Guaranteed-Quality Anisotropic Mesh Generation for Domains with Curves

Yusuke Yokosuka*   Keiko Imai†

Abstract

Anisotropic mesh generation is important for interpolation and numerical modeling. Recently, Labelle and Shewchuk proposed a two dimensional guaranteed-quality anisotropic mesh generation algorithm called Voronoi refinement. This algorithm treats only domains with straight lines as inputs. However, in many applications, input domains have many curves and the exact representation of curves is needed for efficient numerical modeling. In this paper, we extend the Voronoi refinement and propose it as a guaranteed-quality anisotropic mesh generation algorithm for domains with curves. Some experimental results are also shown.

1 Introduction

Mesh generation is used in interpolation including computer graphics, and numerical modeling including the finite element method. It has been shown that anisotropic meshes where the elements are elongated along specified directions are well suited for interpolation and numerical modeling [6]. In this paper, we consider two dimensional anisotropic mesh generation.

In anisotropic mesh generation, many heuristic solutions have been proposed [1, 2, 4, 7]. These algorithms work for all kinds of domains. However, meshes generated by heuristics are not unique and have no guaranteed property. Recently, Labelle and Shewchuk [3] have proposed an anisotropic mesh generation algorithm which guarantees that high quality meshes are generatedD This high quality means that there are no poor-quality triangles in the mesh. The algorithm can be applied to the cases where the input domain is a planar straight line graph (PSLG). However, it cannot be used for input domains with curves. Therefore, we aim to extend this algorithm to treat domains with curves.

There is an approach to handle domains with curves. If we first approximate an input curve with segments, the Voronoi refinement algorithm proposed by Labelle and Shewchuk can be used for the segments. The new points inserted by the algorithm are not on the original curves but on the approximate segments. It follows that the obtained mesh does not precisely approximate the boundary of the input domain. To overcome this, there is a way to divide the curve into smaller segments. If we use this method, the number of triangles in the mesh is undesirably large. From the above discussion, we find that where new points are to be inserted has to be decided during the execution of the algorithm.

*Department of Information and System Engineering, Graduate School of Science and Engineering, Chuo University, yoyu@imai-lab.ise.chuo-u.ac.jp

†Department of Information and System Engineering, Chuo University, imai@ise.chuo-u.ac.jp

2 Anisotropic Mesh

In this section, we first explain the relationship between a metric tensor and anisotropic mesh generation. Next, we define anisotropic Voronoi diagrams and anisotropic Delaunay triangulations. These definitions are introduced by Labelle and Shewchuk [3].

Consider a domain $\Omega \subseteq \mathbb{R}^d$. Suppose that for any point $p \in \Omega$, there is a *metric tensor* M_p. M_p is given as a symmetric positive definite matrix, and is used to measure length and angles from p. The distance between $q_1 \in \mathbb{R}^d$ and $q_2 \in \mathbb{R}^d$ as measured by p is defined as

$$d_p(q_1, q_2) = \sqrt{(q_1 - q_2)^{\mathrm{T}} M_p (q_1 - q_2)}.$$

Let $d_p(q) = d_p(p, q)$. In the same way, the angle $\angle_p q_1 q_2 q_3$ ($q_1, q_2, q_3 \in \mathbb{R}^d$) as measured by p is defined as

$$\angle_p q_1 q_2 q_3 = \arccos \frac{(q_1 - q_2)^{\mathrm{T}} M_p (q_3 - q_2)}{d_p(q_1, q_2) d_p(q_3, q_2)}.$$

Given a metric tensor M_p of a point p, define a *deformation tensor* F_p to be any matrix such that $F_p^{\mathrm{T}} F_p = M_p$ and $\det(F_p) > 0$. The *relative distortion* between p and q is defined as $\tau(p, q) = \tau(q, p) = \max\{||F_q F_p^{-1}||_2,\ ||F_p F_q^{-1}||_2\}$.

In anisotropic mesh generation, the metric tensors show the directions and size of scaling at each point in the domain. Triangles are elongated according to the metric tensors. Consider any point p in a triangle t. If t is equilateral as measured by p, t is elongated as to the metric tensor at p in physical space. Therefore, in anisotropic mesh generation, it is necessary that each triangle is equilateral as measured by any point within itself.

