Spacing Control and Sliver-free Delaunay Mesh

Xiang-Yang Li *

Abstract:— *We are often required to generate a Delaunay mesh whose element size is within a constant factor of a control spacing function, i.e., well-conformed, in addition to the fact that each mesh element has small aspect ratio, i.e., well-shaped. However, generating well-shaped Delaunay meshes is an open problem for a long time. Observe that slivers have small radius-edge ratio thus the Delaunay triangulation of well-spaced point set can not guarantee a sliver-free mesh.*

In this paper, we present a refinement-based method that, given a PLC domain with no acute input angles, guarantees to generate a well-shaped and well-conformed Delaunay mesh. Specifically, for any tetrahedron τ generated by this algorithm, its radius-edge ratio is at most a small constant $\varrho_0 > 2$, which can be given as an input parameter. Moreover, we show that there is a constant $\sigma_0 > 0$ depending on ϱ_0 such that $V/L^3 \geq \sigma_0$, where V is the volume of τ and L is the shortest edge length of τ. Thus, the algorithm generates a well-shaped Delaunay mesh: the aspect ratio of each tetrahedron is at most a constant depending on ϱ_0. The size of each tetrahedron element is also within a small constant factor of the given control spacing.

Keywords: Mesh generation, mesh quality, Delaunay triangulations, slivers, computational geometry, algorithms.

1 Introduction

This paper presents a quality guaranteed refinement method to generate a well-shaped Delaunay mesh respecting to a control spacing. The input could be a three-dimensional polyhedral domain with no small angles or an *almost good* 3D Delaunay mesh. The well-shaped mesh is measured by the largest *aspect ratio* of all its tetrahedron elements. While the almost good Delaunay mesh is measured by the largest *radius-edge ratio* of all its elements. Notice that a bound on the radius-edge ratio eliminates all tetrahedra with large aspect ratio except slivers. In other words, our algorithm generates a sliver-free Delaunay mesh.

Mesh quality measure. In this paper, we exclusively consider three-dimensional meshes whose elements are tetrahedra. The size and shape of the triangles and tetrahedra is important because it influences the convergence and stability of numerical algorithms such as the finite element method, see Strang and Fix [21]. We assume that the spatial domain is given in terms of its piecewise linear complex boundary (*PLC*) [22]. [1] Moreover, we also assume that there are no small input angles in the domain. This assumption will guarantee that there is no infinite loop during the boundary protection step.

The *aspect ratio* of an element is usually defined as the ratio of the radius of its circumsphere to the radius of its inscribed sphere. The smaller the aspect ratio, the better the tetrahedron is. Unfortunately, there is no method that guarantees to generate a Delaunay mesh whose elements have small aspect ratio for a PLC domain until this work.

An alternative but weaker quality measurement is to use the *radius-edge ratio* introduced by Miller *et al.* [16]. It is the ratio of the circumradius to the shortest edge length of the tetrahedron. The mesh whose elements have small radius-edge ratio is called *almost good* mesh. There are numerous methods [5, 18, 19, 17, 12, 15] that guarantee to generate a mesh with small radius-edge ratio. Here, the radius-edge ratio of a mesh is the maximum radius-edge

*Dept. of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL. Email: xli2@cs.uiuc.edu. And Dept. of Computer Science, Illinois Institute of Technology, Chicago, IL. Email: xli@cs.iit.edu.

[1] PLC defines a domain using a set of polytopes S, such that (1) For each polytope of S, its boundary is a union of polytopes in S. (2) S is closed under intersection. (3) If $dim(P \cap Q) = dim(Q)$, then $P \subset Q$ and $dim(P) < dim(Q)$. In two dimensions, PLC is also called *Planar Straight Line Graph (PSLG)*.

ratio among all of its elements. Similarly, the aspect ratio of a mesh is the maximum aspect ratio among all of its elements.

Not any tetrahedron, which has small radius-edge ratio, has small aspect ratio. Slivers are the only tetrahedra that have small radius-edge ratio but have large aspect ratio. The ubiquity of slivers in 3-dimensional Delaunay triangulations has been recognized at least since the experimental study of Cavendish, Field and Frey [3].

To be consistent with previous work [4, 9], we say a mesh has the Ratio Property $[\varrho_0]$ if all its tetrahedra have radius-edge ratio at most ϱ_0. For later convenience, we will use R_τ, L_τ and $\rho(\tau)$ to denote the circumradius, the shortest edge length and the radius-edge ratio of an element τ.

Generating almost good mesh and spacing control In two dimensions, many methods [1, 5, 6, 12, 15, 18, 22] guarantee to generate well-shaped meshes, i.e., with small radius-edge ratio. Surprisingly, in three dimensions, generating well-shaped meshes is considerably more difficult. The main difficulty comes from how to generate a sliver-free Delaunay mesh. Talmor [22] notices that even well-spaced vertices do not prevent slivers from its Delaunay triangulation. Even so, generating a three-dimensional mesh with small radius-edge ratio is well understood. J. Shewchuk [19] extended the Delaunay refinement method to three dimensions by proving that all tetrahedra will have radius-edge ratio no more than 2. The sphere packing based methods, see Miller *et al.* [17] and Li *et al.* [15], guarantee to generate meshes with small radius-edge ratio and the size of each tetrahedron is within a small constant factor of the element size requirement. However, all these methods fail to address the problem of slivers.

For numerical simulations, we have to generate a well-shaped mesh whose element sizes also conform well to a given control spacing function. A control spacing function specifies the desired element size at each point of the domain. Here the element size is typically measured by the nearest neighbor function or edge length function. Several heuristics and algorithms had been developed to generate well-conformed meshes. Splitting the longest edge, or subdividing the simplexes are among the most used heuristics [2, 8]. Shimada [20] used the particle simulation to find a good mesh vertices set, then constructed the final mesh using Delaunay triangulation of the found vertices. However, above algorithms do not provide any theoretical guarantees on qualities of the generated meshes.

