



Grid generation and optimization based on centroidal Voronoi tessellations [☆]

Qiang Du, Max Gunzburger ^{*}

Department of Mathematics, Iowa State University, 400 Carver Hall, Ames, IA 5001-2064, USA

Abstract

Centroidal Voronoi tessellations (CVTs) are Voronoi tessellations of a region such that the generating points of the tessellations are also the centroids of the corresponding Voronoi regions. Such tessellations are of use in very diverse applications, including data compression, clustering analysis, cell biology, territorial behavior of animals, and optimal allocation of resources. In this paper, we explore the use of CVTs in grid generation in connection with finite element approximations of partial differential equations. We begin by describing these tessellations and methods for their determination. We then discuss their application to mesh generation and finish with some examples of their use for the solution of partial differential equations.

© 2002 Elsevier Science Inc. All rights reserved.

Keywords: Unstructured grids; Mesh generation; Mesh optimization; Centroidal Voronoi tessellation; Delaunay triangulation; Finite element methods

1. Introduction

During recent years, there has been a concerted effort in the scientific computing community to develop automated methodologies for the solution of complex problems. A key element of the methodologies is the generation of meshes that are optimal in some sense related to minimizing the costs of solving the complex problem. The search for such mesh generation methodologies

[☆] Supported by the National Science Foundation under grant no. CCR-9988303.

^{*} Corresponding author.

E-mail addresses: qdu@iastate.edu (Q. Du), gunzburg@iastate.edu (M. Gunzburger).

continues and motivates further studies of fast, automatic, and optimal algorithms for generating quality meshes [5].

Mesheres are often classified as either being structured or unstructured [25]. While also applicable to the former, the main focus of this paper is on the latter.

A very popular method for unstructured mesh generation is the Voronoi–Delaunay triangulations [8]. Given a set of input points $\{\mathbf{z}_i\}_{i=1}^k$ belonging to a domain $\Omega \subset \mathbb{R}^N$, the *Voronoi* region \hat{V}_i corresponding to the point \mathbf{z}_i consists of all points in Ω that are closer to \mathbf{z}_i than to any other point in the set. The set $\{\hat{V}_i\}_{i=1}^k$ forms a partition of Ω and is known as a *Voronoi tessellation* or *Voronoi diagram* of Ω . The points $\{\mathbf{z}_i\}_{i=1}^k$ are called *generating points* or *generators*. The dual *Delaunay triangulation* is formed by connecting pairs of generating points which correspond to adjacent Voronoi regions [1,16,22].

Voronoi tessellations and Delaunay triangulations are very useful in a variety of applications. Historically, these concepts have been reinvented, given different names, generalized, studied, and applied many times over in many different fields [6]. For a comprehensive treatment of Voronoi diagrams, see [22].

In the context of unstructured mesh generation, Delaunay, triangulations have often been used as a good starting point. The triangulation may contain triangles with small angles, or triangles with greatly varying area so that most of the algorithms related to the Voronoi–Delaunay triangulations do not provide a guarantee about the quality of the resulting mesh.

The quality of the Voronoi–Delaunay triangulations is often associated with the distribution of the generating points. To find good distribution of points, several techniques have been proposed such as the advancing front method [10,11], the method of sphere packing [20,21], and the method of successive insertion of Steiner points [5,24]. Some of these techniques can indeed produce meshes satisfying certain quality constraints, e.g., the minimum angle is larger than a user provided lower bound [5].

Recently, we have been studying the concept of *centroidal Voronoi tessellations* [6] which offer many superior properties compared to ordinary Voronoi tessellations. Thus, it may be expected that the centroidal Voronoi tessellation based Delaunay triangulation (CVDT) might provide better alternatives to existing methodologies for generating high quality meshes.

The CVDT provides, in some sense, an optimal distribution of generating points; indeed, we show that the construction of CVDT generalizes many existing local smoothing techniques. In fact, a centroidal Voronoi tessellation (CVT) is constructed based on an associated *density function* and *cost* (or *error*, or *distortion measure*, or *energy*) functional. Within such a framework, it becomes possible to establish an important theoretical connection between mesh smoothing techniques and some mesh quality measure. In the context of the numerical solution of partial differential equations, the density function and

cost functional can be related to error estimators for the underlying discrete approximation so that the CVDT should also be useful in an adaptive solution strategy.

The paper is organized as follows. In Section 2, we describe the basic concepts used. In Section 3, we present various algorithms for determining CVDTs and, in Section 4, we discuss applications to mesh generation. In Section 5, we present some preliminary numerical studies on the construction of CVDTs and the use of CVDTs for the solution of model partial differential equations. Some comments are provided in Section 6.

2. Centroidal Voronoi tessellations and Delaunay triangulations

Given a region V and a density function $\rho(\mathbf{y})$ defined on V , the *mass centroid* \mathbf{z}^* of V is defined by

$$\mathbf{z}^* = \frac{\int_V \mathbf{y} \rho(\mathbf{y}) \, d\mathbf{y}}{\int_V \rho(\mathbf{y}) \, d\mathbf{y}}. \quad (1)$$

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

Definition 1. Given the set of points $\{\mathbf{z}_i\}_{i=1}^k$ in the domain Ω and a positive density function ρ defined on Ω , a Voronoi tessellation is called a CVT if

$$\mathbf{z}_i = \mathbf{z}_i^*, \quad i = 1, \dots, k, \quad (2)$$

i.e., the points \mathbf{z}_i that serve as generators for the Voronoi regions V_i are themselves the mass centroids of those regions. The corresponding dual Delaunay triangulation is referred to as the CVDTs.

