INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2003; **56**:1355–1373 (DOI: 10.1002/nme.616)

Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations

Qiang Du[1,2,*,†] and Desheng Wang[2]

[1]*Department of Mathematics, Penn State University, University Park, PA 16802, U.S.A.*
[2]*Lab for Scientific and Engineering Computing, Academy of Sciences, Beijing, China*

SUMMARY

The centroidal Voronoi tessellation based Delaunay triangulation (CVDT) provides an optimal distribution of generating points with respect to a given density function and accordingly generates a high-quality mesh. In this paper, we discuss algorithms for the construction of the constrained CVDT from an initial Delaunay tetrahedral mesh of a three-dimensional domain. By establishing an appropriate relationship between the density function and the specified sizing field and applying the Lloyd's iteration, the constrained CVDT mesh is obtained as a natural global optimization of the initial mesh. Simple local operations such as edges/faces flippings are also used to further improve the CVDT mesh. Several complex meshing examples and their element quality statistics are presented to demonstrate the effectiveness and efficiency of the proposed mesh generation and optimization method. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: centroidal Voronoi Delaunay triangulation; tetrahedral mesh generation; optimization; mesh quality

1. INTRODUCTION

In many scientific and engineering problems ranging from fluid flows to structural analysis, mesh generation often forms a crucial part of the numerical solution procedure. Automatic unstructured tetrahedral mesh generations for complex three dimensional (3D) domains have proved to be very successful tools for the efficient solution of complex application problems. The Advancing Front technique (AFT) [1–4], Octree methods [5] and Voronoi Delaunay-based methods [6–15] are some of the well studied techniques in unstructured mesh generation. Among them, due to the rigorous mathematical theory and systematic implementation, Delaunay-based tetrahedral meshing methods have gained a lot of popularity. Still, this type of

*Correspondence to: Qiang Du, Department of Mathematics, Penn State University, University Park, PA 16802, U.S.A.
†E-mail: qdu@math.psu.edu

Contract/grant sponsor: China State Major Basic Research Project and US NSF; contract/grant number: G199903280 and CCR 9988303

Received 26 November 2001
Revised 21 January 2002
Accepted 16 May 2002

methods cannot eliminate the existence of sliver elements in the mesh, that is, highly distorted tetrahedra (slivers) with the degree of dihedral angles as low as 0.01 or as high as 179.99 are often encountered [16–18]. Such sliver elements affect the accuracy of the numerical solution in the application; thus, for the construction of the initial mesh, further mesh optimization is usually required. In fact, tremendous efforts have been made to achieve better mesh quality through mesh improvement and optimization.

The existing algorithms for tetrahedral mesh improvements fall into three basic categories [15–22]:

1. Geometric optimization, also called mesh smoothing: the vertices are repositioned to improve mesh quality without changing mesh topology (i.e. node connectivity). Several smoothing strategies exist, such as optimal Laplacian smoothing and its variants.
2. Topological optimization, including local reconnection: edges or faces are swapped to locally change nodes connectivity for a given set of vertices.
3. Vertex insertion and deletion: vertices, including those near the boundary, are added or deleted to optimize the mesh through refining or coarsening.

Many of the above methods were often efficiently implemented through local operations, and they are usually combined in an iterative fashion to enhance the mesh quality. However, due to their local nature, they may not offer significant improvement globally. In practice, the global features of the overall quality of the mesh include the mesh conformity with a specified sizing field which generally refers to the function that, at any vertex, gives the length of the edges (or equivalently the parameters of elements) connecting the vertex. In this sense, with only local operations, the improved mesh is not going to be optimal. Most experiments show that, even after combined local optimization operations, some less-ideal elements still remain in the mesh. Such difficulties in high-quality mesh generation motivate the application of global optimization methods which can improve both the mesh quality and the conformity with the given sizing field.

Centroidal Voronoi tessellations (CVT) are Voronoi tessellations of a region such that the generating points of the tessellations are also the mass centroids of the corresponding Voronoi regions with respect to a given density function [23]. CVT are used in diverse applications, including data compression, clustering analysis, cell biology, territorial behaviour of animals and optimal allocation of resources [23, 24]. The dual data structure of CVT is the corresponding Delaunay triangulation called centroidal Voronoi–Delaunay triangulation (CVDT). For an arbitrary domain, if we specify some geometric constraints, such as fixed boundary structure which may include vertices or edges that are not allowed to change, we get the Constrained CVDT (or CCVDT). In Reference [25], the two dimensional CVDTs are constructed based on CVTs with some prescribed density functions. They result in high quality triangular meshes for 2D domains which produce very good results when model partial differential equations are solved using these meshes. The quality of the Delaunay-based triangular mesh is often greatly affected by the placement of the generating points [6, 7, 15], though the point placement is not the unique factor that needs to be addressed in order to produce a good mesh. The results in Reference [25] indicate that CVDT provides an optimal distribution of generating points that leads to high quality meshes. Hence, CVDT can be used as a good alternative for generating high quality meshes that satisfy the desired specification in sizing field or local refinement requirements. This is largely due to the property that the distribution of vertices from the

CVDTs construction conforms with the given density function globally [23]. Thus, if Lloyd's iteration [26] is applied to an initial Delaunay tetrahedral mesh to construct a CCVDT of a 3D domain, the final 3D triangulation may be naturally viewed as an optimization of the initial mesh. In this paper, a systematic implementation of such a strategy is investigated.

To briefly describe the main ingredients, we start with the construction of an initial Delaunay tetrahedral mesh for the given domain. First, a surface triangular mesh is taken as the input. Then, we begin the construction of the volume mesh which consists of two parts: conforming boundary tetrahedronization and interior refinement through Delaunay points insertion. The boundary tetrahedronization is stored for later use and a sizing field $H_B(p)$ (p being a point in the domain) is selected which can either come from the user input or from an interpolation scheme based on the boundary tetrahedral mesh whose vertices have sizing values. The interior refinement is divided into two parts: interior points generation based on a division of interior edges using algebraic or geometric propagation and neighbouring grid filtering [10], and point insertion into the existing mesh based on the classical constrained Delaunay insertion procedure [8, 9]. With the initial mesh, an appropriate density function is constructed from the sizing field. The 3D construction of the CCVDT is then underway with the application of the Lloyd iteration: the Voronoi regions (i.e. Voronoi polyhedra) of the interior vertices (which are allowed to move) are computed from the Delaunay tetrahedronization and the mass centres of these Voronoi regions are computed; then, these mass centres are inserted (as new vertices) into the mesh to replace the original generators. If any generator is close (in some measure or by certain criterion) to the boundary, we project it onto the boundary surface and merge the projection point with the closest vertex on the boundary triangular mesh to the projection of their midpoint on the boundary. The above two procedures, namely, the computation of mass centres and the Delaunay insertion, are iterated alternatively until convergence. The resulting CCVDT is expected to be a significantly improved tetrahedral mesh and it is in more harmony with the sizing field. The mesh vertices are optimally positioned and the mesh's overall structure is improved. This produces dramatic enhancement of element quality as almost all the slivers existing in the initial mesh are removed after the iteration. For further improvement, we apply simple local operations such as edges/faces flippings to the few remaining bad-shaped elements. In the end, a high-quality tetrahedral mesh is obtained as the output.