Over the years, the algorithms that provide theoretical quality guarantees had also been developed. These include the sphere packing based algorithm by Miller *et al.* [17], and the biting method by Li *et al.* [15]. Chew [6] also had developed a refinement based algorithm which generates good surface meshes respecting to control spacing function. Li [12] developed a similar algorithm that works for any dimensions also. It systematically defines what is bad element, and inserts the circumcenter of each bad element as new mesh vertex.

Eliminating slivers. Slivers are notoriously common in tetrahedral meshes. The numerical accuracy usually depends on the smallest dihedral angle among all tetrahedra of the mesh. Different numerical methods may have different dependency on the smallest angle, but it is always necessary to generate a mesh whose smallest dihedral angle is not too small. Observe that several algorithms guarantee to generate almost good meshes. Unfortunately, the smallest angle of a sliver could be as small as possible. Therefore, removing slivers from an almost good mesh is one of the major tasks for mesh generation. There had been numerous efforts [7, 4, 9, 11] that try to achieve this either experimentally or theoretically. However, we will focus only on the theoretical approaches, because they have theoretical guarantees to back up the performance. Before this work, there are only a few theoretical approaches due to the hardness of the problem.

Chew [7] sketched an algorithm that eliminates slivers by adding points near the circumcenters of some tetrahedra. The algorithm terminates with a sliver-free mesh except possibly original slivers in the mesh. Notice that this algorithm results in a constant density mesh: every tetrahedron has circumradius no larger than one unit. In addition, his algorithm does not address the slivers near boundary completely.

Recently, Cheng, Dey, Edelsbrunner, Facello, and Teng [4] developed an algorithm that, given an almost good Delaunay triangulation, constructs an assignment of weights to mesh vertices so the weighted Delaunay triangulation has small aspect ratio, i.e., free of slivers. We call it *weighted Delaunay* based method. We refer the reader to [4] for a description of weighted Delaunay triangulations.

However, the algorithm fails to address the boundary situation completely. It is possible that there are some bad elements near the domain boundary, which is unacceptable by several numerical methods. Recently, Edelsbrunner, Li, Miller *et al.* [9] developed a new algorithm that perturbs the vertices of an almost good Delaunay mesh such that the Delaunay triangulation of the perturbed vertices has small aspect ratio. Unfortunately, both algorithms [4, 9] are lack of boundary treatment.

Our results. The main result of this paper is a refinement-based technique that generates a well-shaped Delaunay mesh whose elements size approximately equals the control spacing. It, by refining bad elements, first generates an almost good mesh whose element size conforms to the control spacing. It then removes slivers by adding point around the circumcenter of bad element (with large radius-edge ratio or slivers) such that it avoids creating new *small* slivers at the same time. [2] It keeps adding points until the mesh has small radius-edge ratio and does not have sliver. We prove that for any bad element τ, there is a point p around its circumcenter that the insertion of p will not introduce new *small* sliver. Observe that, the insertion of p will definitely eliminate τ. However, when the to-be-inserted point p encroaches boundary triangles or segments, we split these boundary triangles or segments instead of adding point p. We prove the termination guarantee of our algorithm by showing that the distance between the closest mesh vertices is just decreased by a constant factor compared with that of the input mesh. This method can be combined with previous method such as [4] and [9] to avoid over-refining the mesh.

Our algorithm differs from Chew's algorithm in that it generates a non-uniform mesh and eliminates all original slivers without introducing new slivers in final mesh. Notice that if there are slivers in the original mesh, then Chew's algorithm can not guarantee to remove them even they are not near the domain boundary. Chew claimed that his algorithm works for non-uniform mesh, but, to the best of our knowledge, there are no any proofs available.

[2] Here, a created sliver is small if its circumradius is less than some constant factor of the circumradius of this bad element. We will give exact definition later.

Outline. The remainder of the paper is structured as follows. Section 2 introduces the basic concept such as Delaunay triangulation, sliver, sliver regions and their basic related properties. Our refinement-based algorithm is presented in Section 3. It specifies how to remove slivers or elements with large radius-edge ratio by avoiding creating small slivers. The termination and quality guarantee of the algorithm is presented in Section 4. Section 5 concludes the paper with discussions.

2 Preliminaries

Generating a mesh whose elements have small aspect ratio and their sizes conform to a given control spacing function is one of the most important steps in numerical simulations. The *aspect ratio* of an element is usually defined as the ratio of the radius of its circumsphere to the radius of its inscribed sphere. An alternative but weaker quality measurement is *radius-edge ratio*, which is the ratio of the circumradius to the shortest edge length of the tetrahedron.

2.1 Delaunay Triangulation

Delaunay triangulation is widely used to generate tetrahedral meshes because it provides a bridge to prove the theoretical quality guarantees on several meshing algorithms. There are abundant of well-studied algorithms to construct them [3, 10]. A triangulation of a vertex set in general position is a Delaunay triangulation if the circumsphere of each tetrahedron does not contain any mesh vertices in its interior. For Delaunay triangulations, there are numerous nice properties. For example, after inserting a new vertex p, all new tetrahedra created in the Delaunay triangulation of new vertex set are incident to p, i.e., have p as one of its vertices. Furthermore, the new triangulation can be updated by local operations. The nearest neighbor graph defined by a vertex set is contained in the Delaunay triangulation of the vertex set. In other words, the shortest edge length of the Delaunay triangulation is the closest distance among mesh vertices. This fact is used in proving the termination guarantee of our algorithm. The Delaunay triangulation maximizes the radius-edge ratio for two-dimensional vertex set, but not always for three-dimensional vertex set.

2.2 Spacing Function

A spacing function $f()$ is used to specify the ideal element size at every point of the domain Ω. Given an input domain Ω, the geometry structure of the domain boundary contributes to the ideal spacing specification of a well-shaped mesh that could be generated on Ω. Ruppert [18] introduced the concept called *local feature size* to capture the geometry features. He has observed that $lfs()$ changes slowly within the domain. Formally, a function $f()$ is α-*Lipschitz* if for any two points x, y in the domain,

$$|f(x) - f(y)| \leq \alpha||x - y||.$$

Then the Lipschitz coefficiency of $lfs()$ is bounded from above by 1 [18]. In addition, the numerical error estimation also determines the size of the mesh elements. Therefore, the spacing function is the combination of the geometry condition and the numerical condition.