CVTs are special in that, given an arbitrary set of points $\{\mathbf{z}_i\}_{i=1}^k$ in $\Omega \subset \mathbb{R}^N$, these points will not in general be the centroids of their associated Voronoi regions; see Fig. 1.

The notion of Voronoi regions and centroids, and therefore, of centroidal Voronoi regions, may be generalized to more abstract spaces and to metrics other than the Euclidean ℓ^2 norm [6]; for example, one can instead use $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_{\ell^p}$ to define the Voronoi partition. The generalized mass centroid of V may then be defined by the point x in V (or \mathbb{R}^N) that minimizes $\int_V \rho(\mathbf{y}) d(\mathbf{x}, \mathbf{y}) \, d\mathbf{y}$. One may also define CVTs for more abstract objects other than points; see [6,22] for more details.

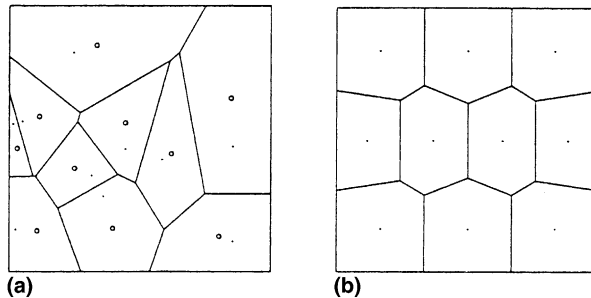


Fig. 1. A Voronoi tessellation of a square with 10 randomly selected points ((a) (o) centroids; (•) generators) and a 10-point centroidal Voronoi tessellation ((b) (•) generators and centroids).

For any tessellation $\{V_i\}_{i=1}^k$ of the domain Ω and a set of points $\{z_i\}_{i=1}^k$ in Ω , we can define the following *cost* (or *error* or *energy*) functional:

$$\mathcal{F}(\{V_i\}_{i=1}^k, \{z_i\}_{i=1}^k) = \sum_{i=1}^k \int_{V_i} \rho(x) \|x - z_i\|^2 dx. \quad (3)$$

The standard CVTs along with their generators are critical points of this cost functional. It was shown in [6] that it is possible to have a CVT as a saddle point, though the minimizers of \mathcal{F} are geometrically more stable.

In practice, the positions of the generators may be limited by various constraints. For example, in the context of mesh generation, it may be required that a certain number of generators be located on the boundary of Ω . This motivates another generalization of the CVT concept, that is, the *constrained* CVT defined as the minimum of \mathcal{F} under some specified constraints. Thus, we may consider the abstract optimization problem with P symbolizing some constraints on the location of the generating points.

Definition 2. Given the set of points $\{z_i\}_{i=1}^k$ in Ω , a density function ρ , and a constraint set P , a Voronoi tessellation is called a *constrained centroidal Voronoi tessellation* (CCVT) if $\{\{V_i\}_{i=1}^k, \{z_i\}_{i=1}^k\}$ is a solution of the problem

$$\min_{\{z_i\}_{i=1}^k \in P, \{V_i\}_{i=1}^k} \mathcal{F}(\{V_i\}_{i=1}^k, \{z_i\}_{i=1}^k). \quad (4)$$

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

It is understood that the set P should be chosen so that the minimization problem is solvable. This is mostly the case in the mesh generation context as geometrical constraints often make P well defined and compact.

Our current research effort is aimed at advancing the theoretical understanding of CVDTs and their properties, developing various algorithms and software tools for the computation of CVDTs on various computation platforms (in particular, on modern, high performance, parallel supercomputers and computer clusters), and, together with scientists and engineers from other disciplines, applying the concepts of CVDTs and the software tools to solve problems in diverse fields of applications.

3. Algorithms for CVDTs

Usually there are two components in the algorithms for finding CVDTs: the first being the calculation of the generators of the CVT and the second being the construction of the corresponding Delaunay triangulation from those generators. Once the generators are found, the second component can be accomplished using standard Delaunay triangulation algorithms. Thus, we focus on methods for finding the generators of CVTs. We only discuss two such methods which are the simplest representatives of the two method classes, i.e., probabilistic and deterministic. More sophisticated and efficient probabilistic algorithms are discussed in [15].

