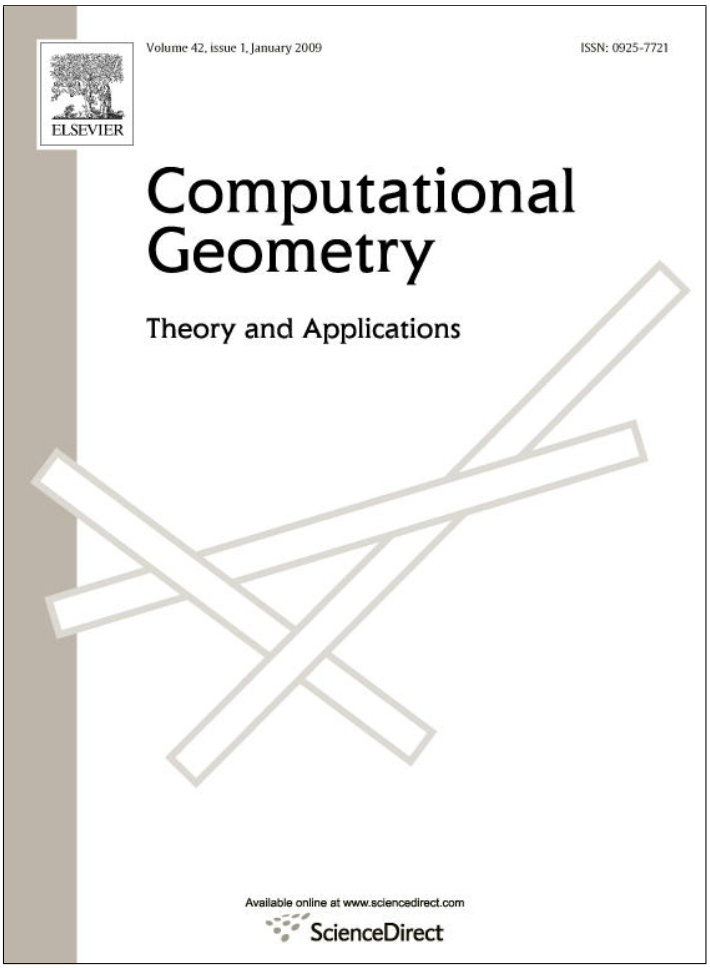

Computational Geometry 42 (2009) 1–19

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Constrained CVT meshes and a comparison of triangular mesh generators

Hoa Nguyen [a], John Burkardt [a], Max Gunzburger [a,1,*], Lili Ju [b,2], Yuki Saka [a]

[a] School of Computational Science, Florida State University, Tallahassee, FL 32306-4120, USA
[b] Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA

ARTICLE INFO

Article history:
Received 14 August 2006
Received in revised form 11 April 2008
Accepted 12 April 2008
Available online 27 April 2008
Communicated by M.T. Goodrich

Keywords:
Grid generation
Point placement method
Delaunay triangulations
Uniformity measures

ABSTRACT

Mesh generation in regions in Euclidean space is a central task in computational science, and especially for commonly used numerical methods for the solution of partial differential equations, e.g., finite element and finite volume methods. We focus on the uniform Delaunay triangulation of planar regions and, in particular, on how one selects the positions of the vertices of the triangulation. We discuss a recently developed method, based on the centroidal Voronoi tessellation (CVT) concept, for effecting such triangulations and present two algorithms, including one new one, for CVT-based grid generation. We also compare several methods, including CVT-based methods, for triangulating planar domains. To this end, we define several quantitative measures of the quality of uniform grids. We then generate triangulations of several planar regions, including some having complexities that are representative of what one may encounter in practice. We subject the resulting grids to visual and quantitative comparisons and conclude that all the methods considered produce high-quality uniform grids and that the CVT-based grids are at least as good as any of the others.

1. Introduction

Grid generation involves two phases: point placement and connection. Once the locations of points in a Euclidean domain are determined, we choose to connect the points so they are the vertices of a Delaunay triangulation of the domain, i.e., a triangulation for which no circumcircle of a triangle contains points in its interior. This property guarantees that the Delaunay triangulation maximizes the minimum angle among all possible triangulations of the point set. Delaunay triangulation is the method of choice for triangulating two-dimensional domains.

Even Delaunay triangulations can result in skinny triangles, i.e., triangles having one or more small angles. The avoidance of such anomalies requires the judicious selection of the locations of the points that become the vertices of the triangulation. There may be other requirements that point locations must satisfy in the grid generation setting. For example, some of the points should be located on the boundary of the domain so that one may apply the specified boundary conditions on the solution of the partial differential equation being discretized. Delaunay triangulations corresponding to point sets that are constrained in this manner are referred to as constrained Delaunay triangulations. In addition, the points may be required to satisfy some desired distributional characteristics, i.e., to be non-uniformly distributed to achieve some specified goal.

* Corresponding author.
E-mail addresses: nguyen@scs.fsu.edu (H. Nguyen), burkardt@scs.fsu.edu (J. Burkardt), gunzburg@scs.fsu.edu (M. Gunzburger), ju@math.sc.edu (L. Ju), saka@scs.fsu.edu (Y. Saka).

[1] Supported in part by the US Department of Energy under grant number DE-FG02-07ER64432 and the US National Science Foundation under grant number DMS-0308845.
[2] Supported in part by the US Department of Energy under grant number DE-FG02-07ER64431 and the US National Science Foundation under grant number DMS-0609575.

0925-7721/$ – see front matter
doi:10.1016/j.comgeo.2008.04.002

The focus of this paper is on how one selects the points that become the vertices of *uniform* Delaunay triangulations of planar regions. For us, uniformity can be interpreted to mean that all the triangles in the triangulation are close to the ideal case of congruent, equilateral triangles, or that the vertices of the triangulation are equally spaced, are in some sense isotropically distributed, and also cover the domain, i.e., there are no large subsets of the domain that contain no vertices. Of course, in many application involving the numerical solution of partial differential equations, one prefers to use non-uniform grids. However, not only are there applications for which uniform grids are preferable, but that setting can also serve to make baseline comparisons between different methods for selecting the locations of the grid points.

Our first goal is to describe a recently introduced means for selecting points in Euclidean regions, and especially for selecting grid points for mesh generation purposes. This strategy is based on the centroidal Voronoi tessellation (CVT) concept and, in particular, on generalizations that are useful for grid generation applications. Detailed discussions of the centroidal Voronoi tessellation (CVT) and constrained centroidal Voronoi tessellation (CCVT) concept can be found in [6] and [9], respectively. In Section 2, in addition to defining and analyzing CVT-based point-placement methods, we provide two algorithms, including one new one, for effecting CVT-based grid generation.

Our second goal is to compare several methods for uniformly triangulating planar regions. Since we want to make quantitative as well as visual comparisons, we gather together, in Section 3, several measures that can be used to assess the uniformity of points sets and triangulations of those sets. Then, in Section 4, we briefly describe the grid generation methods that we use in our comparative studies. In that section, we also provide figures and tables resulting from the application of the methods to several planar regions, including some having complexities that are representative of what one may encounter in practice. The figures may be used to visually compare the various methods; the tables, which contain the results of applying the quality measures defined in Section 3 to the computed grids, may be used for quantitative comparisons. We close Section 4 with a discussion of the results obtained and of current and future directions for CVT-based grid generation.

2. Constrained centroidal Voronoi tessellations

2.1. Centroidal Voronoi tessellations

Let $|\cdot|$ denote the Euclidean norm in $\mathbb{R}^N$. Given a bounded, open set $\Omega \subset \mathbb{R}^N$ with boundary Γ and given an integer $K > 1$, a *tessellation* of Ω into K subsets is a subdivision of Ω into K non-overlapping open subsets, i.e., any set $\{V_k\}_{k=1}^K$ such that $V_k \subset \Omega$, $V_k \cap V_\ell = \emptyset$ for $k \neq \ell$, and $\bigcup_{k=1}^K \overline{V}_k = \overline{\Omega}$. Given a set of points $\{\mathbf{z}_k\}_{k=1}^K$ belonging to $\overline{\Omega}$, for $k = 1, \ldots, K$, let

$$V_k = \left\{\mathbf{x} \in \Omega\colon\ |\mathbf{x} - \mathbf{z}_k| < |\mathbf{x} - \mathbf{z}_\ell| \text{ for } \ell = 1, \ldots, K,\ \ell \neq k\right\}. \tag{1}$$

Note that $\{V_k\}_{k=1}^K$ defines a tessellation of Ω. The set $\{V_k\}_{k=1}^K$ satisfying (1) is referred to as a *Voronoi tessellation* or *Voronoi diagram* of Ω, the points of the set $\{\mathbf{z}_k\}_{k=1}^K$ are referred to as the *generating points* or *generators* of the Voronoi tessellation, and each V_k is referred to as the *Voronoi region* or *Voronoi cell* corresponding to $\mathbf{z}_k$.

Given a non-negative and almost everywhere continuous density function $\rho(\mathbf{x})$ defined on Ω and given any region $V \subset \Omega$, we define its *centroid* or *center of mass* by

$$\bar{\mathbf{z}} = \frac{\int_{V_i} \mathbf{x}\rho(\mathbf{x})\,d\mathbf{x}}{\int_V \rho(\mathbf{x})\,d\mathbf{x}}. \tag{2}$$

Equivalently, $\bar{\mathbf{z}}$ is the unique solution of the problem

$$\min_{\mathbf{z}\in\mathbb{R}^N} F(\mathbf{z}), \quad \text{where } F(\mathbf{z}) = \int_V \rho(\mathbf{x})|\mathbf{x} - \mathbf{z}|^2\,d\mathbf{x}. \tag{3}$$

In particular, for each Voronoi region V_k, $k = 1, \ldots, K$, we can define its centroid $\bar{\mathbf{z}}_k$ by

$$\bar{\mathbf{z}}_k = \frac{\int_{V_k} \mathbf{x}\rho(\mathbf{x})\,d\mathbf{x}}{\int_{V_k} \rho(\mathbf{x})\,d\mathbf{x}}.$$

In general, the generators of a Voronoi tessellation do not coincide with the centers of mass of the corresponding Voronoi cells.

If it so happens that

$$\mathbf{z}_k = \bar{\mathbf{z}}_k \quad \text{for } k = 1, \ldots, K,$$

i.e., for each Voronoi region V_k, its generator $\mathbf{z}_k$ coincides with its center of mass $\bar{\mathbf{z}}_k$, we refer to the Voronoi tessellation as a *centroidal Voronoi tessellation* (CVT). The existence of centroidal Voronoi tessellations of a given set has been proved, but note that, in general, they are not uniquely defined; see [6].

