INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 4, Number 3-4, Pages 531–547

CONFORMING CENTROIDAL VORONOI DELAUNAY TRIANGULATION FOR QUALITY MESH GENERATION

LILI JU

This paper is dedicated to Max Gunzburger on the occasion of his 60th birthday.

Abstract. As the methodology of centroidal Voronoi tessellation (CVT) is receiving more and more attention in the mesh generation community, a clear characterization of the influence of geometric constraints on the CVT-based meshing is becoming increasingly important. In this paper, we first give a precise definition of the geometrically conforming centroidal Voronoi Delaunay triangulation (CfCVDT) and then propose an efficient algorithm for its construction that involves projection and lifting processes in two dimensional space. Finally, we show the high-quality of CfCVDT meshes and the effectiveness and robustness of our algorithm through extensive examples.

Key Words. centroidal Voronoi tessellation, Delaunay triangulation, conforming centroidal Voronoi Delaunay triangulation, mesh generation.

1. Introduction

Mesh generation often forms a crucial part of the numerical solution procedure in many applications. In the past few decades, automatic, unstructured mesh generations for complex 2D/3D domains have provided very successful tools for solving complex application problems, some of those well-studied techniques include AFT [25, 27], Octree [29], and Voronoi/Delaunay-based methods [1, 3–5, 30, 33]. It is well-known that the quality of Delaunay-based triangular/tetrahedral meshes is greatly affected by the placement of the generating points of the associated Voronoi regions. Many work has been devoted to finding robust and efficient algorithms to distribute the generating points by some optimal criteria, for example, the Laplacian smoothing [19], the centroidal Voronoi tessellation (CVT) [11], the optimal Delaunay triangulation (ODT) [7] and DistMesh [28]. We here are specially interested in the CVT approach.

Centroidal Voronoi tessellation proposed in [11] is a special Voronoi tessellation having the property that the generators of the Voronoi diagram are also the centers of mass, with respect to a given density function, of the corresponding Voronoi cells. CVTs are very useful in many applications, including but not limited to image and data analysis, vector quantization, resource optimization, design of experiments, optimal placement of sensors and actuators, cell biology, territorial behavior of animals, numerical partial differential equations, point sampling, meshless computing, mesh generation and optimization, reduced-order modeling, computer graphics, and mobile sensing networks. Recently, CVT and its duality – centroidal Voronoi Delaunay triangulation (CVDT) based mesh generation and mesh optimization have

Received by the editors February 7, 2006, and, in revised form, February 23, 2006.
2000 *Mathematics Subject Classification.* 65Y10, 68N10.

attracted a lot of attention and been used for many applications due to its optimal properties, see [1, 2, 12, 13, 15–17]. For example, in the medical imaging and simulation for surgical operations which are used to not just diagnose the patient's ailments but also test alternative treatments, a mesh generator with good triangle/tetrahedron quality and robust control over the mesh sizing and the number of elements is desired for numerical simulations [32].

In this paper, we propose an effective and efficient algorithm for high-quality triangular mesh generation based on the CVT-methodology, specifically, the geometrically conforming centroidal Voronoi Delaunay triangulation (CfCVDT). The plan of the rest of the paper is as follows. We first give a brief introduction to the concept of CVT in Section 2.1, and then generalize the definition for conforming mesh generation of complicated geometries in Section 2.2. We also propose an algorithm involving projection and lifting processes for approximate CfCVDT mesh construction in two dimensional space in Section 3, and some mesh examples generated by our algorithm for different geometries are given in Section 4 to show the high quality of CfCVDT meshes. Finally we make some concluding remarks in Section 5.

2. Conforming Centroidal Voronoi Delaunay Triangulation

2.1. Centroidal Voronoi tessellation. Given an open bounded domain $\Omega \in \mathbb{R}^d$ and a set of distinct points $\{\mathbf{x}_i\}_{i=1}^n \subset \Omega$. For each point $\mathbf{x}_i$, $i = 1, \ldots, n$, define the corresponding Voronoi region V_i, $i = 1, \ldots, n$, by

$$V_i = \big\{\mathbf{x} \in \Omega \quad | \quad \|\mathbf{x} - \mathbf{x}_i\| < \|\mathbf{x} - \mathbf{x}_j\| \quad \text{for } j = 1, \cdots, n \quad \text{and} \quad j \neq i\big\} \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean distance in $\mathbb{R}^d$. Clearly $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\cup_{i=1}^n \overline{V}_i = \Omega$ so that $\{V_i\}_{i=1}^n$ is a tessellation of Ω. We refer to $\{V_i\}_{i=1}^n$ as the *Voronoi tessellation* (VT) of Ω associated with the point set $\{\mathbf{x}_i\}_{i=1}^n$. A point $\mathbf{x}_i$ is called a *generator*; a subdomain $V_i \subset \Omega$ is referred to as the *Voronoi region* corresponding to the generator $\mathbf{x}_i$. It is well-known that the dual tessellation (in a graph-theoretical sense) to a Voronoi tessellation of Ω is the so-called *Delaunay triangulation* (DT). The Voronoi regions V_i's are convex polygons if Ω is convex and their vertices consist of circumcenters of the corresponding Delaunay triangles.

Given a density function $\rho(\mathbf{x}) \geq 0$ defined on Ω, for any region $V \subset \Omega$, define $\mathbf{x}^*$, the *mass center* or *centroid* of V by

$$\mathbf{x}^* = \frac{\displaystyle\int_V \mathbf{y}\rho(\mathbf{y})\, \mathrm{d}\mathbf{y}}{\displaystyle\int_V \rho(\mathbf{y})\, \mathrm{d}\mathbf{y}}. \tag{2}$$

Then a special family of Voronoi tessellations are defined in the following [11]:

Definition 1. *We refer to a Voronoi tessellation* $\{(\mathbf{x}_i, V_i)\}_{i=1}^n$ *of* Ω *as a* centroidal Voronoi tessellation *if and only if the points* $\{\mathbf{x}_i\}_{i=1}^n$ *which serve as the generators of the associated Voronoi regions* $\{V_i\}_{i=1}^n$ *are also the centroids of those regions, i.e., if and only if we have that*

$$\mathbf{x}_i = \mathbf{x}_i^* \quad \textit{for} \;\; i = 1, \ldots, n\,. \tag{3}$$

The corresponding Delaunay triangulation is then called a centroidal Voronoi Delaunay triangulation.

General Voronoi tessellations do not satisfy the CVT property. It is worth noting that CVT or CVDT may not be unique [11]. The CVT concept also can be generalized to very broad settings that range from abstract spaces and distance metrics to discrete point sets [11, 13].

CVTs possess an optimization property that can be used as a basis for an alternate definition. Given any set of points $\{\widetilde{\mathbf{x}}_i\}_{i=1}^n$ in Ω and any tessellation $\{\widetilde{V}_i\}_{i=1}^n$ of Ω, define the *energy* by

$$\mathcal{K}\big(\{(\widetilde{\mathbf{x}}_i, \widetilde{V}_i)\}_{i=1}^n\big) = \sum_{i=1}^n \int_{\widetilde{V}_i} \rho(\mathbf{y}) \|\mathbf{y} - \widetilde{\mathbf{x}}_i\|^2 \, \mathrm{d}\mathbf{y} \,. \tag{4}$$

Then it can be shown that $\mathcal{K}$ is minimized only if $\{(\widetilde{\mathbf{x}}_i, \widetilde{V}_i)\}_{i=1}^n$ forms a CVT of Ω [11]. Notice that $\{(\widetilde{\mathbf{x}}_i, \widetilde{V}_i)\}_{i=1}^n$ still may not be a minimizer of $\mathcal{K}$ although $\{(\widetilde{\mathbf{x}}_i, \widetilde{V}_i)\}_{i=1}^n$ is a CVT [11]. Although the *energy* $\mathcal{K}$ may not be directly identified with the energy of some physical system, it is often naturally associated with quantities such as *error distortion*, *variance* and *cost* in many applications. Based on the conjecture of equipartition of energy [11] for large number of generators, CVTs have important geometric features, including the following:

- For a constant density function, the generators $\{\mathbf{x}_i\}_{i=1}^n$ are uniformly distributed; the Voronoi regions $\{V_i\}_{i=1}^n$ are all almost of the same size and, in the two-dimensional case, most of them are (nearly) congruent convex hexagons [11].
- For a non-constant density function, the generators $\{\mathbf{x}_i\}_{i=1}^n$ are still locally uniformly distributed, and it is conjectured [11] that, asymptotically, for some constant C,

$$\frac{h_{V_i}}{h_{V_j}} \approx \left(\frac{\rho(\mathbf{x}_j)}{\rho(\mathbf{x}_i)}\right)^{\frac{1}{d+2}}, \tag{5}$$

  where h_{V_i} denotes the diameter of V_i and d is the dimension of Ω.

