MESH SMOOTHING SCHEMES BASED ON OPTIMAL DELAUNAY TRIANGULATIONS

Long Chen*

Math Department, The Pennsylvania State University, State College, PA, U.S.A., lyc102@psu.edu

ABSTRACT

We present several mesh smoothing schemes based on the concept of optimal Delaunay triangulations. We define the optimal Delaunay triangulation (ODT) as the triangulation that minimizes the interpolation error among all triangulations with the same number of vertices. ODTs aim to equidistribute the edge length under a new metric related to the Hessian matrix of the approximated function. Therefore we define the interpolation error as the mesh quality and move each node to a new location, in its local patch, that reduces the interpolation error. With several formulas for the interpolation error, we derive a suitable set of mesh smoothers among which Laplacian smoothing is a special case. The computational cost of proposed new mesh smoothing schemes in the isotropic case is as low as Laplacian smoothing while the error-based mesh quality is provably improved. Our mesh smoothing schemes also work well in the anisotropic case.

Keywords: anisotropic mesh adaptation, Delaunay triangulation, Voronoi tessellation, mesh smoothing

1. INTRODUCTION

In this paper, we will derive several, old and new, mesh smoothing schemes based on optimal Delaunay triangulations. The optimal Delaunay triangulation (ODT) introduced in [1] is the triangulation that minimizes the interpolation error among all triangulations with the same number of vertices. By moving a node to a new position, in its local patch, such that the interpolation error is reduced, we obtain a suitable set of mesh smoothers for both isotropic and anisotropic mesh adaptations.

Many mesh generation methods aim to generate a mesh as good as possible according to some mesh qualities. In the context of finite element methods, it is shown that the angles of triangles should remain bounded away from 0 and π if one wants to control the interpolation error in H^1 norm [2]. Hence certain geometric qualities are defined to exclude the large and small angles in the triangulation. On the other hand in order to approximate an anisotropic function, (with sharp boundary layers or internal layers) long thin elements can be good for linear approximation if we measure the error in L^p norm rather than in H^1 norm [3, 4]. It is shown that the shape of elements should be stretched according to a metric which is assigned by (a modification of) the Hessian matrix of the object function [5, 6, 7, 8, 4], and thus when the metric is highly anisotropic, the desirable mesh may contain large or small angles. In the meantime, the density of the nodes should be distributed according to some norm of the determinant of the Hessian matrix. In other words, the volume of each element under the new metric is almost equidistributed; see [7, 8, 4] for details.

Inspired by the optimal L^p error estimates in [4], we think the mesh quality is a function dependent concept. Therefore, we define the overall quality for a triangulation by $\|f - f_I\|_{L^p(\Omega)}$, where f is the function of interest and f_I is the linear interpolant based on the triangulation. We minimize this error-based quality by several local mesh improvements.

There are mainly three types of mesh improvement methods: (1) refinement or coarsening, (2) edge swapping, and (3) mesh smoothing. According to our understanding of the mesh quality, the refinement and the coarsening mainly try to optimize the mesh density, while edge swapping and mesh smoothing mainly aim to optimize the shape regular-

*This work was supported in part by NSF DMS-0074299, NSF DMS-0209497,NSF DMS-0215392 and the Center for Computational Mathematics and Application at Penn State.

ity. In [9], we have developed the edge-based refinement and coarsening. Our edge-based refinement will automatically result in a conform triangulation and thus save a lot of work of programing. In [1], we show that the empty circle criteria is equivalent to the interpolation error criteria when $f(\mathbf{x}) = \|\mathbf{x}\|^2$. The termination of the edge swapping is trivial since after each iteration the interpolation error is decreased. By choosing f of interest, the edge swapping is generalized to the anisotropic case. We will consider the mesh smoothing based on ODTs in this paper.

There are mainly two types of smoothing methods, namely Laplacian smoothing and optimization-based smoothing. Laplacian smoothing [10], in its simplest form, is to move each vertex to the arithmetic average of the neighboring points. It is easy to implement and require a very low computational cost, but it operates heuristically and does not guarantee an improvement in the geometric mesh qualities. Thus people proposed an optimization-based smoothing: the vertex is moved so as to optimize some mesh quality [11, 12, 13]. The price for the guaranteed quality improvement is that the computational time involved is much higher than that of Laplacian smoothing.

Our mesh smoothing schemes essentially belong to the optimization-based smoothing. Instead of geometric mesh qualities, we try to minimize the interpolation error in the local patch. With several formulas of the interpolation error, in isotropic case, we could solve the optimization problem exactly and thus the computational cost is as low as that of Laplacian smoothing, while the error-based mesh quality is guaranteed to be improved. If we change $f(\mathbf{x}) = \|\mathbf{x}\|^2$ to a general function or a metric, we get anisotropic mesh smoothing schemes which are useful in the mesh adaptation for solving partial differential equations [14, 4]. Of course, the computational cost in the anisotropic case is a little bit higher.

The rest of this paper is organized as follows: in Section 2, we define the error-based and metric-based mesh qualities, introduce the concept of optimal Delaunay triangulations and derive formulas for the error-based mesh quality. In Section 3, we introduce centroid Voronoi tessellations as the dual of ODTs. In Section 4 we present several mesh smoothing schemes by considering the optimization of the interpolation error. We finally report some numerical experiments in two dimensions in Section 5 to show the efficiency of our mesh smoothing schemes.

2. OPTIMAL DELAUNAY TRIANGULATIONS

The Delaunay triangulation (DT) of a finite set of points S, one of the most commonly used unstructured triangulations, can be defined by the empty sphere property: no vertices in S are inside the circumsphere of any simplex in the triangulation. There are many optimality characterizations for Delaunay triangulation [15], among which the most well known is that in two dimensions it maximizes the minimum angle of triangles in the triangulation [16]. In [1], we characterized the Delaunay triangulation from a function approximation point of view.

Let us denote $Q(\mathcal{T}, f, p) = \|f - f_{I,\mathcal{T}}\|_{L^p(\Omega)}$, where $f_{I,\mathcal{T}}(\mathbf{x})$ is the linear interpolation of f based on a triangulation $\mathcal{T}$ of a domain $\Omega \subset \mathbb{R}^n$. Let Ω be the convex hull of S and $\mathcal{P}_S$ be the set of all triangulations of Ω whose vertices are points in S. We have shown in [1] that

$$Q(DT, \|\mathbf{x}\|^2, p) = \min_{\mathcal{T} \in \mathcal{P}_S} Q(\mathcal{T}, \|\mathbf{x}\|^2, p), \qquad (1)$$

for $1 \leq p \leq \infty$, which is a generalization of previous work [6, 3, 17] to higher dimensions. Delaunay triangulation is therefore characterized as the optimal triangulation for piecewise linear interpolation to isotropic function $\|\mathbf{x}\|^2$ for a given point set in the sense of minimizing the interpolation error in $L^p(1 \leq p \leq \infty)$ norm. For a more general function, a function-dependent Delaunay triangulation is then defined to be an optimal triangulation that minimizes the interpolation error for this function and its construction can be obtained by a simple lifting and projection procedure.

The optimal Delaunay triangulation (ODT) introduced in [1] minimizes the interpolation error among all triangulations with the same number of vertices. Let $\mathcal{P}_N$ stand for the set of all triangulations with at most N vertices. $\mathcal{T}^* \in \mathcal{P}_N$ is an optimal Delaunay triangulation if

$$Q(\mathcal{T}^*, f, p) = \inf_{\mathcal{T} \in \mathcal{P}_N} Q(\mathcal{T}, f, p), \qquad (2)$$

for some $1 \leq p \leq \infty$. Such a function-dependent optimal Delaunay triangulation is proved to exist for any given convex continuous function [1].

