connecting all the points (3, 8, 17, 22 ) close to this vertex  $V_{11}$ , a new polygon will be formed. Now, this polygon becomes the face in the subdivided surface. The polygons formed in the process are called Vertex-polygons (or V-polygons). The same procedure can be repeated for all other faces (Figure 6 e).

4. Consider the edge  $V_{10}$ - $V_{11}$  in the initial mesh. There are two faces which are adjacent to this edge and contain the new points. A new polygon is formed by joining all the new points (4, 3, 22, 21) around this edge. This polygon then becomes the face in the new surface. The polygons formed in the process are called Edge-polygons (or E-polygons). The same procedure can be repeated for all other faces (Figure 6 d).

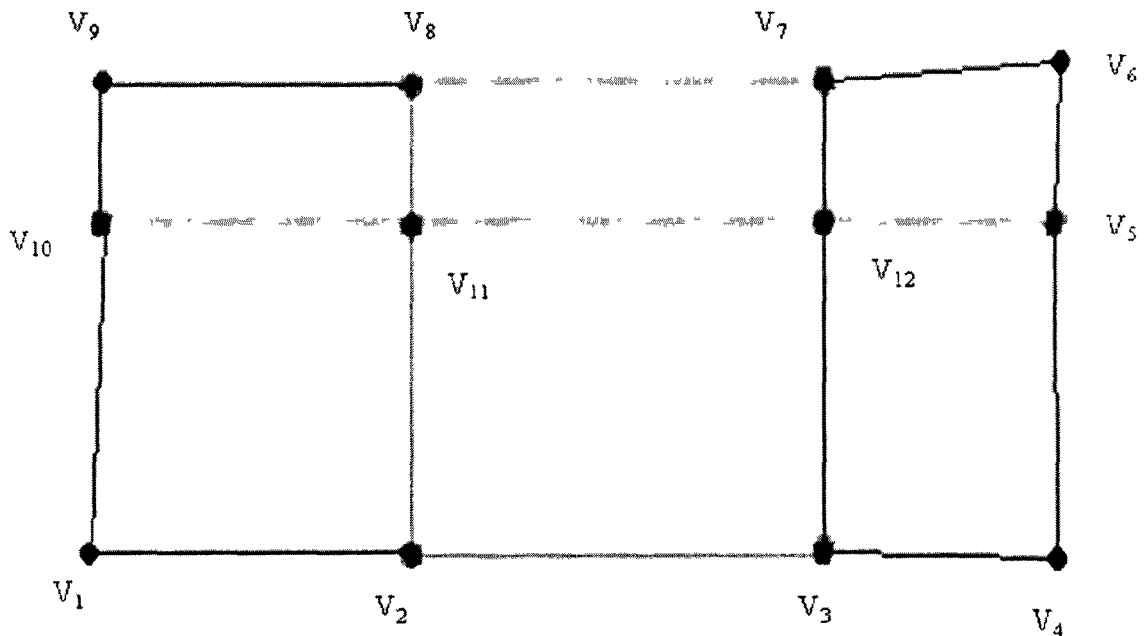


Figure 6 (a) Initial Polygon

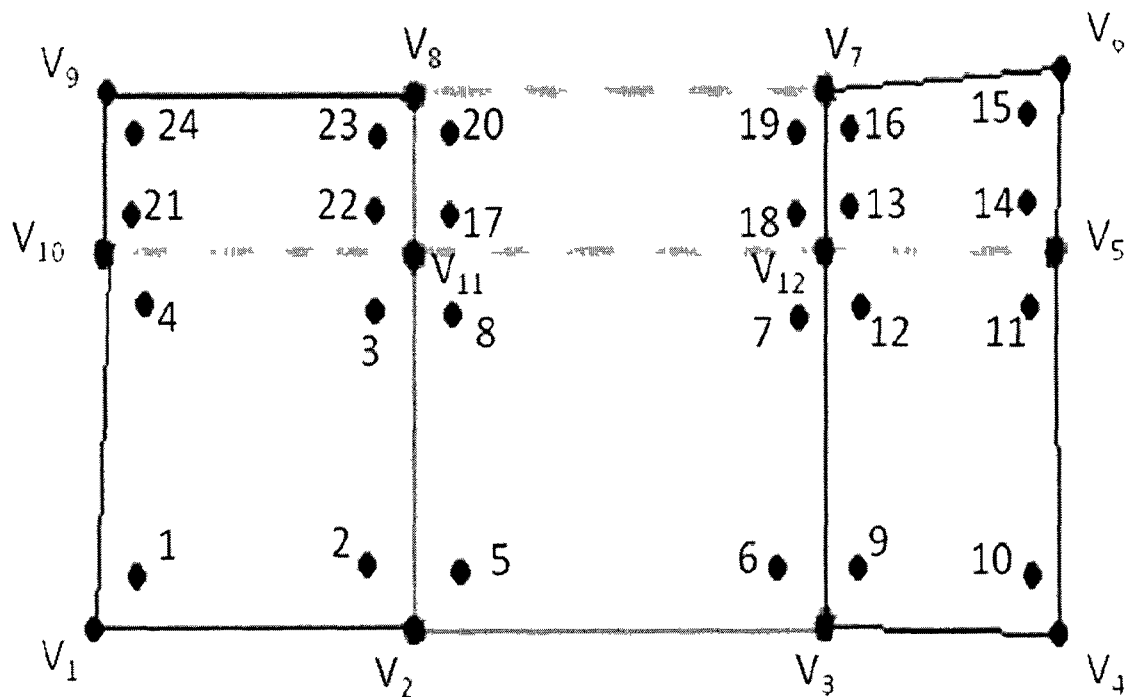


Figure 6 (b) Locating the Points

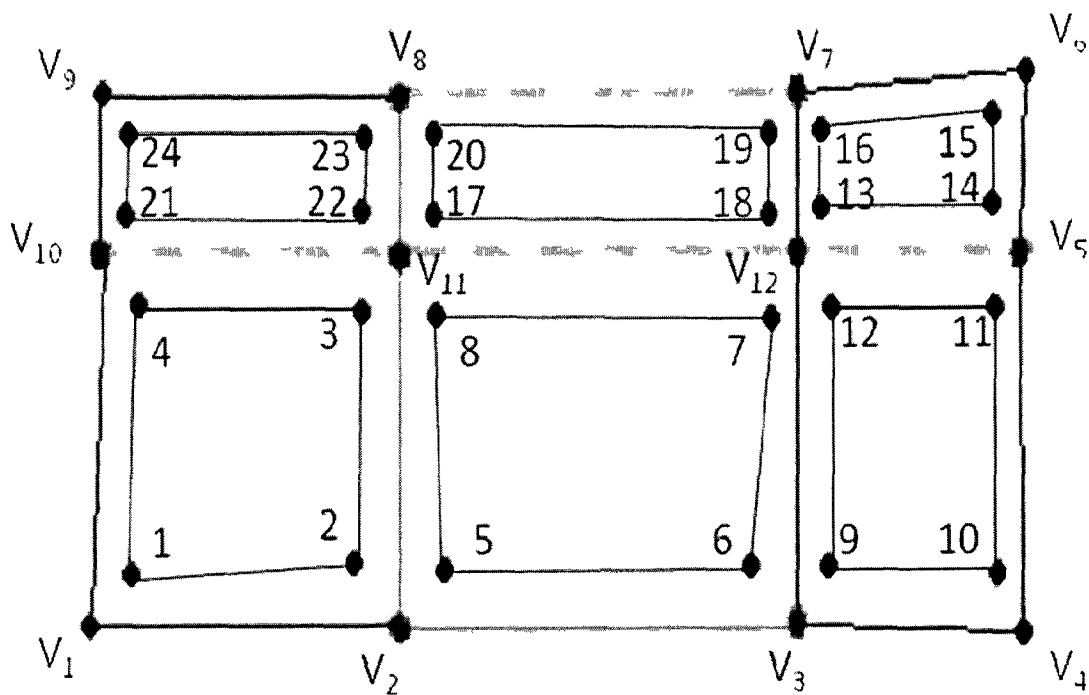


Figure 6 (c) Face Polygons

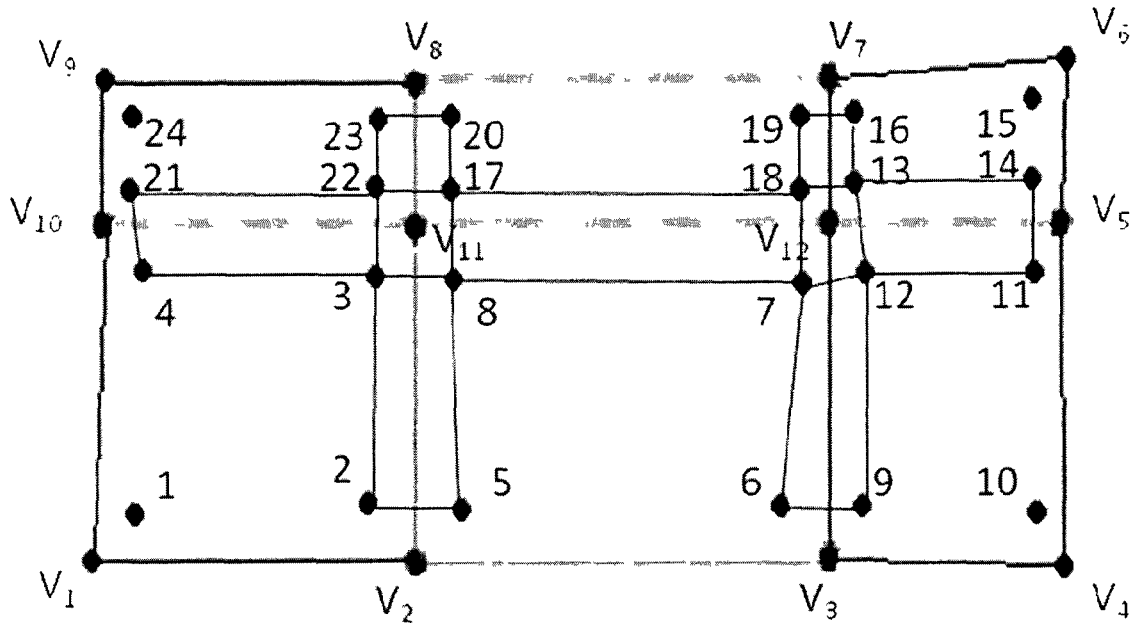


Figure 6 (d) Edge-Polygons

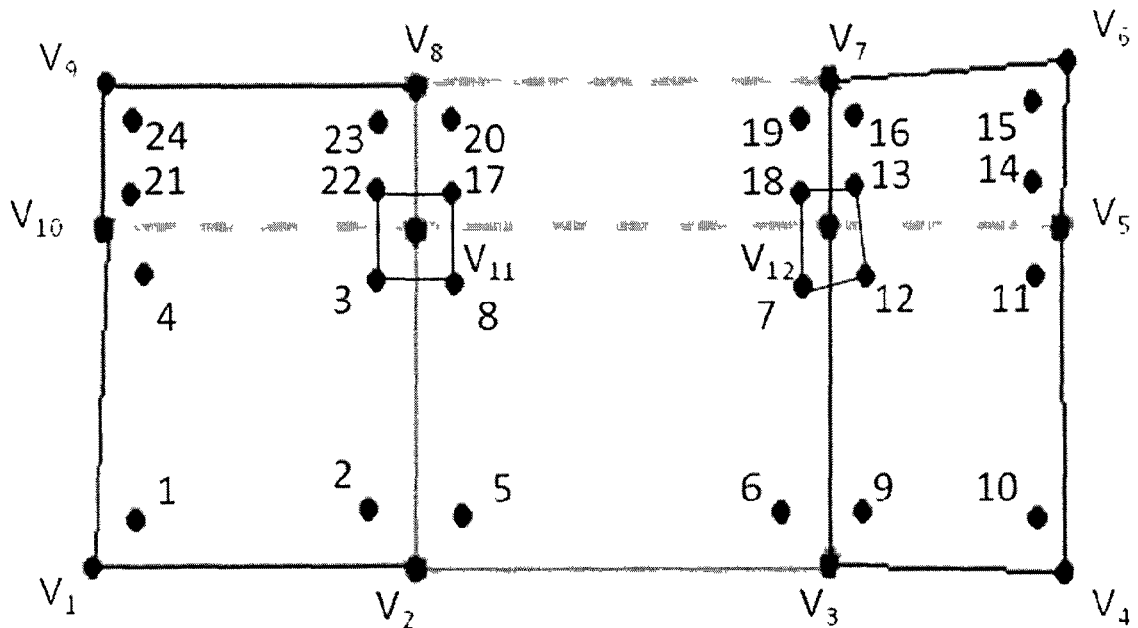


Figure 6 (e) Vertex-Polygons

To get the first refinement of the initial mesh, all the polygons formed by the above steps are joined. This will give a new refined surface to which the refinement scheme can be applied again in the same manner to refine it further. This is shown in figure 6 (f).

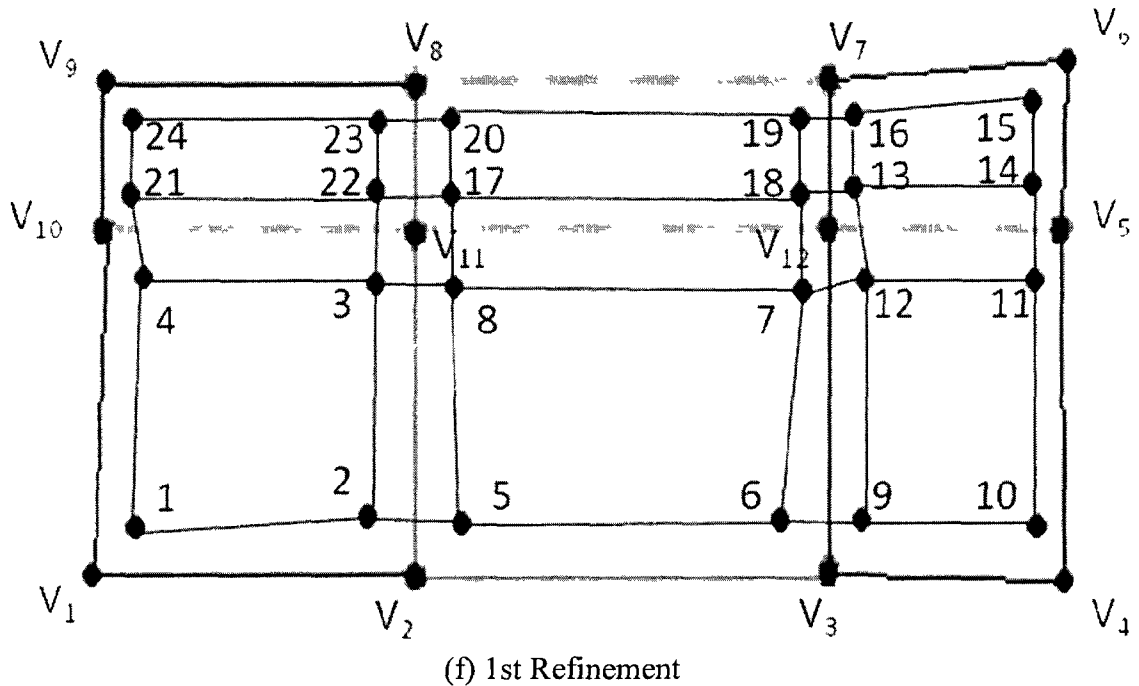


Figure 6 (a)-(f) Doo-Sabin Subdivision Steps and 1st Refinement.

### 3.5 Geometric Rules of Doo-Sabin Subdivision Scheme

Doo-Sabin surfaces only require one subdivision mask. The same subdivision mask can be used for all the faces.

For any regular face containing four sides, the subdivision mask can be described as follows:

$$v' = \frac{9}{16}v_0 + \frac{3}{16}v_1 + \frac{1}{16}v_2 + \frac{3}{16}v_3 \quad \dots\dots\dots(3.1)$$

For an irregular face, the subdivision mask can be defined as:

$$v' = \sum_{i=0}^{n-1} \alpha_i v_i \quad \dots\dots\dots (3.2)$$

To calculate the coefficient  $\alpha_i$ , Doo-Sabin suggested the following:

$$\alpha_i = \frac{n+5}{4n} \quad \text{for } i=0 \quad \dots\dots\dots(3.3)$$

Or,

$$\alpha_i = \frac{3 + 2\cos\left(\frac{2\pi i}{n}\right)}{4n} \quad \text{for } i=1, 2, \dots, n-1 \quad \dots\dots\dots (3.4)$$

The limit surface obtained by using Doo-Sabin subdivision scheme is a quadratic spline. Unlike Chaikin's algorithm, Doo-Sabin subdivision scheme cannot be applied to open loop. The reason behind this is that the Doo-Sabin subdivision scheme thrives on the polygons generated after each refinement.

### 3.6 Doo-Sabin and Mid-edge Scheme

A combination of Doo-Sabin and Mid-edge scheme [18] [19] is applied to the initial mesh in figure 6 is shown below in figure 7 (a) - (e).