3.1. Probabilistic approaches

We focus on the *random* MacQueen's method which has the advantage that it does not require the calculation of the Voronoi tessellation until the final step. The method is defined as follows. Given a set Ω ; a positive integer k , and a probability density function ρ defined on $\overline{\Omega}$:

0. select an initial set of k points $\{\mathbf{z}_i\}_{i=1}^k$ in Ω , using a Monte Carlo method; initialize the index $j_i = 1$ for all $i = 1, \dots, k$;
1. select a $\mathbf{y} \in \Omega$ at random according to the probability density function $\rho(\mathbf{y})$;
2. find a \mathbf{z}_i in $\{\mathbf{z}_i\}_{i=1}^k$ that is closest to \mathbf{y} ; denote the index of that \mathbf{z}_i by i^* ;
3. set

$$\mathbf{z}_{i^*} \leftarrow \frac{j_{i^*} \mathbf{z}_{i^*} + \mathbf{y}}{j_{i^*} + 1} \quad \text{and} \quad j_{i^*} \leftarrow j_{i^*} + 1;$$

- the new \mathbf{z}_{i^*} , along with the unchanged $\{\mathbf{z}_i\}_{i \neq i^*}$, form the new set $\{\mathbf{z}_i\}_{i=1}^k$;
4. if this new set of points meets some convergence criterion, find the corresponding Delaunay triangulation and terminate; otherwise, return to step 1.

Note that the index j_i represents the number of times that the point \mathbf{z}_i has been updated.

MacQueen's algorithm has the following interpretation in the language of clustering methods. At the ℓ th stage of the algorithm, one starts with the positions of the cluster centers, i.e., the means, \mathbf{z}_i , $i = 1, \dots, k$, and a clustering of $(\ell + k)$ points (the original k points of step 0 plus the ℓ points \mathbf{y} 's previously selected in step 1) into k corresponding clusters. One then selects a new random point \mathbf{y} , locates the closest cluster center, and adds it to the corresponding cluster. Of course, one also updates the mean, i.e., the cluster center, of the enlarged cluster. For more detailed discussions, see [18]; see also [6] for a more comprehensive list of references.

An advantage of MacQueen's method lies in the fact that there is no need to compute the Voronoi tessellations or the Delaunay triangulations until the termination step. This is very attractive in a situation where only the generators are sought after. However, in the context of grid generation, the final computation of the Delaunay triangulation cannot be skipped.

3.2. Deterministic approaches

We only describe an algorithm based on the popular Lloyd's method 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 $\bar{\Omega}$:

0. select an initial set of k points $\{\mathbf{z}_i\}_{i=1}^k$, e.g., by using a Monte Carlo method;
1. construct the Voronoi tessellation $\{V_i\}_{i=1}^k$ of Ω associated with the points $\{\mathbf{z}_i\}_{i=1}^k$;
2. compute the mass centroids of the Voronoi regions $\{V_i\}_{i=1}^k$ found in step 1; these centroids are the new set of points $\{\mathbf{z}_i\}_{i=1}^k$;
3. if this new set of points meets some convergence criterion, find the corresponding Delaunay triangulation and terminate; otherwise, return to step 1.

We keep the convention of naming the above algorithm *Lloyd's method*. The original Lloyd's method [17] was used to just find the CVT and it may be viewed as a fixed point iteration for the mapping between generators and centroids. A more detailed discussion along with a list of references can be found in [6].

3.3. Algorithms for constrained CVDTs

Various generalizations are possible for the construction of CCVDTs. We illustrate with a few examples. In the simplest setting of the one-dimensional case, two of the generators may be confined to be the two endpoints of the

given interval $\Omega = [a, b]$. We propose the following modification to MacQueen’s method:

0. let $\mathbf{z}_1 = a$ and $\mathbf{z}_k = b$ and select an initial set of $k - 2$ points $\{\mathbf{z}_i\}_{i=1}^k$, using a Monte Carlo method; set $j_i = 1$ for all $1 < i < k$;
1. select a $\mathbf{y} \in \Omega$ at random, according to the probability density function $\rho(\mathbf{y})$;
2. find that \mathbf{z}_i that is closest to \mathbf{y} ; denote the index of that \mathbf{z}_i by i^* ;
3. if $i^* = 1$ or $i^* = k$, return to step 1; otherwise set

$$\mathbf{z}_{i^*} \leftarrow \frac{j_{i^*}\mathbf{z}_{i^*} + \mathbf{y}}{j_{i^*} + 1} \quad \text{and} \quad j_{i^*} \leftarrow j_{i^*} + 1;$$

the new \mathbf{z}_{i^*} , along with the unchanged $\{\mathbf{z}_i\}_{i \neq i^*}$, form the new set $\{\mathbf{z}_i\}_{i=1}^k$;

4. if this new set of points meets some convergence criterion, construct the Delaunay triangulation and terminate; otherwise, return to step 1.

Generalizations to more complicated constraint sets may be done in the same spirit.

Similarly, let P be a constraint set, we propose the following modification to the deterministic Lloyd method:

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

$$\mathcal{G}(\{\mathbf{z}_i\}_{i=1}^k) = \sum_{i=1}^k \int_{V_i} \rho(x) \|x - z_i\|^2 dx; \tag{5}$$

the minimizer is the new set of points $\{\mathbf{z}_i\}_{i=1}^k$;

3. if this new set of points meets some convergence criterion, find the corresponding Delaunay triangulation and terminate; otherwise, return to step 1.

The modified Lloyd algorithm again has the property that the functional \mathcal{F} is monotonically decreasing throughout the iteration.

4. Applications of CVDTs to mesh generation

In [6], we have discussed many applications of CVTs such as the optimal quadrature rules, optimal representation, quantization, clustering, optimal placement of resources, cell division, and territorial behavior of animals. Here,

we are primarily interested in the application of CVTs and CVDTs to mesh generation and the numerical solution of partial differential equations.

