Chapter 12
Unstructured Methods

12.1 Introduction

Unstructured mesh techniques occupy an important niche in grid generation. The major feature of unstructured grids consists, in contrast to structured grids, of a nearly absolute absence of any restrictions on grid cells, grid organization, or grid structure. Figuratively speaking, unstructured grids manifest the domination of anarchy while structured grids demonstrate adherence to order. The concept of unstructured grids allows one to place the grid nodes locally irrespective of any coordinate directions, so that curved boundaries can be handled with ease and local regions in which the solution is turbulent or its variations are large can be resolved with a selective insertion of new points without unduly affecting the resolution in other parts of the physical domain.

Unstructured grid methods were originally developed in solid mechanics. Nowadays, these methods influence many other fields of application beyond solid modeling, in particular, computational fluid dynamics, where they are becoming widespread.

Unstructured grids can, in principle, be composed of cells of arbitrary shapes built by connecting a given point to an arbitrary number of other points, but are generally formed from tetrahedra and hexahedra (triangles and quadrilaterals in two dimensions). The advantages of these grids lie in their ability to deal with complex geometries, while allowing one to provide natural grid adaptation through the insertion of new nodes.

At the present time, the methods of unstructured grid generation have reached the stage in which three-dimensional domains with complex geometry can be successfully meshed. The most spectacular theoretical and practical achievements have been connected with the techniques for generating tetrahedral (or triangular) grids. There are at least two basic approaches that have been used to generate these meshes: Delaunay and advancing-front. This chapter presents a review of some popular techniques realizing these approaches.

V.D. Liseikin, *Grid Generation Methods*, Scientific Computation,
DOI 10.1007/978-3-319-57846-0_12

Note that the chapter addresses only some general aspects of unstructured grid methods. The interested reader who wishes to learn more about the wider aspects of unstructured grids should study, for example, the monographs by Carey (1997) George and Borouchaki (1998a, b), Frey and George (2008), and Lo (2015).

12.2 Methods Based on the Delaunay Criterion

Much attention has been paid in the development of methods for unstructured discretizations to triangulations which are based upon the very simple geometrical constraint that the hypersphere of each n-dimensional simplex defined by $n + 1$ points is void of any other points of the triangulation. For example, in three dimensions, the four vertices of a tetrahedron define a circumsphere which contains no other nodes of the tetrahedral mesh. This restriction is referred to as the Delaunay or incircle criterion, or the empty-circumcircle property. Triangulations obeying the Delaunay criterion are called Delaunay triangulations. They are very popular in practical applications, owing to the following properties being valid in two dimensions:

(1) Delaunay triangles are nearly equilateral;
(2) the maximum angle is minimized;
(3) the minimum angle is maximized;
(4) the triangulation is unique if the points are in a general position, i.e. no four points are cyclic;
(5) if every triangle in a triangulation is non-obtuse, it is a Delaunay triangulation;
(6) any two-dimensional triangulation can be transformed into a Delaunay triangulation by locally flipping of the diagonals of adjacent triangles.

These properties give some grounds to expect that the grid cells of a Delaunay triangulation are not too deformed.

Based on a sound geometrical concept and the optimality properties, Delaunay triangulation has important applications in many fields, including data visualization, terrain modelling, mesh generation, surface reconstruction and structural networking for arbitrary point sets. The popularity of Delaunay triangulation is attributed to its nice geometric properties as a dual of Voronoi tessellation and the speed with which it can be constructed in two or higher dimensions.

The Delaunay criterion itself is not an algorithm for mesh generation. It merely provides a rule for connecting a set of existing points in space to form a triangulation. As a result, although the boundary of the domain is well specified, it is necessary to devise a scheme to determine the number and the locations of node points to be inserted within the domain of interest.

The Delaunay criterion does not give any indication as to how the grid points should be defined and connected. One more drawback of the Delaunay criterion is that it may not be possible to realize it over the whole region with a prespecified boundary triangulation. This disadvantage gives rise to two grid generation approaches of constrained triangulation which preserve the boundary connectivity and take into account the Delaunay criterion. In the first approach of constrained Delaunay triangulation, the Delaunay property is overriden at points close to the boundaries, and consequently the previously generated boundary grid remains intact. Alternatively, or in combination with this technique, points can be added in the form of a skeleton to ensure that breakthroughs of the boundary do not occur. Another approach, which observes the Delaunay criterion over the whole domain, is to postprocess the mesh by recovering the boundary simplexes which are missed during the generation of the Delaunay triangulation and by removing the simplexes lying outside the triangulated domain.

There are a number of algorithms for generating unstructured grids based on the Delaunay criterion in constrained or unconstrained forms.

Some methods for Delaunay triangulations are formulated for a preassigned distribution of points which are specified by means of some appropriate technique, in particular, by a structured grid method. These points are connected to obtain a triangulation satisfying certain specific geometrical properties which, to some extent, are equivalent to the Delaunay criterion.

Many Delaunay triangulations use an incremental Bowyer–Watson algorithm which can be readily applied to any number of dimensions. It starts with an initial triangulation of just a few points. The algorithm proceeds at each step by adding points one at a time into the current triangulation and locally reconstructing the triangulation. The process allows one to provide both solution-adaptive refinement and mesh quality improvement in the framework of the Delaunay criterion. The distinctive characteristic of this method is that point positions and connections are computed simultaneously.

One more type of algorithm is based on a sequential correction of a given triangulation, converting it into a Delaunay triangulation.

12.2.1 Dirichlet Tessellation

A very attractive means for generating a Delaunay triangulation of an assigned set of points is provided by a geometrical construction first introduced by Dirichlet (1850).

Consider an arbitrary set of points P_i, $i = 1, \ldots, N$, in the n-dimensional domain. For any point P_i, we define a region $V(P_i)$ in R^n characterized by the property that it is constituted by the points from R^n which are nearer to P_i than to any other P_j, i.e.

$$V_i = \{x \in R^n | \mathrm{d}(x, P_i) \leq \mathrm{d}(x, P_j) \ , \quad i \neq j \ , \quad j = 1, \ldots, N\} \ ,$$

where $\mathrm{d}(a, b)$ denotes the distance between the points a and b. These areas V_i are called the Voronoi polyhedrons (see Fig. 12.1 for $n = 2$). Thus, the polyhedra are intersections of half-spaces, and therefore they are convex, though not necessarily bounded. The set of Voronoi polyhedra corresponding to the collection of points P_i is called the Voronoi diagram or Dirichlet tessellation. The common boundary of

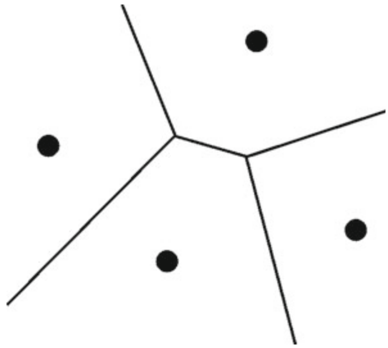


Fig. 12.1 Voronoi polyhedron for 4 points

two facing Voronoi regions $V(P_i)$ and $V(P_j)$ is an $(n-1)$-dimensional polygon. A pair of points P_i and P_j whose Voronoi polyhedra have a face in common is called a configuration pair. By connecting only the contiguous points, a network is obtained. In this network, a set of $n+1$ points which are contiguous with one another forms an n-dimensional simplex. The circumcenter, i.e. the center of the hypersphere, of any simplex is a vertex of the Voronoi diagram. The hypersphere of the simplex is empty, that is, there is no point inside the hypersphere. Otherwise, this point would be nearer to the circumcenter than the points on the hypersphere. Thus, the set of simplexes constructed in such a manner from the Dirichlet tessellation constitutes a new tessellation which satisfies the Delaunay criterion and is, therefore, a Delaunay triangulation. The boundary of the Delaunay triangulation built from the Voronoi diagram is the convex hull of the set of points P_i (see Fig. 12.2 for $n = 2$).

It should be noted that Delaunay triangulations and Dirichlet tessellations can be considered the geometrical duals of each other, in the sense that for every simplex S_i, there exists a vextex P_i of the tessellation and, conversely, for every Voronoi region $V(P_j)$, there exists a vertex P_j of the triangulation. In addition, for every edge of the triangulation, there exists a corresponding $(n-1)$-dimensional segment of the Dirichlet tessellation.

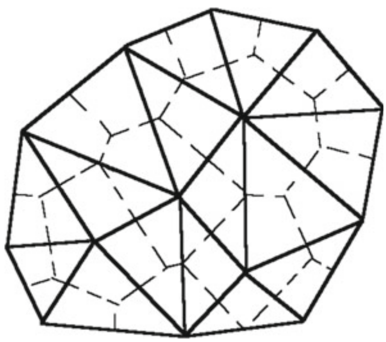


Fig. 12.2 Voronoi diagram and Delaunay triangulation

12.2.2 Incremental Techniques

The empty-hypercircle criterion of the Delaunay triangulation can be utilized to create incremental triangulation algorithms for arbitrary dimensions. Recall that by the Delaunay triangulation of a set V_N of N points in n-dimensional space, we mean the triangulation of V_N by simplexes with the vertices taken from V_N such that no point lies inside the hypersphere of any n-dimensional simplex.

Here, two incremental methods are presented. In the first method, a new n-dimensional simplex is constructed during each stage of the triangulation, using the given set of points for this purpose. In the second technique, each step produces several simplexes which are generated after inserting a new point.

A-Priori-Given Set of Points

Let a set of points V_N in a bounded n-dimensional domain X^n be given. We assume that these points do not lie in any $(n-1)$-dimensional hyperplane. The incremental technique starts by taking an $(n-1)$-dimensional face e (edge in two dimensions and triangle in three dimensions), commonly the one with the smallest size, and constructing hyperspheres through the vertices of e and any one of the remaining points of V_N. One of these hyperspheres formed by a point, say, P_1, does not contain any point of V_N inside it. The $(n-1)$-dimensional simplex e and P_1 define a new n-dimensional simplex. In the next step, the $(n-1)$-dimensional simplex e is taken out of consideration. The algorithm stops, and the triangulation is complete, when every boundary face corresponds to the side of one simplex and every internal $(n-1)$-dimensional simplex forms the common face of precisely two n-dimensional simplexes. It is clear that this algorithm is well suited to generate a Delaunay triangulation with respect to a prescribed boundary triangulation.