Construction of CVT is usually done by either probabilistic methods typified by MacQueen's random algorithm [26] (which is a simple iteration between sampling and averaging points) or deterministic methods typified by Lloyd iteration [24] (which is a simple iteration between constructing Voronoi diagrams and mass centroids). Due to the low convergence rate of MacQueen's method, much attention has been focused on the Lloyd method:

Algorithm 1. (**Lloyd Method for CVT**)
Given a domain Ω, a density function $\rho(\mathbf{x})$ defined on Ω, and a positive integer n.

```
0. Select an initial set of n points $\{\mathbf{x}_i\}_{i=1}^n$ on $\Omega$.
1. Construct the Voronoi regions $\{V_i\}_{i=1}^n$ of $\Omega$ associated with $\{\mathbf{x}_i\}_{i=1}^n$.
2. Determine the centroids of the Voronoi regions $\{V_i\}_{i=1}^n$; these centroids
   form the new set of points $\{\mathbf{x}_i\}_{i=1}^n$.
3. If the new points meet some convergence criterion, return $\{(\mathbf{x}_i, V_i)\}_{i=1}^n$ and
   terminate; otherwise, goto step 1.
```

It is worth noting that the energy $\mathcal{K}$ associated with the Voronoi tessellation $\{(\mathbf{x}_i, V_i)\}_{i=1}^n$ decreases monotonically during Lloyd iteration. A probabilistic version of Lloyd method and its parallel implementation were suggested in [22]. Another possible way to improve the performance of Lloyd method is to use the multigrid acceleration scheme, see [9].

2.2. Conforming CVT and its duality. When CVT/CVDT is applied to numerical solution of partial differential equations (PDEs), e.g., in a finite element method or a finite volume method, some modifications are needed to handle geometric constraints. An obvious one is that the CVT/CVDT-based mesh must conform with the boundary of the target domain Ω, i.e., some of CVT generators (CVDT vertices) have to be constrained to lie on the boundary so that the boundary conditions of the PDE problem can be enforced. There are basically two popular approaches proposed to solve this kind of geometric constraints:

- One approach is to construct the CVT without applying any constraint using standard algorithm such as the Lloyd method, and during the construction process (step 1 in Algorithm 1), for those Voronoi regions that extend to the exterior of the domain, their corresponding generators are projected onto the boundary.
- Another approach is to first distribute some generators on the boundary in accordance with some pre-defined mesh sizing function related to the density function ρ and form the boundary mesh, then determine the interior generators using Lloyd method for CVTs and construct the domain mesh which conforms with the boundary mesh.

The first approach was used in [12] for mesh generation in numerical solution of a 2D model PDE that produced high-quality triangular meshes. In [1, 15–17], a certain combination of both approaches was used for 3D tetrahedral meshing and the constrained Delaunay triangulation (CDT) [30,31] was used instead of standard Delaunay triangulation (DT) to generate the mesh. The main difference between CDT and standard DT is that some geometric constraints such as pre-determined node position and node connectivities are added and strictly enforced during the CDT process. For example, the boundary of the domain can be triangulated first, and the resulting boundary triangulation is then used as a constraint on the conforming triangulation of the whole domain using CDT. However, in most of the above approaches, the number of generators (vertices) on the boundary stays fixed or only increases for the conforming triangulation after first iteration of Lloyd algorithm no matter what density function ρ used. This is mostly due to the absence of lifting of generators from the boundary to the interior and the fact that the generators lying on the boundary can not return to the interior of the domain since their Voronoi regions obviously always extend to the exterior of the domain. Thus, the initial position of generators has to be well chosen in accordance with the density function ρ, otherwise the quality of CVDT mesh could be very bad, especially in the regions close to the domain boundary; for example, with the boundary triangulation not properly treated, some slivers may be produced in 3D meshing [1]. On the other hand, how to choose a good sizing function for distributing generators on the boundary is also not an easy problem if the density function ρ is complicated. Instead, vertex/face flippings were used together with CVDT meshing in [1, 15] to improve the mesh quality. Overall, a more clear characterization of the influence of geometric constraints on the CVT-based meshing remains to be explored.

We propose the following conforming CVT/CVDT concept. Let us assume that the domain $\Omega \in \mathbb{R}^d$ is compact and $\partial\Omega$ is piecewise smooth with singular (corner) points $P_S = \{\mathbf{z}_i\}_{i=1}^k$. Denote by $\mathbf{Proj}(\mathbf{x})$ the process that projects $\mathbf{x} \in \Omega$ to a point on the boundary $\partial\Omega$ closest to $\mathbf{x}$. Denote by P_I the set of generators in the interior of the domain and by P_B the set of generators on the boundary. Then we define

Definition 2. *A Voronoi Tessellation* $\{(\mathbf{x}_i, V_i)\}_{i=1}^n$ *of* Ω *is called a* conforming centroidal Voronoi tessellation (CfCVT) *if and only if the following properties are satisfied:*

- $P_S \subset \{\mathbf{x}_i\}_{i=1}^n$;
- $\mathbf{x}_i = \mathbf{x}_i^*$ for $\mathbf{x}_i \in P_I$;
- $\mathbf{x}_i = \mathbf{Proj}(\mathbf{x}_i^*)$ for $\mathbf{x}_i \in P_B - P_S$.

The corresponding dual triangulation is then called a conforming centroidal Voronoi Delaunay triangulation (CfCVDT)[1].

It is noted that the meaning of singular (corner) points is trivial in two-dimensional space but may need to be more rigorously defined in spaces higher than two dimensions. In many applications, the domain Ω is often non-convex and is possibly very complicated [15], so that a main difficulty associated with Lloyd method for constructing CfCVDT meshes is the construction of the Voronoi regions. For this reason, we next propose an algorithm for constructing approximate CfCVDT meshes in two dimensional space that does not require the construction of exact Voronoi tessellations. More importantly, in addition to the projection process used in the above approaches, a lifting process is also used in our algorithm that will allow the boundary vertices to return to the interior.

3. Approximate CfCVDT Mesh Construction

We will describe our algorithm for the two dimensional case in details; the generalization to higher dimensions follows similar lines. Currently, for mesh generation with conforming boundary requirements, CDT process has been widely used so far as mentioned before, but it is worth noting that the dual tessellations of CDT generally is not an exact Voronoi tessellation, especially near the boundary. Our algorithm for constructing approximate CfCVDTs is based on the CDT process and a procedure similar to Lloyd method for CVTs.

3.1. Polygonal domains. Assume that $\Omega \in \mathbb{R}^2$ is a domain with a polygonal boundary. In this case the projection process onto the domain boundary – $\mathbf{Proj}$ can be directly computed by simple calculations. An initial conforming triangulation $\mathcal{T}_0 = \{T_i\}_{i=1}^m$ of Ω can be easily generated by the "TRIANGLE" software package [30] that uses the CDT process with a boundary mesh as a constraint and interior Delaunay refinement techniques, or by some other means. Denote by $P = \{\mathbf{x}_i\}_{i=1}^n$ the set of vertices of $\mathcal{T}_0$. Let P_S, P_B and P_I be defined as before. The CDT process guarantees that $P_S \subset P$.