Centroidal Voronoi tessellations can also be characterized as solutions of an optimization problem. Let

$$\mathcal{F}\big(\{\mathbf{z}_k, V_k\}_{k=1}^K\big) = \sum_{k=1}^K \int_{V_k} \rho(\mathbf{x})|\mathbf{x}-\mathbf{z}_k|^2\, d\mathbf{x}. \tag{4}$$

Then, we have the following result; see, e.g., [6].

Proposition 1. *Given an integer $K > 1$ and a non-negative and almost everywhere continuous density function $\rho(\mathbf{x})$ defined on Ω. Let $\{V_k\}_{k=1}^K$ denote an arbitrary subdivision of Ω into K non-overlapping, covering subsets and let $\{\mathbf{z}_k\}_{k=1}^K$ denote an arbitrary set of K points in Ω. Then, a necessary condition for $\mathcal{F}(\{\mathbf{z}_k, V_k\}_{k=1}^K)$ to be minimized is that $\{\mathbf{z}_k, V_k\}_{k=1}^K$ define a centroidal Voronoi tessellation of Ω.*

We see that $\mathcal{F}(\cdot)$ is a variance measure; we will refer to it as the *CVT energy*. The special nature of CVTs means that they have to be constructed. Algorithms for determining CVTs are discussed in [5,6,8,9,11,14,15]

2.2. Constrained centroidal Voronoi tessellations

The notion of constraining CVTs was introduced in [7,9]. Here, we discuss constrained CVTs for bounded, Lipschitz regions $\Omega \subset \mathbb{R}^N$ having piecewise smooth boundaries.

We define *constrained centroidal Voronoi tessellations* (CCVTs) as follows. Given a constraint set Q, CCVTs are solutions, if they exist, of the problem

$$\min_{\{\mathbf{z}_k, V_k\}_{k=1}^K} \mathcal{F}\big(\{\mathbf{z}_k, V_k\}_{k=1}^K\big) \quad \text{subject to} \quad \mathbf{z}_k \in Q,\ i = 1, \ldots, K, \tag{5}$$

where $\mathcal{F}(\cdot)$ is defined as in (4). The constraint set Q can take many different forms. For example, *all* of the points $\mathbf{z}_k$, $k = 1, \ldots, K$, could be constrained to lie on a given surface in $\mathbb{R}^N$; in this case, one obtains a centroidal Voronoi tessellation of that surface. This case is treated in [9]. In the context of grid generation for regions in $\mathbb{R}^N$, one may want *some* of the points $\mathbf{z}_k$, $k = 1, \ldots, K$, to be on the boundary of the region Ω. In this setting, there are at least three different ways to define the constraint set Q. One is a *from the boundary out to the interior* method in which a boundary grid is specified and then an interior grid is determined; see Section 2.2.1. Another is a *from the interior to the boundary* method in which the boundary grid is determined simultaneously to the interior grid (see Section 2.2.3), and another method sits somewhere in between the other two (see Section 2.2.2). Note that, depending on the case, points may be constrained by the explicit specification of their coordinates or by the requirement that their coordinates satisfy the formula defining (a portion of) the boundary.

2.2.1. Fixed points on the boundary

First, one could be given a surface mesh of $M < K$ points $\{\mathbf{z}_m^*\}_{m=1}^M$ on the boundary Γ of Ω; this mesh could be the output of a CAD program or could be determined as a CCVT of the surface Γ using the methods of [9]. These points are then fixed during the generation of a CVT. In this case, the problem (5) reduces to

$$\min_{\{\mathbf{z}_k, V_k\}_{k=1}^K} \mathcal{F}\big(\{\mathbf{z}_k, V_k\}_{k=1}^K\big) \quad \text{subject to} \quad \mathbf{z}_m = \mathbf{z}_m^*,\ m = 1, \ldots, M.$$

Thus, the points $\{z_m\}_{m=1}^M$ are fixed and the remaining points $\{\mathbf{z}_k\}_{k=M+1}^K$ are allowed to move freely in Ω. Note that the tessellation $\{V_k\}_{k=1}^K$ is not explicitly constrained.

In this type of CCVT, the boundary mesh is fixed and the quality of the resulting CCVT is largely determined by the quality of the given boundary mesh. Difficulties associated with controlling mesh quality when this approach is used are discussed in [10].

2.2.2. Sliding points along the boundary

Alternately, one could let some of the points $\{\mathbf{z}_m^*\}_{m=1}^M$ move to more advantageous positions, i.e., ones that result in a lower value of the CVT energy. Thus, we have that a subset $\{\mathbf{z}_m^*\}_{m=1}^{M_0}$, $M_0 < M$, of the boundary points remain fixed, e.g., at the corners of a domain in $\mathbb{R}^2$, while the rest of the boundary points are allowed to "slide" along smooth segments of the boundary Γ. More precisely, suppose that the boundary Γ of Ω can be subdivided into J smooth disjoint segments Γ_j, $j = 1, \ldots, J$, and that each Γ_j can be described by the equation $g_j(\mathbf{x}) = 0$, where $g_j(\cdot)$ is at least a C^1 function. A segment Γ_j could be connected to another segment by points (at a corner or vertex) or by curves (along edges in $\mathbb{R}^3$).

Next, with $M_J = M$, without loss of generality, we divide the given boundary mesh points $\{\mathbf{z}_m^*\}_{m=1}^M$ as follows: for some positive integers $M_1, \ldots, M_{J-1}$,

$$g_j\big(\mathbf{z}_m^*\big) = 0 \quad \text{for } m = M_{j-1}+1, \ldots, M_j. \tag{6}$$

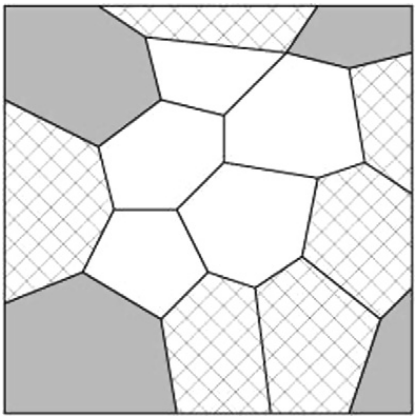

Fig. 1. The corner regions (shaded), boundary regions (cross-hatched), and interior regions (white) of a tessellation of the square.

The remaining given boundary mesh points $\{\mathbf{z}_m^*\}_{m=1}^{M_0}$ may be on any of the segments Γ_j or be exceptional points such as corners. We then define a CCVT by solving the following problem:

$$\min_{\{\mathbf{z}_k, V_k\}_{k=1}^K} \mathcal{F}\big(\{\mathbf{z}_k, V_k\}_{k=1}^K\big) \quad \text{subject to} \quad \begin{cases} \mathbf{z}_m = \mathbf{z}_m^* & \text{for } m = 1, \ldots, M_0 \text{ and} \\ g_j(\mathbf{z}_m) = 0 & \text{for } m = M_{j-1}+1, \ldots, M_j, \ \ j = 1, \ldots, J. \end{cases} \tag{7}$$

Thus, the boundary points $\mathbf{z}_m$, $m = 1, \ldots, M_0$, are fixed while the points $\mathbf{z}_m$, $m = M_0 + 1, \ldots, M$ are now allowed to slide along the boundary, but are not allowed to leave the segment Γ_j on which they were initially located. The remaining points $\{\mathbf{z}_k\}_{k=M+1}^K$ are allowed to move freely in Ω. Note that the tessellation $\{V_k\}_{k=1}^K$ is not explicitly constrained.

In this type of CCVT, only a portion of the boundary points are fixed, e.g., at corners, and the remaining points on the boundary are allowed to find more advantageous, in the sense of the problem (7), positions. However, the number of points in each smooth boundary segment is predetermined by the given point set $\{\mathbf{z}_m^*\}_{m=1}^M$. The quality of the resulting CCVT is still affected by the quality of the given boundary mesh, but to a lesser extent than that for the first approach.

2.2.3. Automatically placed points on the boundary

The third CCVT we consider is one on which neither the number or positions of the points on the boundary are predetermined. For the sake of simplicity, we describe this approach for domains in $\mathbb{R}^2$. We again assume that the boundary Γ of Ω can be subdivided into J smooth disjoint segments Γ_j, $j = 1, \ldots, J$, each of which can be specified as in (6). We assume that there are $M_c \geqslant 0$ corners and that $\Gamma_c = \{\mathbf{z}_m^*\}_{m=1}^{M_c}$ denotes the set of corner points. We label V_k, where $\{V_k\}_{k=1}^K$ is any tessellation of Ω, according to:

V_k is a *corner* region if $\overline{V}_k \cap \Gamma_c \neq \emptyset$

V_k is a *boundary* region if $\overline{V}_k \cap \Gamma_j \neq \emptyset$ for a single j

V_k is an *interior* region if $\overline{V}_k \cap \Gamma = \emptyset$.

The number of corner regions is equal to the number of corners M_c, unless the boundaries of two of the regions V_k intersect at a corner or the boundary of a region V_k includes more than one corner; the first happenstance will not occur for CCVTs and the second can be excluded through sufficient refinement. However, the number of boundary and interior regions can be arbitrary, so long as they add up to $K - M_c$. See Fig. 1 for an illustration.

We define a CCVT by solving the following problem:

$$\min_{\{\mathbf{z}_k, V_k\}_{k=1}^K} \mathcal{F}\big(\{\mathbf{z}_k, V_k\}_{k=1}^K\big) \quad \text{subject to} \quad \begin{cases} \mathbf{z}_m = \mathbf{z}_m^* & \text{for } m = 1, \ldots, M_c \\ g_j(\mathbf{z}_k) = 0 & \text{if } \overline{V}_k \cap \Gamma_j \neq \emptyset. \end{cases} \tag{8}$$

The subtle difference between the problems (7) and (8) is that for the former, the number of points on each boundary segment Γ_j is predetermined, while for the latter, they are determined as part of the solution of the optimization problem.

This type of CCVT is the least constrained of the three types we have introduced so that it is likely to produce the best distribution of points. Certainly, the value of the CVT energy for this type is lower than for the other two. In the sequel we will only consider this type of CCVT, although one could easily develop the other two types along the same lines.

2.3. Geometric view of CCVTs

CCVTs can also be defined geometrically. To this end, we need to define a *constrained centroid* or a *constrained center of mass* of a region $V \subset \mathbb{R}^N$. We use the notations introduced in Section 2.2. The standard center of mass $\bar{\mathbf{z}}$ of V is defined

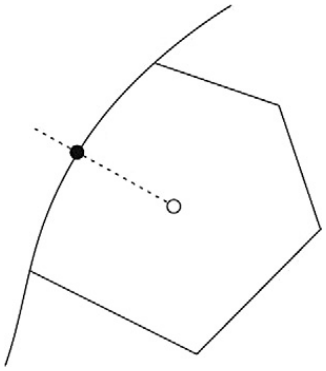

Fig. 2. The ordinary center of mass (open circle) of a region in $\mathbb{R}^2$ and its constrained center of mass (filled circle) with respect to the curve. The line segment joining the two centers of mass is perpendicular to the curve.

by (2) or, equivalently, (3). Given a C^1 surface segment γ, the constrained centroid $\bar{\mathbf{z}}^b$ of γ is defined to be the solution of the problem

$$\min_{\mathbf{z}\in\gamma} F(\mathbf{z}), \tag{9}$$

where $F(\cdot)$ is given as in (3). If γ can be described by the equation $g(\mathbf{x}) = 0$, then the problem (9) has the equivalent form

$$\min_{\mathbf{z}\in V} F(\mathbf{z}) \quad \text{subject to} \quad g(\mathbf{z}) = 0. \tag{10}$$

The following proposition, proven in [9], relates the position of the constrained mass centroid to that of the ordinary mass centroid.