The set of points used to generate the triangulation can be built with a structured method or an octree approach, or by embedding the domain into a Cartesian grid. However, the most popular approach is to utilize the strategy of a sequential insertion of new points.

Modernized Bowyer–Watson Technique

Another incremental method, proposed by Baker (1989) and which is a generalization of the Bowyer–Watson technique, starts with some triangulation, not necessarily that of Delaunay, of the set of N points $V_N = \{P_i | i = 1, \dots, N\}$ by an assembly of simplexes $T_N = \{S_j\}$. For any simplex $S \in T_N$, let R_S be the circumradius and Q_S the circumcenter of S. In the sequential-insertion technique, a new point P is introduced inside the convex hull of V_N. Let $B(P)$ be the set of the simplexes whose circumspheres contain the point P, i.e.

$$B(P) = \{S | S \in T_N,\ \mathrm{d}(P, Q_S) < R_S\} ,$$

where $\mathrm{d}(P, Q)$ is the distance between P and Q. All these simplexes from $B(P)$ form a region $\Gamma(P)$ surrounding the point P. This region is called the generalized cavity. The maximal simply connected area of $\Gamma(P)$ that contains the point P is called the principal component of $\Gamma(P)$ and denoted by Γ_P. The point P is checked to determine if it is visible from all boundary segments of the principal component or if it is obscured by some simplex. In the former case, the algorithm generates new simplexes associated with P by joining all of the vertices of the principal component with the point P. In the latter case, either this point is rejected and a new one is introduced or the principal component Γ_P is reduced by excluding the redundant simplexes from $B(P)$ to obtain an area whose boundary is not obscured from P by any simplex. Then, the new simplexes are formed as in the former case. The union of these simplexes and those which do not form the reduced region of the retriangulation defines a new triangulation of the set of $N+1$ points $V_{N+1} = V_N \bigcup \{P\}$. In this manner, the process proceeds by inserting new points, checking visibility, adjusting the principal component, and generating new simplexes. The new triangulation differs from the previous one only locally around the newly inserted point P.

In two dimensions, we have that if the initial triangulation is the Delaunay triangulation, then the region $\Gamma(P)$ is of star shape, and consequently the boundary is visible from the point P and each step of the Bowyer–Watson algorithm produces a Delaunay triangulation. Thus, in this case, the Bowyer–Watson algorithm is essentially a "reconnection" method, since it computes how an existing Delaunay triangulation is to be modified because of the insertion of new points. In fact, the algorithm removes from the existing grid all the simplexes which violate the empty-hypersphere property because of the insertion of the new point. The modification is constructed in a purely sequential manner, and the process can be started from a very simple initial Delaunay triangulation enclosing all points to be triangulated (for example, that formed by one very large simplex or one obtained from a given set of boundary points) and adding one point after another until the necessary requirements for grid quality have been satisfied.

12.2.3 Approaches for Insertion of New Points

The sequential nature of the Bowyer–Watson algorithm gives rise to a problem of choosing the position where the new point in the existing mesh will be inserted, because a poor point distribution can eventually lead to an unsatisfactory triangulation. The new point should be chosen according to some suitable geometrical and physical criteria which depend on the existing triangulation and the behavior of the physical solution. The geometrical criteria commonly consist in the requirement for the grid to be smooth and for the cells to be of a standard uniform shape and a necessary size. The physical criterion commonly requires the grid cells to be concentrated in some specific zones as the zones of turbulence, large solution variations, or large solution errors. With respect to the geometrical criterion of generating uniform cells, the vertices and segments of the Dirichlet tessellation are promising locations for placing a new point, since they represent a geometrical locus which falls, by construction, midway between the triangulation points. Thus, in order to control the size and shape of the grid cells, two different ways in which the new point is inserted are commonly considered. In the first, the new point is chosen at the vertex of the Voronoi polyhedron corresponding to the "worst" simplex. In the second way, the new point is inserted into a segment of the Dirichlet tessellation, in a position that guarantees the required size of the newly generated simplexes.

12.2.4 Two-Dimensional Approaches

This subsection discusses the major techniques delineated in Sects. 12.2.1–12.2.3 for generating planar triangulations based on the Delaunay criterion.

Voronoi Diagram

The Delaunay triangulation has a dual set of polygons referred to as the Voronoi diagram or the Dirichlet tessellation.

The Voronoi diagram can be constructed for an arbitrary set of points in the domain. Each polygon of the diagram corresponds to the point that it encloses. The polygon for a given point is the region of the plane which is closer to that point than to any other points. These regions have polygonal shapes and the tessellation of a closed domain results in a set of nonoverlapping convex polygons covering the convex hull of the points. It is clear that the edge of a Voronoi polygon is equidistant from the two points which it separates, and is thus a segment of the perpendicular bisector of the line joining these two points. The Delaunay triangulation of the given set of points is obtained by joining with straight lines all point pairs whose Voronoi regions have an edge in common. For each triangle formed in this way, there is an associated vertex of the Voronoi diagram which is at the circumcentre of the three points which form the triangle. Thus, each Delaunay triangle contains a unique vertex of the Voronoi diagram, and no other vertex within the Voronoi structure lies within the circle centered at this vertex. Figure 12.2 depicts the Voronoi polygons and the associated Delaunay triangulation.

It is apparent from the definition of a Voronoi polygon that degeneracy problems can arise in the triangulation procedure when

(1) three points of a potential triangle lie on a straight line;
(2) four or more points are cyclic.

These cases are readily eliminated by rejecting or slightly moving the point which causes the degeneracy from its original position.

Incremental Bowyer–Watson Algorithm

The two-dimensional incremental technique, introduced independently by Bowyer (1981) and Watson (1981), triangulates a set of points in accordance with the requirement that the circumcircle through the three vertices of a triangle does not contain any other point. The accomplishment of this technique starts from some Delaunay triangulation which is considered to be an initial triangulation. The initial triangulation commonly consists of a square divided into two triangles which contain the

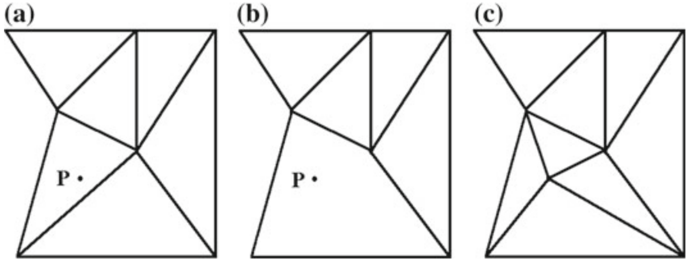


Fig. 12.3 Stages of the planar incremental algorithm

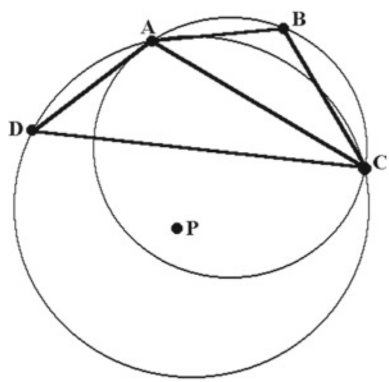
given points. With this starting Delaunay triangulation, a new grid node is chosen from a given set of points or is found in accordance with some user-specified rule to supply new vertices. In order to define the grid cells which contain this point as a vertex, all the cells whose circumcircles enclose the inserted point are identified and removed. The union of the removed cells forms the region which is referred to as the Delaunay or inserting cavity. A new triangulation is then formed by joining the new point to all boundary vertices of the inserting cavity created by the removal of the identified triangles. Figure 12.3 represents the stages of the planar incremental algorithm.

The distinctive feature of the two-dimensional Delaunay triangulations is that all edges of the Delaunay cavity are visible from this inserted point, i.e. each point of the edges can be joined to it by a straight line which lies in the cavity.

Properties of the Planar Delaunay Cavity

In order to prove the fact that all boundary edges of the Voronoi cavity are visible from the introduced point, we consider an edge AB lying on the boundary of the cavity. Let ABC be the triangle with the vertices A, B, and C, which lies in the Delaunay cavity formed by the insertion of the point, denoted by P (Fig. 12.4). It is obvious that all edges of triangle ABC are visible if P lies inside the triangle. Let P

Fig. 12.4 Illustration of the inserted point P and the triangles of the Delaunay cavity



lie outside the triangle. As this triangle lies in the Delaunay cavity, it follows that P lies inside circle ABC. In this case, the quadrilateral whose vertices are the points $ABCP$ is convex. Thus, P has to be visible from edge AB unless we have a situation like the one depicted in Fig. 12.4, in which some triangle ACD separates the edge from P. As triangle ACD belongs to the initial Delaunay triangulation, the vertex D lies outside circle ABC. However, since a chord of a circle subtends equal angles at its circumference, we readily find that P belongs to circle ACD, i.e. the triangle lies inside the Delaunay cavity formed by P. Thus, triangle ACD does not prevent those edges of ABC which are the boundary edges of the cavity from being visible from P. Repeating the argument with the other triangles, the number of which is finite, we come to the conclusion that there are no triangles between the boundary of the Delaunay cavity and P which do not lie in the cavity. Also, we find that the Delaunay cavity is simply connected. We emphasize that these facts are valid if the original triangulation satisfies the Delaunay criterion.

Thus, in accordance with the incremental algorithm, the Delaunay cavity is triangulated by simply connecting the inserted point with each of the nodes of the initial grid that lie on the boundary of the cavity. The union of these triangles with those which lie outside of the cavity (Fig. 12.3c) completes one loop of the incremental grid construction. The subsequent steps are accomplished in the same fashion.

It is apparent that in two dimensions, the creation of these new cells results in a Delaunay triangulation, i.e. the Delaunay criterion is valid for all new triangles. Here, we present a schematic proof of this fact.

