A generic Delaunay triangulation algorithm for finite element meshes

D.A. Field

Mathematic Department, General Motors Research Laboratories, 30500 Mound Road, Warren, MI 48090-9055 USA

ABSTRACT

A mathematical framework for the construction of k dimensional meshes embedded in n dimensions is provided. By extending Delaunay triangulations a generic triangulation algorithm is stated along with methods for assessing the shapes of simplices. Planar, surface and solid meshes illustrate the algorithm.

1. INTRODUCTION

Although Boris Delaunay's initial paper on triangulations [6] dealt with n dimensions, most recent papers on Delaunay triangulations have focused on the efficiency of algorithms for triangulating two and three dimensions, see [21] for a general reference and [5,26] for more recent and extensive bibliographies. A new direction of interest in Delaunay triangulations is emerging with algorithms for constructing triangulations on spheres and on more general two dimensional surfaces [2,11,22-24,28]. This paper presents a mathematical framework for all types of Delaunay triangulations and their applications. By first constructing its dual Dirichlet tesselation (Voronoi diagram) [3] a Delaunay triangulation can then be constructed in $O(n)$ additional steps. Delaunay triangulations can also be constructed directly [31]. The unifying structure presented in this paper is oriented toward the latter method of constructing Delaunay triangulations. This orientation is natural because tesselations are more closely related to nearest neighbors [29] while the focus of this paper is on the triangulation of k dimensional manifolds embedded in n dimensions, $k < n$.

The mathematical properties of Delaunay triangulations have been well documented by Preparata and Shamos [21] and Rogers [25]. The book by Preparata and Shamos gives a general and algorithmic approach whereas Rogers' book is more theoretical. The motivation for creating the mathematical framework presented here is a natural extension of the applications of Delaunay triangulations [4,11,15] which are reviewed in the next section of this paper. This extension is very different from the generalizations found in [21] and for $k, n \in \{2, 3\}$ is related to interpolating scattered data for which there exists an extensive literature, see [1,13,14] for additional references.

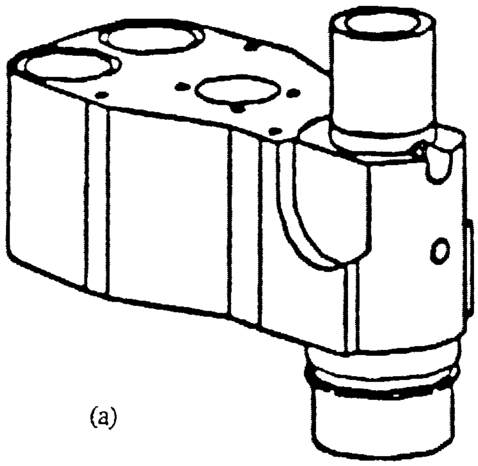
The third section will supply a brief yet complete statement of the terminology and definitions used in this paper. An algorithm for constructing k dimensional Delaunay triangulations embedded in n dimensions will be given in the fourth section. The fifth and final section will supply equations and formulae for constructing the triangulation, for assessing the shape of its simplices, and for general properties of Delaunay triangulations.

2. INITIAL APPLICATIONS

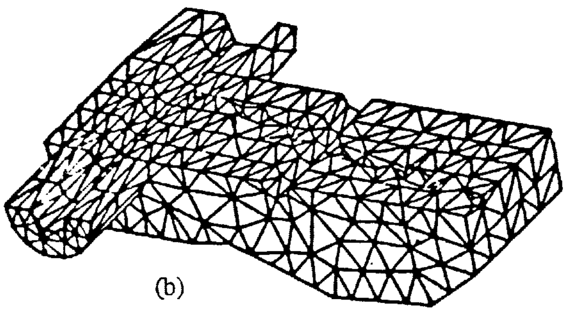
Motivation for the generic Delaunay triangulation algorithm came from applying two and three dimensional Delaunay triangulations to finite element mesh generation [4,15] and to triangulations of surfaces [11]. Since the applications reported in these references are within the framework developed in the next two sections and are contributors to this development, brief abstracts of these applications will be given here.

Among these Delaunay triangulations the first application generated tetrahedral finite element meshes [4]. The aim was to decompose arbitrary solid models created from CAD systems, see Figure 2.1, but the main bottleneck was, and still is, the generation of finite element nodes. Numerical difficulties and the creation of undesirable elements were addressed in [8]. Although the algorithm is ideally suited for adaptive mesh refinement, further improvements with the interface between mesh generation and CAD were needed to make this application truly automatic.

The ability to selectively refine, derefine and smooth graded two dimensional Delaunay triangulations were the goals in [10,15]. In Figure 2.2 [15], a variant of the re-



(a)



(b)

Figure 2.1 (a) Display of CAD mathematical representation and (b) its tetrahedral finite element decomposition.

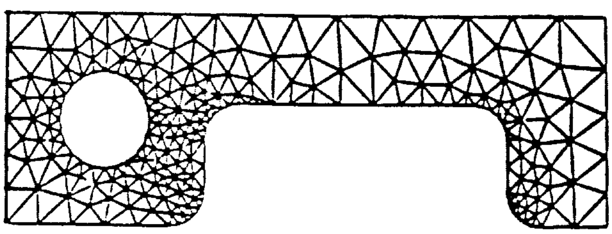
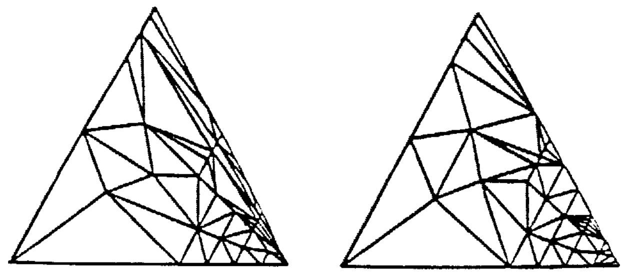


Figure 2.2 A graded mesh based on Delaunay triangulations.

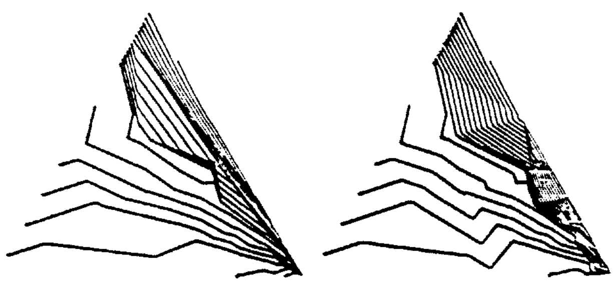
sults in [10, Lemma 6] was applied to guarantee that the boundary edges defining the finite element domain were maintained by the Delaunay triangulation. Improvements in maximizing minimum angles and allowing a fixed number of finite element nodes to be redistributed were given in [12].

