B.Tech Project Report

Unstructured Mesh Generation on Planes and Surfaces using Graded Triangulation

submitted in partial fulfillment of the

requirements for the degree of

Bachelor of Technology

in

Aerospace Engineering

by

Anirudh Modi

93001015

under the guidance of

Prof. G.R. Shevare

Department of Aerospace Engineering

Indian Institute of Technology

Bombay

April 1997

Abstract

Triangulation is one of the most important and widely used methods of unstructured mesh generation today. This report describes *Graded Triangulation*, which is essentially an extension of the advancing front method (AFM) to obtain a smooth gradation in the mesh. The report begins with an introduction to mesh generation and the general concepts involved. It also describes the various mesh generation techniques. Since this triangulation requires both AFM and initial Delaunay triangulation, both are also discussed in brief. The implementation of graded triangulation is explained and its extension to surface triangulation discussed. Finally, a comparison between graded triangulation (in 2D) and AFM is brought about with the help of several examples.

Acknowledgment

8th April, 1997

I am thankful to **Prof. G.R. Shevare**, who amidst his busy schedule, spared time to guide me through the project. Without his support, motivation and useful suggestions, this project would not have taken its present shape. I also thank him for the great computing facilities extended and the freedom allowed to me throughout. I also owe my gratitude to Ravi for his continuous help and support during the first stage of the project, and to Avinash for giving me company and keeping me motivated throughout.

Anirudh Modi

Contents

List of Figures

Chapter 1

Introduction

Many problems in CFD, Computational Physics etc., involve the numerical solution of a set of equations in a complicated shape domain. The solution of such problems requires the domain to be discretized to produce a set of points on which the numerical algorithm can be based. For some problems, the generation of a suitable grid/mesh can be as demanding as the effort required to perform the computations for which the grid was intended. In recent times, considerable attention has been focused on the discretization process, which is commonly called mesh generation.

Numerical mesh generation has now become a fairly common tool for use in the numerical solution of PDEs on arbitrarily shaped regions. This is especially true in CFD, from which came much of the driving force for the development of this technique, but the procedures are equally applicable to all physical problems that involve field solutions. The basic techniques involved in numerical mesh generation are -

1. a means of distributing points over the field in an orderly fashion, so that neighbors may be easily identified and data can be stored and handled efficiently.

2. a means of communication between points, so that a smooth distribution is maintained as points shift their positions.

3. a means of representing continuous functions by discrete values on a collection of points with sufficient accuracy, and a means for evaluation of the error in this representation.

4. a means for communicating the need for a redistribution of points in the light of the error evaluation, and a means of controlling this redistribution.

This technique frees the computational simulation from restriction to certain boundary shapes, and allows general codes to be written in which the boundary shapes is specified simply by input.

Thus, mesh generation or grid generation is the process of decomposition of the domain. With the advent of Finite Element Methods (FEM) and Finite Volume Methods (FVM) which can be applied to grid cells of any arbitrary shape and connectivity, unstructured meshes are now being widely used. Triangulation is the most widely used form of unstructured mesh generation as any given arbitrary complex geometry can be more flexibly filled by triangular elements than by quadrilateral elements. Triangulation plays an important role in FEMs, numerical analysis, computer-aided geometric design (CAGD), approximation theory, and elsewhere. The applicability and accuracy of the finite element analysis is dependent upto a large extent on the validity and quality of the meshes generated. Thus it is important to have an acceptable and efficient process of mesh generation for all types of domains.

In some cases, the type and quality of triangulation is much more important than the time taken to generate the triangulation. For example, in case of boundary layer calculations, like in a flow past an airfoil, more points are required around the boundary than away from it. Here, conventional triangulation methods like Delaunay and Advancing Front often fail posing a need for another method which can give gradual variation or gradation in the triangulation generated. This is where Graded Triangulation comes into focus, which uses an appropriate combination of Delaunay and Advancing Front methods to generate the required mesh for this purpose.

The aim of this project is to develop a fully automatic unstructured triangular mesh generator. Work done in this project also forms an integral part of IITZeus, a surface modeling and mesh generation package being developed by the Department of Aerospace Engineering at IIT, Bombay. The package can generate curves like B-spline curves (BSC) and piecewise linear curves (PLC), and surfaces like B-spline surfaces (BSS), piecewise bi-linear surfaces (PBLS), surfaces of revolution (RS) and Coon's patches. These form the basis for generating the various inputs for the triangulation routines.

Chapter 2

Mesh Generation

2.1 General concepts

2.1.1 Definition

A mesh of a domain Ω, is defined by a set τ_h of finite number of segments in 1D; segments, triangles and quadrilaterals in 2D; and segments, triangles, quadrilaterals, tetrahedra, pentahedra and hexahedra in 3D. [Geo91]

2.1.2 Conformity

The set τ_h is a conformal mesh of domain Ω (Figure 2.1), iff

1. The domain Ω is completely and exactly covered by the mesh τ_h. When the domain Ω is not polygonal (in 2D) or polyhedral (in 3D) (i.e., if it is defined by a smooth curve or a surface), the mesh τ_h will only be an approximate partitioning of the domain.

2. All elements of mesh τ_h must have a non-empty interior.

3. The intersection of any 2 elements in the mesh τ_h is either an empty set, a point, an edge or a face (of both elements).

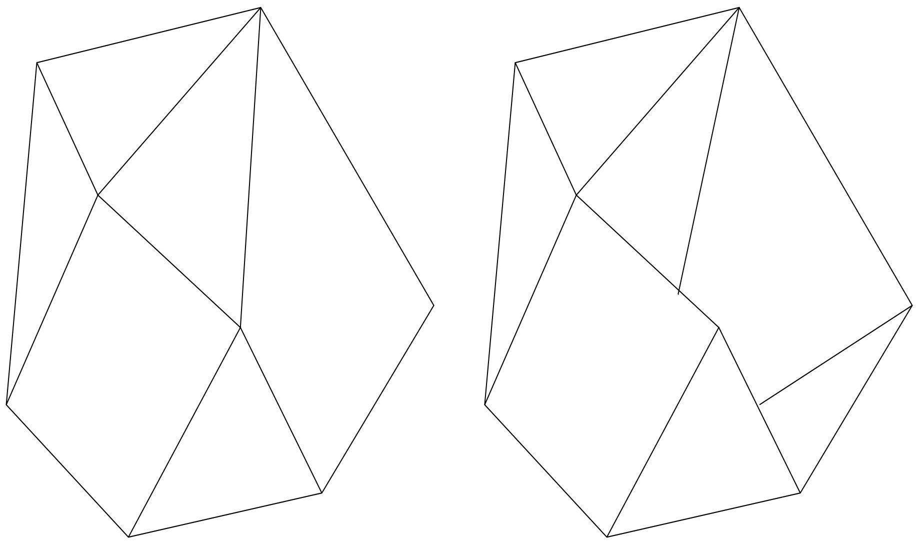

Figure 2.1: Conformal and non-conformal meshes

2.1.3 Quality of the mesh

There are numerous criteria for deciding the quality of a mesh [Geo91]. A few of these criteria as applicable to planar/surface meshes are -

1. The variation in the area of the elements should not be too large.
2. The aspect ratio of triangular elements should be as close to 1 as possible.
   The aspect ratio of a triangular element is defined as the ratio of the circumradius of the triangle to twice its inradius. Hence the aspect ratio of an equilateral triangle is exactly 1.
3. The ratio of the largest to the smallest edge/angle of the element should be close to 1.
4. The ratio of the area of the largest/smallest element to all the immediate neighbours should not be drastically low/high.

2.1.4 Connectivity of the mesh

The connectivity of a mesh is the definition of the connection of its vertices. A mesh is called structured if the connectivity is same throughout the mesh and each element has a fixed num-

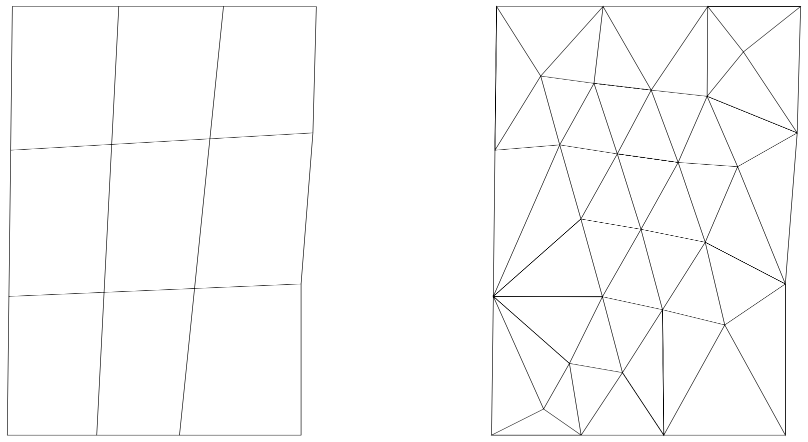
Figure 2.2: Structured and unstructured meshes

ber of neighbours. An unstructured mesh does not have a fixed connectivity among its elements (Figure 2.2).

2.2 Mesh generation methods

The various methods available for mesh generation can be enumerated as [Sha96] -

1. *Manual generation* - In this approach the user defines each element by the vertices. This approach is feasible only when a limited number of elements are required and the domain is very simple.

2. *Transport mapping method* - Also known as *Transfinite Interpolation* method, it involves blending of a mesh generated in a parametric domain into the real domain, as the parametric boundary blends into the real boundary.

3. *Explicit solution of Partial Differential Equations* - In this approach the mesh is first generated in a parametric domain in a simple geometry (quadrilateral, tetrahedron etc.). Then a mapping function is defined to ensure properties such as boundary conformity, orthogonality of elements, variable density of elements and the like. Such methods result in structured meshes.

4. *Advancing Front Methods (AFM)* - These methods are widely used to create planar, surface and volume grids which can have triangular, quadrilateral or higher order elements. The boundary is represented as continuous or discretised curve(s) (for 2D mesh generation) and triangulated or continuous surface(s) (for 3D mesh generation). A front, initialised by the boundary is established. The front is updated as new internal points are created till the front becomes empty, also symbolising the creation of the mesh in the entire domain.