4.1. Unconstrained mesh generation

Generically, there are many geometrical properties associated with the Delaunay triangulation that include the following.

Delaunay criterion: The interior of the circumsphere of a simplex in the triangulation contains no generating points.

Dual Delaunay property: The edge is perpendicular to some face of the Voronoi tessellation.

Empty circle property: For each edge, a sphere can be found which contains the edge's endpoints but does not contain any other generating points.

In two dimensions, the Delaunay triangulation also possesses the smallest angle, the largest circumscribing circle, and the largest minimum enclosing circle properties [20,23]. The construction of Delaunay triangulations is well studied and includes divide-and-conquer, plane-sweep, and randomized incremental algorithms [3,8,14]. Thus, the application of the CVDT to unconstrained mesh generation deals mainly with the distribution of the *generators* according to some density function ρ . The function ρ needs to reflect the properties of the solution. It can either be given based on previously known information or be given by some local error estimators of the underlying solution.

4.2. Constrained mesh generation

In more practical applications, meshes are subject to constraints on the position of the generating points and perhaps as well on the edges and faces of the simplices. We focus on the important constraint of marking the meshes conform with the geometrical boundary. There are many different approaches to handling such constraints, some of which are discussed below.

From the boundary to the interior: One may predetermine a subset of generating points on the boundary, e.g., via a lower dimensional CVT construction based on a pre-defined density function. The advantage of this approach is the consistent use of the CVT concept which results in a relatively simple algorithm. The drawbacks are that it may not be easy to determine a priori the number of generating points on the boundary and the appropriate boundary density function.

From the interior to the boundary: One may, of course, construct the CVT and the CVDT without applying any constraints using a standard algorithm such as the Lloyd iteration. Whenever a new set of generators is found, one determines if any of the corresponding Voronoi regions extend to the exterior of the domain. Some procedure can then be applied to project some or all of

the corresponding generators to the boundary of the domain and then continue the iteration process.

Variational formulation: In more generic settings, consider CVDTs for a bounded domain where we have N generating points \mathbf{z}_i in the interior of the domain Ω and M points \mathbf{y}_k on the boundary. Consider the minimization of the modified functional

$$E(\{\mathbf{z}_i, V_i\}; \{\mathbf{y}_k, W_k\}) = \sum_{i=1}^N \int_{V_i} \rho_1(\mathbf{x}) \|\mathbf{x} - \mathbf{z}_i\|^2 dV \\ + \sum_{k=1}^M \int_{W_k} \rho_2(\mathbf{x}) \|\mathbf{x} - \mathbf{y}_k\|^2 dW$$

with respect to $\{\mathbf{z}_i\} \subset \Omega$, $\{\mathbf{y}_k\} \subset \partial\Omega$, $\{V_i\}$, and $\{W_k\}$, where the sets $\{V_i\}$ and $\{W_k\}$ jointly form a partition of Ω . The density functions may be defined differently to reflect the different weights placed on the interior and boundary generating points. The Delaunay triangulation corresponding to the minimizer can then be used for generating meshes which conform to the boundary. An alternate formulation requires the \mathbf{y}_k 's to be fixed points on the boundary so that one minimizes with respect to $\{\mathbf{z}_i\}$, $\{V_i\}$, and $\{W_k\}$. The variational problem covers both cases mentioned above.

Again, the density functions ρ_1 and ρ_2 may be based on prior information or on a local error estimator. They can be defined as the restrictions of the same function but in general, they may also be allowed to take different forms. The latter case may cause the resulting CVTs to have special geometric properties near the boundary. For practical applications, the relation between ρ_1 and ρ_2 needs to be further investigated.

4.3. Adaptation and local smoothing

Adaptation of meshes in general involves refinement, coarsening, and smoothing. For the smoothing part, various approaches have been studied [7,9,19].

One popular approach is Laplacian smoothing [7,19]. By successively moving each point to the centroid of its neighbors, it is often the case that the resulting grids are improved. This practical procedure has been widely used in engineering applications due to its simplicity, but theoretically, no connection is made to guarantee any specific mesh quality criterion [4].

The CVDT concept provides a good theoretical explanation to the smoothing process: by successively moving generators to the mass centers (of the Voronoi regions), the cost functional is reduced. With a suitable choice for the density function, the cost functional can be related to error estimators for the underlying problem. The averages of the neighboring generators

provide approximations to the mass centers of the Voronoi regions. Thus, the smoothing process mimics the process of iteratively constructing CVTs such as the Lloyd algorithm, and thus, contributes to the reduction of the discretization error.

4.4. Equi-partition of errors

An interesting question is related to the distribution of the cost functional (also referred to as the variance or distortion error, etc., in some contexts) when the number of generators becomes large. Let us consider the one-dimensional case and let $h_i = |V_i|$. Under the condition that the density is bounded and strictly positive, we have $h = \max h_i \rightarrow 0$, as k , the number of generators, goes to infinity. Then, it has been shown that [6]

$$|V_i| \approx c_1 \rho^{1/3}(m_i) \quad \text{and} \quad \int_{V_i} \rho(x)(x - x_i)^2 dx \approx c_2 \quad \forall i,$$

for some constants c_1, c_2 , where m_i denotes the midpoint of the interval V_i . That is, asymptotically speaking, the error is equally distributed in the Voronoi intervals and the sizes of the Voronoi intervals are inversely proportional to the one-third power of the underlying density at the midpoints of the intervals. For multidimensional CVTs, in [12] an important conjecture is made 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 but the conjecture remains open for three and higher dimensions. The equi-distribution of error principle can be established based on this conjecture [12,13].