The remainder of the article is organized as follows. In Section 2, we briefly describe the generation of the initial constrained Delaunay tetrahedral mesh and the derivation of the sizing field. In Section 3, we present some basic concepts and results concerning CVT, CVDT and CCVDT. And in Section 4, a detailed description for constructing the three dimensional CCVDT is provided. In Section 5, local optimization techniques are discussed for further improvement of the mesh. In Section 6, some application examples and the mesh quality statistics are presented. Finally, a few concluding remarks are given in Section 7.

2. INITIAL DELAUNAY TETRAHEDRONIZATION

For a given 3D domain, we use the classical Delaunay-based methods to generate the initial Delaunay tetrahedronization. Here, a surface triangular mesh of the domain boundary is taken as the input. The tetrahedronization consists of two steps: the conforming boundary tetrahedronization; and the interior Delaunay refinement.

2.1. Conforming boundary tetrahedronization

The input surface triangular mesh of the boundary can be generated using various methods, including parametric mapping, paving, advancing front, and parametric space meshing and mapping [6, 27–31]. Each surface mesh point contains an element sizing value $H_B(p)$, (p being a boundary point) which defines the approximate length of the edges (or equivalently the elements) emanating from p. These sizing values will be used to construct the sizing field of the entire domain.

For the conforming boundary tetrahedronization, we first make an unconstrained Delaunay tetrahedronization of all the surface mesh points (occasionally including some prescribed interior points) by the classic incremental Delaunay insertion procedure. There are three steps involved: the construction of *base*, *cavity* and *balls* [8, 9]. This tetrahedronization forms the triangulation of the convex hull of the boundary points and it usually does not match the surface mesh or even the geometry of the domain, i.e. the boundary integrity is not preserved [10–14]. To reconstruct the missing elements of the surface mesh in a conforming manner, a mixture of boundary recovery techniques is used [13]. The local operations such as edges/faces flippings are first used to recover the missing edges/faces for the constrained setting [10, 11] and subsequently, the edges/faces splittings [12, 13] technique are used for the Delaunay refinement. A conforming boundary tetrahedronization is completed after the boundary recovery.

With the conforming boundary tetrahedronization, we derive a sizing field $H(p)$ for any point p of the domain (as the measurement of the approximate average length of the edges emanating from it) through either interpolation or user specification. Together with the boundary tetrahedronization, they form the control space [7] of the tetrahedra mesh generation. The user specified sizing field could also be based on either *a priori* known information or *a posteriori* local error estimation.

2.2. Interior Delaunay refinement

The vertices of the boundary tetrahedronization include only the boundary mesh points and possibly some added Steiner points. The partially constructed mesh obviously needs interior refinement. The refinement is divided into two parts: interior points generation and point insertion into the existing mesh.

For the interior points generation, we use a simple method proposed in Reference [10]: division of interior edges using algebraic or geometric propagation and neighbouring grid filtering. Other generation methods may be considered, including the method of advancing Front and sphere packing. Once the points are generated, we apply the classical constrained version of the incremental Delaunay insertion procedure to successively insert these points into the existing mesh. Then, the above process of points generation and insertion is iterated until no interior points can be generated which signals the end of the initial Delaunay tetrahedronization.

The initial generated tetrahedral mesh usually has a substantial portion of badly shaped elements, like slivers, which need improvements. This is due to many reasons, for example, the points placement may not be optimal when using the interior edges subdivision method and there are also inherent problems of 3D Delaunay insertion procedure [14]. Even using the AFT points generation [7], some less-ideal elements still exist in the mesh. In the following sections, we present a global optimization method for the removal of slivers through the

construction of constrained centroidal Voronoi–Delaunay-based triangulation (CVDT) so that the overall quality of the mesh may be greatly enhanced.

For reference, we here give a definition of the quality of a tetrahedron. Let K be a tetrahedron, its quality is given by

$$Q(K) = \alpha \frac{\rho}{h_{\max}}$$

where $h_{\max}$ is the diameter of K (length of the longest edge of K), ρ the radius of the inscribed sphere and $\alpha = 2\sqrt{6}$ is a normalization factor such that the quality of an equilateral tetrahedron is 1. This quality function varies in the interval $(0, 1]$ (a well-shaped tetrahedron has a quality close to 1, while an ill-shaped element has a quality close to 0). Obviously, much of our work can be used in conjunction with many other quality measures for tetrahedral meshing [32]. The particular measure chosen here does not greatly affect the outcome of the meshing algorithm.

3. CVT, CVDT AND CCVDT

Given a set of input points $\{z_i\}_1^k$ belonging to a domain $\Omega \subset R^N$, the *Voronoi* region $\hat{V}_i$ corresponding to the point z_i consists of all points in Ω that are closer to z_i than to any other point in the set. The set $\{\hat{V}_i\}_1^k$ forms a partition of Ω and is known as a *Voronoi tessellation* or *Voronoi diagram* of Ω. The points $\{z_i\}_1^k$ are called *generating points* or *generators*. The dual *Delaunay triangulation* is formed by connecting pairs of generating points which correspond to adjacent Voronoi regions [33, 34].

Recently, the centroidal Voronoi tessellation (CVT) and a wide range of its applications have been studied in Reference [23]. It has been subsequently applied to 2D grid generation and optimization [25]. The CVT also provides an effective algorithms for nodal generation in some meshless methods [24]. In the following, we recall some of the relevant concepts and results of CVT given in Reference [25].

3.1. Basic concepts

Given a density function ρ defined on a region V, the *mass centroid* $\mathbf{z}^*$ of V is defined by

$$\mathbf{z}^* = \frac{\int_v \mathbf{y}\rho(\mathbf{y})\,\mathrm{d}\mathbf{y}}{\int_v \rho(\mathbf{y})\,\mathrm{d}\mathbf{y}}$$

Thus, given k points $\mathbf{Z}_i$, $i = 1, \ldots, k$, in the domain Ω, we can define their associated Voronoi regions V_i, $i = 1, \ldots, k$, which form a tessellation of Ω. On the other hand, given the regions V_i, $i = 1, \ldots, k$, we can define their mass centroids $\mathbf{Z}_i^*$, $i = 1, \ldots, k$.