5. *Delaunay-Voronoi Triangulation* - This algorithm is widely used for tetrahedralisation of 3D domains (also known as 3D triangulation). This results in a mesh which is optimal for a set of given points in the domain. However, the quality of the mesh depends upon the points specified and a bad choice of points may result in poor triangulation. This method is much faster than the AFM.

6. *Sweepline method* - This algorithm constructs the Voronoi tessellation by searching for Voronoi region boundaries in an orderly fashion. The nodes are sorted in a specific coordinate direction. A sweepline is defined as the description of the intersection of a moving line perpendicular to the specified direction with the Voronoi boundaries. The sweepline starts from the least value of sorted coordinates and moves to the other extreme. As each node is encountered by the sweepline, new bisectors of the lines joining two nodes are introduced in it. The aim is to detect the point of intersection of three bisectors to get the circumcenter of the triangle formed by three nodes. This is a relatively new and involved algorithm but a time complexity better than Delaunay's algorithm is claimed.

Chapter 3

Advancing Front Triangulation

3.1 Introduction

This method is particularly suited for the boundary representation of domains. If the boundary is continuous, it is discretised to give a piecewise linear curve (PLC) based roughly upon the number of triangles required in the domain. The initial front is a set of line segments defining the boundary completely. Usually, the internal and the external curves are specified in opposite sense in order to get a convergent algorithm. This is useful if new points are placed only on one side of the boundary as the curves are traversed in the specified sense. A method of selection of trial point which leads to direct graded triangulation is discussed here.

3.2 Algorithm

The general algorithm [Geo91] is

1. Define the boundary of the domain to be discretised.

2. Initialise the front as a piecewise linear curve in conformity with the boundary.

3. The edge to be deleted from the front is chosen based upon some criterion (generally smallest edge is chosen as it gives good quality meshes).

4. For the edge to be deleted -

    (a) Select the trial point position (trial point is the point lying inside the domain and making an equilateral triangle with the edge to be deleted).

    (b) Search for any already existing point within a certain proximity of the trial point. If any such point exists it becomes the trial point. Continue the search.

    (c) Determine whether the element formed with the new ideal point crosses any edges. If yes, select a new trial point from the front and try again (go to 4b).

5. Add the new point, edges and triangles to the respective lists.

6. Delete the base edge from the front and add the new edges.

7. If the front is non-empty, go to 3.

3.3 New point insertion

Depending on α, the angle between two consecutive edges of the front, three possibilities arise regarding the new point insertion and edge deletion [Geo91].

- $\alpha < \pi/2$. The two segments are retained and become the edges of the new triangles created. (Figure 3.1)

- $\pi/2 \leq \alpha \leq 2\pi/3$. An internal point and two triangles are generated from the two segments. (Figure 3.2)

- $2\pi/3 < \alpha/2$. One segment is retained while a triangle with this edge and an internal point is created. (Figure 3.3)

A more detailed information and implementation details can be found in [Sha96] or [Geo91].

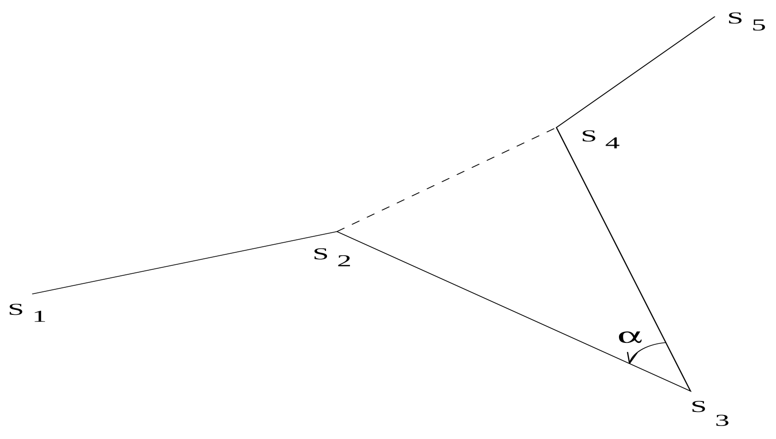

Figure 3.1: New point insertion : $\alpha < \pi/2$

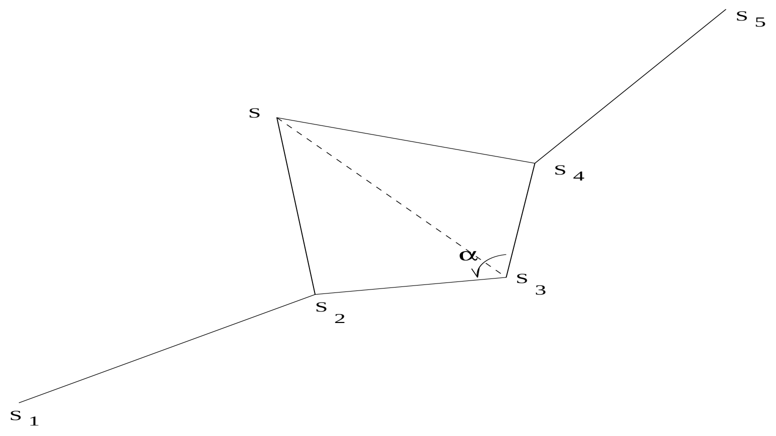

Figure 3.2: New point insertion : $\pi/2 \leq \alpha \leq 2\pi/3$

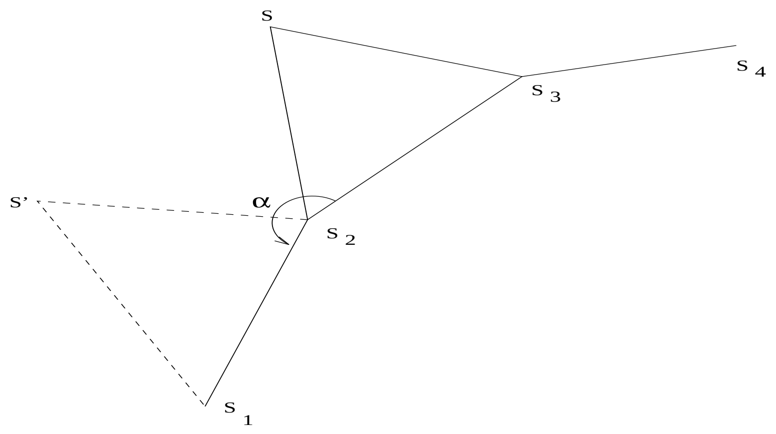

Figure 3.3: New point insertion : $\alpha > 2\pi/3$

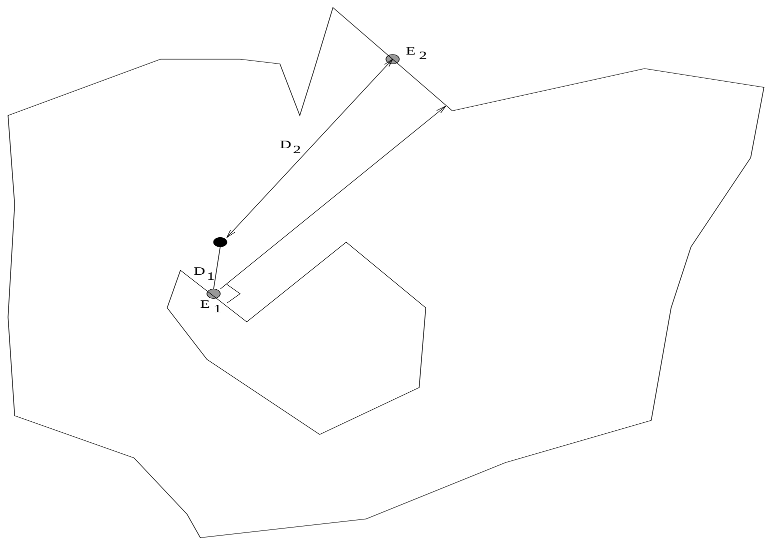

Figure 3.4: Direct graded triangulation

3.4 Determination of the trial point

The edge of the initial front closest to the mid-point P of the base edge is found. A normal is drawn through the mid-point of that edge and the closest edge of the initial front that it intersects is determined (Figure 3.4). If the distance of point P from mid-point of both the edges is D_1 and D_2 respectively and if L_1 and L_2 are the respective edge lengths then the required length of the new edges is calculated as

$$L = \frac{D_1 L_2 + D_2 L_1}{D_1 + D_2}$$

Triangulation using this method for selection of trial point is termed as direct graded triangulation which is implemented here in IITZeus (Figure 3.5). This algorithm requires many calculations and hence slows down the triangulation slightly. [Sha96]

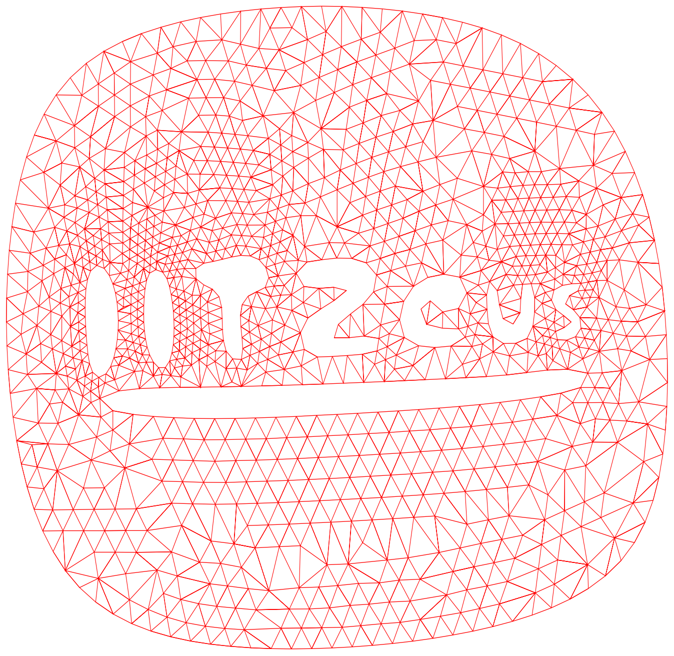

Figure 3.5: Advancing front triangulation - an example

Chapter 4

Delaunay Triangulation

4.1 Introduction

Delaunay triangulation is the most widely used triangulation method in unstructured mesh generation. It is one of the fastest triangulation methods with relatively easier implementation, giving excellent results for most applications.