5. Numerical experiments

In our preliminary numerical experiments, we have computed the CVDTs for some sample density functions $\rho(x)$ in a two-dimensional square and have used CVDTs to solve model Poisson equations.

5.1. Centroidal Voronoi–Delaunay triangulations

We first present examples of CVDTs and make comparisons with some generic Delaunay triangulations. In Fig. 2, two pairs of grids are presented with the top pair corresponding to a constant, uniform density and the bottom pair to the density $\exp(-2(x+1)^2 - 2(y+1)^2)$. For each pair, the Delaunay grid on the right is determined from a CVT and are of much better quality than

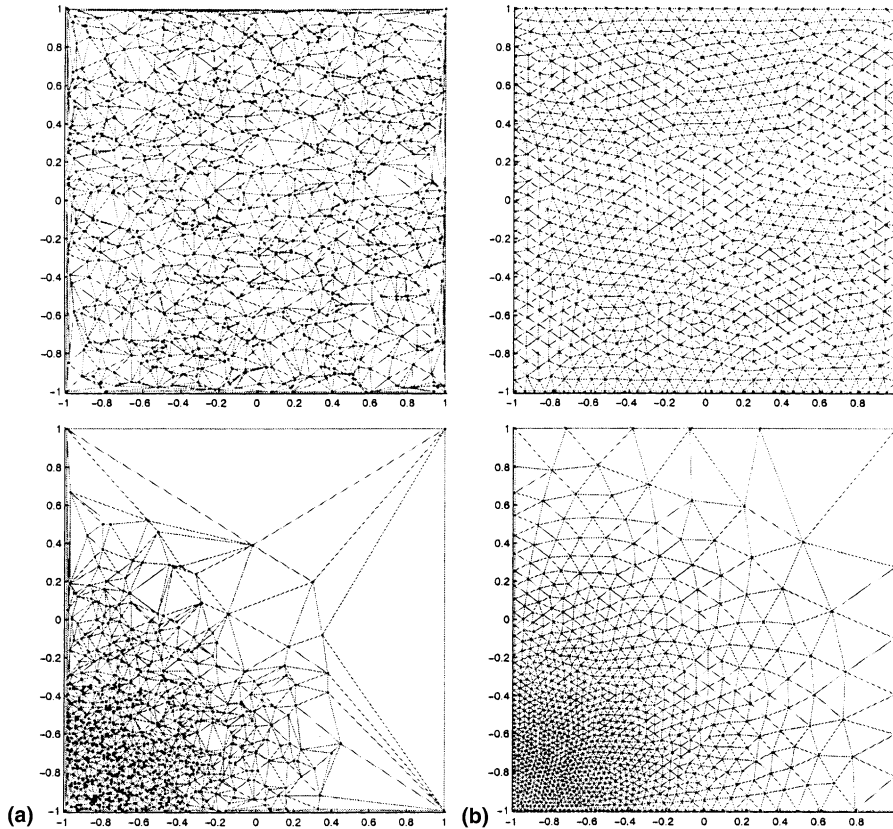


Fig. 2. Two-dimensional Delaunay triangulations for random Voronoi diagrams (a) and CVTs (b).

the corresponding Delaunay grid on the left which is obtained from randomly generated Voronoi diagrams, both based on the same density function.

5.2. Applications to the numerical solution of partial differential equations

CVTs and their dual Delaunay triangulations can obviously be linked to the *numerical solution of partial differential equations*. The optimal grid generation strategy can be incorporated into the discretization methodology for PDEs. It is hoped that a posteriori error estimates can be used to determine an effective density function. This would lead to a CVT-based grid adaptation strategy which can be incorporated into useful PDE solvers. Here, as a first small step towards that goal, we present some examples for solving the Poisson equation with Dirichlet boundary conditions in a unit square domain. In all cases, we use standard continuous, piecewise quadratic finite element spaces based on a

triangulation of the domain to effect the discretization. In the examples we present, we compare computations on logically Cartesian grids with those of comparable centroidal Voronoi-based grids. By comparable, we mean that two grids have a similar number of nodes and triangles.

We first compare uniform grid computations. In the Cartesian setting, it is well understood what one means by a uniform grid; see the figure on the left in Fig. 3. For centroidal Voronoi-based grids and their dual Delaunay triangulations, “uniform” implies that the grids were generated using a constant density function; see the figure on the right in Fig. 3. The Cartesian grid has 98 triangles, 225 nodes, and 169 unknowns; the CVDT has 96 triangles, 223 nodes, and 163 unknowns.

In Table 1, we give the L^2 -norm and the H^1 -seminorm of the error between the exact solution and its finite element approximation. From the table, it seems that there is not much to choose between using uniform Cartesian or CVDT grids for well-behaved exact solutions.