On the other hand, we can deduce the spacing function defined by a well-shaped mesh M over a domain Ω. There are several ways to do so. Edge length function $El()$ and nearest neighbor function $N()$ [12, 22] are two of the most used ones. Here $El(x)$ is the length of the longest edge of all elements containing point x; $N(x)$ is the distance from x to the second closest mesh vertex. Therefore, a mesh conforms well to a given control spacing $f()$ if $N()$ (or $EL()$) is within a small constant factor of $f()$.

It is simple to show that we can not generate well-shaped and well-conformed mesh for an arbitrary spacing function. Specifically, we require that the spacing function to have an α-Lipschitz condition for a small constant α.

2.3 Parameterizing Slivers

A tetrahedron has small aspect ratio implies that it has small radius-edge ratio, but not vice versa. Sliver is the tetrahedron that has small radius-edge ratio, but the aspect ratio could be as large as possible. Let's study in detail how to define sliver by considering a tetrahedron $pqrs$. Let$V$ be its volume and L be its shortest edge length. As [4], we define $\sigma = \sigma(pqrs) = V/L^3$ as a measure of its quality. Call tetrahedron $pqrs$ a *sliver* if $\rho(pqrs) \leq \varrho_0$ and $\sigma(pqrs) < \sigma_0$, where $\sigma_0 > 0$ is a constant that we specify later. It is useful to relate this measure with a distance-radius ratio defined in [4, 9]. Let D be the Euclidean distance of point p from the plane passing through qrs and let Y be the radius of the circumcircle of qrs, see Figure 1. We call triangle qrs the *base-triangle*. The following lemma shows that the value D/Y is no more than a constant factor of σ for any tetrahedron.

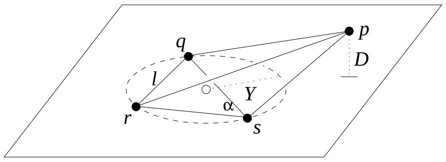

Figure 1. D/Y depends on the ordering of the four vertices, but all four such values are no more than a constant factor of σ for any tetrahedron $pqrs$.

Lemma 2.1 [DISTANCE-RADIUS QUALITY] *For any tetrahedron $pqrs$, $D \leq 12\sigma Y$, where $\sigma = \frac{V}{L^3}$.*

PROOF. Let l be the shortest edge length of triangle qrs, S be the area of triangle qrs. Then $S \geq l^2 \sin(\alpha)/2)$, where α is the smallest angle of qrs, i.e., $\sin(\alpha) = \frac{l}{2Y}$. It follows that $S \geq l^3/(4Y)$. Notice that $V = S \cdot D/3 = \sigma L^3$. It implies that

$$\begin{aligned} D &= 3\sigma L^3/S \\ &\leq \frac{3\sigma L^3}{l^3/(4Y)} \\ &\leq 12\sigma Y, \end{aligned}$$

because $L \leq l$. □

Observe that the above lemma is true for any tetrahedron and does not depends on which vertex is ordered first. Notice that, it is also proved in [4] that $D \geq \frac{3}{\pi\varrho_0^3}\sigma Y$, if tetrahedron has Ratio Property $[\varrho_0]$. For convenience, we will use $D \leq c_1 Y$, where $c_1 = 12\sigma$. The following lemma will verify our definition of $\sigma(\tau)$ for tetrahedron τ.

Lemma 2.2 [ASPECT RATIO, DISTANCE-RADIUS] *[14] For any tetrahedron τ, if $\sigma(\tau) \geq \sigma_0$ and $\rho(\tau) \leq \varrho_0$, then the aspect ratio of τ is at most $\frac{\sqrt{3}\varrho_0^3}{\sigma_0}$.*

PROOF. Let R_τ, r_τ be the circumradius and inradius of τ. Let S_i, $i = 1, 2, 3, 4$ be the area of four face triangles. Then we have $S_i \leq \frac{3\sqrt{3}}{4}R_\tau^2$. Then

$$V = \sum_{i=1}^{4} \frac{1}{3} S_i \cdot r_\tau \leq \sqrt{3} R_\tau^2 r_\tau$$

And we also have

$$V \geq \sigma_0 L_\tau^3 \geq \sigma_0 (\frac{R_\tau}{\varrho_0})^3$$

The lemma follows from $\sqrt{3}R_\tau^2 r_\tau \geq \sigma_0(\frac{R_\tau}{\varrho_0})^3$. $\boxminus$

In the rest of paper, we will use (c, R) to denote a sphere centered at point c with radius R.

2.4 Picking Region

The refinement algorithm is based on the following observation. Any tetrahedron τ will not be in the Delaunay triangulation if we add any vertex that is inside the circumsphere of τ. However, to avoid creating a very short edge in new triangulation, we only pick a point from sphere $(c_\tau, \delta R_\tau)$, where $\delta < 1$ is a constant to be specified later in 4.2.3. For convenience, we will always use c_τ, and R_τ to denote the circumcenter and circumradius of τ. We call $(c_\tau, \delta R_\tau)$ the *picking region* for tetrahedron τ.

When c_τ is close to domain boundary, the distance of points from $(c_\tau, \delta R_\tau)$ to the domain boundary can be very small. Or to be worse, the sphere $(c_\tau, \delta R_\tau)$ could be outside the domain thus we can not select any point. Therefore, we protect the domain boundary by splitting some boundary triangles or segments instead of selecting points from $(c_\tau, \delta R_\tau)$.

The basic scheme of boundary protection is same as the classic Delaunay refinement methods. For a triangle qrs, its *equatorial sphere* is the smallest sphere containing points q, r, s. For a segment qr, its *diametric sphere* is the smallest sphere containing points q, r. So qr is a diameter of its diametric sphere. Without confusion, sometimes we will just use the circumsphere to denote the smallest sphere containing an element (triangle or segment). A point c encroaches boundary if it is contained in the circumsphere of any boundary triangle or segment. Let (c_τ, R_τ) be the circumsphere of tetrahedron τ. Assume c_τ encroaches the circumsphere (c_μ, R_μ) of boundary triangle or segment μ. By using triangular inequality, it can be proved [19] that $R_\mu \geq R_\tau/2$ if the element μ contains the projection of c_τ on μ.