Proposition 2. *Let $V \subset \mathbb{R}^N$ be a bounded measurable subset and let $\gamma \subset \mathbb{R}^N$ be a C^1 surface. Let $\rho(\mathbf{x})$ denote a given measurable density function defined on V that is positive almost everywhere. Then, the constrained centroid $\bar{\mathbf{z}}^b$ of V with respect to γ exists and is given by any point on γ such that the line segment joining that point and the mass centroid $\bar{\mathbf{z}}$ of V is normal to γ, i.e., $\bar{\mathbf{z}}^b$ is a projection of $\bar{\mathbf{z}}$ onto γ.*

Fig. 2 provides an illustration of a constrained mass centroid. Note that according to Proposition 2, multiple choices may exist for the constrained mass centroid, e.g., consider a circular region V. In such cases, we choose the one closest to the ordinary centroid (breaking ties arbitrarily) that is also in $\overline{V}$, and if no such point exists, we simply choose the one that is closest to the ordinary centroid. Note that Proposition 2 provides a framework for developing algorithms for determining a constrained center of mass of a region in $\mathbb{R}^N$.

The following theorem gives a geometric characterization of the generators of a CCVT as defined by the problem (8).

Theorem 1. *Let $\Omega \subset \mathbb{R}^N$ denote a bounded, open domain with a C^1 boundary Γ that is defined by the points $\mathbf{x} \in \mathbb{R}^N$ such that $g_1(\mathbf{x}) = 0$. Let $\rho(\cdot)$ denote a measurable density function defined on Ω that is positive almost everywhere. For an integer $K > 1$, let $\{\mathbf{z}_k\}_{k=1}^K$ denote any set of K points belonging to Ω and let $\{V_k\}_{k=1}^K$ denote any tessellation of Ω into K open subregions for which the number of tessellating subregions having a boundary that intersects the boundary Γ is equal to M. Without loss of generality, we assume that the subregions $\{V_k\}_{k=1}^M$ are the ones that intersect Γ. A necessary condition for $\{\mathbf{z}_k, V_k\}_{k=1}^K$ to be a CCVT of Ω is that each V_k is the Voronoi region corresponding to $\mathbf{z}_k$ and, simultaneously, each $\mathbf{z}_k$ is the constrained centroid of V_k.*

Proof. In the case we consider here, CCVTs are solutions of the problem (8) with $M_c = 0$ and $J = 1$. Solutions of that problem are stationary points of the Lagrangian functional

$$\mathcal{L}\big(\{(\mathbf{z}_i, V_i)\}_{i=1}^K, \{\lambda_m\}_{m=1}^M\big) = \sum_{i=1}^K \int_{V_i} \rho(\mathbf{x})|\mathbf{x} - \mathbf{z}_i|^2\, d\mathbf{x} \; - \sum_{m=1}^M \lambda_m g_1(\mathbf{z}_m)^2,$$

with respect to variations in the points $\{\mathbf{z}_k\}_{k=1}^K$, the tessellating subregions $\{V_k\}_{k=1}^K$, and the Lagrange multipliers $\{\lambda_m\}_{m=1}^M$.

Suppose first that $k > M$ so that V_k is an interior region, i.e., $\overline{V}_k$ does not intersect Γ. Then, as in [6], one easily finds that

$$\mathbf{z}_k = \bar{\mathbf{z}}_k = \frac{\int_{V_k} \mathbf{x}\rho(\mathbf{x})\, d\mathbf{x}}{\int_{V_k} \rho(\mathbf{x})\, d\mathbf{x}} \quad \text{for } k = M+1, \ldots, K.$$

If $k \leqslant M$ so that V_k is a boundary region, we find, as in Proposition 2, that $\mathbf{z}_k = \mathbf{z}_k^b$, i.e.,

$$g_1(\mathbf{z}_k) = 0 \quad \text{and} \quad \mathbf{z}_k = \bar{\mathbf{z}}_k^b = \bar{\mathbf{z}}_k - \frac{\lambda_k}{2\int_{V_k} \rho(\mathbf{x})\, d\mathbf{x}} \nabla_{\mathbf{z}} g_1(\mathbf{z}_k) \quad \text{for } k = 1, \ldots, M.$$

The fact that V_k is the Voronoi region for $\mathbf{z}_k$ can be proven in exactly the same way as for the ordinary CVT case; see [6]. □

For the sake of simplicity, we have assumed that the domain Ω has no corners or edges; a similar result can easily be proved if indeed there are corners and some of the points $\{\mathbf{z}_k\}_{k=1}^K$ are constrained to lie on corners and other points are forced to lie on edges if their Voronoi regions have boundaries that include corners and edges.

2.4. Algorithms for constructing CCVTs

We consider two algorithms for constructing CCVTs. The first was discussed in [8,9] and is a modification of Lloyd's method for constructing ordinary CVTs; see [6] for a discussion of Lloyd's method in the latter context.

Algorithm 1 *(CCVT).* Given a bounded, open domain $\Omega \subset \mathbb{R}^2$, a density function $\rho(\mathbf{x})$ defined for all $\mathbf{x} \in \Omega$, and a positive integer K,

0. select an initial set of K points $\{\mathbf{z}_i\}_{i=1}^K$ in $\overline{\Omega}$, e.g., by uniform random sampling or by sampling a superimposed equilateral triangular grid or by constructing an ordinary CVT;
1. construct the Voronoi tessellation $\{V_i\}_{i=1}^K$ of Ω associated with $\{\mathbf{z}_i\}_{i=1}^K$;
2. compute the (ordinary) mass centroids of the Voronoi regions $\{V_i\}_{i=1}^k$ found in step 1;
3. move the points $\{\mathbf{z}_i\}_{i=1}^k$ to the centroid positions;
4. determine the boundary and corner Voronoi regions;
5. if V_k is a boundary region, move $\mathbf{z}_k$ to its projection onto the boundary Γ;
6. if V_k is a corner region, move $\mathbf{z}_k$ to the corner;
7. if the new points meet some convergence criterion, terminate; otherwise, return to step 1.

We have found that we obtain slightly better results if we amend this algorithm by defining the Voronoi regions for an extension Ω_ϵ of the domain Ω instead of for Ω itself. The centroids nearest the boundary of an unconstrained CVT tend to line up with the boundary at a nearly uniform distance from the boundary. In Algorithm 1, these centroids are pulled to the boundary resulting in a certain amount of mesh distortion. By using a slightly larger domain, in Algorithm 2 described below, an attempt is made to find a slightly larger region for which the unconstrained centroids will nearly fall on the original boundary.

The effect of using an extended domain is that the (ordinary) centroids of the boundary Voronoi regions are located nearer the boundary of the original domain Ω than when one uses Algorithm 1. Of course, we still project those centroids onto the boundary of Ω since that is where we want the boundary points to be located. We now describe the amended algorithm.

Given $\epsilon > 0$ and a bounded, closed domain $\Omega \subset \mathbb{R}^2$, *the ϵ-expansion Ω_ϵ of Ω* is a set of points whose distances to Ω are less than or equal to ϵ. In Algorithm 2, ϵ is determined adaptively. To keep things simple, we consider the case of constant densities.

Algorithm 2 *(Modified CCVT).* Given a bounded domain $\Omega \subset \mathbb{R}^N$ and a mesh size δ, set $\epsilon = 0$ and $\Omega_\epsilon = \Omega$. Then,

0. select an initial set of K points $\{\mathbf{z}_i\}_{i=1}^K$ in $\overline{\Omega}$, e.g., by uniform random sampling or by sampling a superimposed equilateral triangular grid;
1. construct the (ordinary) CVT of Ω_ϵ as follows:
   i. determine the Voronoi tessellation $\{V_i\}_{i=1}^K$ of Ω_ϵ corresponding to the points $\{\mathbf{z}_i\}_{i=1}^K$;
   ii. determine the (ordinary) centers of mass of the Voronoi regions $\{V_i\}_{i=1}^K$;
   iii. move the points $\{\mathbf{z}_i\}_{i=1}^K$ to the centroid positions;
   iv. if the new points meet some convergence criteria, go to step 2; otherwise, return to step 1i;
2. determine ϵ_{new}, the average of the distances from the centroids of each boundary Voronoi region of step 1 to the furthest point of that region;
3. if $|\epsilon_{new} - \epsilon|$ is less than a prescribed tolerance, go to step 4; otherwise, set $\epsilon = \epsilon_{new}$, determine the ϵ expansion Ω_ϵ of the domain Ω, and return to step 1;
4. use Algorithm 1 to construct the CCVT on the ϵ-expansion Ω_ϵ of Ω, except that in step 5 of that algorithm, the centroids of the boundary Voronoi regions are projected onto the boundary of Ω (and not the boundary of Ω_ϵ).

3. Measures of mesh quality

In order to effect comparisons between triangulations obtained using the algorithms of Sections 2 and 4, we use several quality measures of mesh uniformity that have been suggested in the literature. Here, we briefly discuss the quality measures we use, confining ourselves to the uniform grid case. All measures apply to a given set of K points $\{\mathbf{z}_k\}_{k=1}^K$ belonging

to the region $\overline{\Omega} \in \mathbb{R}^2$. See [1,3,13,18–20] for additional details. For any point set, all the mesh quality measures we describe are non-negative and have an ideal value zero. Thus, in all cases, the smaller the value of the measure, the more uniform is the point distribution, at least according to the measure being used. Note that although the definitions of some of the quality measures are given for the planar case only, they can be extended to apply to meshes in higher dimensions.

The definition of some of the quality measures of the uniformity of point sets involve moments of regions in Euclidean space. Given a region V in N-dimensional Euclidean space, the *zeroth moment* or *volume* of V (a scalar) is defined by

$$|V| = \int_V d\mathbf{x},$$

the *first moment* or the *center of mass* or the *centroid* of V (a vector) is defined by

$$\bar{\mathbf{z}} = \frac{1}{|V|} \int_V \mathbf{x}\, d\mathbf{x},$$

the *second moment* (relative to its center of mass) of V (a tensor) is defined by

$$\mathbb{M} = \frac{1}{|V|} \int_V (\mathbf{x} - \bar{\mathbf{z}})(\mathbf{x} - \bar{\mathbf{z}})^T\, d\mathbf{x}.$$

The deviatoric tensor associated with V is given by

$$\mathbb{D} = \mathbb{M} - m\mathbb{I},$$

where $\mathbb{I}$ denotes the identity tensor, $m = T/N$, and $T = \text{trace}(\mathbb{M})$.