Furthermore, we have the following asymptotic lower bound for strictly convex functions [4]:

$$\begin{aligned} &\liminf_{N \to \infty} N^{2/n} Q(\mathcal{T}_N, f, p) \\ &\geq LC_{n,p} \| \sqrt[n]{\det(\nabla^2 f)} \|_{L^{\frac{pn}{2p+n}}(\bar{\Omega})}, \end{aligned}$$

where $LC_{n,p}$ is a constant only depending on n and p. The equality holds if and only if all edges are asymptotic equal under the metric

$$H_p = (\det \nabla^2 f)^{-\frac{1}{2p+n}} \nabla^2 f. \qquad (3)$$

When $f(\mathbf{x}) = \|\mathbf{x}\|^2$, H_p is the Euclidean metric and the equality holds if all edge lengths of the triangulation are equal. In $\mathbb{R}^2$, the optimal one consists of equilateral triangles which is the ideal case for many mesh adaptation schemes. By choosing f of interest, we can obtain anisotropic meshes by minimizing the interpolation error. We, thus, consider mesh adaptation techniques as optimization methods to minimize the interpolation error and define the interpolation error as our mesh quality. It is worthy noting that Berzins [18] gave a solution dependent mesh quality and Bank and Smith [11] also derived the distortion quality from error point of view. More recently, Shewchuk [19] also looked at the mesh quality from the interpolation point of view.

Definition. Suppose Ω is a domain in $\mathbb{R}^n$ with triangulation $\mathcal{T}$, $f \in \mathcal{C}^1(\bar{\Omega})$, f_I the piecewise linear and global continuous interpolation of f based on $\mathcal{T}$ and p, $1 \le p \le \infty$, we define *error-based mesh quality* $Q(\mathcal{T}, f, p)$ as

$$Q(\mathcal{T}, f, p) = \|f - f_I\|_{L^p(\Omega)}.$$

For a quadratic convex function f and an integer $p \ge 1$, it was shown in [4] that

$$|\tau|_{H_p}\Big(\sum_{i,j,i<j}^{n} d^2_{\tau_{ij},H_p}\Big)^p \approx Q(\tau, f, p),$$

where d_{τ_{ij},H_p} and $|\tau|_{H_p}$ are the edge length and the volume of τ under the metric H_p given in (3), respectively. $A \approx B$ means there exist two constant C_1, C_2 such that $C_1 A \le B \le C_2 A$.

We will derive several formulas of $Q(\mathcal{T}, f, 1)$ for later use. The following lemma can be found in [20, 4].

Lemma 1. For a convex quadratic function f

$$Q(\tau, f, 1) = \frac{|\tau|}{2(n+2)(n+1)} \sum_{i,j}^{n} \|\mathbf{x}_i - \mathbf{x}_j\|^2_{\nabla^2 f},$$

where $\|\mathbf{v}\|_{\nabla^2 f} = \mathbf{v}^T \nabla^2 f \mathbf{v}$.

In particular, $f(\mathbf{x}) = \|\mathbf{x}\|^2$ corresponds to the Euclidean metric. By Lemma 1 and the inequality between the total edge length and the volume of a simplex, we see

$$Q(\tau, \|\mathbf{x}\|^2, 1) \ge C_n |\tau|^{1+2/n} \tag{4}$$

and the equality holds if and only if τ is equilateral. In (4), C_n is a constant only depending on the dimension n and it can be calculated by taking an equilateral simplex τ. Thus the distortion metric [11, 18] for a simplex can be defined by the following ratio (or its reciprocal)

$$\frac{Q(\tau, \|\mathbf{x}\|^2, 1)}{C_n |\tau|^{1+2/n}} = \frac{\sum d^2_{i,j}}{C_n |\tau|^{2/n}},$$

where C_n is used to normalize the quality, and the optimization of the distortion metric will lead to equilateral simplexes. Lemma 1 shows the relation between the distortion metric and the interpolation error. We will look at the interpolation error and optimize it directly.

With Lemma 1, we get an interesting formula for our error-based mesh quality.

Theorem 1. For a convex quadratic function f

$$Q(\mathcal{T}, f, 1) = \frac{1}{n+1} \sum_{i=1}^{N} \int_{\Omega_i} \|\mathbf{x} - \mathbf{x}_i\|^2_{\nabla^2 f} d\mathbf{x}. \tag{5}$$

Proof. Let $\{\lambda_i(\mathbf{x})\}_{i=1}^{n+1}$ be the barycenter coordinate of $\mathbf{x}$ in the simplex τ. Then $\mathbf{x} = \sum_{i=1}^{n+1} \lambda_i \mathbf{x}_i$ and

$$\begin{aligned}
&\sum_{k=1}^{n+1} \int_\tau \|\mathbf{x} - \mathbf{x}_k\|^2_{\nabla^2 f} \\
=\ & \sum_{i,j,k=1}^{n+1} \int_\tau \lambda_i \lambda_j (\mathbf{x}_i - \mathbf{x}_k)^T \nabla^2 f (\mathbf{x}_j - \mathbf{x}_k) \\
=\ & \frac{|\tau|}{(n+2)(n+1)} \sum_{i,j,k=1}^{n+1} (\mathbf{x}_i - \mathbf{x}_k)^T \nabla^2 f (\mathbf{x}_j - \mathbf{x}_k) \\
=\ & \frac{n+1}{2} \frac{|\tau|}{(n+2)(n+1)} \sum_{i,j}^{n+1} \|\mathbf{x}_i - \mathbf{x}_j\|^2_{\nabla^2 f} \\
=\ & (n+1) \int_\tau |f_I(\mathbf{x}) - f(\mathbf{x})|.
\end{aligned}$$

The last equality follows from Lemma 1. The third one is obtained by summing up the following basic identity:

$$\begin{aligned}
\|\mathbf{x}_i - \mathbf{x}_j\|^2_{\nabla^2 f} =\ & \|\mathbf{x}_i - \mathbf{x}_k\|^2_{\nabla^2 f} + \|\mathbf{x}_j - \mathbf{x}_k\|^2_{\nabla^2 f} \\
& -2(\mathbf{x}_i - \mathbf{x}_k)^T \nabla^2 f (\mathbf{x}_j - \mathbf{x}_k).
\end{aligned}$$

Noting that

$$\sum_{i=1}^{N_E} \sum_{k=1}^{n+1} \int_{\tau_i} \|\mathbf{x} - \mathbf{x}_{\tau,k}\|^2_{\nabla^2 f} = \sum_{i=1}^{N} \int_{\Omega_i} \|\mathbf{x} - \mathbf{x}_i\|^2_{\nabla^2 f},$$

we get the result. Here N_E is the number of elements in the triangulation. Q.E.D.

This formula motivates a natural definition of a metric-based mesh quality.

Definition. For a given triangulation $\mathcal{T}$ and metric G, we define a metric-based mesh quality $Q(\mathcal{T}, G, 1)$ as

$$Q(\mathcal{T}, G, 1) = \frac{1}{n+1} \sum_{i=1}^{N} \int_{\Omega_i} \|\mathbf{x} - \mathbf{x}_i\|^2_G d\mathbf{x}.$$

For a convex function we can get another formula which can be found in our recent work [1]. Since it is the basis of our mesh smoothing schemes, we present the proof here.