Next, we examine the errors between the exact solution

$$u(x, y) = \frac{1}{1 + 40x^2 + 40y^2} \tag{6}$$

and its quadratic finite element approximation. The function in (6) decays quickly away from the origin (one of the corners of the unit square domain)

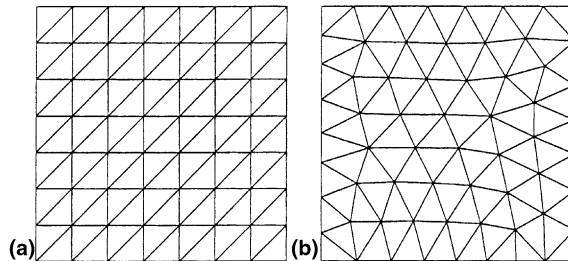


Fig. 3. Uniform Cartesian (a) and centroidal Voronoi (b) grids.

Table 1
Errors for uniform grid examples using Cartesian and CVDT grids

Exact solution	Cartesian grid		CVDT grid	
	L^2 error	H^1 error	L^2 error	H^1 error
$\cos \pi x \cos \pi y$	0.000820	0.0440	0.000590	0.0318
$e^{-x/2}e^{2y}$	0.000269	0.0132	0.000342	0.0168
$\sin \pi x \sin \pi y$	0.000827	0.0441	0.000683	0.0323
$\sin(4xy^2)$	0.002348	0.1211	0.001141	0.0588
$\sin 2\pi x \sin 2\pi y$	0.006742	6.2833	0.005576	6.2842

and thus has a large gradient near the origin. We approximate the solution of the Poisson equation using a series of grids, of both Cartesian and CVDT type, which are successively more refined near the origin. Of course, if the number of points remains fixed, then refinement near the origin results in coarsening away from the origin, i.e., near the upper right corner of the unit square.

For Cartesian grids, refinement in each coordinate direction is effected by monomial mappings, i.e., the interval $[0, 1]$ is mapped into itself using the mapping x^s which, for $s > 1$, has the effect of piling up points near the left end of the interval. We use the values $s = 1, 2, 3$, and 4 . For the case of an 8×8 grid (225 nodes, 98 triangles, 169 unknowns), $s = 1$, of course, yields the uniform Cartesian grid of the figure on the left in Fig. 3. The other values of s yield the grids of Fig. 4. We also consider the 16×16 grids (961 nodes, 450 triangles, 841 unknowns) of Fig. 5 which correspond to the cases $s = 1, 2$, and 3 . The L^2 -norm and the H^1 -seminorm of the error between the exact solution and its quadratic finite element approximation are given in Table 2. We see that if the number of grid points remains fixed, then refinement near the origin (with the corresponding coarsening away from the origin) at first yields better approximations; compare the $s = 1$ and $s = 2$ cases. However, more refinement, e.g., the $s = 3$ and $s = 4$ cases, near the origin causes sufficient coarsening away from the origin so that the overall error increases. Of course, adding grid points

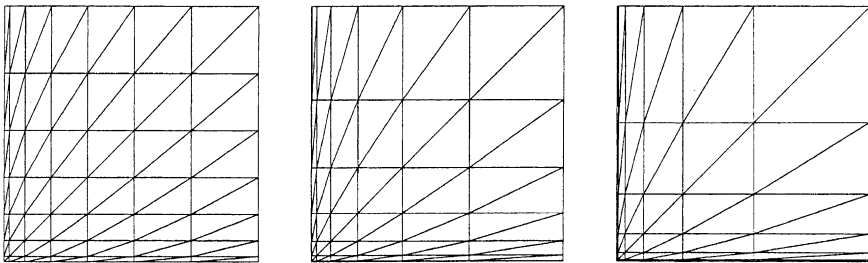


Fig. 4. 8×8 Cartesian grids increasingly refined near the origin: $s = 2, 3, 4$ (left to right).

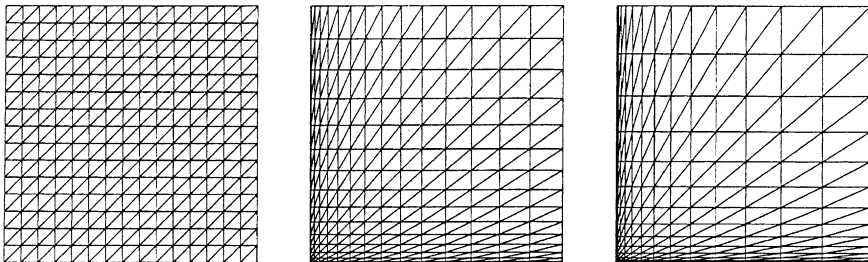


Fig. 5. 16×16 Cartesian grids increasingly refined near the origin: $s = 1, 2, 3$ (left to right).

Table 2

Errors for Cartesian grid with different levels of refinement for the exact solution of (6)

Refinement exponent, s	8×8 Cartesian grid		16×16 Cartesian grid	
	L^2 error	H^1 error	L^2 error	H^1 error
1	0.002931	0.1333	0.0003038	0.03041
2	0.000580	0.0363	0.0000592	0.00818
3	0.000975	0.0427	0.0000977	0.00960
4	0.001660	0.0567	–	–

for the same refinement exponent reduces the error. Comparisons between the 8×8 and 16×16 certainly illustrate the quadratic and cubic convergence of H^1 -seminorm and L^2 -norm of the error, respectively.