3.1. Quality measures based on the coordinates of the points

We first consider measures that depend directly on the coordinates of the points in the given set $\{\mathbf{z}_k\}_{k=1}^K$.

The covariance (COV) measure λ. Given a set of K points $\{\mathbf{z}_i\}_{i=1}^K$ in $\mathbb{R}^2$, let

$$\gamma_k = \min_{j=1,\ldots,K,\, j\neq k} |\mathbf{z}_k - \mathbf{z}_j| \quad \text{for } k = 1, \ldots, K \quad \text{and} \quad \overline{\gamma} = \frac{1}{K}\sum_{k=1}^K \gamma_k$$

so that γ_k is the minimum distance between the point $\mathbf{z}_k$ and any of the other points. Then, the *COV (covariance) measure* λ is given by

$$\lambda = \frac{1}{\overline{\gamma}}\left(\frac{1}{K}\sum_{k=1}^K (\gamma_k - \overline{\gamma})^2\right)^{1/2} = \left(-1 + \frac{1}{\overline{\gamma}^2 K}\sum_{k=1}^K \gamma_k^2\right)^{1/2} = \left(K\frac{\sum_{k=1}^K \gamma_k^2}{(\sum_{k=1}^K \gamma_k)^2} - 1\right)^{1/2}.$$

For an ideal uniform mesh, $\gamma_1 = \gamma_2 = \cdots = \gamma_K = \overline{\gamma}$ so that $\lambda = 0$. Thus, the smaller the value of λ, the more uniform the mesh.

The mesh ratio γ. Given a set of K points $\{\mathbf{z}_k\}_{k=1}^K$ in $\mathbb{R}^2$, the *mesh ratio* γ is given by

$$\gamma = \left(\frac{\max_{k=1,\ldots,K} \gamma_k}{\min_{k=1,\ldots,K} \gamma_k}\right) - 1.$$

For an ideal uniform mesh, $\gamma_1 = \gamma_2 = \cdots = \gamma_K$ so that, ideally, $\gamma = 0$. Thus, the smaller the value of γ, the more uniform is the point distribution.

3.2. Quality measures based on the Voronoi regions

Given a set of K points $\{\mathbf{z}_k\}_{k=1}^K$ in a region $\overline{\Omega}$, we can use those points to generate a Voronoi tessellation $\{V_k\}_{k=1}^K$ of Ω. We can then associate with each point a corresponding Voronoi region and use those regions to determine various quantities that can be used to measure the quality of the set of points.

The point distribution measure h. Given a Voronoi tessellation $\{\mathbf{z}_k, V_k\}_{k=1}^K$, let

$$\widetilde{h} = \max_{k=1,\ldots,K} h_k, \quad \text{where } h_k = \max_{\mathbf{y}\in V_k} |\mathbf{z}_k - \mathbf{y}| \text{ for } k = 1, \ldots, K.$$

Thus, h_k gives the maximum distance between the particular generator $\mathbf{z}_k$ and the points in its associated cell V_k and $\widetilde{h}$ gives the maximum distance between any generator and the points in its associated Voronoi cell. The ideal value h^* of $\widetilde{h}$ is that for a tessellation into congruent regular hexagons in which case $h^* = (\sqrt{12}|\Omega|/9K)^{1/2} \approx 0.6204(|\Omega|/K)^{1/2}$, where $|\Omega|$ denotes the area of Ω. The *point distribution measure* h is then given by

$$h = \frac{\widetilde{h}}{h^*} - 1 = \left(\left(\frac{9K}{\sqrt{12}|\Omega|}\right)^{1/2}\widetilde{h}\right) - 1.$$

Ideally, $h = 0$ so that the smaller the value of h, the more uniform is the point distribution.

The point distribution ratio μ. Given a Voronoi tessellation $\{\mathbf{z}_k, V_k\}_{i=k}^{K}$, the *point distribution ratio* μ is given by

$$\mu = \left(\frac{\max_{k=1,\ldots,K} h_k}{\min_{k=1,\ldots,K} h_k}\right) - 1.$$

For an ideal uniform point set, $\mu = 0$ so that the smaller the value of μ, the more uniform is the point distribution.

The regularity measure χ. Given a Voronoi tessellation $\{\mathbf{z}_k, V_k\}_{k=1}^{K}$, the *regularity measure* χ is given by

$$\chi = \left(\max_{k=1,\ldots,K} \chi_k\right) - 1, \quad \text{where } \chi_k = \frac{\sqrt{3}h_k}{\gamma_k} \text{ for } k = 1, \ldots, K.$$

For an ideal uniform, regular hexagonal mesh, $h_k = \widetilde{h}$ and $\gamma_k = \sqrt{3}\widetilde{h}$ for all k and thus $\chi_k = 1$ for all k so that, ideally, $\chi = 0$. Thus, the smaller χ is, the more uniform is the point distribution. In addition, the value of χ provides us a measure of the mesh regularity, i.e., the *local uniformity* of a mesh. Again, if a mesh is locally uniform in the sense that the cells in a neighborhood of any cell are nearly congruent to that cell, then the value of χ will again be small.

Cell volume deviation ν. Given a Voronoi tessellation $\{\mathbf{z}_k, V_k\}_{k=1}^{K}$, the *cell volume deviation* ν is given by

$$\nu = \left(\frac{\max_{k=1,\ldots,K} |V_k|}{\min_{k=1,\ldots,K} |V_k|}\right) - 1,$$

where $|V_k|$ denotes the volume of the Voronoi cell V_k. For an ideal uniform mesh, the volumes $|V_k|$ would all be equal, i.e., $|V|_1 = |V|_2 = \cdots = |V|_K$ so that $\nu = 0$. Thus, the smaller the value of ν, the more uniform is the point distribution. Note that both boundary and interior Voronoi cells are included in the determination of ν; smaller values of ν can be obtained for all methods if one discriminates between the two types of cells.

The second moment trace measure τ. Given a Voronoi tessellation $\{\mathbf{z}_k, V_k\}_{k=1}^{K}$, let τ_k denote the trace of the second moment tensor associated with each Voronoi region V_k. Let $\overline{\tau} = \frac{1}{K}\sum_{k=1}^{K} \tau_k$ denote the average of the trace over the K regions. Then, the *second moment trace measure* τ is given by

$$\tau = \max_{k=1,\ldots,K} |\tau_k - \overline{\tau}|.$$

For an ideal uniform mesh, $\tau_1 = \tau_2 = \cdots = \tau_K = \overline{\tau}$ so that, ideally, $\tau = 0$. Thus, the smaller the value of τ, the more uniform is the point distribution.

The second moment determinant measure d. Given a Voronoi tessellation $\{\mathbf{z}_k, V_k\}_{k=1}^{K}$, let D_k denote the determinant of the deviatoric tensor associated with each Voronoi region V_k. Then, the *second moment determinant measure* d is given by

$$d = \max_{k=1,\ldots,K} |D_k|.$$

For an ideal uniform mesh, $D_1 = D_2 = \cdots = D_K = 0$ so that $d = 0$. Thus, the smaller the value d, the more uniform is the mesh.

3.3. Quality measures based on the Delaunay triangulation

Given a set of K points $\{\mathbf{z}_k\}_{k=1}^{K}$ in a region Ω, we can use those points to generate a Delaunay triangulation $\{\triangle_j\}_{j=1}^{\widetilde{K}}$ of Ω. We can use the triangles $\triangle_j$, $j = 1, \ldots, \widetilde{K}$, to determine various quantities that can be used to measure the quality of the set of points. If the points are to be used as nodes for, e.g., a finite element discretization of a partial differential equation, then perhaps these measures are of the most direct relevance since they directly involve the triangles.

Maximum area measure α**.** Given a Delaunay triangulation $\{\triangle_j\}_{j=1}^{\widetilde{K}}$ corresponding to the point set $\{\mathbf{z}_k\}_{k=1}^{K}$, let $|\triangle_j|$ denote the area of the triangle $\triangle_j$. Then, the *maximum area measure* is defined by

$$\alpha = \left(\frac{\widetilde{K}}{|\Omega|} \max_{j=1,\ldots,\widetilde{K}} |\triangle_j| \right) - 1.$$

For an ideal uniform mesh, $|\triangle_1| = |\triangle_2| = \cdots = |\triangle_{\widetilde{K}}| = |\Omega|/\widetilde{K}$ so that, ideally, $\alpha = 0$. Thus, the smaller the value of α, the more uniform is the point distribution.

Minimum angle measure β**.** Given a Delaunay triangulation $\{\triangle_j\}_{j=1}^{\widetilde{K}}$ corresponding to the point set $\{\mathbf{z}_k\}_{k=1}^{K}$, let β_j denote the minimum angle of the triangle $\triangle_j$. Note that $\beta_j \leqslant \pi/3$ radians. Then, the *minimum angle measure* is defined by

$$\beta = \left(\frac{\pi/3}{\min_{j=1,\ldots,\widetilde{K}} \beta_j} \right) - 1.$$

For an ideal uniform mesh, $\beta_1 = \beta_2 = \cdots = \beta_{\widetilde{K}} = \beta = \pi/3$ radians so that, ideally, $\beta = 0$. Thus, the smaller the value of β, the more uniform is the point distribution.

Circle ratio measure q**.** Given a Delaunay triangulation $\{\triangle_j\}_{j=1}^{\widetilde{K}}$ corresponding to the point set $\{\mathbf{z}_k\}_{k=1}^{K}$, let q_j denote half the ratio of the radius r_j of the inscribed circle to the radius R_j of the circumscribed circle of the triangle $\triangle_j$, i.e.,

$$q_j = \frac{R_j}{2r_j} = \frac{abc}{(b+c-a)(c+a-b)(a+b-c)} \quad \text{for } j = 1, \ldots, \widetilde{K},$$

where a, b, and c denote the lengths of the sides of the triangle $\triangle_j$. Note that $q_j \geqslant 1$. Then, the *circle ratio measure* is given by

$$q = \left(\max_{j=1,\ldots,\widetilde{K}} q_j \right) - 1.$$

For an ideal uniform mesh, $q_1 = q_2 = \cdots = q_{\widetilde{K}} = 1$ so that, ideally, $q = 0$. Thus, the smaller the value of q, the more uniform is the point distribution.