Theorem 2. For a convex function f,

$$Q(\mathcal{T}, f, 1) = \frac{1}{n+1} \sum_{\mathbf{x}_i \in \mathcal{T}} f(\mathbf{x}_i)|\Omega_i| - \int_\Omega f(\mathbf{x}) d\mathbf{x}. \tag{6}$$

Proof. Because $f_I(\mathbf{x}) \ge f(\mathbf{x})$ in Ω, we get

$$\begin{aligned}
& Q(\mathcal{T}, f, 1) \\
=\ & \sum_{\tau \in \mathcal{T}} \int_\tau f_I(\mathbf{x}) d\mathbf{x} - \int_\Omega f(\mathbf{x}) d\mathbf{x} \\
=\ & \frac{1}{n+1} \sum_{\tau \in \mathcal{T}} \left(|\tau| \sum_{k=1}^{n+1} f(\mathbf{x}_{\tau,k}) \right) - \int_\Omega f(\mathbf{x}) d\mathbf{x} \\
=\ & \frac{1}{n+1} \sum_{\mathbf{x}_i \in \mathcal{T}} f(\mathbf{x}_i)|\Omega_i| - \int_\Omega f(\mathbf{x}) d\mathbf{x}.
\end{aligned}$$

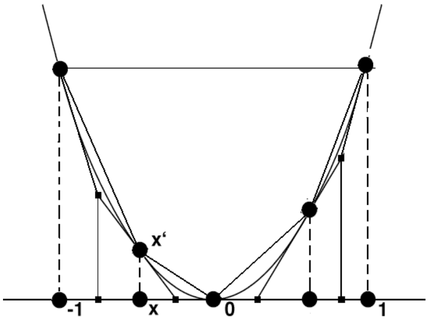


Figure 1: Lifting and Projection of a Delaunay Triangulation and a Voronoi Tessellation in one dimension

Q.E.D.

We conclude this section by considering the geometric meaning of the interpolation error. To make it clear, let us introduce some notation first. We identify $\mathbb{R}^{n+1}$ as $\mathbb{R}^n \times \mathbb{R}$. A point in $\mathbb{R}^{n+1}$ can be written as $(\mathbf{x}, x_{n+1})$, where $\mathbf{x} \in \mathbb{R}^n$ and $x_{n+1} \in \mathbb{R}$. For a point $\mathbf{x} \in \mathbb{R}^n$, we can lift it to the paraboloid $(\mathbf{x}, \|\mathbf{x}\|^2)$ living in $\mathbb{R}^{n+1}$ and denote this lifting operator as $'$, namely $\mathbf{x}' = (\mathbf{x}, \|\mathbf{x}\|^2)$. For a given point set S in $\mathbb{R}^n$, we then have a set of points S' in $\mathbb{R}^{n+1}$ by lifting points in S to the paraboloid. For the sake of simplicity, in the sequel we choose Ω as an inscribed polytope of B_n, the unit ball in $\mathbb{R}^n$. The graph of function $f(\mathbf{x}) = \|\mathbf{x}\|^2$ is the paraboloid and $f(B_n) \cup (B_n, 1)$ bound a convex body C. For a triangulation $\mathcal{T}$, the graph of f_I and $(\Omega, 1)$ will bound a polytope P^i. Since $f_I(\mathbf{x}) \geq f(\mathbf{x})$ and $f_I(\mathbf{x}_i) = f(\mathbf{x}_i)$, the polytope P^i can be seen as an inscribed polytope approximation to the convex body C. P^i is convex if and only if the underling triangulation is a Delaunay triangulation. Actually this is a characterization of Delaunay triangulation and called the lifting method [21]. See Figure 1 for an illustration in one dimension. From a function approximation point of view, it is easy to see that the convex polytope P^i is the optimal linear approximation to the paraboloid for a fixed points set since $Q(\mathcal{T}, f, 1)$ is nothing but their volume difference.

Optimal Delaunay triangulations with respect to $Q(\mathcal{T}, \|\mathbf{x}\|^2, 1)$ is the optimal inscribed polytope $P^i \in \mathcal{P}_N^i$ in the sense of minimizing the volume difference, where we use superscript i to indicate that it is the set of inscribed polytopes. The optimal inscribed polytope approximation to a general convex body is also well studied in the literature (see, for example, Gruber [22]). Note that the graph of f_I can be thought as an approximation of the boundary surface of the convex body C. The results and algorithms developed in the optimal polytope approximation can be applied to surface mesh generation and simplification. We would like to point out that in this case the metric should correspond to the second fundamental form of the surface [23].

3. CENTROID VORONOI TESSELLATIONS

In this section, we understand the Voronoi tessellations as circumscribe polytopes approximation of the paraboloid. We measure the approximation error by the volume difference. The optimal one is called a centroid Voronoi tessellation (CVT) and it is, more or less, the dual of an ODT.

We begin with the classic definition of Voronoi tessellations (or Voronoi diagrams).

Definition. Let Ω be an open set in $\mathbb{R}^n$ and $S = \{\mathbf{x}_i\}_{i=1}^N \subset \Omega$. For any $\mathbf{x}_i \in S$, we define the Voronoi region of $\mathbf{x}_i$ as

$$V_i = \{\mathbf{x} \in \Omega, s.t. \|\mathbf{x} - \mathbf{x_i}\| < \|\mathbf{x} - \mathbf{x_j}\|\}.$$

Then $\bar{\Omega} = \sum \bar{V}_i$. We call this partition $\mathcal{V}$ a *Voronoi tessellation* or *Voronoi diagram* of Ω and points $\{\mathbf{x}_i\}$ *generators*.

If we lift generators to the paraboloid $(\mathbf{x}, \|\mathbf{x}\|^2)$, we can characterize the Voronoi tessellation as the vertical projection of an upper convex envelope of tangential hyperplanes at those points [24]. Note that the envelope will form a circumscribed polytope P^c of C. Thus we can understand the VT as a circumscribe polytope approximation; See Figure 1. The duality of VT and DT can be understand as the polar duality [15] of the inscribed and circumscribe polytopes.

Theorem 3. The volume difference between P^c and C is

$$D(\mathcal{V}, \|\mathbf{x}\|^2, 1) := \sum_{i=1}^{N} \int_{V_i} \|\mathbf{x} - \mathbf{x}_i\|^2 d\mathbf{x}. \tag{7}$$

Proof. Let $\{\mathbf{x}_i\}_{i=1}^{n+1}$ be vertices of a simplex τ and $TM_{\mathbf{x}_i'}$ the tangential hyperplane of paraboloid at $\mathbf{x}_i'$ which is

$$x_{n+1} = \|\mathbf{x}\|^2 - \|\mathbf{x} - \mathbf{x}_i\|^2. \tag{8}$$

It is clear that the point $(\mathbf{x}_o, \|\mathbf{x}_o\|^2 - R^2)$ satisfies (8) for $i = 1, 2, ...n+1$, where $\mathbf{x}_o$ and R are the center and radius of the circumscribe sphere of τ. The vertical projection of the upper convex envelope $\mathcal{V}'$ of $TM_{\mathbf{x}_i'}$ is the Voronoi tessellation.

By the construction of VT, we see that the part of boundary of P^c which is projected to Voronoi region V_i is supported by the tangent hyperplane $TM_{\mathbf{x}_i'}$. Thus by (8) the difference of the volume is:

$$\sum_{i=1}^{N} \int_{V_i} \left(\|\mathbf{x}\|^2 - x_{n+1}\right) = \sum_{i=1}^{N} \int_{V_i} \|\mathbf{x} - \mathbf{x}_i\|^2 d\mathbf{x}.$$

We can generalize this quality with respect to any density function $\rho(\mathbf{x})$, which is a positive function defined on Ω and $\int_\Omega \rho(\mathbf{x}) d\mathbf{x} = 1$.

