Title
Automatic Unstructured Grid Generation Based on Iterative Point Insertion

Journal
Computing, 55

Authors
Hamann, Bernd
Thornburg, H. J.
Hong, G.

Publication Date
1995

Peer reviewed

Computing 55, 135–161 (1995)

Computing

Automatic Unstructured Grid Generation Based on Iterative Point Insertion

B. Hamann, H. J. Thornburg, and **G. Hong** Mississippi State

Received January 16, 1995; revised February 10, 1995

Abstract — Zusammenfassung

Automatic Unstructured Grid Generation Based on Iterative Point Insertion. Unstructured grid generation is concerned with discretizing surfaces and volumes in 3D space by triangles and tetrahedra. The tetrahedra discretize a volume in a surface's interior or surrounding a surface on its outside. This paper presents a new technique for unstructured grid generation based on intersecting lines with a given geometry and inserting grid points yielding an adapted grid whose point density decreases with increasing distance to the geometry. The technique is completely automatic.

AMS Subject Classifications: 65D17, 68U05

Key words: Approximation, grid generation, interpolation, simulated annealing, triangulation, unstructured grid generation.

Ein automatisches Verfahren zur Generierung unstrukturierter Gitter mittels iterativer Punkteinfügungen. Unstrukturierte Diskretisierungsverfahren dienen der diskreten Beschreibung von Flächen und Volumina in Form von Dreiecken und Tetraedern. Tetraeder werden zur Diskretisierung eines Volumens benutzt, das auf der Innen- oder Außenseite einer vorgegebenen Fläche liegt. Diese Arbeit behandelt ein automatisches Verfahren zur Generierung von Tetraedern. Das Verfahren basiert auf einem Schnittalgorithmus, der Geraden mit einer vorgegebenen Fläche schneidet und iterative Punkte einfügt. Die resultierende Punktdichte nimmt mit zunehmender Distanz zur Fläche ab.

1. Introduction

Grid generation is concerned with discretizing surfaces and volumes by polygonal and polyhedral "cells." Typically, an unstructured grid of a surface/volume is set of triangles/tetrahedra. A structured grid, on the other hand, consists of quadrilaterals (surface) and hexahedra (volume). While most numerical algorithms used in computational field simulation (CFS) are based on structured grids, these grids can usually not be generated automatically. Structured grid generation is extremely tedious and time-consuming, especially the interactive process of specifying "blocks" (see [7] and [19]). Unstructured grid generation techniques are usually automated to a high degree and require little user input, which is its main advantage over the structured grid generation approach. Also, the use of unstructured grids for CFS applications has increased significantly in recent years (see [20]). Most algorithms for generating unstructured-triangular

and unstructured-tetrahedral grids are based on the *Delaunay triangulation* or the so-called *advancing front* method.

The Delaunay triangulation has been used for various applications, including scattered data interpolation. In this context, function values f_i are given at scattered locations $\mathbf{x}_i$ for which no connectivity is given. Typically, the Delaunay triangulation is computed for the points $\mathbf{x}_i$ and triangular (tetrahedral) interpolants are constructed. The Delaunay triangulation is characterized by the fact that the circle (sphere) passing through the vertices of any triangle (tetrahedron) does not contain any point of the original point set in its interior. In the 2D case, the Delaunay triangulation is the max-min angle triangulation of the given point set, i.e., it maximizes the minimum angle in the triangulation (see [17]). This property is very desirable for many applications, particularly for grid generation.

The Delaunay triangulation has found wide popularity in the finite element method (FEM) and unstructured grid generation communities. In general, triangles (tetrahedra) must be generated for a simply connected region in 2D (3D). They are generated in two steps. The first step is the generation of a boundary grid for the boundary curves (boundary surfaces) of the space to be discretized, i.e., a set of boundary conforming line segments (triangles) is computed. The second step is the generation of triangles (tetrahedra) inside this boundary grid (see [21, 22, 24]).

The unstructured grid can be generated by inserting points and perofrming local retriangulation iteratively or by first generating the entire point set and then triangulating it, possibly useing parallel programming paradigms. Grid points are placed such that certain quality measures (edge lengths, areas, volumes, angles, ratios thereof, etc.) and specified distributions are satisfied.

The grid points are typically chosen according to geometrical properties of the boundary (e.g., arc length or absolute curvature) and distribution functions. Spacing parameters are used for the boundary point distribution, and so-called *sources* (i.e., point, line, and plane sources) are used to further control grid point distributions in the interior. Grid point densities decrease in a predefined fashion with increasing distance from the sources. Grids must be *graded* in this fashion due to the sometimes sudden occurrence of discontinuities in certain field parameters, e.g., shocks in flow fields.

The advancing front method generates unstructured grids in a completely different way. The input—for the generation of a triangular (tetrahedral) grid—is a set of oriented line segments (oriented triangles) discretizing the boundary curves (surfaces) of some geometry. Triangles (tetrahedra) are then constructed by advancing a *front* into the interior of the field until it is completely filled with elements. Depending on local edge and angle configurations of the current front, new elements are created by connecting existing

points or by inserting new points—according to some distribution function—and connecting them with existing ones (see [13]).

The desired point distribution is typically defined by a set of points with associated spacing parameters (background grid). Interpolating these spacing parameters yields the desired spacing at any point in the field. Points are inserted such that the desired spacing is optimally satisfied. The advancing front strategy is also used for the generation of unstructured grids consisting of quadrilaterals (hexahedra).

When generating an unstructured grid, the local density of triangles/tetrahedra is usually controlled by a distribution function and/or a so-called background mesh. For CFS solution algorithms, the density of triangles and tetrahedra must be higher in surface regions and boundary layers with complex field/flow phenomena. Unfortunately, the local complexity of the field/flow is not known prior to a first CFS solution, and only geometric properties can be considered for computing an initial unstructured grid. It is obvious to require that the initial density of grid points should purely depend on surface curvature (surface grid) and on distance to the surface (volume grid). This approach is used in the method discussed in this paper.

Recent advances in grid generation are discussed in [3]. There are two basic approaches when generating an unstructured grid. The first approach is based on an *advancing front* technique (see [13 and 16]), and the second approach is based on the *Delaunay triangulation* for point sets (see [1], [2], [10], [11], [12], [21], [22]).

The unstructured grid generation technique discussed in this paper is based on computational geometry and geometric modeling concepts and methods. An initial volume grid is generated by intersecting lines with the given geometry. This initial volume triangulation, consisting of a uniform "density" of tetrahedra, is iteratively modified by inserting points until a desired "density" of tetrahedra is obtained. A general reference to computational geometry is [17], and an excellent guide to geometric modeling is [5].

A technique called *marching cubes* is used to generate in initial volume triangulation. The *marching cubes* algorithm was first introduced for computing isosurfaces for rectilinear, scalar-valued data sets. The original technique is described in [14], and a simple error in the original technique is corrected in [15]. The approximation of surface curvature is discussed in [8], the reduction of surface triangulations in [9], and the optimization of surface triangulations in [4].

In the following, it is assumed that a closed surface is given such that each 3D point can be characterized as an exterior, as an interior, or as a point on the surface. The overall algorithm for automatically creating an unstructured volume grid in the interior of a closed surface or surrounding a closed surface on its

exterior is divided into these steps:

(i) Compute a first volume triangulation of a finite volume containing the given geometry completely.
(ii) Intersect families of lines defined by edges of tetrahedra in the first volume triangulation with a given surface triangulation of the closed surface.
(iii) Characterize each vertex of each tetrahedron in the first volume triangulation as an exterior point ("−1"), as a point on the surface ("0", or as an interior point ("1"), depending on the orientation of the surface.
(iv) Compute/extract a second volume triangulation for the part of the bounding volume being inside (outside) the closed surface.
(v) Map all boundary vertices in the second volume triangulation to points lying exactly on the surface, provided that an analytical, parametric definition of the surface is given.
(vi) Weight the tetrahedra in the second volume triangulation with respect to their centroids' distance to the surface and, optionally, with respect to the surface's local absolute curvature.
(vii) Insert additional grid points, update the triangulation locally, and improve it by considering various quality criteria.
(viii) Optionally, extract the surface grid from the volume grid.

These steps are described in detail in the following sections.

2. Generating the Unstructured Volume Grid

The first step is the computation of a volume triangulation of a bounding volume of the given closed surface. This bounding volume is a "scaled bounding box," and it is denoted by

$$\mathscr{V} = [X_0, X_1] \times [Y_0, Y_1] \times [Z_0, Z_1]. \tag{2.1}$$

It must contain the given surface completely. The surface itself is given by a set of points, denoted by

$$\mathscr{P} = \{\mathbf{x}_i = (x_i, y_i, z_i) | i = 1, \ldots, n\} \subset \mathscr{V}, \tag{2.2}$$

and an associated surface triangulation, denoted by

$$\mathscr{T} = \{(v_1^j, v_2^j, v_3^j) | j = 1, \ldots, n_t\}, \tag{2.3}$$

where a triple (v_1^j, v_2^j, v_3^j) refers to three points in $\mathscr{P}$ defining a triangle. It is assumed that a parametric surface description is known. The triangulation $\mathscr{T}$ defines a piecewise linear (triangular) approximation of the surface.

In the following, boldface characters are used for 3D points (e.g., $\mathbf{x} = (x, y, z)$). Denoting the corner points (X_I, Y_J, Z_K), $I, J, K \in [0, 1]$, of the bounding volume

by $\mathbf{b}_{I,J,K}$, the bounding volume $\mathscr{V}$ is initially discretized by using the points

$$\mathbf{y}_{i,j,k} = \mathbf{y}(u_i, v_j, w_k) = \sum_{K=1}^{1} \sum_{J=1}^{1} \sum_{I=0}^{1} \mathbf{b}_{I,J,K} B_I^1(u_i) B_J^1(v_j) B_K^1(w_k),$$
$$i = 0, \ldots, n_x,\ j = 0, \ldots, n_y,\ k = 0, \ldots, n_z, \tag{2.4}$$

where $B_n^1(t) = (1-t)^{1-n} t^n$, $t \in [0, 1]$, $u_i = i/n_x$, $v_j = j/n_y$, and $w_k = k/n_z$.