For a boundary triangle qrs, let (c_{qrs}, R_{qrs}) be its circumcircle. We call circle $(c_{qrs}, \delta R_{qrs})$ the *picking region* of triangle qrs. An analog definition for boundary segment is defined as follows. Let qr be a boundary segment and c be its middle point. Then

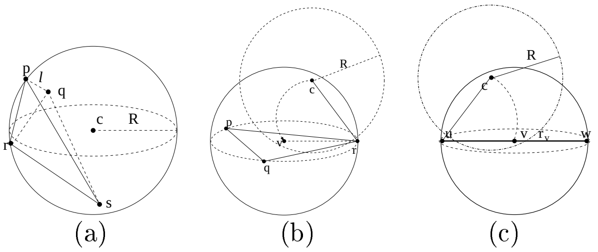

Figure 2. Circumcenter c does not encroach any boundary; c encroaches a boundary triangle; c encroaches a boundary segment.

the segment on qr centered at c with length $2\delta||qr||$ is called the *picking region* of qr. Without confusion, we will use $(c_\tau, \delta R_\tau)$ to denote the picking region of element τ. Here R_τ is the circumradius of τ; τ can be a tetrahedron, triangle or segment. The picking region of τ has the same dimension as τ.

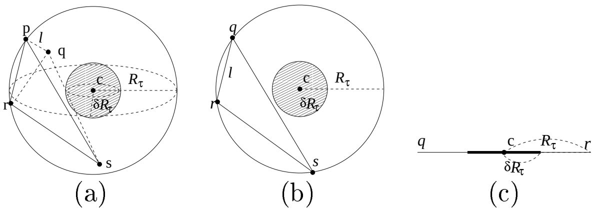

Figure 3. The picking region of a tetrahedron, a boundary triangle and a boundary segment.

Notice that, as classic Delaunay refinement, we require that the circumspheres of all tetrahedra, boundary triangles and segments do not contain any vertex inside. In other words, if any boundary triangle μ is encroached by the circumcenter of a bad tetrahedron τ, then we select a point p from the picking region of μ instead of from that of τ. We protect the boundary segment similarly. Observe that we can not select an arbitrary point inside the picking region to refine the mesh. We will study in detail how to select point inside the picking region.

2.5 Sliver Regions

We then begin studying conditions under which a point p creates sliver $pqrs$ with base-triangle qrs based on above Lemma 2.1. This had been done in [9], but for completeness, we still include it here.

Let (z, Z) be the circumsphere of a tetrahedron $pqrs$ and let (y, Y) be the circumcircle of the triangle qrs. If $pqrs$ has the Ratio Property $[\varrho_0]$ then $Y \leq Z \leq \sqrt{3}\varrho_0 Y$.

Recall that, in the current context, we identify tetrahedron $pqrs$ as a *sliver* if $\sigma(pqrs) = \frac{V}{L^3}$ is less than some threshold σ_0 and it has Ratio Property $[\varrho_0]$. We now prove that if $pqrs$ is a sliver, then the distance P from p to the closest point on the circle (y, Y) can not be too large. This lemma had been proved in [9, 14], but for completeness of presentation, we restate the lemma for later reference. The bound on P presented here is better than [9].

Lemma 2.3 [TORUS LEMMA] *[14] If tetrahedron $pqrs$ is a sliver ($\sigma(pqrs) \leq \sigma_0$ and $\rho(pqrs) \leq \varrho_0$), then $P \leq c_2 Y$, where $c_2 = 48\varrho_0\sigma_0$.*

Notice that, the constant c_2 is over-estimated. The actual constant could be much smaller.

Forbidden regions. For any base triangle qrs, there is a set of point p such that $pqrs$ is a sliver (i.e., $\sigma(pqrs) \leq \sigma_0$ and $\rho(pqrs) \leq \varrho_0$). We call them the *forbidden region* F_{qrs} of base triangle qrs. The actual shape of the forbidden region is an hourglass, see [7] By the Torus Lemma 2.3, F_{qrs} is contained in the solid torus of points at distance at most P from the circle (y, Y), as illustrated in Figure 4. The volume of that torus is the perimeter of the circle times the area of the sweeping disk, which is $2\pi Y \cdot \pi P^2$. Notice that this torus region is an over-estimate of the forbidden region F_{qrs} for base triangle qrs.

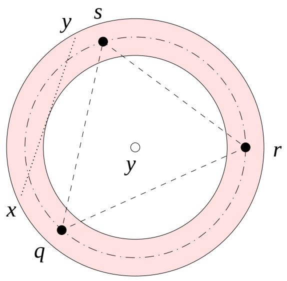

Figure 4. Every base triangle qrs that forms a tetrahedron with p defines a forbidden region inside a torus of points around the circumcircle of qrs.

We summarize the above discussions by the following lemma for later reference.

Lemma 2.4 [FORBIDDEN VOLUME] *[14] For any base triangle qrs, the volume of the forbidden region F_{qrs} is at most $c_3 Y^3$, where $c_3 = 2\pi^2(48\varrho_0\sigma_0)^2$.*

Recall that, when the circumcenter c of a bad element τ is contained inside some equatorial spheres or diametric spheres, we will split the boundary triangles and/or segments instead of selecting a point around c. In other words, we only select points from a plane or from a segment for protecting boundary. To make it possible that the selected point will not create new small slivers, we first need that the intersection of the forbidden region with any plane or segment is small. The following lemma will bound the area of the intersection of the forbidden region with any plane.

Lemma 2.5 [FORBIDDEN AREA] *[14] For any base triangle qrs and a plane $\mathcal{H}$, the area of the intersection of the forbidden region F_{qrs} with $\mathcal{H}$ is at most $c_4 Y^2$, where c_4 is a constant depending on σ_0 and ϱ_0.*

PROOF. We prove it for $c_4 = 192\pi\varrho_0\sigma_0$. The area is at most $\pi(Y+P)^2 - \pi(Y-P)^2 = 4\pi YP$. It is at most $192\pi\varrho_0\sigma_0 Y^2$ from Lemma 2.3. $\square$

Similarly, we have the following lemma to bound the length of the intersection of the forbidden region F_{qrs} with any line.