Definition 3.1 (Du *et al.* [23, 25])
Given the set of points $\{\mathbf{z}_i\}_1^k$ in the domain Ω and a positive density function ρ defined on Ω, a Voronoi tessellation is a *centroidal Voronoi tessellation* (CVT) if $\mathbf{z}_i = \mathbf{z}_i^*$, $i = 1, \ldots, k$, i.e. the generators of the Voronoi regions V_i, $\mathbf{z}_i$, are themselves the mass centroids of those

regions. The dual Delaunay triangulation is referred to as the *Centroidal Voronoi–Delaunay triangulation* (CVDT).

For any tessellation $\{V_i\}_1^k$ of the domain Ω and a set of points $\{z_i\}_1^k$ (independent of $\{V_i\}_1^k$) in Ω, we can define the following *cost* (or *error* or *energy*) functional:

$$F(\{V_i\}_1^k, \{z_i\}_1^k) = \sum_{i=1}^{k} \int_{V_i} \rho(x)\|x - z_i\|^2 \, \mathrm{d}x$$

The standard CVTs along with their generators are critical points of this cost functional. Using the concept of cost functional, we also have:

Definition 3.2
Given the set of points $\{\mathbf{z}_i\}_1^k$ in Ω a density function ρ, and a constraint set P, a Voronoi tessellation is called a *constrained centroidal Voronoi tessellation* (CCVT) if $\{\{V_i\}_1^k, \{z_i\}_1^k\}$ minimizes the problem:

$$\min_{\{z_i\}_1^k \in P, \{V_i\}_1^k} F(\{V_i\}_1^k, \{z_i\}_1^k) = \sum_{i=1}^{k} \int_{V_i} \rho(x)\|x - z_i\|^2 \, \mathrm{d}x$$

The dual Delaunay triangulation is referred to as the *constrained centroidal Voronoi Delaunay triangulation* (CCVDT).

For constrained CVTs on general surfaces, we refer to Reference [35] for further studies. In the case of constrained Delaunay meshing, the above set P can be confined to a set that may include fixed boundary vertices, edges or even faces (in 3D).

3.2. Algorithms for CVDT and CCVDT

Constructing CVDT involves the calculation of the generators of the CVT and the construction of the corresponding Delaunay triangulation based on those generators. The latter construction step can be accomplished using standard Delaunay triangulation algorithms. Hence, the key part of CVDTs generation lies in finding the generators. For this, there are two kinds of methods: probabilistic and deterministic methods [23, 36]. Here, we just describe one particular algorithm for our use. The algorithm is based on the popular Lloyd's method [26] which is an obvious iteration between constructing Voronoi tessellations and centroids. Given a set Ω, a positive integer k, and a probability density function ρ defined on $\bar{\Omega}$, and a constraint set P:

0. select an initial set of k points $\{\mathbf{z}_i\}_1^k$ belonging to P.
1. construct the Voronoi tessellation $\{V_i\}_1^k$ of Ω associated with the points $\{\mathbf{z}_i\}_1^k$;
2. compute, in the set P, the minimum of the functional

$$G(\{\mathbf{z}_i\}_1^k) = \sum_{i=1}^{k} \int_{V_i} \rho(x)\|x - z_i\|^2 \, \mathrm{d}x$$

   the minimizer is the new set of points $\{\mathbf{z}_i\}_1^k$;
3. if this new set of points meets some convergence criterion, find the corresponding Delaunay triangulation, terminate; otherwise, return to step 1.

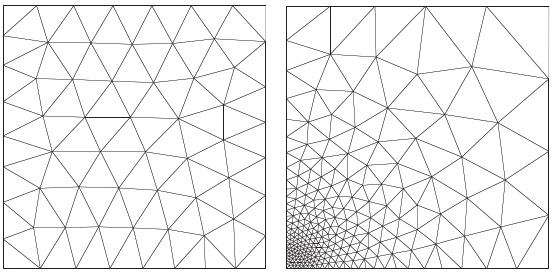

Figure 1. CVDT mesh on a two dimensional square for some given densities.

The above algorithm enjoys the property that the functional F is monotonically decreasing throughout the iteration. When the set P is simply taken to be Ω and no additional requirement is made, it reduces to the standard Lloyd's iteration [26].

3.3. Application to mesh generation

One useful application of CVDT (CCVDT) is in mesh generation, including unconstrained and constrained mesh generation. A couple of examples on the two dimensional CVT-based triangular meshes on a square domain with a uniform and a non-uniform density distribution of vertices are given in Figure 1. Numerical solutions of some model PDEs demonstrate that these meshes, produced by properly chosen density functions, can provide more accurate solutions compared to other structured or unstructured meshes [25].

For unconstrained mesh generation, the construction of CVDT deals mainly with the distribution of the generators according to some given density function ρ. Such a function ρ is used to reflect the properties of the underlying solution to be calculated on the mesh. Its selection can either be based on an a priori estimate or on some posteriori error estimates [37].

3.4. Geometric constraints due to the boundary

For constrained mesh generation, several different approaches were proposed in Reference [25] to handle the geometrical boundary constraints of the given domain in the construction of CCVDT:

1. From the boundary to the interior: a subset of generating points on the boundary is predetermined.
2. From the interior to the boundary: construct the CVT and CVDT without applying any constraints using a standard algorithm such as the Lloyd method and during the construction process, for those Voronoi regions that extend to the exterior of the domain, their corresponding generators are projected to the boundary.
3. Variational formulation: to formulate a general variational approach which covers both of the above. For details, please refer to Reference [25].

The example given in Figure 1 is constructed based on the second approach [25], the densities used in the calculation are different with one being uniform while the other being related to the derivatives of the PDE solution.

3.5. The Lloyd iteration and the mesh smoothing

The construction of CVDT (or CCVDT) through the Lloyd iteration can be viewed from a different angle as a smoothing process of an initial mesh. The CVDT concept provides a good theoretical explanation to the smoothing process: by successively moving generators to the mass centres (of the Voronoi regions), the cost functional is reduced. Here, *smoothing* means both node-movement and node reconnection. With a suitable choice of the density function, the cost functional can be related to error estimators for the underlying problem. Since the averages of the neighbouring generators provide approximation to the mass centres of the Voronoi regions, such smoothing process mimics the process of iteratively constructing CVTs such as the Lloyd's algorithm, and thus, contributes to reduction of the discretization error.

3.6. Equidistribution of cost

In Reference [23], interesting properties concerning the distribution of the cost functional were presented. For example, it was shown that in the one dimensional case,

$$\int_{V_i} \rho(x)(x - x_i)^2 \, \mathrm{d}x \approx c \quad \forall i$$

for some constant c when the number of generators is large. This means, asymptotically speaking, the error or cost is equally distributed in the Voronoi intervals [23].