4.2 Voronoi tessellation

Before attempting to learn about Delaunay triangulation it is helpful to learn about Voronoi tessellation also known as Theissen or Dirichlet [Bow81] tessellation. Voronoi tessellation is a geometric dual of Delaunay triangulation and one can be derived from the other. Given a set of N points in a plane, Voronoi tessellation divides the domain in a set of polygonal regions, the boundaries of which are the perpendicular bisectors of the lines joining the points (Figure 4.1). Furthermore each tile contains only one of the N points. If both these conditions are satisfied then the lines joining the points form a mesh known as the Delaunay triangulation. According to the Delaunay criterion, the circumcircle of every triangle is so formed that it does not contain any point of the mesh. Except in degenerate cases, the vertices of Voronoi tessellation occur where three tiles meet. Each Voronoi vertex is circumcenter of a Delaunay triangle. In degenerate cases more than one triangle

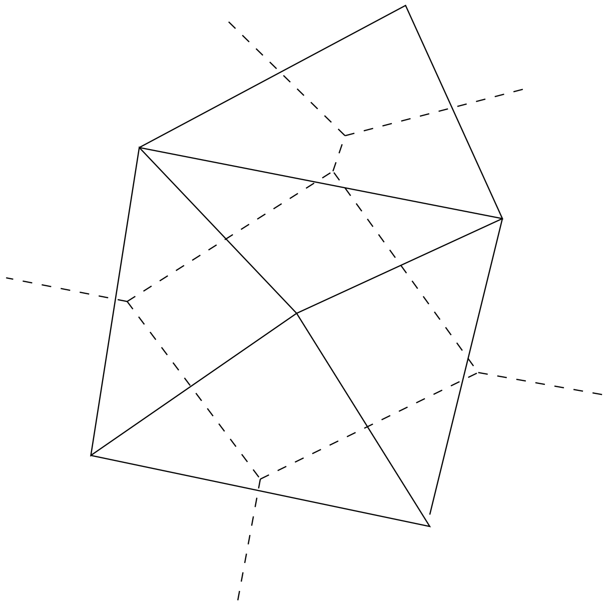

Figure 4.1: Voronoi tessellation and Delaunay triangulation

may be validly possible.

Since Delaunay triangulation is the geometric dual of Voronoi tessellation, many methods have been formulated to arrive at the former using the latter, though many methods for direct triangulation have also been formulated [HS88, LS80]. In Delaunay triangulation the boundary triangulation is not difficult but placing the interior points at inappropriate places may result in bad meshes even though the Delaunay criterion is satisfied.

4.3 Algorithms

A few algorithms for Delaunay triangulation are enumerated as [Sha96] -

1. *Watson's algorithm* - Watson's algorithm [Nan95] starts by forming a super-triangle which is a triangle encompassing all the given points of the domain. Initially the super-triangle is flagged as incomplete (Figures 4.2, 4.3, 4.4 and 4.5). Then the algorithm proceeds by incrementally inserting new points in the existing triangulation. A search is made for all the triangles whose circumcircles contain the new point and they are deleted to give what is

known as an insertion polygon. This gives the new triangulation. This process is continued till all points to be inserted are exhausted and then all triangles having the vertices of the super-triangle are deleted.

This is one of the simplest and the most extensively used algorithm for Delaunay triangulation.

2. *Lawson's algorithm or the diagonal swapping algorithm* - If a point is added to an existing triangular mesh then circumcircles are formed for all new triangles formed. If any of the neighbours lie inside the circumcircle of any triangle, then a quadrilateral is formed using the triangle and its neighbour. The diagonals of this quadrilateral are swapped to give a new triangulation. This process is continued till there are no more faulty triangles and no more swaps are required.

3. *Fang and Piegl algorithm* - This algorithm for triangulation places the given points in a uniform mesh and creates triangles in a circular manner. The authors claim a linear time complexity for the algorithm. The algorithm also produces the convex hull of the given set of points at no extra cost. The algorithm is so designed that points only on one side the current point under consideration have to be checked for the Delaunay criterion. However, this algorithm runs into troubles when too many points lie on a straight line.

4. *Constrained Delaunay Triangulation (CDT)* - If a set of points in a plane and a set of non-crossing edges are specified then the CDT creates a mesh such that -

   (a) All the specified edges are included in the triangulation.

   (b) It is as close to Delaunay triangulation as possible.

   This algorithm has been tested to work in a time complexity of $O(N \log N)$ for N specified points in the domain. If a set of points and non-crossing edges are specified then the CDT of the given set has the property that for all new edges there exists a circle such that,

   - the endpoints of the edge lie on the circle, and

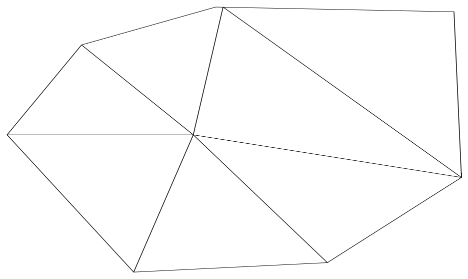
Figure 4.2: Watson's algorithm : Initial triangulation

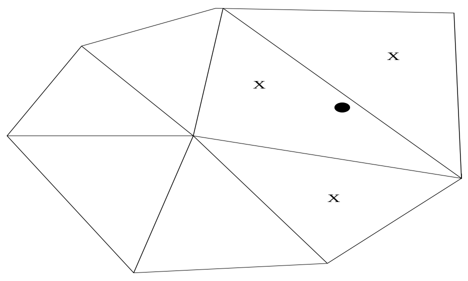

Figure 4.3: Watson's algorithm : Point insertion

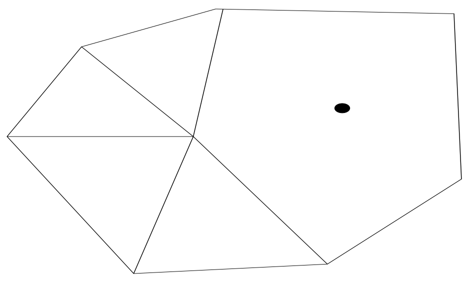
Figure 4.4: Watson's algorithm : Element deletion

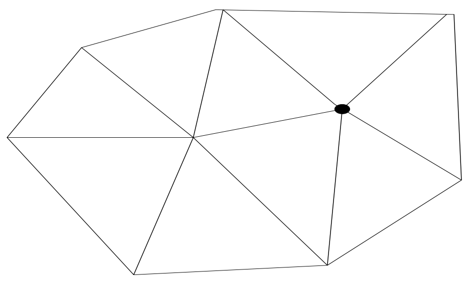
Figure 4.5: Watson's algorithm : Final mesh

- if any new node is in the circle, then it is invisible from at least one of the nodes of the edge, i.e., if lines are drawn from the point to the two nodes of the edge then at least one of them will intersect with one of the edges of the mesh.

Thus, if no edges are specified, the CDT is the same as Delaunay triangulation. CDT is a divide-and-conquer algorithm since the given data is sorted according to any one dimension and the domain is divided into strips. The CDT for each strip is then calculated and pasted to form new strips whose CDT is then calculated. This process is repeated till the CDT for the entire domain is obtained.

5. *Sweepline algorithm* - In a Voronoi tessellation, a sweepline is defined as the intersection of a moving horizontal line with the Voronoi boundaries. Circumcircles are located by finding the point of intersection of three bisectors of the lines joining the nodes. But the Voronoi polygons obtained this way are not very robust and there have been only a few attempts to tackle this problem. A one-to-one geometric transformation maps a bisector onto a hyperbola or a vertical half-line. Any bisector between any two nodes, say p and q, is mapped as

$$\begin{aligned} x^* &= x, \\ y^* &= y + \sqrt{(x_p - x)^2 + (y_p - y)^2} \end{aligned}$$

The transformation maps x as such and y as the original value and the distance between the point and the node p. The mapped Voronoi regions $R_p{}^*$ and $R_q{}^*$ are bounded by the mapped bisector, say C_{pq}, which is either a hyperbola or a straight half-line. A bisector is generated only if the sweepline has reached both forming nodes. An expected vertex is the intersection point of two neighbouring bisectors.

4.4 Initial Delaunay triangulation

When Delaunay triangulation is carried out on only the boundary points specified without any extra point being inserted in the domain, the triangulation generated is known as initial Delaunay or background triangulation (Figure 4.6). This triangulation is utilised in the graded triangulation described later. A final Delaunay triangulation of the same domain using the area criterion [Nan95] is also shown (Figure 4.7).