An algorithm which eliminates the extra triangle surrounding all triangulation points was developed in [10] order to create 2/3-Delaunay triangulations; see the next section for the definition of k/n-Delaunay triangulations. A 2/3-Delaunay triangulation was then applied to lubrication data [27]. The dependency of the swell of elastomers on the percentages of water, methanol and oil allowed these four variables to be transformed into a representation $z = f(x, y)$ where the domain of f is the equilateral triangle of barycentric coordinates. The projection of the 2/3-Delaunay triangulation onto the domain in Figure 2.3(a) is characterized by the presence of long narrow triangles and is contrasted with the corresponding planar Delaunay triangulation in Figure 5.3(b). This projection phenomenon has been encountered recently in other planar triangulations which seek to optimize other criteria, [7,20]. Contours of increasing z-values are displayed in Figure 2.4.



(a) (b)

Figure 2.3 (a) Projection of the 2/3-Delaunay triangulation into the xy-plane, and (b) the corresponding planar Delaunay triangulation.



(a) (b)

Figure 2.4 (a) Contours of the 2/3-Delaunay triangulation and (b) contours of the projection of a planar Delaunay triangulation.

2/3-and 3/4-delaunay triangulations are now being investigated for adaptive finite element mesh generation. An assumption is that creating initial meshes in two and three dimensions is a geometric problem, albeit an extremely difficult one in three dimensions. Refinement of meshes are then attempts to improve approximations of solutions which are two- and three- dimensional manifolds embedded in three and four dimensions respectively. Since the finite element method is a method for approximating the solution manifold, the aim is to create $(n-1)/n$-Delaunay triangulations of the solution manifold which will be projected onto their physical domains.

3. DELAUNAY TRIANGULATIONS

A *k-simplex*, S_k, in E^n is the convex hull of $k+1$ affinely independent n-dimensional vertices $\mathbf{P}_1, \ldots, \mathbf{P}_{k+1}$, $k < n$, where

$$S_k = \{a_1\mathbf{P}_1 + \ldots + a_{k+1}\mathbf{P}_{k+1} \mid a_1 + \ldots + a_{k+1} = 1, \quad 0 < a_i, \quad 1 < i < k+1\} \quad . \tag{3.1}$$

An *r-face* of a k-simplex is an r-simplex whose $r+1$ vertices are a subset of the $k+1$ vertices of the k-simplex.

A *k/n-triangulation* is a collection of k-simplices in n dimensions such that the intersection of two k-simplices is either empty or an r-face and that each k-simplex must share a $(k-1)$-face with at least one other k-simplex.

The *minsphere* of a k-simplex is the smallest sphere of dimension $n-1$ passing through its $k+1$ vertices. The interior of the minsphere is the n-dimensional open set whose boundary is the minsphere. Thus the minsphere of a triangle imbedded in three dimensions is a 2-sphere whose equator, a 1-sphere, is the circumcircle in the plane of the triangle. The center and the radius of a minsphere are respectively called its *mincenter* and *minradius*. A k-simplex in a k/n-triangulation satisfies the *empty minsphere property* and is a *Delaunay k-simplex* if the interior of its minsphere contains no vertex of any k-simplex. A *k/n-Delaunay triangulation* is a k/n-triangulation when every k-simplex is a Delaunay k-simplex.

If $k = n$ the minsphere of an n-simplex is called a *circumsphere* and its center and radius are called *circumcenter* and *circumradius* respectively. With the restriction that the n-simplices satisfy the same conditions as the k-simplices in a Delaunay k/n-triangulation, the union of all such n-dimensional Delaunay triangulations is the convex hull of the triangulationpoints. A *separation plane* of an n-dimensional Delaunay triangulation is the unique $n-1$-dimensional hyperplane passing through an $(n-1)$-simplex on the boundary of the convex hull. A separation plane divides n-dimensional space into two open half spaces where the intersection of the triangulation with one of the open half spaces is empty.

Separation planes generalize to k/n-Delaunay triangulations. Of the infinitely many $n-1$-dimensional planes passing through a $(k-1)$-simplex on the boundary of a k/n-Delaunay triangulation, there is a unique plane perpendicular to the line passing through distinct mincenters of the k-simplex and its $(k-1)$-face on the boundary. The special case when these mincenters coincide is resolved in section 5. This plane may not "separate" E^n into two open half spaces where one is disjoint from the triangulation.

Finally, note that a k-simplex sponsors spheres in many dimensions. Its minsphere is $n-1$-dimensional as are the minspheres of each of its r-faces, $0 < r < k$. However, the k-simplex has a $k-1$-dimensional circumsphere and each of its r-faces has an $r-1$-dimensional circumspheres.

4. GENERIC TRIANGULATION ALGORITHM

A crucial assumption in the following algorithm is that triangulation points are randomly distributed in the sense stated by Rogers [25]. The vertices in a k/n-Delaunay triangulation are random if no subset of $k+2$ points lie on the minsphere of a Delaunay k-simplex. This assumption avoids the possibility of creating degenerate k-simplices in k-dimensional Delaunay triangulations. Implementations of the algorithm for realistic applications must account for this type of nonrandomness which can produce degenerate or nearly degenerate simplices that must be detected and handled as special cases. See [4] for an example of postprocessing a special case. The following algorithm can triangulate closed manifolds.

GENERIC DELAUNAY TRIANGULATION ALGORITHM

INITIALIZATION: Create a k-simplex with the first $k+1$ affinely independent points. With this simplex initialize a *master list* of simplices and include the mincenters and minradii of each simplex. With the $(k-1)$-faces of the original k-simplex initialize a *boundary list* of $(k-1)$-simplices and include their separation planes.

CLASSIFICATION: The remaining points are inserted one at a time until all points have been inserted or all remaining points cannot be inserted without creating simplices which do not satisfy the empty minsphere property. The location of a new triangulation point falls into one of four cases wherein each case STEP(*) and insertion patch refer to the INSERTION section of the algorithm.

CASE 1: The new point is separated from the k/n-Delaunay triangulation by at least one separation plane (Figures 4.1 and 4.2). Execute STEPa and STEPb. If the insertion patch is not connected execute STEPc, otherwise execute STEPd. If STEPd is executed and if the candidate k-simplices are Delaunay k- simplices execute STEPe and STEPf, otherwise execute STEPc.

CASE 2: No separation plane contains or separates the new point and the insertion patch is not empty (Figure 4.3). Execute STEPa. If the insertion patch is not connected execute STEPc, otherwise execute STEPd. If STEPd is executed and if the candidate k-simplices are Delaunay k-simplices execute STEPe and STEPf, otherwise execute STEPc.