For the multidimensional CVT, a conjecture has been made [38] which states that asymptotically, as the number of generators becomes large, all Voronoi regions are approximately congruent to the same basic cell that only depends on the dimension. The basic cell was shown to be the regular hexagon in two dimensions, which tells us that the CVDT provides high quality mesh. The conjecture remains open for three and higher dimensions [38], but its validity has been verified through extensive numerical studies and it is widely used in practical applications such as in the area of vector quantizations [39]. For mesh generations, it is also a good and practical strategy to construct a similar equi-distribution of error principle based on the conjecture which in turn provides an appropriate relation between the density function and the sizing field $H(p)$. The sizing field can thus be directly related to the error estimators of the solutions while the corresponding costs or error functionals are assumed to be related implicitly to the distortion of the elements quality. Also, based on the numerical evidence on the validity of Gersho conjecture in 3D, the construction of a three dimensional CCVDT, like in 2D, would produce an optimal high-quality tetrahedral mesh in the end. Such an assertion is indeed numerically supported by our various meshing examples. In the next section, we will discuss how to construct this three dimensional CCVDT using the Lloyd method which effectively turns the construction into a natural global optimization for the initial Delaunay tetrahedral mesh generated in Section 2.

4. THE CONSTRUCTION OF 3D CONSTRAINED CVDT

Since our goal is to improve the global quality of the initial constrained Delaunay tetrahedral mesh both geometrically and topologically, Lloyd's iteration to be used starts from the initial Delaunay mesh and the whole process is as follows:

0. Construct an initial conforming or constrained Delaunay tetrahedronization of a given 3D domain (actually the input being a surface triangular mesh) under a given sizing field $H(p)$. Here, $H(p)$ is derived from the boundary points' sizing. And also store all the data of the conforming boundary tetrahedronization.
1. Construct the Voronoi region for each of interior points that are allowed to change their positions, and construct the mass centre of the Voronoi region with a properly defined density function ρ derived for the sizing field $H(p)$.
2. Insert the computed mass centres into the conforming Delaunay tetrahedronization in the constrained Delaunay insertion procedure.
3. Compute the difference $D = \sum_{i=1}^{k} \|P_i - P_{\text{imc}}\|^2$, $\{P_i\}$ is the set of interior points allowed to change, $\{P_{\text{imc}}\}$ is the set of corresponding computed mass centre.
4. If D is less than a given tolerance, terminate; otherwise, return to step 1.

There are a few issues to be addressed in the above iterative process, namely: (1) how to relate an appropriate density function to the sizing field $H(p)$? (2) due to the complexity of 3D cases, computing each Voronoi region's mass centre exactly is impractical, then, how to obtain the approximate mass centres? and (3) when the mass centre of a Voronoi region is *near* (in some measure) the boundary and if it was inserted in the existing mesh, a highly distorted element might be generated; how to modify the construction of CCVDT in such cases? We will address these issues next.

4.1. Density function definition

To define the density function, we rely on the equidistribution of the error (cost) functional in the 3D case which is based on the Gersho conjecture [38].

Let V_i be a three-dimensional Voronoi region, h_i be the radius of V_i, and $h = \max h_i \to 0$ as k, the number of generators, goes to infinity. Then we have the error equidistribution:

$$\int_{V_i} \rho(Z)\|Z - Z_i\|^2 \, \mathrm{d}Z \approx c, \quad \forall i$$

The cost functional $\int_{V_i} \rho(Z)\|Z - Z_i\|^2 \, \mathrm{d}Z$ reflects the approximation properties on the given element and thus we assume that it may provide control on the element quality. The CVDT minimizes the sum of all cost functionals, or equivalently, it obtains an asymptotic cost equi-distribution. Therefore, the equi-distribution principle leads naturally to a suitable definition of the density function.

By the assumption $h = \max h_i \to 0$, there is a point Z^* in V_i such that

$$\int_{V_i} \rho(Z)\|Z - Z_i\|^2 \, \mathrm{d}Z \approx \rho(Z^*)h^2|V_i|$$

where $|V_i|$ denotes the volume of V_i. The right side roughly equals $\alpha h^5 \rho(Z^*)$ for some constant α. Hence, combining with the equidistribution principle, we have:

$$\rho(Z^*)h^5 = C(=c/\alpha)$$

which leads to the following relation between the density function and the sizing field:

$$\rho(Z) = C/H(Z)^5$$

The choice of C is not important, hence, we take $C = 1$. It is now transparent that the density function ρ is directly related to the sizing field $H = H(p)$. The mesh of CCVDT with the density function ρ will conform well with the sizing field. Our numerical examples show that a large number of nearly equilateral tetrahedra appear in the CCVDT. Also, we have tested density functions that vary as the inverse 2nd, 3rd, 4th, and 6th powers of H, and we find the element shape quality improvements are all less than that of the 5th power. This gives, to certain extent, another numerical indication for the validity of the Gersho conjecture. Accordingly, this implies that our choice of the density function is appropriate for the construction of CCVDT.

4.2. Computation of mass centres

Let V_i be the Voronoi region of the interior unconstrained point P_i. In the 3D case, V_i is a polyhedron containing the point P_i. One possible way to compute the mass centre of V_i with the density function being ρ is as follows:

First, V_i is decomposed into N tetrahedra. The decomposition approach is very simple: let $\{e_j,\ j = 1, \ldots, n(V_i)\}$ be the set of edges (of the mesh) connecting P_i. From the construction of V_i, we know V_i is composed of $n(V_i)$ *simpler* polyhedra. Each simpler polyhedron is formed by a 3D planar polygon S and the point P_i, as illustrated in Figure 2. We then find the intersection point Q of the edge e_j with S. P_i, Q, and an edge of S forms a tetrahedron. This gives a complete decomposition of V_i into $N = \sum_{j=1}^{n(V_i)} m_j$ tetrahedra where m_j are the number of edges of S.

We next compute the mass centre of each tetrahedron using numerical integration together with linear interpolation for the sizing field values (or density function values) of quadrature points. If any vertex of some tetrahedron is outside the given domain, the contribution of this vertex is deleted in the mass centre's computation. We denote the mass centre and the mass of each tetrahedron by X_j and M_j. The mass centre Y_i of V_i can be then computed by:

$$Y_i = \frac{\displaystyle\int_{V_i} Y\rho(Y)\,\mathrm{d}Y}{\displaystyle\int_{V_i} \rho(Y)\,\mathrm{d}Y} = \frac{\displaystyle\sum_{j=1}^{N} X_j M_j}{\displaystyle\sum_{j=1}^{N} M_j}$$

4.3. Changes to the boundary points

In general, the boundary points, i.e. the surface triangular mesh points and some Steiner points added during the boundary recovery are fixed and not allowed to change. But when carrying

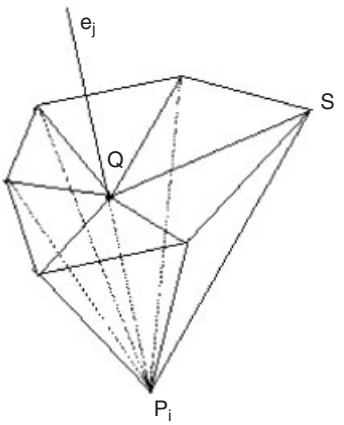

Figure 2. Tetrahedra division with respect to a connecting edge.