Lemma 2.6 [FORBIDDEN LENGTH] *[14] For any base triangle qrs and a line $\mathcal{L}$, the length of the intersection of the forbidden region F_{qrs} and $\mathcal{L}$ is at most $c_5 Y$, where c_5 is a constant depending on σ_0 and ϱ_0.*

PROOF. We prove it for $c_5 = 16\sqrt{3\varrho_0\sigma_0}$. The length is at most $2\sqrt{(Y+P)^2 - (Y-P)^2} = 4\sqrt{YP}$. As shown by segment xy in Figure 4. It is at most $16\sqrt{3\varrho_0\sigma_0}Y$ from Lemma 2.3. $\square$

2.6 Bounds on Small Slivers

For any bad tetrahedron τ (sliver or tetrahedron with large radius-edge ratio) in the mesh, we add a point p inside its circumsphere. The insertion of new point p removes the previous bad element, but may create new bad elements incident to p. One may try to avoid creating any new slivers by selecting a point from the picking region of τ. Unfortunately, it is not always possible to do so, see [14]. We give the following example to show that inserting any point in the picking region will create a new sliver $pqrs$, see Figure 5.

However, notice that the new slivers that can not be avoided by inserting p always have large circumradius. In other words, we could avoid creating

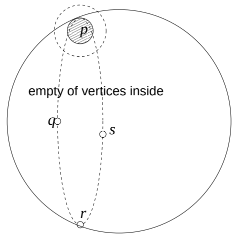

Figure 5. A base triangle qrs that forms a sliver with any point p inside the picking region of an element τ.

any new small sliver whose circumradius is within a small constant factor of R_τ. To be specific, we call a new sliver $pqrs$ a *small sliver* if its circumradius $R_{pqrs} \le bR_\tau$, where b is a constant to be specified later. For all other slivers with $R_{pqrs} > bR_\tau$, we call them *large slivers* respected to τ. The following lemma shows that we only have to avoid creating a constant number of small slivers during any single refinement step.

Lemma 2.7 [CONSTANT SMALL SLIVERS] *[14] Let p be any point in the picking region $(c, \delta R_\tau)$ of an element τ. Then there is at most constant number of new slivers $pqrs$ incident on p such that $R_{pqrs} \le bR_\tau$.*

PROOF. First, we show that the circumradius of small sliver is not too small. The length E of longest edge of $pqrs$ satisfies $E \ge (1-\delta)R_\tau$. From the Ratio Property [ϱ_0], we have $R_{pqrs} \ge E/2 \ge \frac{(1-\delta)R_\tau}{2}$. For any edge e of small sliver $pqrs$, the length of e is at most $2R_{pqrs} \le 2bR_\tau$. In other words, all vertices of small sliver $pqrs$ are in sphere $(p, 2bR_\tau)$. And from the Ratio Property [ϱ_0], we have $||e|| \ge \frac{R_{pqrs}}{\varrho_0} \ge \frac{1-\delta}{2\varrho_0} R_\tau$. In other words, all vertices of small slivers incident on p are not too close.

Then the lemma follows from a volume argument. □

We will use W to denote the number of small slivers incident on a point p in the picking region. Notice that, in above lemma 2.7, the element τ could be tetrahedron, boundary triangle or boundary segment. Observe that this lemma and the lemmas bounding the volume, area, length of the forbidden region of an element together imply the existence of point p in the picking region that avoids creating small slivers.

3 Refinement Algorithm

Herein, I build upon the algorithmic framework of Ruppert[18] and Shewchuk [19] to design a new triangular/tetrahedral Delaunay refinement algorithm. This algorithm generates meshes whose simplex elements have radius-edge ratio no greater than ϱ_0 in 3D, where $\varrho_0 > 2$ is a user specified constant. Any generated tetrahedron τ will have $\sigma(\tau) \ge \sigma_0$ for a constant σ_0 depending on ϱ_o. In other words, all mesh elements have small aspect ratio. In addition, the element size is approximately equal to the control spacing function.

We first study how to make the mesh conforming to the spacing function. Notice that, given a control spacing $f()$, a mesh element is good if both the radius-edge ratio is bounded from above and the element size is within a small constant factor of $f()$. We use the following definition to distinguish the good elements from the bad elements in our refinement algorithm.

Definition 3.1 [B-BAD ELEMENT] *Given an α-Lipschitz control spacing $f()$, a simplex is B-bad element if $\frac{R}{f(c)} > B$, where $f(c)$ is the control spacing value at its circumcenter c. The ratio $\frac{R}{f(c)}$ is called the* radius-center-spacing ratio.

We found that Chew also use the similar definition. His algorithm inserts the circumcenter of the tetrahedron whose circumradius is larger than the spacing defined at its circumcenter. After no such tetrahedron exists, it then inserts the circumcenter of the tetrahedron that has large radius-edge ratio. However, in this algorithm, we only first insert the circumcenter of tetrahedron whose radius-center-spacing ratio is larger than B. It automatically guarantees to generate an almost good and well-conformed mesh. For later convenience, we will use $\xi(\tau)$ to denote the radius-center-spacing value of tetrahedron τ. Recall that $\rho(\tau)$ is the radius-edge value of τ; $\sigma(\tau)$ is the volume per cube of shortest edge length for τ. Then slivers are the only possible bad elements after this refinement procedure. We then try to find a point in the picking region of a sliver to eliminate each sliver.

3.1 Algorithm Outline

The algorithm is built upon the classic Delaunay refinement method. It can be implemented by a small modification from classic Delaunay refinement code. As calssic Delaunay refinement, we assume that the input domain does not have small angles. Recall that small angles in the input domain may cause infinite splittings to protect domain boundary. We give the formal description of our refinement method as follows.

Algorithm: REMOVE-SLIVER(ϱ_0, σ_0, δ, **b**, **B**)

Enforce Empty Encroachment: For any diametric sphere of boundary segment, if it contains any vertex inside, then add its middle point and update the Delaunay triangulation. For any equatorial sphere of boundary triangle, if it contains any vertex inside, then add its circumcenter and update the Delaunay triangulation. If the circumcenter encroaches any boundary segment, we split the boundary segment instead of adding that circumcenter.