Let AB be an edge of the Delaunay cavity formed by the insertion of point P. Suppose that the new triangle ABP does not satisfy the Delaunay criterion. Then, there exists some point D on the same side of AB as P and which lies inside circle ABP (Fig. 12.5). Consider the original triangle that had AB as an edge. There are two possibilities: either ABD is this original triangle or there is another point, say, E, on the cavity boundary lying outside circle ABP. In the former case, P lies outside circle ABD, i.e. triangle ABD does not lie in the Delaunay cavity, and consequently edge AB is not the edge of the cavity, contrary to our assumption. In the latter case, arc ABP lies inside circle ABE. However, this contradicts the assumption that

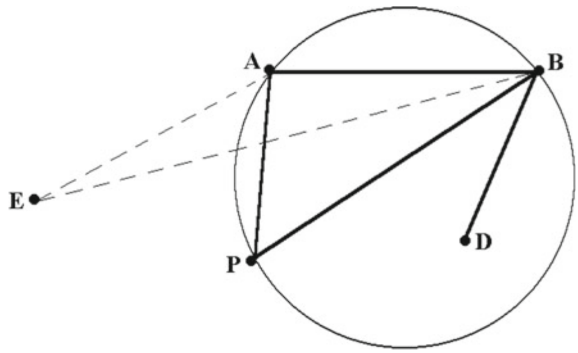


Fig. 12.5 Illustration of the proof that the Delaunay criterion is satisfied by all new triangles created by the incremental algorithm

the original triangulation was of Delaunay type. Therefore,,,, circle ABP does not contain other points, i.e. the triangle ABP satisfies the Delaunay criterion.

Thus, we find that the planar Bowyer–Watson algorithm is a valid procedure for generating Delaunay triangulations. One more issue that has received attention is that the point placement selected to generate Delaunay triangulations can be used to generate meshes with a good aspect ratio.

Initial Triangulation

Because the mesh points are introduced in a sequential manner, in the initial stages of this construction, an extremely coarse grid containing a small subset of the total number of mesh points and consisting of a small number of very large triangles can be chosen. For example, for generating grids in general two-dimensional domains, an initial triangulation can be formed by dividing a square lying in the domain or containing it into two triangles. Then, interior and boundary points are successively added to build successive triangulations until the necessary requirements of domain approximation are observed.

It is desirable to make the initial triangulation boundary-conforming, i.e. all boundary edges are included in the triangulation. One natural way is to triangulate initially only the prescribed boundary nodes, by means of the Bowyer–Watson algorithm. Since the Delaunay triangulation of a given set of points is a unique construction, there is no guarantee that the triangulation built through the boundary points will be boundary-conforming. However, through repeated insertion of new mesh points at the midpoints of the missing boundary edges, a boundary-conforming triangulation may be obtained. Another way to maintain boundary integrity is obtained by rejecting any point that would result in breaking boundary connectivity.

Diagonal-Swapping Algorithm

Diagonal swapping is a topological operation in which the diagonal of a quadrilateral formed by two adjacent triangles is swapped to the other position to improve the overall quality of the two triangles. The diagonal-swapping algorithm makes use of the equiangular property of a Delaunay-type triangulation, which states that the minimum angle of each triangle in the mesh in maximized.

Assuming we have some triangulation of a given set of points, the swapping algorithm transforms it into a Delaunay triangulation by repeatedly swapping the positions of the edges in the mesh in accordance with the equiangular property. For this purpose, each pair of triangles which constitutes a convex quadrilateral is considered. This quadrilateral produces two of the required triangles when one takes the diagonal which maximizes the minimum of the six interior angles of the quadrilaterals, as shown in Fig. 12.6. Each time an edge swap is performed, the triangulation becomes more equiangular. The end of the process results in the most equiangular (the Delaunay) triangulation.

This technique based on the Delaunay criterion retriangulates a given triangulation in a unique way, such that the minimum angle of each triangle in the mesh is maximized. This has the advantage that the resulting meshes are optimal for the

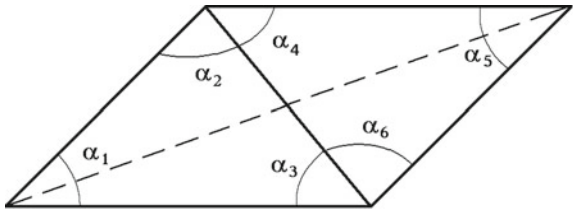


Fig. 12.6 The triangulation which maximizes the minimum angle. The dashed line indicates a possible original triangulation

given point distribution, in that they do not usually contain many extremely skewed cells.

12.2.5 Constrained Form of Delaunay Triangulation

One way to ensure that the boundary triangulation remains intact in the process of retriangulation by inserting new points is to use a constrained version of the Delaunay triangulation algorithm that does not violate the point connections made near the boundary.

Principal Component

For the purpose of generating a constrained two-dimensional triangulation, we consider the modernized Bowyer–Watson algorithm for an arbitrary triangulation T that may not satisfy the Delaunay criterion. Let P be a new, introduced point. The Delaunay cavity is the area constituted by all triangles whose circumcircles contain P. Let this be denoted by $\Gamma(P)$.

An important fact is that the Delaunay cavity created by the introduction of the point P contains no points other than P in its interior. In order to show this, we consider a point A in the triangulation T that is a vertex of at least one triangle in $\Gamma(P)$. If there is a triangle $S \notin \Gamma(P)$ that has A as a vertex, then the point A is not an interior point of $\Gamma(P)$. Thus, we need to show that there exists such a triangle. Let $\{S_i\}$ be the set of all triangles that have A as a vertex, and let C_i be the circumcircle associated with triangle S_i. Now $S_i \in \Gamma(P)$ if and only if the new point P lies inside C_i. Thus, for vertex A to be an interior point of $\Gamma(P)$, point P must lie inside $\bigcap C_i$. However, if the point A is an interior point of $\Gamma(P)$, then the interior of $\bigcap C_i$ is empty, since the vertex A is the only point that lies on all the circles of $\{S_i\}$. Thus, at least one triangle of $\{S_i\}$ does not lie in $\Gamma(P)$, and hence the vertex A is not an interior point of $\Gamma(P)$.

In the case of a general triangulation, the cavity $\Gamma(P)$ need no longer be simply connected. For the purpose of retriangulation, we consider the maximal simply connected region of the cavity that contains the new point P. This region is called the principal component of the Delaunay cavity and is designated by Γ_P.

It is apparent that the principal component possesses the property that all its boundary edges are visible from P. To prove this, we first note that Γ_P is not empty,

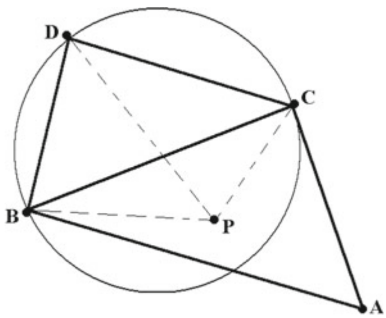


Fig. 12.7 Illustration of the principal component

since it includes the triangle containing P. Let this be the triangle ABC (Fig. 12.7). Now consider all neighboring triangles sharing a common edge with the triangle ABC. In particular, let triangle BCD lie in Γ_P. Point P must, therefore, lie inside circle BCD. As points P, B, C, and D define a convex quadrilateral, all edges of this quadrilateral are visible from P. Continuing this process by means of a tree search through all triangles in Γ_P, we clearly see that all edges of Γ_P are visible from P.

Formulation of the Constrained Triangulation

Now we can formulate the generation of a constrained planar Delaunay triangulation developed by Baker (1989).

We assume that certain triangles of a triangulation T are fixed, in particular, those adjacent to the boundary. Let this subset of T be denoted by $\overline{T}$. The triangles from $\overline{T}$ do not participate in the building of any Delaunay cavity, i.e. if the cavity created by the introduction of a new point contains one or more of the fixed triangles, we restrict the reconnections to the part of the cavity that does not contain any fixed triangle. Let $\Upsilon(P)$ be this part of the cavity, i.e. $\Upsilon(P) = \Gamma(P) - \overline{T}$. By Υ_P, we denote the maximal simply connected region of $\Upsilon(P)$ that contains P. In analogy with Γ_P, we call the region Υ_P the principal component of $\Upsilon(P)$. It is clear that the principal component Υ_P exists only if P does not lie inside any of the triangles belonging to the collection $\overline{T}$ of the fixed triangles.

It is apparent that the boundary edges of the principal component Υ_P are visible from P. As the analogous fact has been proved for Γ_P, we can restrict our consideration to the case $\Gamma_P \bigcap \overline{T} \neq \emptyset$. Let the edges of the principal component Γ_P be given by $\overline{A_1, A_2}$, $\overline{A_2, A_3}, \ldots, \overline{A_{n-1}, A_n}$, $\overline{A_n, A_1}$, where $\{A_i\}i = 1, n$ are the vertices on the boundary of Γ_P. These edges, and consequently the vertices A_i, are visible from P. The subcavity obtained by removing one of the triangles from Γ_P contains at most three new edges. These internal edges lie wholly inside the cavity Γ_P and divide Γ_P into disjoint polygonal regions. The principal component Υ_P is the polygonal region that contains the point P, and this polygon is made up of one of these internal cavity edges and other edges which come from the cavity boundary. The vertices of the polygon containing P must, therefore, remain visible from P. Hence, all edges of the polygon are visible from P. By repeating this argument with other triangles removed from Γ_P, we conclude that the boundary edges of Υ_P are visible from P.

Now the vertices of Υ_P can be connected with P, thus building the constrained retriangulation. This retriangulation keeps the fixed triangles of $\overline{T}$ intact.

Boundary-Conforming Triangulation

A key requirement of a mesh generation procedure is to ensure that the mesh is boundary-conforming, i.e. the edges of the assembly of triangles conform to the boundary curve. The procedure of constrained triangulation allows one to keep a subset of the boundary triangles, built from the edges forming the boundary, intact. These boundary triangles can be generated by any one of the suitable procedures. Thus, the resulting triangulation will be boundary-conforming and its interior triangles obey the Delaunay criterion.

Another approach developed by Weatherill and Hassan (1994) to applying the Delaunay criterion to generate boundary-conforming grids consists in recovering the boundary edges which are missing during the process of Delaunay triangulation and then deleting all triangles that lie outside the domain.