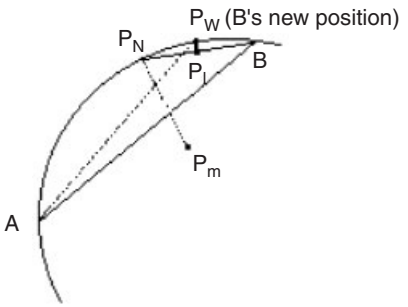

Figure 3. Projection and merging.

out the Lloyd iteration, the mass centres of some Voronoi regions may move close to the boundary, or even outside the domain. This is due to the following causes:

1. The input surface triangular mesh may not have good quality, e.g. there are some badly shaped elements with small angles.
2. The input domain has a complicated geometry, e.g. it contains small cavities or holes.
3. The given sizing field or the distribution of boundary sizing field varies substantially.
4. The initial Delaunay tetrahedral mesh contains some badly shaped elements, such as slivers, that are near the boundary.

Obviously, if we directly insert these generators into the boundary tetrahedronization, some highly-distorted elements would be generated in the mesh or the insertion procedure may fail. In these cases, we have the following remedies: if the mass centre moves outside the domain, we just delete this candidate generator; if the mass centre moves close to the boundary and the distance is less than some given criterion, we apply the following projection and merging technique (for simplicity, the 2D case is illustrated in Figure 3):

Let P_{m} be the mass centre. P_{m} is first projected to the boundary surface in the normal direction via the surface parameterization (see related discussion in Reference [35]). Denote the projection point by P_N. Let B be the nearest boundary point to P_N. If B is not allowed to move on the boundary due to the domain's geometric requirement (for example, the eight vertices of a cube domain are not allowed to move on the boundary), the generator P_{m} is deleted. Otherwise, the midpoint P_l of the line segment $\overline{BP_N}$ is projected onto the boundary surface similarly as P_{m}; then the mass centre P_{m} is deleted and B is moved to P_l's projection point P_w. This is to say, the possibly problematic mass centres are not inserted into the existing mesh; instead, their nearest boundary points are changed to appropriate new positions on the boundary surface if these boundary points are allowed to move without violating any boundary geometric constraints. Such measures keep a more reasonable points placement near the boundary and enhances the quality of the nearby elements. A direct consequence is the

removal of slivers or badly shaped elements near the boundary. This technique has been used by many other researchers for element quality improvement [17].

With the construction of CCVDT, the initial conforming Delaunay tetrahedral mesh is improved in several aspects which is numerically demonstrated in our meshing experiments. Most importantly, the points distribution provided by the CCVDT is globally optimal in the sense of cost or error functional being minimized, which in turn reflects the global quality of mesh elements. The topological structure of the mesh, especially the connectivity of vertices are globally improved. A large percentage of elements are improved to be nearly equilateral tetrahedra, and the number of badly shaped elements (quality under a given threshold) or slivers is greatly reduced, all reflect the enhancement of the overall mesh quality or average quality. Secondly, the points placement becomes more conforming to the given or derived sizing field. This makes the transition of element sizing smoother and results in fewer badly shaped transitional elements. The third aspect is that the badly shaped elements due to the existence of the badly shaped triangles in the surface mesh are eliminated in whole or by a large ratio, and the mesh structure near the boundary are much improved, in spirit similar to those generated using the advancing front method [7].

However, because the above optimization is a global process, there are still a very few number of local and isolated less-ideal elements in the mesh. Some classical local optimization techniques are applied for the further improvement of these elements.

5. LOCAL OPTIMIZATION FOR FURTHER IMPROVEMENT

We consider three simple topological operations which are discussed in detail in References [16–18, 22]:

(1) Flipping2-3 (shown in Figure 4): Given two elements sharing a face, this operation consists of connecting the opposite vertices, deleting the common face. The resulting structure has three elements. This operation is performed if the connecting edge passes through the common face and *the worst element of the resulting three elements has better quality than that of the worst of the initial two*.
(2) Flipping3-2 (shown in Figure 4): The inverse of Flipping2-3. The operation is performed under the condition that the quality of the worst element should be enhanced.

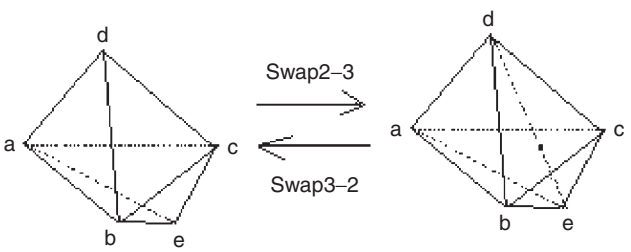

Figure 4. Flipping2-3 and the inverse Flipping3-2.

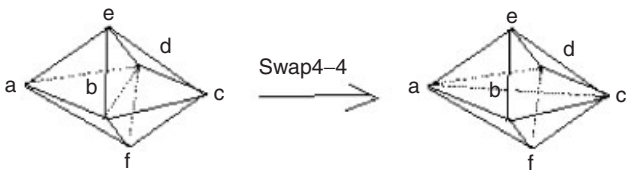

Figure 5. Flipping4-4.

(3) Flipping4-4 (shown in Figure 5): If an edge in the mesh is shared by four tetrahedra, and the union of these elements is an octahedron, the operation swaps the edge among the three pairs of opposite nodes in the octahedron. The operation is performed if the quality of the worst element is improved.

The above three operations are conducted in the following order:

1. Find the badly shaped (or worst) elements (quality less than a prescribed criterion) and their surrounding elements.
2. Analyse the faces of the elements and perform the operation Flipping2-3.
3. Analyse the edges of the elements and perform the operation Flipping3-2 or Flipping4-4.
4. Go back to step 1 until no operation can be performed.

Because the global optimization through the construction of CCVDT has eliminated almost all slivers or badly shaped elements and greatly enhanced the overall mesh quality, we need not to perform the complicated local operations like the cluster reconnection in References [3, 17] for their further improvement. The above three simple operations are sufficient for the improvement of the remaining badly shaped tetrahedra. This will be demonstrated by various numerical examples in the following section. And these swaps affect minimally the cost or error functional. Moreover, when the distortion of some elements is improved through swaps, the goodness of fit to the sizing function may be reduced on some occasions, but the reduction is very little. These insignificant changes are due to the fact that our swaps are local and usually the number of swaps needed to perform is small. Furthermore, the elements that need swapping are often isolated. Hence, we can omit the adverse effects of the swaps on the global properties of the mesh.