Definition. Let $\rho(\mathbf{x})$ be a density function in Ω. For a Voronoi tessellation $\mathcal{V}$ of Ω corresponding to generators $\{\mathbf{x}_i\}_{i=1}^N$, we define

$$D(\mathcal{V}, \rho(\mathbf{x}), 1) = \sum_{i=1}^{N} \int_{V_i} \rho(\mathbf{x}) \|\mathbf{x} - \mathbf{x}_i\|^2 d\mathbf{x}. \tag{9}$$

A dual concept of the optimal Delaunay triangulations or the optimal inscribed polytope approximations is the optimal Voronoi tessellations or the optimal circumscribe polytope approximations by minimizing $D(\mathcal{V}^*, \rho(\mathbf{x}), 1)$.

Definition. $\mathcal{V}^*$ is a centroid Voronoi tessellation if and only if

$$D(\mathcal{V}^*, \rho(\mathbf{x}), 1) = \min_{\mathcal{V} \in \mathcal{P}_N} D(\mathcal{V}, \rho(\mathbf{x}), 1).$$

Here $\mathcal{P}_N$ stands for the set of all Voronoi tessellation with at most N generators.

Why is it called centroid Voronoi tessellation? Because for a CVT, the generator $\mathbf{x}_i$ is also the centroid of its Voronoi region V_i, i.e.

$$\mathbf{x}_i = \frac{\int_{V_i} \mathbf{x}\rho(\mathbf{x})}{\int_{V_i} \rho(\mathbf{x})}.$$

The proof is very simple. Let $\mathbf{x}_i$ be the centroid of V_i. For any point $\mathbf{z}_i \in V_i$, we have

$$\int_{V_i} ||\mathbf{x} - \mathbf{x_i}||^2 \rho(\mathbf{x}) = \int_{V_i} (\mathbf{x} - \mathbf{x_i}) \cdot (\mathbf{x} - \mathbf{z_i})\rho(\mathbf{x})$$
$$\leq (\int_{V_i} ||\mathbf{x} - \mathbf{x_i}||^2 \rho(\mathbf{x}))^{1/2} (\int_{V_i} ||\mathbf{x} - \mathbf{z_i}||^2 \rho(\mathbf{x}))^{1/2}.$$

Thus

$$\int_{V_i} ||\mathbf{x} - \mathbf{x_i}||^2 \rho(\mathbf{x}) \leq \int_{V_i} ||\mathbf{x} - \mathbf{z_i}||^2 \rho(\mathbf{x}).$$

As we know, VT is the dual of DT. A natural question arises: is a CVT the dual of an ODT? It is interesting to compare (5) with (7). The difference of those two quantities mainly lies in the different decomposition of Ω. For a VT, it is a partition of Ω, while for a triangulation it is an overlapping decomposition of Ω. We conjecture that in the Euclidean metric, they are the dual of each other asymptotically. Indeed, this conjecture is true for one and two dimensions since in both cases the ODT and CVT for the Euclidean metric are known and happen to be the dual of each other. It is also true if we measure the difference in L^∞ norm since both of them asymptotically coincide with the optimal sphere covering scheme [20, 25]. But for general L^p norm in dimensions $n \geq 3$, the answer is not known yet.

For various important and interesting applications of CVTs, we refer to a nice review of Du et. al. [26]. Nowadays the theories and algorithms of CVTs are successfully applied to mesh generation and adaptation [27], both for general surface grid generation [28], anisotropic mesh generation [29] and mesh optimization in three dimensions [30]. We believe ODT shall also play an important role in the mesh generation and adaptation. This paper is to show the application of ODTs to the mesh smoothing.

4. MESH SMOOTHING SCHEMES

Mesh smoothing is a local algorithm which aims to improve the mesh quality, mainly the shape regularity, by adjusting

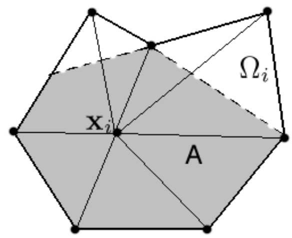


Figure 2: The feasible region in a local patch

the location of a vertex $\mathbf{x}_i$ in its local patch Ω_i, which consists of all simplexes containing $\mathbf{x}_i$, without changing the connectivity. To ensure that the moving will not destroy a valid triangulation, namely non-overlapping or inverted simplexes generated, we perform an explicit check, which is necessary when the patch is concave. Several sweeps through the mesh can be performed to improve the overall mesh quality. A general mesh smoothing algorithm is listed below:

General mesh smoothing algorithm
For k=1:step
For i=1:N
$\mathbf{x}^* = smoother(\mathbf{x}_i, \Omega_i)$
If $\mathbf{x}^*$ is acceptable then $\mathbf{x}_i = \mathbf{x}^*$
End
End

The key in the mesh smoothing is the smoother. Namely how to compute the new location by using the information in the local patch. Because the mesh may contain millions of vertices, it is critical that smoother function is computationally inexpensive. Laplacian smoothing, the simplest inexpensive smoother, is to move each vertex to the arithmetic average of the neighboring points.

Laplacian smoother

$$\mathbf{x}^* = \frac{1}{k} \sum_{\mathbf{x}_j \in \Omega_i, \mathbf{x}_j \neq \mathbf{x}_i} \mathbf{x}_j, \tag{10}$$

where k is the number of vertices of Ω_i. It is low-cost and works in some heuristic way since it is not directly related to most geometrical mesh qualities. Later we will derive Laplacian smoother by minimizing our error-based mesh quality.

An optimization-based smoothing has been proposed in [11, 12, 13]. An objected function $\phi(\mathbf{x})$ is composed by combining the element qualities in the patch. A typical choice [13] is $\phi(\mathbf{x}) = \min_{1 \leq j \leq k} q_j(\mathbf{x})$, where $q_j(\mathbf{x})$ is the quality for simplex $\tau_j \in \Omega_i$. Then one uses the steepest descent optimization or GLP (generalized linear program) [31] to find the optimal point x^*.

Optimization-based smoother

$$\mathbf{x}^* = \text{argmax}_{\mathbf{x}\in\Omega_i}\phi(\mathbf{x}). \quad (11)$$

The domain of $\phi(\mathbf{x})$ is restricted to the feasible region A, which is the biggest convex set contained in Ω_i such that $\mathbf{x} \in A$ will not result in overlapping simplexes; see Fig. 2. The optimization-based smoother is designed to improve the mesh quality and the theoretical results developed for GLP ensure that the expected time for one sweep is a linear function of the problem size [31]. But it is often expensive than Laplacian smoothing. Numerical comparison can be found at [32]. It is worthy noting that in two dimensions Zhou and Shimada [33] proposed an angle-based approach mesh smoothing that strikes a balance between geometric mesh quality and computational cost.

All the mesh smoothing schemes we discussed above are designed for isotropic mesh adaptation. For anisotropic mesh smoothing, the first step is to update our understanding of mesh quality which we have done in Section 2. We shall develop several mesh smoothers by minimizing the error-based or the metric-based mesh quality locally, which will be a unified way to derive isotropic and anisotropic mesh smoothers.

We first consider the isotropic case $Q(\Omega_i, \|\mathbf{x}\|^2, 1)$ or $Q(\Omega_i, E, 1)$, where E is the identity matrix representing the Euclidean metric. We replace the vertex $\mathbf{x}_i$ by any $\mathbf{x} \in \Omega_i$, keeping the connectivity, and try to minimize the error locally as a function of $\mathbf{x}$.