Each cuboid defined by the eight vertices $\mathbf{y}_{i+r,j+s,k+t}$, $i \in \{0, \ldots, n_x - 1\}$, $j \in \{0, \ldots, n_y - 1\}$, $k \in \{0, \ldots, n_z - 1\}$, and $r, s, t \in \{0, 1\}$, is split into six tetrahedra (see Fig. 1). These tetrahedra are given by the point sets

$$T_{i,j,k}^1 = \{\mathbf{y}_{i,j,k}, \mathbf{y}_{i,j+1,k}, \mathbf{y}_{i+1,j+1,k}, \mathbf{y}_{i+1,j+1,k+1}\},$$
$$T_{i,j,k}^2 = \{\mathbf{y}_{i,j,k}, \mathbf{y}_{i,j+1,k}, \mathbf{y}_{i,j+1,k+1}, \mathbf{y}_{i+1,j+1,k+1}\},$$
$$T_{i,j,k}^3 = \{\mathbf{y}_{i,j,k}, \mathbf{y}_{i,k+1}, \mathbf{y}_{i,j+1,k+1}, \mathbf{y}_{i+1,j+1,k+1}\},$$
$$T_{i,j,k}^4 = \{\mathbf{y}_{i,j,k}, \mathbf{y}_{i+1,j,k}, \mathbf{y}_{i+1,j+1,k}, \mathbf{y}_{i+1,j+1,k+1}\},$$
$$T_{i,j,k}^5 = \{\mathbf{y}_{i,j,k}, \mathbf{y}_{i+1,j,k}, \mathbf{y}_{i+1,j,k+1}, \mathbf{y}_{i+1,j+1,k+1}\}, \quad \text{and}$$
$$T_{i,j,k}^6 = \{\mathbf{y}_{i,j,k}, \mathbf{y}_{i,j,k+1}, \mathbf{y}_{i+1,j,k+1}, \mathbf{y}_{i+1,j+1,k+1}\}. \tag{2.5}$$

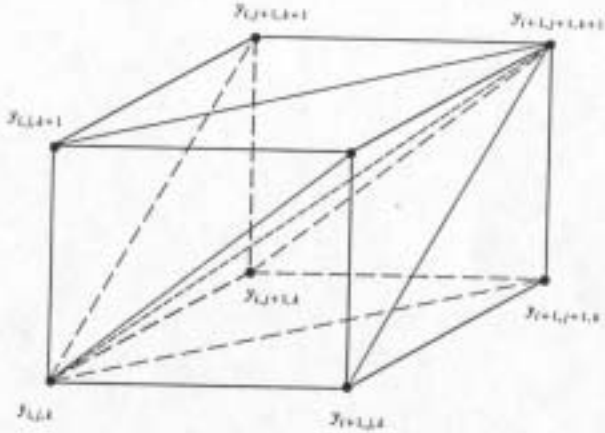

Figure 1. Volume triangulation of single cuboid

The edges of the tetrahedra in this first volume triangulation define families of lines. Some of the lines possibly pass through the interior of the given surface. The line families are defined parametrically. They are

$$\mathrm{l}^1(t) = \mathbf{y}_{0,j,k} + t(\mathbf{y}_{1,j,k} - \mathbf{y}_{0,j,k}),\ j = 0, \ldots, n_y,\ k = 1, \ldots, n_z,$$
$$\mathrm{l}^2(t) = \mathbf{y}_{i,0,k} - t(\mathbf{y}_{i,1,k} - \mathbf{y}_{i,0,k}),\ i = 0, \ldots, n_x,\ k = 0, \ldots, n_z,$$

$$
\begin{aligned}
&\mathbf{l}^{3}(t) = \mathbf{y}_{i,j,0} + t(\mathbf{y}_{i,j,1} - \mathbf{y}_{i,j,0}),\ i = 0,\ldots,n_x,\ j = 0,\ldots,n_y,\\
&\mathbf{l}^{4a}(t) = \mathbf{y}_{0,j,k} + t(\mathbf{y}_{1,j+1,k} - \mathbf{y}_{0,j,k}),\ j = 1,\ldots,(n_y-1),\ k = 0,\ldots,n_z,\\
&\mathbf{l}^{4b}(t) = \mathbf{y}_{i,0,k} + t(\mathbf{y}_{i+1,1,k} - \mathbf{y}_{i,0,k}),\ i = 0,\ldots,(n_x-1),\ k = 0,\ldots,n_z,\\
&\mathbf{l}^{5a}(t) = \mathbf{y}_{0,j,k} + t(\mathbf{y}_{1,j,k+1} - \mathbf{y}_{0,j,k}),\ j = 0,\ldots,n_y,\ k = 1,\ldots,(n_z-1),\\
&\mathbf{l}^{5b}(t) = \mathbf{y}_{i,j,0} + t(\mathbf{y}_{i+1,j,1} - \mathbf{y}_{i,j,0}),\ i = 0,\ldots,(n_x-1),\ j = 0,\ldots,n_y,\\
&\mathbf{l}^{6a}(t) = \mathbf{y}_{i,0,k} + t(\mathbf{y}_{i,1,k+1} - \mathbf{y}_{i,0,k}),\ i = 0,\ldots,n_x,\ k = 1,\ldots,(n_z-1),\\
&\mathbf{l}^{6b}(t) = \mathbf{y}_{i,j,0} + t(\mathbf{y}_{i,j+1,1} - \mathbf{y}_{i,j,0}),\ i = 0,\ldots,n_x,\ j = 0,\ldots,(n_y-1),\\
&\mathbf{l}^{7a}(t) = \mathbf{y}_{0,j,k} + t(\mathbf{y}_{1,j+1,k+1} - \mathbf{y}_{0,j,k}),\ j = 1,\ldots,(n_y-1),\ k = 1,\ldots,(n_z-1),\\
&\mathbf{l}^{7b}(t) = \mathbf{y}_{i,0,k} + t(\mathbf{y}_{i+1,1,k+1} - \mathbf{y}_{i,0,k}),\ i = 0,\ldots,(n_x-1),\ k = 1,\ldots,(n_z-1),\quad \text{and}\\
&\mathbf{l}^{7c}(t) = \mathbf{y}_{i,j,0} + t(\mathbf{y}_{i+1,j+1,1} - \mathbf{y}_{i,j,0}),\ i = 0,\ldots,(n_x-1),\ j = 0,\ldots,(n_y-1),
\end{aligned}
\tag{2.6}
$$

where t is a real number. All edges of all tetrahedra in the first volume triangulation belong to one of these line families. The line families are illustrated in Fig. 2 for the volume triangulation of a single cuboid.

It is assumed that the entire closed surface is given by a set of triangles approximating the surface ((2.2) and (2.3)). Each line in each line family is used to compute intersections with the triangles in $\mathcal{T}$. Thus, it is possible to assign indicators to each point $\mathbf{y}_{i,j,k}$ based on the given surface triangulation. A point $\mathbf{y}_{i,j,k}$ can lie outside the closed surface ("-1"), lie within a surface triangle ("0"), or lie inside the closed surface ("1"). Such indicators are needed for the proper construction of the unstructured volume grid, which must be comprised solely of points in the interior (exterior) and points on the surface.

One remark must be made with respect to generating interior and exterior grids: The following discussion is restricted to the case of generating a grid for a volume in the interior of the given surface. If one needs to compute a grid for a volume surrounding the given geometry on its outside, one simply inverts the outside/inside indicators for all vertices $\mathbf{y}_{i,j,k}$; inside becomes outside and outside becomes inside.

Once all vertices in the first volume triangulation are characterized as exterior, interior, or surface points, it is possible to distinguish between tetrahedra entirely lying on the outside of the given surface, entirely lying on the inside, or lying partially on the outside and partially on the inside. In the next step, a second set of tetrahedra is constructed, which is obtained from the tetrahedra in the first volume triangulation. The part of a tetrahedron in the first volume triangulation that lies inside the closed surface is extracted and contributes to the second volume triangulation, which is used to discretize the interior volume.

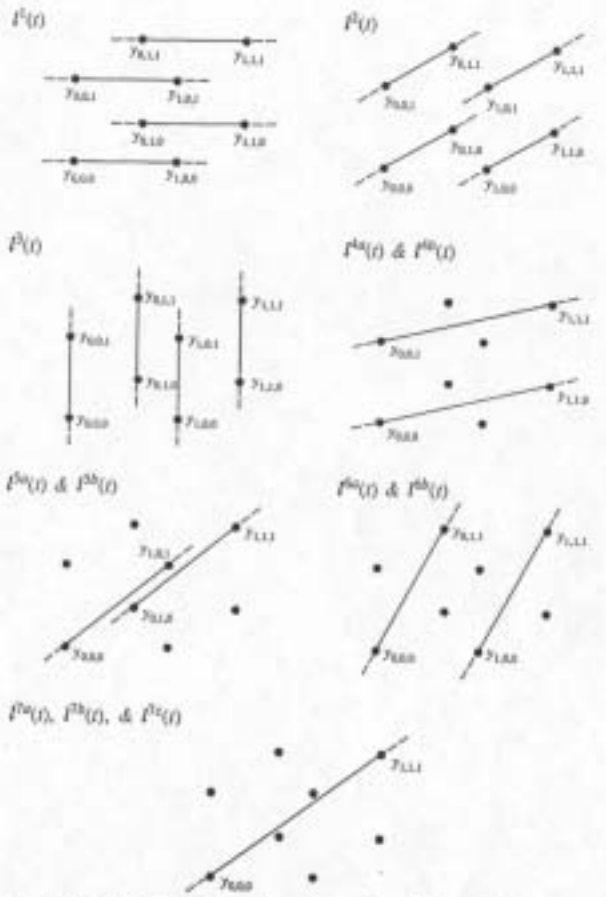

Figure 2. Families of lines determined by edges of tetrahedra within single cuboid

The vertex indicators "-1," "0," and "1," computed for each vertex $\mathbf{v}_i$ of each tetrahedron in the first volume triangulation, are used to determine whether an edge $\overline{\mathbf{v}_i\mathbf{v}_j}$ of a tetrahedron has an intersection point $\mathbf{p}_{i,j}$ with the given surface. Overall, there are fourteen cases to be considered, assuming that an edge intersects a surface triangle in at most one point. If an edge intersects a surface triangle in more than one point, special care must be taken. This is discussed later. The fourteen different cases given in Table 1 are obtained by considering all possible combinations for vertex indicators.