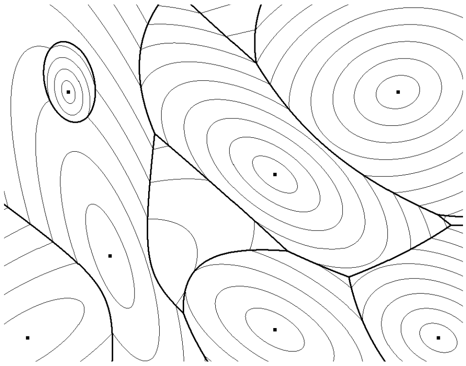


Figure 1: An anisotropic Voronoi diagram.

We briefly explain anisotropic Voronoi diagrams as proposed by Labelle and Shewchuk [3]. Let V be a set of points called sites. The *Voronoi cell* of a site $v \in V$ is

$$\mathrm{Vor}(v) = \{p \in \mathbb{R}^d \mid d_v(p) \leq d_w(p), \forall w \in V\}.$$

Any subset of sites $W \subseteq V$ define a Voronoi face $\mathrm{Vor}(W) = \bigcap_{w \in W} \mathrm{Vor}(w)$, which is a set of points equally close to the sites in W but no closer to any other. Every site of W is said to *own* $\mathrm{Vor}(W)$. The *anisotropic Voronoi diagram* of V is the arrangement of the non-empty faces $\{\mathrm{Vor}(W) \mid W \subseteq V, W \neq \emptyset, \mathrm{Vor}(W) \neq \emptyset\}$. Figure 1 shows an example of an anisotropic Voronoi diagram. 0-faces and 1-faces in the Voronoi diagram are called *Voronoi vertices* and *Voronoi arcs*, respectively.

The dual of an ordinary Voronoi diagram is the Delaunay triangulation. But the dual of an anisotropic Voronoi diagram is not generally a triangulation. Here, we describe the condition where the dual of the anisotropic Voronoi diagram is a triangulation. The *wedge* between two sites v, w is defined as

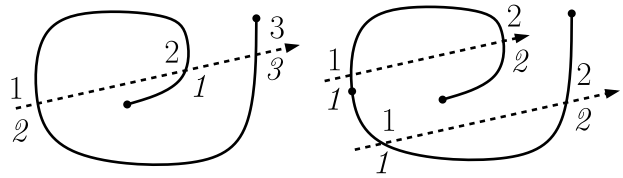
$$\begin{aligned}\mathrm{wedge}(v, w) \quad = \quad & \{q \in \mathbb{R}^d \mid (q - v)^{\mathrm{T}} M_v (w - v) > 0 \\ & \text{and } (q - w)^{\mathrm{T}} M_w (v - w) > 0\}.\end{aligned}$$

A Voronoi k-face $f \in \mathrm{Vor}(W)$ $(0 \leq k < d,\ W \subseteq V)$ is said to be *wedged* if for any pair of sites $v_1, v_2 \in W$ $(v_1 \neq v_2)$, any point $q \in f$ is inside the wedge(v_1, v_2).

We know the following property [3]. Let V be a set of sites in a general position and let D be the anisotropic Voronoi diagram of V. If all the Voronoi arcs and vertices in D are wedged, the dual of D is a triangulation. This triangulation is called the *anisotropic Delaunay triangulation*.

3 Voronoi refinement for Domains with Curves

The Voronoi refinement algorithm can be applied to polygonal domains with straight line segments. We want to extend the algorithm for curve-bounded domains with internal curves.



(a) A counter example (b) An example

Figure 2: A linear order.

First, we precisely describe the conditions of the input domains. The input in our algorithm is a Planar Regular Curve Graph X and a metric tensor field M. The definition of a Planar Regular Curve Graph is as follows.