By Theorem 1, we consider the following local optimization problem

$$\min_{\mathbf{y}\in A}\int_{\Omega_i}\|\mathbf{x}-\mathbf{y}\|^2 d\mathbf{x}.$$

By the discussion of the CVT, we know that the minimizer is the centroid of Ω_i, namely $\mathbf{x}^* = \int_{\Omega_i}\mathbf{x}d\mathbf{x}/|\Omega_i|$. Thus we get the following smoother.

CVT smoother I

$$\mathbf{x}^* = \frac{\sum_{\tau\in\Omega_i}\mathbf{x}_\tau|\tau|}{|\Omega_i|}, \quad (12)$$

where $\mathbf{x}_\tau$ is the centroid of τ, i.e. $\mathbf{x}_\tau = \sum_{\mathbf{x}_k\in\tau}\mathbf{x}_k/(n+1)$.

If the mesh density is nonuniform, for example, the mesh around the transition layer will quickly change from a small size to a much larger size. In order to keep the smoother from stretching the elements in the high density region out into the low density region, we have to incorporate the mesh density function into our mesh quality. Since we still need the isotropic mesh, we choose the metric $\rho(\mathbf{x})E$. The nonuniform function $\rho(\mathbf{x})$ is to control the mesh density, which aims to equidistribute the error or the volume of element under this metric, while the matrix E is to improve the shape regularity of elements.

Let us consider the following optimization problem:

$$\min_{\mathbf{y}\in A}\int_{\Omega_i}\|\mathbf{x}-\mathbf{y}\|^2\rho(\mathbf{x})d\mathbf{x}.$$

Again the minimizer is the centroid of Ω_i with respect to the density $\rho(\mathbf{x})$, namely

$$\mathbf{x}^* = \frac{\int_{\Omega_i}\mathbf{x}\rho(\mathbf{x})d\mathbf{x}}{\int_{\Omega_i}\rho(\mathbf{x})d\mathbf{x}}.$$

We use ρ_τ, the average of ρ over a simplex, to get our second mesh smoother.

CVT smoother II

$$\mathbf{x}^* = \frac{\sum_{\tau\in\Omega_i}\mathbf{x}_\tau\rho_\tau|\tau|}{\sum_{\tau\in\Omega_i}\rho_\tau|\tau|}. \quad (13)$$

What is the right choice of the density function ρ_τ? It could be *a priori* one. Namely the density is given by the user according to *a priori* information about the function. In practice, especially when solving partial differential equations, the density is given by *a posteriori* error estimate, which of course depends on the function and the problem.

A universal choice of the density function is related to the volume of the element. Recall that the mesh smoothing mainly takes care of the isotropic property of the mesh. It is reasonable to assume that after refinement and coarsening the mesh density is almost equidistributed. Namely the volumes of elements are almost equal under the metric $\rho_\tau E$. Since $|\tau|_{\rho_\tau E} = \rho_\tau^{n/2}|\tau|$, we may choose $\rho_\tau = |\tau|^{-n/2}$. With this choice, the mesh smoother (13) becomes

$$\mathbf{x}^* = \frac{\sum_{\tau\in\Omega_i}\mathbf{x}_\tau|\tau|^{1-n/2}}{\sum_{\tau\in\Omega_i}|\tau|^{1-n/2}}. \quad (14)$$

When $n = 2$, the formula (14) is

$$\mathbf{x}^* = \frac{2}{3}\frac{\sum\mathbf{x}_j}{k} + \frac{1}{3}\mathbf{x}_i.$$

It is a lumped Laplacian smoothing. This relation shows that why Laplacian works in some sense. In three dimensions, no such a relation exists since for a vertex $\mathbf{x}_j$ of Ω_i, the number of simplexes which containing $\mathbf{x}_j$ in Ω_i is not fixed.(In two dimensions, this number is two.)

Since $\cup_i\Omega_i$ is an overlapping decomposition of Ω, the change of Ω_i will affect other patches and thus the overall error will not necessarily be reduced. We shall make use of the formula of $Q(\mathcal{T}, f, 1)$ in Theorem 2 to minimize the interpolation error directly.

By Theorem 2,

$$\begin{aligned} Q(\Omega_i, f, 1) &= \frac{1}{n+1}\sum_{\tau_j\in\Omega_i}\Big(|\tau_j(\mathbf{x})|\sum_{\mathbf{x}_k\in\tau_j,\mathbf{x}_k\neq\mathbf{x}} f(\mathbf{x}_k)\Big) \\ &+ \frac{|\Omega_i|}{n+1}f(\mathbf{x}) - \int_{\Omega_i} f(\mathbf{x})d\mathbf{x}. \end{aligned}$$

Since we only adjust the location of $\mathbf{x}_i$, Ω_i is fixed and $\int_{\Omega_i} f(\mathbf{x})d\mathbf{x}$ is a constant. We only need to minimize $E(\mathbf{x})$

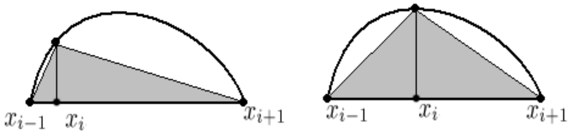


Figure 3: Moving a grid point in its local patch

which is defined by the following expression

$$\sum_{\tau_j \in \Omega_i} \Big(|\tau_j(\mathbf{x})| \sum_{\mathbf{x}_k \in \tau_j, \mathbf{x}_k \neq \mathbf{x}} f(\mathbf{x}_k) \Big) + \frac{|\Omega_i|}{n+1} f(\mathbf{x}).$$

The domain of $E(\mathbf{x})$ is the feasible region A. Since there exists a small neighborhood of $\mathbf{x}_i$ in A, A is not empty and $\mathbf{x}_i$ is an interior point of A. If the triangulation is already optimal , we conclude that $\mathbf{x}_i$ is a critical point of $E(\mathbf{x})$. We then have the following theorem.

Theorem 4. If the triangulation $\mathcal{T}$ is optimal in the sense of minimizing $Q(\mathcal{T}, f, 1)$ for a convex function $f \in \mathcal{C}^1(\Omega)$, then for an interior vertex $\mathbf{x}_i$, we have

$$\nabla f(\mathbf{x}_i) = -\frac{1}{|\Omega_i|} \sum_{\tau_j \in \Omega_i} \Big(\nabla |\tau_j|(\mathbf{x}) \sum_{\mathbf{x}_k \in \tau_j, \mathbf{x}_k \neq \mathbf{x}_i} f(\mathbf{x}_k) \Big).$$

When $n = 1$, Theorem 4 says that if the grid optimize the interpolation error in L^1 norm, it should satisfy

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}}. \tag{15}$$

We use Figure 3 to illustrate (15). We move the grid point x_i in its local patch $[x_{i-1}, x_{i+1}]$. It is easy to see that minimizing $Q(\Omega_i, f, 1)$ is equivalent to maximize the area of the shadowed triangle. Since the base edge is fixed, it is equivalent to maximizing the height. Thus (15) holds.

In two dimensions, since

$$|\tau_j|(x, y) = \begin{vmatrix} x_{j+1} - x_j & x - x_j \\ y_{j+1} - y_j & y - y_j \end{vmatrix},$$

we can get a similar formula

$$\begin{aligned} f_x(x_i, y_i) &= \sum_j \omega_j^x f(x_j, y_j), \\ f_y(x_i, y_i) &= \sum_j \omega_j^y f(x_j, y_j), \end{aligned}$$

where

$$\omega_j^x = \frac{y_{j+1} - y_{j-1}}{|\Omega_i|}, \text{ and } \omega_j^y = \frac{x_{j-1} - x_{j+1}}{|\Omega_i|}.$$

The significance of Theorem 4 is that we can recover the derivative exactly from the nodal values of the function if the triangulation is optimized. With the gradient information, we can approximate f by higher degree polynomials or construct *a posteriori* error indicator.