The fourteen cases are illustrated in Fig. 3. Circles indicate exterior points ("-1"), stars indicate points in a surface triangle ("0"), and boldface points indicate interior points ("1").

For these fourteen cases, possible volume triangulations for the interior part of

Table 1. Vertex indicators and intersections for single tetrahedron ($\mathbf{p}_{i,j} = 1$ (0): edge $\overline{\mathbf{v}_i\mathbf{v}_j}$ does (does not) intersect surface in point $\mathbf{p}_{i,j}$)

Case	Vertex indicators for $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$	Possible intersection points $\mathbf{p}_{1,2}$	$\mathbf{p}_{1,3}$	$\mathbf{p}_{1,4}$	$\mathbf{p}_{2,3}$	$\mathbf{p}_{2,4}$	$\mathbf{p}_{3,4}$
1	−1	−1	−1	−1	0	0	0	0	0	0
2	0	−1	−1	−1	0	0	0	0	0	0
3	0	0	−1	−1	0	0	0	0	0	0
4	0	0	0	−1	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	1	−1	−1	−1	1	1	1	0	0	0
7	1	0	−1	−1	0	1	1	0	0	0
8	1	0	0	−1	0	0	1	0	0	0
9	1	0	0	0	0	0	0	0	0	0
10	1	1	−1	−1	0	1	1	1	1	0
11	1	1	0	−1	0	0	1	0	1	0
12	1	1	1	−1	0	0	1	0	1	1
13	1	1	1	0	0	0	0	0	0	0
14	1	1	1	1	0	0	0	0	0	0

a tetrahedron are summarized in Table 2. By extracting the interior part, one obtains the second volume triangulation for the interior volume.

Considering case 12 in Fig. 3, there are two other volume triangulations that must be considered. Denoting the centroid of the six points $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, $\mathbf{v}_{1,4}$, $\mathbf{v}_{2,4}$, and $\mathbf{v}_{3,4}$ by $\mathbf{c}$, one might have to generate a triangulation for these seven points. Thus, the volume triangulation 12h is given by the eight tetrahedra $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{c})$, $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_{1,4}, \mathbf{c})$, $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_{3,4}, \mathbf{c})$, $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_{2,4}, \mathbf{c})$, $(\mathbf{v}_1, \mathbf{v}_{1,4}, \mathbf{v}_{3,4}, \mathbf{c})$, $(\mathbf{v}_2, \mathbf{v}_{1,4}, \mathbf{v}_{2,4}, \mathbf{c})$, $(\mathbf{v}_3, \mathbf{v}_{3,4}, \mathbf{c})$, and $(\mathbf{v}_{1,4}, \mathbf{v}_{2,4}, \mathbf{v}_{3,4}, \mathbf{c})$, and 12i is given by the eight tetrahedra $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{c})$, $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_{2,4}, \mathbf{c})$, $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_{1,4}, \mathbf{c})$, $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_{3,4}, \mathbf{c})$, $(\mathbf{v}_1, \mathbf{v}_{1,4}, \mathbf{v}_{2,4}, \mathbf{c})$, $(\mathbf{v}_2, \mathbf{v}_{2,4}, \mathbf{v}_{3,4}, \mathbf{c})$, $(\mathbf{v}_3, \mathbf{v}_{1,4}, \mathbf{v}_{3,4}, \mathbf{c})$, and $(\mathbf{v}_{1,4}, \mathbf{v}_{2,4}, \mathbf{v}_{3,4}, \mathbf{c})$. Since the part of the first volume triangulation that lies inside the closed surface must be triangulated consistently, i.e., all tetrahedra must "match," and all the volume triangulations listed in Table 2, including 12h and 12i, much be considered.

As mentioned above, it is possible that an edge of a tetrahedron in the first volume triangulation has more than one intersection point with the given surface, or an edge might lie completely or partially in one of the triangles in the surface triangulation. Special care is required for these cases. The end points of each edge of each tetrahedron in the first volume triangulation are characterized by indicators defining whether an end point is outside the surface ("−1"), in some surface triangle ("0"), or on the inside of the surface ("1"). A line-polyhedron intersection test is used for the case of multiple intersections. This is necessary for the proper characterization of vertices as interior or exterior points. Since the closed surface is given as a closed polyhedron with triangular faces, the test uses a theorem for intersecting a line segment with a closed polyhedron.

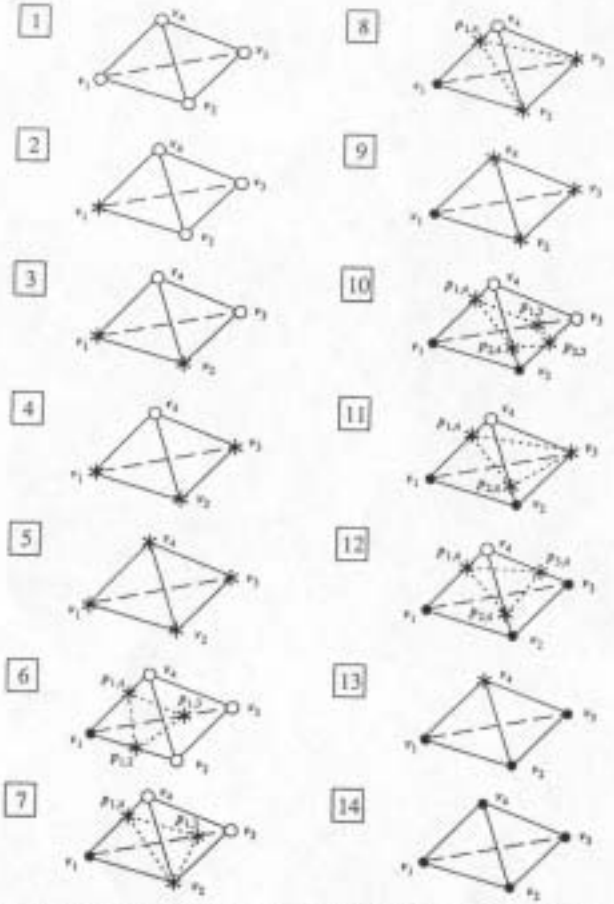

Figure 3. Vertex indicators and intersection points for single tetrahedron

Theorem. *If $\mathbf{v}_i$ is an exterior point and the edge $\overline{\mathbf{v}_i\mathbf{v}_j}$ has an even number of intersections with the closed polyhedron, then $\mathbf{v}_j$ is an exterior point, otherwise $\mathbf{v}_j$ is an interior point. If $\mathbf{v}_i$ is an interior point and the edge $\overline{\mathbf{v}_i\mathbf{v}_j}$ has an even number of intersections with the closed polyhedron, then $\mathbf{v}_i$ is an interior point, otherwise $\mathbf{v}_j$ is an exterior point.*

Depending on the outcome of this test, the indicators of vertices of tetrahedra in the first volume triangulation are modified before constructing the second volume triangulation of the volume on the inside of the surface. Table 3 shows the different possibilities that can occur when performing the test.

The cases listed in Table 3 are shown in Fig. 4. Again, circles indicate exterior points, stars indicate points in a surface triangle, and boldface points indicate interior points.

Table 2. Possible volume triangulations for cases 1–14 (quadruples defining the vertices of a tetrahedron)

Case	First tetrahedron	Second tetrahedron	Third tetrahedron
1–4	—	—	—
5	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$	—	—
6	$(\mathbf{v}_1, \mathbf{p}_{1,2}, \mathbf{p}_{1,3}, \mathbf{p}_{1,4})$	—	—
7	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{p}_{1,3}, \mathbf{p}_{1,4})$	—	—
8	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{p}_{1,4})$	—	—
9	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$	—	—
10a	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{p}_{1,3}, \mathbf{p}_{1,4})$	$(\mathbf{v}_2, \mathbf{p}_{1,3}, \mathbf{p}_{1,4}, \mathbf{p}_{2,4})$	$(\mathbf{v}_2, \mathbf{p}_{1,3}, \mathbf{p}_{2,3}, \mathbf{p}_{2,4})$
10b	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{p}_{1,3}, \mathbf{p}_{2,4})$	$(\mathbf{v}_1, \mathbf{p}_{1,3}, \mathbf{p}_{1,4}, \mathbf{p}_{2,4})$	$(\mathbf{v}_2, \mathbf{p}_{1,3}, \mathbf{p}_{2,3}, \mathbf{p}_{2,4})$
10c	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{p}_{1,4}, \mathbf{p}_{2,3})$	$(\mathbf{v}_1, \mathbf{p}_{1,3}, \mathbf{p}_{1,4}, \mathbf{p}_{2,3})$	$(\mathbf{v}_2, \mathbf{p}_{1,4}, \mathbf{p}_{2,3}, \mathbf{p}_{2,4})$
10d	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{p}_{2,3}, \mathbf{p}_{2,4})$	$(\mathbf{v}_1, \mathbf{p}_{1,3}, \mathbf{p}_{1,4}, \mathbf{p}_{2,4})$	$(\mathbf{v}_1, \mathbf{p}_{1,3}, \mathbf{p}_{2,3}, \mathbf{p}_{2,4})$
11a	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{p}_{1,4})$	$(\mathbf{v}_2, \mathbf{v}_3, \mathbf{p}_{1,4}, \mathbf{p}_{2,4})$	—
11b	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{p}_{2,4})$	$(\mathbf{v}_1, \mathbf{v}_3, \mathbf{p}_{1,4}, \mathbf{p}_{2,4})$	—
12a	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{p}_{1,4})$	$(\mathbf{v}_3, \mathbf{v}_2, \mathbf{p}_3, \mathbf{p}_{3,4})$	$(\mathbf{v}_2, \mathbf{p}_{1,4}, \mathbf{p}_{2,4}, \mathbf{p}_{3,4})$
12b	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{p}_{1,4})$	$(\mathbf{v}_2, \mathbf{v}_3, \mathbf{p}_{1,4}, \mathbf{p}_{2,4})$	$(\mathbf{v}_3, \mathbf{p}_{1,4}, \mathbf{p}_{2,4}, \mathbf{p}_{3,4})$
12c	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{p}_{3,4})$	$(\mathbf{v}_2, \mathbf{v}_3, \mathbf{p}_{1,4}, \mathbf{p}_{3,4})$	$(\mathbf{v}_2, \mathbf{p}_{1,4}, \mathbf{p}_{2,4}, \mathbf{p}_{3,4})$
12d	$(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_3, \mathbf{p}_{2,4})$	$(\mathbf{v}_1, \mathbf{v}_3, \mathbf{p}_{1,4}, \mathbf{p}_{2,4})$	$(\mathbf{v}_3, \mathbf{p}_{1,4}, \mathbf{p}_{2,4}, \mathbf{p}_{3,4})$
12e	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{p}_{2,4})$	$(\mathbf{v}_1, \mathbf{v}_3, \mathbf{p}_{2,4}, \mathbf{p}_{3,4})$	$(\mathbf{v}_1, \mathbf{p}_{1,4}, \mathbf{p}_{2,4}, \mathbf{p}_{3,4})$
12f	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{p}_{3,4})$	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{p}_{2,4}, \mathbf{p}_{3,4})$	$(\mathbf{v}_1, \mathbf{p}_{1,4}, \mathbf{p}_{2,4}, \mathbf{p}_{3,4})$
13	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$	—	—
14	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$	—	—