For each triangle $T_i = (\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \mathbf{x}_{i_3}) \in \mathcal{T}_0$, we define

$$\mathbf{x}_{T_i} = \begin{cases} \text{circumcenter of } T_i & \text{if } T_i \text{ is an acute triangle;} \\ \text{the middle point of the longest edge of } T_i & \text{otherwise.} \end{cases} \tag{6}$$

Clearly, $\mathbf{x}_{T_i} \in \overline{T}_i$. For each vertex $\mathbf{x}_i$, we denote by $\{T_{i_k}\}_{k=1}^{m_i} \subset \mathcal{T}_0$ the set of triangles for which $\mathbf{x}_i$ is a vertex, counting in the counterclockwise direction.

Interior vertices. First, consider the case $\mathbf{x}_i \in P_I$, i.e., $\mathbf{x}_i$ is an interior vertex. Define U_i by

$$U_i = \text{the polygon formed by } \{\mathbf{x}_{T_{i_k}}\}_{k=1}^{m_i}; \tag{7}$$

see Fig. 1. The polygon U_i can be regarded as an approximation to the Voronoi region V_i associated with $\mathbf{x}_i$. Let $\overline{\mathbf{x}}_i$ denote the center of mass of the U_i with

[1] A concept of constrained centroidal Voronoi Delaunay triangulation (abbreviated as CCVDT) was used in [13, 15, 16, 18], but its meaning is slightly different from the usage here.

respect to the density function ρ. Denote by $\{\alpha_{i_k}\}_{k=1}^{m_i}$ the associated angles around $\mathbf{x}_i$ corresponding to $\{T_{i_k}\}_{k=1}^{m_i}$. Define

$$\alpha = \begin{cases} \max\{\alpha_{i_k} \mid \overline{T}_{i_k} \cap \partial\Omega \neq \emptyset\} & \text{if } \overline{T}_{i_k} \cap \partial\Omega \neq \emptyset \text{ for some } i_k; \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

and $\mathbf{e}_i$ denote the corresponding boundary edge opposite to the angle α_{i_k} such that $\alpha_{i_k} = \alpha$; see Fig. 1 for illustrations of some cases.

Now, select a parameter θ_{max} ($\pi > \theta_{max} > \pi/2$). Then, define

$$\mathbf{y}_i = \begin{cases} \overline{\mathbf{x}}_i & \text{if } \alpha < \theta_{max}; \\ \mathbf{Proj}_{\mathbf{e}_i}\mathbf{x}_i & \text{otherwise} \end{cases} \tag{9}$$

where $\mathbf{Proj}_{\mathbf{e}_i}\mathbf{x}_i$ denotes the projection of $\mathbf{x}_i$ onto the boundary edge $\mathbf{e}_i$. It is clear that $\mathbf{y}_i$ is still an interior vertex if $\alpha < \theta_{max}$; otherwise, it is a boundary vertex although $\mathbf{x}_i$ is an interior node. In our computational experiments, we set $\theta_{max} = 5\pi/9$, a good value to choose, which is validated through many experiments.

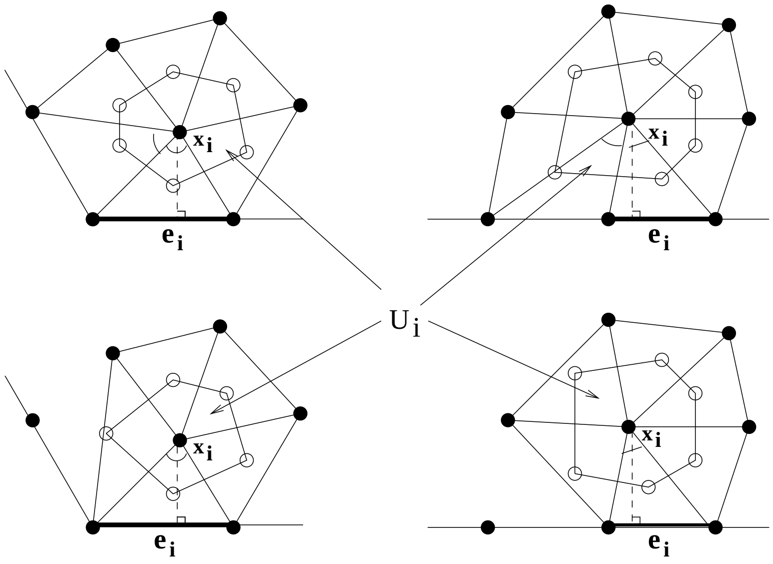

Figure 1: The approximate Voronoi region U_i for the interior vertex $\mathbf{x}_i$.

Boundary vertices. Next, consider the case $\mathbf{x}_i \in P_B$, i.e., $\mathbf{x}_i$ is a boundary vertex. Let $\mathbf{e}_1$ and $\mathbf{e}_2$ denote the two boundary edges having $\mathbf{x}_i$ as the common end point and denote by $\mathbf{z}_1$ and $\mathbf{z}_2$ the midpoints of $\mathbf{e}_1$ and $\mathbf{e}_2$, respectively; see Fig. 2. The approximate Voronoi region U_i of $\mathbf{x}_i$ is defined by

$$U_i = \text{the polygon formed by } \mathbf{z}_1,\ \{\mathbf{x}_{T_{i_k}}\}_{k=1}^{m_i}, \text{ and } \mathbf{z}_2; \tag{10}$$

see Fig. 2. Let $\overline{\mathbf{x}}_i$ denote the center of mass of the U_i associated with the density function ρ. If $\mathbf{x}_i \in P_B - P_S$, denote by β_1 and β_2 the angles facing the boundary edges $\mathbf{e}_1$ and $\mathbf{e}_2$, respectively, in $\{T_{i_k}\}_{k=1}^{m_i}$; see Fig. 2 (right). Let

$$\beta = \max(\beta_1, \beta_2) \tag{11}$$

and select a parameter θ_{min} ($\pi/3 > \theta_{min} > 0$). Then, define

$$\mathbf{y}_i = \begin{cases} \mathbf{x}_i & \text{if } \mathbf{x}_i \in P_S; \\ \mathbf{Proj}_{\overline{\mathbf{z}_1\mathbf{z}_2}}\overline{\mathbf{x}}_i & \text{if } \mathbf{x}_i \in P_B - P_S \text{ and } \beta > \theta_{min}; \\ \overline{\mathbf{x}}_i & \text{if } \mathbf{x}_i \in P_B - P_S \text{ and } \beta \leq \theta_{min} \end{cases} \tag{12}$$

where $\mathbf{Proj}_{\overline{\mathbf{z}_1\mathbf{z}_2}}\mathbf{x}_i$ denotes the projection of $\mathbf{x}_i$ onto the segment $\overline{\mathbf{z}_1\mathbf{z}_2}$. It is clear that $\mathbf{y}_i$ is also a boundary vertex if $\mathbf{x}_i$ is a corner vertex or $\mathbf{x}_i$ is non-corner vertex but $\beta > \theta_{min}$; otherwise, $\mathbf{y}_i$ becomes an interior vertex (the lifting process) although $\mathbf{x}_i$ is on the boundary. In our numerical experiments, we set $\theta_{min} = \pi/6$.

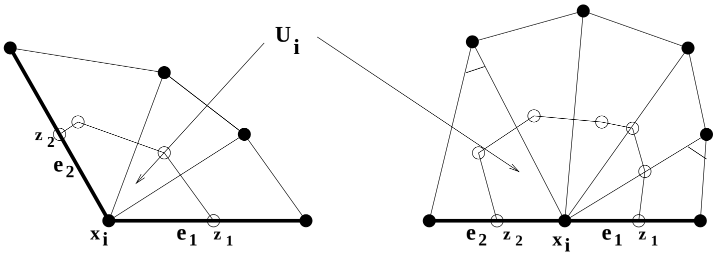

Figure 2: The approximate Voronoi region U_i for the boundary vertex $\mathbf{x}_i$. left: corner vertex; right: non-corner vertex.