If the triangulation is not optimized, Theorem 4 can be used to solve the critical point. And the critical point can be used as the new location for the mesh smoother. When $f(\mathbf{x}) = \mathbf{x}^T H \mathbf{x}$ is a non-degenerate quadratic function, i.e. H is a $n \times n$ nonsingular matrix. We can solve the critical point exactly and get a mesh smoother based on ODTs.

ODT smoother I

$$\mathbf{x}^* = -\frac{H^{-1}}{|\Omega_i|} \sum_{\tau_j \in \Omega_i} \Big(\nabla |\tau_j(\mathbf{x})| \sum_{\mathbf{x}_k \in \tau_j, \mathbf{x}_k \neq \mathbf{x}_i} \|\mathbf{x}_k\|_H^2 \Big). \tag{16}$$

When the goal of the mesh adaptation is to get a uniform and shape regular mesh, we choose $f(\mathbf{x}) = \|\mathbf{x}\|^2$ and get

$$\mathbf{x}^* = -\frac{1}{2|\Omega_i|} \sum_{\tau_j \in \Omega_i} \Big(\nabla |\tau_j(\mathbf{x})| \sum_{\mathbf{x}_k \in \tau_j, \mathbf{x}_k \neq \mathbf{x}_i} \|\mathbf{x}_k\|^2 \Big). \tag{17}$$

Comparing with the CVT, Theorem 4 says that for an ODT, the node $\mathbf{x}_i$ is also a kind of center of its local patch. In general, it is not the centroid of the patch. This is the difference of the ODT smoother with the CVT smoothers including Laplacian smoother. For example, if vertices of the patch lie on a common sphere, then the optimal location is the sphere center not the centroid. In deed, since the approximation error only depends on the second derivative,

$$Q(\mathcal{T}, \|\mathbf{x}\|^2, p) = Q(\mathcal{T}, \|\mathbf{x} - \mathbf{x}_o\|^2, p).$$

For function $f(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_o\|^2$, $f_I(\mathbf{x}) = R^2$ and

$$(f_I - f)(\mathbf{x}) = R^2 - \|\mathbf{x} - \mathbf{x}_o\|^2$$

attains the minimum value at $\mathbf{x} = \mathbf{x}_o$. As a byproduct, (17) gives a simple formula to compute the circumcenter of a simplex, which is not easy in high dimensions.

When f is a convex quadratic function, the optimization of interpolation error is a quadratic optimization. After we get the global critical point x^*, we can further simplify our optimization problem to be

$$\min_{\mathbf{x} \in A} \|\mathbf{x} - \mathbf{x}^*\|^2_{\nabla^2 f}. \tag{18}$$

The problem (18) is to find the projection (under the metric $\nabla^2 f$) of $\mathbf{x}^*$ to the convex set A. For the efficiency of algorithm, we only compute the projection when the global minimum point x^* is not acceptable. The cost of this algorithm is a little bit higher if we need to compute the projection and change the topological structure of the mesh. But the overall cost for one sweep will not increase too much since it operates like a smart-Laplacian smoothing [12].

It may happen that the new location $\mathbf{x}^*$ is on the boundary of the patch; see Figure 4. For the sake of conformity we need to connect this hanging point to the related points which will reduced the error since $\|\mathbf{x}\|^2$ is convex. For two dimensional triangulations, it looks like we perform an edge swapping after a local smoothing. If the point $\mathbf{x}^*$ is on the boundary

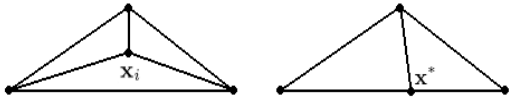


Figure 4: Moving a point to the element's boundary

of Ω, we will eliminate an element by moving an interior point to the boundary. Conversely a point on the boundary can be moved into the interior. Some boundary points, which are called corner points, are fixed to preserve the geometric shape of the domain. But we free other boundary points. This freedom can change the density of points near the boundary and yield a better mesh since the interpolation error is reduced after each local adjustment.

For a general function f, we can use line search to solve the following optimization problem.

ODT smoother II

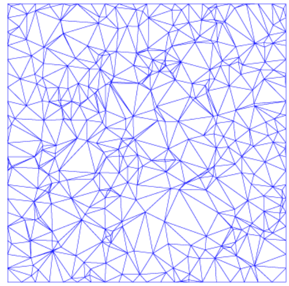
$$x^* = \text{argmin}_{\mathbf{x} \in A} E(\mathbf{x}). \qquad (19)$$

An alternative approach to solve (19) approximately is to compute an average Hessian matrix H_{Ω_i} in the local patch, and using ODT smoother I for the quadratic function $f_q(\mathbf{x}) := \mathbf{x}^T H_{\Omega_i} \mathbf{x}$. This approach is successfully applied in the construction of optimal meshes in [4]. We will include several pictures in the next section. On those optimal meshes, the interpolation error attains the optimal convergence rate; see in [4] for details.

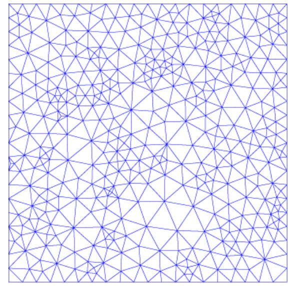
5. NUMERICAL EXPERIMENTS

In this section, we shall present several examples to show the efficiency of our new smoothers in the isotropic grid adaptation as well as the anisotropic case.

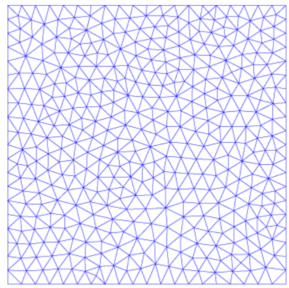
The first example is to compare our new smoothers with Laplacian smoother for the isotropic grid adaptation and to show the reduction of the interpolation error for those smoothers. We place 20 equally spaced nodes on each edge of the boundary of square $[0,1] \times [0,1]$ and 361 nodes in the square. The nodes in the domain are placed randomly while the nodes on the boundary is equally spaced since in this example we only move the interior nodes. We use 'delaunay' command of the Matlab 6.1 to generate the original mesh; see Fig 5(a). In this example, the goal of the mesh smoothing is to get an equilateral mesh. Namely triangles are almost equilateral and the density is uniform. We implemented Laplacian smoothing, CVT smoothing I and ODT smoothing I. In one iteration we apply the mesh smoothing for each node and then do the edge swapping once. We incorporate the edge swapping in our mesh smoothing since it can change the topological structure of the mesh. In practice, the edge swapping always come with the mesh smoothing. We perform 10 iterations and present meshes obtained by different smoothers in Figure 5. According to our theory, Laplaican smoothing is not for the uniform density. Figure 5(b) shows that the triangle size is not uniform. We also test CVT smoother I and ODT smoother I which are designed for the uniform density. Both of them get better meshes than Laplacian smoothing; see Figure 5(c) and 5(d).



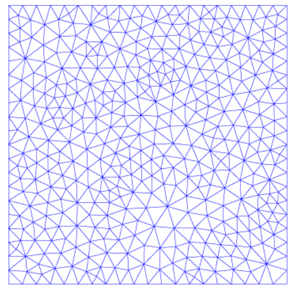
(a) Original mesh



(b) Laplacian smoother



(c) ODT smoother I



(d) CVT smoother I

Figure 5: Comparison of Laplacian smoother, CVT smoother I and ODT smoother I

In Figure 6, we plot the interpolation error of each mesh smoother. In this example, $f(\mathbf{x}) = \|\mathbf{x}\|^2$. Therefore we only need to compare $\int_\Omega f_I(\mathbf{x}) d\mathbf{x}$ which can be evaluated exactly. See the proof of Theorem 2. The initial interpolation error is plotted in the location 'step 1'. Figure 6 clearly shows the reduction of the interpolation after each iteration. The ODT smoother I is better than the others since it has a provably error reduction property. The numeric convergence of the interpolation errors for those smoothers is very clear from this picture.