Table 3. Adjusting vertex indicators of edges in first volume triangulation in case of multiple intersections of edges with given surface

Case	Initial indicators for $\mathbf{v}_i$	Initial indicators for $\mathbf{v}_j$	No. of intersections between $\mathbf{v}_i$ and $\mathbf{v}_j$	New indicators for $\mathbf{v}_i$	New indicators for $\mathbf{v}_j$	Selected intersection point $\mathbf{p}_{i,j}$ between $\mathbf{v}_i$ and $\mathbf{v}_j$
1	−1	−1	any number	−1	−1	none
2	−1	0	any number	−1	−1	none
3a	−1	1	even number	−1	−1	none
3b	−1	1	odd number	−1	1	point closest to $\mathbf{v}_i$
4	0	0	any number	1	1	none
5	0	1	any number	1	1	none
6	1	1	any number	1	1	none

If an entire edge or a part of an edge of a particular tetrahedron in the first volume triangulation is contained in one or many triangles in the surface triangulation, the two end points of the intersection are computed for that particular edge, and the end points of that edge obtain the appropriate indicators. Two possible cases are shown in Fig. 5.

Each point $\mathbf{p}$ lying in the interior or on an edge of a surface triangle is mapped onto the exact parametric surface. This is done by expressing the point $\mathbf{p}$ in terms of barycentric coordinates, using the barycentric coordinates to obtain a

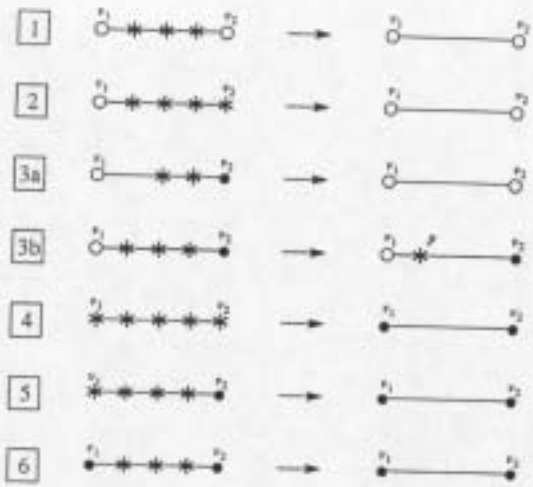

Figure 4. Adjusting vertex indicators of edge in first volume triangulation in case of multiple intersections of edges with given surface

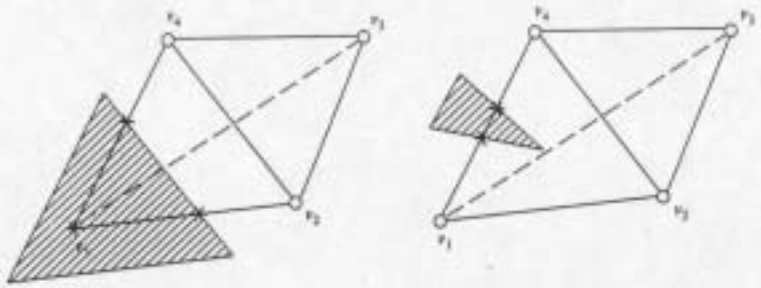

Figure 5. Edges of tetrahedra contained in surface triangle

parameter tuple in the surface's domain, and computing an exact surface point. By writing the point $\mathbf{p}$ as the convex combination

$$\mathbf{p} = \bar{u}_1\mathbf{v}_1 + \bar{u}_2\mathbf{v}_2 + \bar{u}_3\mathbf{v}_3, \tag{2.7}$$

where $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ are the vertices of the triangle containing $\mathbf{p}$, one obtains the parameter tuple

$$(\bar{u}, \bar{v}) = \bar{u}_1(u_1, v_1) + \bar{u}_2(u_2, v_2) + \bar{u}_3(u_3, v_3), \tag{2.8}$$

where (u_k, v_k), $k = 1, 2, 3$, is the parameter tuple associated with the vertex $\mathbf{v}_k$. One evaluates the parametric surface $\mathbf{s}$ containing $\mathbf{v}_k$ at $(\bar{u}, \bar{v})$ and replaces $\mathbf{p}$ by $\mathbf{s}(\bar{u}, \bar{v})$. This principle is illustrated in Fig. 6.

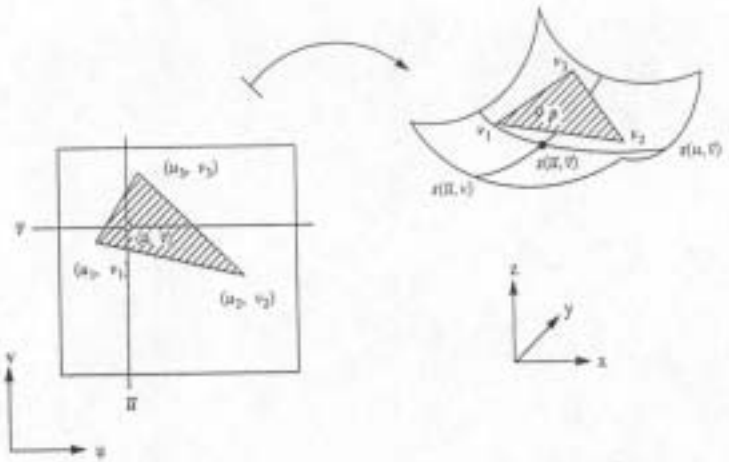

Figure 6. Computing exact surface point using barycentric coordinates

3. Weighting Tetrahedra and Point Insertion

Once an initial interior volume grid is known as the result of constructing the second volume triangulation consisting of interior tetrahedra, one must still increase the number of grid points. It is assumed that the number of tetrahedra in the second volume triangulation is extremely small and not yet sufficiently "dense" enough for later field simulation. Since fields are typically more complex in the boundary layers close to the given surface, the density of grid points should be higher in these boundary regions. Each tetrahedron can be weighted according to its volume and its centroid's distance to the surface. In other words, the volumes of the tetrahedra should decrease with increasing distance to the surface.

A linear function V is used to define the "desired" volume of a particular tetrahedron T_i. This function is

$$V(d_i) = \left[1 - \left(\frac{d_i}{d_1}\right)\right] V_0 + \left(\frac{d_i}{d_1}\right) V_1, \tag{3.1}$$

where V_0 and V_1 are specified tetrahedral volumes at distance 0 and $d_1 > 0$. Here, the distance d_i between a tetrahedron T_i having vertices $\mathbf{v}_1^i$, $\mathbf{v}_2^i$, $\mathbf{v}_3^i$, and $\mathbf{v}_4^i$ and the surface is defined as the smallest distance between the tetrahedron's centroid and the surface, i.e.,

$$d_i = \min\left\{\left\| \frac{1}{4} \sum_{j=1}^{4} \mathbf{v}_j^i - \mathbf{x}_k \right\|, k = 1, \ldots, n\right\}, \tag{3.2}$$

where "|| ||" indicates the Euclidean norm, and $\mathbf{x}_k$ is an original surface point. Equation (3.1) can be interpreted in an alternative way: If V_i is the volume of tetrahedron T_i, and T_j has a distance d_j associated with it, then T_i should have a volume $V(d_j)$. Therefore, the difference between actual and "desired" volume is defined as

$$E_i = |V_i - V(d_i)|. \tag{3.3}$$

The *average difference* of the whole set of tetrahedra and the "desired" volumes is defined as

$$E_{avg} = \frac{1}{n_t} \sum_{i=1}^{n_t} E_i, \tag{3.4}$$

where n_t is the total number of tetrahedra.

It is possible to improve the overall quality of the volume triangulation with respect to (3.4) by iteratively splitting tetrahedra into multiple ones such that E_{avg} becomes smaller. Thus, the tetrahedron $T' \in \{T_i\}_{i=1}^{n_t}$ must be identified such that splitting it into four new ones results in the smallest possible new value E_{avg}'. Splitting a tetrahedron into four new ones results in the new average difference

$$E_{avg}' = \frac{1}{n_t + 3} \sum_{i=1}^{n_t+3} E_i' \tag{3.5}$$

Equation (3.1) can be generalized to

$$V(d_i) = \left[1 - \left(\frac{d_i}{d_1}\right)^p\right] V_0 + \left(\frac{d_i}{d_1}\right)^0 V_1, \qquad 0 \le p < \infty. \tag{3.6}$$

Thus, tetrahedral volumes increase in a linear fashion with increasing surface distance for $p = 1$.