6. APPLICATION EXAMPLES

To present numerical examples, we use the criterion that a tetrahedron is badly shaped if its quality is less than 0.1 and is well-shaped if its quality is greater than 0.4.

In the following, we provide three examples to illustrate the effectiveness of the proposed optimization method. Each example includes the mesh, its cutting view and a table containing detailed elements quality statistics of the relevant meshes: the initial Delaunay tetrahedral mesh, the mesh of the CCVDT, the mesh after the final local optimization of the CCVDT (referred by CCVDT + LOCAL), and the mesh obtained from the initial mesh with only local optimization (referred by LOCAL), i.e. by coupling the constrained Laplacian smoothing with

the three kinds of swappings discussed above. The comparisons of the mesh quality data in the first three phases show the progress made in the improvement. The comparison with the last mesh shows the effectiveness of the CCVDT. All elements in each phase are divided into four groups. The percentages of badly shaped and well-shaped elements, average element quality and the minimum quality are given in the tables.

Besides the element quality statistics, we have also computed the values of the cost or error functional of each numerical example, and we find a general decreasing trend in these values, albeit a very small fluctuation which is due to the accuracy of computation in the procedure of Delaunay kernel, i.e. the Delaunay insertion. Overall, the error functionals all converge after a number of iterations.

Obviously, the CPU time for constructing the CCVDT equals the total time of each regular Delaunay meshing multiplied by the number of iterations. In cases where the sizing variation is not very large and the geometry is not very complex, the convergence is very fast and accordingly the meshing time is very reasonable. For instance, the number of Lloyd iterations in the construction of CCVDT for Examples 1 and 3 are both less than 200 while for Example 2, the iterations number goes up to 1000. Since the number of elements that need swapping is usually small, the additional CPU time of local swap is very small due to the locality of the local swaps. Naturally, the speeding up of the Lloyd's iteration is a very important direction for our future research, for instance, Newton's type schemes or over-relaxation schemes may be applied. Localization through domain decomposition procedures may offer significant reduction in the run time. Parallel algorithms also offer great potential to make the CCVDT meshing competitive for large scale problems. Recently, algorithms have been proposed for computing the CVTs on distributed computation systems [36], and applications of such algorithms to the meshing procedure are currently under exploration.

6.1. Example 1: a simple cube

Example 1 is the tetrahedronization of a simple cube with sizing refinement along one edge. Figure 6 shows the mesh (of CCVDT + Local) and its cutting view. Table I shows the elements quality statistics. From the table, we see that the initial mesh has 30 badly shaped elements and the minimum quality is 0.012, i.e., there are numerous slivers. Also, there are a fraction of tetrahedra whose quality falls between 0.1 and 0.4, and this results in a low average quality 0.674. Overall, they indicate the poor quality of the initial Delaunay mesh. The second column of the table offers dramatic elements quality enhancement: the badly shaped elements are decreased to zero; the number of elements falls in the quality interval 0.1–0.4 are substantially decreased, the average quality was improved to be 0.757 and the minimum quality is 0.145. The third column shows the mesh after the final local optimization which is a high-quality mesh: there is no badly shaped elements, the average quality is up to 0.774 and the percentage of well-shaped elements is up to 98.2. Even though the example is for a very simple geometry, the effectiveness of the optimization can be seen clearly.

6.2. Example 2: a mesh having fixed boundary vertices for a domain with hollow cylinders

Example 2, shown in Figure 7 and Table II, is a complex geometry which has many intersections of hollow cylinders, but the variation of the element sizing values is kept to be small. The mesh shown is also of CCVDT + Local. In this example, in the construction of the CCVDT, for comparison, first we fixed all boundary points and no one is allowed to move,

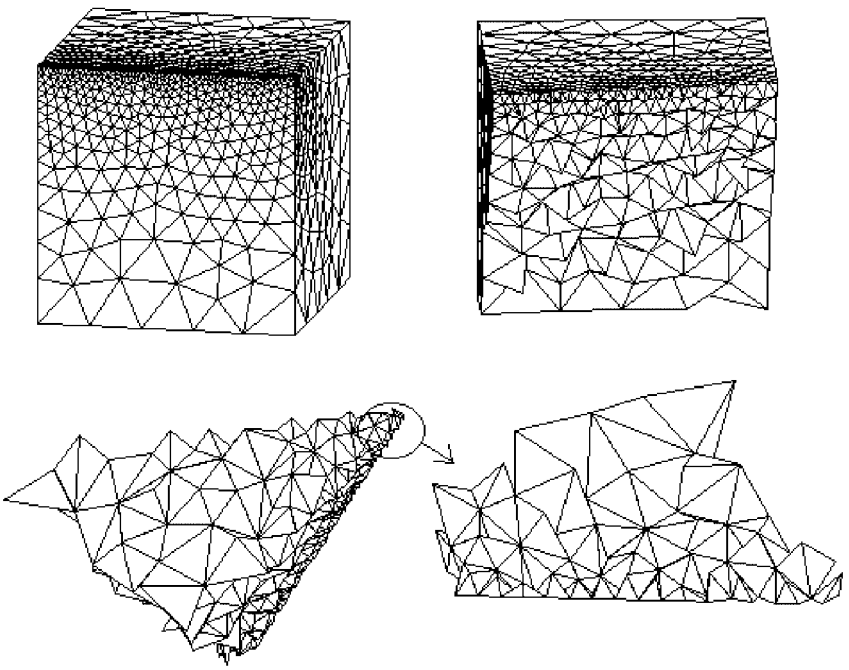

Figure 6. A cube's tetrahedral mesh and its cutting view.

Table I. Elements quality statistics of Example 1.