12.2.6 Point Insertion Strategies

The Bowyer–Watson algorithm proceeds by sequentially inserting a point inside the domain at selected sites and reconstructing the triangulation so as to include new points. This subsection presents two approaches to sequential point insertion which provide a refinement of planar Delaunay triangulations. In both cases, bounds on some measures of grid quality, such as the minimum angle, the ratio of maximum to minimum edge length, and the ratio of circumradius to inradius, are estimated.

Point Placement at the Circumcenter of The Maximum Triangle

One simple but effective approach consists in placing a new point at the circumcenter of the cell with the largest circumradius and iterating this process until the maximum circumradius is less than some prescribed threshold. In this way, by eliminating bad triangles, the quality of the grid is improved at every new point insertion, terminating with a grid formed only by suitable triangles. In this subsection, it will be shown that the Bowyer–Watson incremental algorithm together with point insertion at the circumcenters of maximal triangles will lead to a triangulation with a guaranteed level of triangle quality.

Unconstrained Triangulation

Let $\{T_n\}$, $n = 0, 1, \ldots,$ be a sequence of Delaunay triangulations built by the repeated application of the Bowyer–Watson algorithm with point insertion at the circumcenter of the maximal triangle. By the maximal triangle of a triangulation, we mean the triangle with the maximum value of its circumradius. We assume that the initial Delaunay triangulation T_0 conforms to a prescribed set of boundary edges.

Now let $l_n,\ L_n,\ n = 0, 1, \ldots,$ be the minimum and maximum edge lengths, respectively, of T_n, and let R_n be the radius of the maximal triangle of T_n. Furthermore, for any triangle S, we denote its circumradius by R_S and its inradius by r_S. Thus, $R_n = \max\{R_S,\ S \in T_n\}$. We have the following relations:

(1) $R_{i+1} \leq R_i$;
(2) when $R_{n-1} \geq l_0$, then $l_n = l_0$, and when $R_{n-1} < l_0$, then $l_n = R_{n-1}$;
(3) when $R_n \leq l_0$, then $L_n/l_n \leq 2$, $\theta \geq 30°$ for all angles of the triangulation T_n, and $\min R_S/r_S \leq 2 + 4\sqrt{3}$ for all triangles S of T_n.

To prove the first relation, we consider an edge e_n of the Delaunay cavity of the triangulation T_n formed by an inserted point P. There exist triangles S_1 and S_2 in T_n which share the common edge e_n, such that S_1 lies inside while S_2 lies outside the Delaunay cavity. Let S_1 be defined by the points $A,\ B$, and C and S_2 be defined by the points $A,\ B$, and D. Then, edge e_n is the line segment $\overline{AB}$. Since P lies outside circle ABD, P lies on the same side of e_n as C. If the center of circle ABP lies on the same side of e_n as D, then angle APB is obtuse and, consequently, the circumradius of triangle ABP is smaller than the circumradius of triangle ABD. We denote these circumradii by R_{ABP} and R_{ABD}, respectively.

If the center of circle ABP lies on the same side of e_n as C, then the angle θ_1 subtended by chord AB at C is less than the angle θ_2 subtended at P. Since the centers of circles ABP and ABC lie on the same side of $\overline{AB}$ as points C and P, it follows that $\theta_1 < \pi/2$ and $\theta_2 < \pi/2$. If the length of chord $\overline{AB}$ is l, then

$$R_{ABP} = \frac{l}{2\sin\theta_2} < \frac{l}{2\sin\theta_1} = R_{ABC} \ ,$$

where R_{ABC} is the circumradius of ABC. Thus, we obtain

$$R_{ABP} < R_{ABD} \quad \text{and} \quad R_{ABP} < R_{ABC} \ .$$

Since this is true for all edges of the Delaunay cavity, we obtain the proof of the first relation, that the maximum circumradius R_n decreases, i.e. $R_{n+1} \leq R_n$, with strict inequality if there is only one triangle with the maximum radius R_n. As there can be only a limited number of maximal triangles in T_n, after several applications of the procedure, we obtain $R_{n+k} < R_n$.

It follows that the maximum radius can be reduced to any required size after a sufficiently large number of iterations. When R_n falls below the value of l_0, so that $l_{n+1} = R_n$, we obtain the following obvious inequality:

$$L_{n+1} \leq 2R_{n+1} \leq 2R_n = 2l_{n+1} \ . \tag{12.1}$$

It is evident that repeated point insertion at the circumcenter reduces the value $\lambda = L_n/l_n$ to a value no greater than 2. The upper bound of 2 for λ is achieved when $R_n \leq l_0$. Let $\theta_{\min}$ be the minimum angle. We have

$$\sin\theta_{\min} \geq \frac{l_{n+1}}{2R_{n+1}} , \tag{12.2}$$

with equality if the minimum edge length of any maximal triangle is equal to l_{n+1}, the minimum edge length for the triangulation T_{n+1}. From the inequalities (12.1) and (12.2), we obtain

$$\sin\theta_{\min} \geq \frac{l_{n+1}}{2R_{n+1}} = \frac{R_n}{2R_{n+1}} \geq \frac{1}{2} , \tag{12.3}$$

so that

$$\theta_{\min} \geq \pi/6 .$$

For each triangle, the quantity $\mu = R/r$, where R is the circumradius and r is the inradius, is a characteristic of cell deformity. The maximum value of μ occurs for an isosceles triangle with an angle between sides of $\theta_{\min}$ and assumes the value

$$\mu_{\max} = \frac{1}{2\cos\theta_{\min}(1-\cos\theta_{\min})} .$$

From (12.3), we obtain

$$\mu \leq 2 + 4/\sqrt{3}$$

after a sufficient number of retriangulations with the insertion of new points at the circumcenters of maximal triangles.

These considerations prove the properties (2) and (3) stated above.

Generalized Choice of the Insertion Triangles

In the approach considered, a new point is inserted at the circumcentre of the largest triangle. The choice of the insertion triangle, namely the triangle where the point is inserted, can be formulated in accordance with more general principles.

One simple formulation is based on the specification of a function $f(\boldsymbol{x})$ which prescribes a measure of grid size or quality, say the radius of the circumscribed circle, at the point $\boldsymbol{x}$. The actual expression for $f(\boldsymbol{x})$ can be obtained by interpolating prescribed nodal values over a convenient background mesh. The function $f(\boldsymbol{x})$ defines a quantity $\alpha(S)$ for each triangle S:

$$\alpha(S) = \frac{R_S}{f(Q_S)} ,$$

where Q_S is the position of the centre of the circle circumscribed around the triangle S. The largest value of $\alpha(S)$ determines the choice of the insertion triangle S. By repeatedly inserting the new point at the circumcenters of such triangles, it is possible eventually to reach a mesh in which $\max_S \alpha(S) < 1$.

Voronoi-Segment Point Insertion

The second approach proposed by Rebay (1993) to placing a new point consists in inserting the point along a segment of the Dirichlet tessellation. In contrast to the first approach, in which the position of the inserted point is predetermined, and the required cell size is reached after a number of iterations, this technique provides an opportunity to generate one or possibly several new triangles having, from the very beginning, the size prescribed for the final grid. This is achieved by choosing a suitable position for point placement in the Dirichlet tessellation, between a triangle whose circumradius falls below the required value and a neighboring triangle whose circumradius is still too large. This point insertion results in almost equilateral triangles over most of the interior of the domain.

Formulation of the Algorithm

At each stage of the process of generating the triangulations $T_n,\ n = 1, 2, \ldots,$ the triangles of T_n are divided into two groups, which are referred to as the groups of accepted (small enough) and nonaccepted (too large) triangles, respectively. In most cases, the accepted triangles are the boundary triangles and those whose circumradii are below $3/2$ times the prescribed threshold. The remaining triangles constitute the group of nonaccepted triangles.

The algorithm proceeds by always considering a maximal nonaccepted triangle which borders one of the accepted triangles (Fig. 12.8). Let ABC be the accepted triangle and ADB the nonaccepted triangle. The Voronoi segment connecting the circumcenters of these triangles is the interval EF which is perpendicular to the common edge AB and divides it into two equal parts. In the algorithm, a new point X is inserted on the Voronoi segment edge EF in a position chosen so that the triangle

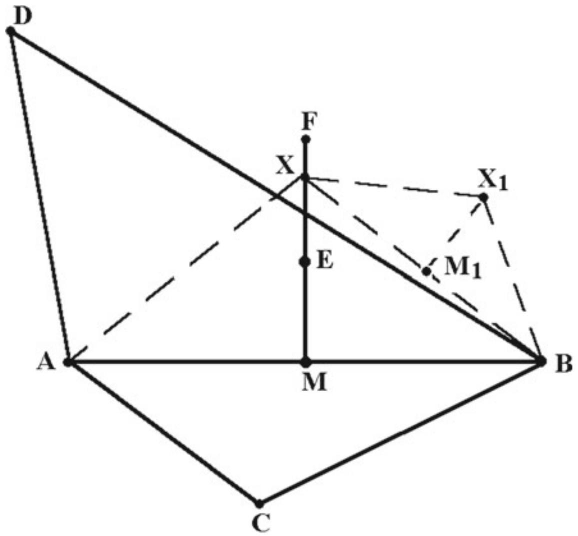


Fig. 12.8 Voronoi-segment point insertion

formed by connecting X with A and B has the prescribed size. This point is inserted into the interval between the midpoint M of the common edge and the circumcenter F of the nonaccepted triangle ADB.

Let p be one half the length of edge AB, and q the length of FM. As point F is the circumcenter of the triangle ADB, we find that $q \geq p$. Let f_M be the prescribed threshold value for the circumradius at the point M. It may seem that we can locate the new point X on segment EF at the intersection of EF with the circle that passes through points A and B and has a radius equal to f_M. However, it might happen that this exact value f_M for the circumradius is not appropriate, since any circle through A and B has a radius $\rho \geq p/2$. Furthermore, a real intersection point X exists only for circles having a radius ρ smaller than that of the circle passing through AB and F, i.e. $\rho \leq (p^2 + q^2)/2q$. For these reasons, the circumradius for the triangle AXB is defined by the equation

$$R_{AXB} = \min\left[\max(f_M, p)\ ,\quad \frac{p^2 + q^2}{2q}\right] . \qquad (12.4)$$

Since

$$\frac{p^2 + q^2}{2q} = \frac{(p - q)^2 + 2pq}{2q} \geq p\ ,$$

we find that $R_{AXB} \geq p$. In accordance with the algorithm, the new point X will lie on the interval EF between M and F at a distance

$$d = R_{AXB} + \sqrt{(R_{AXB})^2 - p^2} \qquad (12.5)$$

from the point M.