CASE 3: No separation plane contains or separates the new point and the insertion patch is empty (Figure 4.4). Determine the k-simplex or k-simplices of minimum distance to the new point and let these simplices be the insertion patch. If the insertion patch is not connected execute STEPc, otherwise execute STEPd. If STEPd is executed and if the candidate k-simplices are Delaunay k-simplices execute STEPe and STEPf, otherwise execute STEPc.

CASE 4: The new point lies on at least one separation plane and no separation plane separates the new point (Figure 4.5). Execute STEPa. From the boundary list determine the k-simplex whose $(k-1)$-face on the boundary is least distant to the new point and add to the insertion patch the k-simplices formed by connecting the new point to the other $(k-1)$-faces of the boundary k-simplex. If the insertion patch contains the boundary k-simplex delete this simplex from the patch. If the insertion patch is not connected or if there is no unique $(k-1)$-face of minimum distance, execute STEPc, otherwise execute STEPd. If STEPd is executed and the candidate simplices are Delaunay k-simplices execute STEPe and STEPf, otherwise execute STEPc.

INSERTION: This section of the algorithm lists the steps executed by the CLASSIFICATION section.

STEPa: Create an *insertion patch* as the union of k-simplices which contain the new point inside their minspheres.

STEPb: Add to the insertion patch new k-simplices created from the boundary list by connecting the new point to the vertices of a $(k-1)$-face whose separation plane separates the new point from the vertex opposite the $(k-1)$-face.

STEPc: Append the new point to the end of the list of triangulation points.

STEPd: Create candidate Delaunay k-simplices by connecting the new point to the insertion patch boundary faces which do not contain the new point.

STEPe: Replace the original k-simplices of the insertion patch with the new Delaunay k-simplices and update the master list of Delaunay k-simplices.

STEPf: Whenever a new Delaunay k-simplex is inserted into the master list and has a $(k-1)$-face on the boundary of the triangulation update the list of separation planes.

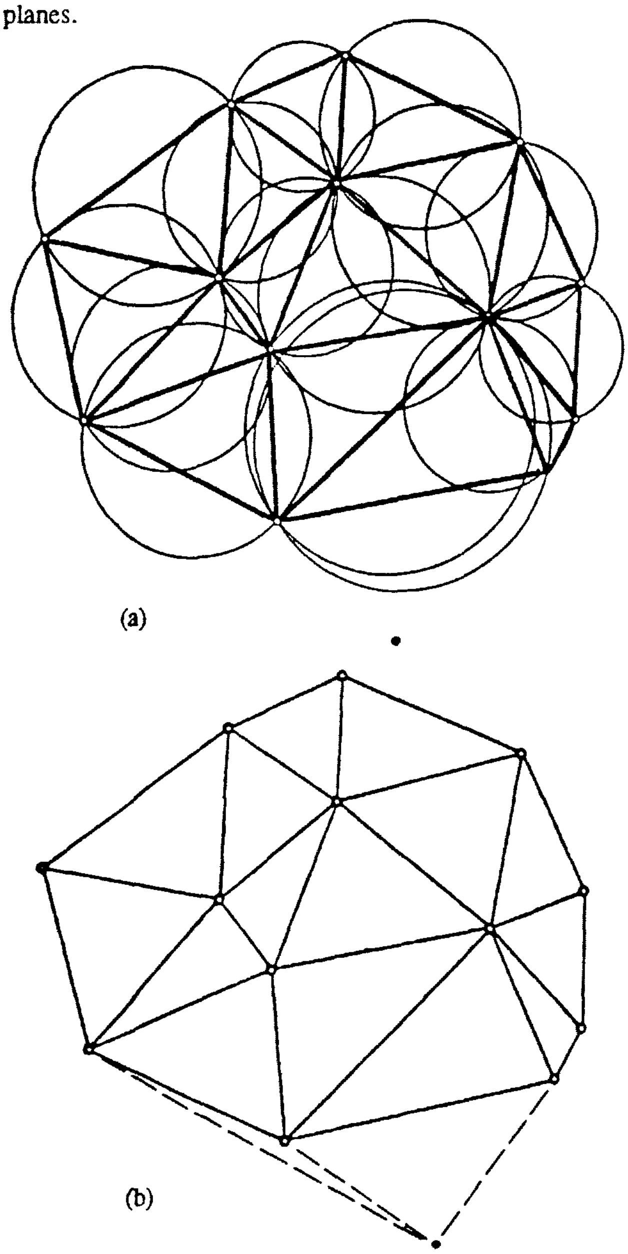


Figure 4.1 CASE 1: (a) the empty circumcircles and (b) the retriangulation.

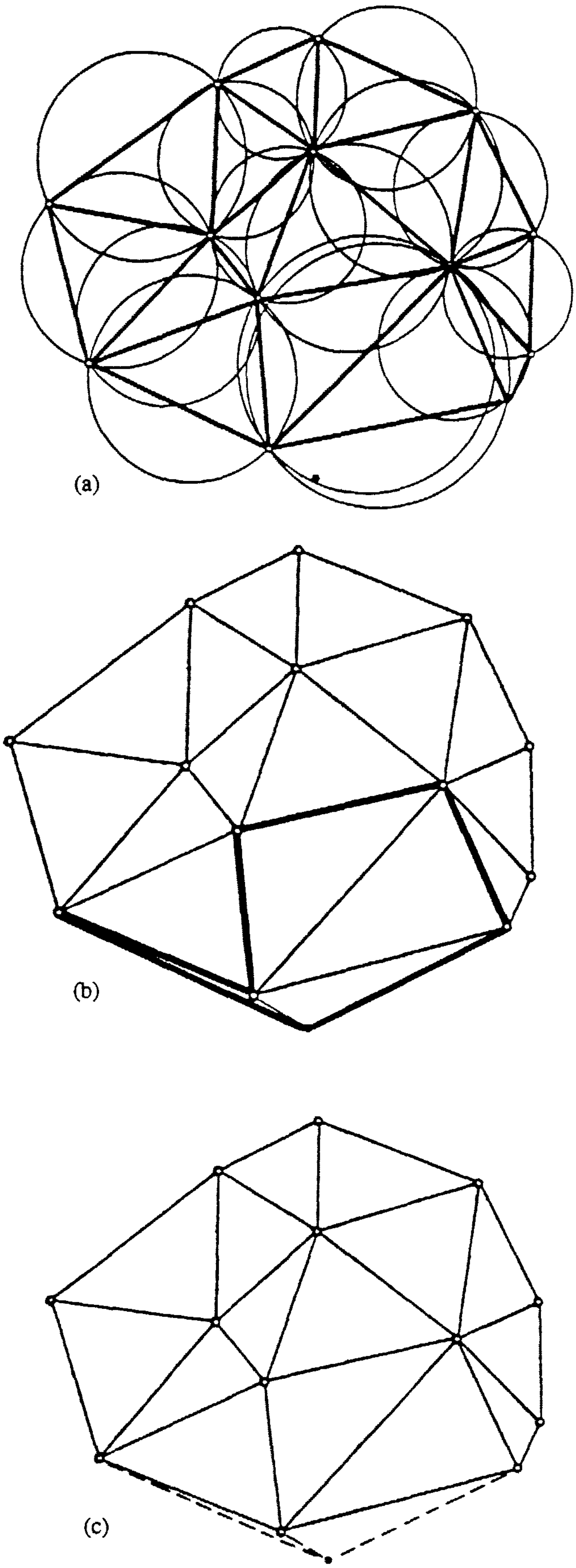


Figure 4.2 CASE 1: (a) the circumcircles (b) the insertion patch, and (c) the retriangulation of the insertion patch.