	Initial mesh	CCVDT	CCVDT + Local mesh	Local mesh
Elements number	7263	6886	6858	7209
$0.7 < Q < 1.0$	3410	4752	4851	3489
$0.4 < Q < 0.7$	3505	1933	1883	3471
$0.1 < Q < 0.4$	318	201	124	247
$0.0 < Q < 0.1$	30	0	0	2
$Q_{\min}$	0.012	0.145	0.237	0.079
Bad elements (%)	0.4	0.0	0.0	0.0^{+}
Good elements (%)	95.2	97.1	98.2	96.5
Average quality	0.674	0.757	0.774	0.705

i.e. no projection and merging operation is performed. In Table II, after the construction of CCVDT, there are still 14 badly shaped elements remaining in the mesh. Moreover, even after the final local flippings, there are still 4 such elements. But when we tried to perform boundary changing operation, i.e. projection and merging, we found that no such operations are allowed to perform. Close examination indicates that these four elements are all near the boundary and there are very few interior points in the neighbour, i.e. the boundary dominates the problem. That is to say, in this case, we cannot improve the mesh further if the boundary points distribution is not modified and accordingly adding more free interior points. This again illustrates that for complex geometry, the projection of boundary points and merging

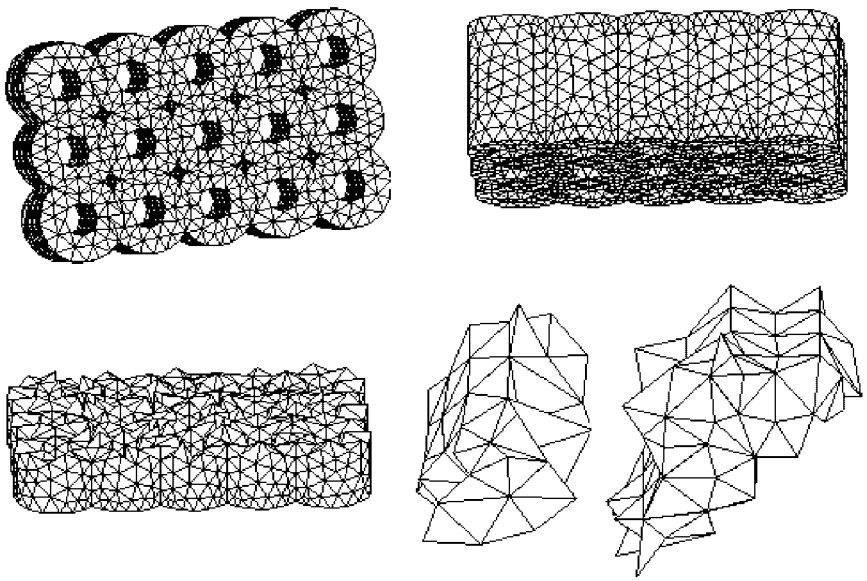

Figure 7. A complex model's tetrahedral mesh and its cutting view.

Table II. Elements quality statistics of Example 2.

	Initial mesh	CCVDT	CCVDT + Local mesh	Local mesh
Elements number	9627	9328	9214	9547
$0.7 < Q < 1.0$	848	2150	2269	871
$0.4 < Q < 0.7$	7016	5988	6023	7369
$0.1 < Q < 0.4$	1703	1176	918	1286
$0.0 < Q < 0.1$	60	14	4	21
$Q_{\min}$	0.004	0.045	0.076	0.043
Bad elements (%)	0.6	0.1	0.0^{+}	0.2
Good elements (%)	81.6	87.2	89.9	86.3
Average quality	0.586	0.699	0.712	0.632

operation in the construction of CCVDT is in general very necessary. However, other statistics, including average quality and the ratio of well-shaped elements, tell us the global optimization through CCVDT is indeed effective in improving the overall quality.

6.3. Example 3: an example for composite material simulation

The final example shown in Figure 8 and Table III is a mesh for a unit cell with nine stiff inclusions (spheres) for a composite material simulation. The sizing variation is not large (0.02–0.1). Here, we performed boundary-point treatments: projection and merging for 18 generators. The good news are reported in Table III: there is a tremendous decrease of badly shaped elements, from 158 to 7 after the CCVDT construction, and subsequently, the badly shaped elements are completely eliminated through the final local optimization. Similar to

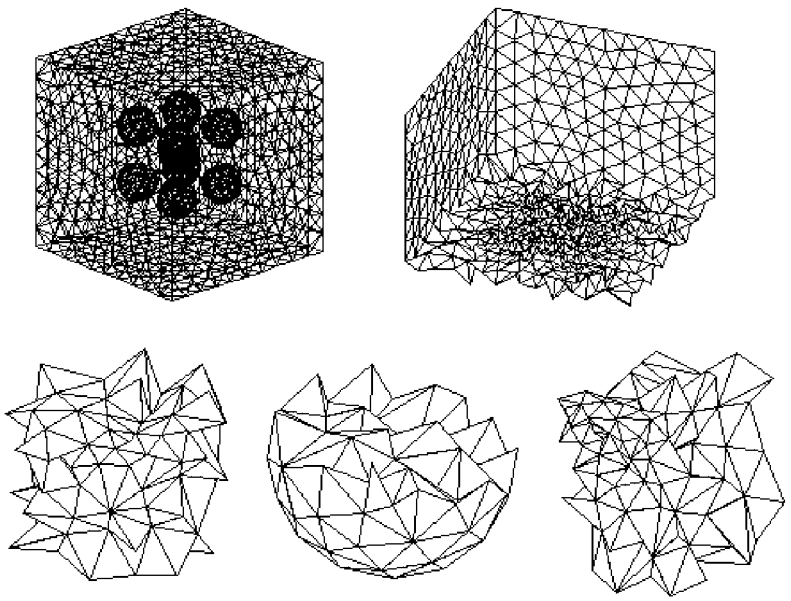

Figure 8. A mesh for composite material simulation and its cutting view.

Table III. Elements quality statistics of Example 3.

	Initial mesh	CCVDT	CCVDT + Local mesh	Local mesh
Elements number	28429	27126	27108	28188
$0.7 < Q < 1.0$	12742	21197	21215	15650
$0.4 < Q < 0.7$	13831	5471	5488	11054
$0.1 < Q < 0.4$	1698	451	405	1465
$0.0 < Q < 0.1$	158	7	0	19
$Q_{\min}$	0.005	0.052	0.152	0.049
Bad elements (%)	0.55	0.0^{+}	0.0	0.0^{+}
Good elements (%)	93.4	98.3	98.5	94.7
Average quality	0.661	0.769	0.773	0.716

Example 2, all the statistics clearly demonstrate the effectiveness of the proposed optimization method.

7. CONCLUDING REMARKS

In conclusion, the proposed tetrahedronal mesh generation and optimization method is based on the optimization properties of the centroidal Voronoi tessellations and is numerically reliable. It offers dramatic element quality enhancement which meets the goal set in the introduction: to globally optimize the points distribution and the topological structure of the mesh through the construction of the CCVDT and to further improve the mesh with local optimization

techniques. The extensions concerning the adaptive 3D CVT and CVDT are currently under investigation along with the application to the numerical simulation of partial differential equations arising in practical problems. In order to make our method competitive not only in the final outcome but also in terms of computational efforts, much of our future studies will be devoted to the speeding up of CCVDTs construction. At the same time, the theoretical analysis of the relation between the cost functional and the distortion of the element shape, and theoretical studies of the validity of the Gersho conjecture, are some challenging issues to be investigated further that would offer a more solid theoretical foundation for the methods proposed here.