For CVDT grids, refinement is effected by appropriately choosing the density function. Ultimately, we would like to relate the density function to an a posteriori error estimate for the solution (see, e.g., [2]). Here, we merely choose a series of density functions to see their effect on the accuracy of the solution; we again use the example of the Poisson equation with Dirichlet boundary conditions for which (6) is the exact solution. Since a priori error estimates for quadratic finite element approximations involve local norms of the third derivatives of the exact solution, we choose the density functions to be proportional to powers of these local third derivative norms,

$$\rho(x, y) = \left\{ \sum_{i=0}^3 |\partial_x^i \partial_y^{3-i} u(x, y)|^2 \right\}^{k/2}.$$

In Fig. 6, we give some example CVDT grids generated from the three density functions corresponding to $k = 0.5$, $k = 1.0$, and $k = 1.5$; of course, $k = 0$ corresponds to the constant density function which generates the grid of the figure on the right in Fig. 3. These grids all correspond to 64 Voronoi gener-

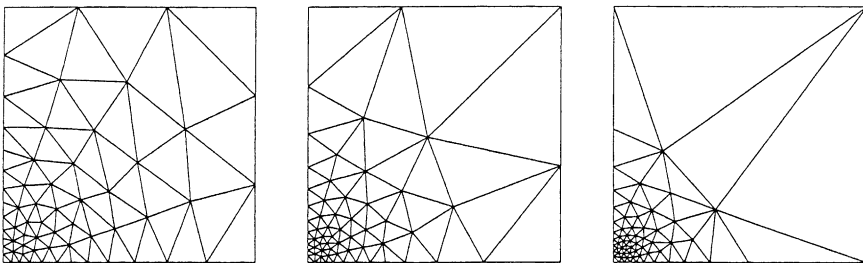


Fig. 6. CVDT grids having roughly 100 triangles and increasing refinement near the origin; $k = 0.5, 1.0, 1.5$ (left to right).

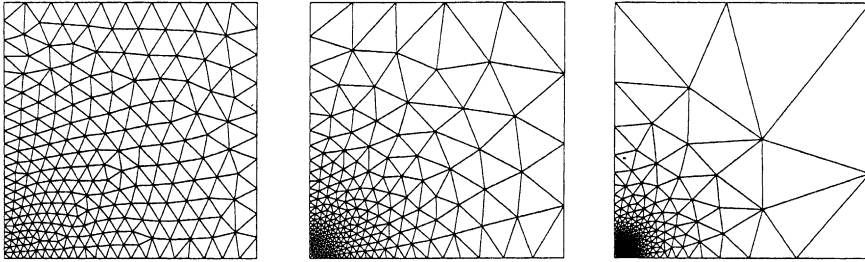


Fig. 7. CVDT grids having roughly 450 triangles and increasing refinement near the origin; $k = 0.25, 0.50, 0.75$ (left to right).

ators. The number of triangles is roughly the same as that for an 8×8 Cartesian grid. The number of triangles differ slightly as one changes the density function since it is in number of Voronoi generators that we are keeping fixed. In Fig. 7, we give some example CVDT grids having 256 Voronoi generators and different density functions corresponding to $k = 0.25, 0.5, 0.75$; the number of triangles for these grids is roughly the same as for a 16×16 Cartesian grid. The L^2 -norm and the H^1 -seminorm of the error between the exact solution and its quadratic finite element approximation are given in Table 3 for CVDT grids generated by several density functions. As was the case for Cartesian grids, refinement near the origin reduces the error from that of a uniform grid; however, with a fixed number of Voronoi generators, too much refinement near the origin causes excessive coarsening away from the origin and results in increases in the error. To achieve further reduction in the error, refinement near the origin should be accompanied by the addition of Voronoi generators, i.e., more triangles. We certainly see that having more points greatly reduces the error and in fact, as was the case for Cartesian grids, we see, for CVDT grids,

Table 3
Errors for CVDT grid with different levels of refinement for the exact solution of (6)

Exponent in density function	Number of triangles	CVDT grid	
		L^2 error	H^1 error
$k = 0.00$	96	0.001698	0.0841
$k = 0.25$	95	0.000289	0.0271
$k = 0.50$	95	0.000111	0.0120
$k = 1.00$	98	0.000504	0.0136
$k = 1.50$	102	0.003387	0.0459
$k = 2.00$	104	0.009145	0.1135
$k = 0.25$	443	0.0000178	0.00436
$k = 0.50$	447	0.0000259	0.00262
$k = 1.75$	460	0.0002880	0.00801

the expected quadratic and cubic convergence of the H^1 -seminorm and L^2 -norm of the error, respectively.