The computational cost of those smoothing schemes in each iteration is listed in the Table 1. In order to compare the efficiency of the smoothing schemes, we do not include the computational time for the edge swapping in each iteration. Table 1 clearly shows that all of those three mesh smoothing schemes have similar computational cost. Thus it is fair to say that ODT smoother I is very desirable for isotropic and uniform mesh generation and adaptation.

Our second example is to use ODT smoothers to generate

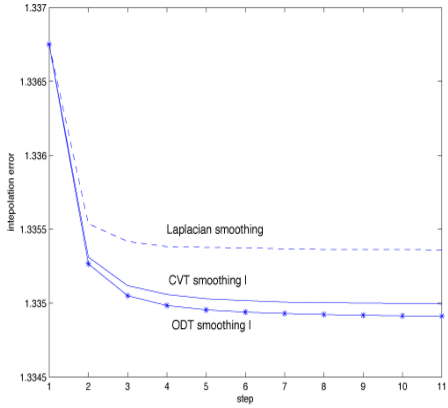


Figure 6: Error comparison of Laplacian smoother, CVT smoother I and ODT smoother I

Step	Laplacian	CVT I	ODT I
1	0.19	0.18	0.20
2	0.15	0.16	0.16
3	0.15	0.16	0.16
4	0.15	0.16	0.15
5	0.15	0.16	0.17
6	0.15	0.15	0.16
7	0.15	0.15	0.16
8	0.16	0.16	0.15
9	0.15	0.17	0.16
10	0.13	0.16	0.17

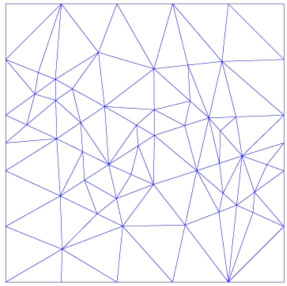
Table 1: Computational cost comparison of Laplacian smoother, CVT smoother I and ODT smoother I

an anisotropic mesh. We set $f(x, y) = 10x^2 + y^2$ to be an anisotropic function. The optimal mesh under the Hessian matrix of f should be long and thin vertically. We also include the edge swapping. In Figure 7 we list several meshes after different iterations. Since the desirable mesh is anisotropic, the number of boundary points on the vertical edges should be much less than that of points on the horizontal edges. Therefore we free the boundary points except four corner points. From those pictures, it is clear that some points are projected to the boundary and some are moved into the square. We also plot the interpolation error in the Figure 8. Since the local mesh smoothing is a Gauss-Seidel like algorithm, we see the Gauss-Seidel type convergence result for those mesh smoothing schemes; see Figure 6 and 8. An ongoing project is to develop a multigrid-like mesh smoothing schemes. It is essentially a multilevel constraint nonlinear optimization problem which is well studied in the literature(see, for example, Tai and Xu [34]).

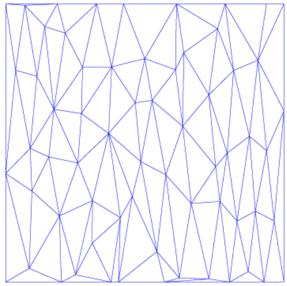
The third example is to show a successful application of the ODT smoother II in the anisotropic mesh generation. The function we approximate is

$$f(x, y) = e^{-(\frac{r-0.5}{\epsilon})^2} + 0.5r^2$$

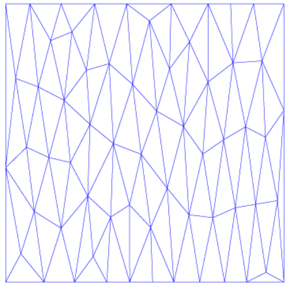
where $r^2 = (x+0.1)^2 + (y+0.1)^2$ and $\epsilon = 10^{-3}$. This function changes dramatically at the ϵ neighborhood of $r = 0.5$. We use offset $(x + 0.1, y + 0.1)$ to avoid the non-smoothing Hessian matrix at $(0, 0)$ and quadratic function $0.5r^2$ to ensure that Hessian matrix is not singular when r is far away from the circle so that we can focus our attention on the interior layer. We use our local refinement, edge swapping and ODT smoother II to improve the mesh. Here we present several pictures of our meshes. For the optimality of the L^p norm of the interpolation error, see [4] for details. We have applied the mesh adaptation strategies based on ODTs in solving partial differential equations, especially for the anisotropic problems; see our recent work [9].



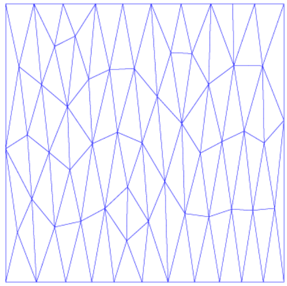
(a) Original mesh



(b) Mesh after 1 iteration



(c) Mesh after 5 iterations



(d) Mesh after 20 iterations

Figure 7: Anisotropic meshes obtained by ODT smoother I

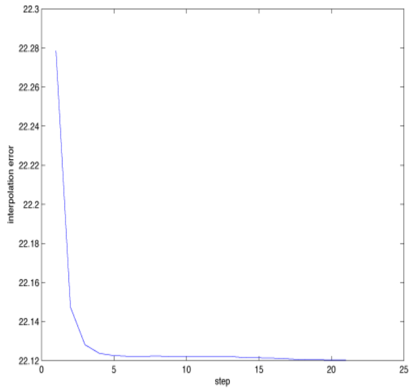
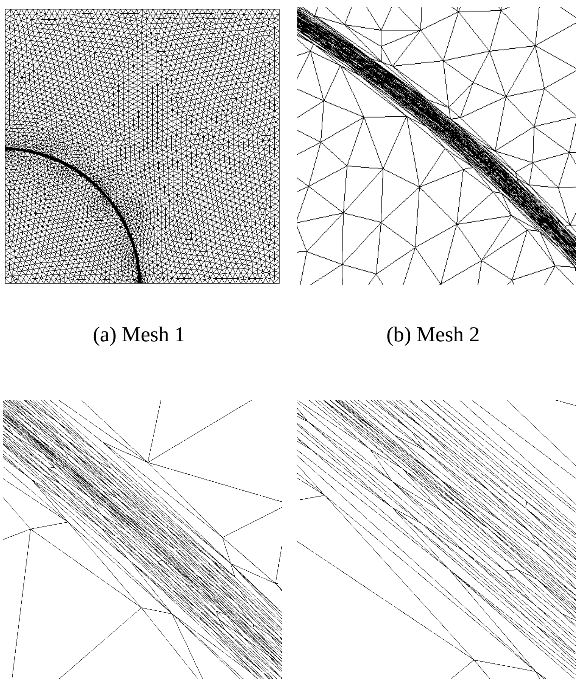


Figure 8: Interpolation of the second example



(a) Mesh 1

(b) Mesh 2

(c) Mesh 3

(d) Mesh 4

Figure 9: An anisotropic mesh and its details

6. CONCLUDING REMARKS AND FUTURE WORK

In this paper, we have developed several mesh smoothing schemes using optimal Delaunay triangulations as a framework. The proposed mesh smoothers are designed to reduce the interpolation error. Our error estimates of the interpolation error ensures that the optimization of the interpolation error aims to equidistribute the edge length under some metric related the Hessian matrix of the approximated function. Since the L^p norm ($p < \infty$) is somehow an average norm, we can not promise that the reduction of the interpolation error will improve the geometric qualities, for example the minimal angle. However, from the function approximation point of view, the minimal angle condition is not necessary if we measure the interpolation error in L^p norm.