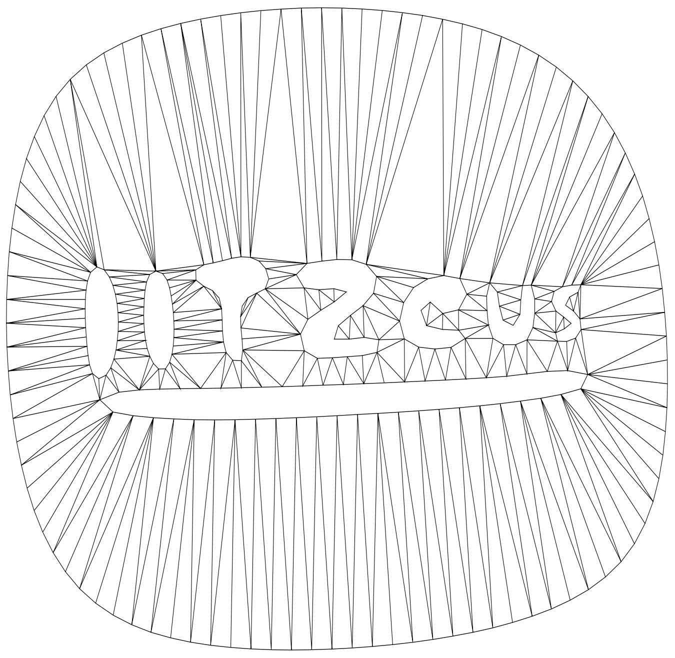

Figure 4.6: Initial Delaunay triangulation - an example

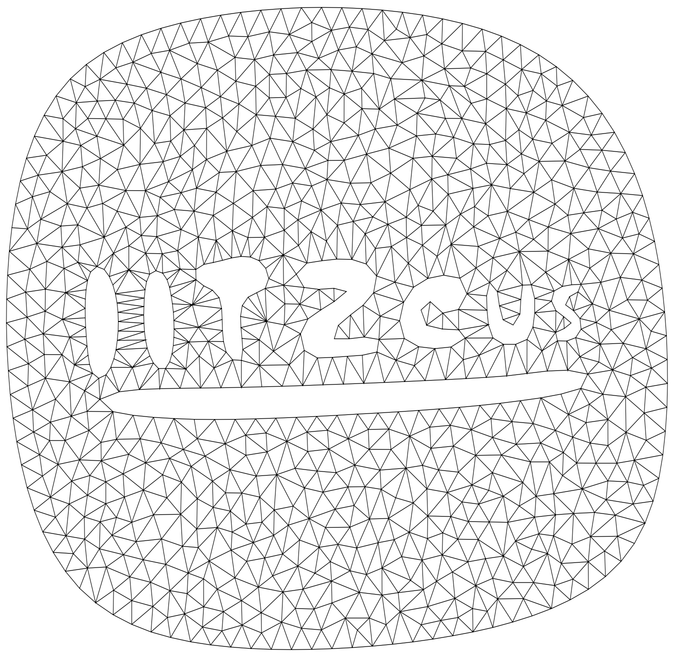

Figure 4.7: Final Delaunay triangulation using area criterion - an example

Chapter 5

Graded Triangulation

5.1 Introduction

Although very popular, the AFM has many shortcomings. One of them is that since no information about the background mesh is known (i.e., the rest of the front), there is a high chance of the clashing of fronts (Figures 7.4, 7.14) or occurrence of sliver triangles (Figure 7.22). Also, in physical problems like high speed flows over bodies (like flow over airfoil - Figure 7.1), the mesh is usually required to be adapted or graded in a particular direction. More number of triangles (i.e., dense triangulation) is required around the boundary and fewer number of triangles are required away from the boundary. Delaunay triangulation, although very fast, can give uniform triangulation, but cannot give this kind of gradation required. Hence, a new approach is required to achieve this. This approach uses both Delaunay triangulation and AFM in succession to achieve the desired results.

5.2 Algorithm

A popular method of graded triangulation is by use of a background mesh for the entire domain. The background mesh is a triangulated mesh generated using only the boundary points. Such a mesh can be generated by the use of initial Delaunay triangulation. Now the graded triangulation differs from the AFM only in that the function for calculating the trial point is modified to accommodate the grading. This function now uses the information of the background mesh such that the length of the triangle's edges increase from the minimum value to the maximum. At any point in the domain the edge length of the sides of the triangle under construction are calculated to ensure gradual variation by finding the background triangle that encloses the point. The barycentric coordinates of the point with respect to the three nodes of the triangle are calculated. The mesh density at any node is defined as the mean of the lengths of the edges meeting at that point. If the mesh density at the nodes of the triangle containing the point are L_1, L_2 and L_3 and the respective barycentric coordinates of the point are α, β and γ (note that $\alpha+\beta+\gamma=1$), then the required lengths of the new edges are given by

$$L = \alpha L_1 + \beta L_2 + \gamma L_3$$

The disadvantage of this method is that the Delaunay triangulation for initial triangulation should be done before starting the actual triangulation, but it results in a far better mesh than direct graded triangulation which does not use any background mesh.

5.3 Implementation

The above algorithm requires the implementation of both the Delaunay triangulation and the AFM. These have been done as described in the previous chapters. The algorithm can now be described as -

- The domain/boundary is read (only one level of holes are allowed for need of continuum) and converted to a PLC if not already so.
- Initial Delaunay triangulation is done using these boundary points and the triangles are stored in a TRI structure [Nan95].
- The node density of each boundary point is now computed using the above information and stored. This is simply the mean of all the edges in the triangulation meeting at that point.

- Now, the AFM proceeds and the function for determining the new insertion point is modified as follows -

    1. A first approximation to the insertion point (called trial point) is made in the same way as described earlier for the conventional AFM.
    2. Using this point, the background triangle containing the point is found using a simple search algorithm.
    3. Upon finding this background triangle, the barycentric coordinates (α, β and γ) of the trial point w.r.t. the triangle is computed. These are simply the ratios of the area of the three triangles formed by the trial point and the background triangle to the area of the background triangle.
    4. The already computed node densities of the nodes of the background triangle (L_1, L_2 and L_3) are now recalled.
    5. The new length of the trial point from the edge of the front is now recomputed based on a suitable function of the above 6 parameters α, β, γ, L_1, L_2 and L_3.
    6. Having got a new trial point, iteration starts again from 2, till the next point obtained by the procedure is the same as the previous point or within a certain tolerance band of the previous point.

- Having got the final insertion point, the AFM proceeds till the triangulation is completed.

Also, apart from the above mentioned 6 parameters, the gradation function is implemented to accept two additional parameters x and y which specify the location of the current trial point.

5.4 Efficiency

The efficiency of the algorithm implemented above not only depends upon the number of boundary points N but also on the area of the continuum to be discretised and the gradation function used. Since the AFM itself is iterative, an accurate statement about the efficiency in terms of the number of boundary points N cannot be made.

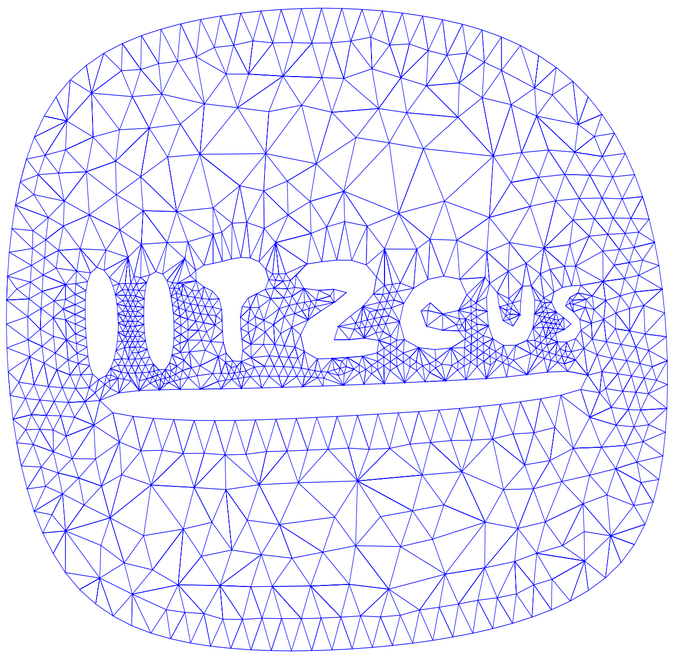

Figure 5.1: Graded triangulation - an example

Chapter 6

Surface Triangulation

Surface triangulation is an important milestone in 3D mesh generation as it forms the input for the tetrahedralisation of volumes which is a 3D analog of triangulation in 2D. Efficient methods like Delaunay triangulation cannot be applied to surfaces as the Delaunay criterion is not defined for surfaces like it is for planar domains (2D) or volumes (3D).
The following two methods can be employed for surface triangulation -

1. *Triangulation in Parametric Domain* - One-to-one mapping of the surface component onto a 2D parametric domain is done. Graded triangulation is then applied to the parametric plane taking into consideration the actual edge lengths and curvature in 3D. The generated grid is then transformed back to 3D surface. Since the basic triangulation takes place in the 2D (u,v) plane, this method is just an extension of the 2D algorithm described earlier, and is expected to provide quality grids for various applications.

2. *Direct Triangulation* - Triangulation done in 2D can be extended to a surface by performing direct triangulation (i.e., attempting to triangulate the surface as it is) [NS95]. The graded triangulation as is cannot be applied here, as Delaunay criterion is not defined for surfaces, hence initial triangulation cannot be carried out as described.

6.1 Input : Linear Coons surface

The input for the surface triangulation routine implemented here is the bilinear Coons surface/patch [GH73, RA90].

If the four boundary curves in (u,v) space, $P(u,0)$, $P(u,1)$, $P(0,v)$ and $P(1,v)$ are known, and a bilinear blending function is used for the interior of the surface patch, a linear Coons surface is obtained. It is given by

$$\begin{aligned} P'(u,v) &= (1-v)P(u,0) + vP(u,1) + (1-u)P(0,v) + uP(1,v) \\ &\quad -(1-u)(1-v)P(0,0) - (1-u)vP(0,1) \\ &\quad -u(1-v)P(1,0) - uvP(1,1) \end{aligned}$$

where $0 \le u \le 1$, $0 \le v \le 1$. It can be easily seen that, at the corners,

$$P'(0,0) = P(0,0),\ P'(0,1) = P(0,1),\ P'(1,1) = P(1,1),\ P'(1,0) = P(1,0)$$

and along the boundaries,

$$P'(u,0) = P(u,0),\ P'(u,1) = P(u,1),\ P'(0,v) = P(0,v),\ P'(1,v) = P(1,v)$$

More compactly, the equation can written as

$$P'(u,v) = \begin{bmatrix} 1-u & u & 1 \end{bmatrix} \begin{bmatrix} -P(0,0) & -P(0,1) & P(0,v) \\ -P(1,0) & -P(1,1) & P(1,v) \\ P(u,0) & P(u,1) & 0 \end{bmatrix} \begin{bmatrix} 1-v \\ v \\ 1 \end{bmatrix}$$

A mapping function then gives (x,y,z) corresponding to every (u,v). In the implementation here, the four boundary curves happen to be B-spline curves (BSCs). The functions $(1-u)$, u, $(1-v)$, and v are called blending functions because they blend the boundary curves to produce the internal shape of the surface. The linear Coons surface is the simplest of the Coons surfaces. Examples of these surfaces are Figures 6.1 and 6.2. A C code for IITZeus using X/Motif was written as a part of this project to obtain this.