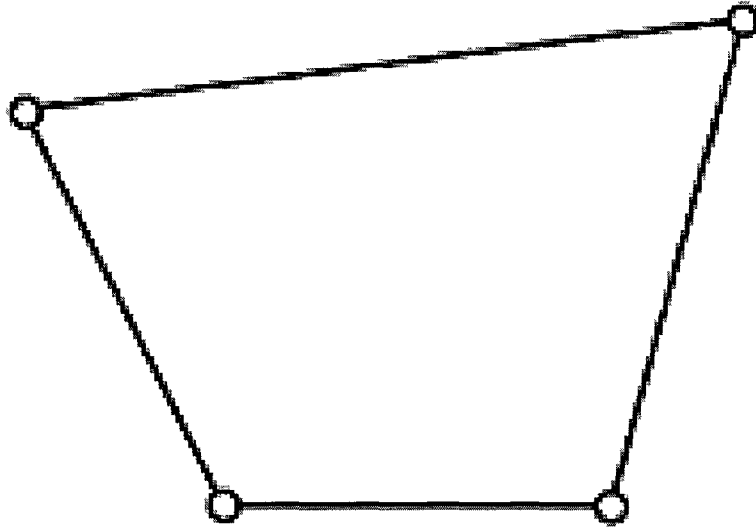


Figure 7(a)-(e) Doo-Sabin and Mid-edge scheme

To apply the Doo-Sabin/Midedge scheme, the initial four vertices are subjected to Doo-Sabin's subdivision mask. The subdivision mask is given by :

$$1 = \frac{9}{16}v_1 + \frac{3}{16}v_2 + \frac{1}{16}v_3 + \frac{3}{16}v_4 \quad \dots\dots\dots (3.5)$$

$$2 = \frac{9}{16}v_2 + \frac{3}{16}v_3 + \frac{1}{16}v_4 + \frac{3}{16}v_1 \quad \dots\dots\dots (3.6)$$

$$3 = \frac{9}{16}v_3 + \frac{3}{16}v_2 + \frac{1}{16}v_1 + \frac{3}{16}v_4 \quad \dots\dots\dots (3.7)$$

$$4 = \frac{9}{16}v_4 + \frac{3}{16}v_3 + \frac{1}{16}v_2 + \frac{3}{16}v_1 \quad \dots\dots\dots (3.8)$$

where 1, 2, 3, 4 are the four new vertices as shown in figure 7 (b).

Now, consider the new point 1. To calculate the point 1', auxiliary lines parallel to the adjacent edges are drawn from 1 to intersect the adjacent sides at 1a and 1b. Then, 1' becomes the centre point of the polygon  $V_1-1a-1-1b$ . Similarly, the points 2', 3', 4' are also located as shown in figure 7 (b).



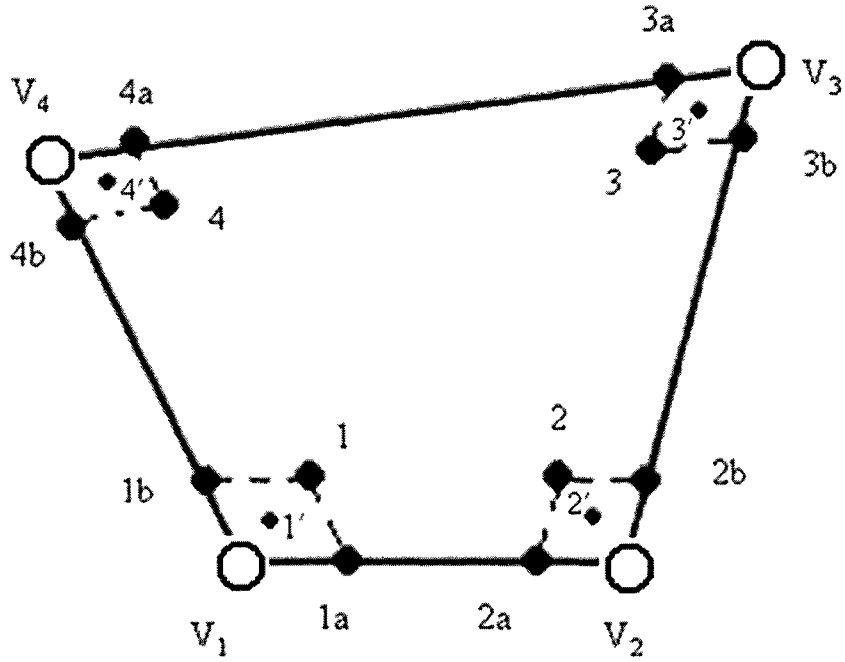


Figure 7 (b)

The 1<sup>st</sup> refinement of the initial mesh is shown in figure 7 (c). This is the polygon that is formed by joining all the new points. Each corner in the previous mesh generates three new vertices after each refinement. By following the same procedure, next refinement is computed as shown in figure 7 (e).

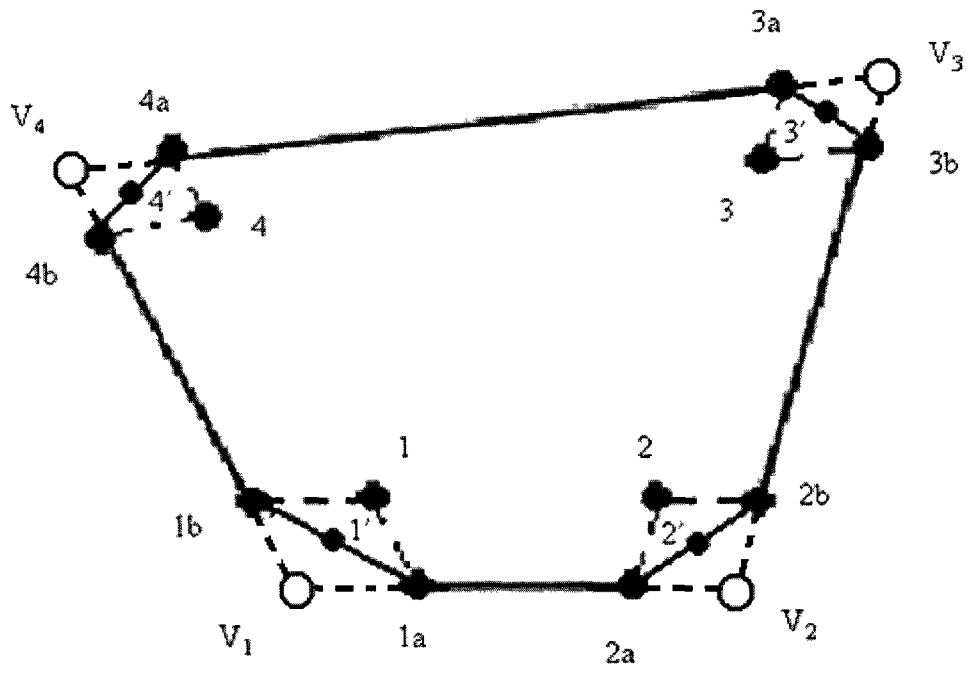


Figure 7 (c)

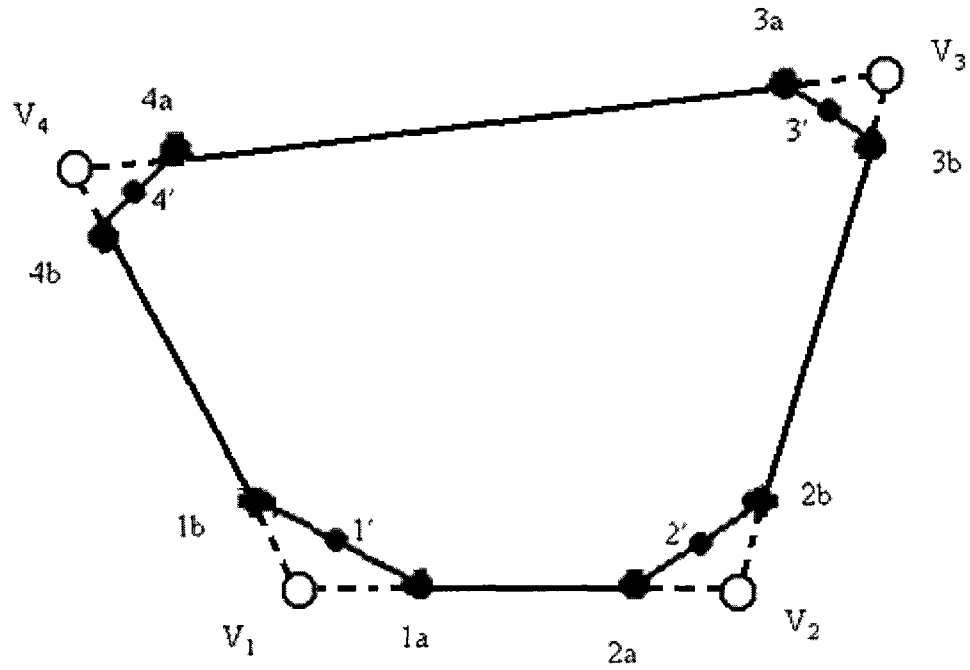


Figure 7(d)

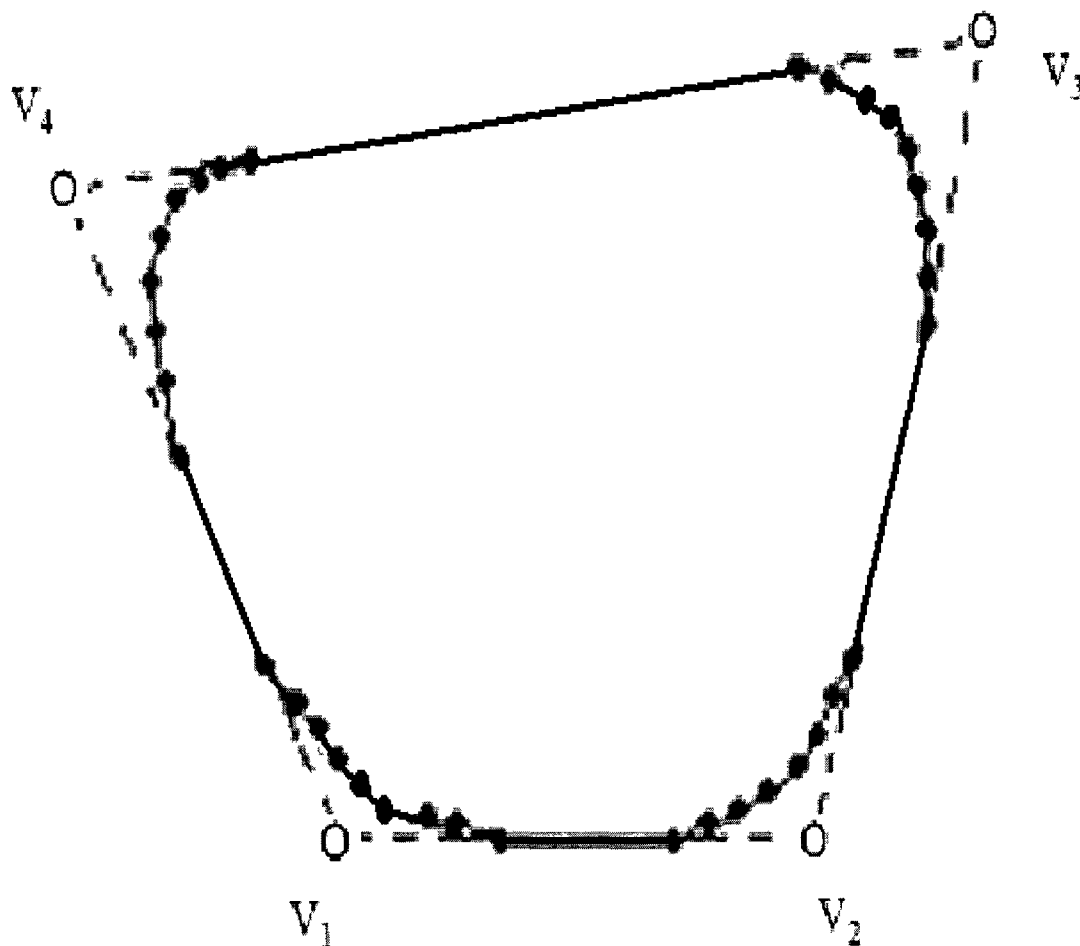


Figure 7 (e) 2<sup>nd</sup> Refinement of Initial Mesh

### 3.7 Loop Subdivision Scheme

The Loop subdivision scheme was formulated by Charles Loop. It is an approximating scheme. The surfaces generated by this scheme are C2 continuous at all the points except at the extraordinary points. At the extraordinary points, the surfaces generated are C1 continuous. By Loop's algorithm, each triangle in the original mesh is divided into 4 new triangles. This is also called binary Loop subdivision algorithm. [21]

In the binary Loop subdivision scheme, a vertex point is generated for all the old vertices in the initial mesh. Each edge of the triangle also generates a new edge point. These new vertices and edge points form the new refined small triangles for the next step of refinement.

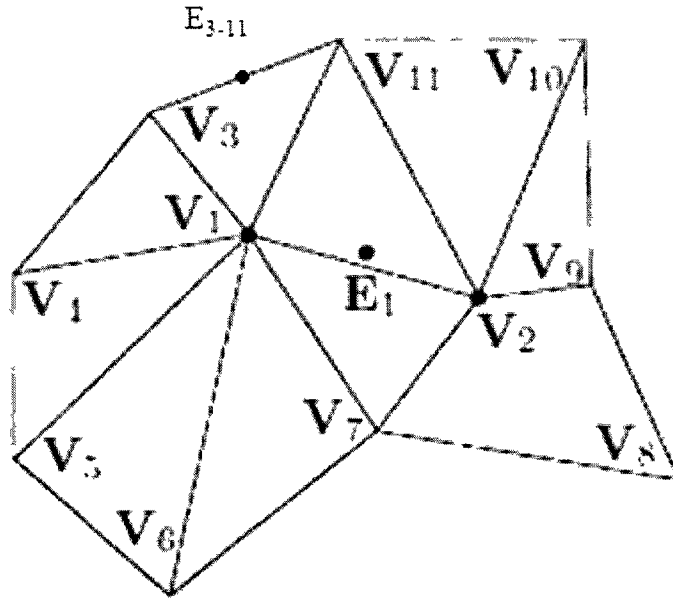


Figure 8 Binary Loop Subdivision [16]

To get a better understanding of the binary Loop subdivision, consider the edge  $V_1-V_2$  as shown in figure 8. This edge is shared by the triangles  $V_1-V_{11}-V_2$  and  $V_1-V_7-V_2$ . Now, the edge-point for this edge is calculated as:

$$E_1 = \frac{3}{8}(V_1 + V_2) + \frac{1}{8}(V_{11} + V_7) \dots\dots\dots(3.9)$$

However, for the edges that form the boundary, the edge point can be calculated by averaging the vertices forming the boundary edges. So for edge  $V_3$  and  $V_{11}$ , the edge point is calculated by

$$E_{3-11} = \frac{1}{2}(V_3 + V_{11}) \dots\dots\dots (3.10)$$

In this case, the edge point is always located on the edge.

Now to calculate the vertex point, consider the vertex  $V_2$ . The new vertex point for this vertex is calculated by

$$V'_2 = \frac{5}{8}V_2 + \frac{3}{8}Q_2 \dots\dots\dots(3.11)$$

Here  $Q_2$  is an average of all nodes that connect to  $V_2$ .

$$Q_2 = (V_7 + V_1 + V_{11} + V_{10} + V_9)/5 \dots\dots\dots(3.12)$$

For the vertices that are located on the boundary edge such as  $V_3$ , the new vertex point can be calculated as the weighted addition of

$$\frac{6}{8}V_3 + \frac{1}{8}(V_4 + V_{11}) \dots\dots\dots(3.13)$$

The main disadvantage of this method is that after some iterations, the surface is not smooth at “extraordinary” vertices. This is because the tangent plane loses its continuity at extraordinary vertices. An improvement to this scheme called the Ternary Loop method [22] was also introduced. The vertex point is calculated as the eweighted sum  $\alpha_n V_i + (1-\alpha_n)Q_i$ . Here  $n$  is the number of triangles,  $V_i$  is the vertex that  $n$  triangles share and  $Q_i$  can be calculated as the average of the vertices that surround  $V_i$ .  $\alpha_n$  is calculated by

$$\alpha_n = \left(\frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{n}\right)^2 + \frac{3}{8} \dots\dots\dots(3.14)$$

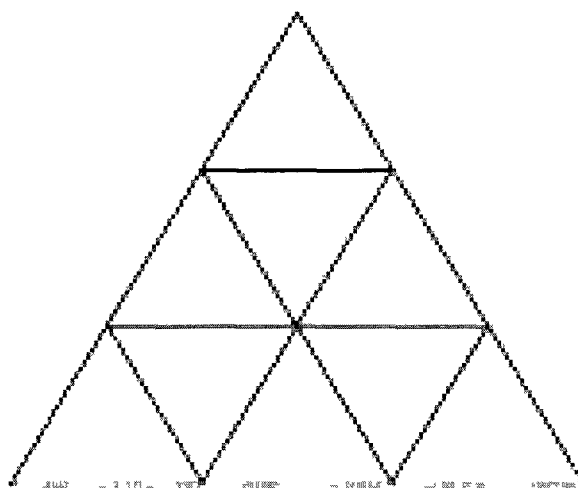


Figure 9 Ternary Loop Subdivision [16]

In this method, a triangle is divided into 9 smaller triangle (figure 9) and the new vertices and edges are calculated by using the suitable subdivision mask.