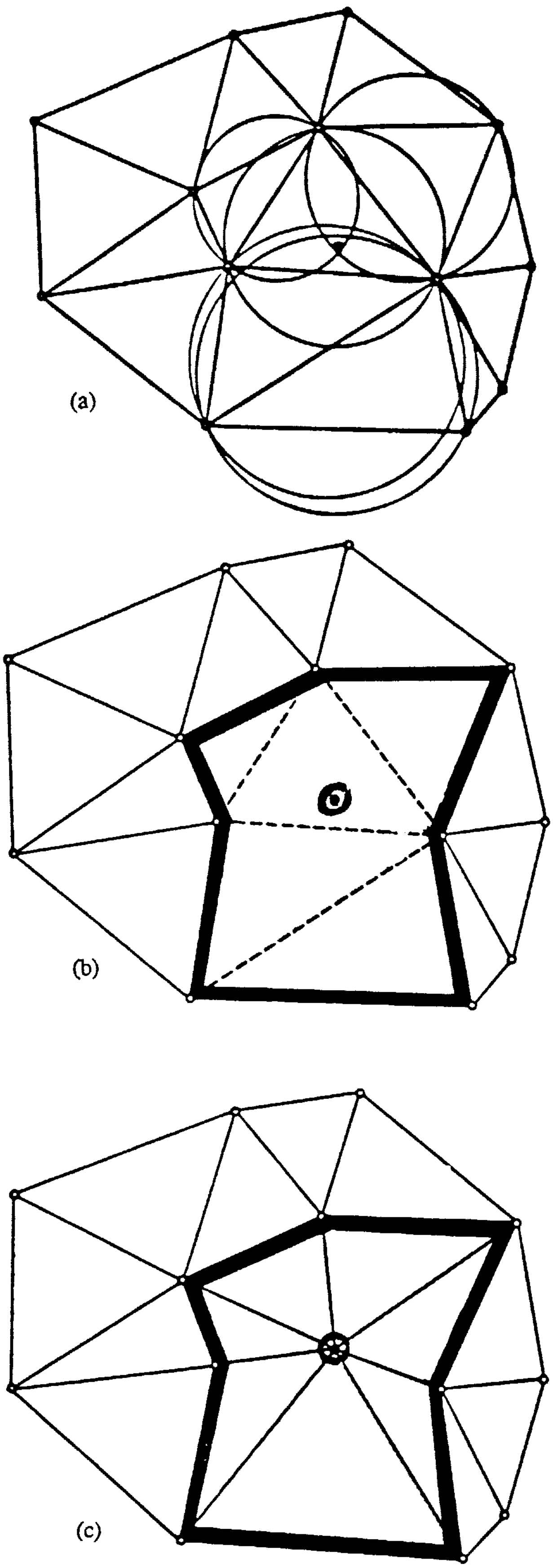


Figure 4.3 CASE 2: (a) the nonempty circumcircles (b) the insertion patch, and (c) the retriangulation of the insertion patch.

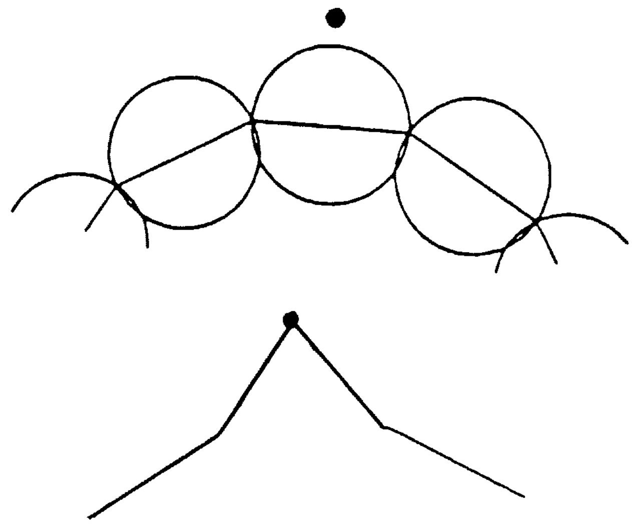


Figure 4.4 CASE 3: an example with $k = 1$ and $n = 2$.

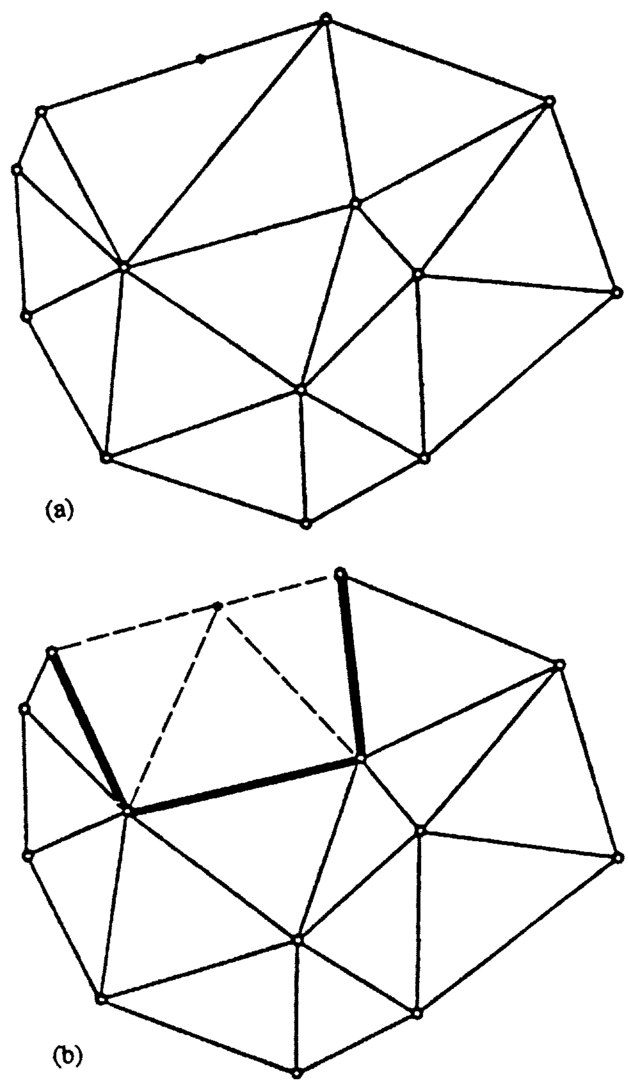


Figure 4.3 CASE 4: (a) the new point and (b) the retriangulation of the insertion patch.