6. Concluding remarks

We have presented some preliminary studies on the use of CVTs and CVDTs for mesh generation. As CVTs possess a number of desirable geometric and optimization properties, it seems very sensible to utilize the properties of the dual Delaunay triangulations corresponding to CVTs especially for three-dimensional settings. Though not being proven, we believe that CVDTs can be very effective in avoiding *slivers* and for obtaining special-looking grids, e.g., long thin triangles aligned with the flow in boundary layers, wherever desirable. It remains to investigate further the effects of nonuniform densities on Delaunay triangulations. Such results evidently will be useful for the generation of optimal grids. Another question of both theoretical and practical importance is CVTs and the corresponding CVDTs in general metrics and how to compute them. For various applications, the use of Euclidean metric appears overly restrictive. We plan to explore the use of various generalized CVTs to achieve the purpose of designing anisotropic grids. For example, the metric could reflect the anisotropic properties. Another interesting investigation would be to incorporate *grid adaptation*, e.g., for time-dependent problems, into MacQueen grid generation strategy. One would also like to have the ability for *refining and coarsening* the grid. All of these topics will be pursued in our future work.

References

- [1] F. Aurenhammer, Voronoi diagrams – a survey of a fundamental geometric data structure, *Comp. Surveys* 23 (1991) 345–405.
- [2] I. Babuška, W. Rheinboldt, Error estimates for adaptive finite element computations, *SIAM J. Numer. Anal.* 15 (1978) 736–754.
- [3] C. Barber, D. Dobkin, H. Huhdanpaa, The quickhull algorithm for convex hulls, *ACM Trans. Math. Soft.* 22 (1996) 469–483.
- [4] M. Bern, D. Eppstein, Mesh generation and optimal triangulation, in: D. Du, F. Hwang (Eds.), *Computing in Euclidean Geometry*, World Scientific, River Edge, NJ, 1992, pp. 23–90.
- [5] L. Chew, Guaranteed-quality triangular meshes, Department of Computer Science, Technical Report 89–983, Cornell University, 1989.
- [6] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: applications and algorithms, *SIAM Rev.* 41 (1999) 637–676.
- [7] P. Eiseman, Adaptive grid generation, *Comput. Methods Appl. Mech. Engrg.* 64 (1987) 321–376.
- [8] S. Fortune, Voronoi diagrams and Delaunay triangulations, in: D. Du, F. Hwang (Eds.), *Computing in Euclidean Geometry*, World Scientific, River Edge, NJ, 1992, pp. 193–233.
- [9] L. Freitag, M. Jones, P. Plassmann, A parallel algorithm for mesh smoothing, *SIAM J. Sci. Comput.* 20 (1999) 2023–2040.

- [10] P. Frey, H. Borouchaki, P. George, Delaunay tetrahedralization using an advancing-front approach, in: Proc. Fifth Internat. Meshing Roundtable, Sandia National Lab, 1996, pp. 87–106.
- [11] P. Frey, H. Borouchaki, P. George, 3D Delaunay mesh generation coupled with an advancing-front approach, *Comput. Methods Appl. Mech. Engrg.* 157 (1998) 115–131.
- [12] A. Gersho, Asymptotically optimal block quantization, *IEEE Trans. Inform. Theory* 25 (1979) 373–380.
- [13] R. Gray, D. Neuhoff, Quantization, *IEEE Trans. Inform. Theory* 44 (1998) 2325–2383.
- [14] L. Guibas, D. Knuth, M. Sharir, Randomized incremental construction of Delaunay and Voronoi diagrams, *Algorithmica* 7 (1992) 381–413.
- [15] L. Ju, Q. Du, M. Gunzburger, Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations, to appear in *Parallel Computing*.
- [16] R. Klein, *Concrete and Abstract Voronoi Diagrams*, Lecture Notes in Computer Science, vol. 400, Springer, Berlin, 1989.
- [17] S. Lloyd, Least square quantization in PCM, *IEEE Trans. Inform. Theory* 28 (1982) 129–137.
- [18] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: L. Le Cam, J. Neyman (Eds.), *Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability*, University of California, vol. I, 1967, pp. 281–297.
- [19] D. McRae, K. Laflin, Dynamic grid adaptation and grid quality, in: *CRC Handbook of Grid Generation*, CRC Press, Boca Raton, 1998, pp. 34-1–34-33.
- [20] G. Miller, D. Talmor, S. Teng, N. Walkington, A Delaunay based numerical method for three dimensions: generation, formulation and partition, in: *Proc. 36th Symp. on Foundation of Computer Science*, IEEE Press, New York, 1995, pp. 683–692.
- [21] G. Miller, D. Talmor, S. Teng, N. Walkington, H. Wang, Control volume meshes using sphere packing: generation, refinement and coarsening, in: *Proc. Fifth Internat. Meshing Roundtable*, Sandia National Lab, 1996, pp. 47–62.
- [22] A. Okabe, B. Boots, K. Sugihara, *Spatial Tessellations, Concepts and Applications of Voronoi Diagrams*, Wiley, Chichester, 1992.
- [23] V. Rajan, Optimality of the Delaunay triangulation in R^d , *Discrete Comput. Geom.* 12 (1994) 189–202.
- [24] J. Ruppert, A Delaunay refinement algorithm for quality 2-dimensional mesh generation, *J. Algorithms* 18 (1995) 548–585.
- [25] J. Thompson, B. Soni, N. Weatherill (Eds.), *Handbook for Grid Generation*, CRC Press, Boca Raton, 1999.