Definition 1 *A Planar Regular Curve Graph (PRCG) is a set of sites and regular curves in a plane that satisfies three conditions:*

1. *For any curve contained in a PRCG, two endpoints of the curve are sites in the PRCG.*
2. *The site is an endpoint of a curve, or a point that is not on a curve.*
3. *Curves are permitted to intersect only at their endpoints.*

These conditions are equal to the conditions of a Planar Straight Line Graph (PSLG), which is a set of sites and line segments. If a set of curves is that of segments, a PRCG is identical with a PSLG. Additionally, we assume that any pair of curves has no two common endpoints. Moreover, we suppose that the collinear points on any curve occur in a linear order (Figure 2(b)), and for any two curves the two convex hulls are disjoint with the exception of shared endpoints. Even if these additional conditions are not satisfied in a PRCG, we can make changes to satisfy them in the preprocessing.

Let $\Omega \subset \mathbb{R}^2$ be a finite domain in which we want to generate a triangulation. We assume that curves and points in the interior of Ω consist of a PRCG and the boundary of Ω is also a PRCG. In total, the PRCG is designated as $X(\Omega)$. The domain Ω that satisfies this condition is called a PRCG domain.

Our algorithm generates a quality anisotropic mesh so that the following condition is satisfied: Each angle in every triangle t in the mesh is larger than or equal to θ_{bound} as measured by any point in t. In this paper, a triangle that doesn't satisfy this condition is called a poor-quality triangle.

We define some concepts to give an outline of our algorithm for the generation of a quality anisotropic mesh. If a Voronoi cell $\mathrm{Vor}(w)$ of a site w contains a curve c that does not contain w, c is said to be encroached upon w (Figure 3). If c is encroached, split it by inserting a site z in $c \cap \mathrm{Vor}(w)$. At this time,

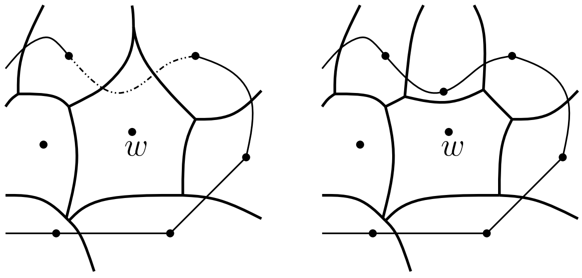


Figure 3: A curve encroached upon a site w is split.

if possible, the curve c is split so that $d_a(z) = d_b(z)$ where a and b are the endpoints of c. We consider the case that the point z such that $d_a(z) = d_b(z)$ does not lie in $\mathrm{Vor}(w)$ of an encroaching site w. In this case, let z' be the point that is in $c \cap \mathrm{Vor}(w)$ and as close to the point z as possible, and z' is inserted as a new site. This operation is called a "split".

Definition 2 *A point q in domain Ω is called a* violator *if q satisfies either one of the next two conditions.*

- *q lies on a Voronoi arc $\mathrm{Vor}(\{v, w\})$ $(v, w \in X)$ that is not wedged, and q doesn't lie in* $\mathrm{wedge}(v, w)$. *In this case, q is called a wedge violator.*
- *q is a Voronoi vertex whose dual is an inverted or a poor-quality triangle. q is called a poor violator.*

Outline of our algorithm
Step 1: Construct the anisotropic Voronoi diagram D of sites in X. For each encroached curve c, split the curve c. After this, every curve is contained in the Voronoi cells of its endpoints.
Step 2: Select any violator q. If q does not encroach, then q is inserted as a new site. Otherwise, q is not inserted and c is split. After that, update the anisotropic Voronoi diagram.

In our algorithm, $\sin \theta_{\mathrm{bound}}$ is strictly less than $1/4$ ($\arcsin(1/4) \approx 14.4°$) to guarantee termination of our algorithm. The detail of the proof is described in the next section.

4 Termination of the algorithm

In this section, we prove that our algorithm terminates and guarantees good quality. Most of the proof follows the same approach as in [3]. We give only a short summary of the proof here, further details about it are given in the full paper.