Computational experience with $k\epsilon\{2,3\}$ and $n = 3$ has shown that whenever a new triangulation point is inserted on or nesrly on an existing minsphere it is best to postpone the insertion of the new point because in the meanwhile other points may be inserted and the local geometry of the triangulation may change. If the list on points yet to be triangulated becomes filled with points which lie on minspheres and these points must be inserted, then it is best to consider these points inside the minspheres. Postponing the insertion of new points is not necessary in two-dimensional Delaunay triangulations. On the other hand postponing insertion significantly increased robustness of the generic algorithm for three-dimensional and 2/3-dimensional Delaunay triangulations. Except for two-dimensional triangulations checks for degenerate or nearly degenerate simplices must be included in the triangulation algorithm. This is most effectively accomplished by using the normalized shape ratio found in the next section.

Lawson's edge swapping algorithm for planar Delaunay triangulations is particularly efficient when the triangle containing the new triangulation point is known. Unfortunately this algorithm has no analog for swapping edges in a tetrahedral triangulation. For this reason Lawson's method is not used in the Generic Delaunay Triangulation algorithm. In all dimensions however, Lawson's data structure can be very useful.

A set of N points in n dimensions, $N > n$, can be triangulated into k-simplices, $0 < k < d < n$, where the zero-dimensional triangulation is the set of N points and $d + 1$ is the maximum number of affinely independent points.

Constructions of n-dimensional Delaunay triangulations by the Dirichlet tesselation approach do not appear to extend to k/n-tesselations. There is however a natural way to define k/n-tesselations. The *k/n-Dirichlet tesselation*, dual to a k/n-Delaunay triangulation, is the intersection of the n-dimensional Dirichlet tesselation with the k/n-Delaunay triangulation. Note that the vertices on the n-dimensional Dirichlet tiles are circumcenters of n- simplices and they need not be located on the k/n-Dirichlet tesselation.

The number of triangles in a planar triangulation is given by $v_b + 2v_i + 2h - 2$ where v_b is the number of vertices on the boundary, v_i is the number of internal vertices and h is the number of holes in the triangulation. The number of k-simplices in a k/n-Delaunay triangulation as a function of the number of vertices v appears to behave as $O(k!v)$. In general the number of k-simplices generated by v vertices is not unique when $k > 2$.

In order to form insertion patches the generic triangulation algorithm specifies that all nonempty minspheres must be tested for containment of a new point. This testing is necessary when the algorithm constructs a k/n-Delaunay triangulation because the algorithm constructs a manifold which depends on the order of insertion and the density of trangulation points. If a known manifold supplies the triangulation points, the location of these points is another important factor which can influence the faithfulness of the approximation of the k/n-triangulation to the original manifold. In planar Delaunay triangulations this testing process leads to $O(v^2)$ arithmetic operations. The use of pointers to march through adjacent triangles can reduce the number of arithmetic operations to an expected $O(v^{3/2})$ [17]. Although this type of search would yield connected insertion patches in $2/n$-Delaunay triangulations and a possible retriangulation of the patch into Delaunay triangles, there could be other triangles not included in the insertion patch which no longer have empty minspheres. The new triangulation would therefore not be a $2/n$-Delaunay triangulation. For this reason the $O(v^2)$ search is used to determine the insertion patches of k/n-Delaunay triangulations. For n-dimensional Delaunay triangulations the search time complexity can be reduced with tree structures, see [17] for an example. The data structures become more complex to maintain as does code development.

That candidate Delaunay k-simplices are always Delaunay simplices was proven for two and three dimensional Delaunay triangulations in [9]. Since a proof of the two dimensional version of the Generic Delaunay Triangulation Algorithm has not appeared in journal literature, a proof which extends to n dimensions is documented here.

The algorithm produces a convex hull of triangles. Since STEPc is the only step that increases the area of the triangulation and it conforms with the dynamic convex hull algorithm in [21], a convex hull is produced. The following Lemmas are useful, see [17] for proofs.

Lemma 1: The line defined by the intersection of circles ABC and ABD separates C from D if and only if either C lies outside circle ABD and D lies outside circle ABC or C lies inside circle ABD and D lies inside circle ABC

Lemma 2: Let the line defined by A and B separate C from D and let C lie outside circle ABD. If a point E lies inside circle ABD and the line separates E and D, then E lies inside circle ABC.

Theorem 3: The insertion patches of the 2-dimensional Generic Delaunay Triangulation Algorithm are connected polygons.

Proof: Note that a point on a circle is considered to be in the circle. In CASE1 if a new node P is not contained in any circumcircle then the insertion patch must be connected because the set, T, of triangles added to the convex hull in STEPc preserves the convex hull. Let P lie in the circumcircle of triangle DEF which lies inside the convex hull and does not share an edge with any of the triangles in T. Circle DEF is partitioned into triangle DEF and segments whose chords are DE, DF and EF. Since P is not in triangle DEF, P must lie in a segment say, DE. DE cannot be a boundary edge of the convex hull because triangle DEF does not share an edge with any triangle in T. As an interior edge DE is shared by a Delaunay triangle DEG and, by Lemma 1, G is separated from F by DE. By Lemma 2, P lies in circle DEG.

A sequence of triangles DEF, DEG, ... is created where each circle DEF, DEG, ... contains P. The sequence cannot end by repeating a triangle because each edge inside the convex hull is shared by exactly two triangles which implies that P would lie in two disjoint segments of a circle. Since there are only a finite number of triangles, the sequence must end with an edge shared by a triangle in T and therefore the patch is a connected polygon.

The other cases have similar proofs. T is replaced by a triangle containing P, or by the union of two triangles if P lies on an edge.

Theorem 4: The insertion patches of the 2-dimensional Generic Delaunay Triangulation Algorithm are retriangulated into Delaunay triangles.

Proof: Since the insertion patch is connected by Theorem 3, the boundary of the patch is a simple closed polygon. That the retriangulation of the patch is a valid triangulation is proven first.