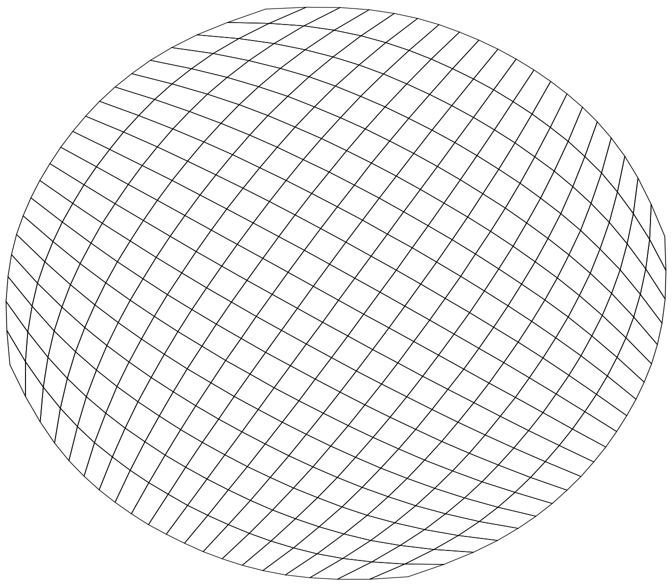

Figure 6.1: Linear Coons surface (20×20)

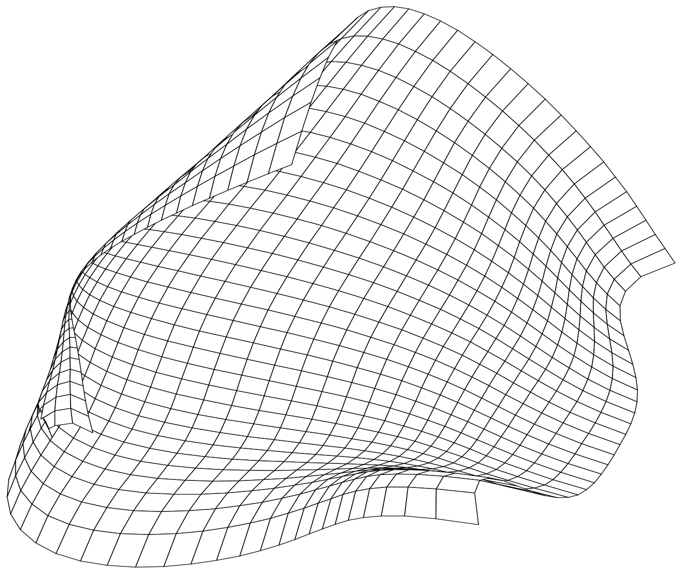

Figure 6.2: Linear Coons surface (30×30)

6.2 Algorithm and implementation

1. Four BSCs with C^0 continuity (end-points being common) are read from the database, and a Coons Surface is made from it, i.e., a one-to-one mapping function from (u,v)-space ($[0,1] \times [0,1]$) to (x,y,z)-space ($\Re^3$) is established.

2. The 2D (u,v)-space is now triangulated using graded triangulation described earlier. Here, the major change is the use of the actual 3D lengths of the segments as one of the additional grading parameters. This means that whenever a trial point in (u,v)-space is chosen, the corresponding point in $\Re^3$ is determined and its 3D distances with the neighbouring points in the front is also used to re-adjust the original point. This is done by simple iteration till some kind of convergence is obtained between the two consecutive points.

3. Thus, the triangulation takes place in pseudo-3D and whenever a point in (u,v)-space is inserted, it is mapped back to $\Re^3$and the corresponding triangle is displayed.

Chapter 7

Results and Conclusions

The results generated by the graded triangulation routine implemented in ANSI C can be seen here. The routine is integrated with the IITZeus environment which facilitates the input and output of the data. Each result of the graded triangulation routine is compared with the AFM routine already implemented earlier.

For all the results obtained by the graded triangulation shown here, the same gradation function $L = (\alpha L_1 + \beta L_2 + \gamma L_3)/3$ is used. It can be clearly seen from these results that the triangulation obtained is somewhat dense around the specified boundary and relatively sparse away from the boundary and this gradation occurs much more smoothly than the corresponding AFM triangulation.

The time comparison between graded triangulation and AFM is not shown as it greatly depends on the number of triangles generated which is different for both the methods even for the same test data depending on the gradation function. However, in the case of same number of triangles in the output, the time difference between the two is small, AFM being only slightly faster.

It can be further concluded that the user now has a much better control over the triangulation generated by means of the gradation function to be specified, which is absent in both AFM and Delaunay triangulation.