Properties of the Triangulation

The condition

$$R_{AXP} \leq \frac{p^2 + q^2}{2q}$$

and (12.5) ensure that $d \leq q$. We also have, from (12.5), that $d \geq p$. Angle AXB is a right angle when $d = p$, and it decreases as d increases.

If the accepted triangle ABC is equilateral, then angle AFB must be no greater than $2\pi/3$, since otherwise the Delaunay triangulation would have given rise to an edge connecting C to F.

At the first stage, we expect $p \ll q$. Recall that the threshold of f_M is such that $f_M < p < 3f_M/2$. It follows that $f_M < p \leq (p^2 + q^2)/2q$, and hence $d = p$ and $R_{AXB} = p$. Thus, triangle AXB has a right angle at vertex X. Since $2 < 3f_M/2$, triangle AXB will be tagged as accepted and each segment AX and XB will be a candidate for the next accepted triangle, built in the same way as AXB. Now we denote the quantity p, equal to one half the length of the accepted edge of the ith iteration, by p_i, and thus $p_1 = p_0/\sqrt{2}$. Analogously, we use d_i, R_i, and M_i at the ith iteration of the procedure. It turns out that on repeating the procedure, p_i and d_i show the following behavior:

$$p_i \to \sqrt{3} f_{M_i}/2 \;, \qquad d_i \to 3 f_{M_i}/2 \;,$$

i.e. the generated triangles tend to become equilateral, with circumradius f_{M_i}. To show this, let

$$f_{M_n} = \left(\frac{2}{\sqrt{3}} + \epsilon_n\right) p_n = \left(\frac{2}{\sqrt{3}} + \epsilon_{n+1}\right) p_{n+1} \;. \tag{12.6}$$

If $|\epsilon_n|$ is sufficiently small, we obtain $p_n < f_{M_n}$ so that $R_n = f_{M_n}$ and, from (12.5),

$$d_n = f_{M_n} + \sqrt{f_{M_n}^2 - p_n^2} \;.$$

Furthermore, we have

$$4p_{n+1}^2 = p_n^2 + d_n^2 = 2\left(f_{M_n}^2 + f_{M_n}\sqrt{f_{M_n}^2 - p_n^2}\right) .$$

Thus,

$$\frac{p_{n+1}^2}{p_n^2} = \frac{1}{2}\frac{f_{M_n}^2}{p_n^2} + \frac{1}{2}\frac{f_{M_n}}{p_n}\sqrt{\frac{f_{M_n}^2}{p_n^2} - 1} \;.$$

Using (12.6), we obtain

$$\frac{p_{n+1}^2}{p_n^2} = \frac{1}{2}\left(\frac{2}{\sqrt{3}} + \epsilon_n\right)^2 + \frac{1}{2}\left(\frac{2}{\sqrt{3}} + \epsilon_n\right)\sqrt{\left(\frac{2}{\sqrt{3}} + \epsilon_n\right)^2 - 1} \;,$$

which results in

$$\frac{p_{n+1}}{p_n} = 1 + \frac{3}{4}\sqrt{3} + \epsilon_n + O(\epsilon_n^2) \;.$$

From (12.6), we also have

$$\frac{p_{n+1}}{p_n} = \frac{2/\sqrt{3} + \epsilon_n}{2/\sqrt{3} + \epsilon_{n+1}} \;.$$

Comparing the last two equations and neglecting terms $O(\epsilon_n^2)$, we find that

$$\epsilon_{n+1} \simeq -\epsilon_{n/2} \;.$$

Thus, for $|\epsilon_n|$ sufficiently small, the algorithm ensures that $\epsilon_n \to 0$ and

$$p_n \to \sqrt{3} f_{M_n}/2 \; .$$

Therefore, it can be expected that a large number of the interior triangles will be nearly equilateral. Close to the boundary, there may be isosceles right-angled triangles, and in regions where the boundary has large curvature, there may be some obtuse triangles. A maximum angle of 120° and minimum angle of 30° may be realized by an obtuse triangle formed when the vertex D of a nonaccepted triangle is sufficiently close to an active edge.

In analogy with the first approach to inserting new points, the choice of the triangle into which the new point is inserted can be modified by introducing a quality measure function $f(\boldsymbol{x})$ and a corresponding control quantity $\alpha(S)$.

12.2.7 Surface Delaunay Triangulation

A surface Delaunay triangulation is defined by analogy with the planar Delaunay triangulation.

Let P_i be the vertices of the surface triangulation T. A triangle S from T satisfies the Delaunay criterion if the interior of the circumsphere through the vertices of S and centered on the plane formed by S does not contain any points. If all triangles satisfy the Delaunay criterion, then the triangulation T is called a surface Delaunay triangulation.

In practice, all methods for planar Delaunay triangulations are readily modified and extended for a surface Delaunay triangulation taking into account various surface geometric characteristics (see Frey and George (2008) and Lo (2015)).

12.2.8 Three-Dimensional Delaunay Triangulation

In three dimensions, the network of the Delaunay triangulation is obtained by joining the vertices of the Voronoi polyhedrons that have a common face. Each vertex of a Voronoi polyhedron is the circumcenter of a sphere that passes through four points which form a tetrahedron and no other point in the construction can lie within the sphere.

Unconstrained Technique

The most popular three-dimensional algorithm providing a Delaunay structure is the one based on the Bowyer–Watson sequential process: each point of the grid is introduced into an existing Delaunay triangulation, which is broken and then reconnected to form a new Delaunay triangulation.

In general, the algorithm follows the same steps as in the two-dimensional construction described above. It starts with an initial Delaunay triangulation formed by a supertetrahedron or supercube, partitioned into five tetrahedrons which contain all other points. The remaining points which comprise the mesh to be triangulated are introduced one at a time, and the Bowyer–Watson algorithm is applied to create the Delaunay cavity and the corresponding retriangulation after each point insertion.

An important feature of a mesh generation procedure is its ability to produce a boundary-conforming mesh, i.e. the triangular faces of the assembly of tetrahedrons conform to the boundary surface. Unfortunately, the unconstrained technique does not guarantee that the boundary faces will be contained within such a triangulation. Thus, a modified procedure must be introduced to ensure that the resulting triangulation is boundary-conforming.

Constrained Triangulation

The purpose of the constrained Delaunay triangulation is to generate a triangulation which preserves the connections imposed on the boundary points. The three-dimensional constrained triangulation is carried out in the same way as for two-dimensional triangulations.

In the first approach, the tetrahedrons whose faces constitute the boundary surface are fixed during the process of retriangulation. These boundary tetrahedrons are generated in the first step of triangulation. The next steps include the insertion of a point, the definition of a star-shaped cavity containing the point, and retriangulation of the cavity. The resulting grid is boundary-conforming and its interior subtriangulation is a Delaunay triangulation.

The second approach to the constrained triangulation of a domain developed by Weatherill and Hassan (1994) starts with inputting the boundary points and boundary point connectivities of the faces of the boundary triangulation. After performing a Delaunay triangulation of the boundary points, a new Delaunay triangulation is built by inserting interior points and applying the Bowyer–Watson algorithm. After this, the tetrahedrons intersecting the boundary surface are transformed to recover the boundary triangulation. If a boundary face is not present in the new Delaunay triangulation, this is due to the fact that edges and faces of the tetrahedrons of the Delaunay triangulation intersect this face. Since the face is formed from three edges, it is necessary first to recover the face edges and then the face. This is achieved by first finding the tetrahedrons which are intersected by the face edges. There is a fixed combination of possible standard intersections of each tetrahedron by any mixed boundary edge, which allows one to perform direct transformations to recover the edge. Having established the intersection types, these tetrahedrons are then locally transformed into new tetrahedrons so that the required edges are present. A similar procedure then follows to recover the boundary faces.

12.3 Advancing-Front Methods

Advancing-front techniques extend the grid into the region in the form of marching layers, starting from the boundary and proceeding until the whole region has been covered with grid cells. The region separating the part of the domain already meshed from those that are still unmeshed is referred to as a front. Advancing-front techniques need some initial triangulation of the boundaries of the geometry, and this triangulation forms the initial front. The name of this class of methods refers to a strategy that consists of creating the mesh sequentially, element by element, creating new points and connecting them with previously created elements, thus marching into as-yet-unmeshed space and sweeping a front across the domain. The marching process includes the construction of a new simplex, which is built by connecting either some appropriate points on the front or some inserted new point with the vertices of a suitable face on the front. The process stops when the front is empty, i.e. when the domain is entirely meshed.

The advancing-front approaches offer the advantages of high-quality point placement and integrity of the boundary. The efficiency of the grid-marching process largely depends on the arrangement of grid points in the front, especially at sharp corners. A new grid point is placed at a position which is determined so as to result in a simplex with prescribed optimal quality features. In some approaches, the grid points are positioned along a set of predetermined vectors. To ensure a good grid quality and to facilitate the advancing process, these vectors are commonly determined once at each layer mesh point by simply averaging the normal vectors of the faces sharing the point and then smoothing the vectors. Other approaches to selecting new points for moving the front use the insertion techniques applied in the Delaunay triangulations described above.

The fronts continue to advance until either

(1) opposite fronts approach to within a local cell size; or
(2) certain grid quality criteria are locally satisfied.

Grid quality measures which are to be observed in the process of grid generation by means of the advancing-front method include the cell spacing and sizes of angles. The desired mesh spacings and other gridding preferences in the region are commonly specified by calculations on a background grid.

12.3.1 Procedure of Advancing-Front Method

In order to generate cells with acceptable angles and lengths of edges by a marching process, the advancing-front concept inherently requires a preliminary specification of local grid spacing and directionality at every point of the computational mesh. The spacing is prescribed by defining three (two in two dimensions) orthogonal directions together with some length scale for each direction. The directions and length scales are commonly determined from background information, in particular, by carrying out computations on a coarse grid and interpolating the data.