Let AB and BC be successive patch boundary edges inside the convex hull. Let P be the inserted point and suppose PC intersects AB. This implies that P and A lie on the same side of BC. Since BC is a patch boundary edge, there existed triangles BCE and BCD before the retriangulation. Let E and P lie on opposite sides of BC. P cannot lie in circle BCE because P lies in circle BCD and, by Lemma 2, this contradicts that BC is a boundary edge. If D is not equal to A, BD cannot be a boundary edge of the insertion patch because three boundary edges (AB, BC and BD) cannot share B. Thus there exists triangle BDG whose circle BDG, by Lemma 2, contains P. The sequence of triangles BCD, BDG, ... must end and the last triangle must have edge AB, and this contradicts that edge AB is a patch boundary edge. The cases where AB or BC lies on the boundary of the convex hull are similar.

To show that for edge AB on the insertion patch boundary and inside the convex hull triangle PAB is a Delaunay triangle, Lemma 2 is first applied to the segment of circle PAB with chord AB that lies outside the insertion patch. This segment lies inside the circumcircle of a Delaunay triangle and thus does not contain any triangulation points in its interior. The triangle PAB cannot contain another triangulation point because AB is an interior edge and therefore belongs to a Delaunay triangle ABC where C lies on the same side of AB as P. This implies that the circumcircle of triangle ABC contained P and therefore also contains triangle PAB and the segments PA and PB of the circumcircle of triangle PAB. Since circle ABC contains no triangulation point in its interior other that P, segments PA, PB and triangle PAB cannot contain a triangulation point. Therefore the circumcircle of triangle PAB is empty making triangle PAB a Delaunay triangle. If AB were a boundary edge of the convex hull, segment AB of the circumcirle of triangle PAB cannot contain triangulation points in its interior either. The proof for the other cases are similar.

5. FUNDAMENTAL CALCULATIONS

The mincenter $\mathbf{C}_k$ and minradius R_k of a k-simplex, S_k, in n dimensions are determined with the following equations. The vertices $\mathbf{P}_i$ of S_k define a k-dimensional hyperplane H^k in E^n and the points in H^k are of the form

$$w = (1 - t_1 - \ldots - t_k)\mathbf{P}_1 + t_1\mathbf{P}_2 + \ldots + t_k\mathbf{P}_{k+1} \quad , \quad -\infty < t_k < \infty. \tag{5.1}$$

$\mathbf{C}_k$ satisfies (5.1) and

$$\|\mathbf{C}_k - \mathbf{P}_i\| = R_k \quad , 1 \le i \le k+1, \tag{5.2}$$

where $\| * \|$ denotes Euclidean distance. Substituting $\mathbf{C}_k$ in the form represented in (5.1) into (5.2) yields $k+1$ nonlinear equations in the unknowns $t_1, \ldots, t_k$ and R_k. Equating these equations pairwise and squaring yields k linear equations

$$\begin{aligned} &2\sum_{j=1}^{k}(\mathbf{P}_{j+1} - \mathbf{P}_1) \bullet (\mathbf{P}_{i+1} - \mathbf{P}_i)t_j \\ &= \|\mathbf{P}_{i+1}\|^2 - \|\mathbf{P}_i\|^2 - 2(\mathbf{P}_{i+1} - \mathbf{P}_i) \bullet \mathbf{P}_1 \end{aligned} \tag{5.3}$$

where $\mathbf{P}_j = (p_{j1}, \ldots, p_{jn})$, $1 \le j \le k+1$. The minradius is calculated by substituting the solution of (5.3) into (5.2) for some i_0, $1 \le i_0 \le k+1$.

If S_k lies on boundary of the k/n-Delaunay triangulation, then the separation hyperplane of its $(k-1)$-face on the boundary is represented by the equation

$$\mathbf{A} \bullet \mathbf{X} = d \tag{5.4}$$

where $d = \mathbf{A} \bullet \mathbf{C}_{k-1}$, $\mathbf{A} = \mathbf{C}_k - \mathbf{C}_{k-1}$, $\mathbf{X} = (x_1, \ldots, x_n)$ and $\mathbf{C}_{k-1}$ is the mincenter of the $(k-1)$-face on the boundary. In the special case $\mathbf{C}_k = \mathbf{C}_{k-1}$, as in a right triangle, the vector $\mathbf{A} = \mathbf{P} - \mathbf{C}_k$, where $\mathbf{P}$, expressed as in (5.1) with $t_j = 1$ if $\mathbf{P}_j$ is the vector opposite the $(k-1)$-face containing $\mathbf{C}_k$, is the solution to the following system of $k-1$ linear equations

$$(\mathbf{P} - \mathbf{C}_k) \bullet (\mathbf{P}_i - \mathbf{P}_l) = 0 \quad , 1 \le i \le k+1 \quad j \ne i \ne l. \tag{5.5}$$

For a random set of points the empty minsphere property of two-dimensional Delaunay triangulations is equivalent to the unique triangulation which maximizes the smallest angle in the triangulation [19]. Due to this property, planar Delaunay triangulation algorithms are said to produce as nearly equilateral triangles as possible. This angle property, considered desirable for finite element analysis, was the motivation for defining normalized shape ratios to measure how much the shapes of triangles and tetrahedra deviate from the equilateral triangle and tetrahedron [8]. This ratio has been extremely valuable in constructing three-dimensional finite element meshes.

The *normalized shape ratio*, η_r, of a k-simplex is $(r/R)\rho_k$, where r is its inradius (the radius of its inscribed $(k-1)$-dimensional sphere), R is its circumradius and ρ_k is the circumradius to inradius ratio of the regular k-simplex. The value of R is determined in (5.2) and the values of r and ρ_k can be determined by using the following theorem.

Theorem 5: The incenter $\mathbf{I}_k$ of a k-simplex S_k is

$$\mathbf{I}_k = \left[\sum_{i=1}^{k-1} vol_i \mathbf{P}_i\right] \Big/ \left[\sum_{i=1}^{k-1} vol_i\right] \tag{5.6}$$

where vol_i is the volume of the $(k-1)$-face opposite vertex $\mathbf{P}_i$. The inradius of the k-simplex is

$$r = |\mathbf{A}_i \bullet \mathbf{I}_k - d_i|/||\mathbf{A}_i|| \tag{5.7}$$

where $\mathbf{A}_i \bullet \mathbf{X} = d_i$ is the equation of the $(n-1)$-dimensional separation hyperplane passing through the $(k-1)$-face opposite vertex $\mathbf{P}_i$.

Proof: The inradius satisfies (5.7) because the insphere is tangent to each $(k-1)$-face and (5.7) is the distance of the incenter to a $(k-1)$-face. $\mathbf{I}_k$ and the vertices of a $(k-1)$-face form a k-simplex whose volume is $r(vol_i/k)$. The ratio of this volume to the volume of S_k is the i^{th} barycentric coordinate of $\mathbf{I}_k$. Thus $\mathbf{I}_k$ can be represented as in (5.6) [30].