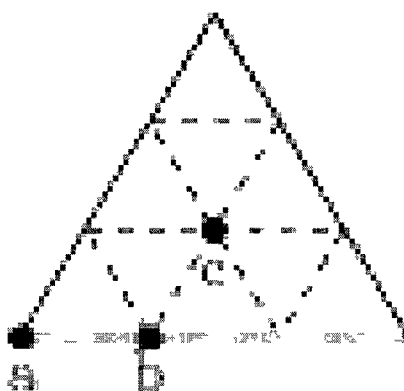


Figure 10 (a)

For subdividing a triangle into 9 small triangles, it requires one face point, 2 edge points for every edge and 3 new vertex nodes. This is shown in figure 10 (a).

In figure 10 (b), 'b' can be calculated by the subdivision mask shown. It can be calculated by adding weights of 7 vertices obtained from 6 triangles. To normalize these weights as displayed in figure 10 (b), the weights are divided by the total sum of weights which is 81. Similarly all the edge points are calculated.

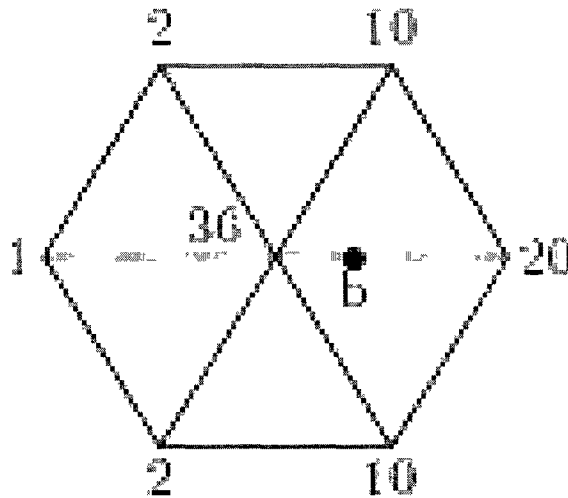


Figure 10 (b)

Figure 10 (c) shows the subdivision mask for calculating the face point 'c'. It can be calculated by adding weights of 6 vertices obtained from 4 triangles. To normalize the weights, the weights have to be divided by their total sum of 27. All the triangles which share same vertex determines the position of the vertex point. If  $n$  triangles share the same vertex 'a', then all the vertices that lie on the same edge as the vertex 'a' have the weights  $(1-\alpha)/n$  and 'a' has the weight  $\alpha$ . It was found that  $\alpha=5/9$  was the most suitable weight which worked in most of the cases.

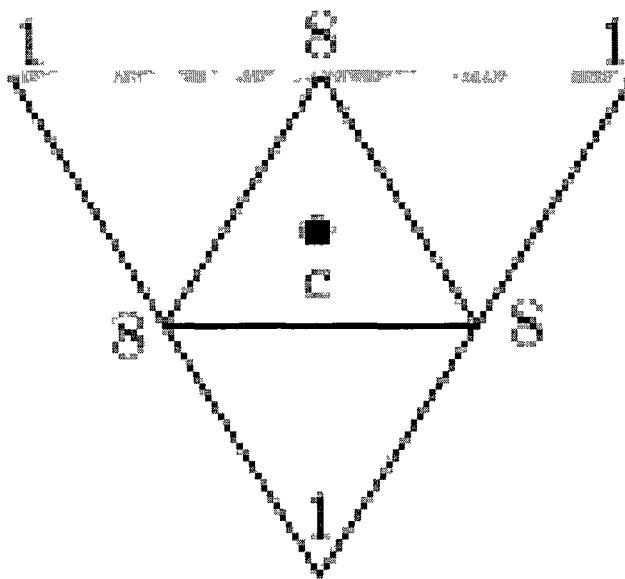


Figure 10 (c)

Figure 10 (a)-(c) Subdivision Masks for Ternary Loop Subdivision [16]

For applying the binary Loop subdivision scheme to the initial mesh shown in figure 6, the initial mesh is divided into 2 triangles as shown in 11 (b). This is because the Loop subdivision algorithm works only for triangles.



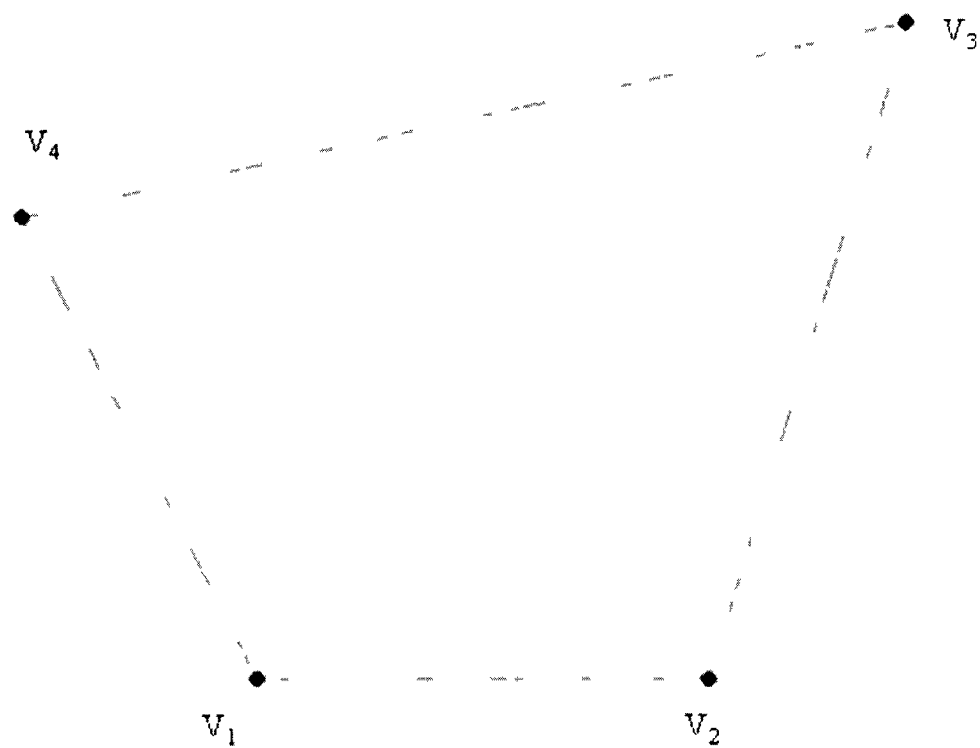


Figure 11 (a)

Figure 11 (a)-(d) Binary Loop Subdivision for initial mesh with four points

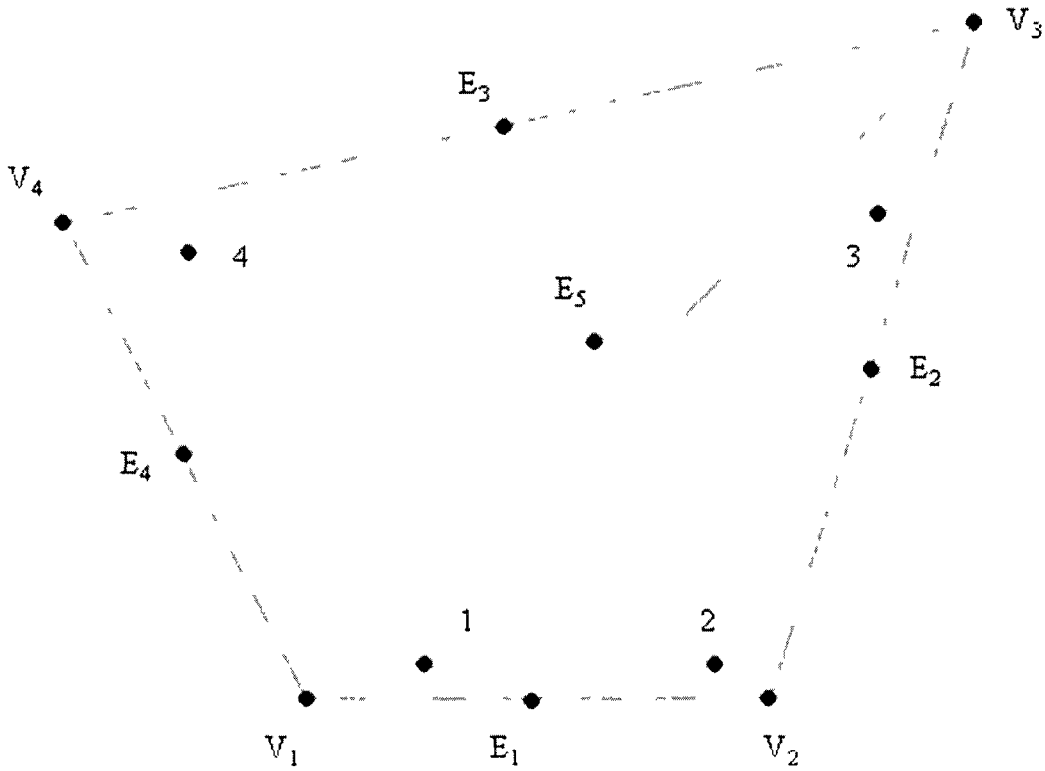


Figure 11 (b)

Since the edges  $V_1-V_2$ ,  $V_2-V_3$ ,  $V_3-V_4$  &  $V_4-V_1$  form the boundary edges, their edge points can be located as the midpoint each edge namely  $E_1$ ,  $E_2$ ,  $E_3$  and  $E_4$  respectively. For edge  $V_1-V_3$ , the edge point  $E_5$  can be calculated as:

$$E_5 = \frac{3}{8}(V_1 + V_3) + \frac{1}{8}(V_2 + V_4) \quad \dots\dots\dots(3.15)$$

The vertex points 1, 2, 3, 4 as shown in figure are generated from the old nodes  $V_1$ ,  $V_2$ ,  $V_3$  &  $V_4$  respectively. New vertex point 1 can be calculated as the weighted sum of neighboring boundary vertices.

$$1 = \frac{6}{8}V_1 + \frac{1}{8}(V_2 + V_4) \quad \dots\dots\dots(3.16)$$

Similarly, new vertex points 2, 3, and 4 are also located. The vertex points and edge points are joined as shown in figure 11 (c) to get the first refinement of the old mesh.

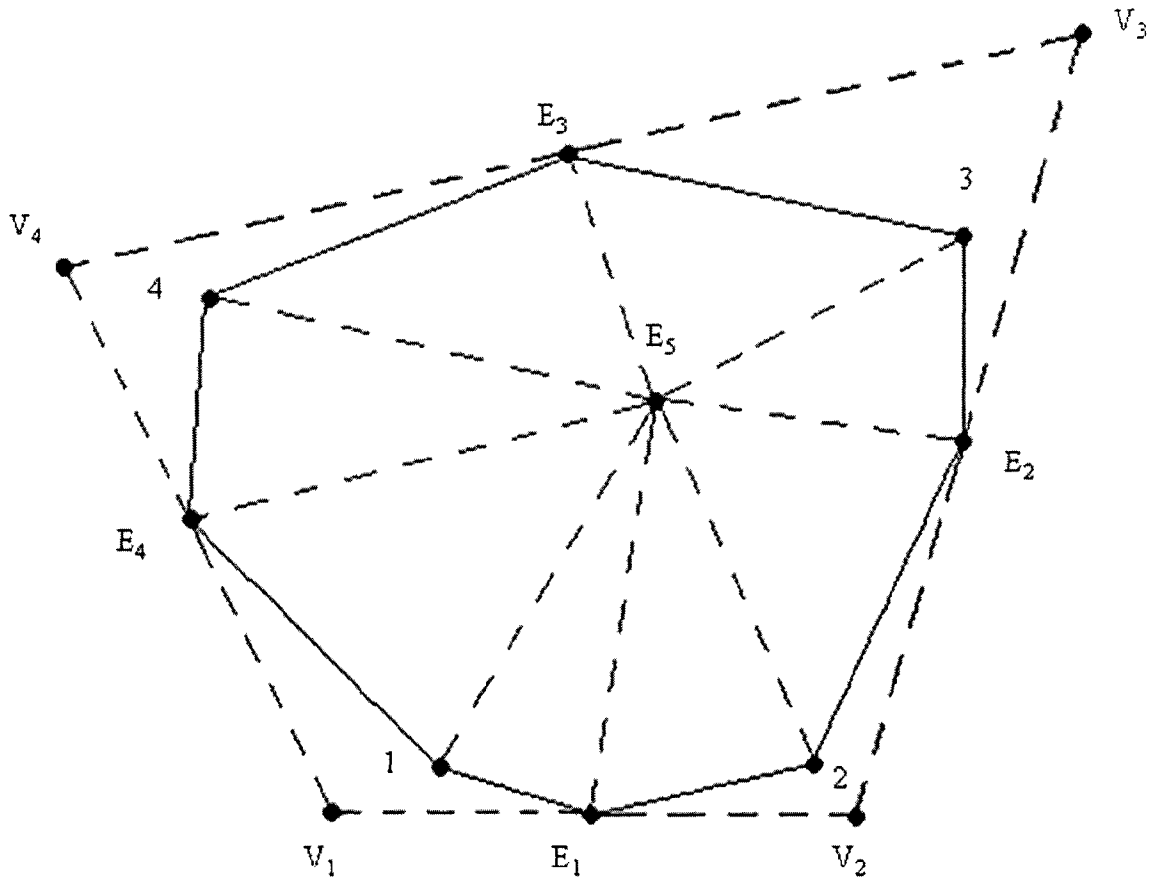


Figure 11 (c) 1<sup>st</sup> Refinement

Figure 11 (d) shows the second refinement of the initial mesh. For this refinement, the edge point \$E\_5\$ becomes an internal vertex. For an internal vertex, the new vertex is calculated by

$$E'_5 = \frac{5}{8} E_5 + \frac{3}{8} Q_5 \quad \dots\dots\dots(3.17)$$

Here,

$$Q_5 = (1 + E_1 + 2 + E_2 + 3 + E_3 + 4 + E_4) / 8 \quad \dots\dots\dots(3.18)$$

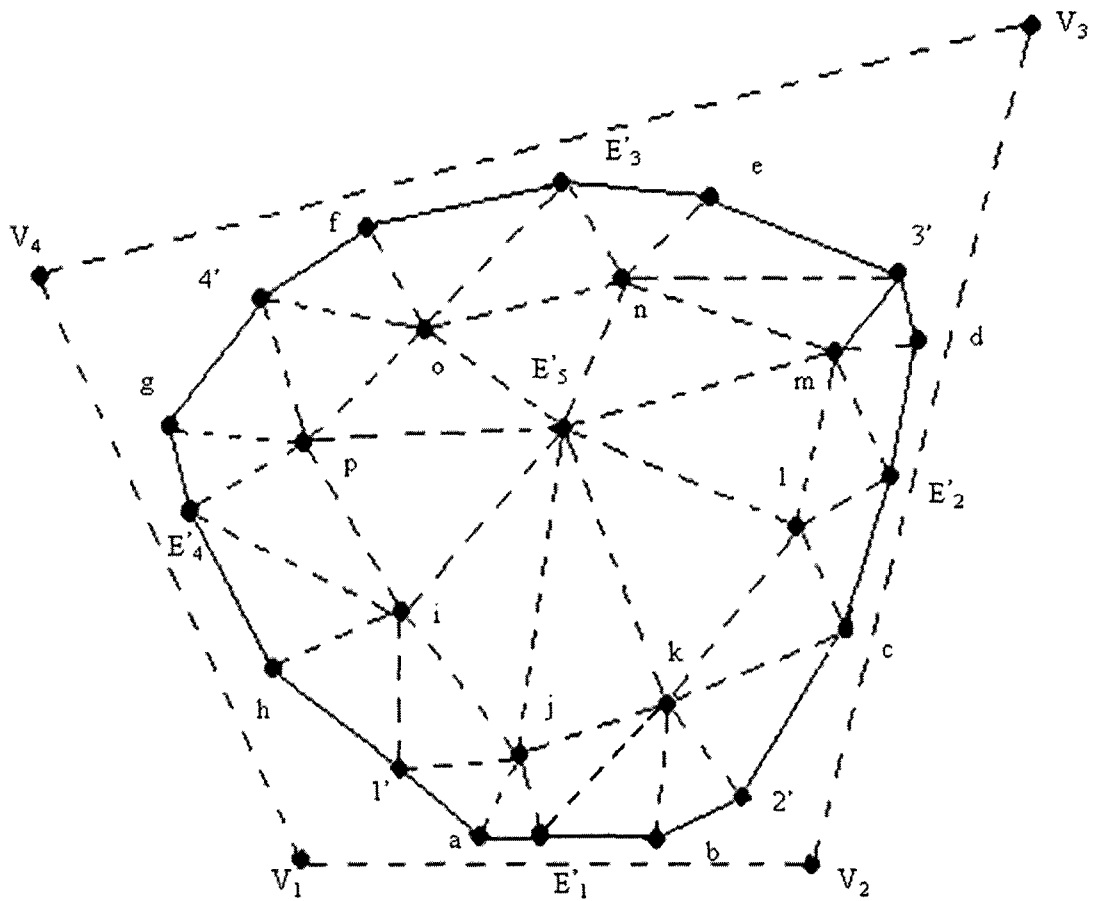


Figure 11 (d) 2nd Refinement

For the second refinement, the edge points and the vertex points are calculated as described earlier. After each refinement, the number of triangles becomes four times the triangles in the previous refinement step. By Loop's algorithm, the curve produced is cubic spline.