We assume that each angle of any two curves in the input PRCG X is more than $60°$. Under this assumption, we can prove that our algorithm terminates for $\theta_{\mathrm{bound}} < \arcsin(1/4)$. We will show that the algorithm does not bring segments with shorter distances than that of any segment existing in each step. In the proof, we use the Euclidean distance because we can prove the same fact in the same way even where we use a metric tensor.

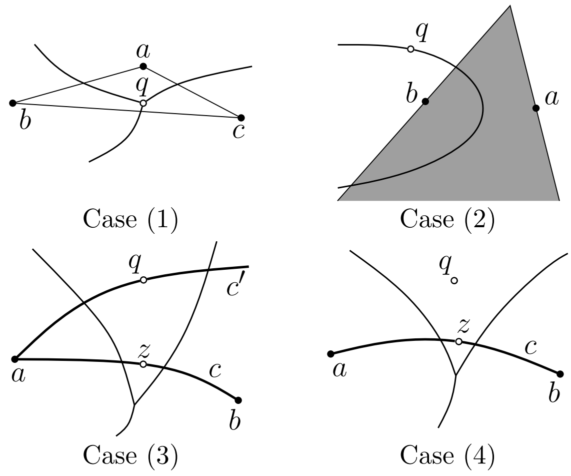


Figure 4: Inserting a new site.

There are four cases in which a new site is inserted during the execution of the algorithm (Figure 4); (1) The new site is a poor violator; (2) It is a wedge violator; (3) A site on a curve c' encroaches another curve c and a point on c is inserted; (4) The algorithm tries to insert a violator q whose $\mathrm{Vor}(q)$ encroaches another curve, and inserts a new site on the curve instead of q.

First, we consider the case of (1). Let q be a poor violator. In this case, the dual triangle of q has an angle that is less than θ_{bound}. l denotes the shortest edge of the triangle. We can show that the length of the new segment that appears by inserting q is more than Bl, where $B = 1/(2 \sin \theta_{\mathrm{bound}})$. In the case of (2), with further investigation into details, we also have it that the length of the new segment that appears by inserting q is more than Bl.

Next, we examine the case of (3). In this case, the site q on a curve c' encroaches another curve c. The situation that c and c' are disjoint is of no matter in the proof of the termination. Therefore, we assume that c and c' have a common endpoint a. Let b be the other endpoint of c. The new site z is chosen as the nearest point in $\mathrm{Vor}(q) \cap c$ from the Voronoi edge determined by a and b. In the triangle $\triangle qaz$, the shortest edge among the newly created edges is qz because z is in $\mathrm{Vor}(q)$. Therefore, $\overline{qz} < \overline{az}$ and this means $\angle zqa > \angle qaz$. It is shown that $\angle qaz$ is the smallest in the triangle $\triangle qaz$ from our assumption that $\angle qaz > 60°$. Therefore, we get $\overline{qz} \geq \overline{qa}$ and the new segment is longer than the existing segments.

We consider the last case (4) is which a violator q encroaches a curve c, and c is split. Note that q is not inserted in this case. The algorithm inserts the point z, which is the nearest point in $\mathrm{Vor}(q) \cap c$ from

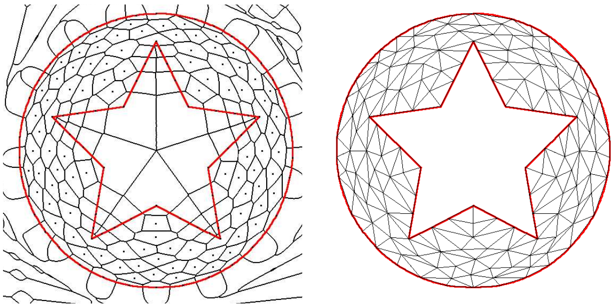


Figure 5: Input with straight lines and a circle.

the Voronoi edge determined by a and b, where a and b are endpoints of c. Without loss of generality, we assume that q is nearer to b than a. In the triangle $\triangle qbz$, we can get that $\overline{qz} + \overline{bz} > \overline{qb}$ from the triangle inequality. Moreover, $\overline{qz} < \overline{bz}$ because z is in $\mathrm{Vor}(q)$. Summarizing the above discussion, we have $\overline{bz} > \overline{qb}/2$. q is a violator and the distance between any other point and q is greater than Bl. If $B > 2$, the distance from the other site to q is not shorter than the length of any existing segment. This condition is satisfied when $\theta_{\mathrm{bound}} < \arcsin(1/4)$.