Clean Bad Elements: For any bad tetrahedron τ ($\xi(\tau) > B$, $\rho(\tau) > \varrho_0$ or $\sigma(\tau) \leq \sigma_0$), add a point p in the picking region of τ such that it avoids creating small slivers. If the circumcenter c_τ encroaches boundary, we applied the following rules instead of adding point p. Here the element with $\xi(\tau) > B$ has priority to be refined.

Encroach Equatorial Sphere: For any equatorial sphere of boundary triangle μ, if it contains point c_τ inside, then add a point p in the picking region of μ that avoids creating new small slivers. Update the Delaunay triangulation accordingly. But if the circumcenter c_μ of μ is contained in the diametric sphere of any boundary segment, we apply the following rule instead of adding p from $(c_\mu, \delta R_\mu)$.

Encroach Diametric Sphere: For any diametric sphere of boundary segment, if it contains point c_τ or c_μ inside, then add a point p in the segment's picking region that avoids creating new small slivers. Update the Delaunay triangulation accordingly.

Notice that we can also apply the third and fourth steps to enforce the empty boundary encroachment property (the first step). Let M_C be the mesh generated after the first step. We call it *Delaunay-refinement-conforming* mesh. In other words, all equtorial spheres of boundary triangles and dimetric spheres of boundary segments are empty of mesn vertices inside.

Notice that we have to select a point p in the picking region of bad tetrahedron (or boundary triangle, segment) that avoids creating new small slivers. Let (c_τ, R_τ) be the circumsphere of the element defining this picking region. We randomly select a point p from the picking region and construct a local mesh whose elements are all incident on p. If there is a sliver introduced in the local mesh with circumradius less than bR_τ, we discard p and reselect new point. The above procedure is continued until the local mesh is free of small slivers. Here, the local mesh is all tetrahedra that are incident on p. By defining sliver and small slivers properly, we can show that the above procedure is expected to terminate in constant rounds.

For the sake of easy analyzing, we will assume that we first refine tetrahedra that is large compared with the spacing defined at its circumcenter; followed by eliminating the slivers.

3.2 Enforcing Element Size

The main idea to guarantee the element size is as follows. Recall that the given control spacing function $f()$ is α-Lipschitz; (x, r) denote the sphere centered at point x with radius r. Let β be the constant such that there is no intersection among a set of spheres $S = \{(x, \beta f(x)) \mid x \in M_c\}$ defined on the mesh vertices of M_c. In other words, S is a sphere packing. We call sphere $(x, \beta f(x))$ the *protection sphere* of mesh vertex x. By carefully selecting B and β, the B-bad quality measure makes sure that adding points around the circumcenter of any B-bad element will not introduce any overlap among all protection spheres. Then by a simple volume argument, we know that the algorithm is guaranteed to terminate by just refining the tetrahedron with large radius-center-spacing ratio. After the algorithm terminates, the definition of the B-bad element will ensure that the resulted mesh elements are well-conformed, if B and β are selected properly. Moreover, we can also show that the radius-edge ratio of each tetrahedron is bounded from above by a constant after enforcing the control spacing function.

4 Termination and Quality Guarantee

Notice that, the elements with large $\xi(\tau)$ will have priority to be refined over the elements with large $\rho(\tau)$, or small $\sigma(\tau)$. Then we first prove that after some finite steps, all elements will have $\xi(\tau) \leq B$.

4.1 Termination without Large Element

It had been established in [15]: if $N(x) \geq \frac{2\beta}{1-\alpha\beta} f(x)$, then the protection sphere $(x, \beta f(x))$ does not overlap with any other protection sphere. Recall that after the Delaunay-refinement-conforming mesh is constructed, a point p around the circumcenter of B-bad element is "responsible" for all point insertions. If the circumcenter encroaches any subfacets or subsegments, then we add a point around the circumcenter of the encroached boundary triangle or the middle point of the encroached segments. If the circumcenter of the encroached boundary triangle also encroaches some segments, we add a point near the middle point of each encroached segment. instead of adding a point around the circumcenter of boundary triangle.

If we only add the circumcenter, instead of adding a point around the circumcenter in the picking region, [12] proved the following theorem.

Theorem 4.1 [TERMINATION BY CIRCUMCENTER] *If $B \geq \frac{4\beta}{1-(5+2\sqrt{2})\alpha\beta}$, then the algorithm will not introduce any vertex whose protection sphere is overlapped with already existed protection spheres.*

It is proved by analyzing the three cases of inserting points to a mesh: inserting the circumcenter of a bad tetrahedron, inserting the circumcenter of an encroached triangle; inserting the middle point of an encroached segment. [12] shows that the refinement algorithm is guaranteed to terminate by a simple volume argument. It is achieved by defining the protection sphere by constant $\beta = \frac{B}{4+(5+2\sqrt{2})\alpha B}$. Here we assume that there exists $\epsilon > 0$ such that $f(x) \geq \epsilon$ for all x.

The following theorem, by a similar analysis, shows that our method is guaranteed to terminate with no large element, if B and β are selected properly.

Theorem 4.2 [TERMINATE WITHOUT LARGE ELEMENT] *If $B \geq \frac{4\beta}{(1-\delta)(1-(5+2\sqrt{2})\alpha\beta)}$, then the refinement algorithm will not introduce any vertex whose protection sphere is overlapped with already existed protection spheres.*

The following theorem guarantees a good radius-edge ratio for all mesh simplices after each tetrahedron τ with $\xi(\tau) > B$ is refined.