## CHAPTER IV

### SHAPE OPTIMIZATION OF COMPLIANT MECHANISMS

In this chapter, two examples of quadrilateral discretization and modified quadrilateral discretization for compliant mechanisms will be discussed. Compliant displacement inverter and amplifier, and compliant gripper are considered for both the cases of discretizations. Chaikin's algorithm will be applied on both the discretization conditions for both compliant mechanisms. MATLAB will be used for the application of Doo-Sabin scheme.

#### 4.1 Input

The inputs to a Chaikin's algorithm are the vertices of the polygonal mesh. In each refinement level of the scheme, vertices are the only input; the only difference is that the vertices will keep on changes in each step.

#### 4.2 Output

After the input vertices are fed to the refinement scheme, new positions of the new vertices are calculated by it. The old vertices are cut off and the new vertices are generated. The length of the edges formed by joining the new vertices goes on decreasing with every refinement. The process is continued till the limit surface is obtained.

The two examples will be discussed here in detail.

#### **Example 1: Compliant Displacement Inverter and Amplifier.**

##### **Case (a) : Regular Quadrilateral Discretization**

Step 1: The first step of the Chaikin's subdivision scheme is shown in figure 12.

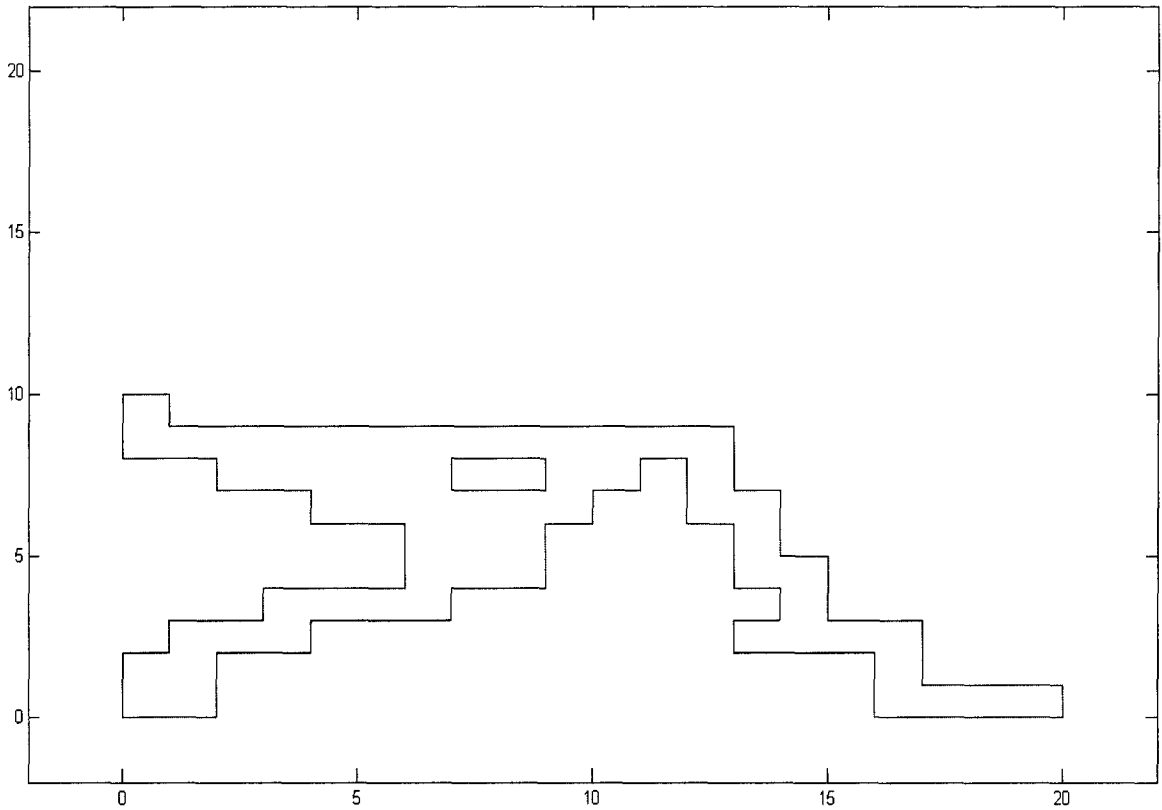


Figure 12 Initial Mesh

Figure 12 shows the initial mesh for the regular discretization of compliant displacement inverter and amplifier. The figure contains three outer open loops and 1 closed loop.

Step 2: Figure 13 shows the 1<sup>st</sup> refinement of the initial mesh.

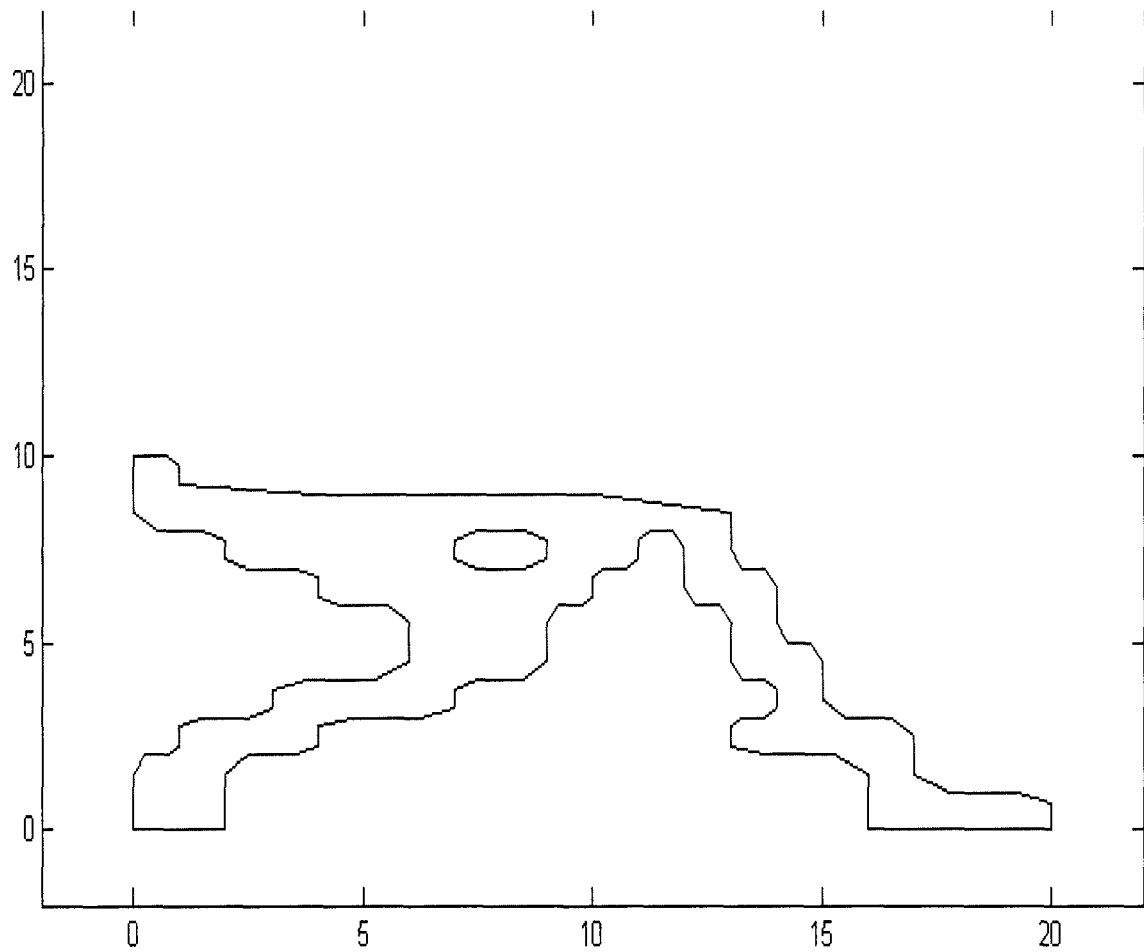


Figure 13 1st Refinement

Step 3: Figure 14 shows the 2<sup>nd</sup> refinement of the initial mesh.

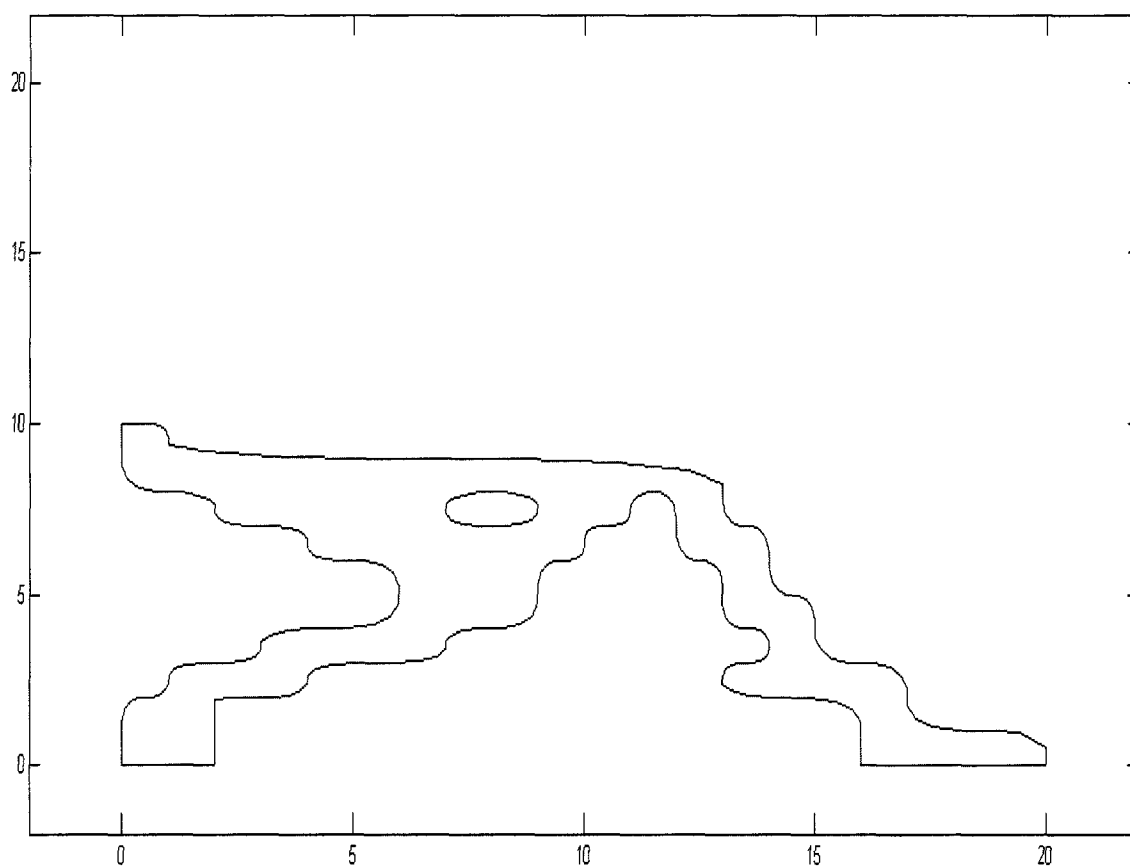


Figure 14 2nd Refinement



Step 4: Figure 15 shows the 3<sup>rd</sup> refinement of the initial mesh.

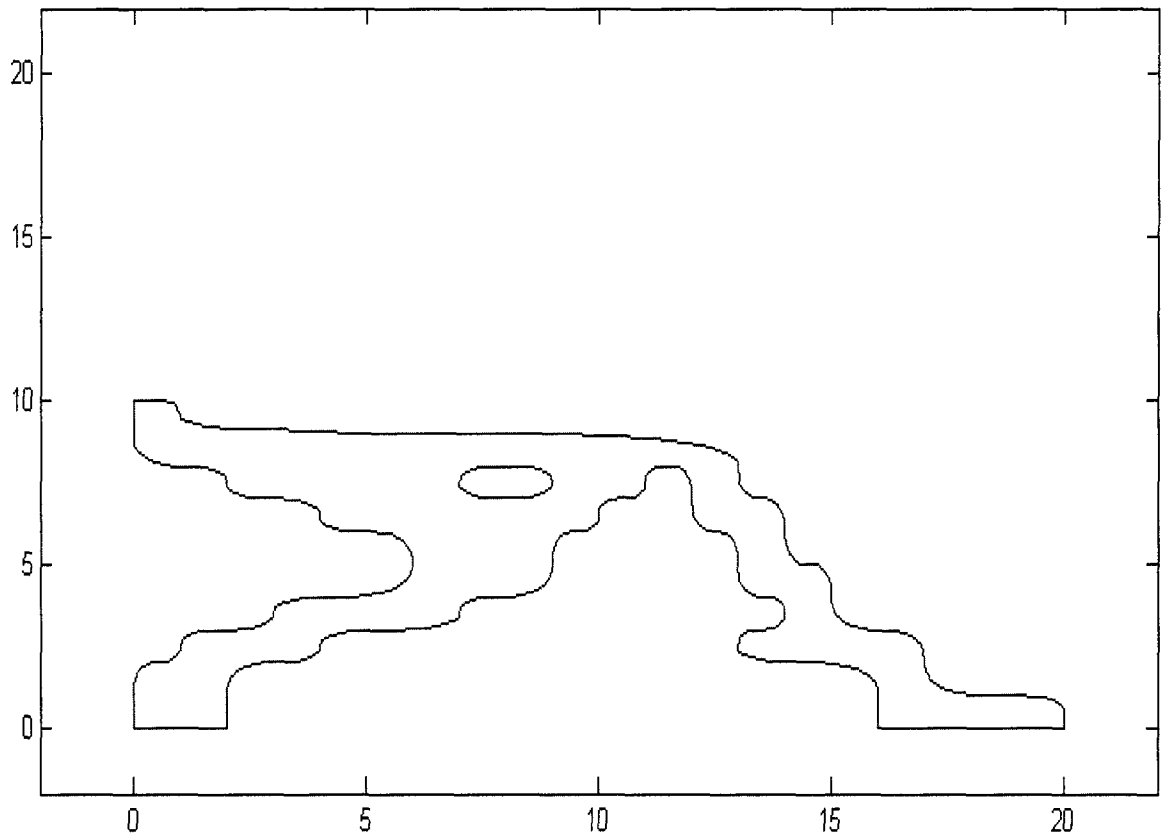


Figure 15 3rd Refinement

Step 5: Figure 16 shows the 4th refinement of the initial mesh.

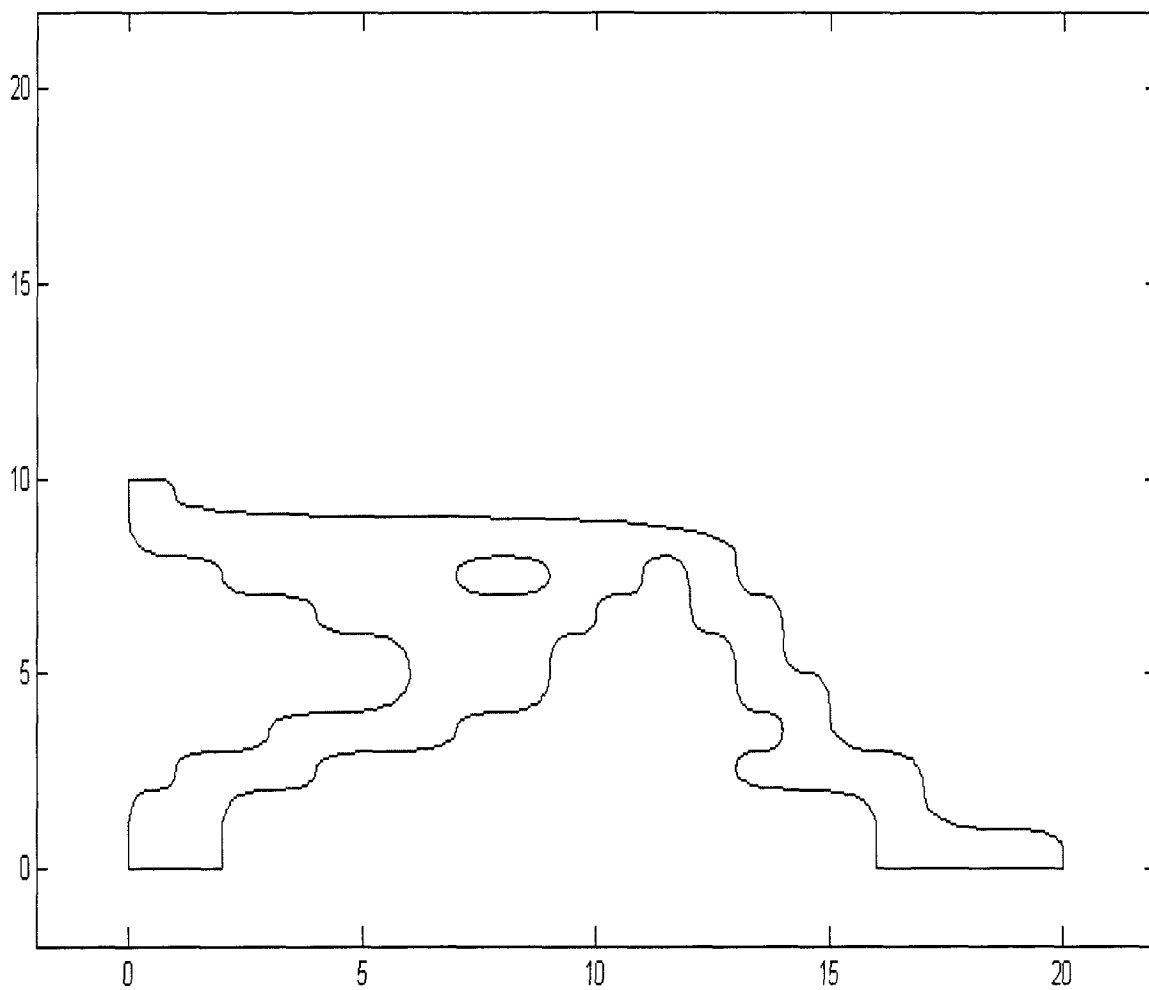


Figure 16 4th Refinement

Step 6: Figure 17 shows the 5th refinement of the initial mesh.