Another generalization is feasible as well. The distance d_i of a tetrahedron can be weighted by the absolute curvature of the surface point that is closest to the tetrahedron's centroid. The absolute curvature at the closest surface point that is closest to the tetrahedron's centroid. The absolute curvature at the closest surface point is denoted by κ_i, and it should have the following effect on the tetrahedral volumes in the overall mesh: The higher the absolute curvature of a surface point, the smaller the volume of tetrahedra in the region close to this surface point. It is proposed to use the value

$$\omega_i = \frac{\kappa_{\max} - \kappa_i}{\kappa_{\max} - \kappa_{\min}} + \frac{1}{10} \frac{\kappa_i - \kappa_{\min}}{\kappa_{\max} - \kappa_{\min}} \in \left[\frac{1}{10}, 1\right] \tag{3.7}$$

as a weight for d_i, provided that $\kappa_{\min} \neq \kappa_{\max}$. Here, $\kappa_{\min}$ is the minimum absolute curvature and $\kappa_{\max}$ the maximum absolute curvature in the set of all surface points (see [8] for the approximation of curvature at points in a surface triangulation).

Incorporating this principle, the tetrahedral volumes are proportional to their distances to the surface and inversely proportional to the absolute surface

curvature at the closest surface point. In order to use the absolute surface curvature in (3.1) and (3.6), the distance terms are replaced. This generalization of (3.6) yields

$$V(d_i) = \left[1 - \left(\frac{\omega_i d_i}{d_1}\right)^p\right] V_0 + \left(\frac{\omega_i d_i}{d_1}\right)^p V_1, \qquad 0 \le p < \infty. \tag{3.8}$$

Since the given surface might contain tangent plane discontinuities and absolute curvature discontinuities (e.g., cusps) that must be preserved, surface points where such discontinuities occur are assigned infinite weight. Figure 7 illustrates the whole process for the planar case. A closed curve is given, and a triangulation is constructed considering distance to and curvature of the surface. Triangle T_j is split by inserting its centroid $\mathbf{c}$ as an additional vertex.

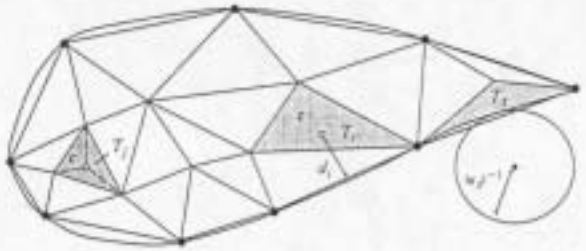

Figure 7. Distance and curvature associated with triangles in planar triangulation and splitting triangles

Instead of considering purely geometrical criteria, such as distance to the geometry and surface curvature, one can also use a solution-based strategy for grid adaptation. Assuming one prescribes tetrahedral volumes V_I at certain 3D locations (x_I, y_I, z_I), $I = 1, \ldots, N$, based on some flow solution, one can construct a scattered data interpolant $V(x, y, z)$ that interpolates this data (see [6]). The scattered data interpolant is defined by a linear system of the form

$$V(x_I, y_I, z_I) = \sum_{i=1}^{N} c_i \psi_i(x_I, y_I, z_I), \qquad I = 1, \ldots N, \tag{3.9}$$

where ψ_i are appropriately chosen basis function. The interpolant $V(x, y, z)$ can be evaluated at the centroid of any tetrahedron in the grid, and the difference between $V(x, y, z)$ and the actual tetrahedral volumes must be minimized.

One of the most common methods for scattered data interpolant is Hardy's reciprocal multiquadric (see [6]), which is given by

$$V(x_I, y_I, z_I) = \sum_{i=1}^{N} c_i \left(R + (x_I - x_i)^2 + (y_I - y_i)^2 + (z_I - z_i)^2\right)^{-0.5}, \quad I = 1, \ldots, N, \tag{3.10}$$

where R is a non-negative number. Obviously, an $N \times N$ linear system must be solved for the unknown coefficients c_i. This can be avoided by localizing the method. In this case, one computes several scattered data interpolants $V_k(x, y, z)$. Each interpolant V_k is computed by considering only a fixed number of given data closest to the point (x, y, z). This is done in the implementation.

Obviously, a stopping criterion for splitting tetrahedra can either relate to the overall number of tetrahedra or the overall quality of the mesh with respect to E_{avg}, i.e., splitting can terminate when either $n_t \ge n_t^{\max}$ or $E_{\text{avg}} \le \epsilon$.

Special care must be taken when splitting tetrahedra having surface points among their vertices. In this case, the insertion of new points depends on the number of surface points among the vertices and the neighbors of the particular tetrahedron being split. Case distinctions are made based on the number of surface points among the vertices. The different cases are listed in Tables 4 and 5. Table 4 characterizes the different cases, and Table 5 summarizes how to split tetrahedra containing surface points among their vertices.

Table 4. Configurations used to determine strategy for splitting (all other cases obtained by symmetry)

Case	Surface points	Configuration
1	$\mathbf{v}_1$	—
2a	$\mathbf{v}_1, \mathbf{v}_2$	Another tetrahedron shares the edge $\overline{\mathbf{v}_1\mathbf{v}_2}$.
2b	$\mathbf{v}_1, \mathbf{v}_2$	No other tetrahedron shares the edge $\overline{\mathbf{v}_1\mathbf{v}_2}$.
3a	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$	Another tetrahedron shares the face $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$.
3b	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$	No other tetrahedron shares the face $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$.
4a	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$	All four faces are shared by other tetrahedra.
4b	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$	The three faces $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$, $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4)$, and $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4)$ are shared by other tetrahedra, $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$ is not shared.
4c	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$	The two faces $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ and $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4)$ are shared by other tetrahedra, $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4)$ and $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$ are not shared.
4d	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$	The face $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$ is shared by another tetrahedron, the other three faces are not shared.
4e	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$	No face is shared by another tetrahedron.

In Table 5, $\mathbf{c}$ denotes the centroid of a tetrahedron, $\mathbf{c}_{i,j}$ the midpoint of the edge $\overline{\mathbf{v}_i\mathbf{v}_j}$, and $\mathbf{c}_{i,j,k}$ the centroid of the face with vertices $\mathbf{v}_i$, $\mathbf{v}_j$, and $\mathbf{v}_k$. Midpoints of edges (e.g., $\mathbf{c}_{1,2}$) and centroids of faces (e.g., $\mathbf{c}_{1,2,3}$) are mapped onto the given parametric surface by expressing such points in terms of barycentric coordinates and evaluating the given surface at corresponding parameter values. The chosen volume triangulations for splitting tetrahedra containing surface points among their vertices are shown in Fig. 8.

Table 5. Splitting tetrahedra containing surface points among their vertices (inserted points and resulting volume triangulation)

Case	Surface points	Inserted vertex	New tetrahedra
1	$\mathbf{v}_1$	$\mathbf{c}$	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{c})$ $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4, \mathbf{c})$ $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4, \mathbf{c})$ $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{c})$
2a	$\mathbf{v}_1, \mathbf{v}_2$	$\mathbf{c}$	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{c})$ $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4, \mathbf{c})$ $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4, \mathbf{c})$ $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{c})$
2b	$\mathbf{v}_1, \mathbf{v}_2$	$\mathbf{c}_{1,2}$	$(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4, \mathbf{c}_{1,2})$ $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{c}_{1,2})$
3a	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$	$\mathbf{c}$	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{c})$ $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4, \mathbf{c})$ $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4, \mathbf{c})$ $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{c})$
3b	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$	$\mathbf{c}_{1,2,3}$	$(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4, \mathbf{c}_{1,2,3})$ $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4, \mathbf{c}_{1,2,3})$ $(\mathbf{c}_{2,3}, \mathbf{v}_1, \mathbf{v}_4, \mathbf{c}_{1,2,3})$
4a	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$	$\mathbf{c}$	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{c})$ $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4, \mathbf{c})$ $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4, \mathbf{c})$ $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{c})$
4b	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$	$\mathbf{c}_{2,3,4}$	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4, \mathbf{c}_{2,3,4})$
4c	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$	$\mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4}$	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_1, \mathbf{v}_4, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_3, \mathbf{v}_4, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4})$
4d	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$	$\mathbf{c}_{1,2,4}, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4}$	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_1, \mathbf{v}_4, \mathbf{c}_{1,2,4}, \mathbf{c}_{1,3,4})$ $(\mathbf{v}_2, \mathbf{v}_4, \mathbf{c}_{1,2,4}, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_3, \mathbf{v}_4, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_1, \mathbf{c}_{1,2,4}, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4})$
4e	$\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4$	$\mathbf{c}_{1,2,3}, \mathbf{c}_{1,2,4}, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4}$	$(\mathbf{v}_1, \mathbf{v}_2, \mathbf{c}_{1,2,3}, \mathbf{c}_{1,2,4})$ $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{c}_{1,2,3}, \mathbf{c}_{1,3,4})$ $(\mathbf{v}_1, \mathbf{v}_4, \mathbf{c}_{1,2,4}, \mathbf{c}_{1,3,4})$ $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{c}_{1,2,3}, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_2, \mathbf{v}_4, \mathbf{c}_{1,2,4}, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_3, \mathbf{v}_4, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_1, \mathbf{c}_{1,2,3}, \mathbf{c}_{1,2,4}, \mathbf{c}_{1,3,4})$ $(\mathbf{v}_2, \mathbf{c}_{1,2,3}, \mathbf{c}_{1,2,4}, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_3, \mathbf{c}_{1,2,3}, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4})$ $(\mathbf{v}_4, \mathbf{c}_{1,2,4}, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4})$ $(\mathbf{c}_{1,2,3}, \mathbf{c}_{1,2,4}, \mathbf{c}_{1,3,4}, \mathbf{c}_{2,3,4})$