ACKNOWLEDGEMENT

The authors would like to thank the referees for valuable suggestions that improved the discussion made in the paper.

REFERENCES

1. Lohner R, Parikh P. Generation of three-dimensional unstructured grids by the advancing-front method. *International Journal for Numerical Methods in Fluids* 1988; **8**:1135–1149.
2. Moller P, Hansbo P. On advancing front mesh generation in three dimensions. *International Journal for Numerical Methods in Engineering* 1995; **38**:3551–3569.
3. Rassineux A. Generation and Optimization of tetrahedral meshes by advancing front technique. *International Journal for Numerical Methods in Engineering* 1998; **41**:647–651.
4. Marcum, David L, Nigel P. Weatherill, Unstructured Grid Generation Using Iterative Point Insertion and Local Reconnection. *AIAA Journal* 1995; **33**(9):1619–1625.
5. Shephard MS, Georges MK. Automatic three-dimensional mesh generation technique by the finite element octree technique. *International Journal for Numerical Methods in Engineering* 1991; **32**:709–749.
6. Borouchaki H, Laug P, George PL. Parametric surface meshing using a combined advancing-front generalized Delaunay approach. *International Journal for Numerical Methods in Engineering* 2000; **49**:233–259.
7. Frey PJ, Borouchaki H, George P. 3D Delaunay mesh generation coupled with an advancing-front approach. *Computer Methods in Applied Mechanics and Engineering* 1998; **157**:115–131.
8. Borouchaki H, George PL. Optimal Delaunay point insertion. *International Journal for Numerical Methods in Engineering* 1996; **39**:3407–3437.
9. Borouchaki H, Lo SH. Fast Delaunay triangulation in three dimensions. *Computer Methods in Applied Mechanics and Engineering* 1995; **128**:153–167.
10. George PL, Hecht F, Saltel E. Automatic mesh generation with specified boundary. *Computer Methods in Applied Mechanics and Engineering* 1991; **92**:268–288.
11. Weatherill N, Hassan O. Efficient three dimensional delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering* 1994; **37**: 2005–2039.
12. Shewchuk J. Delaunay refinement mesh generation. *PhD Dissertation*, Computer Science Department, Carnegie Mellon University, 1997.
13. Wright JP, Jack AG. Aspects of three-dimensional constrained Delaunay meshing. *International Journal for Numerical Methods in Engineering* 1994; **37**:1841–1861.
14. George PL. *Delaunay Triangulation and Meshing*: *Application to Finite Elements*. Editions HERMES, Paris, 1998.
15. Amenta N, Bern M, Eppstein D. Optimal point placement for mesh smoothing. *Journal of Algorithms* 1999; **30**:302–322.
16. Freitag LA, Ollivier-Gooch C. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering* 1997; **40**:3979–4002.
17. Zavattieri PD, Dari EA, Buscaglia GC. Optimization strategies in unstructured mesh generation. *International Journal for Numerical Methods in Engineering* 1996; **39**:2055–2071.
18. Dari EA, Buscaglia GC. Mesh optimization: how to obtain good unstructured 3-D finite element meshes with not-so-good mesh generators. *Structural Optimization* 1994; **8**:181–188.
19. Cheng S, Dey T, Edelsbrunner H, Facello M, Teng S. Sliver Exudation. *Journal of the ACM* 2000; **47**: 883–904.

20. Miller G, Talmor D, Teng S, Walkington N. A Delaunay based numerical method for 3 dimensions: generation, formulation, and partition. In Proceedings of the 27th ACM Symposium on Theory and Computing 1995; 683–692.
21. Edelsbrunner H, Li X, Miller G, Stathopoulos A, Talmor D, Teng S, Ungor A, Walkington N. Smoothing and cleaning up slivers, Proceedings of the 32nd ACM Symposium on Theory and Computing 2000; 273–277.
22. Buscaglia GC, Dari EA. Anisotropic mesh optimization and its application in adaptivity. *International Journal for Numerical Methods in Engineering* 1997; **40**:4119–4136.
23. Du Q, Faber V, Gunzburger M. Centroidal Voronoi tessellations: applications and algorithms. *SIAM Review* 1999; **41**:637–676.
24. Du Q, Gunzburger M, Ju L. Meshfree probabilistic determination of point sets and support regions for meshless computing. *Computer Methods in Applied Mechanics and Engineering*, 2001, accepted for publication.
25. Du Q, Gunzburger M. Grid generation and optimization based on centroidal Voronoi tessellations. *Applied and Computational Mathematics*, 2001, accepted for publication.
26. Lloyd S. Least square quantization in PCM. *IEEE Transactions on Information Theory* 1982; **28**:129–137.
27. Lohner R. Regridding surface triangulation. *Journal of Computational Physics* 1996; **126**(6):1–10.
28. Hartmann E. A marching method for the triangulation of surfaces. *The Visual Computer* 1998; **14**:95–108.
29. Sheng X, Hirsch B. Triangulation of trimmed surfaces in parametric space. *Computer-Aided Design* 1992; **24**(8):437–444.
30. Shimada K. Anisotropic triangular meshing of parametric surfaces via close packing of ellipsoidal bubbles. *Sixth International Meshing Roundtable 97 Proceedings*, 1997; 63–74.
31. Tristano JR, Owen SJ, Canann SA. Advancing front surface mesh generation in parametric space using a Riemannian surface definition. *Seventh International Meshing Roundtable*, *98 Proceedings*, 1998; 429–445.
32. Bern M, Plassman P. Mesh generation. *Handbook of Computational Geometry*. Elsevier: Amsterdam, 2000; 291–332.
33. Okabe A, Boots B, Sugihara K. *Spatial tessellations*: *Concepts and Applications of Voronoi Diagrams*. Wiley: Chichester, 1992.
34. Aurenhammer F. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys* 1991; **23**:345–405.
35. Du Q, Gunzburger M, Ju L. Constrained centroidal Voronoi tessellations on general surfaces. *SIAM Journal on Scientific Computing*, 2001, submitted for publication.
36. Du Q, Gunzburger M, Ju L. Probablistic methods for centroidal Voronoi tessellations and their parallel implementations. *Journal of Parallel Computing* 2002; **28**:1477–1500.
37. Babuska I, Rheinboldt W. Error estimates for adaptive finite element computations. *SIAM Journal on Numerical Analysis* 1978; **15**:736–754.
38. Gersho A. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory* 1979; **25**: 373–380.
39. Gray R, Neuhoff D. Quantization. *IEEE Transactions on Information Theory* 1998; **44**:2325–2383.
40. Baker TJ. Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation. *Engineering Computations* 1989; **5**:161–175.