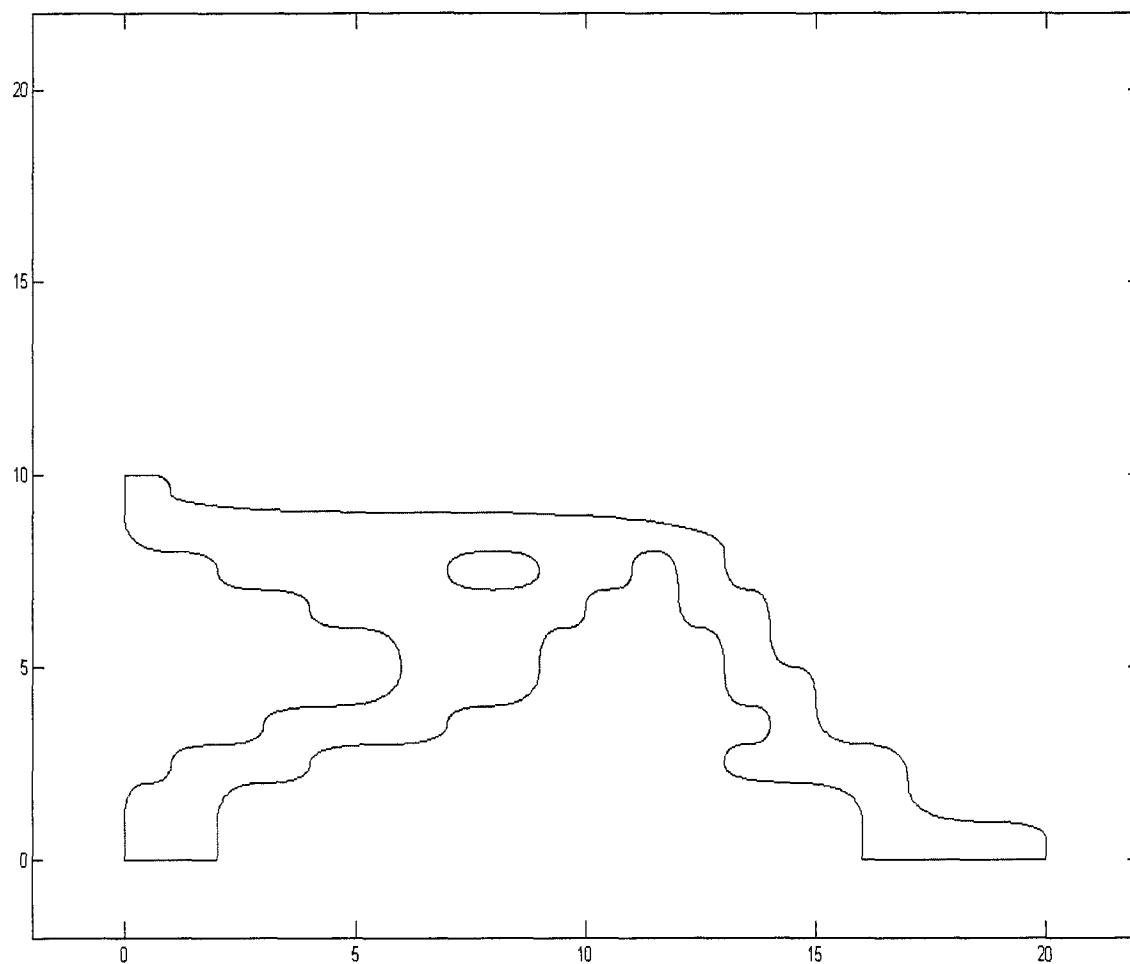


Figure 17 5th Refinement

**Case (b) : Modified Quadrilateral Discretization**

Step 1: Figure 18 shows the initial mesh.

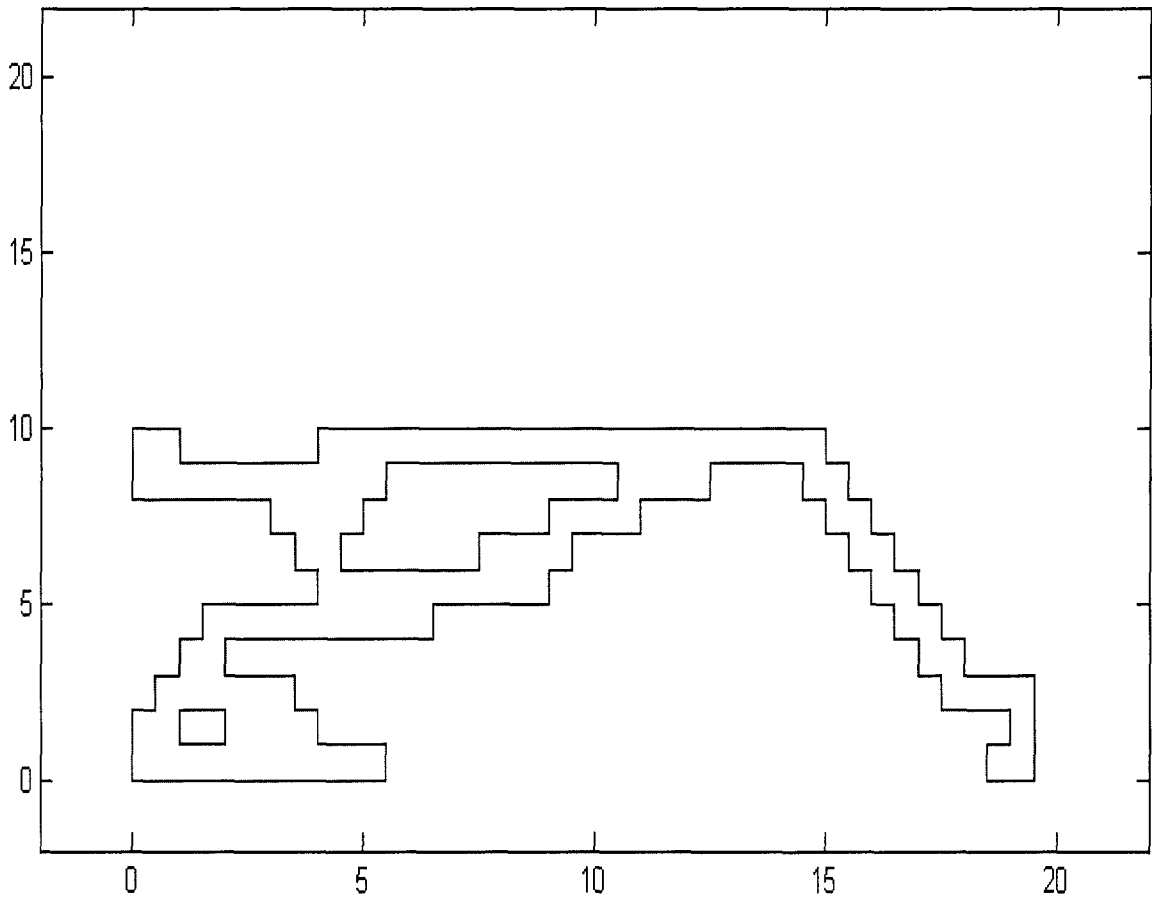


Figure 18 Initial Mesh

Step 2: Figure 19 shows 1<sup>st</sup> refinement the initial mesh.

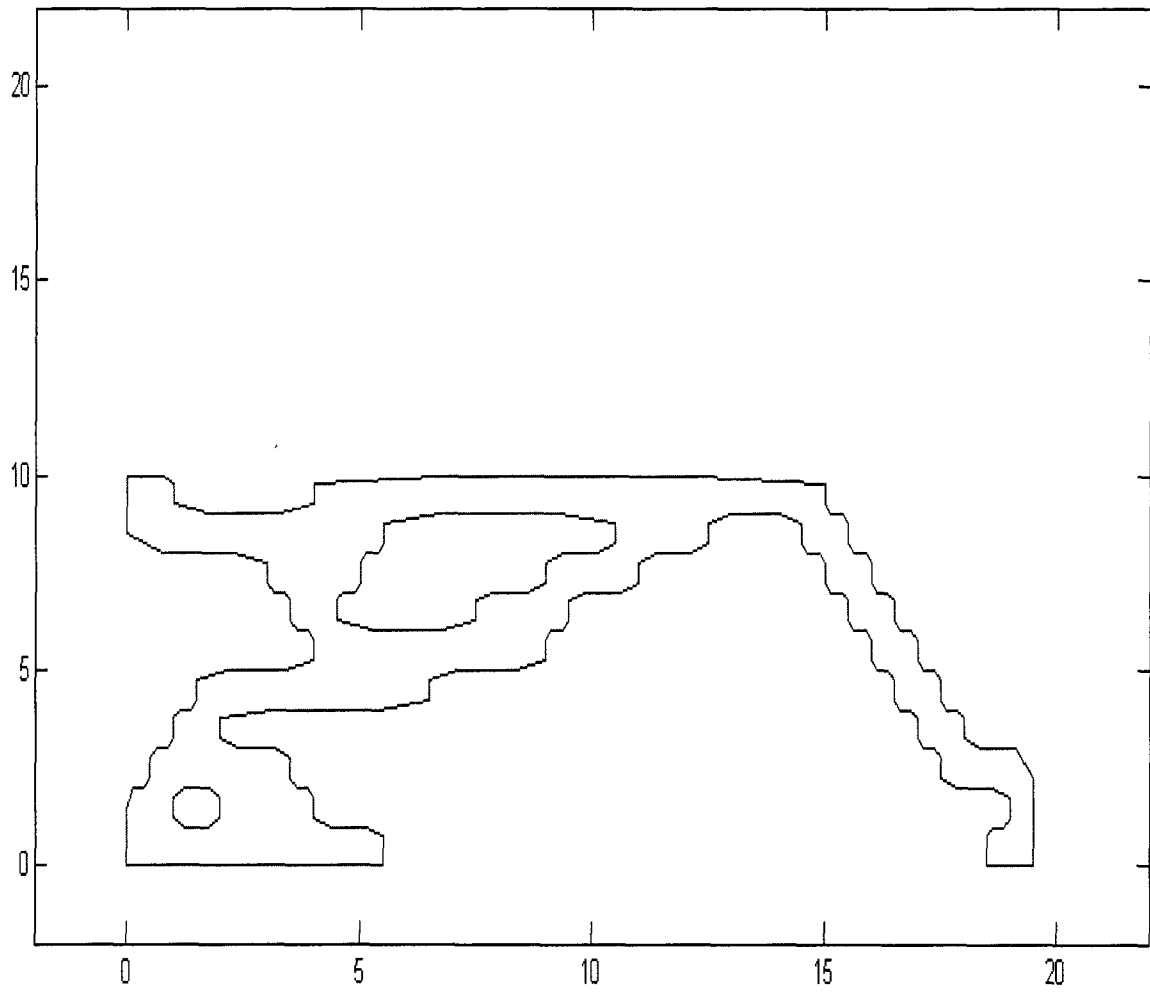


Figure 19 1st Refinement

Step 3: Figure 20 shows 2<sup>nd</sup> refinement the initial mesh.

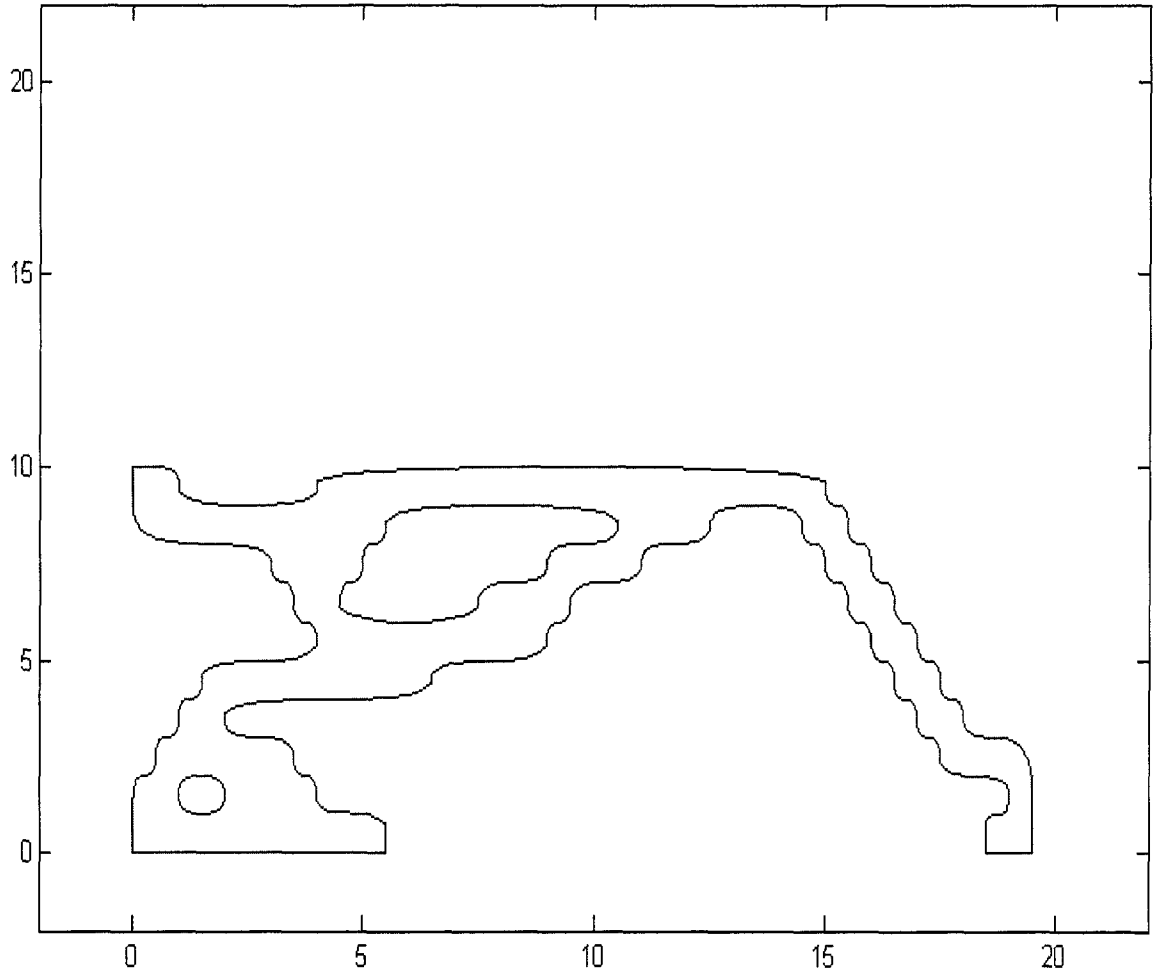


Figure 20 2nd Refinement

Step 4: Figure 21 shows 3<sup>rd</sup> refinement the initial mesh.

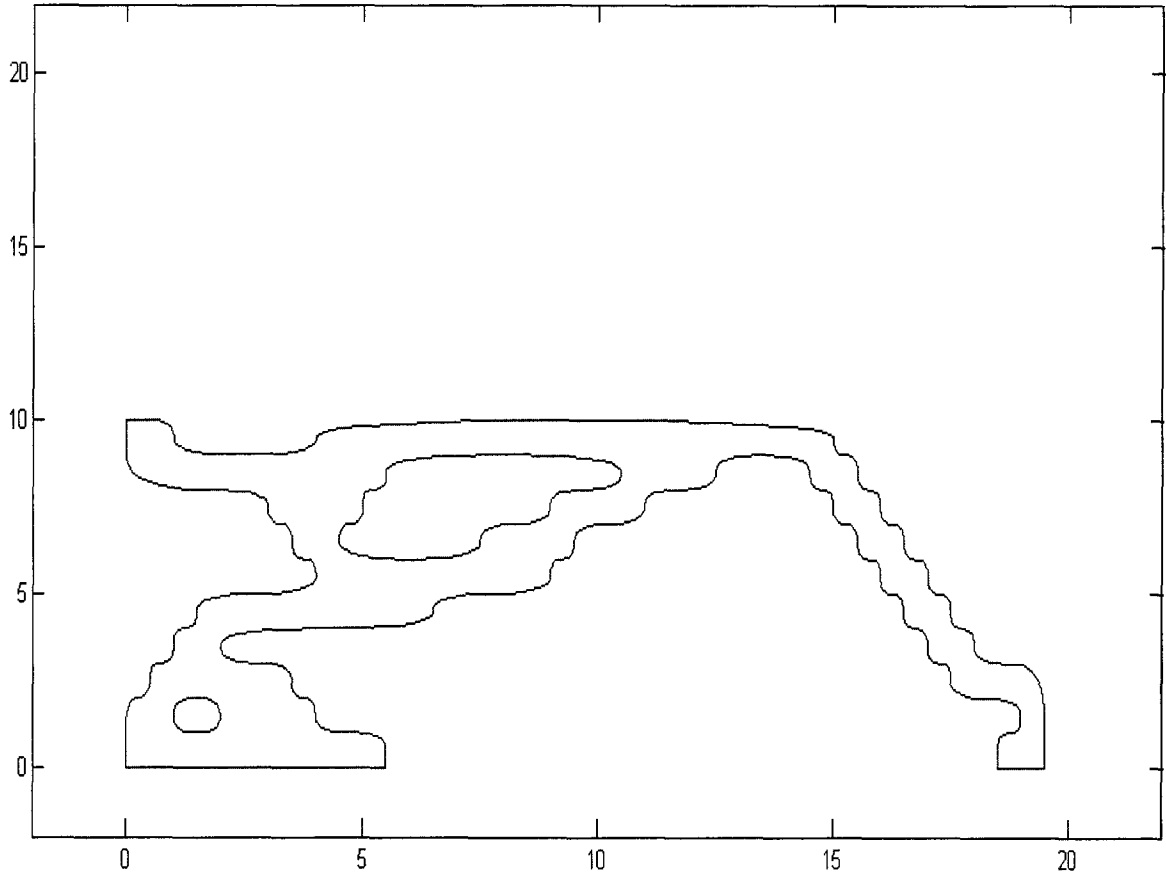


Figure 21 3rd Refinement

Step 5: Figure 22 shows 4th refinement the initial mesh.