The advancing-front procedure proceeds by first listing all faces which constitute the front and then selecting an appropriate face (edge in two dimensions) on the front. The operation of the selection is very important, since the quality of the final grid may by affected by the choice. According to a common rule, the face is selected where the grid spacing is required to be the smallest. A collection of vertices on the front which are appropriate for connection to the vertices of the selected face to form a tetrahedron (triangle in two dimensions) is searched. The collection may be formed by the vertices which lie inside a sphere centered at the barycenter of the face, with an appropriate radius based upon the height of a unit equilateral tetrahedron. A new point is also created which is consistent with the ideal position determined from the background information about grid spacing and directionality. The selected vertices and the new point are ordered according to their distance from the barycenter of the selected face. Each sequential tetrahedron formed by the face and the ordered points is then checked to find out whether it intersects any face in the front. The first point which satisfies the test and gives a tetrahedron of good quality is chosen as the fourth vertex for the new tetrahedron. The current triangle is then removed from the list of front faces, since it is now obscured by the new tetrahedron. This process continues until there are no more faces in the list of front faces.

In many cases, the use of the background mesh to define the local grid spacing can be replaced by sources in the form of points, lines, and surfaces.

One of the advantages of such a procedure is that all operations are performed locally, on neighboring faces only. Additionally, boundary integrity is observed, since the boundary triangulation constitutes the initial front.

The disadvantages of the advancing-front approach relate mainly to the phase in which a local direction and length scale are determined and to the checking phase for ensuring the acceptability of a new tetrahedron.

12.3.2 Strategies for Selecting Out-of-Front Vertices

One of the critical items of advancing-front methods is the placement of new points. Upon generating a new simplex, a point is placed at a position which is determined so as to result in the required shape and size of the new simplex. The parameters which define the desirable cell at each domain position are specified by a function which is determined a priori or found in the process of computation.

In one approach, the new point is placed along a line which is orthogonal to a chosen face on the front and passes through its circumcenter. This placement is aimed at the creation of a new simplex whose boundary contains the chosen face.

If the simplex generated with the new point results in a crossover with the front, it is discarded. Alternately, if the new point is located very close to a vertex on the front, it is replaced by this vertex in order to avoid the appearance of a cell with a very small edge at some later stage.

Another approach, generally applied in two dimensions, takes into account a vertex on the front and the angle at which the edges cross at this point. The point is created with the aim of making the angles in the new triangles as near to 60° as possible. In particular, a very large angle between the edges is bisected or even trisected. On the other hand, if the vertex has a small interior angle, the two adjacent vertices on the front are connected. This approach can be extended to three dimensions by analyzing a dihedral angle at the front.

12.3.3 Grid Adaptation

The frontal approach is well suited to generating adaptive grids near the boundary segments, where the grid cells are commonly required to be highly stretched.

Highly stretched grid cells begin forming individually from the boundary and march into the domain. However, unlike the conventional procedure in which cells are added in no systematic sequence, the construction of a stretched grid needs to be performed by advancing one layer of cells at a time, with the minimum congestion of the front and a uniform distribution of stretched cells. The new points are positioned along a set of predetermined vectors in accordance with the value of a stretching function. The criterion by which the points are evaluated has a significant impact on the grid quality and the marching process. Because of the requirement of a high aspect ratio of cells in the boundary layer, the conventional criteria based on the cell angles are not appropriate for building highly stretched cells.

In a criterion based on a spring analogy, the points forming a new layer are assumed to be connected to the end points of the face by tension springs. Among these points, the one with the smallest spring force is considered the most suitable for forming the new cell, and consequently for changing the front boundary. The spring concept allows one to indicate when an opposing front is very close to the new location, namely, when an existing point on the front has the smallest spring force. The adaptive advancing process terminates on a front face when the local grid characteristics on the front, influenced by the stretching function, no longer match those determined by the background grid in that location. When the proximity and/or grid quality criteria are satisfied on all faces of the front, the process switches from an advancing-layers method to the conventional advancing-front method to form regular isotropic cells in the rest of the domain.

12.3.4 Advancing-Front Delaunay Triangulation

A combination of the advancing-front approach and the Delaunay concept gives rise to the advancing-front Delaunay methods.

If the boundary of a domain is triangulated and a set of points to be triangulated is given in the interior of the domain, then the advancing-front Delaunay triangulation is carried out by forming the cells adjoining the front in accordance with the empty-circumcircle property.

The procedure for the triangulation can be outlined as follows. A face on the front is chosen, and a new simplex is tentatively built by joining the vertices of the face to an arbitrary point on the front, in the interior of the domain with regard to the front. If this simplex contains any points within its circumcircle, it is not added to the triangulation. By checking all points, the appropriate vertex which produces a simplex containing no points interior to its circumcircle is eventually found. The simplex formed through this vertex is accepted and the front is advanced.

Another algorithm is based on the strategy of placing new points ahead of the front and triangulating them according to the Delaunay criterion.

12.4 Meshing by Quadtree-Octree Decomposition

Quadtree-octree meshing is based on the idea of partitioning a domain in a progressive manner so as to produce cells of size compatible with the node spacing requirement. The use of quadtree decomposition for two-dimensional mesh generation was developed in the 1980s by Shephard et al. (1988).

In this approach, applied to mesh generation, the n-dimensional domain to be gridded is first enclosed in a bounding root box (an n-dimensional parallelepiped) which is approximated with a union of disjoint and variably sized cells whose union constitutes the final mesh of the domain. The cells are obtained from a recursive refinement of the root parallelepiped. The current cell is subdivided into four equally sized cells in a two-dimensional case and into eight equally sized cells in a three-dimensional case. The stopping criterion used to subdivide a cell can be based on the local geometric properties of the boundary of the domain (e.g. the local curvature of the boundary) or user defined level of refinement.

The set of cells composes the tree structure associated with the spatial decomposition. At each stage of the tree construction, each cell of the tree is analyzed and refined into 2^n (with n the space dimension) equally sized cells, based on a specific criterion. The level of a cell corresponds to its depth in the related tree (i.e. the number of subdivisions required to reach a cell of this size). The bounding box is at level 0. The depth of the tree corresponds to the maximum level of subdivision.

12.5 Three-Dimensional Prismatic Grid Generation

The use of prismatic cells is justified by the fact that the requirement of high aspect ratio can be achieved without reducing the values of the angles between the cell edges.

The procedure for generating a prismatic grid begins by triangulating the boundary surface of a domain. The next stage in the procedure computes a quasinormal direction at each node of the surface triangulation. Then, the initial surface is shifted along these quasinormal directions by a specified distance d. This gives the first layer of prismatic cells. This shifting process is repeated a number of times using suitable values of d at each stage and either the same or newly computed normal directions. The value of the quantity d can be chosen in the form of any of the stretching functions described in Chap. 4.

The efficiency of the algorithm is essentially dependent on the choice of quasinormal directions. The generation of the quasinormals is carried out in three stages, depending on a position of the vertices.

(1) Normals are first computed at the vertices which lie on the corners of the boundary. These are calculated as the angle-weighted average of the adjacent surface normals. The angle used is the one between the two edges adjacent to the boundary surface and meeting at the corner.
(2) Normals at grid points on the geometrical edges of the boundary surface are computed. These normals are the average of the two adjacent surface normals.
(3) Finally, the normals at grid nodes on the boundary surfaces are calculated.

12.6 Comments

Unstructured grid methods were originally developed in solid mechanics. The paper by Field (1995) reviews some early techniques for unstructured mesh generation that rely on solid modelling. An informal survey that illustrates the wide range of unstructured mesh generation was conducted by Owen (1998) and described in the handbook of grid generation edited by Thompson et al. (1999).

Though unstructured technology deals chiefly with tetrahedral (triangular in two dimensions) elements, some approaches rely on hexahedrons (or quadrilaterals) for the decomposition of arbitrary domains. Recent results have been presented by Tam and Armstrong (1991) and Blacker and Stephenson (1991).

Properties of n-dimensional triangulations were reviewed by Lawson (1986). The relations between the numbers of faces were proved in the monograph by Henle (1979) and in the papers by Steinitz (1922), Klee (1964), Lee (1976).

The Delaunay triangulation and Voronoi diagram were originally formulated in the papers of Delaunay (1934, 1947), Voronoi (1908), respectively. Algorithms for computing Voronoi diagrams have been developed by Green and Sibson (1978), Brostow et al. (1978), Finney (1979), Bowyer (1981), Watson (1981), Tanemura et al. (1983), Sloan and Houlsby (1984), Fortune (1985) and Zhou et al. (1990). Results of studies of geometrical aspects of Delaunay triangulations and their dual Voronoi diagrams were presented in the monographs by Edelsbrunner (1987), Du and Hwang (1992), Okabe et al. (1992), Preparata and Shamos (1985). Proofs of the properties of planar Delaunay triangulations were given by Guibas and Stolfi (1985) and by Baker (1987, 1989).

A technique for creating the Delaunay triangulation of an a priori given set of points was proposed by Tanemura et al. (1983). The incremental two-dimensional Delaunay triangulation which starts with an initial triangulation was developed by Bowyer (1981), Watson (1981). Watson has also shown the visibility of the edges of the cavity associated with the inserted point. Having demonstrated that the Delaunay criterion is equivalent to the equiangular property, Sibson (1978) devised and later Lee and Schachter (1980) investigated a diagonal-swapping algorithm for generating a Delaunay triangulation by using the equiangular property.

A novel approach, based on the aspect ratio and cell area of the current triangles, to the generation of points as the Delaunay triangulation proceeds was developed by Holmes and Snyder (1988). In their approach, a new point is introduced into the existing triangulation at the Voronoi vertex corresponding to the worst triangle. Ruppert (1992), Chew (1993) have shown that in the planar case, the procedure leads to a Delaunay triangulation with a minimum-angle bound of 30 degrees. An alternative procedure of inserting the new point on a Voronoi segment was proposed by Rebay (1993). A modification of the Rebay technique was made by Baker (1994). Haman et al. (1994) inserted points into a starting Delaunay grid in accordance with the boundary curvature and distance from the boundary, while Anderson (1994) added nodes while taking into account cell aspect ratio and proximity to boundary surfaces.