3.2. Additional interface constraints. There could be other kind of geometric constraints imposed for specific numerical problems. For example, for the interface singularity problems [6, 21], it is often required that no triangles/tetrahedra can straddle the interface, i.e., the triangulations have to conform to the interfaces, so that the discontinuities of the coefficients only occur across mesh edges/faces. This will surely also pose some constraints on the position of generators and triangulation process during CfCVDT construction. Let us assume that the initial $\mathcal{T}_0$ satisfies the above geometric constraint. In the following, we will describe some additional techniques for this case. Let Γ be the interface inside the domain imposed by the physical system. Denote by $P_F \in P_I$ the set of interior vertices located on the interface Γ and by $P_{FS} \in P_F$ the set of interior interface vertices that are corner points of the interface.

Interface vertices. Let us first consider the case $\mathbf{x}_i \in P_F$, i.e., $\mathbf{x}_i$ is an interior vertex located on the interface. Let $\mathbf{e}_1$ and $\mathbf{e}_2$ denote the two interface edges having $\mathbf{x}_i$ as the common end point respectively. The approximate Voronoi region U_i of $\mathbf{x}_i$ is defined as same as that for the interior points; see Fig. 3. Let $\overline{\mathbf{x}}_i$ denote the centroid of the U_i associated with the density function ρ. If $\mathbf{x}_i \in P_F - P_{FS}$, denote by $\{\beta_i\}_{i=1}^4$ the angles facing the interface edges $\mathbf{e}_1$ and $\mathbf{e}_2$, respectively, in $\{T_{i_k}\}_{k=1}^{m_i}$; see Fig. 3 (right). Let

$$\beta_{max} = \max\{\beta_1, \beta_2, \beta_3, \beta_4\}, \quad \beta_{min} = \min\{\beta_1, \beta_2, \beta_3, \beta_4\} \tag{13}$$

and $\mathbf{z}$ be the corresponding interior non-interface vertex forming the angle $\beta_i = \beta_{min}$; see Fig. 3 (right). Then, define

$$\mathbf{y}_i = \begin{cases} \mathbf{x}_i & \text{if } \mathbf{x}_i \in P_{FS}; \\ \mathbf{Proj}_{\mathbf{e}_1\mathbf{e}_2}\overline{\mathbf{x}}_i & \text{if } \mathbf{x}_i \in P_F - P_{FS} \text{ and } \beta_{max} > \theta_{min}; \\ (\overline{\mathbf{x}}_i + \mathbf{z})/2 & \text{if } \mathbf{x}_i \in P_F - P_{FS} \text{ and } \beta_{max} \leq \theta_{min} \end{cases} \tag{14}$$

where $\mathbf{Proj}_{\mathbf{e}_1\mathbf{e}_2}\mathbf{x}_i$ denotes the projection of $\mathbf{x}_i$ onto the segment $\mathbf{e}_1\mathbf{e}_2$ ($\mathbf{e}_1\mathbf{e}_2$ is a straight line when $\mathbf{x}_i$ is not a corner vertex of the interface). It is clear that $\mathbf{y}_i$ is still a corner vertex of the interface Γ if $\mathbf{x}_i$ is a corner vertex of the interface, or $\mathbf{x}_i$ is a non-corner vertex on the interface but $\beta > \theta_{min}$; otherwise, $\mathbf{y}_i$ becomes an interior vertex not located on the interface Γ although $\mathbf{x}_i$ is located on the interface.

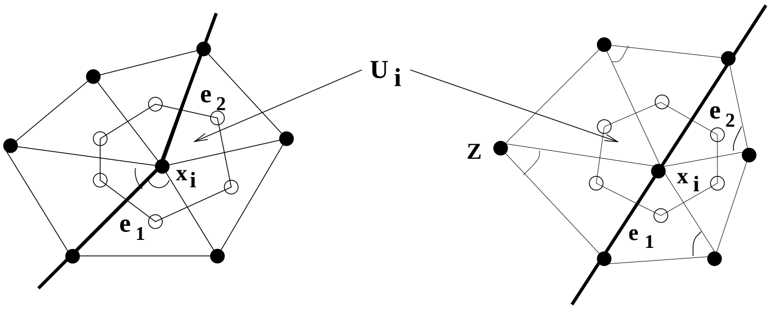

Figure 3: The approximate Voronoi region U_i for the interior vertex $\mathbf{x}_i$ located on the interface. left: a corner vertex of the interface ; right: a non-corner vertex of the interface.

Interior vertices near the interface. Let $\mathbf{x}_i$ and $\mathbf{x}_j$ be the two interior vertices facing the common interface edge $\mathbf{e}_{ij}$. Their approximate Voronoi regions U_i and U_j are defined as same as that for the interior points, see Fig. 4. Let $\overline{\mathbf{x}}_i$ and $\overline{\mathbf{x}}_j$ denote the centers of mass of the U_i and U_j with respect to the density function ρ respectively. Denote by $\{\alpha_{i_k}\}_{k=1}^{m_i}$ the associated angles around $\mathbf{x}_i$ corresponding to $\{T_{i_k}\}_{k=1}^{m_i}$. Define

$$\alpha_i = \begin{cases} \max\{\alpha_{i_k} \mid \overline{T}_{i_k} \cap \Gamma \neq \emptyset\} & \text{if } \overline{T}_{i_k} \cap \Gamma \neq \emptyset \text{ for some } i_k; \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

and let $\mathbf{e}_i$ denote the corresponding interface edge opposite to the angle α_{i_k} such that $\alpha_{i_k} = \alpha$. Similarly, we can define α_j and $\mathbf{e}_j$ for the vertex $\mathbf{x}_j$.

Then, we define

$$\mathbf{y}_i = \begin{cases} \overline{\mathbf{x}}_i & \text{if } \alpha_i < \theta_{max}; \\ \mathbf{Proj}_{\mathbf{e}_i}\mathbf{x}_i & \text{if } \alpha_i > \theta_{max} \text{ and } \mathbf{e}_i \neq \mathbf{e}_j; \\ \mathbf{Proj}_{\mathbf{e}_i}\mathbf{x}_i & \text{if } \alpha_i > \alpha_j > \theta_{max} \text{ and } \mathbf{e}_{ij} = \mathbf{e}_i = \mathbf{e}_j; \\ \overline{\mathbf{x}}_i & \text{otherwise} \end{cases} \tag{16}$$

where $\mathbf{Proj}_{\mathbf{e}_i}\mathbf{x}_i$ denotes the projection of $\mathbf{x}_i$ onto the interface edge $\mathbf{e}_i$. From this construction, it is easy to see that $\mathbf{x}_i$ and $\mathbf{x}_j$ are not allowed to be projected onto $\mathbf{e}_{ij}$ at the same step and $\mathbf{x}_i$ (or $\mathbf{x}_j$) can not go across the interface into the other side of the interface in one step.

3.3. Curved domains. For domains with curved boundary, we take the approach of level set function used by [28]. Let us assume that the boundary of the domain Ω is the zero level set of a given function $f(\mathbf{x})$. For each point $\mathbf{x}_0 = (x_0, y_0) \in \Omega$, the $\mathbf{Proj}(\mathbf{x}_0)$ process requires us to find the closest point $\mathbf{x} = (x, y)$ on the zero level set, i.e., $f(\mathbf{x}) = 0$ and $\mathbf{x} - \mathbf{x}_0$ is parallel to the gradient (f_x, f_y) at $\mathbf{x}$:

$$Ł(\mathbf{x}) = \left[\begin{array}{c} f(x,y) \\ (x-x_0)f_y - (y-y_0)f_x \end{array} \right] = 0. \tag{17}$$

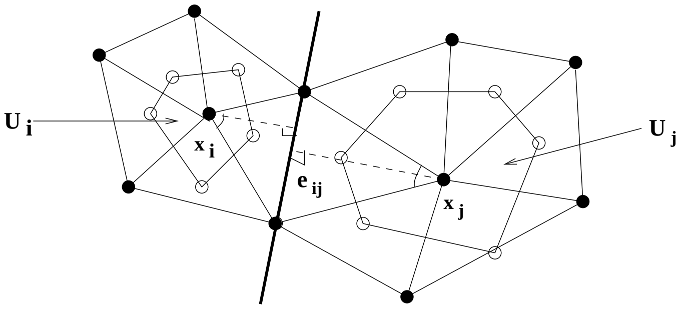

Figure 4: The approximate Voronoi regions for the interior vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ near the interface.