Theorem 4.3 [RADIUS-EDGE RATIO] *If $B < \frac{1}{\alpha}$, then for all mesh elements τ*

$$\rho(\tau) \leq \frac{B}{2\beta(1-\alpha B)}.$$

PROOF. Assume that τ has four vertice p, q, r, s. For any point v on the circumsphere (c, R) of tetrahedron $pqrs$, we have $f(v) \geq f(c) - \alpha R \geq (1 - \alpha B) f(c)$. Then the length l of the shortest edge of τ satisfies $l \geq 2\beta f(v) \geq 2\beta(1-\alpha B) f(c)$. Recall that $R \leq B f(c)$. Then it follows that $\frac{R}{l} \leq \frac{B}{2\beta(1-\alpha B)}$, if $B < \frac{1}{\alpha}$. ◻

The theoretic guarantee of the radius-edge ratio of the generated mesh is almost 2 (by setting B sufficiently small), which matches the result of Shewchuk's 3D Delaunay refinement algorithm [19]. Moreover, after the algorithm refines all tetrahedra with large radius-center-spacing ratio, we know that all protection spheres centered at mesh vertices do not overlap. Thus it implies the following statement about the nearest neighbor of every mesh vertex.

Theorem 4.4 [NEAREST-NEIGHBOR] *For every mesh vertex p,*

$$N(p) \geq 2\beta/(1+\alpha\beta) f(p).$$

The above lemma had been proved in [12]. Notice that this lemma only studies the nearest neighbor function after all tetrahedra with large radius-center-spacing ratio are refined. Observe that the nearest neighbor values of the mesh vertex may be decreased after eliminating slivers. However, we can show that it only decreases by at most a constant factor.

4.2 Termination of Algorithm

We then prove that the algorithm will terminate in removing slivers if we define what is bad element properly. After the algorithm terminates, we know that all tetrahedron τ in the final mesh satisfies Ratio Property $[\varrho_0]$ and $\sigma(\tau) \geq \sigma_0$. In other words, the generated mesh elements have small aspect ratio. In addition, the mesh conforms well to the spacing function.

First of all, we classify the bad elements to three classes: *original slivers* in the mesh, *created slivers* by inserting some points, *large radius-edge elements.* Notice that, for the classical Delaunay refinement method, the termination is guaranteed by showing that the shortest distance among any two mesh vertices is not decreased. However, the distance among mesh vertices generated by this method will possibly decrease along the insertion of new points. For example, the insertion of a point p in the picking region of a sliver τ could possibly decrease the shortest distance by a constant factor. But, on the other hand, we will show that the shortest distance will not decrease too much. Then by a volume argument, we know that the algorithm will terminate. To be convenient, we will use l_{pre} to denote the shortest edge length before inserting a point p; use l_{aft} to denote the shortest edge length after inserting a point p; use l_{org} to denote the shortest edge length of original mesh after the first step of our algorithm (enforce empty boundary encroachment).

4.2.1 Eliminate Original Slivers.

Assume τ is an original sliver after large elements are removed. Notice that $R_\tau \geq l_{pre}/2$. Then we have $l_{aft} \geq \frac{1-\delta}{2} l_{pre}$. Assume that there are m original slivers in the input mesh. Then after inserting points in the picking region of all initial slivers, the shortest edge of the mesh is at least $(\frac{1-\delta}{2})^m l_{org}$. This may be very bad, even there is a bound on the decreasing of shortest edge length. We will show that the shortest edge after eliminating all original slivers is decreased by only a constant factor that is not dependent on m.

Lemma 4.5 [ORIGINAL SLIVERS] *After eliminating all original slivers, the shortest edge of the mesh is at least* $(1-\delta)/4$ *of that of the original mesh.*

PROOF. For simplicity, assume that p_i, $i = 1, 2, \ldots$, are the points inserted to remove original slivers τ_j, $j = 1, 2, \ldots$ respectively. Notice that, it may need insert many points to remove a sliver whose circumcenter encroaches domain boundary. Some of the original slivers may be eliminated by selecting point from the picking region of other sliver.

Let (c_i, R_i) be the sphere responsible for selecting point p_i, i.e, p_i is selected from $(c_i, \delta R_i)$. Then for any $i < j$, inserting p_j implies that point p_i is not contained in sphere (c_j, R_j). This is from the Delaunay property. In other words, we have $||p_i - c_j|| \geq R_j$. It implies that $||p_i - p_j|| \geq (1-\delta) R_j$.

If point p_j is selected from the picking region of an original sliver τ_k, then $R_i = R_k$. It implies that

$$||p_i - p_j|| \geq (1-\delta) R_k \geq (1-\delta) l_{org}/2.$$

Assume that p_j is selected from the picking region of a boundary triangle or boundary segment. In other words, there is an original sliver τ_k, whose circumcenter c_τ is contained in sphere (c_j, R_j); or c_τ encroaches the circumsphere (c_μ, R_μ) of a boundary triangle μ, and c_μ is contained in the circumsphere (c_j, R_j) of a boundary segment. In both cases, we have $R_j \geq R_\tau/2$. Notice that $R_\tau \geq l_{org}/2$. We have

$$||p_i - p_j|| \geq (1-\delta) R_\tau/2 \geq (1-\delta) l_{org}/4.$$

Then the lemma follows. □

4.2.2 Eliminate Other Bad Elements

Let τ be a tetrahedron that has large radius-edge ratio or be a sliver created during the refinement. Notice that to eliminate τ, we have to insert at least one point inside the circumsphere of τ. There are two sources for inserting points: a point from the picking region of τ; or the point from the picking region of boundary triangles or segments. Then tetrahedron τ will be eliminated in finite steps [14].

Then we show that the shortest edge of the mesh does not change much by eliminating the tetrahedron with large radius-edge ratio or a created sliver. Let τ be such a bad element. Notice that to eliminate τ, we have to insert at least one point inside the circumsphere of τ. There are two sources for inserting points: a point from the picking region of τ; or the point from the picking region of other tetraherdon, boundary triangles or segments.

Lemma 4.6 [OTHER BAD ELEMENTS] *Assume that tetrahedron τ has large radius-edge ratio or is a sliver created along point insertion. Then the shortest edge length of the mesh after eliminating τ is at least*

$$\min(\frac{(1-\delta)\varrho_0}{2}, \frac{(1-\delta)b}{4})$$

factor of the shortest edge length before τ is eliminated.