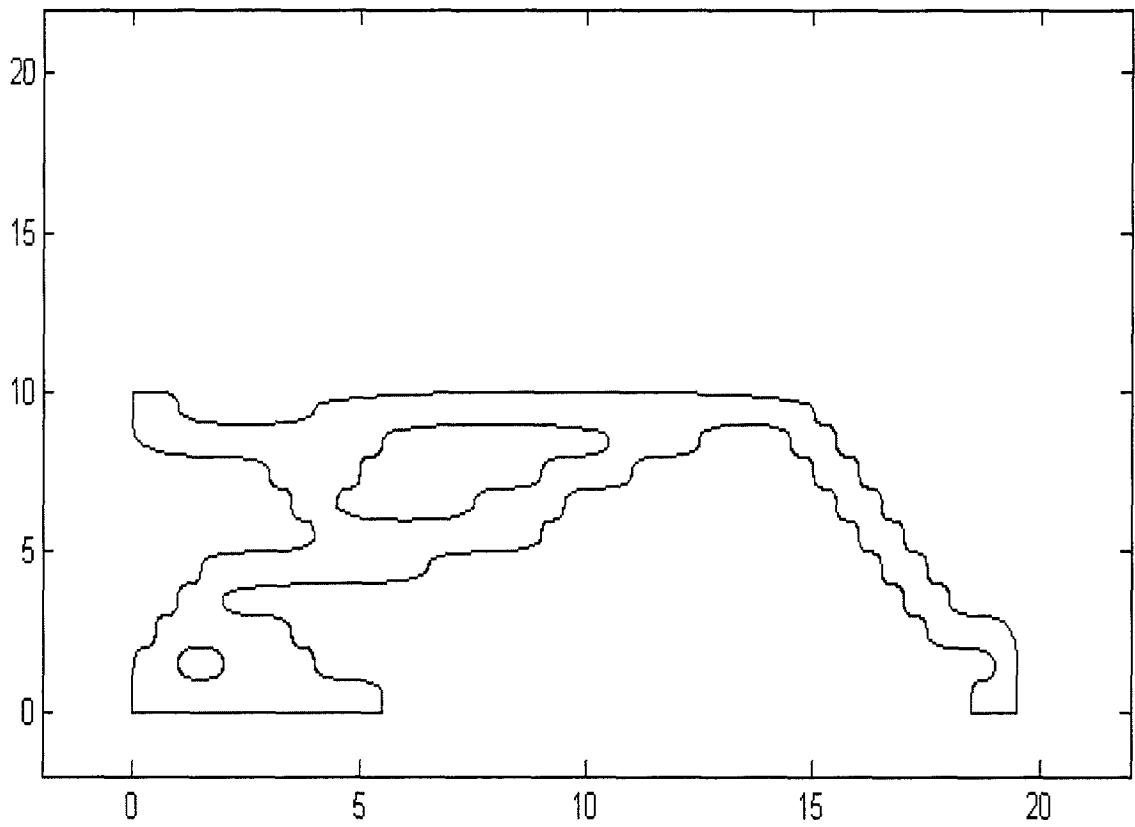


Figure 22 4th Refinement



Step 6: Figure 23 shows 5th refinement the initial mesh.

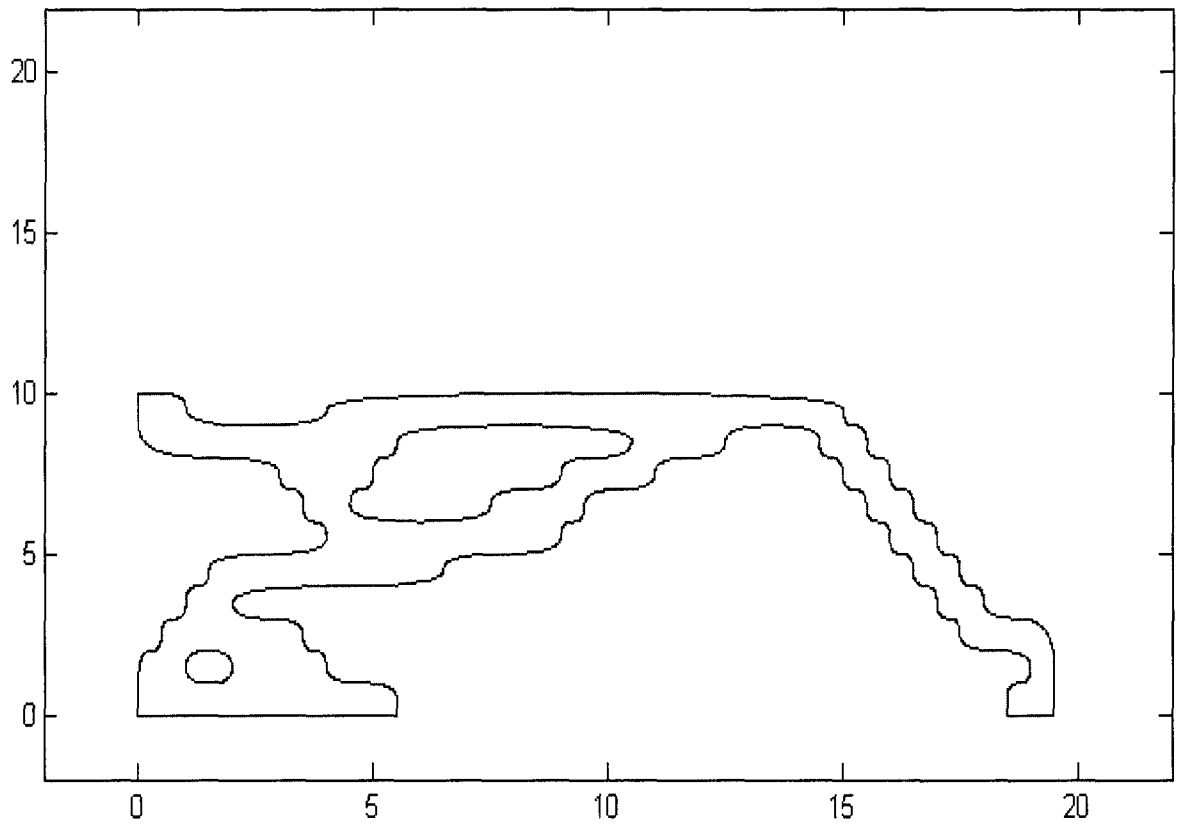


Figure 23 5th Refinement

**Example 2: Compliant Gripper.****Case (a) Regular Quadrilateral Discretization.**

Step 1: Figure 24 shows the initial mesh.

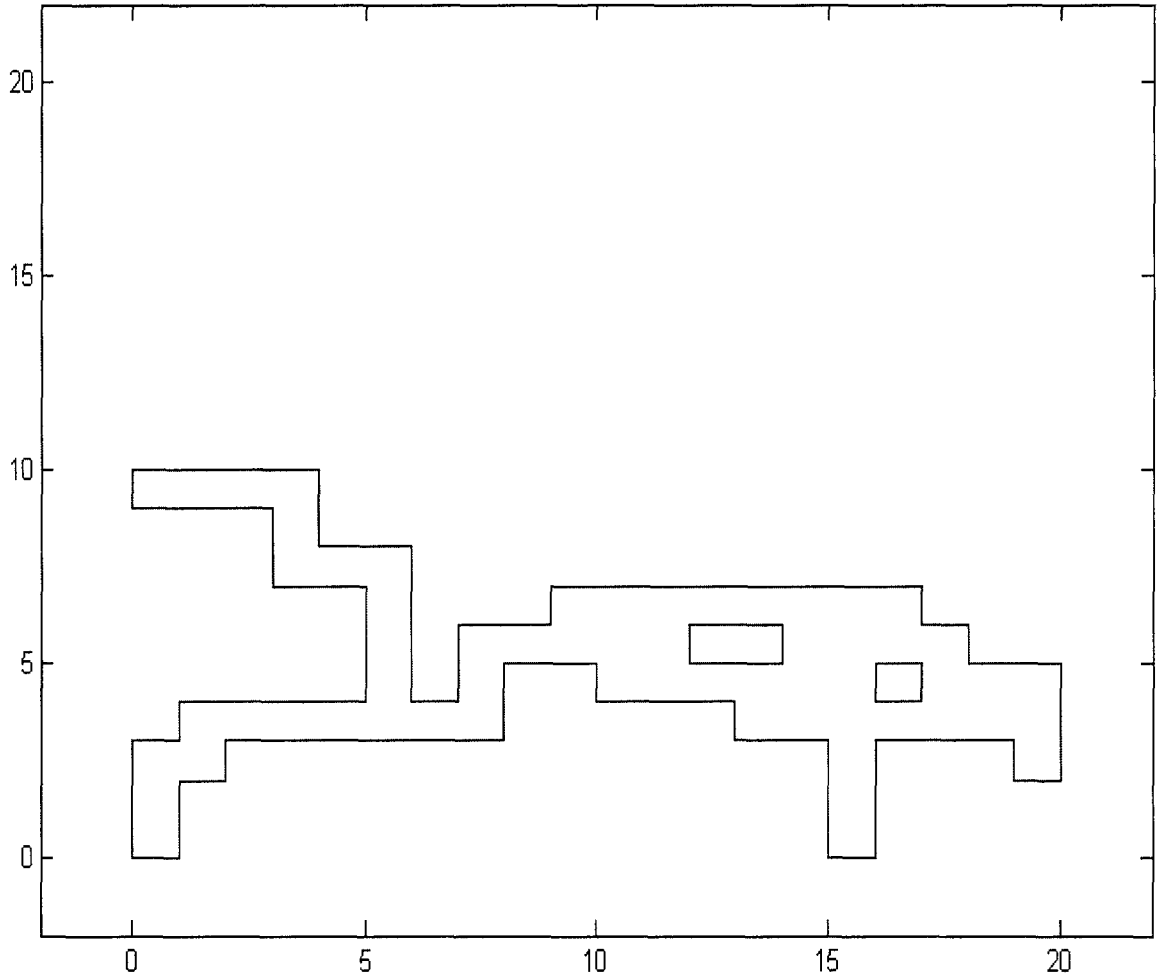


Figure 24 Initial Mesh

Step 2: Figure 25 shows 1st refinement the initial mesh.

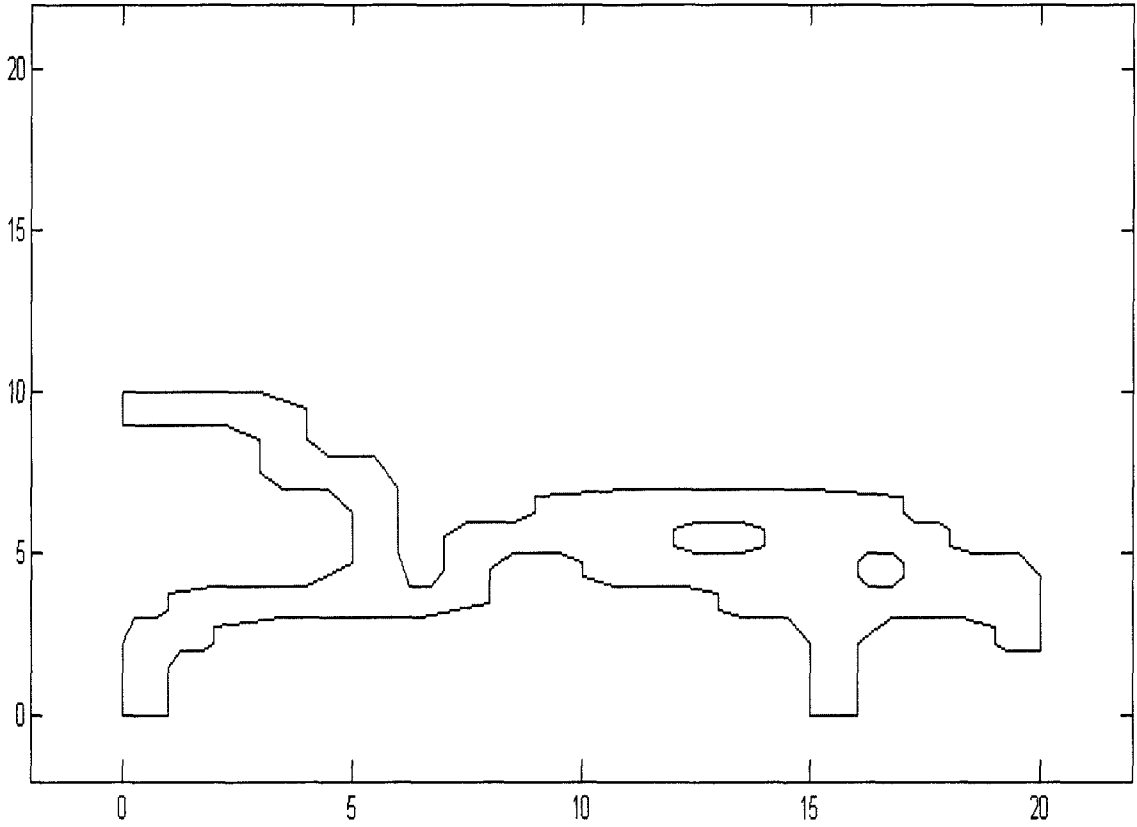


Figure 25 1st Refinement

Step 3: Figure 26 shows 2nd refinement the initial mesh.

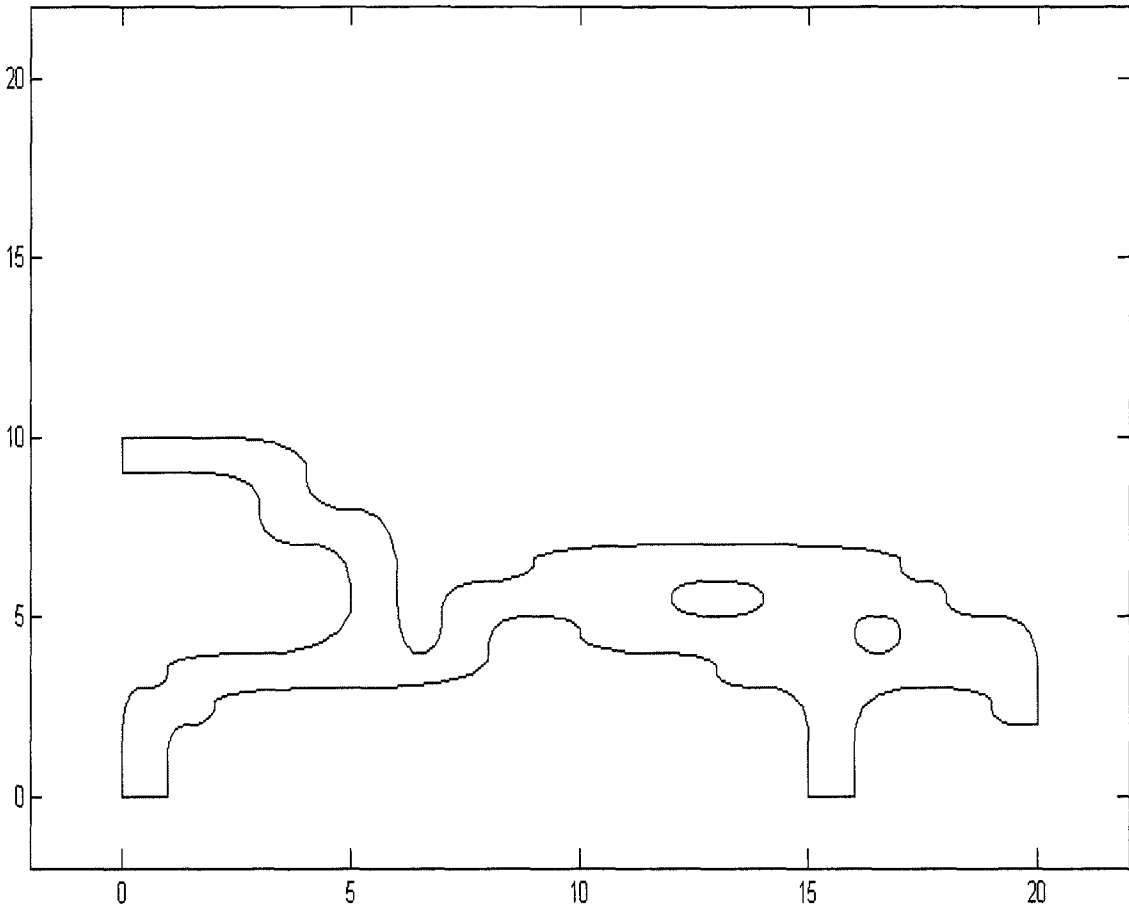


Figure 26 2nd Refinement

Step 4: Figure 27 shows 3rd refinement the initial mesh.