To solve the above problem, we use the damped Newton's method with $\mathbf{x}_0$ as initial guess proposed in [28]. The Jacobian of Ł is

$$\mathbf{J}(\mathbf{x}) = \frac{\partial Ł}{\partial \mathbf{x}} = \begin{bmatrix} f_x & f_y + (x-x_0)f_{xy} - (y-y_0)f_{xx} \\ f_y & -f_x - (y-y_0)f_{xy} + (x-x_0)f_{yy} \end{bmatrix}^{\mathbf{T}}, \tag{18}$$

and we iterate

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \mathbf{J}^{-1}(\mathbf{x}_k)Ł(\mathbf{x}_k), \qquad k = 0, 1, 2, \cdots, \tag{19}$$

until the residual $Ł(\mathbf{x}_k)$ is small enough. We then take the approximate solution $\mathbf{x}_k$ generated by the above procedure as $\mathbf{Proj}(\mathbf{x}_0)$. The damping factor is often set to be 1.

One of the convenient ways to define the level set function $f(x,y)$ is to use the signed distance functions [28]. The distance function $d(x,y)$ measures the (signed) distance for (x,y) to the domain boundary $\partial\Omega$ and satisfies

$$d(x,y) = \begin{cases} < 0 & \text{if } (x,y) \in \Omega - \partial\Omega; \\ = 0 & \text{if } (x,y) \in \partial\Omega; \\ > 0 & \text{otherwise,} \end{cases} \tag{20}$$

so the domain boundary is the zero level set of $d(x,y)$. Some examples of d for simple geometries, and various help functions for the generation of distance functions for more complicated domains have been given in [28] in great details.

3.4. Construction algorithm. Based on the above discussions, now we describe an algorithm for constructing an approximate CfCVDT of the domain Ω.

Algorithm 2. (**Lloyd method for approximate CfCVDT**)
Given a domain $\Omega \in \mathbb{R}^2$, a density function $\rho(\mathbf{x})$ defined on Ω, and an initial triangulation $\mathcal{T}_0$ of Ω with vertices $\{\mathbf{x}_i\}_{i=1}^n$ generated using CDT.

1. *Determine $\{\mathbf{y}_i\}_{i=1}^n$ from $\{\mathbf{x}_i\}_{i=1}^n$ according to* (9), (12), (14) *and* (16).
2. *Set $\{\mathbf{x}_i\}_{i=1}^n = \{\mathbf{y}_i\}_{i=1}^n$. If the domain consists of curved boundary, set $\mathbf{x}_i = \mathbf{Proj}(\mathbf{x}_i)$ by* (19) *for all $\mathbf{x}_i$ regarded as boundary nodes.*
3. *Reconstruct the boundary segments $\mathcal{E}_B$ from the new $\{\mathbf{x}_i\}_{i=1}^n$.*
4. *Re-triangulate the domain Ω using CDT with $\{\mathbf{x}_i\}_{i=1}^n$ as the vertices and $\mathcal{E}_B$ as the boundary edges; the resulting triangulation is the new $\mathcal{T}$.*
5. *If the triangulation $\mathcal{T}$ meets some convergence criterion, return $\mathcal{T}$ and terminate; otherwise, go to step* 1.

Of course, to prevent some vertices from frequently jumping back and forth between the boundary and the interior of the domain, more sophisticated controls are needed. For the sake of simplicity, we omit some details in Algorithm 2. We expect the convergence of this Algorithm 2 to be similar to the Algorithm 1 that has been extensively studied [10].

4. Computational Examples

In this section, through a number of mesh examples from simple geometries to relatively complicated geometries[2], we illustrate the high-quality of the CfCVDT meshes and the robustness and effectiveness of our construction Algorithm 2. Given a triangulation $\mathcal{T}$ of Ω, let h_T denote the diameter of T for any triangle $T \in \mathcal{T}$ and define

$$h_{max} = \max_{T\in\mathcal{T}} h_T, \quad h_{min} = \min_{T\in\mathcal{T}} h_T,$$

then h_{max}/h_{min} will be used to measure the distribution of the nodes: the larger h_{max}/h_{min} is, the more the nodes are distributed non-uniformly. We apply the commonly used q-measure [20] to evaluate the quality of the triangular mesh, where, for any triangle T, q is defined to be twice the ratio of the radius R_T of the largest inscribed circle and the radius r_T of the smallest circumscribed circle, i.e.,

$$q(T) = 2\frac{R_T}{r_T} = \frac{(b+c-a)(c+a-b)(a+b-c)}{abc}, \tag{21}$$

where a, b, and c are side lengths of T. Clearly $0 < q \leq 1$ and $q = 1$ corresponds to the equilateral triangle. For a given triangulation $\mathcal{T}$, we define

$$q_{min} = \min_{T\in\mathcal{T}} q(T) \qquad \text{and} \qquad q_{avg} = \frac{1}{\text{card}(\mathcal{T})}\sum_{T\in\mathcal{T}} q(T).$$

q_{min} measures the quality of the worst triangle and q_{avg} measures the average quality of the mesh $\mathcal{T}$. The measurements of mesh quality and node distribution of all examples discussed below are reported in Table 1.

(1) Unit Circle. We first set $\Omega = \{(x,y) \mid x^2+y^2 \leq 1\}$ be the unit circle and then use the signed distance function $d(x,y) = dcircle(x,y,0,0,1)$ where

$$dcricle(x,y,x_c,y_c,r) = \sqrt{(x-x_c)^2+(y-y_c)^2} - r.$$

Figure 5 shows the resulting CfCVDT meshes for $n = 217$ and $n = 817$ with different density functions

$$\rho(x,y) = 1, \tag{22}$$

$$\rho(x,y) = e^{-10.0\sqrt{x^2+y^2}}, \tag{23}$$

$$\rho(x,y) = e^{-10.0|\sqrt{x^2+y^2}-1|} \tag{24}$$

respectively.

(2) Square with Hole. We set $\Omega = [0,1]^2 - \{(x,y) \mid x^2+y^2 < 0.5^2\}$ with signed distance function

$$d(x,y) = \max(drectangle(x,y,-1,1,-1,1),\ -dcircle(x,y,0,0,0.5))$$

where

$$\begin{aligned} &drectangle(x,y,x_1,x_2,y_1,y_2) = \\ &\qquad -\min(\min(\min(-y_1+y, y_2-y), -x_1+x), x_2-x). \end{aligned}$$

[2] Some sample domains can be found in [28].

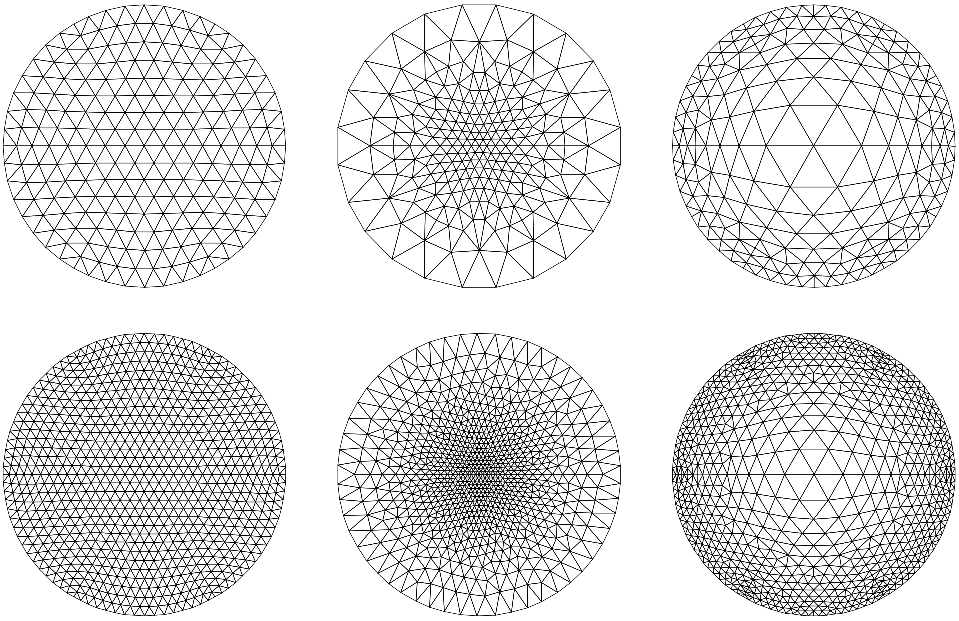

Figure 5: CfCVDT meshes of a unit circle. Top: $n = 217$; bottom: $n = 817$; left: with density function (22); middle: with density function (23); right: with density function (24).