The standard formula [18] for calculating the volume of a $(k-1)$-simplex with $(k-1)$-dimensional vertices $\mathbf{P}_1$, $\ldots$, $\mathbf{P}_k$ is an $(k-1) \times (k-1)$-determinant

$$Vol = |det(\mathbf{Y}_1, \ldots, \mathbf{Y}_{k-1})|/(k-1)! \tag{5.8}$$

with $(k-1)$-dimensional column vectors $\mathbf{Y}_i = \mathbf{P}_{i+1} - \mathbf{P}_1$. Although the $k+1$ vertices $\mathbf{P}_i$ have n-dimensional coordinates which prevent the direct use of (5.8), vol_i can still be calculated by applying $n-k+1$ Householder transformations [16] followed by the evaluation of a k-dimensional determinant.

Without loss of generality translate the k-simplex so that one of its vertices coincides with the origin. Let $\mathbf{P}_1^{(1)}$, $\ldots$, $\mathbf{P}_k^{(1)}$ be the new coordinates of the remaining vertices and let H_1 be the vector space spanned by these vectors. Extend this linearly independent set of vectors to a basis for E^n by solving the underdetermined set of equations

$$\mathbf{P}_i^{(1)} \bullet \mathbf{X} = 0 \quad ,1 \leq i \leq k \quad . \tag{5.9}$$

A Gram-Schmidt orthonormalization on the solution vectors produces orthonormal vectors $\mathbf{U}_1^{(1)}, \ldots, \mathbf{U}_{n-k}^{(1)}$ which span K_1, the orthogonal complement of H_1. Let T_1 be the Householder transformation which maps $\mathbf{U}_{n-k}^{(1)}$ to $\mathbf{e}_n$ the n-dimensional unit vector $(0, \ldots ,0,1)$ and let $H_2 = \{\mathbf{P}_1^{(2)}, \ldots, \mathbf{P}_k^{(2)}\}$ and $K_2 = \{\mathbf{U}_1^{(2)}, \ldots, \mathbf{U}_{n-k-1}^{(2)}\}$ where

$$\begin{aligned} T_1(\mathbf{P}_i^{(1)}) &= \mathbf{P}_i^{(2)} \quad ,1 \leq i \leq k \quad and \\ T_1(\mathbf{U}_i^{(1)}) &= \mathbf{U}_i^{(2)} \quad ,1 \leq i \leq n-k-1 \quad . \end{aligned} \tag{5.10}$$

Since the vectors in K_2 remain mutually orthogonal, are perpendicular to the vector space spanned by the vectors in H_2, and $\mathbf{e}_n$ is perpendicular to both H_2 and K_2, the projections of H_2 and K_2 into E^{n-1} span E^{n-1}. Redefine H_2 and K_2 with this $n-1$-dimensional projection and iterate with $(n-1)$-, $\ldots$, $(k-1)$-dimensional Householder transformations to produce a set of vectors which span E^k and with the origin is the image the vertices of the original k-simplex.

To determine the $(k-1)$-volumes vol_i in (5.6) use the Householder transformations which map the unique normal of a $(k-1)$-dimensional face sharing the origin to $\mathbf{e}_k$ and project the remaining vectors in H_{n-k} to $(k-1)$-dimensional vectors as required in the determinant formula in (5.8). The volume of the $(k-1)$-face opposite the origin is calculated similarly after translating it so that one of its vertices lies at the origin.

With equations (5.6) and (5.7) the $(k-1)$-volumes vol_i can be calculated recursively without reference to determinants. vol_i of the $(k-1)$-face opposite $\mathbf{P}_i$ is $1/(k-1)$ times the product of its inradius and the sum of the $(k-2)$-volumes of its $(k-2)$-faces. Iterate down dimensions to compute 2-volumes (areas of triangles) by Herron's formula. With (5.6) determine the incenters of the 3-faces and with (5.7) determine the inradii of the 3-faces. With these inradii and 2-volumes, calculate the 3-volumes in (4.6) and determine the incenters of the 4-faces and so on. Thus $(k-1)$-volumes are calculated recursively with $\binom{k+1}{3}$ 2-simplices (triangles), $\binom{k+1}{4}$ 3-simplices (tetrahedra) and $\binom{k+1}{j+1}$ j-simplices, $2 < j < k-1$. Since the coefficients of $\mathbf{P}_i$ in (5.6) are ratios of volumes, the multiplicative constants $1/j$ can be ignored.

The normalizing factors $\rho_2 = 2$ and $\rho_3 = 3$ are easily derived. That the succeeding factors satisfy $\rho_{k+1} = k+1$ is not obvious and, for completeness, a derivation is presented here.

Theorem 6: $\rho_{k+1} = k+1$

Proof: First, the equal r-volumes of the r-faces of a regular k-simplex imply that the incenter and circumcenter of a regular k-simplex must coincide and the orthogonal projection of a regular k-simplex vertex onto its opposite face is the incenter of that $(k-1)$-face.

Let the $k+1$ vertices of a regular k-simplex be $\mathbf{U}_1$, $\ldots$, $\mathbf{U}_{k+1}$ where $\mathbf{U}_1$ is the origin, $\mathbf{U}_2 = (1, 0, \ldots ,0)$ and

$$\mathbf{U}_i = (u_{1i}, \ldots, u_{ni}), u_{ji} = 0, \quad 3 \leq i \leq j \leq n \quad .$$

As a k-face of the regular $(k+1)$-simplex the opposite vertex, $\mathbf{U}_{k+2}$, of this k-simplex is

$$\mathbf{U}_{k+2} = (\mathbf{U}_1 + \ldots + \mathbf{U}_{k+1})/(k+1) + h_{k+1}\mathbf{e}_{k+1} \quad ,$$

where $\mathbf{e}_{k+1}$ is the unit vector along the $(k+1)^{th}$-coordinate axis. From (5.6), the incenter of the $(k+1)$-simplex is

$$\mathbf{I}_{k+1} = (\mathbf{U}_1 + \ldots + \mathbf{U}_{k+2})/(k+2)$$

so that the inradius, the distance between the incenters of the $(k+1)$-simplex and the k-face opposite $\mathbf{U}_{k+2}$ is $h_{k+1}/(k+2)$. The circumradius, the distance from the $(k+2)^{th}$ vertex to the incenter, is given by $h_{k+1} - (h_{k+1}/(k+2))$ and thus $\rho_{k+1} = k+1$.

The next theorem establishes a relaionship between k/n-Delaunay triangulations and n-dimensional Delaunay triangulations.

Theorem 7: A k/n-Delaunay triangulation of random points, subject to the restriction that no subset of $n+2$ points lie on the same $(n-1)$-dimensional sphere, is a subset of the k-faces in the n-dimensional Delaunay triangulation of the random points.