From the above discussion, the algorithm does not create a shorter segment than the existing segments in the case of $\theta_{\mathrm{bound}} < \arcsin(1/4)$. Let l be the shortest length among the segments created during the algorithm. We draw a circle whose center is a site with radius $l/2$. There are only a finite number of such circles in Ω. From this fact, we have that our algorithm terminates. When there is no violator, the algorithm also terminates. Therefore, in the obtained mesh, there is no angle that is smaller than θ_{bound}.

5 Experimental Results

In this section, we show some experimental results. We generate anisotropic meshes for 10 domains with curves by using the algorithm we described in Section 3. We assign $\theta_{\mathrm{bound}} = 14.4°$ to guarantee that our algorithm terminates. Some of these results are shown in Figure 5 and 6. Moreover, our algorithm often terminates in practice even if the specified angle is greater than $\arcsin(1/4)$.

6 Conclusion

In this paper, we proposed a guaranteed-quality anisotropic mesh generation algorithm for domains with curves. This algorithm has as its basis a Voronoi refinement algorithm [3]. This proposed algorithm generates an anisotropic mesh in which no triangle has an angle smaller than 14.4°, as measured by any point in the triangle. We also gave some experimental results. The results showed that the proposed algorithm

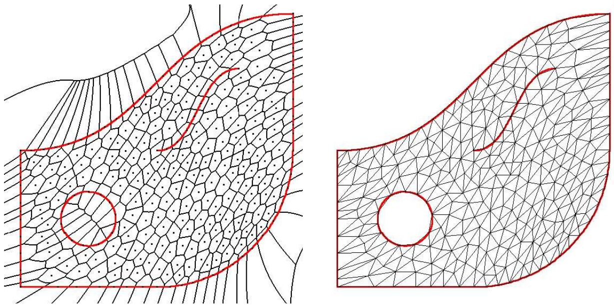


Figure 6: Input with straight lines, circles, and Bezier curves.

does indeed generate guaranteed-quality anisotropic meshes.

There are three proposed future works. The upper bound of the specified angle θ_{bound} in the original Voronoi refinement algorithm is 20.7°. However, our upper bound is 14.4°. Therefore, one future work is to improve the upper bound of the specified angle θ_{bound}. The second will be to take away the assumption of the lower bound of input angles. In the Delaunay refinement algorithm, Pav and Walkington's approach [5] accepts inputs without a lower bound for the input angles. But as yet there is no Voronoi refinement algorithm with such a property. The third work will be to improve our method for three dimensional anisotropic mesh generation.

References

[1] F. J. Bossen, P. S. Heckbert, A Pliant Method for Anisotropic Mesh Generation. *Fifth International Meshing Roundtable*, 63–74, 1996.

[2] P. George, H. Borouchaki, Delaunay Triangulation and Meshing: Application to Finite Elements. Hermes, Paris, 1998.

[3] F. Labelle, J. R. Shewchuk, Anisotropic Voronoi Diagrams and Guaranteed-Quality Anisotropic Mesh Generation. *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, 191–200, 2003.

[4] X. Li, S. Teng, A. Üngör, Biting Ellipses to Generate Anisotropic Mesh. *Eighth International Meshing Roundtable*, 97–108, 1999.

[5] S. Pav, N. Walkington, Delaunay Refinement by Corner Lopping. *Fourteenth International Meshing Roundtable*, 165–182, 2005.

[6] J. R. Shewchuk, What Is a Good Linear Element? Interpolation, Conditioning, and Quality Measures. *Eleventh International Meshing Roundtable*, 115–126, 2002.

[7] K. Shimada, A. Yamada, T. Itoh, Anisotropic Triangulation of Parametric Surfaces via Close Packing of Ellipsoids. *International Journal of Computational Geometry and Applications*, 10(4), 400–424, 2000.