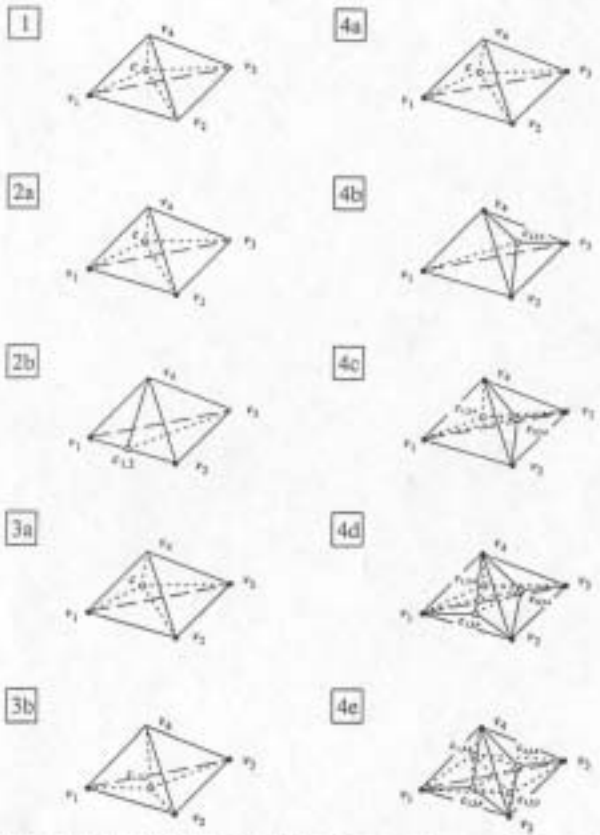

Figure 8. Splitting tetrahedra containing surface points among their vertices

4. Improving the Volume Triangulation

In order to avoid tetrahedra that have small aspect ratios D/A or small solid angles S, it is necessary to locally alter the volume triangulation. Here, the aspect ratio D/A of a tetrahedron is defined as the smallest distance D of any of the four vertex to an opposite face with area A. Considering a tetrahedron with vertices $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, and $\mathbf{v}_4$, the solid angle S at $\mathbf{v}_1$ is computed as follows: First, compute the intersection points of the three lines passing through $\mathbf{v}_1$ and having directions $(\mathbf{v}_2 - \mathbf{v}_1)$, $(\mathbf{v}_3 - \mathbf{v}_1)$, and $(\mathbf{v}_4 - \mathbf{v}_1)$ with the unit sphere having $\mathbf{v}_1$ as its center. Second, compute the (interior) angles of the spherical traingle that is defined by the three intersection points and the three great circles passing through the intersection points. Denoting these angles by α, β, and γ, the solid angle at $\mathbf{v}_1$ is given by the ratio of the surface area of the spherical triangle $(= \alpha + \beta + \gamma - \pi)$ and the surface area of the unit sphere $(= 4\pi)$.

The volume triangulation should be altered locally if this results in a better volume triangulation. Figure 9 shows two possible volume triangulations for a

volume bounded by a convex polyhedron consisting of five vertices, nine edges, and six triangles, which will be called a V5E9T6-polyhedron. The volume triangulation of this configuration is obtained by choosing one among two possible volume triangulations. The first one is defined by the two tetrahedra $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4)$ and $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_5)$, and the second one is defined by the three tetrahedra $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_4, \mathbf{v}_5)$, $(\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_5)$, and $(\mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_5)$. Among the two alternatives, the one that maximizes the minimum solid angle in the local volume triangulation is chosen.

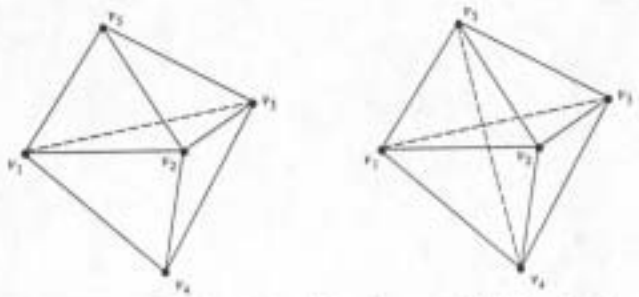

Figure 9. Possible volume triangulations of convex V5E9T6-polyhedron

Whenever the insertion of another grid point introduces such a convex V5E9T6-polyhedron, it is triangulated by either two or three tetrahedra, depending on which configuration maximizes the minimum solid angle. Thus, the edge $\overline{\mathbf{v}_4\mathbf{v}_5}$ might or might not be used in the volume triangulation. For more detail about "edge swapping" and local optimization schemes for triangulations see [2] and [11, 12].

In order to describe the procedure used in the implementation, a definition of "neighborhood" is needed. Again, a tetrahedron T_i is identified with its associated index quadruple $(v_1^i, v_2^i, v_3^i, v_4^i)$ referring to its vertices.

Definition. Given a volume triangulation, the **neighbor set** $\mathcal{N}_i$ associated with the tetrahedron $T_i = (v_1^i, v_2^i, v_3^i, v_4^i)$ is the set of all tetrahedra $T_j = (v_1^j, v_2^j, v_3^j, v_4^j)$ sharing exactly one face with T_i, i.e.,

$$\mathcal{N}_i = \bigcup \{T_j = (v_1^j, v_2^j, v_3^j, v_4^j), \quad i \neq j |$$
$$\{v_1^j, v_2^j, v_3^j\} \subset \{v_1^i, v_2^i, v_3^i, v_4^i\} \vee \{v_1^j, v_2^j, v_4^j\} \subset \{v_1^i, v_2^i, v_3^i, v_4^i\}$$
$$\vee \{v_1^j, v_3^j, v_4^j\} \subset \{v_1^i, v_2^i, v_3^i, v_4^i\} \vee \{v_2^j, v_3^j, v_4^j\} \subset \{v_1^i, v_2^i, v_3^i, v_4^i\}. \quad (4.1)$$

The tetrahedra in $\mathcal{N}_i$ are called **neighbors** of T_i.

For each tetrahedron T_i that is split into a set of new tetrahedra when inserting an additional grid point, the potential modifications of the volume triangulation

are restricted to the neighbor set $\mathcal{N}_i$ and all the tetrahedra replacing T_i. These are the steps of the optimization procedure:

(i) Determine the union of the neighbor set $\mathcal{N}_i$ and the set of all new tetrahedra replacing T_i when inserting a grid point; denote the union by $\mathcal{O}_i$.
(ii) Perform steps (iii) and (iv), while the set $\mathcal{O}_i$ contains at least one convex V5E9T6-polyhedron and replacing of edges is possible.
(iii) Construct the other possible volume triangulation for the convex V5E9T6-polyhedron and select the volume triangulation that maximizes the minimum solid angle among the two alternatives.
(iv) Update the set $\mathcal{O}_i$ if the volume triangulation has been altered.

Step (iii) constructs a locally optimal triangulation, which—in general—does not lead to a globally optimal triangulation. By allowing edge insertion/deletion that sometimes leads to a worse quality measure, one might eventually obtain a globally optimal triangulation (see [18]). It is proposed to use simulated annealing when performing edge insertion/deletion.

Since any edge insertion/deletion procedure might lead to a volume triangulation whose quality has "decreased" with respect to the measures defined in (3.1), (3.6), and (3.8), one should consider edge insertion/deletion only if the insertion of a grid point has created solid angles that are smaller than some tolerance. This is done in the implementation. Once the insertion of grid points is completed, it is possible to apply such an optimization scheme to the set of all tetrahedra.

In the following, the concept of simulated annealing is reviewed, and various optimization criteria for the construction of globally optimal volume triangulations are discussed. Simulated annealing is discussed, for example, in [18]. Globally optimal triangulations for point sets can very often be obtained by using this method. The criterion used can be any quality measure that is desirable for a triangulation. Examples in 2D are the max-min angle triangulations (Delaunay triangulation), a min-max angle triangulation, or a "data dependent" triangulation minimizing some approximation error. Simulated annealing can be incorporated into the algorithm discussed in this paper for obtaining a better—possibly globally optimal—volume triangulation with respect to both geometrical shape and desired volume of a tetrahedron. The simulated annealing algorithm has this general structure:

Algorithm 4.1. (Simulated Annealing)

Input: volume triangulation,
initial temperature t_0,
annealing schedule $\{t_1 > t_2 > t_3 > \ldots > t_n\}$,
number of swap attempts at temperature t_i (*Swap Attempts*(t_i)),
maximum number of good swaps at temperature t_i
(*MaxGoodSwaps*(t_i));

Output: volume triangulation, possibly maximizing its quality;

```
for i = 1 to n do
  for j = 1 to SwapAttempts(t_i) do
    while the number of good swaps < MaxGoodSwaps(t_i) do
    {  choose a random V5E9T6-polyhedron in the triangulation;
       if the second triangulation of the V5E9T6-polyhedron is possible
         construct the second triangulation;
         compute q = QualityOfGivenTriangulationOfV5E9T6Polyhedron
                     - QualityOfSecondTriangulationOfV5E9T6Polyhedron;
         if the second triangulation improves the quality (i.e., q < 0)
           perform the swap and update the triangulation;
         else
         {  compute a random number 0 ≤ r ≤ 1;
            if f ≤ exp(-q/t_i)
              perform the swap and update the triangulation;
         }
  }
```

Initial results suggest that the following parameter choices lead to good triangulation: $t_0 = 2 \max\{|q_i|\}$, where the q_i values correspond to quality-improving swaps and are obtained by applying a local swap algorithm to the initial volume triangulation, $t_i = 0.95^i\, t_0$, $n = 10$, $SwapAttempts(t_i) = 5\ NumberOfFacesInTriangulation$, and $MaxGoodSwaps(t_i) = 5\ NumberOfFacesInTriangulation$. In practice, it is ensured that the absolute values of the quality differences (the $|q_i|$ values) and the temperatures t_i lie in the interval $[0,1]$.