Figure 6 shows the resulting CfCVDT meshes for $n = 305$ and $n = 1126$ with different density functions

$$\rho(x, y) = 1, \tag{25}$$

$$\rho(x, y) = e^{-10.0|\sqrt{x^2+y^2}-0.5|}, \tag{26}$$

$$\rho(x, y) = e^{-10.0\min(|\sqrt{x^2+y^2}-0.5|,\ dp)} \tag{27}$$

respectively with

$$dp = \min\left\{\sqrt{(x-x_i)^2+(y-y_i)^2}\right\}_{i=1}^{4}$$

where $\{(x_i, y_i)\}_{i=1}^4$ are four corners of the square.

(3) Hexagon with Hexagonal Hole. We choose $\Omega = \Omega_1 - \Omega_2$ be a hexagon with a hexagonal hole where Ω_1 is the hexagon formed by corner points $\{(\cos\theta_i, \sin\theta_i)\}_{i=1}^6$ with $\theta_i = (i - 1/2)\pi/3$ for $i = 1, \cdots, 6$ and Ω_2 is the hexagon formed by corner points $\{(0.5\cos\beta_i, 0.5\sin\beta_i)\}_{i=1}^6$ with $\beta_i = (i-1)\pi/3$ for $i = 1, \cdots, 6$. Figure 7 shows the resulting CfCVDT meshes for $n = 809$ with different density functions

$$\rho(x, y) = 1, \tag{28}$$

$$\rho(x, y) = e^{-20.0dp} \tag{29}$$

respectively with

$$dp = \min\left\{\sqrt{(x-x_i)^2+(y-y_i)^2}\right\}_{i=1}^{6}$$

where $\{(x_i, y_i)\}_{i=1}^6$ are six corner points of the interior hexagon.

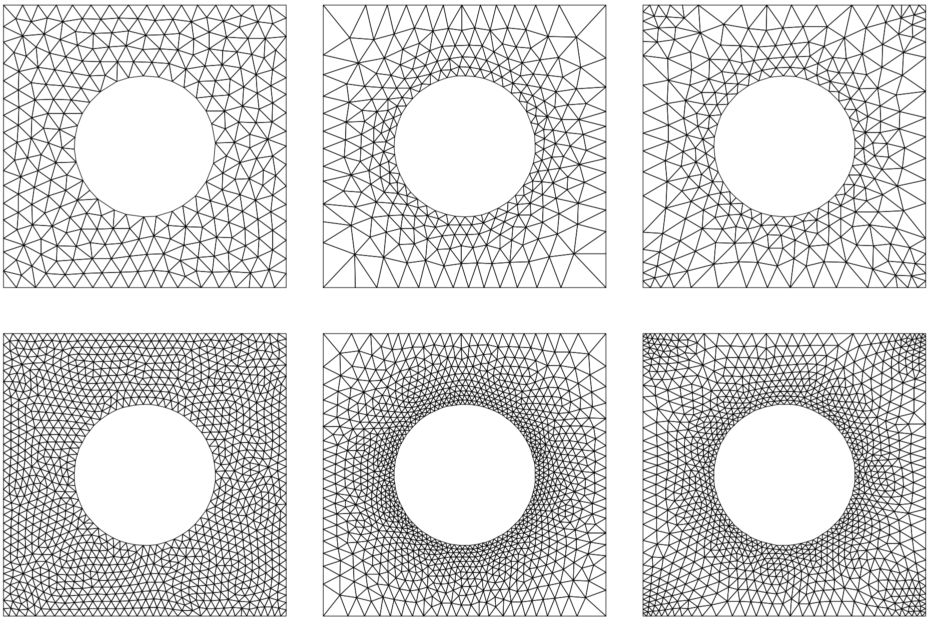

Figure 6: CfCVDT meshes of a square with hole. Top: $n = 305$; bottom: $n = 1126$; left: with density function (25); middle: with density function (26); right: with density function (27).

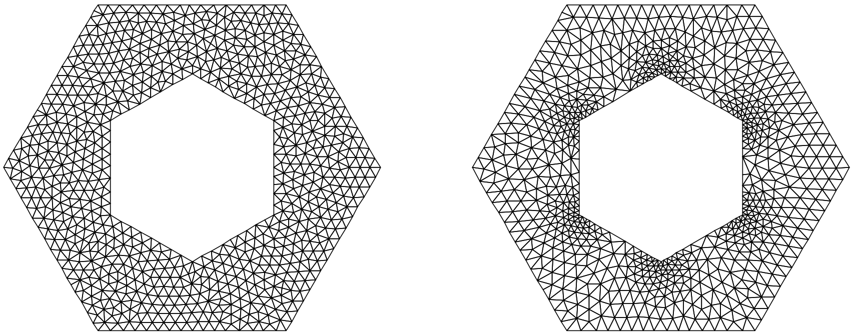

Figure 7: CfCVDT meshes of a hexagon with a hexgonal hole for $n = 809$. Left: with density function (28); right: with density function (29).

(4) Square with Square Holes. We choose the domain Ω be a square with four square holes defined by $\Omega = [-1,1]^2 - \Omega_1 \cup \Omega_2 \cup \Omega_3 \cup \Omega_4$ where

$$\begin{aligned}\Omega_1 &= \big([0.25, 0.75]^2, \quad \Omega_2 = [-0.75, -0.25] \times [0.25, 0.75],\\ \Omega_3 &= [-0.75, -0.25]^2), \quad \Omega_4 = [0.25, 0.75] \times [-0.75, -0.25]\big)\end{aligned}$$

Figure 8 (left) shows the resulting CfCVDT mesh for $n = 1661$ with the density function

$$\rho(x, y) = e^{-20.0|dp - \sqrt{2}/4|}. \tag{30}$$

where

$$dp = \min \left\{ \sqrt{(x - x_i)^2 + (y - y_i)^2} \right\}_{i=1}^{4}$$

with $\{(x_i, y_i)\}_{i=1}^4 = \{(0.5, 0.5), (-0.5, 0.5), (-0.5, -0.5), (0.5, -0.5)\}$.

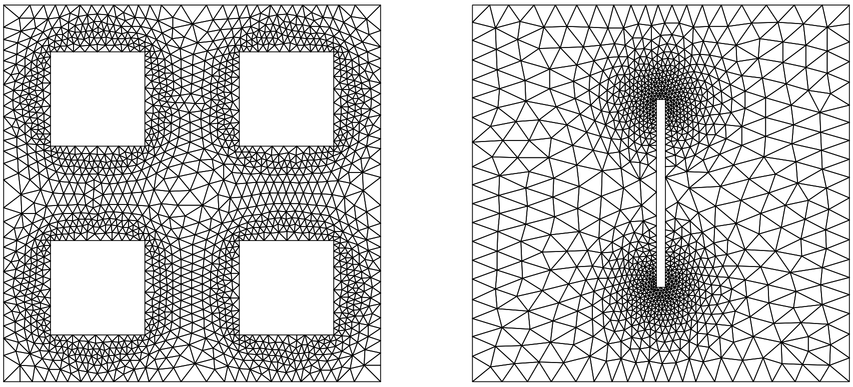

Figure 8: CfCVDT meshes of a square with four square holes with $n = 1661$ and the density function (30) (left) and a slit with $n = 1112$ and the density function (31) (right).