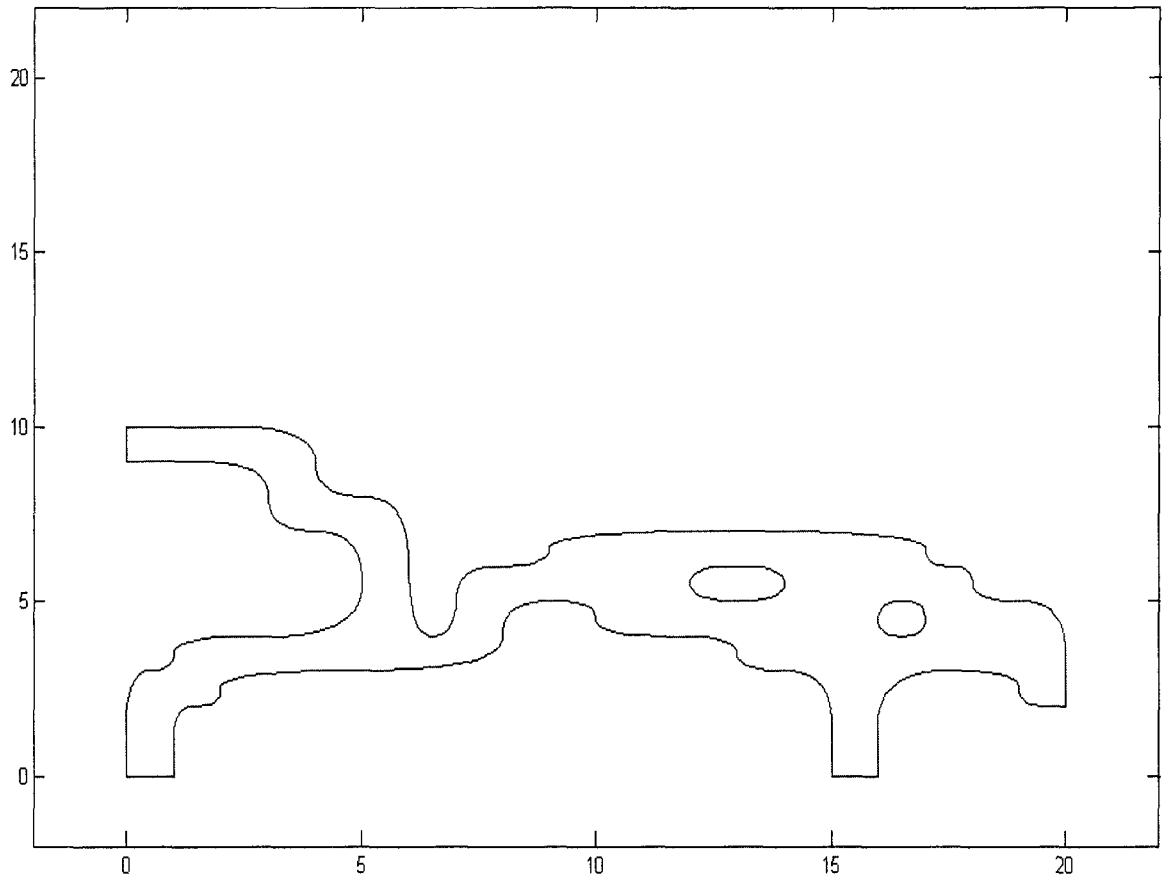


Figure 27 3rd Refinement

Step 5: Figure 28 shows 4th refinement of the initial mesh.

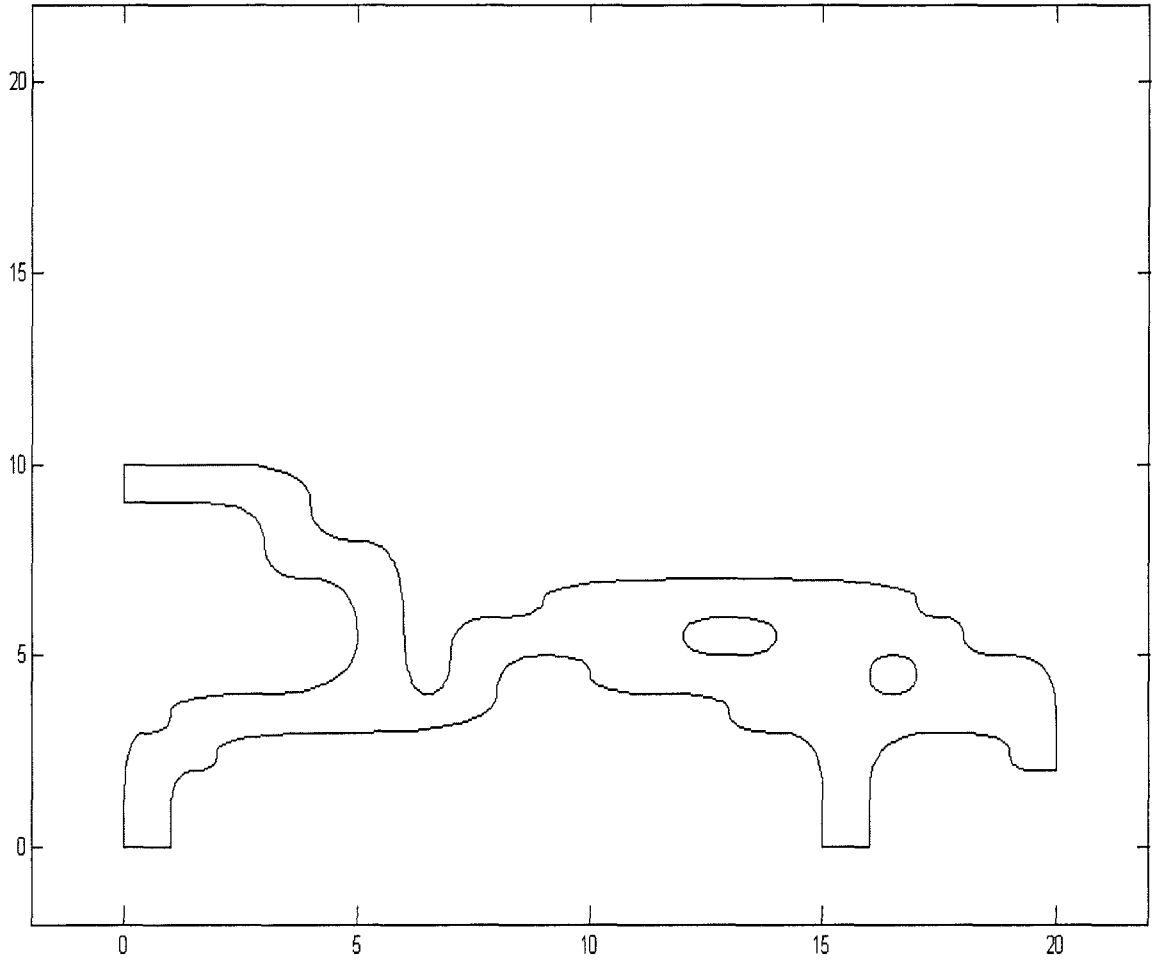


Figure 28 4th Refinement

Step 6: Figure 29 shows 5th refinement of the initial mesh.

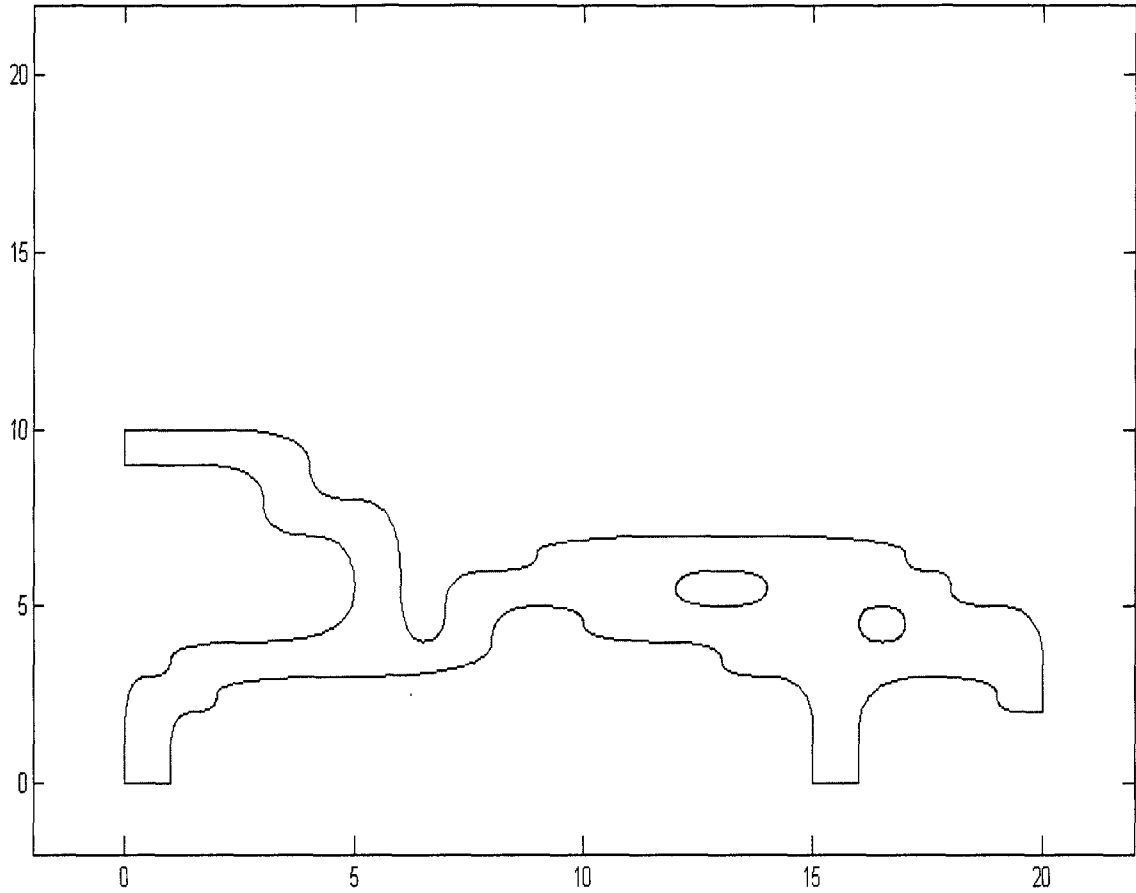


Figure 29 5th Refinement





Step 2: Figure 31 shows 1st refinement of the initial mesh.

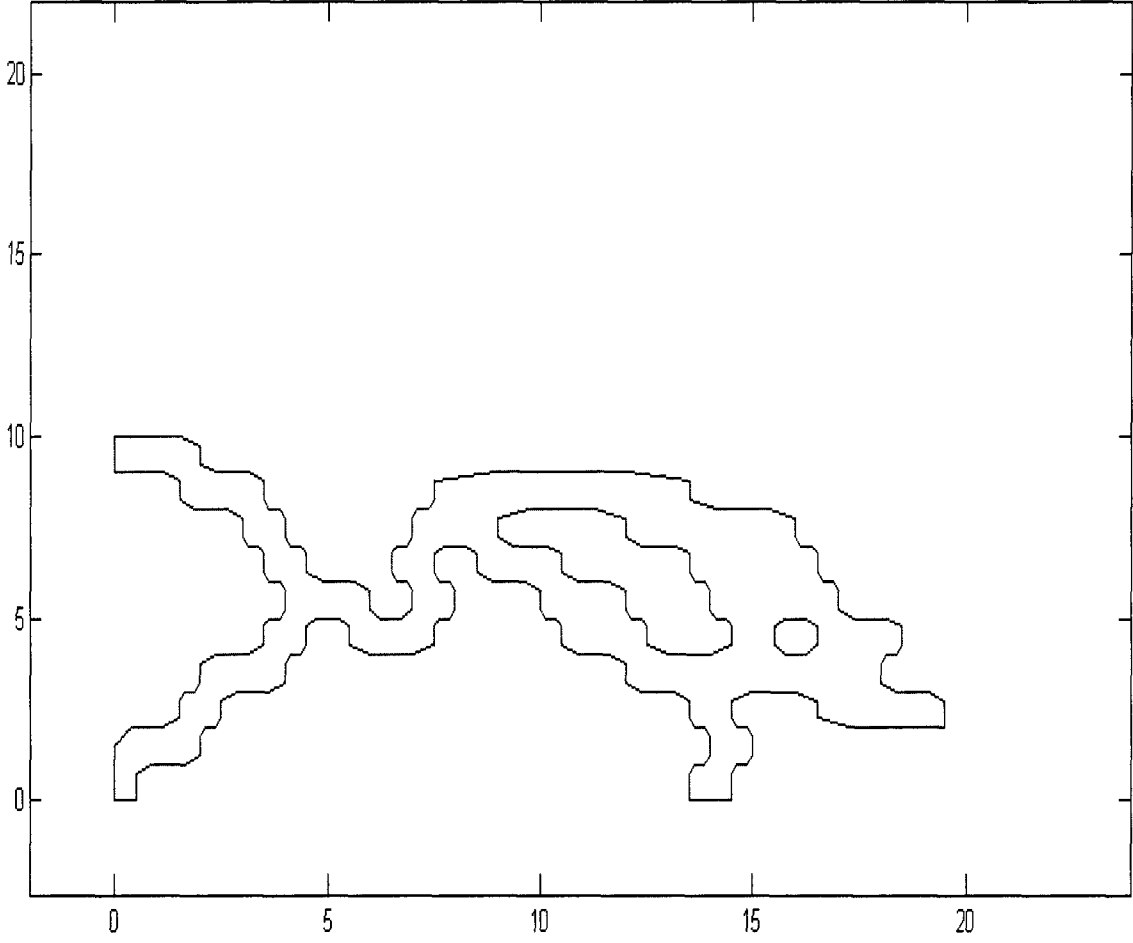


Figure 31 1st Refinement

Step 3: Figure 32 shows 2nd refinement of the initial mesh.

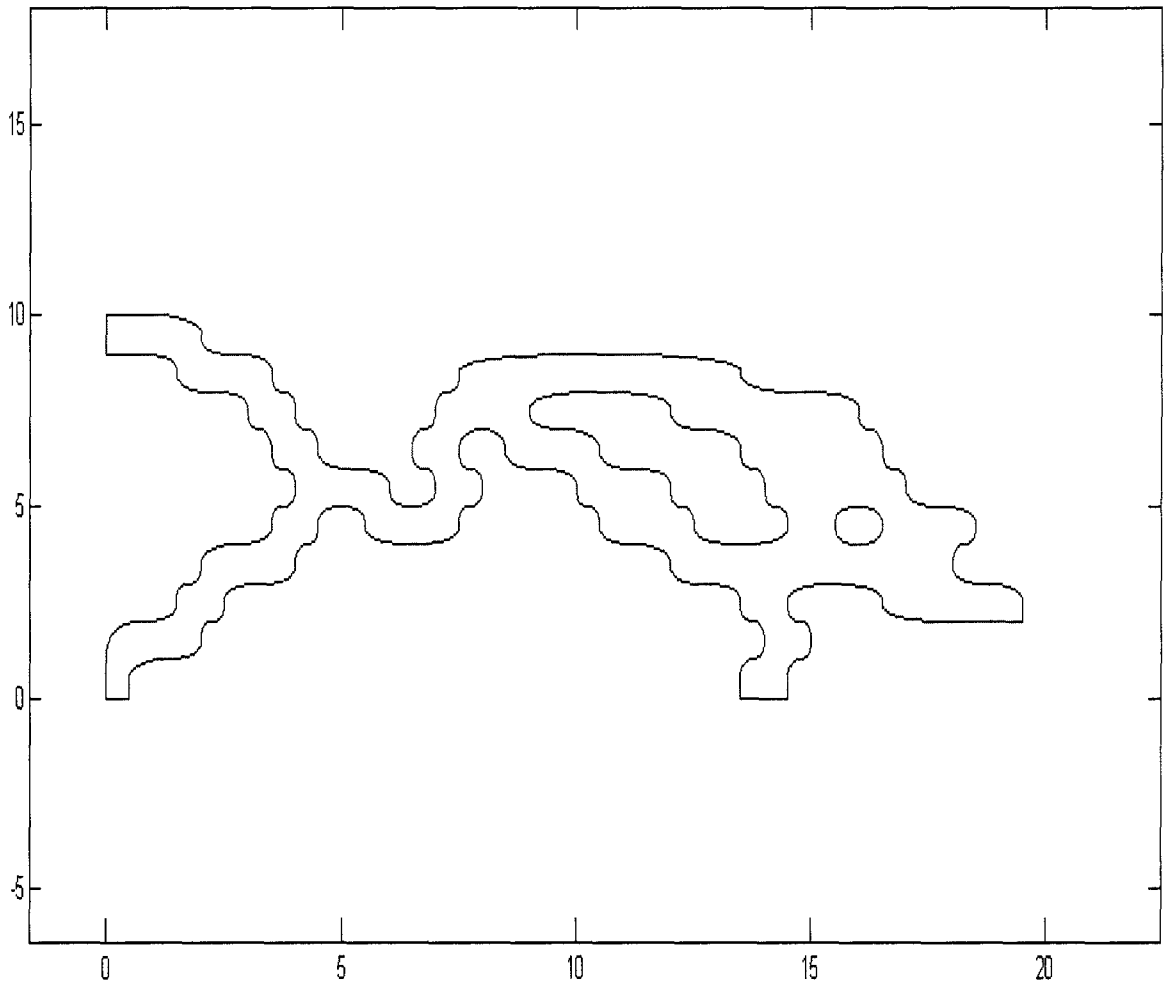


Figure 32 2nd Refinement

Step 4: Figure 33 shows 3rd refinement of the initial mesh.