Approaches to the generation of boundary-conforming triangulations based upon the Delaunay criterion have been proposed by Lee (1978), Lee and Lin (1986), Baker (1989), Chew (1989), Cline and Renka (1990), George et al. (1990), Weatherill (1990), George and Hermeline (1992), Field and Nehl (1992), Hazlewood (1993), Weatherill and Hassan (1994). All techniques and methods considered in the present chapter for proving the results associated with the constrained Delaunay triangulation were described on the basis of papers by Weatherill (1988), Baker (1989, 1994), Mavriplis (1990), Rebay (1993), Weatherill and Hassan (1994).

Further development of unstructured grid techniques based on the Delaunay criterion and aimed at the solution of three-dimensional problems has been performed by Cavendish et al. (1985), Shenton and Cendes (1985), Perronet (1988), Baker (1987, 1989), Jameson et al. (1986), Weatherill (1988), Frey and George (2008), Lo (2015). The application of the Delaunay triangulation for the purpose of surface interpolation was discussed by DeFloriani (1987).

The octree approach originated from the pioneering work of Yerry and Shephard (1985). The octree data structure has been adapted by Lohner (1988b) to produce efficient search procedures for the generation of unstructured grids by the moving-front technique. Octree-generated cells were used by Shephard et al. (1988); Yerry and Shephard (1990) to cover the domain and the surrounding space, and then to derive a tetrahedral grid by cutting the cubes. The generation of hexahedral unstructured grids was developed by Schneiders and Bunten (1995).

The moving-front technique has been successfully developed in three dimensions by Peraire et al. (1987), Lohner (1988a) and Formaggia (1991). Some methods using Delaunay connectivity in the frontal approach have been created by Merriam (1991), Mavriplis (1991, 1993), Rebay (1993), Muller et al. (1993), Marcum and Weatherill (1995).

Advancing-front grids with layers of prismatic and tetrahedral cells were formulated by Lohner (1993). A more sophisticated procedure, basically using bands of prismatic cells and a spring analogy to stop the advancement of approaching layers, was described by Pirzadeh (1992). The application of adaptive prismatic meshes to the numerical solution of viscous flows was demonstrated by Parthasarathy and Kallinderis (1995).

Some procedures for surface triangulations have been developed by Peraire et al. (1988), Lohner and Parikh (1988), Weatherill et al. (1993).

A survey of adaptive mesh refinement techniques was published by Powell et al. (1992). The combination of the Delaunay triangulation with adaptation was performed by Holmes and Lamson (1986), Mavriplis (1990), Muller (1994). The implementation of solution adaptation into the advancing-front method with directional refinement and regeneration of the original mesh was studied by Peraire et al. (1987). Approaches based on the use of sources to specify the local point spacing have been developed by Pirzadeh (1993), Pirzadeh (1994), Weatherill et al. (1993).

The prospects and trends for unstructured grid generation in its application to computational fluid dynamics were discussed by Baker (1995), Venkatakrishan (1996). The first application of the Delaunay triangulation in computational fluid dynamics was carried out by Bowyer (1981), Baker (1987). The advancing-front technique was introduced, in computational fluid dynamics, primarily by Peraire et al. (1987), Lohner (1988a), Lohner and Parikh (1988). The techniques of George (1971); Wordenweber (1981), Wordenweber (1983), Lo (1985), Peraire (1986) foreshadowed the more recent advancing-front methods. Muller (1994), Marchant and Weatherill (1994) applied a combination of frontal and Delaunay approaches to treat problems with boundary layers. Muller (1994) generated triangular grids in the boundary layer through a frontal technique, with high-aspect-ratio triangles, and filled the remainder of the domain with triangles built through the Delaunay approach. Another way to treat a boundary layer with the advancing-front approach was applied by Hassan et al. (1994). In the first step, the boundary layer is covered by a single layer of tetrahedral cells. Then, the newly generated nodes are moved along the cell edges towards the boundary by a specified distance. These steps, in the original layer, are repeated until a required resolution has been reached. After this, the advancing front proceeds to fill up the remainder of the domain.

An algorithm for the generation of a high-quality well-graded quadrilateral element mesh from a triangular element mesh was presented by Lee and Lo (1994), Lo (2015). Very important applications to parallel unstructured mesh generation were discussed by Chrisochoides (2006) and Ivanov (2008).

References

Anderson, W. K. (1994). A grid generation and flow solution method for the Euler equations in unstructured grids. *Journal of Computational Physics*, *110*, 23–38.

Baker, T. J. (1987). Three-dimensional mesh generation by triangulation of arbitrary points sets. AIAA Paper 87-1124-CP.

Baker, T. J. (1995). Prospects and expectations for unstructured methods. In *Proceedings of the Surface Modeling, Grid Generation and Related Issues in Computational Fluid Dynamics Workshop.* NASA Conference Publication 3291. (pp. 273–287). Cleveland, OH: NASA Lewis Research Center.

Baker, T. J. (1989). Automatic mesh generation for complex three-dimensional region using a constrained Delaunay triangulation. *Engineering with Computers*, *5*, 161–175.

Baker, T. J. (1994). Triangulations, mesh generation and point placement strategies. In D. Caughey (Ed.), *Computing the Future* (pp. 1–15). New York: Wiley.

Blacker, T. D., & Stephenson, M. B. (1991). Paving a new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, *32*, 811–847.

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

Brostow, W., Dussault, J. P., & Fox, B. L. (1978). Construction of Voronoi polyhedra. *Journal of Computational Physics*, *29*, 81–92.

Carey, G. F. (1997). *Computational grids: Generation, adaptation, and solution strategies*. London: Taylor and Francis.

Cavendish, J. C., Field, D. A., & Frey, W. H. (1985). An approach to automatic three-dimensional finite element mesh generation. *International Journal for Numerical Methods in Engineering*, *21*, 329–347.

Chew, P. (1993). Mesh generation, curved surfaces and guaranteed quality triangles. Technical report, IMA, Workshop on Modeling, Mesh Generation and Adaptive Numerical Methods for Partial Differential Equations, University of Minnesota, Minneapolis.

Chew, L. P. (1989). Constrained Delaunay triangulations. *Algorithmica*, *4*, 97–108.

Chrisochoides, N. (2006). *Chapter 7. Parallel Mesh Generation.* In *Numeriacal Solution of PDEs on Parallel Computers. Lecture Notes in Computational Science and Engineering* (vol. 51, pp. 237–264).

Cline, A. K., & Renka, R. L. (1990). A constrained two-dimensional triangulation and the solution of closest node problems in the presence of barriers. *SIAM Journal on Numerical Analysis*, *27*, 1305–1321.

DeFloriani, L. (1987). Surface representations on triangular grids. *The Visual Computer*, *3*, 27–50.

Delaunay, B. (1947). *Petersburg School of Number Theory*. USSR, Moscow (Russian): Ak. Sci.

Delaunay, B. (1934). Sur la sphere vide. *Bull. Acad Sci. USSR VII: Class Sci. Mat. Nat.*, *6*, 793–800.

Dirichlet, G. L. (1850). Uber die Reduction der positiven quadratischen Formen mit drei underbestimmten ganzen Zahlen. *Z. Reine Angew. Math.*, *40*(3), 209–227.

Edelsbrunner, H. (1987). *Algorithms in combinatorial geometry*. Berlin: Springer.

Du, D.-Z., & Hwang, F. (Eds.). (1992). *Computing in euclidean geometry*. Singapore: World Scientific.

Field, D. A., Nehl, T. W. (1992). Stitching together tetrahedral meshes. In Field, D., Komkov, V. (Eds.), *Geometric Aspects of Industrial Design.* Philadelphia: SIAM, Chap. 3, 25–38

Field, D. A. (1995). The legacy of automatic mesh generation from solid modeling. *Computer Aided Geometric Design*, *12*, 651–673.

Finney, J. L. (1979). A procedure for the construction of Voronoi polyhedra. *Journal of Computational Physics*, *32*, 137–143.

Formaggia, L. (1991). An unstructured mesh generation algorithm for three-dimensional aeronautical configurations. In A. S. Arcilla, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields* (pp. 249–260). New York: North-Holland.

Fortune, S. (1985). *A sweepline algorithm for Voronoi diagrams*. Murray Hill, NJ: AT&T Bell Laboratory Report.

Frey, P. J., & George, P.-L. (2008). *Mesh generation application to finite elements*. ISTE Ltd and Wiley Inc.

George, A. J. (1971). Computer Implementation of the Finite Element Method. Stanford University Department of Computer Science, STAN-CS-71-208.

George, P. L., Hecht, F., & Saltel, E. (1990). Automatic 3d mesh generation with prescribed meshed boundaries. *IEEE Transactions on Magnetics*, *26*(2), 771–774.

George, P. L., & Borouchaki, H. (1998a). *Delaunay triangulation and meshing: application to finite elements*. Paris: Editions Hermes.

George, P. L., & Borouchaki, H. (1998b). *Delaunay triangulation and meshing*. Paris: Editions Hermes.

George, P. L., & Hermeline, F. (1992). Delaunay's mesh of convex polyhedron in dimension *d*: application for arbitrary polyhedra. *International Journal for Numerical Methods in Engineering*, *33*, 975–995.

Green, P. J., & Sibson, R. (1978). Computing Dirichlet tessellations in the plane. *The Computer Journal*, *21*(2), 168–173.

Guibas, L., & Stolfi, J. (1985). Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, *4*, 74–123.

Haman, B., Chen, J.-L., & Hong, G. (1994). Automatic generation of unstructured volume grids inside or outside closed surfaces. In N. P. Weatherill, P. R. Eiseman, J. Hauser, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Field Simulation and Related Fields* (p. 187). Swansea: Pineridge.

Hassan, O., Probert, E. J., Morgan, K., & Peraire, J. (1994). Unstructured mesh generation for viscous high speed flows. In N. P. Weatherill, P. R. Eiseman, J. Hauser, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Field Simulation and Related Fields* (p. 779). Swansea p: Pineridge.