One can choose from an endless list of quality criteria for a volume triangulation. The ones that are important in the context of grid/mesh generation are

- a **volume** criterion that relates the tetrahedral volume to aboslute surface curvature and shortest surface distance,
- an **aspect ratio** criterion that relates the aspect ratio (i.e., height of a tetrahedron measured in surface normal direction versus length of a tetrahedron measured parallel to the surface) to absolute surface curvature and shortest surface distance,
- a **vertex degree** criterion that considers the square variance of vertex degrees in the triangulation, and
- a **shape** criterion that considers geometrical characteristics of a tetrahedron independently of the given geometry (solid angle, ratio of a radii of inscribed sphere and circumsphere, etc.).

It is possible to combine these criteria into a single criterion by means of convex combinations. This is outlined in the following paragraphs.

The desired volume V of a tetrahedron is defined as a function of absolute surface curvature κ and absolute shortest distance d between the tetrahedron's centroid and the surface. The volume should decrease with increasing curvature and should increase with increasing distance. This behavior can be modeled by a sum of an exponential and a power function. The function is

$$V(\kappa,d) = (v_{0,0} - v_{\infty,0})\exp(-\alpha\kappa) + \frac{v_{0,D} - v_{0,0}}{D^{\beta}} d^{\beta} + v_{\infty,0},$$
$$\kappa \in [0,\infty),\ d \in [0,D], \tag{4.2}$$

where D is a maximal positive distance, $v_{0,0} = V(0,0)$, $v_{\infty,0} = V(\infty,0)$, and $v_{0,D} = V(0,D)$. The parameters α and β control the degree of change in volume with increasing absolute surface curvature and distance.

Defining $v_{\min} = \min\{v(0,0),\ v(\infty,0),\ v(0,D),\ v(\infty,D)\}$ and $v_{\min} = \max\{v(0,0), v(\infty,0), v(0,D), v(\infty,D)\}$, the normalized desired volume is given by

$$V_{\text{norm}}(\kappa,d) = \frac{V(\kappa,d) - v_{\min}}{v_{\max} - v_{\min}}. \tag{4.3}$$

Similarly, the desired aspect ratio R of a tetrahedron is defined as a function of absolute surface curvature and absolute shortest distance. The aspect ratio should increase with increasing curvature and increasing distance. Again, this can be modeled by the function

$$R(\kappa,d) = (r_{0,0} - r_{\infty,0})\exp(-\gamma\kappa) + \frac{r_{0,D} - r_{0,0}}{D^{\delta}} d^{\delta} + r_{\infty,0},$$
$$\kappa \in [0,\infty),\ d \in [0,D], \tag{4.4}$$

where $r_{0,0} = R(0,0$, $r_{\infty,0} = R(\infty,0)$, and $r_{0,D} = R(0,D)$. The parameters γ and δ control the degree of change in aspect ratio with increasing curvature and distance. Defining $r_{\min} = \min\{r(0,0),\ r(\infty,0),\ r(0,D),\ r(\infty,D)\}$ and $r_{\max} = \max\{r(0,0), r(\infty,0), r(0,D), r(\infty,D)\}$, the normalized desired aspect ratio is given by

$$R_{\text{norm}}(\kappa,d) = \frac{R(\kappa,d) - r_{\min}}{r_{\max} - r_{\min}}. \tag{4.5}$$

From a topological/connectivity point of view it is desirable to have a triangulation whose variance of *vertex degrees* is minimal—the degree of a vertex being the number of edges sharing the vertex. This can be achieved by minimizing the square of the variance of the vertex degrees. The square of the variance of the vertex degrees is

$$\sigma^2 = \sum_{i=1}^{n} (\deg(v_i) - \mu)^2, \tag{4.6}$$

where n is the number of vertices, $\deg(v_i)$ denotes the degree of vertex v_i, and $\mu = 1/n\sum_{i=1}^{n} \deg(v_i)$.

The square variance of the vertex degrees is normalized by dividing by the maximal square variance, denoted by $\sigma^2_{\max}$, i.e.,

$$\sigma^2_{\text{norm}} = \frac{\sigma^2}{\sigma^2_{\max}}, \tag{4.7}$$

where $\sigma^2_{\max}$ is initially obtained from the given volume triangulation. It is possible that $\sigma^2_{\max}$ must be updated during the execution of the simulated annealing algorithm. This must be done when a swap leads to a value greater than 1 for the actual square variance of vertex degrees in a triangulation. The value of 0 is associated with the best normalized square variance, and the value of 1 is associated with the worst normalized square variance.

Assuming than an appropriate shape measure $S \in [0,1]$ can be associated with the geometrical characteristics of a tetrahedron (0 corresponding to worst shape and 1 corresponding to best shape), the four functions to be minimized are

- $|V_{\text{norm}}(\kappa, d) - V_{\text{act}}(\kappa, d)|$,
- $|R_{\text{norm}}(\kappa, d) - R_{\text{act}}(\kappa, d)|$,
- σ^2_{act}, and
- $|1 - S_{\text{act}}|$.

Here, $V_{\text{act}}(\kappa, d)$, $R_{\text{act}}(\kappa, d)$, σ^2_{act}, and S_{act} refer to the actual, normalized values of the tetrahedral volume, the tetrahedral aspect ratio, the tetrahedral shape, and the square variance of vertex degrees in the triangulation.

These four functions can be combined into a single function by means of a convex combination. Thus, the "quality function" $q \in [0,1]$ to be maximized is

$$\begin{aligned} q(\kappa, d) = 1 &- \omega_v |V_{\text{norm}}(\kappa, d) - V_{\text{act}}(\kappa, d)| - \omega_r |R_{\text{norm}}(\kappa, d) - R_{\text{act}}(\kappa, d)| \\ &- \omega_d \sigma^2_{\text{act}} - \omega_s |1 - S_{\text{act}}|, \end{aligned} \tag{4.8}$$

where ω_v, ω_r, ω_d, $\omega_s \geq 0$ and $\omega_v + \omega_r + \omega_d + \omega_s = 1$. By choosing different weights ω_v, ω_r, ω_d, and ω_s it is possible to emphasize the volume, the aspect ratio, the square variance of the vertex degrees, or the shape of tetrahedra.

Once one has identified a convex V5E9T6-polyhedron it must be determined what the "qualities" of the two possible triangulations are. The quality of the triangulation consisting of two tetrahedra is defined as

$$q^{(2)}(\kappa, d) = \tfrac{1}{2}(q_1(\kappa, d) + q_2(\kappa, d)), \tag{4.9}$$

and the quality of the triangulation consisting of three tetrahedra is defined as

$$q^{(3)}(\kappa, d) = \tfrac{1}{3}(q_1(\kappa, d) + q_2(\kappa, d) + q_3(\kappa, d)), \tag{4.10}$$