The normalized standard deviation measure p**.** Given a Delaunay triangulation $\{\triangle_j\}_{j=1}^{\widetilde{K}}$ corresponding to the point set $\{\mathbf{z}_k\}_{k=1}^{K}$, let R_j denote the radius of the circumscribed circle. Let

$$\overline{R} = \frac{1}{\widetilde{K}} \sum_{j=1}^{\widetilde{K}} R_j \quad \text{and} \quad R_{\text{std}} = \text{standard deviation of } R_j \text{ over } j = 1, \ldots, \widetilde{K}.$$

Then, the *normalized standard deviation measure* is given by

$$p = \frac{R_{\text{std}}}{\overline{R}}.$$

For an ideal uniform mesh, $R_1 = R_2 = \cdots = R_{\widetilde{K}} = \overline{R}$ so that $p = 0$. Thus, the smaller the value of p, the more uniform is the point distribution.

4. Computational experiments

We now turn to the second goal of the paper, namely, comparing several methods for generating uniform triangular meshes on general regions in $\mathbb{R}^2$. We will apply the methods to a variety of test problems, using several measures of quality to evaluate their relative merits; the specific measures we use are those described in Section 3. We first list the mesh generation methods we will test in addition to, of course, CCVT.

TRIANGLE. A well-known triangular mesh generator is the TRIANGLE method of [17–19]. An initial Delaunay triangulation is refined by halving edges and/or inserting circumcenters in such a way that triangles having an area greater than a specified area are subdivided and angles smaller than a specified angle are eliminated.

DISTMESH. In the DISTMESH method, a Delaunay triangulation is viewed as a system of point masses connected by springs. The point masses are moved from an initial position so that a static equilibrium is achieved.

MESHGEN. The MESHGEN method is a variant of CVT methods; see [14,16]. Voronoi regions are approximated by the easier-to-construct region formed by joining the circumcenters of acute triangles and the mid-sides of the longest sides of obtuse triangles that surround a vertex in the triangulation.

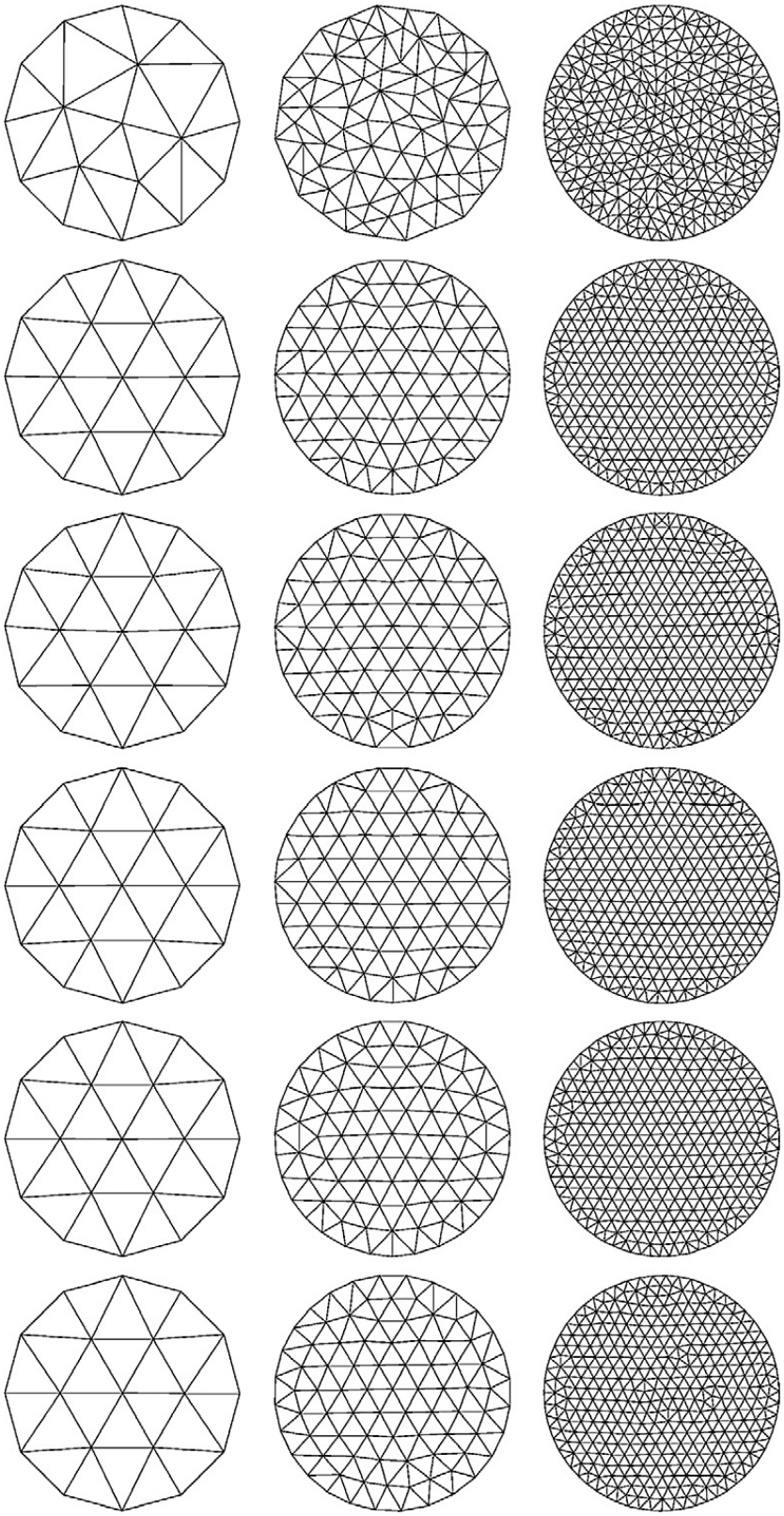

Fig. 3. Meshes for a unit circle; from top to bottom: TRIANGLE, DISTMESH, MESHGEN, VTM, CCVT-Algorithm1, CCVT-Algorithm2.

VTM. The variational tetrahedral meshing (VTM) method of [1] is another variant of CVT methods. Instead of working with Voronoi regions associated with the points, one instead considers patches of Delaunay triangles that surround each point. Unlike the CVT case, the cells associated with the points overlap.

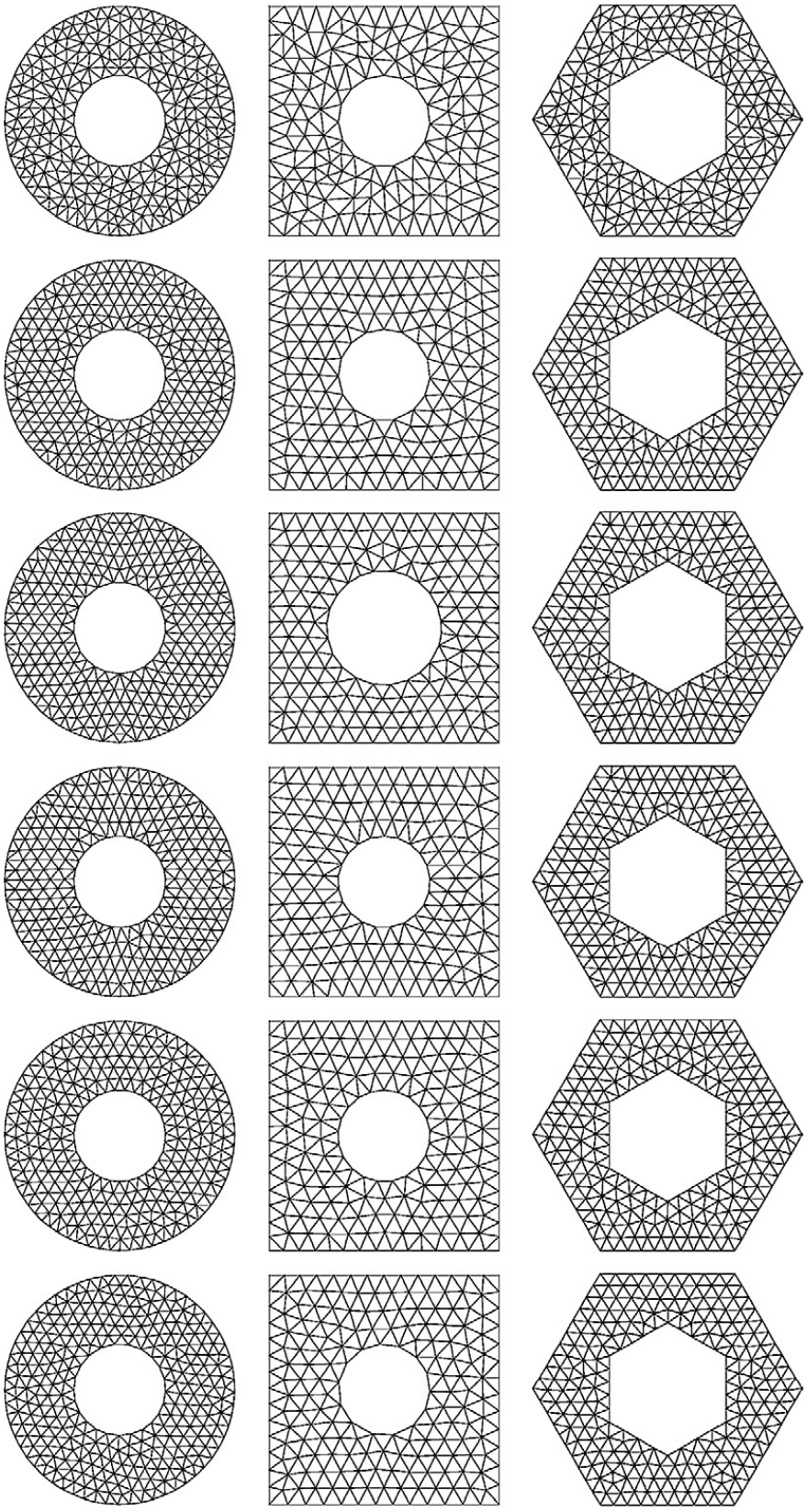

Fig. 4. Meshes for an annulus, a square with a circular hole, and hexagon with a hexagonal hole: from top to bottom: TRIANGLE, DISTMESH, MESHGEN, VTM, CCVT-Algorithm1, CCVT-Algorithm2.