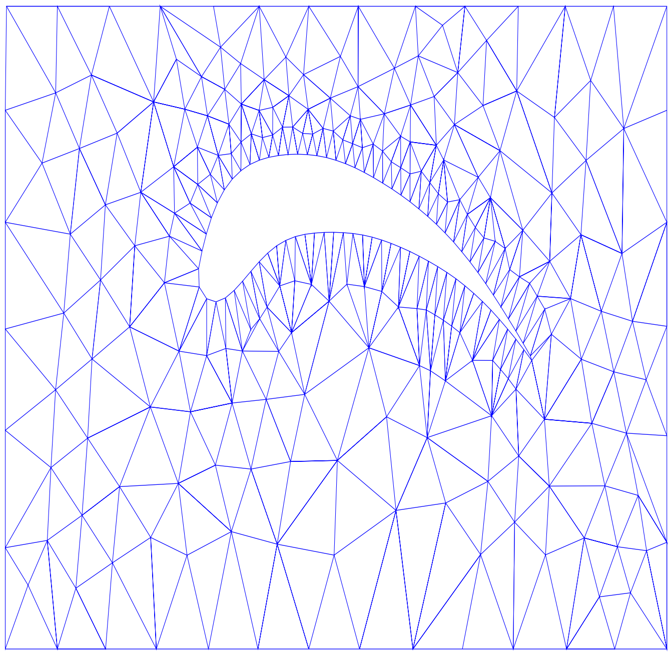

Figure 7.1: Graded triangulation - Test data 1

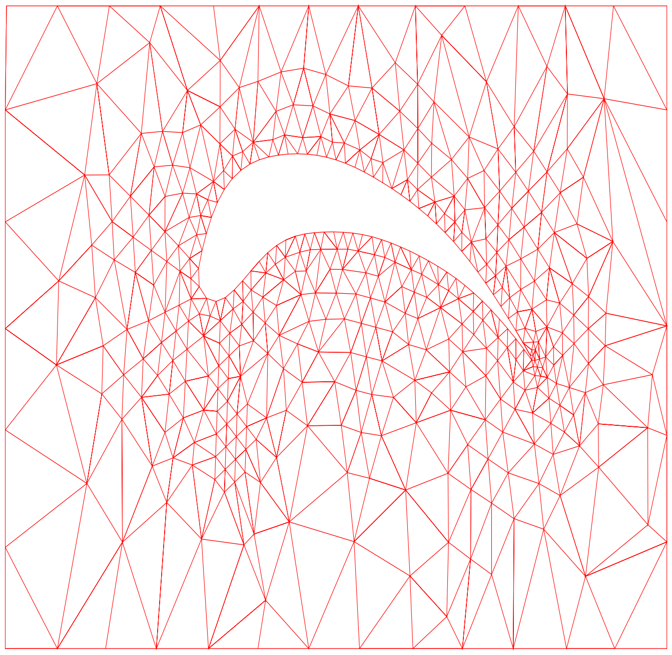

Figure 7.2: Advancing front triangulation - Test data 1

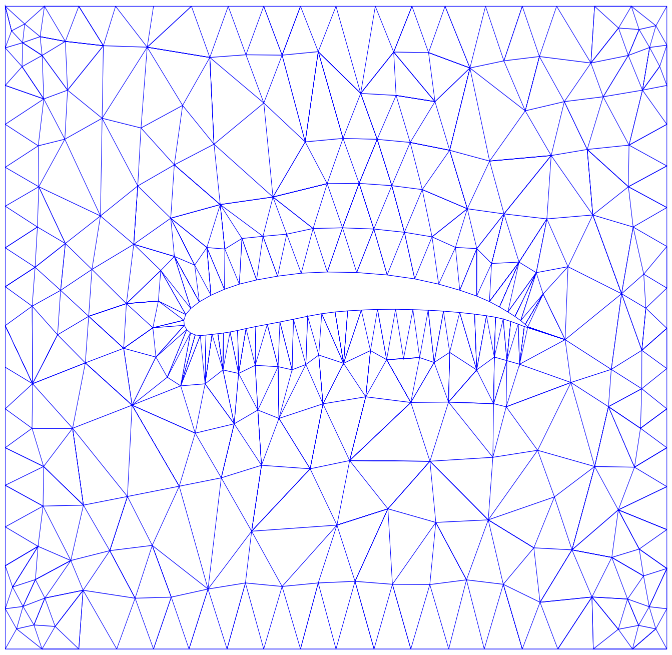

Figure 7.3: Graded triangulation - Test data 2

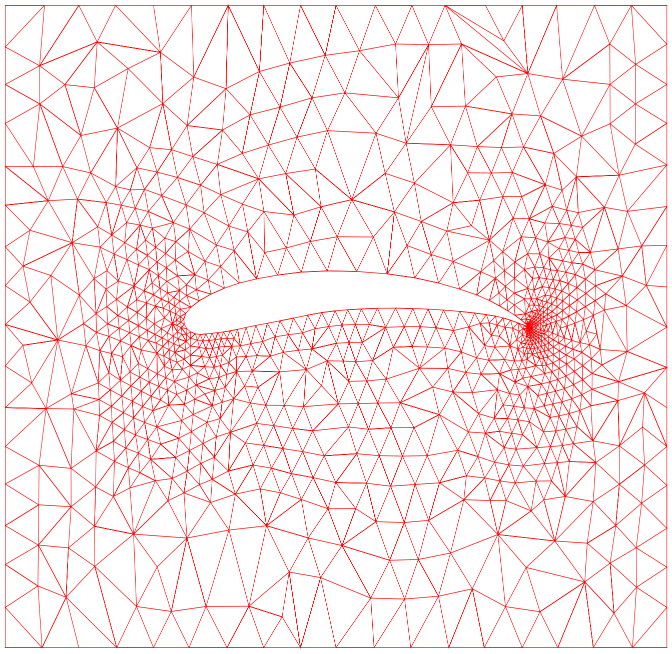

Figure 7.4: Advancing front triangulation - Test data 2

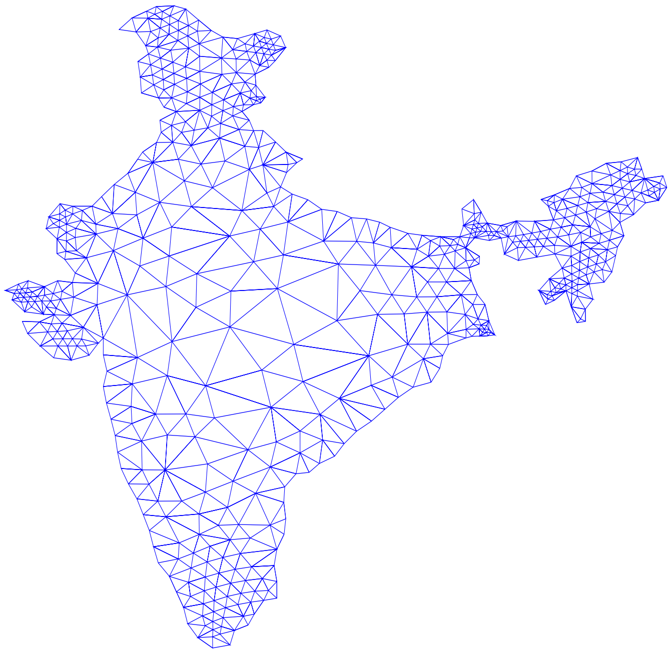

Figure 7.5: Graded triangulation - Test data 3

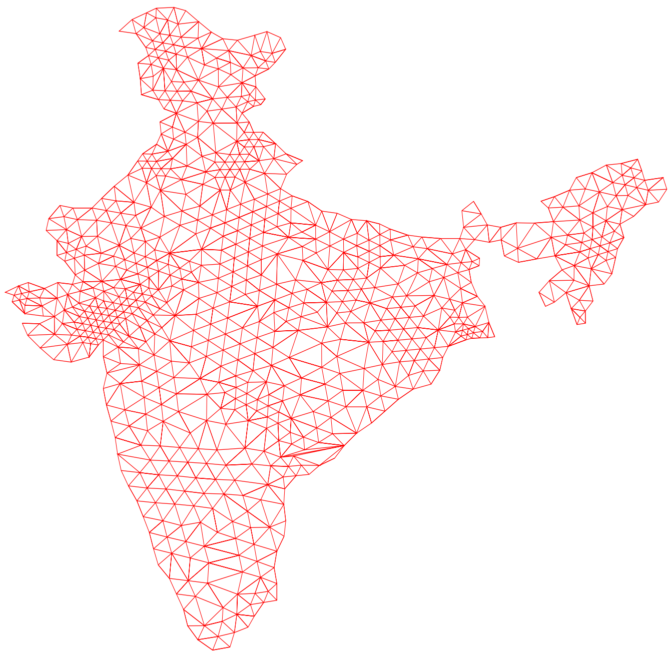

Figure 7.6: Advancing front triangulation - Test data 3

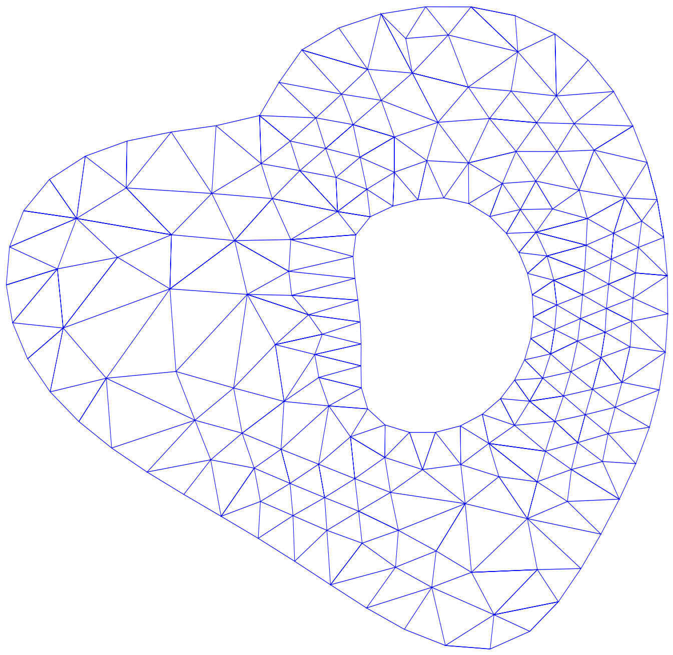

Figure 7.7: Graded triangulation - Test data 4

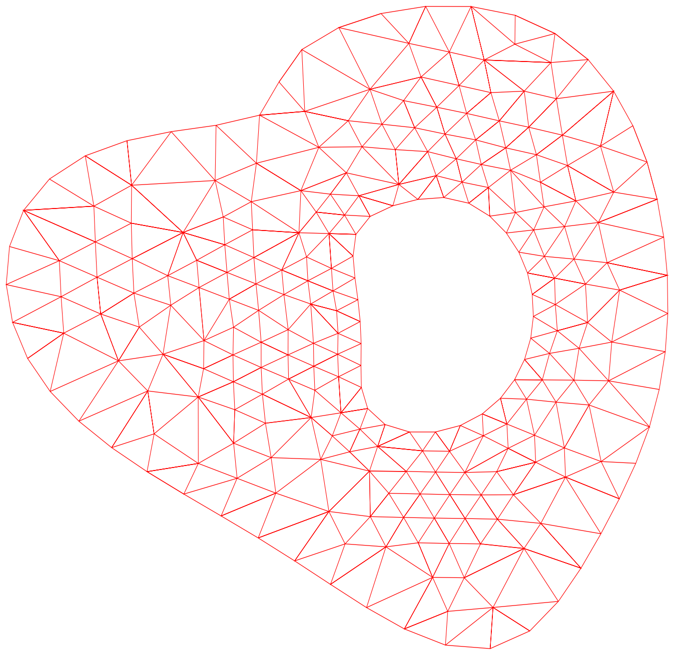

Figure 7.8: Advancing front triangulation - Test data 4

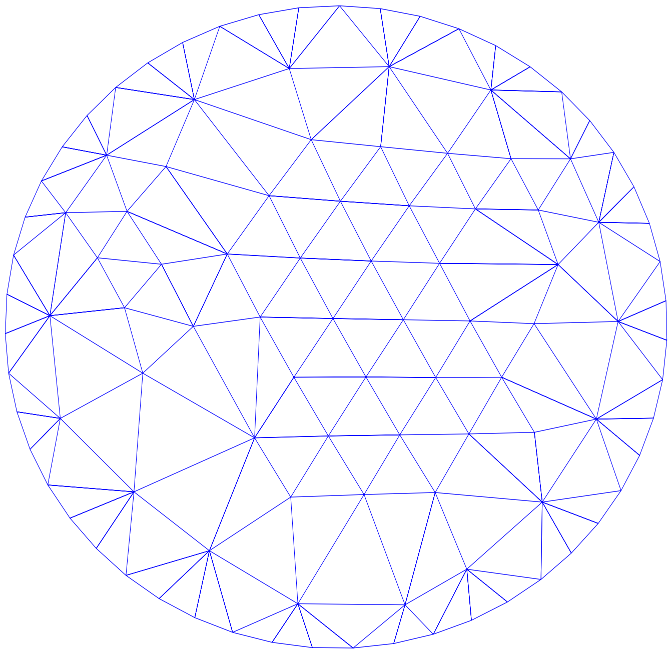

Figure 7.9: Graded triangulation - Test data 5

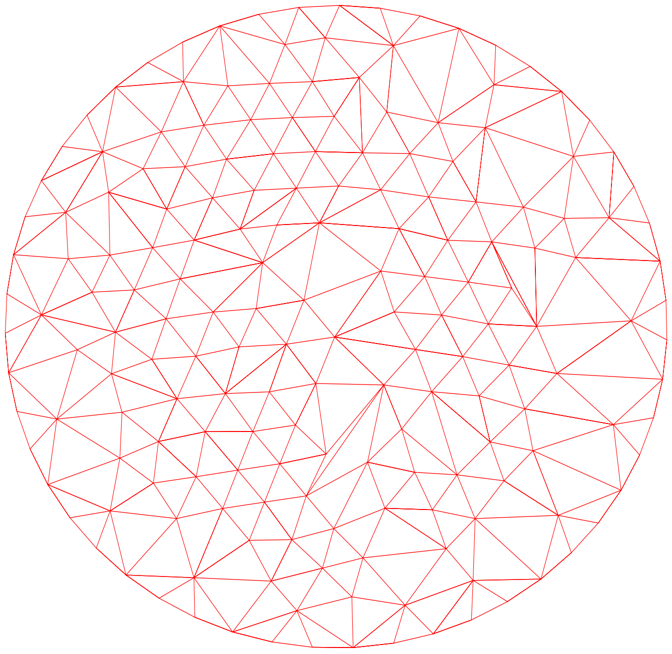

Figure 7.10: Advancing front triangulation - Test data 5

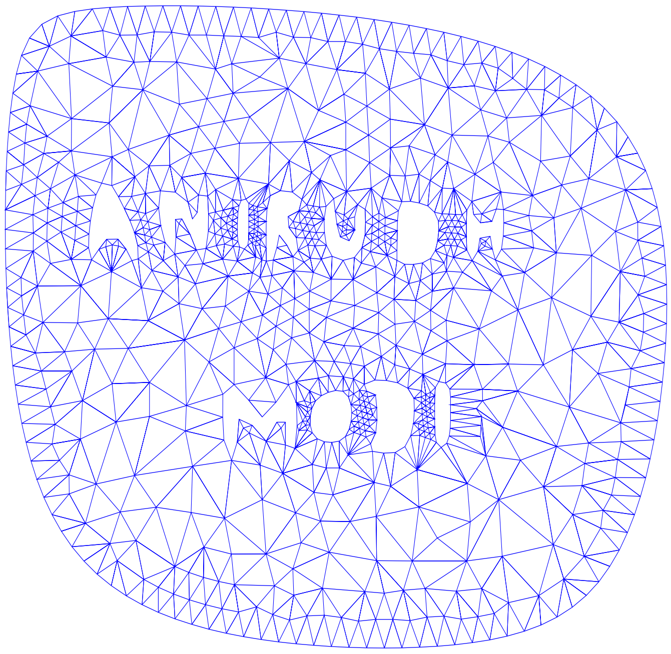
Figure 7.11: Graded triangulation - Test data 6