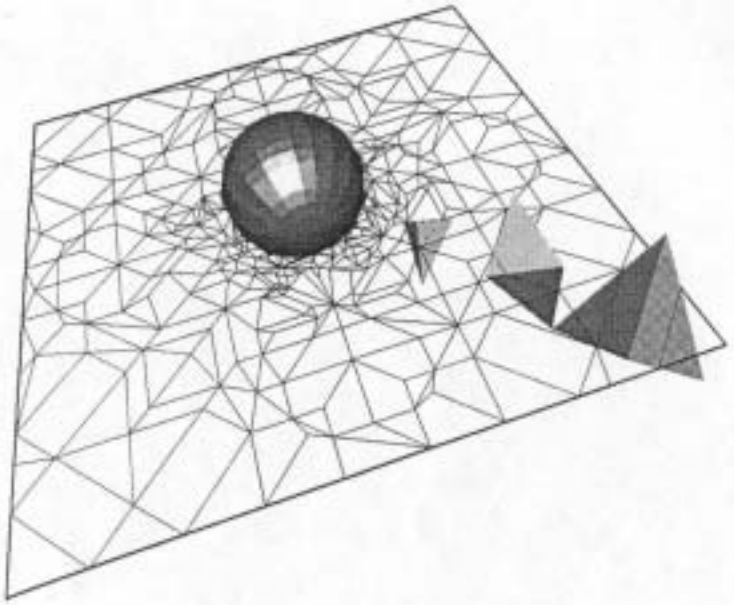

Figure 10. Unstructured grid around sphere

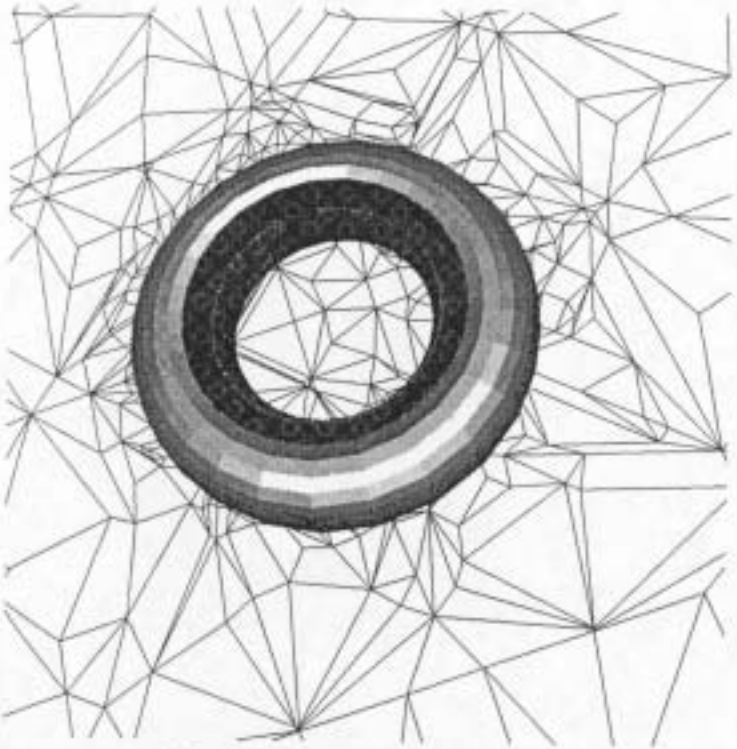

Figure 11. Unstructured grid around torus

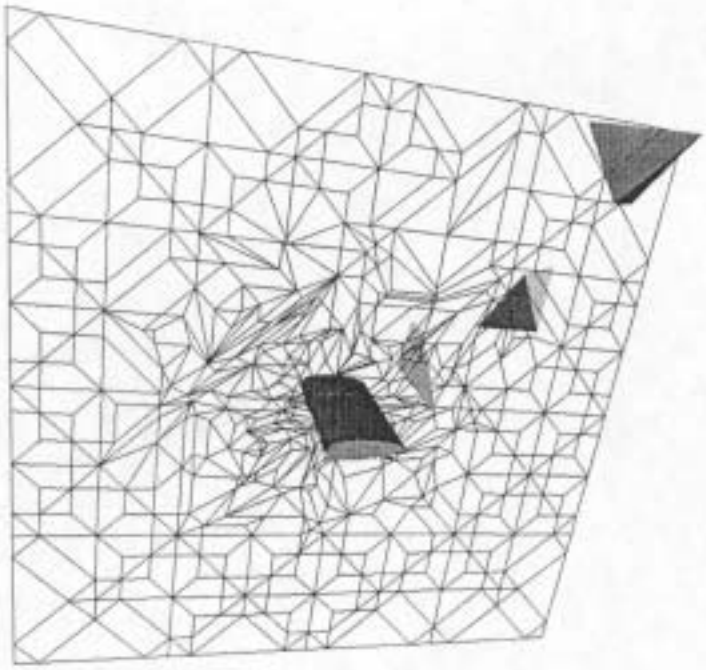

Figure 12. Unstructured grid around wing

where $q_j(\kappa, d)$ is the value of (4.8) for one tetrahedron in the triangulation of the V5E9T6-polyhedron.

If the value of $q^{(3)}(\kappa, d)$ is greater than (smaller than or equal to) the value of $q^{(2)}(\kappa, d)$, then changing a given triangulation consisting of two tetrahedra to a triangulation consisting of three tetrahedra is a quality-improving (quality-deteriorating) swap. If the value of $q^{(2)}(\delta, d)$ is greater than (smaller than or equal to) the value of $q^{(3)}(\kappa, d)$, then changing a given triangulation consisting of three tetrahedra to a triangulation consisting of two tetrahedra is a quality-improving (quality-deteriorating) swap.

5. Examples

The method has been tested for parametrically defined closed surfaces (Figs. 10–12) and for a closed surface triangulation without known parametric representation (Fig. 13). The surface triangulation without known parametric representation is an isosurface of a computerized axial tomography (CAT) data set. In all four examples, grid points have been inserted by considering distance to the surface, but not considering surface curvature. The volume triangulations have been optimized locally by maximizing the minimum aspect ratio D/A after point insertion (see Section 4). The volume triangulations are intersected with cutting planes, and the intersections of the tetrahedra and the cutting planes (triangles and quadrilaterals) are shown. Certain tetrahedra have been made visible to illustrate their increasing/decreasing sizes.

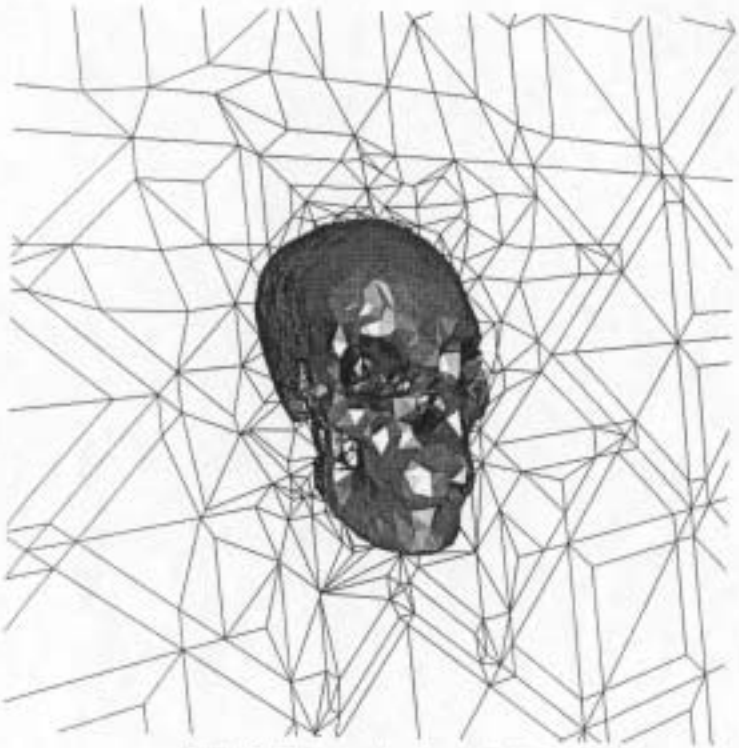

Figure 13. Unstructured grid around skull

6. Conclusions

A completely automatic algorithm for constructing unstructured volume grids has been presented. The method is computationally fairly involved, but the test results confirm the validity of the approach. The fact that no user interaction is required during the entire grid generation process makes this method useful for complicated closed geometries. It is planned to improve the method by reducing computing time and storage requirement.

Acknowledgements

This work was supported by the National Science Foundation (Research Initiation Award 1992) under contract ASC-9210439 to Mississippi State University and by the National Grid Project consortium. Special thanks go to the members on the National Grid Project team, in particular to Joe F. Thompson and Nigel P. Weatherill.

References

[1] Baker, T. J.: Three-dimensional mesh generation by triangulation of arbitrary point sets. AIAA-87-1124-CP, 255–271 (1987).
[2] Barth, T. J., Wiltberger, N. L., Gandhi, A. S.: Three-dimensional unstructured grid generation via incremental insertion and local optimization. NASA-CP-3143, pp. 449–461 (1992).
[3] Castillo, J. E.: Mathematical aspects of numerical grid generation. Philadelphia: SIAM 1991.
[4] Choi, B. K., Shin, H. Y.,Yoon, Y. I., Lee, J. W.: Triangulation of scattered data in 3D space. CAD *202*, 239–48 (1988).
[5] Farin, G.: Curves and surfaces for computer aided geometric design, 3rd ed. San Diego: Academic Press 1993.
[6] Franke, R.: Scattered data interpolation: Tests of some methods. Math. Comp. *38*, 181–200 (1982).
[7] George, P. L.: Automation mesh generation. New York: Wiley & Sons 1991.
[8] Hamann, B.: Curvature approximation for triangulated surfaces. In: Geometric modelling (Farin, G., Hagen, H., Noltemeier, H., eds.), pp. 139–153. Wien New York: Springer 1993.
[9] Hamann, B.: A data reduction scheme for triangulated surfaces. Comput. Aided Geom. Des. *11*, 197–214 (1994).
[10] Holmes, G., Snyder, D.: The generation of unstructured triangular meshes using Delaunay triangulation. In: Numerical grid generation in computational fluid mechanics (Sengupta, S., Häuser, J., Eiseman, P.R., Taylor, C., eds). Swansea: Pineridge Press 1988.
[11] Lawson, C. L.: Software for C^1 surface interpolation. In: Mathematical software III (Rice, J. R., ed.), pp. 161–194. San Diego: Academic Press 1977.
[12] Lawson, C. L.: Properties of *n*-dimensional triangulations. Comput. Aided Geom. Des. *3*, 231–246 (1986).
[13] Löhner, R., Parikh, P.: Three-dimensional grid generation by the advancing front method. Numer. Meth. Fluids *8*, 1135–1149 (1988).
[14] Lorensen, W. E., Cline, H. E.: Marching cubes: A high resolution 3D surface construction algorithm. Comput. Graphics *21*, 163–169 (1987).
[15] Nielson, G. M., Hamann, B.: The asymptotic decider: Resolving the ambiguity in marching cubes. In: Visualisation '91 (Nielson, G. M., Rosenblum, L. J., eds.), pp. 83–91. Los Alamitos: IEEE Computer Society Press 1991.
[16] Parikh, P., Pirzadeh, S.: Recent advances in unstructured grid generation—Program VGRID3D. NASA-CP-3143, pp. 435–448 (1992).
[17] Preparata, F. P., Shamos, M. I.: Computational geometry. New York: Springer 1990.
[18] Schumaker, L. L.: Computing optimal triangulations using simulated annealing. Comput. Aided Geom. Des. *10*, 329–345 (1993).
[19] Thompson, J. F., Warsi, Z. U. A., Mastin, C. W.: Numerical grid generation. New York: North-Holland 1985.
[20] Thompson, J. F., Weatherill, N. P.: Aspects of numerical grid generation: Current science and art. Proceedings of the 11th AIAA Applied Aerodynamics Conference, Monterey, CA, August 1993.
[21] Weatherill, N. P.: The generation of unstructured grids using Dirichlet tessellations, Princeton University, MAE Report 1715 (1985).
[22] Weatherill, N. P.: A method for generating irregular computational grids in multiply connected planar domains. Int. J. Numer. Methods Fluids *8*, 181–197 (1988).
[23] Weatherill, N. P.: The Delaunay triangulation in CFD. Comput. Math. Appl. *24*, 129–150 (1992).

[24] Weatherill, N. P., Hassan, O., Marcum., D. L., Marchant, M. J.: Grid generation by the Delaunay triangulation. Von Karman Institute for Fluid Dynamics, 1993–1994 Lecture Series (1994).

B. Hamann
Department of Computer Science and
NSF Engineering Research Center
for Computational Field Simulation
Mississippi State University
P.O. Box 6176
Mississippi State
MS 39762
USA

H. J. Thornburg
NSF Engineering Research Center
for Computational Field Simulation
Mississippi State University
P.O. Box 6176
Mississippi State
MS 39762
USA

G. Hong
SL Corporation Suite 110, Hunt Plaza
240 Tamal Vista Blvd.
Corte Madera, CA 94925
USA