(5) Slit. We choose the domain Ω be a slit such that

$$\Omega = [-1, 1]^2 - (-0.025, 0.025) \times (-0.5, 0.5).$$

Figure 8 (right) shows the resulting CfCVDT mesh for $n = 1112$ with the density function

$$\rho(x, y) = e^{-20.0 \min\left(\sqrt{x^2+(y-0.5)^2},\ \sqrt{x^2+(y+0.5)^2}\right)}. \tag{31}$$

(6)&(7) Implicit Expressions. We first choose Ω (Example 6) to be the region between the level sets 0.5 and 1.0 of the super-ellipse $f(x, y) = (x^4 + y^4))^{1/4}$ (a super-ellipse with a super-ellipse hole). Figure 9 (left) show the resulting CfCVDT mesh for $n = 851$ with the density function

$$\rho(x, y) = e^{-10.0\left|\sqrt{x^2+y^2}-0.5\right|}. \tag{32}$$

Another example of the domain Ω (Example 7) is chosen to be the intersection of the following two regions (a PI-shape domain):

$$y \leq \cos(x) \quad \text{and} \quad y \geq 5\left(\frac{2x}{5\pi}\right)^4 - 5.$$

Figure 9 (right) shows the resulting CfCVDT mesh for $n = 473$ with the density function

$$\rho(x, y) = 1. \tag{33}$$

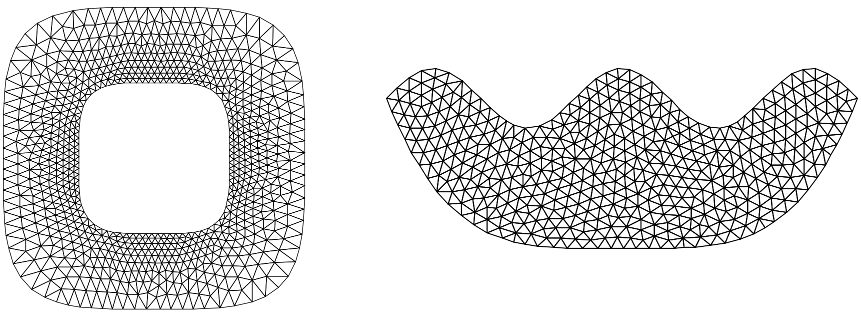

Figure 9: CfCVDT meshes of domains with implicit expressions. Left: a super-ellipse with a super-ellipse hole with $n = 851$ and the density function (32); right: a PI-shape domain with $n = 473$ and the density function (33).

(8) More complex Geometry. The domain of this example is a more complicated construction involving set operations on circles, $\Omega = \Omega_1 - \Omega_2 - \Omega_3$ where

$$\begin{aligned}
\Omega_1 &= \{(r\cos\theta, r\sin\theta) \mid 0 \le r \le 1,\ -\pi/12 \le \theta \le \pi/12\},\\
\Omega_2 &= \{(x,y) \mid (x-0.6)^2 + y^2 < 0.1^2\},\\
\Omega_3 &= \{(0.9 + rcos\theta, r\sin\theta) \mid 0 < r < 1,\ -\pi/4 < \theta < \pi/4\}.
\end{aligned}$$

Figure 10 (right) shows the resulting CfCVDT mesh for $n = 735$ with the density function

$$\rho(x,y) = e^{-40.0\min\left(|\sqrt{(x-0.6)^2+y^2}-0.05|,\ \sqrt{(x-0.9)^2+y^2},\ \sqrt{x^2+y^2},\ 0.15\right)}. \tag{34}$$

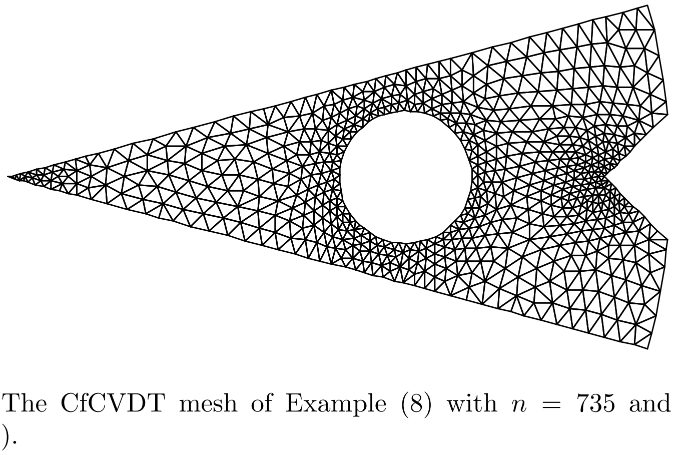

Figure 10: The CfCVDT mesh of Example (8) with $n = 735$ and the density function (34).

We skipped the description of signed distance function for Example 8 since it is a little bit complicated.

(9)&(10) Domains with Interface Constraints. These examples are used to show the effectiveness of our algorithm for constructing CfCVDT meshes with interface constraints. We first choose the square domain $\Omega = [-1,1]^2$ (Example 9) with the interface

$$\Gamma = \{(x,0) \mid -1 < x < 1\} \cup \{(0,y) \mid 1 < y < 1\}.$$

Figure 11 (left) shows the resulting CfCVDT mesh for $n = 1089$ with the density function

$$\rho(x,y) = e^{-15.0 \min\left(\sqrt{(x-0.5)^2+(y-0.5)^2},\ \sqrt{(x+0.5)^2+(y+0.5)^2}\right)}. \tag{35}$$

Another example of domain Ω (Example 10) is chosen to be a square with a square hole such that $\Omega = [-1,1]^2 - (0.5,0.5)^2$ and the interface is set to

$$\begin{aligned}\Gamma = \{(x,0) \mid -1 < x < -0.5 \ \text{ or } \ 0.5 < x < 1\} \\ \cup\{(0,y) \mid -1 < y < -0.5 \ \text{ or } \ 0.5 < y < 1\}.\end{aligned}$$

Figure 11 (right) shows the resulting CfCVDT mesh for $n = 1072$ with the density function

$$\rho(x,y) = e^{-(10.0dp_1+15.0dp_2)}. \tag{36}$$

where

$$\begin{aligned} dp_1 &= \sqrt{(x+0.5)^2+(y+0.5)^2}, \\ dp_2 &= \min\Big(\sqrt{(x-0.5)^+(y-0.5)^2}, \sqrt{(x+0.5)^+(y-0.5)^2}, \\ &\qquad \sqrt{(x+0.5)^+(y+0.5)^2}, \sqrt{(x-0.5)^+(y+0.5)^2}\Big). \end{aligned}$$

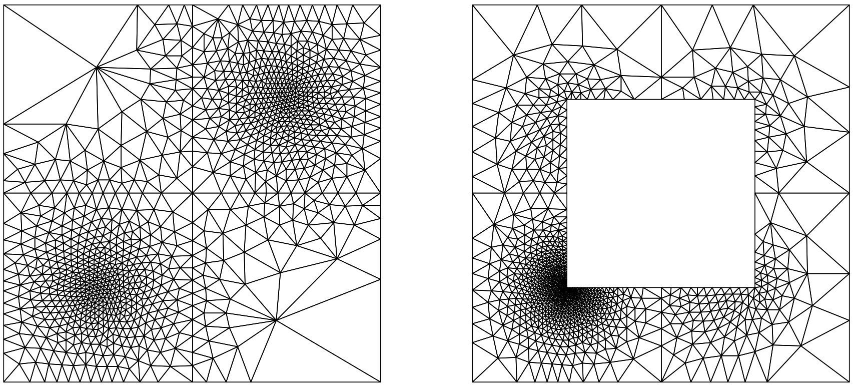

Figure 11: CfCVDT meshes with interface constraints. Left: Example (9) with $n = 1089$ and the density function (35); right: Example (10) with $n = 1072$ and the density function (36).