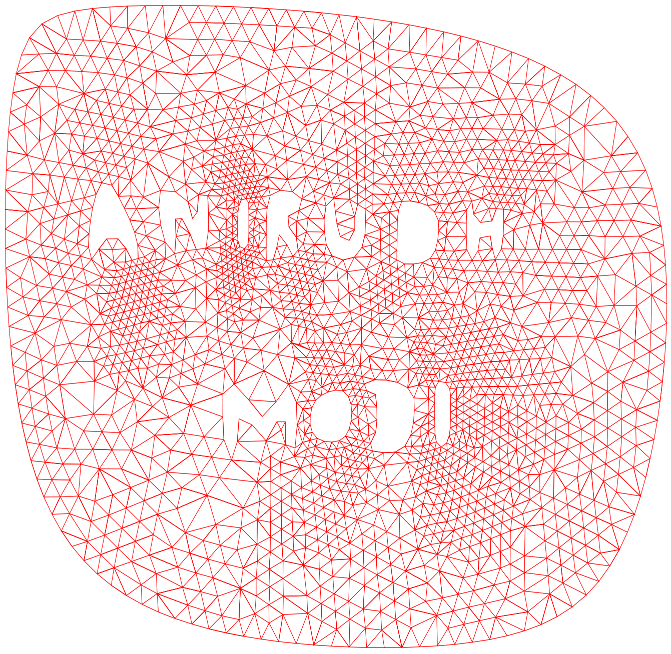
Figure 7.12: Advancing front triangulation - Test data 6