Proof: Without loss in generality assume that there are at least $n+1$ random and affinely independent points so that the n-dimensional Delaunay triangulation is unique. Let M_0 be the center of the minsphere of a k/n-Delaunay triangulation simplex S_k. Translate this simplex so that M_0 lies at the origin and let H_k be the k-dimensional vector space spanned by the vertices of T_k, the translated k-simplex.

Let K_{n-k} be the orthogonal complement of H_k. Each triangulation point that is not in H_k lies on an $(n-1)$-dimensional sphere whose center lies in K_{n-k} and passes through the vertices of T_k. The assumption of affine independence and of a finite number of triangulation points implies that there exists an $(n-1)$-dimensional sphere of least radius which passes through the vertices of T_k and at least one additional triangulation point v_1. Let M_1 be the center of the new sphere and translate the triangulation points so that M_1 is now at the origin. Let T_{k+1} be the new Delaunay $(k+1)$-simplex, H_{k+1} be the $(k+1)$-dimensional vector space spanned by the vertices of T_{k+1}, and $K_{n-(k+1)}$ be the orthogonal complement of H_{k+1}.

Construct a sequence of simplices, T_i, and vector spaces, H_i, and orthogonal complements, K_{n-i}, $k \leq i \leq n$. Since the image of S_k under the sequence of transformations used to contruct T_n is a face of T_n and since T_n is an n-simplex in a translated n dimensional Delaunay triangulation, the theorem is proved.

Finally, let a set of random points undergo a distance preserving transformation before the Generic Delaunay Triangulation algorithm is executed. If the points are inserted in the same order, then the essential numerical values, such as distances and radii of minspheres, will not change. In this sense *k/n-Delaunay triangulations are rotation and translation invariant.* In practice however, the nonuniform distribution of rational numbers represented on a computer can introduce a difference in the final triangulation when triangulation points are nonrandom or nearly so in the sense of Rogers [25].

REFERENCES

1 Alfeld, P. Scattered data interpolation in three or more variables, a survey of the field. *Research Report, Dept. of Math., Univ. of Utah.* 1988, 1-34.

2 Augenbaum, J.M. and Peskin, C.S. On the Construction of the Voronoi Mesh on a Sphere. *J. Comp. Phys.* 1985, 59, 177-192.

3 Bowyer, A. Computing Dirichlet Tesselations. *The Computer Journal.* 1981, 24, 162-166.

4 Cavendish, J.C., Field, D.A. and Frey, W.H. An Approach to Automatic Three-Dimensional Mesh Generation. *Int. J. Num. Meth. Eng.* 1985, 21, 329-347.

5 De Floriani, L. Surface representations based upon triangular grids. *The Visual Computer.* 1987, 3, 27-50.

6 Delaunay, B. Sur la sphere vide. *Bull. L'Acad. des Sci. de L'URSS.* 1934, 793-800.

7 Dyn, N., Levin, D. and Rippa, S. Data dependent triangulations for piecewise linear interpolation. *School of Mathematical Sciences, Tel-Aviv University Preprint.*

8 Field, D.A. Implementing Watson's Algorithm in Three Dimensions. *Proc.Second Ann. ACM Symp. Comp. Geom.* 1986, 246-259.

9 Field, D.A. Watson's Delaunay triangulation without a polygon. *General Motors Res. Pub.* 1985, GMR-5140.

10 Field, D.A. A flexible Delaunay triangulation algorithm. *General Motors Res. Pub.* 1987, GMR-5675.

11 Field, D.A. A surface Delaunay triangulation algorithm. *General Motors Res. Pub.* 1987, GMR-5819.

12 Field, D.A. Laplacian smoothing and Delaunay triangulations. *Comm. Appl. Num. Meth.* 1988, 4, 709-712.

13 Franke, R. Scattered data interpolation: tests of some methods. *Math. Comp.* 1982, 38, 181-200.

14 Franke, R.H. and Schumaker, L.L. A bibliography of multivariate approximation. *Topics in Multivariate Approximation, Proceedings of International Workshop, University of Chile, Dec. 15-19, 1986.* Chui, C.K., Schumaker, L.L. and Uteras, F.I. eds. Academic Press, 1987.

15 Frey, W.H. Selective refinement: a new strategy for automatic node placement in graded triangular meshes. *Int. J. Num. Met. Eng.*. 1987, 24, 2183-2200.

16 Golub, G.H. and Van Loan C.F. *Matrix Computations.* The John Hopkins University Press, Baltimore, MD, 1983.

17 Guibas, L.J. and Stolfi, J. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics.* 1985, 2, 74-123.

18 Hausner, M. *A Vector Space Approach to Geometry.* Prentice-Hall, Englewood Cliffs, NJ, 1965.

19 Lawson, C.L. Software for C1 surface interpolation. *Comp. Aided Geom. Des.*. 1986, 3, 231-246.

20 Nadler, E.J. Piecewise linear approximation on triangulations of a planar region. *Ph.D. thesis,* Brown University, 1985.

21 21. Preparata, F.P. and Shamos, M.I. *Computational Geometry an Introduction.* Springer-Verlag, 1985.

22 Renka, R.J. Interpolation of data on the surface of a sphere. *ACM Trans. Math. Soft.* 1984, 10, 417-436.

23 Renka, R.J. Algorithm 623: Interpolation of data on the surface of a sphere. *ACM Trans. Math. Soft.*. 1984, bf Vol. 10, 437-439.

24 Renka, R.J. Algorithm 624: triangulation and interpolation of arbitrarily distributed points in the plane. *ACM Tran. Math. Soft. 1984.* 1984 10, 440-442.

25 Rogers, C.A. *Packing and Covering.* Cambridge Mathematical Tracts, No. 54, Cambridge University Press 1964.

26 Schumaker, L.L. Triangulation Methods. *Topics in Multivariate Approximation, Proceedings of International Workshop, University of Chile, Dec. 15-19. 1986.* Chui, C.K., Schumaker, L.L. and Uteras, F.I. eds. Academic Press, 1987.

27 Schwartz, S.E. Effects of methanol, water, and engine oil on engine- lubrication-systems elastomers. *Proc. of VII Int. Symp. Alchohol Fuels*, Paris, France 1986.
28 Scott, D.S. The complexity of interpolation given data in three space with a convex function of two variables. *J. Approx. Theory*. 1984, **42**, 52-63.
29 Sibson, R. Locally equiangular triangulations. *Comp. J.* 1978, **21**, 243-245.
30 Vaughan, H.E. and Gabai, H.Hyperspheres associated with an n-simplex. *Amer. Math. Mon.* 1967, **74**, 384-392.
31 Watson, D.F. Computing the n-dimensional Delaunay tesselation with applications to Voronoi polytopes. *Comp. J.* 1981, **24**, 167-172.