Hazlewood, C. (1993). Approximating constrained tetrahedrizations. *Computer Aided Geometric Design*, *10*, 67–87.

Henle, M. (1979). *A combinatorial introduction to topology*. San Francisco: W.H Freeman.

Holmes, D. G., & Lamson, S. H. (1986). Adaptive triangular meshes for compressible flow solutions. In J. Hauser & C. Taylor (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics* (p. 413). Swansea: Pineridge.

Holmes, D. G., & Snyder, D. D. (1988). The generation of unstructured triangular meshes using Delaunay triangulation. In S. Sengupta, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics* (pp. 643–652). Swansea: Pineridge.

Ivanov, E. (2008). *Parallel Tetrahedral meshing based on a-priori domain decomposition: from scratch to results by utilizing off-the-shelf sequential software*. Saarbrucken: VDM Verlag Dr. Muller.

Jameson, A., Baker, T. J., Weatherill, N. P. (1986). Calculation of inviscid transonic flow over a complete aircraft. AIAA Paper 86-0103.

Klee, V. (1964). The number of vertices of a convex polytope. *Can. Math.*, *16*, 37.

Lawson, C. L. (1986). Properties of *n*-dimensional triangulations. *Computer Aided Geometric Design*, *3*, 231–246.

Lee, K. D. (1976). On finding *k*-nearest neighbours in the plane. Technical Report 76–2216. University of Illinois, Urbana, IL.

Lee, D. T. (1978). Proximity and reachibility in the plane. Techical Report R-831, University of Illinois, Urbana, IL.

Lee, D. T., & Lin, A. K. (1986). Generalized Delaunay triangulation for planar graphs. *Discrete & Computational Geometry*, *1*, 201–217.

Lee, C. K., & Lo, S. H. (1994). A New Scheme for the Generation of a Graded Quadrilateral Mesh. *Computers and Structures.*, *52*, 847–857.

Lee, D. T., & Schachter, B. J. (1980). Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Science*, *9*(3), 219–241.

Lo, S. H. (1985). A new mesh generation scheme for arbitrary planar domains. *International Journal for Numerical Methods in Engineering*, *21*, 1403–1426.
Lo, S. H. (2015). *Finite element mesh generation*. Boca Raton: CRC Press Taylor and Francis Group.
Lohner, R. (1988a). Generation of three-dimensional unstructured grids by the advancing-front method. AIAA Paper 88-0515.
Lohner, R. (1993). Matching semi-structured and unstructured grids for Navier–Stokes calculations. AIAA Paper 933348-CP.
Lohner, R. (1988b). Some useful data structures for the generation of unstructured grids. *Communications in Applied Numerical Methods*, *4*, 123–135.
Lohner, R., & Parikh, P. (1988). 3-dimensional grid generation by the advancing front method. *Intertnational Journal for Numerical Methods in Fluids*, *8*, 1135–1149.
Marchant, M. J., & Weatherill, N. P. (1994). Unstructured grid generation for viscous flow simulations. In N. P. Weatherill, P. R. Eiseman, J. Hauser, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Field Simulations and Related Fields* (p. 151). Swansea, UK: Pineridge.
Marcum, D. L., & Weatherill, N. P. (1995). Unstructured grid generation using iterative point insertion and local reconnection. *AIAA Journal*, *33*(9), 1619–1625.
Mavriplis, D. J. (1993). An advancing front Delaunay triangulation algorithm designed for robustness. AIAA Paper 93-0671.
Mavriplis, D. J. (1990). Adaptive mesh generation for viscous flows using Delaunay triangulation. *Journal of Computational Physics*, *90*, 271–291.
Mavriplis, D. J. (1991). Unstructured and adaptive mesh generation for high Reynolds number viscous flows. In A. S. Arcilla, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields* (pp. 79–92). Amsterdam: North-Holland.
Merriam, M. (1991). An efficient advancing front algorithm for Delaunay triangulation. Technical report, AIAA Paper 91-0792.
Muller, J. D., Roe, P. L., & Deconinck, H. (1993). A frontal approach for internal node generation in Delaunay triangulations. *International Journal for Numerical Methods in Fluids*, *17*(3), 241–256.
Muller, J. D. (1994). Quality estimates and stretched meshes based on Delaunay triangulation. *AIAA Journal*, *32*, 2372–2379.
Okabe, A., Boots, B., & Sugihara, K. (1992). *Spatial tessellations concepts and applications of voronoi diagrams*. New York: Wiley.
Owen, S. (1998). A survey of unstructured mesh generation technology. In 7th International Roundtable
Parthasarathy, V., & Kallinderis, Y. (1995). Directional viscous multigrid using adaptive prismatic meshes. *AIAA Journal*, *33*(1), 69–78.
Peraire, J. (1986). *A Finite Element Method for Convection Dominated Flows*. Ph.D. dissertation, University of Wales.
Peraire, J., Peiro, J., Formaggia, L., Morgan, K., Zienkiewicz, O. C. (1988). Finite element Euler computations in three dimensions. AIAA Paper 88-0032.
Peraire, J., Vahdati, M., Morgan, H., & Zienkiewicz, O. C. (1987). Adaptive remeshing for compressible flow computations. *Journal of Computational Physics*, *72*, 449–466.
Perronet, A. (1988). A generator of tetrahedral finite elements for multimaterial objects or fluids. In S. Sengupta, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical Grid Generation in Computational Fluid Mechanics* (pp. 719–728). Swansea: Pineridge.
Pirzadeh, S. (1992). Recent progress in unstructured grid generation. AIAA Paper 92-0445.
Pirzadeh, S. (1994). Viscous unstructured three-dimensional grids by the advancing-layers method. AIAA Paper 94-0417.
Pirzadeh, S. (1993). Structured background grids for generation of unstructured grids by advancing front method. *AIAA Journal*, *31*(2), 257–265.
Powell, K. G., Roe, P. L., & Quirk, J. J. (1992). Adaptive-mesh algorithms for computational fluid dynamics. In M. Y. Hussaini, A. Kumar, & M. D. Salas (Eds.), *Algorithmic Trends in Computational Fluid Dynamics* (pp. 301–337). New York: Springer.

Preparata, F. P., & Shamos, M. I. (1985). *Computational geometry: an introduction*. New York: Springer.

Rebay, S. (1993). Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer–Watson algorithm. *Journal of Computational Physics*, *106*, 125–138.

Ruppert, J. (1992). *Results on Triangulation and High Quality Mesh Generation.* Ph.D. thesis, University of California, Berkeley.

Schneiders, R., & Bunten, R. (1995). Automatic generation of hexahedral finite element meshes. *Computers Aided Geometry Design*, *12*(7), 693.

Shenton, D. N., Cendes, Z. J. (1985). Three-dimensional finite element mesh generation using Delaunay tessellation. IEEE Transactions on Magnetics **MAG-21**, 2535–2538.

Shephard, M. S., Guerinoni, F., Flaherty, J. E., Ludwig, R. A., & Bachmann, P. L. (1988). Finite octree mesh generation for automated adaptive three-dimensional flow analysis. In S. Sengupta, J. Hauser, P. R. Eiseman, & J. F. Thompson (Eds.), *Numerical grid generation in computational fluid mechanics* (pp. 709–718). Swansea: Pineridge.

Sibson, R. (1978). Locally equiangular triangulations. *The Computer Journal*, *21*(3), 243–245.

Sloan, S. W., & Houlsby, G. T. (1984). An implementation of Watson's algorithm for computing 2D Delaunay triangulations. *Advances in Engineering Software*, *6*(4), 192–197.

Steinitz, E. (1922). Polyeder and Raumeintailungen. *Enzykl. Mathematischen Wiss.*, *3*, 163.

Tam, T. K. H., & Armstrong, C. G. (1991). 2D finite element mesh generation by medial axis subdivision. *Advance in Engineering Software and Workstations*, *13*, 313–344.

Tanemura, M., Ogawa, T., & Ogita, N. (1983). A new algorithm for three-dimensional Voronoi tessellation. *Journal of Computational Physics*, *51*, 191–207.

Thompson, J. F., Soni, B. K., & Weatherill, N. P. (Eds.). (1999). *Handbook of grid generation*. Boca Raton: CRC Press.

Venkatakrishan, V. (1996). Perspective on unstructured grid flow solvers. *AIAA Journal*, *34*(3), 533–547.

Voronoi, G. F. (1908). Nouvelles applications des parameters countinus a la theorie des formes quadratiques. Deuxieme Memoire: Recherches sur la parallelloedres primitifs. *J. Reine Angew. Math.*, *134*, 198–287.

Watson, D. (1981). Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal*, *24*(2), 167–172.

Weatherill, N. P., Marchant, M. F., Hassan, O., Marcum, D. L (1993). Adaptive inviscid flow solutions for aerospace geometries on efficiently generated unstructured tetrahedral meshes. AIAA Paper 93-3390

Weatherill, N. P. (1988). A method for generating irregular computational grids in multiply connected planar domains. *International Journal for Numerical Methods Fluids*, *8*, 181–197.

Weatherill, N. P. (1990). The integrity of geometrical boundaries in the two-dimensional Delaunay triangulation. *Communications in Applied Numerical Methods*, *6*, 101–109.

Weatherill, N. P., & Hassan, O. (1994). Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods Engineering*, *37*, 2005–2039.

Wordenweber, B. (1981). Automatic mesh generation of 2- and 3-dimensional curvilinear manifolds. Ph.D. Dissertation (available as Computer Laboratory Report N18), University of Cambridge).

Wordenweber, B. (1983). Finite-element analysis from geometric models. *The International Journal for Computational and Mathematics in Electrical and Electronics Engineering*, *2*, 23–33.

Yerry, M. A., & Shephard, M. S. (1985). Automatic three-dimensional mesh generation for three-dimensional solids. *Computers & Structures*, *20*, 31–39.

Yerry, M. A., & Shephard, M. S. (1990). Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering*, *20*, 1965–1990.

Zhou, J. M., Ke-Ran, S., Ke-Ding, Z., & Quing-Hua, Z. (1990). Computing constrained triangulation and Delaunay triangulation: a new algorithm. *IEEE Transactions on Magnetics*, *26*(2), 692–694.