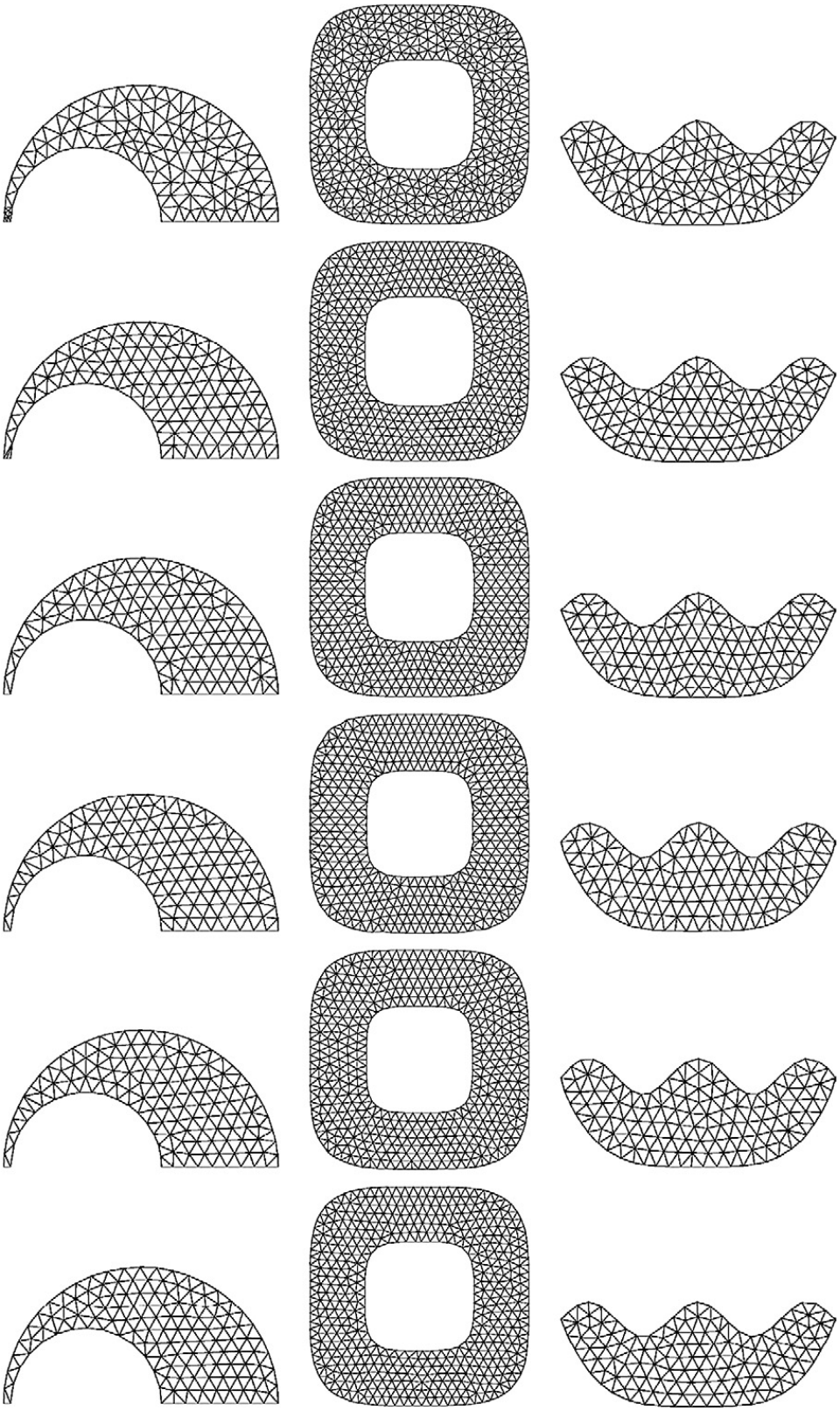

Fig. 5. Meshes for a “horn”, a super-ellipse, and a “bicycle seat”; from top to bottom: TRIANGLE, DISTMESH, MESHGEN, VTM, CCVT-Algorithm1, CCVT-Algorithm2.

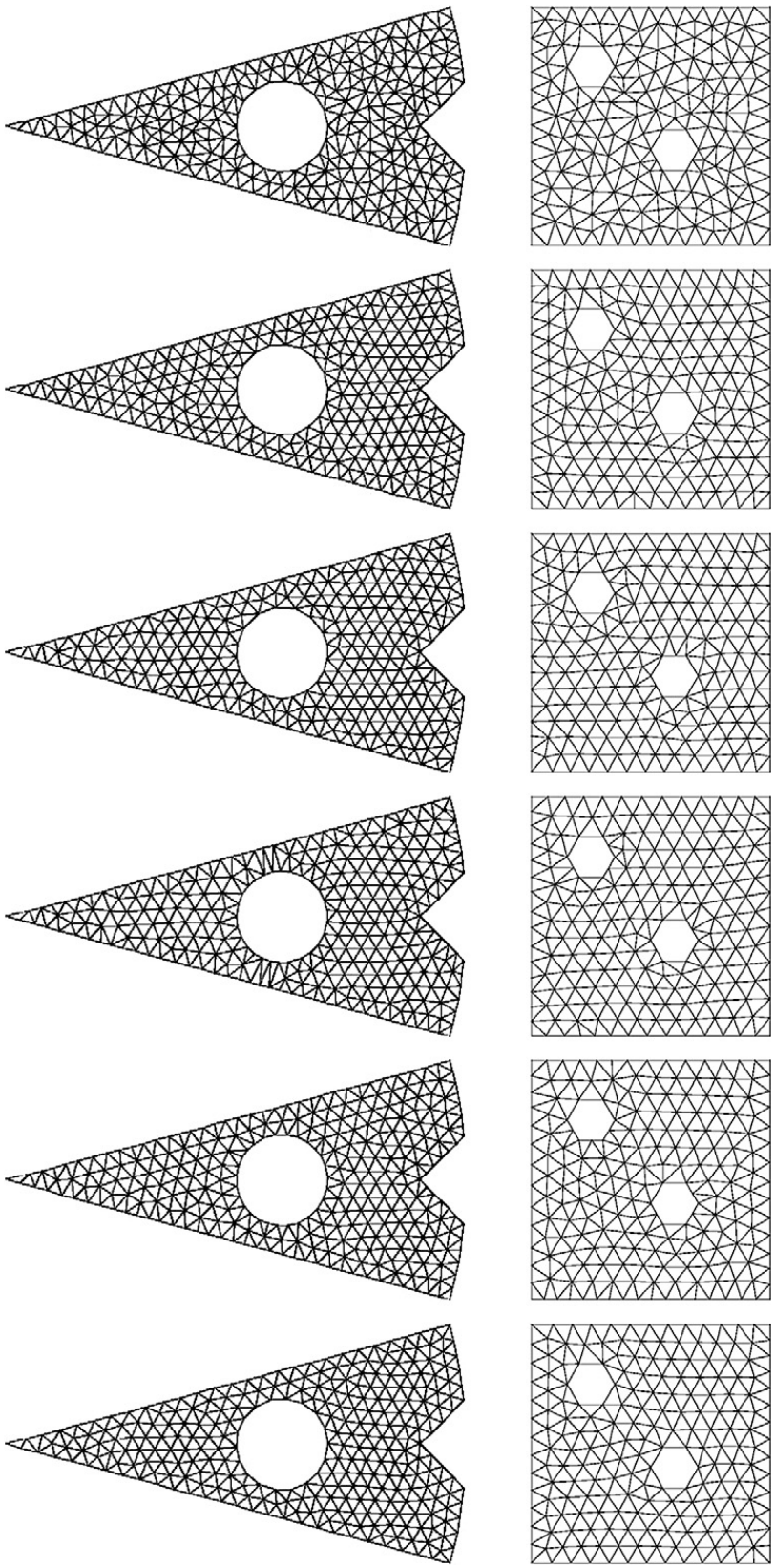

Fig. 6. Meshes for a "pie-slice" with a circular hole and a square with 2 hexagonal holes; from top to bottom: TRIANGLE, DISTMESH, MESHGEN, VTM, CCVT-Algorithm1, CCVT-Algorithm2.

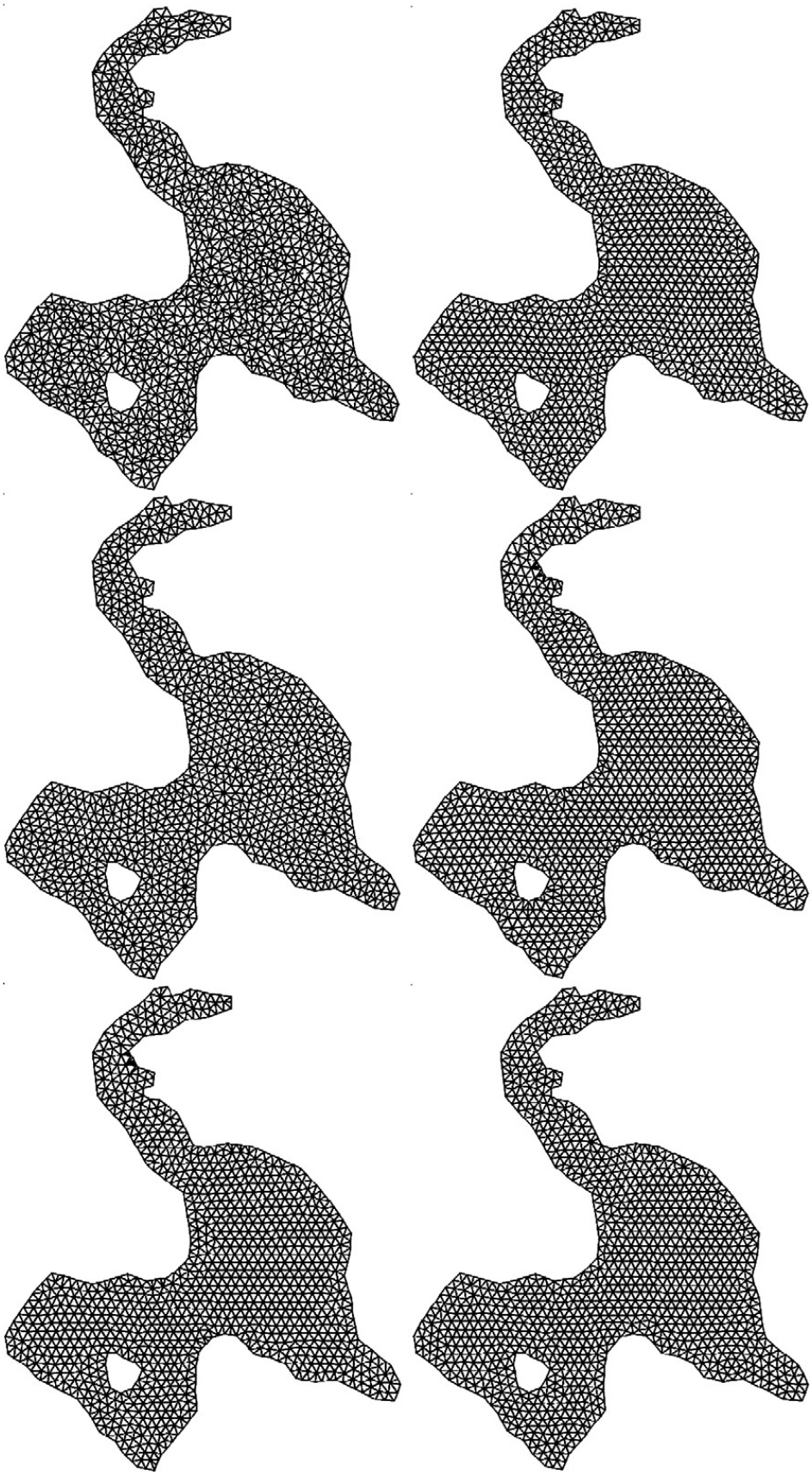

Fig. 7. Meshes for a "lake" with an island; from left to right, top to bottom: TRIANGLE, DISTMESH, MESHGEN, VTM, CCVT-Algorithm1, CCVT-Algorithm2.