We presented two formulations of the interpolation error. The identity (5) in Theorem 1 seems to be new in the literature and shows the close relation to the functional used in the centroid Voronoi tessellations. We also presented a conjecture about the duality between the ODT and the CVT .

The mesh smoothing schemes proposed in this paper have a strong mathematical background. In the isotropic case, the error-based mesh quality is guaranteed to be improved while the computational cost is as low as that of Laplacian smoothing. Laplacian smoothing can be mathematically justified under this framework. Another advantage of our approach is the unification of the isotropic and anisotropic mesh adaptations. By choosing anisotropic function or metric, our smoothing schemes can be used to generate or improve the anisotropic mesh.

Although the formulation of our mesh smoothing schemes hold in any spatial dimension, the numerical experiments, so far, are restricted in two dimensional triangulations. The three dimensional case will be investigated later.

Acknowledgments

The author is grateful to Professor Jinchao Xu for numerous discussions and kind edit of the English and to the referee for many helpful suggestions.

References

[1] Chen L., Xu J. "Optimal Delaunay triangulation." *Journal of Computational Mathematics*, vol. 22(2), 299–308, 2004

[2] Babuška I., Aziz A.K. "On the angle condition in the finite element method." *SIAM J. Numer. Anal.*, vol. 13, no. 2, 214–226, 1976

[3] Rippa S. "Long and thin triangles can be good for linear interpolation." *SIAM J. Numer. Anal.*, vol. 29, 257–270, 1992

[4] Chen L., Sun P., Xu J. "Optimal anisotropic simplicial meshes for minimizing interpolation errors in L^p-norm." *Submitted to Math. Comp.*, 2003

[5] Nadler E. "Piecewise linear best L_2 approximation on triangulations." C.K. Chui, L.L. Schumaker, J.D. Ward, editors, *Approximation Theory*, vol. V, pp. 499–502. Academic Press, 1986

[6] D'Azevedo E., Simpson R. "On optimal interpolation triangle incidences." *SIAM J. Sci. Statist. Comput.*, vol. 6, 1063–1075, 1989

[7] Huang W. "Variational mesh adaptation: isotropy and equidistribution." *J. Comput. Phys.*, vol. 174, 903–924, 2001

[8] Huang W., Sun W. "Variational mesh adaptation II: Error estimates and monitor functions." *J. Comput. Phys.*, vol. 184, 619–648, 2003

[9] Chen L., Sun P., Xu J. "Multilevel Homotopic Adaptive Finite Element Methods for Convection Dominated Problems." *The Proceedings for 15th Conferences for Domain Decomposition Methods*. 2004

[10] Field D. "Laplacian Smoothing and Delaunay Triangulation." *Communications in Applied Numerical Methods*, vol. 4, 709–712, 1988

[11] Bank R., Smith R.K. "Mesh Smoothing Using A Posteriori Error Estimates." *SIAM J. Numer. Anal.*, vol. 34, 979–997, 1997

[12] Shephard M., Georges M. "Automatic three-dimensional mesh generation by the finite octree technique." *Internat. J. Numeri. Methods Engrg.*, vol. 32, 709–749, 1991

[13] L.A. Freitag M.J., Plassmann P. "An efficient parallel algorithm for mesh smoothing." *4th International Meshing Roundtable*, pp. 47–58. Sandia National Laboratories, 1995

[14] Habashi W.G., Fortin M., Dompierre J., Vallet M.G., Ait-Ali-Yahia D., Bourgault Y., Robichaud M.P., Tam A., Boivin. S. *Anisotropic mesh optimization for structured and unstructured meshes*. In 28th Computational Fluid Dynamics Lecture Series. von Karman Institute, March 1997

[15] F.Aurenhammer, R.Klein. *Handbook of Computational Geometry*. Amsterdam, Netherlands: North-Holland, 2000

[16] Lawson C. "Software for C^1 surface interpolation." *Mathematical Software III*, pp. 161–194 J.R. Rice, ed., Academic Press, 1977

[17] Rajan V. "Optimality of the Delaunay triangulation in R^d." *Proc. of the Seventh Annual Symp. on Comp. Geom*, pp. 357–363, 1991

[18] Berzins M. "A Solution-Based Triangular and Tetrahedral Mesh Quality Indicator." *SIAM J. Scientific Computing*, vol. 19, 2051–2060, 1998

[19] Shewchuk J. "What is a Good Linear element? Interpolation, Conditioning, and Quality measures." *11th International Meshing Roundtable*, pp. 115–126. Sandia National Laboratories, 2002

[20] Chen L. "New Analysis of the Sphere Covering Problems and Optimal Polytope Approximation of Convex Bodies." *Submitted to J. Approx.*, 2004

[21] H.Edelsbrunner, R.Seidel. "Voronoi diagrams and arrangements." *Disc. and Comp. Geom.*, vol. 8, no. 1, 25–44, 1986

[22] Gruber P. "Aspects of approximation of convex bodies." G. PM, W. JM, editors, *Handbook of Convex Geometry.*, vol. A, pp. 319–345. Amsterdam: North-Holland, 1993

[23] Heckbert P., Garland M. "Optimal Triangulation and Quadric-Based Surface Simplification." *Journal of Computational Geometry: Theory and Applications*, 1999

[24] Fortune S. "Voronoi Diagrams and Delaunay Triangulations." *Computing in Euclidean Geometry, Edited by Ding-Zhu Du and Frank Hwang*. World Scientific, Lecture Notes Series on Computing – Vol. 1, 1992

[25] Graf S., Luschgy H. *Foundations of quantization for probability distributions*, vol. LMN 1730 of *Lecture Notes in Mathematics*. Springer Verlag, Berlin, Heidelberg, 2000

[26] Du Q., Faber V., Gunzburger M. "Centroidal Voronoi Tessellations: Applications and Algorithms." *SIAM Review*, vol. 41(4), 637–676, 1999

[27] Du Q., Gunzburger M. "Grid generation and optimization based on centroidal Voronoi tessellations." *Appl. Math. Comp.*, vol. 133, 591–607, 2002

[28] Du Q., Gunzburger M., Ju L. "Constrained CVTs in general surfaces." *SIAM J. Scientific Computing*, vol. 24, 1488–1506, 2003

[29] Du Q., Wang D. "Anisotropic Centroidal Voronoi Tessellations and Their Applications." *SIAM J. Scientific Computing*, vol. Accepted, 2003

[30] Du Q., Wang D. "Tetradehral mesh optimization based on CVT." *Inter. J. Numer. Meth. Eng.*, vol. 56, no. 9, 1355–1373, 2002

[31] N.Amenta, Bern M., Eppstein D. "Optimal Point Placement for Mesh Smoothing." *Journal of Algorithms*, pp. 302–322, February 1999

[32] Freitag L. "On combining Laplacian and optimization-based mesh smoothing techniques." *AMD Trends in Unstructured Mesh Generation, ASME*, vol. 220, 37–43, July 1997

[33] Zhou T., Shimada K. "An Angle-Based Approach to Two-Dimensional Mesh Smoothing." *9th International Meshing Roundtable*, pp. 373–384. Sandia National Laboratories, October 2000

[34] Tai X., Xu J. "Global convergence of subspace correction methods for convex optimization problems." *Math. Comp.*, vol. 71, no. 237, 105–124, 2002