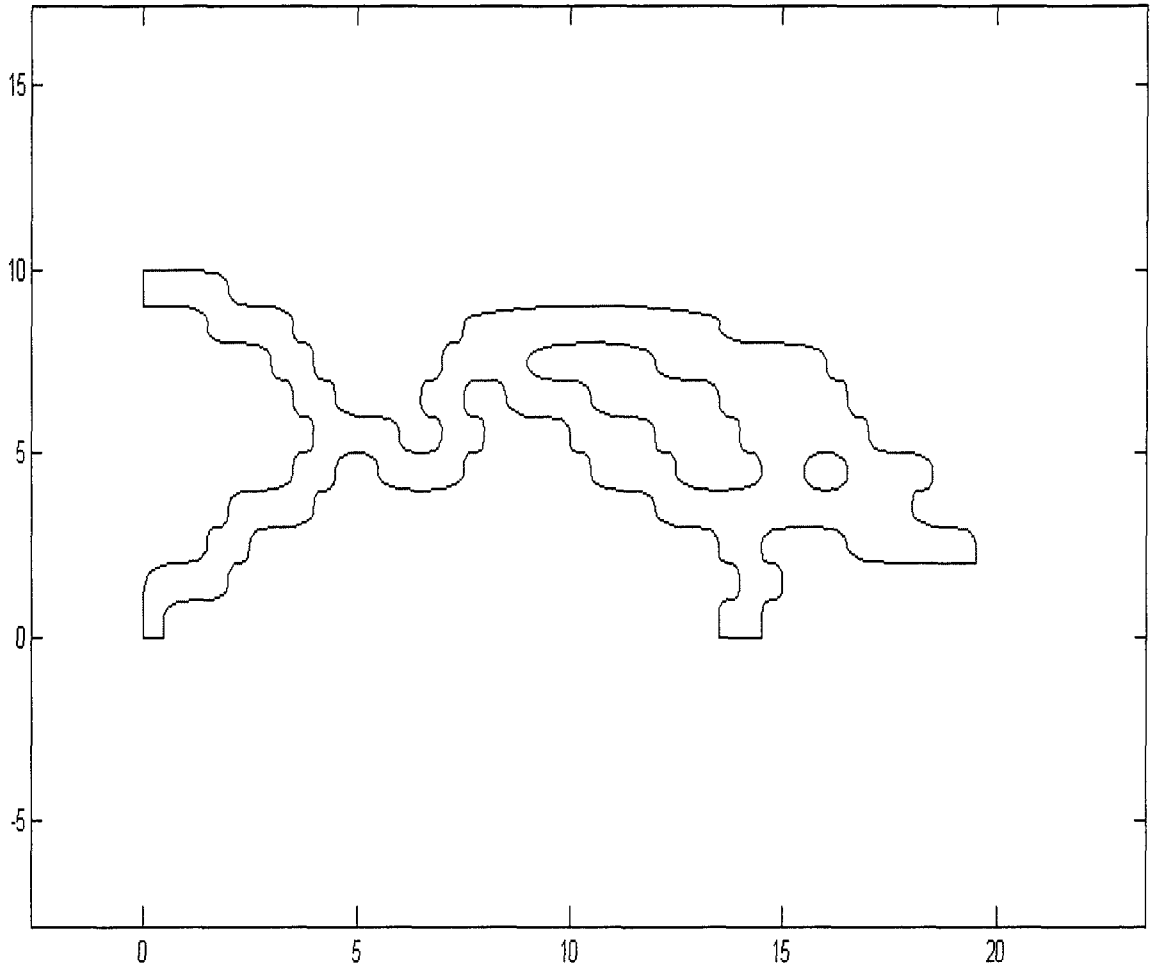


Figure 33 3rd Refinement

Step 5: Figure 34 shows 4th refinement of the initial mesh.

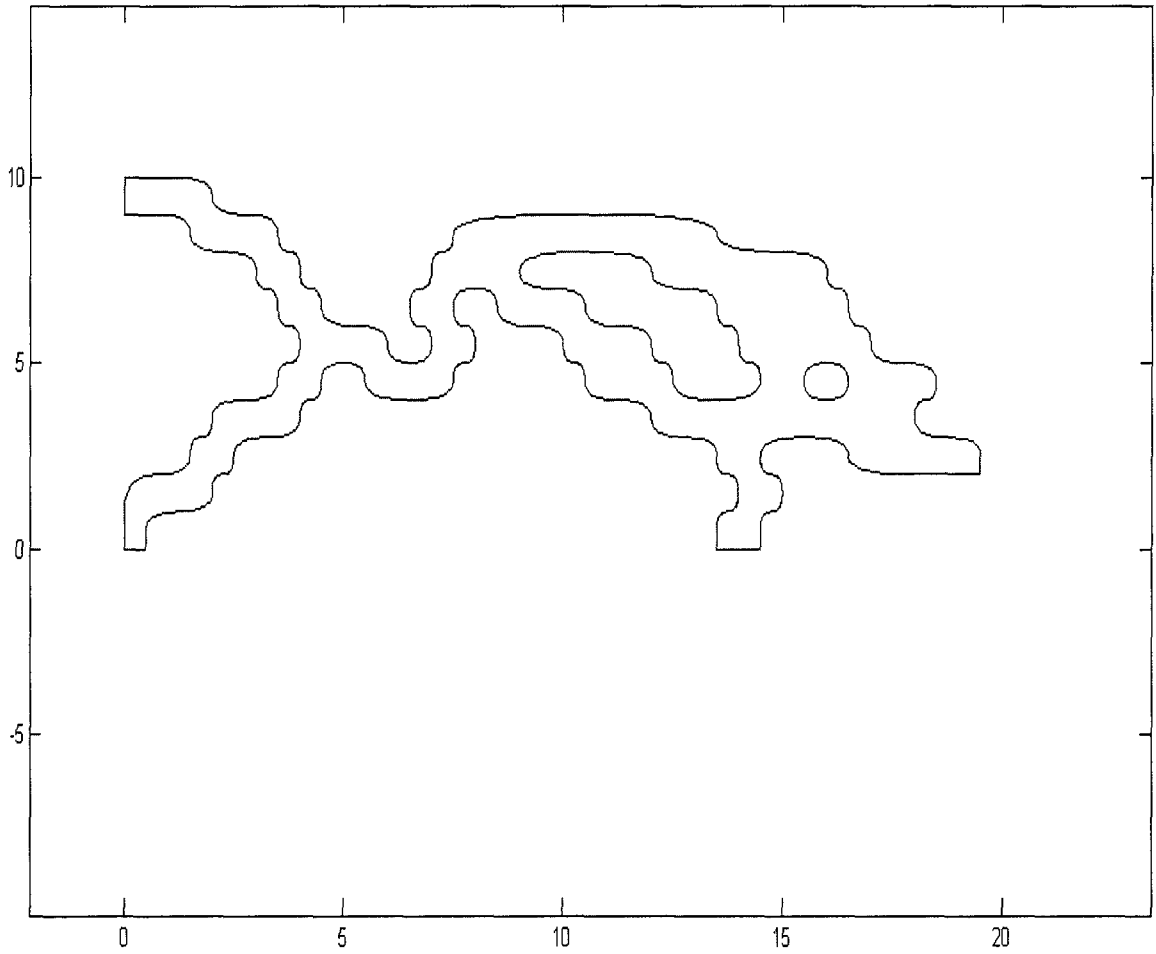


Figure 34 4th Refinement

Step 6: Figure 35 shows 5th refinement of the initial mesh.

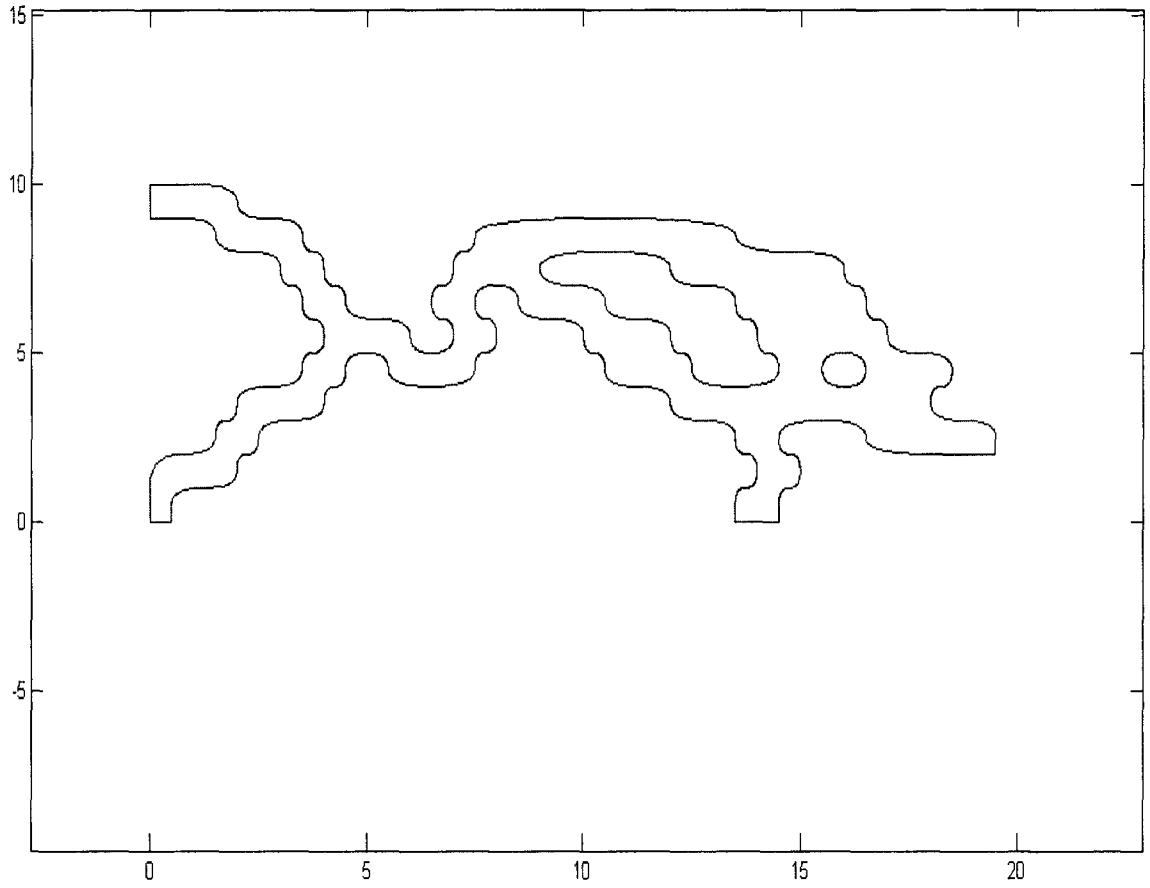


Figure 35 5th Refinement

## CHAPTER V

### DISCUSSION AND CONCLUSION

In this thesis, shape optimization of compliant mechanisms was done through subdivision. The compliant displacement inverter and amplifier, and compliant gripper were used as two examples of compliant mechanisms. Regular quadrilateral discretizations and modified quadrilateral discretizations were considered as the two topological discretizations for both the compliant mechanisms for the purpose of subdivision.

To subdivide the initial mesh of the discretized compliant mechanisms, Chaikin's algorithm for corner cutting was used. This method holds benefit over the other popular subdivision scheme, i.e., Doo-Sabin subdivision scheme because Chaikin's algorithm can be easily applied to both closed and open loops. Since for our analysis purposes, we considered only one half of the mechanisms, the figure contained both open loops and closed loops. Since the input and output nodes were also fixed, Chaikin's algorithm was easily applicable to other control nodes in the control mesh. With each refinement after applying Chaikin's algorithm, smooth edges were generated which resulted in the overall smoothness of the shape.

The thesis involved analysis of the two dimensional surface for the compliant mechanisms. The analysis can be extended to three dimensional problems as well. Since Chaikin's algorithm converges the initial mesh to the desired limit surface slower than other subdivision methods. Other popular methods of Doo-Sabin subdivision scheme, Catmull-Clark's algorithm for subdivision etc. could be used for faster convergence. Work can also be extended where; by selecting different control points on the initial

surface, different control polygons can be formed. Subdivision of control polygons formed by a combination of different initial control points can also be compared.

## REFERENCES

1. W. Zhixue (2005). "An efficient approach for shape optimization of components". *International Journal of Mechanical Sciences*, 1595–1610.
2. V. Braibant and C. Fleury (1984). "Shape Optimal Design Using B-Splines". *Computer Methods In Applied Mechanics And Engineering*, 247-267.
3. D. Zorin and D. Kristjansson (2002). Evaluation of piecewise smooth subdivision surfaces. *The Visual Computer*, Springer Berlin / Heidelberg, 299-315.
4. J. Stam (1998). "Exact evaluation of Catmull–Clark subdivision surfaces at arbitrary parameter values". *Proceedings of ACM SIGGRAPH 98*, Orlando, Florida, 395–404.
5. H. Suzuki, S. Takeuchi and T. Kanai (1999). "Subdivision Surface Fitting to a Range of Points". *Proc. IEEE Pacific Graphics '99*, 158-167
6. N. Litke, A. Levin and P. Schroder (2001). "Fitting Subdivision Surfaces". *Visualization Conference*, IEEE.
7. K.C. Hui, Y.H. Lai(2007). "Generating subdivision surfaces from profile curves", *Computer-Aided Design*, 783–793.
8. J.-a. Lian and Y. Yang (2011). "A new cross subdivision scheme for surface design", *Journal of Mathematical Analysis and Applications*, 244-257.
9. H. Biermann (2001). Sharp Features on Multiresolution Subdivision Surfaces. *Computer Graphics and Applications*, Pacific Conference on Computer Graphics and Applications.
10. J. Stam, C. Loop (2001). "Computer Graphics Forum", Volume 22, Number 1, 79-85.
11. I. Daubechies, I. Guskov, W. Sweldens (1985). "Regularity of Irregular Subdivision", *Springer*, vol. 15, Number 3, pp. 381-426.



12. H. Prautzsch (1985). "Generalized subdivision and convergence", *Computer Aided Geometric Design*, 69-75.
13. N. Litke, A. Levin, et al. (2001). "Trimming for subdivision surfaces." *Computer Aided Geometric Design*, 463-481.
14. Wang (2009). "Integration of CAD and boundary element analysis through subdivision methods." *Computers & Industrial Engineering*, 691-698.
15. Kwan Pyo Ko (2007). "A Study on Subdivision scheme-Draft." Dongseo University Busan, South Korea. See also <http://kowon.dongseo.ac.kr/~kpko/publication/2004book.pdf>.
16. D. Salomon (2005). "Curves and surfaces for computer graphics." Springer Verlag, New York.
17. David Jonsson (2005). "Visualization of CAD models - NURBS vs. Subdivision." University of Umeå. See also <http://www.cs.umu.se/~c99djn/thesis/report.pdf>.
18. P. V. Hull, S. Canfield (2006). "Optimal Synthesis of Compliant Mechanisms Using Subdivision and Commercial FEA". *J. Mech. Design*. volume 128, Issue 2, 337.
19. Peter Schröder, Denis Zorin et al. (2000) "Subdivision for Modeling and Animation" SIGGRAPH 2000 Course Notes.
20. W. Ma. (2005). "Subdivision surfaces for CAD--an overview." *Computer-Aided Design* 37(7): 693-709.
21. C.Loop (1987). "Smooth Subdivision Surfaces Based on Triangles." M.S. Thesis, University of Utah, Mathematics. See also <http://www.cs.berkeley.edu/~sequin/CS284/TEXT/loop87.pdf>

22. C. Loop (2002). "Smooth Ternary Subdivision of Triangle Meshes". Curve and Surface fitting:St. Malo 2002.
23. G. Chaikin (1974). "An Algorithm for High Speed Curve Generation". Computer Graphics and Image Processing, 3:346-349.
24. D Doo, M. Sabin (1978). "Behavior of Recursive Division Points Near Extraordinary Points". Computer Aided Design, 10(6) : 356-360.

## VITA

Akshat Singh received his Bachelor's of Engineering in Mechanical Engineering (2008) from Veer Narmad South Gujarat University, Gujarat, India. He is presently pursuing his Master's of Science in Mechanical engineering at Texas A&M University- Kingsville, Texas.

Permanent Address:

c-794, Kribhco Township

Hazira, Surat-394515

Gujarat, India.