Table 1
Grid-quality measures for circle (coarsest mesh)

	TRIANGLE	DISTMESH	MESHGEN	VTM	CCVT(1)	CCVT(2)
K	18	19	19	19	19	19
$\widetilde{K}$	22	24	24	24	24	24
$\lambda \times 10^2$	0.38	5.15	5.33	5.75	4.77	4.80
$\gamma \times 10$	0.10	1.38	1.60	1.59	1.23	1.23
$h \times 10$	6.12	2.83	2.89	2.83	2.86	2.87
$\mu \times 10$	2.08	1.26	1.07	1.38	1.14	1.13
$\chi \times 10$	4.05	2.30	2.37	2.40	2.23	2.24
ν	1.93	1.66	1.68	1.65	1.62	1.60
$\tau \times 10^3$	6.49	3.29	3.24	3.36	3.33	3.34
$d \times 10^6$	2.50	3.41	3.90	3.33	3.31	3.30
$\alpha \times 10$	6.16	0.99	1.11	0.97	0.96	0.95
$\beta \times 10$	3.41	2.40	2.68	2.41	2.27	2.25
$q \times 10$	2.05	0.64	0.75	0.53	0.53	0.53
$p \times 10^2$	11.50	3.60	3.70	3.50	3.70	3.70

There are triangulation methods in addition to those listed above, e.g., the Laplacian smoothing method proposed in [12] that is useful for improving mesh quality and the ODT (optimal Delaunay triangulation) method proposed in [4]. ODTs minimize the interpolation error among all triangulations with the same number of vertices. It is very similar to CVTs, but there is so far no convergence theory available for the local patch iteration used in computing ODTs such as is available for Lloyd's algorithm for CVTs [5]. The same comment applies to the Laplace smoothing method; in this regard, only CVT is a proven technology.

4.1. Results and discussion

Uniform triangulations of several regions in the plane have been determined using the methods just listed as well as the two CVT algorithms presented in Section 2.4. Many of these test cases were drawn from [17–20]. These examples contain features, e.g., acute and obtuse corners, non-convexity, holes, etc., that are representative of what one encounters in practice. Figs. 3–7 provide means for visually comparing the meshes produced by the various methods. Quantitative comparison using the 12 grid-quality measures listed in Section 3 are found in Tables 1–12. In the tables, K and $\widetilde{K}$ refer to the number of vertices and the number of triangles, respectively, in the triangulations.[3]

From the figures and tables, one concludes that all the methods tested yield good results. In most cases (with perhaps the TRIANGLE algorithm being the exception), both visual and quantitative comparisons show that the quality of the grids produced by the methods is pretty much indistinguishable. If one were to keep score of which method is best for each domain and for each quality measure, one finds that DISTMESH and the second CCVT algorithm do "best", with the latter being slightly "better". However, the differences between the performance of the methods (with again the TRIANGLE algorithm being a possible exception) are statistically insignificant. What does seem to be clear is that CVT-based uniform triangulations of planar regions are at least as good as those generated using the other methods considered.

In this paper, we presented two CVT-based algorithms for the triangulation of general planar regions. The reason we have confined ourselves to examples involving uniform triangulations is that, for this case, several quantitative mesh-quality measures are available in the literature. This enabled us to not only make visual comparisons between different triangulation methods, but to also make perhaps more discriminating numerical comparisons. For more general grid generation settings, e.g., non-uniform grids, such quantitative comparisons are much more difficult, if not impossible, to either make or interpret. However, most (but not all) of the methods considered have been or can be extended to at least some additional settings such as three-dimensional grid generation, surface grid generation, non-uniform grid generation, and anisotropic grid generation, including, in the last two cases, adaptive mesh refinement. It is possible that noticeable quality differences between the methods may appear if one compares them for these settings. CVT-based grid generation has been extended in all of these directions, with some results being presented in [2,9–11,14,16]. Current efforts are being devoted to the further development of CVT-based grid generation algorithms for all the settings just listed.

[3] In the figures and tables, the number of vertices for the results for TRIANGLE are often different from those for the other methods. The cause of this difference is that for TRIANGLE, unlike what is the case for the other methods, the user does not have direct control of the number of vertices. We have endeavored to adjust the parameters used in TRIANGLE so that the number of vertices is very close to those used for the other methods. An additional note about TRIANGLE is that, unlike the other methods, it implements a strategy that attempts to avoid having three vertices of a triangle to all be boundary points so that no triangle is missed in a finite element method. That is the reason for relatively high concentration of points in the top-left figure in Fig. 5. As the number of points gets larger and larger, this feature will disappear. We point this out since it may account for the unusually poor relative performance of TRIANGLE for this example; see Table 7.

Table 2
Grid-quality measures for circle (medium mesh)

	TRIANGLE	DISTMESH	MESHGEN	VTM	CCVT(1)	CCVT(2)
K	86	88	88	88	88	88
$\widetilde{K}$	146	143	144	143	143	146
$\lambda \times 10$	1.10	0.61	0.83	1.02	0.57	0.59
$\gamma \times 10$	5.84	2.19	2.92	4.51	2.02	2.94
$h \times 10$	4.93	2.66	2.32	2.66	2.35	2.47
$\mu \times 10$	5.21	2.12	1.48	2.71	1.68	2.15
$\chi \times 10$	7.45	3.48	4.01	4.54	3.02	3.35
ν	2.47	1.83	2.12	2.42	1.95	1.59
$\tau \times 10^4$	6.87	6.33	6.16	6.36	6.51	5.02
$d \times 10^7$	2.29	0.91	1.05	0.98	1.00	1.30
$\alpha \times 10$	4.82	1.62	1.55	2.51	2.18	2.14
β	1.01	0.49	0.49	0.52	0.43	0.48
q	1.04	0.37	0.25	0.37	0.20	0.28
$p \times 10^2$	12.30	4.00	3.60	4.30	3.50	3.80

Table 3
Grid-quality measures for circle (finest mesh)

	TRIANGLE	DISTMESH	MESHGEN	VTM	CCVT(1)	CCVT(2)
K	362	362	362	362	362	362
$\widetilde{K}$	661	661	657	650	651	661
$\lambda \times 10$	0.98	0.51	0.79	1.07	0.85	0.51
$\gamma \times 10$	5.48	2.56	4.39	5.47	3.75	3.18
$h \times 10$	4.85	1.42	1.87	1.65	1.80	1.66
$\mu \times 10$	8.75	1.48	2.42	3.84	2.10	1.86
$\chi \times 10$	8.89	3.04	4.76	5.46	5.26	2.82
ν	3.25	1.73	2.24	3.14	2.26	1.86
$\tau \times 10^4$	2.06	1.54	1.84	2.03	1.85	1.51
$d \times 10^8$	1.09	0.60	0.63	0.47	0.45	0.63
$\alpha \times 10$	5.15	1.36	2.55	2.25	2.60	1.87
β	1.22	0.44	0.66	0.72	0.74	0.40
$q \times 10$	9.61	2.66	3.51	5.15	3.70	2.20
$p \times 10^2$	13.30	2.20	2.80	3.70	2.90	2.90

Table 4
Grid-quality measures for an annulus

	TRIANGLE	DISTMESH	MESHGEN	VTM	CCVT(1)	CCVT(2)
K	303	303	303	303	303	303
$\widetilde{K}$	520	520	519	507	508	524
$\lambda \times 10$	0.75	0.49	0.72	1.00	0.94	0.39
$\gamma \times 10$	5.21	3.03	5.77	6.18	5.81	2.81
$h \times 10$	5.34	2.58	2.99	3.80	2.62	2.29
$\mu \times 10$	6.63	2.39	3.08	4.08	2.29	2.22
$\chi \times 10$	8.83	3.43	5.12	9.86	7.20	2.71
ν	3.46	1.90	2.35	3.30	2.03	1.75
$\tau \times 10^4$	2.34	1.59	1.98	1.93	1.70	1.36
$d \times 10^9$	8.69	6.80	8.44	5.66	6.00	8.41
$\alpha \times 10$	4.93	2.30	1.91	4.68	1.98	2.36
β	1.00	0.51	0.71	1.32	1.02	0.40
q	7.54	3.33	4.49	9.61	3.16	2.20
$p \times 10^2$	12.20	3.20	4.70	6.50	3.70	3.80

Table 5
Grid-quality measures for a square with a circular hole

	TRIANGLE	DISTMESH	MESHGEN	VTM	CCVT(1)	CCVT(2)
K	193	194	194	194	194	194
$\widetilde{K}$	313	315	308	308	308	320
$\lambda \times 10$	0.83	0.64	0.99	0.96	1.02	0.50
$\gamma \times 10$	4.51	3.19	5.32	6.80	4.91	3.57
$h \times 10$	5.54	2.24	5.03	4.34	2.69	2.66
$\mu \times 10$	5.63	1.73	5.27	4.33	2.69	2.07
$\chi \times 10$	8.31	4.19	10.37	8.22	5.62	3.90
ν	6.03	4.87	5.87	6.60	5.71	4.77
$\tau \times 10^3$	2.50	2.39	2.48	2.44	2.39	2.35
$d \times 10^7$	9.02	5.77	5.94	2.75	1.88	7.44
$\alpha \times 10$	5.18	1.84	1.24	5.65	2.77	2.33
β	1.13	0.65	0.72	0.86	0.77	0.55
$q \times 10$	6.67	2.50	4.71	6.67	3.33	2.82
$p \times 10^2$	12.90	2.90	4.60	7.10	3.60	4.20

Table 6
Grid-quality measures for a hexagon with a hexagonal hole

	TRIANGLE	DISTMESH	MESHGEN	VTM	CCVT(1)	CCVT(2)
K	242	240	240	240	240	240
$\widetilde{K}$	412	385	385	381	381	393
$\lambda \times 10$	1.10	0.62	0.90	0.91	0.84	0.53
$\gamma \times 10$	6.91	3.81	4.98	4.27	4.25	2.99
$h \times 10$	6.91	3.81	4.98	4.27	4.25	2.99
$\mu \times 10$	5.45	2.07	2.29	3.89	1.96	2.62
$\chi \times 10$	9.45	5.17	6.06	7.46	5.62	4.90
ν	4.21	3.01	3.48	3.55	3.40	3.31
$\tau \times 10^4$	8.83	9.25	9.19	8.98	8.94	8.47
$d \times 10^7$	3.36	1.53	1.09	0.94	0.72	1.72
$\alpha \times 10$	4.68	2.81	2.15	4.65	2.80	2.10
β	0.99	0.69	0.86	1.00	0.74	0.53
q	1.00	0.37	0.30	0.69	0.39	0.37
$p \times 10^2$	11.60	3.60	4.60	6.00	4.30	4.20