From the results in Table 1, we can conclude that the quality of CfCVDT meshes are really excellent even when the nodes are highy non-uniformly distributed (i.e., local mesh size varies a lot across the whole domain).

5. Conclusions and Future work

The two dimensional CfCVDT mesh generator discussed in this paper has been implemented in C language based on the very popular software package for triangular mesh generation, "TRIANGLE". The three dimensional CfCVDT mesh (tetrahedral mesh) generator is still under development. In [23], the CfCVDT mesh has been used to design optimal adaptive finite element methods for elliptic PDEs based on the combination of local a posterior estimators and the special CVT property (5), and has obtained promising results. We believe that the CfCVDT

Examples	Density Function	n	q_{min}	q_{avg}	h_{max}/h_{min}
Example 1	(22)	217	0.845	0.980	1.11
		817	0.827	0.983	1.21
	(23)	217	0.634	0.908	5.10
		817	0.679	0.947	6.01
	(24)	217	0.661	0.861	3.56
		817	0.641	0.889	5.05
Example 2	(25)	305	0.671	0.951	1.41
		1126	0.663	0.963	1.54
	(26)	305	0.676	0.925	5.59
		1126	0.657	0.948	6.98
	(27)	305	0.705	0.926	4.51
		1126	0.700	0.949	4.72
Example 3	(28)	571	0.722	0.954	1.41
	(29)	809	0.585	0.931	4.05
Example 4	(30)	1661	0.678	0.936	4.77
Example 5	(31)	1112	0.659	0.931	11.17
Example 6	(32)	851	0.728	0.940	3.51
Example 7	(33)	473	0.697	0.947	1.43
Example 8	(34)	735	0.559	0.934	5.32
Example 9	(35)	1089	0.458	0.933	30.78
Example 10	(36)	1072	0.471	0.934	59.94

Table 1: Measurements of mesh quality and node distribution of all sample CfCVDT meshes.

meshes will attract more and more attention and be very useful for many scientific and engineering problems in the future.

Acknowledgments

This research is partially supported by the University of South Carolina Research and Productive Scholarship under grant number RPS-13060-06-12306.

References

[1] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun, Variational tetrahedral meshing, *Proceedings of ACM SIGGRAPH'05*, pp. 617-625, 2005.

[2] P. Alliez, T.C. De Verdiere, O. Devillers O, and M. Isengurg, Centroidal Voronol diagrams for isotropic surface remeshing, *GRAPHICAL MODELS* **67**, pp. 204-231, 2005.

[3] T. Baker, Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation, *Engrg. Comp.* **5**, pp. 161-175, 1989.

[4] H. Borouchaki and P. George, Optimal Delaunay point insertion, *Int. J. Num. Meth. Engrg.* **39**, pp. 3407-3437, 1996.

[5] H. Borouchaki and S. Lo, Fast Delaunay triangulation in three dimensions, *Comp. Meth. Appl. Mech. Engrg.* **128**, pp. 153-167, 1995.

[6] S. Brenner and L.-Y. Sung, Multigrid methods for the computation of singular solutions and stress intensity factors III: Interface singularities, *Comp. Meth. Appl. Mech. Engrg.* **192**, pp. 4687-4702, 2003.

[7] L. Chen and J. Xu, Optimal Delaunay triangulation, *J. Comp. Math.* **22**, pp. 299-308, 2004.

[8] P.G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, 1978.

[9] Q. Du and M. Emelianenko, Acceleration schemes for the computation of centroidal Voronoi tessellations, *Numerical Linear Algebra and Application*, in press, 2006.

[10] Q. Du, M. Emelianenko, and L. Ju, Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations, *SIAM J. Numer. Anal.* **44**, pp. 102-119, 2006.

[11] Q. Du, V. Faber, and M. Gunzburger, Centroidal Voronoi tessellations: Applications and algorithms, *SIAM Review* **41**, pp. 637-676, 1999.

[12] Q. Du and M. Gunzburger, Grid generation and optimization based on centroidal Voronoi tessellations, *Applied Math. Comptut.* **133**, pp. 591-607, 2002.

[13] Q. Du, M. Gunzburger, and L. Ju, Constrained centroidal Voronoi tessellations on general surfaces, *SIAM J. Sci. Comput.* **24**, pp. 1488-1506, 2003.

[14] Q. Du, M. Gunzburger, and L. Ju, Voronoi-based finite volume methods, optimal Voronoi meshes and PDEs on the sphere, *Comp. Meth. Appl. Mech. Engrg.* **192**, pp. 3933-3957, 2003.

[15] Q. Du and D. Wang, Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations, *Int. J. Numer. Methods Engrg.* **56**, pp. 1355-1373, 2003.

[16] Q. Du and D. Wang, Constrained boundary recovery for 3D Delaunay triangulations, *Int. J. Num. Method. In. Eng.* **61**, pp. 1471-1500, 2004.

[17] Q. Du and D. Wang, Anisotropic centroidal Voronoi tessellations and their applications, *SIAM. J. Sci. Comput.* **26**, pp. 737-761, 2005.

[18] Q. Du and D. Wang, New progress in robust and quality Delaunay mesh generation, *J. Computational and Applied Mathematics*, in press, 2006.

[19] D. Field, Laplacian smoothing and Delaunay triangulation, *Comm. Appl. Numer. Methods* **4**, pp. 709-712, 1988.

[20] D. Field, Quantitative measures for initial meshes, *Int. J. Numer. Methods Engrg.* **47**, pp. 887-906, 2000.

[21] P. Grisvard, *Elliptic Problems in Nonsmooth Domains*, Pitman, Boston, 1985.

[22] L. Ju, Q. Du, and M. Gunzburger, Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations, *Parallel Comput.* **28**, pp. 1477-1500, 2002.

[23] L. Ju, M. Gunzburger, and W.D. Zhao, Adaptive finite element methods for elliptic PDEs based on conforming centroidal Voronoi Delaunay triangulations, submitted, 2005.

[24] S. Lloyd, Least squares quantization in PCM, *IEEE Trans. Infor. Theory* **28**, pp. 129-137, 1982.

[25] R. Lohner and P. Parikh, Generation of three-dimensional unstructured grids by the advancing-front method, *Int. J. Num. Meth. Fluid* **8**, pp. 1135-1149, 1988.

[26] J. MacQueen, Some methods for classification and analysis of multivariate observations, *Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Vol I, Ed. by L. Le Cam and J. Neyman, University of California, pp. 281-297, 1967.

[27] P. Moller and P. Hansbo, On advancing front mesh generation in three dimensions, *Int. J. Num. Meth. Engrg.* **38**, pp. 3551-3569, 1995.

[28] P.-O. Persson and G. Strang, A simple mesh generator in Matlab, *SIAM Review* **46**, pp. 329-345, 2004.

[29] M. Shephard and M. Georges, Automatic three-dimensional mesh generation technique by the finite element octree technique, *Int. J. Num. Meth. Eng.* **32**, pp. 709-749, 1991.

[30] J. Shewchuk, Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator, *Lecture Notes in Comput. Sci.* **1148**, Springer, New York, pp. 203-222, 1996.

[31] J. Shewchuk, What is a good linear element? Interpolation, conditioning and quality measures, *Proceedings of the 11th International Meshing Roundtable*, Sandia National Laboratories, Albuquerque, pp. 115-126, 2002.

[32] C.A. Taylor, M.T. Draney, J. P. Ku, D. Parker, B. N. Steele, K. Wang, and C.K. Zarins, Predictive medicine: Computational techniques in therapeutic decision-making, *Computer Aided Surgery* **4**, pp. 231-247, 1999.

[33] N. Weatherill and O. Hassan, Efficient three dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints, *Int. J. Num. Meth. Eng.* **37**, pp. 2005-2039, 1994.

Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA
E-mail: ju@math.sc.edu