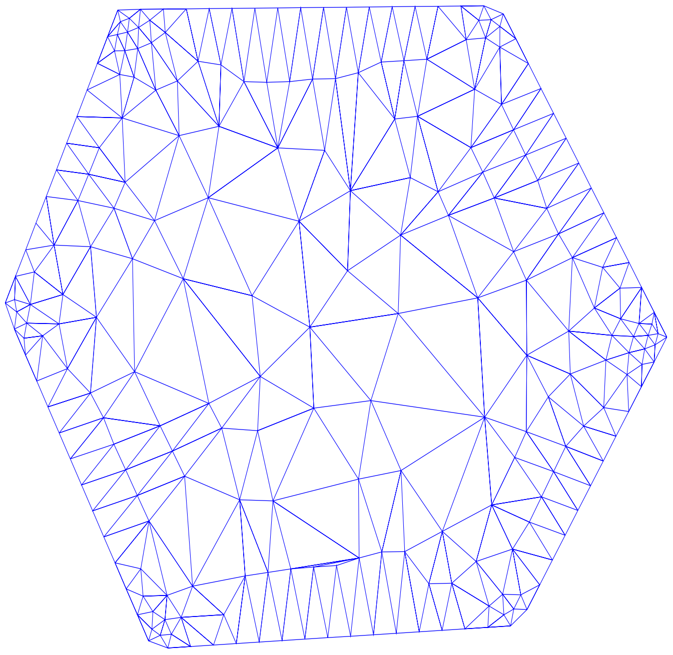

Figure 7.13: Graded triangulation - Test data 7

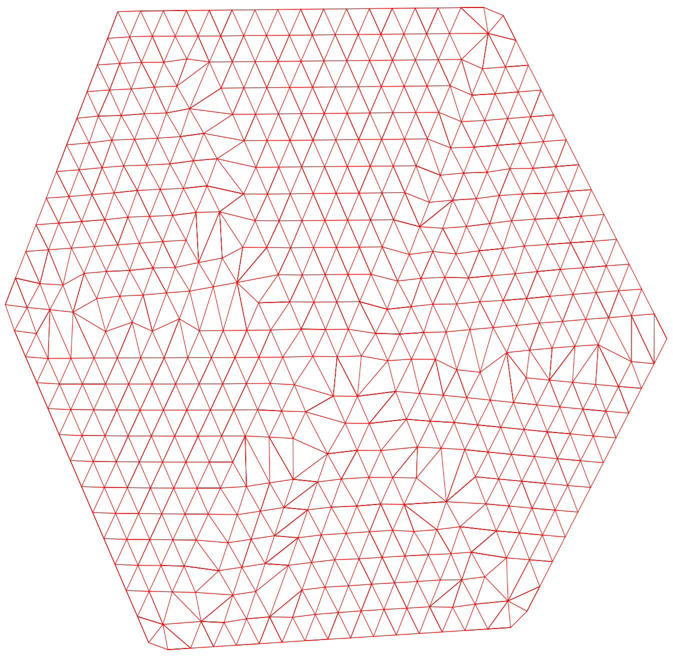

Figure 7.14: Advancing front triangulation - Test data 7

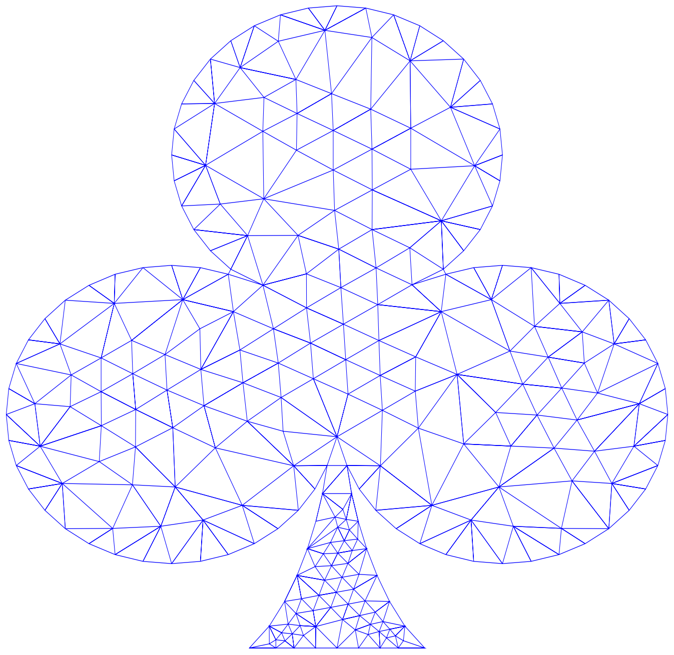

Figure 7.15: Graded triangulation - Test data 8

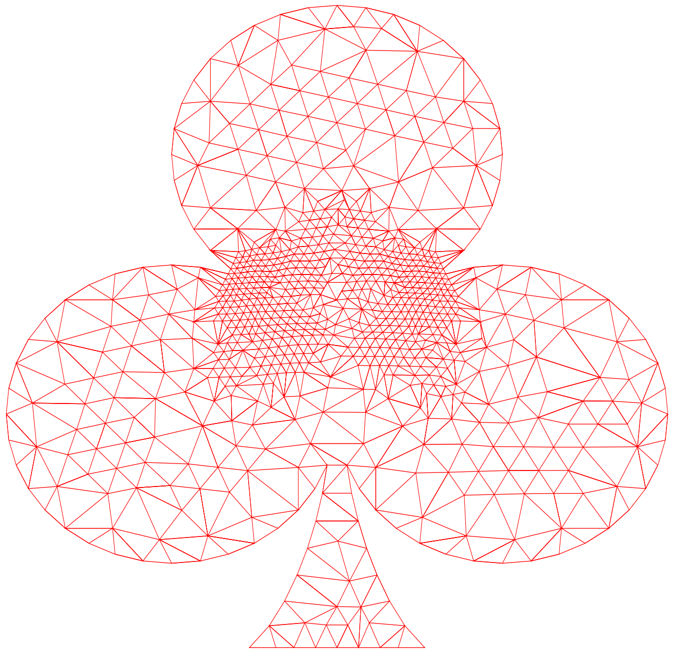

Figure 7.16: Advancing front triangulation - Test data 8

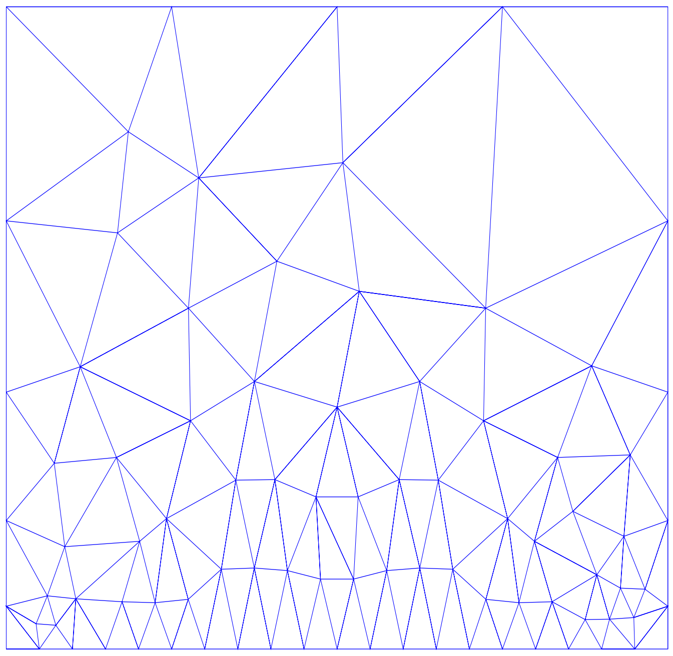

Figure 7.17: Graded triangulation - Test data 9

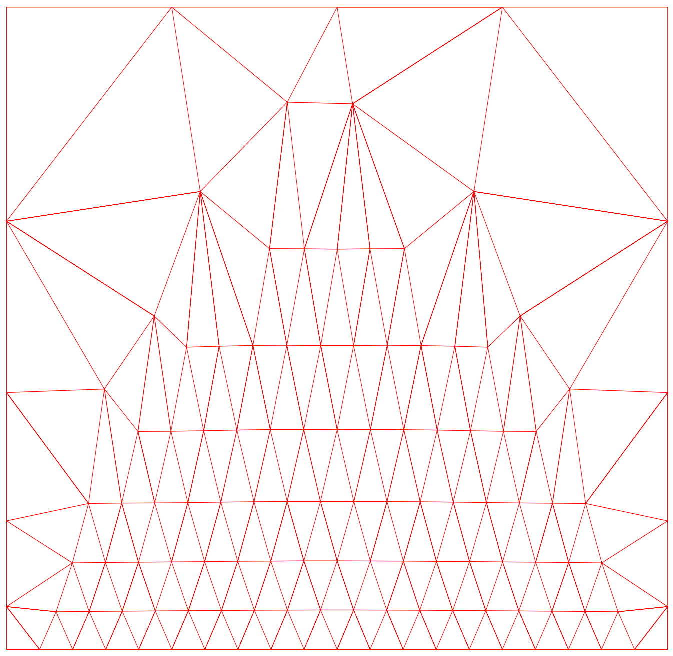

Figure 7.18: Advancing front triangulation - Test data 9

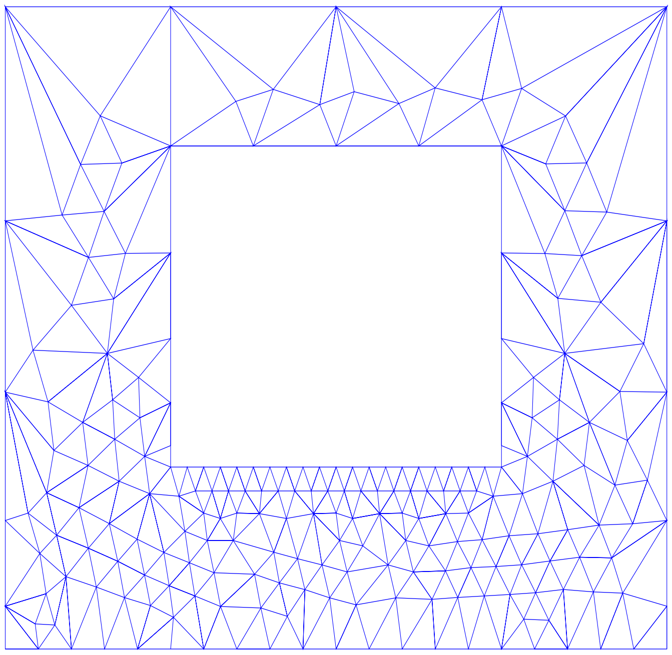

Figure 7.19: Graded triangulation - Test data 10

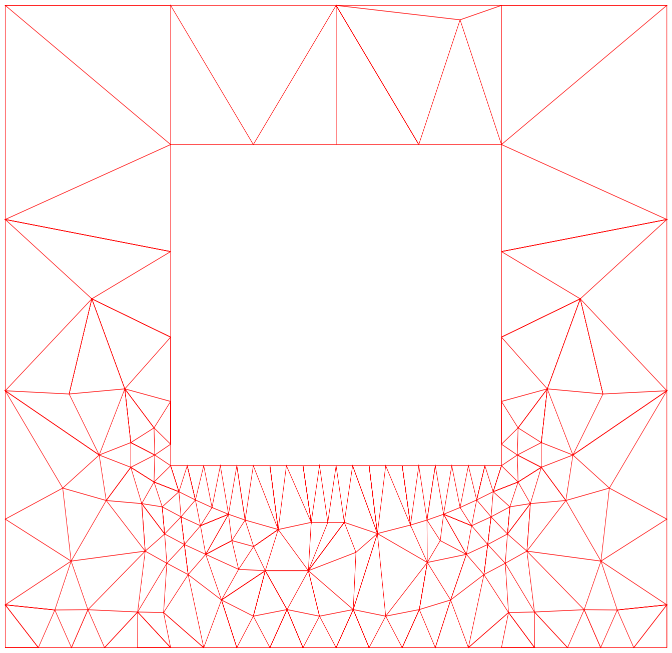

Figure 7.20: Advancing front triangulation - Test data 10

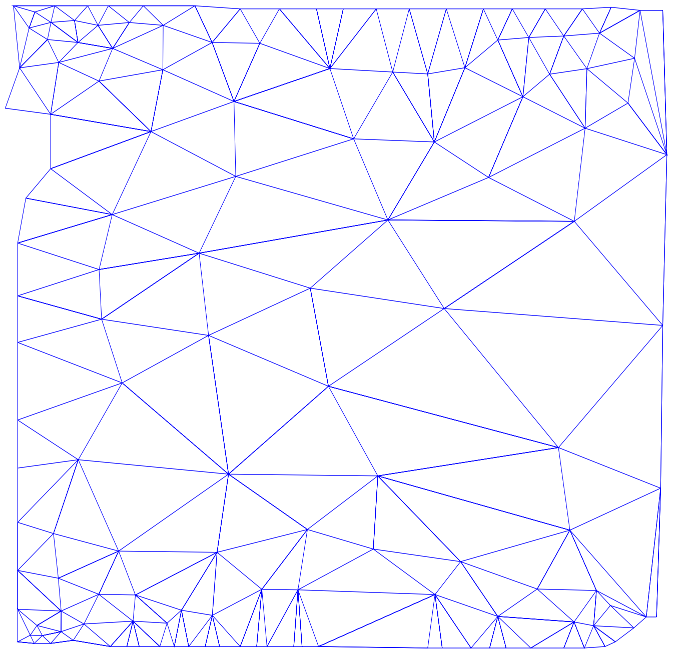

Figure 7.21: Graded triangulation - Test data 11

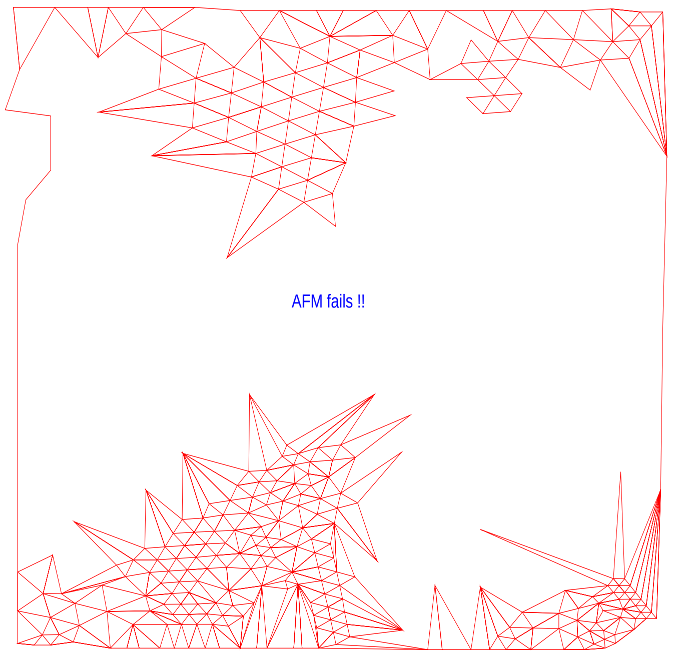

Figure 7.22: Advancing front triangulation - Test data 11

Chapter 8

Scope of the Project

- The graded triangulation routine implemented here uses two main data structures for storage. The TRI structure [Nan95] is used for the initial Delaunay triangulation and the EDGE structure [Sha96] is used for the AFM. This leads to many redundancies in storage and inefficiency in coding of several routines due to constant interaction between the two data structures. A new data structure optimized for the above work will help greatly in speeding up the routine and reducing memory requirements.

- The efficiency of the routine for determining the background triangle in which the trial point is located can be greatly enhanced by using a better (although more complex) search algorithm. The present implementation is a brute force method which searches through the entire list of triangles for the location of the given point. The information of the neighbours of each triangle in the list stored by the TRI structure can be possibly utilized to our advantage here.

- The input routine can be extended to include B-spline surface (BSS) and other parametrically defined surfaces as well, as well as for closed surfaces.

- The algorithm can be extended to use the Jacobian (curvature) of the surface as an additional parameter for point insertion. This should give more control on the triangulation obtained, and hence a better mesh.

References

[AHET91] A.S. Arcilla, J. Häuser, P.R. Eisman, and J.F. Thompson. Numerical Grid Generation Techniques in Computational Fluid Dynamics and Related Fields. In *Proceedings*, Barcelona, Spain, 1991. Elsevier Science Publishers.

[Bow81] A. Bowyer. Computing Dirichlet tessellations. *The Computer Journal*, 24(2):162–166, 1981.

[Geo91] P.L. George. *Automatic Mesh Generation : Applications to Finite Element Method*. John Wiley & Sons, 1991.

[GH73] William J. Gordon and Charles A. Hall. Construction of Curvilinear Co-ordinate Systems and Applications to Mesh Generation. *International Journal for Numerical Methods in Engineering*, 7:461–477, July 1973.

[HS88] D.G. Holmes and D.D. Snyder. The Generation of Unstructured Triangular Meshes using Delaunay Triangulation. In *Numerical Grid Generation in Computational Fluid Mechanics '88*, pages 643–652, Miami, 1988. Pineridge Press, Swansea.

[LP88] R. Löhner and P. Parikh. Three-Dimensional Grid Generation by Advancing Front Method. *International Journal for Numerical Methods in Fluids*, 8:1135–1149, 1988.

[LS80] D.T. Lee and B.J. Schachter. Two Algorithms for Constructing a Delaunay Triangulation. *International Journal of Computer and Information Sciences*, 9(3):219–242, February 1980.

[Nan95] Milind Nanal. Mesh Generation for FEM. M.Tech. Dissertation, Indian Institute of Technology - Bombay, Department of Civil Engineering, April 1995.

[NS95] Kazuhiro Nakahashi and Dmitri Sharov. Direct Surface Triangulation using the Advancing Front Method. *AIAA-95-1686-CP*, pages 442–451, 1995.

[RA90] David F. Rogers and J. Alan Adams. *Mathematical Elements for Computer Graphics*. McGraw Hill International, 1990.

[Sch93] Larry L. Schumaker. Triangulations in CAGD. *IEEE Computer Graphics and Applications*, pages 47–52, January 1993.

[Sha96] Anurag Sharma. Unstructured Mesh Generation by Advancing Front Method. B.Tech. Project Report, Indian Institute of Technology - Bombay, Department of Aerospace Engineering, April 1996.

[Tho84] J.F. Thompson. Grid Generation Techniques in Computational Fluid Dynamics. *AIAA Journal*, 22(11):1505–1523, 1984.

[Wea88] N.P. Weatherill. A Method for Generating Irregular Computational Grids in Multiply Connected Planar Domains. *International Journal for Numerical Methods in Fluids*, 8:181–197, 1988.