Table 7
Grid-quality measures for a “horn”-shaped region

	TRIANGLE	DISTMESH	MESHGEN	VTM	CCVT(1)	CCVT(2)
K	137	138	138	138	138	138
$\widetilde{K}$	213	219	216	214	214	224
$\lambda \times 10$	2.09	1.30	1.18	1.20	1.01	0.93
γ	4.42	3.99	1.21	1.22	1.24	1.32
$h \times 10$	6.19	3.06	4.06	3.70	3.93	2.40
μ	4.23	0.26	0.51	0.51	0.48	0.22
χ	1.68	4.21	1.54	0.93	0.93	1.16
ν	78.65	13.80	7.44	8.17	7.66	6.74
$\tau \times 10^3$	1.37	1.09	1.13	1.09	1.09	1.08
$d \times 10^7$	3.45	1.37	2.99	0.95	0.87	1.61
$\alpha \times 10$	7.24	2.90	4.79	3.42	3.27	2.58
β	0.96	5.16	1.90	1.10	1.12	1.37
q	1.08	2.57	0.89	0.61	0.64	0.49
$p \times 10^2$	20.30	6.50	8.10	6.00	5.50	3.70

Table 8
Grid-quality measures for a super-ellipse

	TRIANGLE	DISTMESH	MESHGEN	VTM	CCVT(1)	CCVT(2)
K	505	505	505	505	505	505
$\widetilde{K}$	882	882	868	864	862	873
$\lambda \times 10$	0.90	0.48	1.07	1.02	1.07	0.71
$\gamma \times 10$	5.09	3.45	5.63	6.19	6.84	8.72
$h \times 10$	4.88	2.07	2.67	4.21	3.10	2.64
$\mu \times 10$	6.20	1.83	2.70	4.59	2.87	2.47
$\chi \times 10$	7.93	3.82	7.28	8.35	8.45	9.90
ν	3.20	2.11	2.97	3.91	3.56	2.17
$\tau \times 10^4$	5.28	3.85	4.82	5.87	5.50	4.57
$d \times 10^8$	5.54	5.24	3.44	4.76	3.27	5.00
$\alpha \times 10$	5.21	1.94	3.39	4.67	2.42	2.39
β	0.94	0.46	1.12	1.14	0.88	1.06
q	0.96	0.27	0.41	1.08	0.59	0.33
$p \times 10^2$	11.40	3.00	4.30	6.00	4.00	3.80

Table 9
Grid-quality measures for a “bike” seat-shaped region

	TRIANGLE	DISTMESH	MESHGEN	VTM	CCVT(1)	CCVT(2)
K	137	136	136	136	136	136
$\widetilde{K}$	222	220	220	210	213	219
$\lambda \times 10$	0.75	0.53	0.74	1.29	1.12	1.08
$\gamma \times 10$	4.07	2.73	3.82	7.97	5.21	8.48
h	6.40	3.13	3.11	4.49	3.56	2.76
$\mu \times 10$	6.52	2.41	2.45	3.48	2.76	2.90
χ	0.82	0.39	0.48	1.04	0.65	0.86
ν	3.71	2.67	3.12	4.71	3.51	4.40
$\tau \times 10^2$	5.55	5.30	5.54	5.69	4.98	6.08
$d \times 10^4$	3.99	3.83	4.51	3.26	2.78	3.46
$\alpha \times 10$	5.31	2.27	1.85	5.04	4.04	2.10
β	1.00	0.55	0.70	1.30	0.90	1.04
$q \times 10$	8.87	2.66	3.16	6.67	4.71	4.08
$p \times 10^2$	13.70	3.70	4.50	6.40	4.50	4.60

Table 10
Grid-quality measures for a “pie slice”-shaped regions with a hole

	TRIANGLE	DISTMESH	MESHGEN	VTM	CCVT(1)	CCVT(2)
K	285	287	287	287	287	287
$\widetilde{K}$	470	474	479	452	451	484
$\lambda \times 10$	0.89	0.66	0.88	1.53	1.40	0.69
γ	0.44	0.38	0.63	1.13	0.81	0.57
$h \times 10$	5.99	3.98	4.42	7.97	4.06	3.32
$\mu \times 10$	6.96	4.78	3.28	7.66	2.92	2.43
χ	0.84	0.42	0.59	1.87	0.95	0.58
ν	21.47	18.73	12.55	20.70	15.15	11.95
$\tau \times 10^4$	1.24	1.20	1.13	1.14	1.12	1.14
$d \times 10^9$	2.81	1.84	1.19	0.97	0.51	1.37
α	0.71	0.36	0.66	1.04	0.49	0.54
β	1.00	1.00	1.00	2.42	1.28	1.00
q	1.13	1.13	0.75	1.08	1.04	0.56
$p \times 10^2$	11.70	4.30	5.50	10.30	5.80	4.10

Table 11
Grid-quality measures for a square with 2 hexagonal holes

	TRIANGLE	DISTMESH	MESHGEN	VTM	CCVT(1)	CCVT(2)
K	202	203	203	203	203	203
$\widetilde{K}$	336	338	339	334	332	343
$\lambda \times 10$	1.09	0.94	1.19	1.42	1.32	0.59
$\gamma \times 10$	8.33	7.15	8.59	8.75	8.18	3.22
$h \times 10$	5.65	3.86	2.88	3.34	3.62	3.16
μ	1.01	0.31	0.36	0.53	0.45	0.30
χ	1.07	0.82	1.05	1.17	0.95	0.47
ν	5.82	4.88	5.98	6.52	7.02	4.50
$\tau \times 10^4$	6.30	6.12	6.53	6.28	6.42	6.13
$d \times 10^8$	5.80	8.56	3.56	4.49	5.79	8.27
$\alpha \times 10$	5.76	4.65	3.45	4.04	4.30	4.03
β	1.00	0.95	1.09	1.60	1.10	0.53
$q \times 10$	5.38	3.70	4.08	8.18	5.15	2.50
$p \times 10^2$	13.10	3.60	4.10	6.20	4.90	4.20

Table 12
Grid-quality measures for a "lake" with an island

	TRIANGLE	DISTMESH	MESHGEN	VTM	CCVT(1)	CCVT(2)
K	1109	1109	1109	1109	1109	1109
$\widetilde{K}$	2020	1979	2004	1948	1945	1986
$\lambda \times 10$	1.02	0.89	0.94	1.51	1.35	0.80
γ	0.80	1.50	0.71	3.33	4.28	1.27
$h \times 10$	5.43	3.03	3.20	4.26	3.52	2.73
$\mu \times 10$	8.60	3.15	3.53	8.67	4.70	3.52
χ	0.97	1.40	0.82	2.92	4.14	1.20
ν	5.07	5.50	3.55	9.65	8.50	4.46
τ	14.11	14.89	12.97	15.71	14.71	13.49
d	71.85	32.15	49.78	42.46	39.06	31.36
$\alpha \times 10$	5.43	3.03	3.40	7.49	5.69	2.80
β	1.16	1.77	1.15	3.74	5.15	1.58
q	1.18	0.73	0.44	2.02	2.71	0.73
$p \times 10$	1.27	0.33	0.56	0.66	0.45	0.39

References

[1] P. Alliez, D. Cohen-Steiner, M. Yvinec, M. Desbrun, Variational tetrahedral meshing, in: Proc. ACM SIGGRAPH'05, 2005, pp. 617–625.
[2] P. Alliez, T. De Verdiere, O. Devillers, M. Isengurg, Centroidal Voronoi diagrams for isotropic surface remeshing, Graph. Model 67 (2005) 204–231.
[3] J. Burkardt, M. Gunzburger, J. Peterson, R. Brannon, User manual and supporting information for library of codes for centroidal Voronoi point placement and associated zeroth, first, and second moment determination, SAND Report SAND2002-0099, Sandia National Laboratories, Albuquerque, 2002.
[4] L. Chen, J. Xu, Optimal Delaunay triangulation, J. Comp. Math. 22 (2004) 299–308.
[5] Q. Du, M. Emelianenko, L. Ju, Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations, SIAM J. Numer. Anal. 44 (2006) 102–119.
[6] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: Applications and algorithms, SIAM Rev. 41 (1999) 637–676.
[7] Q. Du, M. Gunzburger, Grid generation and optimization based on centroidal Voronoi tessellations, Appl. Math. Comput. 133 (2002) 591–607.
[8] Q. Du, M. Gunzburger, L. Ju, Voronoi-based finite volume methods, optimal Voronoi meshes and PDEs on the sphere, Comp. Meth. Appl. Mech. Engrg. 192 (2003) 3933–3957.
[9] Q. Du, M. Gunzburger, L.-L. Ju, A constrained centroidal Voronoi tessellations on surfaces, SIAM J. Sci. Comput. 24 (2003) 1488–1506.
[10] Q. Du, D. Wang, Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations, Int. J. Numer. Meth. Engrg. 56 (2003) 1355–1373.
[11] Q. Du, D. Wang, Anisotropic centroidal Voronoi tessellations and their applications, SIAM J. Sci. Comput. 26 (2005) 737–761.
[12] D. Field, Laplacian smoothing and Delaunay triangulation, Comm. Appl. Numer. Meth. 4 (1988) 709–712.
[13] D. Field, Qualitative measures for initial meshes, Inter. J. Numer. Meth. Engrg. 47 (2000) 887–906.
[14] L. Ju, Conforming centroidal Voronoi Delaunay triangulation for quality mesh generation, Inter. J. Numer. Anal. Model. 4 (2007) 531–547.
[15] L. Ju, Q. Du, M. Gunzburger, Meshfree, probabilistic determination of point sets and support regions for meshless computing, Comp. Meth. Appl. Mech. Engrg. 191 (2002) 1349–1366.
[16] L. Ju, M. Gunzburger, W. Zhao, Adaptive finite element methods for elliptic PDEs based on conforming centroidal Voronoi Delaunay triangulations, SIAM J. Sci. Comput. 28 (2006) 2023–2053.
[17] J. Shewchuk, Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator, in: Lecture Notes in Comput. Sci., vol. 1148, Springer, New York, 1996, pp. 203–222.
[18] J. Shewchuk, Delaunay refinement algorithms for triangular mesh generation, Comput. Geom.: Theory Appl. 22 (2002) 21–74.
[19] J. Shewchuk, What is a good linear finite element? Interpolation, conditioning, anisotropy and quality measures, Technical Report, Department of Computer Science, University of California, Berkeley, 2003.
[20] P.-O. Persson, G. Strang, A simple mesh generator in Matlab, SIAM Rev. 46 (2004) 329–345.