PROOF. First, we consider the case when the circumcenter c of τ does not encroach the domain boundary. Assume p is inserted to remove element τ with radius-edge ratio $\frac{R_\tau}{l_\tau} > \varrho_0$. The shortest edge introduced after inserting p is at least $(1-\delta)\varrho_0 l_\tau$. In other words, we have $l_{aft} \geq (1-\delta)\varrho_0 l_{pre}$.

Then assume that τ is a sliver created along the insertion of point from the picking region of bad elements. Let $f(\tau)$ be that bad element. In other words, $f(\tau)$ is responsible for creating the new sliver τ. Then we have $R_\tau \geq bR_{f(\tau)}$. The shortest edge introduced after inserting point p is at least $(1-\delta)R_\tau$, which is at least $(1-\delta)bR_{f(\tau)}$. Recall that for any tetrahedron $f(\tau)$, we have $R_{f(\tau)} > l_{f(\tau)}/2$. Then the theorem follows.

It remains to show that the theorem is also true when the circumcenter c encroaches the domain boundary. Assume that c encroaches the circumsphere (v, R_v) of a boundary triangle or segment. Notice that $R_v \geq R_\tau/2$. Thus the shortest edge introduced after selecting point p from $(v, \delta R_v)$ is at least $\frac{1-\delta}{2}R_\tau$.

If τ is an element that does not satisfy the Ratio Property $[\varrho_0]$, then we have $R_\tau \geq \varrho_0 L_\tau$. Therefore, we have $l_{aft} \geq \frac{(1-\delta)\varrho_0}{2} l_{pre}$.

If τ is a sliver created during eliminating some other bad elements $f(\tau)$, we have $R_\tau > bR_{f(\tau)}$. Then we have $l_{aft} \geq \frac{(1-\delta)b}{4} L_{f(\tau)}$. The lemma then follows. ◻

4.2.3 Main Theorem

Combining all above analysis, we have the following theorem.

Theorem 4.7 [SHORTEST EDGE LENGTH] *[14] If we select b, δ and ϱ_0 such that $(1-\delta)b \geq 4$ and $(1-\delta)\varrho_0 \geq 2$, the shortest edge length of the mesh will never decrease after all original slivers are eliminated,*

Then given $\varrho_0 > 2$, we select $\delta \leq 1 - \frac{2}{\varrho_0}$ to define the picking region and use $b \geq 2\varrho_0$ to define large slivers.

Notice that we have to make sure that there is a point in the picking region that avoids creating new small slivers. Thus the following conditions are sufficient: (1) $Wc_3(bR_\tau)^3 < (\delta R_\tau)^3$; (2) $Wc_4(bR_\tau)^2 < (\delta R_\tau)^2$; (3)$Wc_5 bR_\tau < \delta R_\tau$. Recall that $c_3 = 2\pi^2(48\varrho_0\sigma_0)^2$; $c_4 = 192\pi\varrho_0\sigma_0$ and $c_5 = 16\sqrt{3\varrho_0\sigma_0}$. In other words, the σ_0 used to define sliver has to satisfy all three inequalities.

Then we have the following main theorem.

Theorem 4.8 [MAIN THEOREM] *The refinement algorithm generates a mesh whose elements have small aspect ratio. In other words, for any tetrahedron τ, we have $\rho(\tau) \leq \varrho_0$ and $\sigma(\tau) \geq \sigma_0$.*

In [14], it is showed that the element size after eliminating slivers of an almost good mesh is at least a constant factor of that of the input mesh. Therefore, we know that the generated final mesh not only is well-shaped but also it is well-conformed.

5 Discussions

In this paper, we present refinement based method that generates well-shaped mesh respecting to a control spacing, i.e., the final mesh is a sliver-free Delaunay mesh. In other words, any tetrahedron generated in the mesh has radius-edge ratio no more than ϱ_0 and the volume is at least σ_0 times the cube of its shortest edge length. It solves a long standing open problem.

Notice that the σ_0 derived from all the proofs may be too small for any practical use (even, it is better than previous results [4, 9]). We would like to conduct some experiments to see what σ_0 can guarantee that there are no small slivers created. Observe that the picking region is sufficiently large, so typically, the generated tetrahedron elements will have much large σ values than σ_0.

Recall that the termination guarantee does not depend on the definition of sliver. Only the existence of point p, which will not introduce small slivers, in the picking region depends on the sliver definition. Based on this observation, we can have a variation of this algorithm. We remove sliver τ only if we find a point p in the picking region of τ such that the new tetrahedra with circumradius less than bR_τ is better. If the algorithm terminates, each tetrahedron τ of the generated mesh has $\sigma(\tau) \geq \sigma_0$. However, the termination guarantee is not so obvious. Unlike the algorithm presented by this paper, this variation may refine the small slivers, and which in turn introduces shorter edges to the mesh. We leave the termination guarantee as an open problem. For more discussions, the reader is referred to [13].

6 Acknowledgements

The author thanks the anonymous referees very much for many helpful reviews to improve the paper.

References

[1] Bern, M., Eppstein, D., and Gilbert, J. R. Provably good mesh generation. In *the 31st Annual Symposium on Foundations of Computer Science, IEEE* (1990), pp. 231–241.

[2] Buscaglia, G. C., and Dari, E. A. Anisotropic mesh optimization and its application in adaptivity. *International Journal for Numerical Methods in Engineering 40* (1997), 4119–4136.

[3] Cavendish, J. C., Field, D. A., and Frey, W. H. An approach to automatic three-dimensional finite element mesh generation. *Internat. J. Numer. Methods Engrg. 21* (1985).

[4] Cheng, S. W., Dey, T. K., Edelsbrunner, H., Facello, M. A., and Teng, S. H. Silver exudation. In *Proc. 15th ACM Symposium on Computational Geometry* (1999), pp. 1–13.

[5] Chew, L. P. Constrained Delaunay triangulations. *Algorithmica 4* (1989), 97–108.

[6] Chew, L. P. Guaranteed-quality mesh generation for curved surfaces. In *9th ACM Sym. on Comp. Geometry* (1993), pp. 274–280.

[7] Chew, L. P. Guaranteed-quality delaunay meshing in 3d (short version). In *13th ACM Sym. on Comp. Geometry* (1997), pp. 391–393.

[8] Diaz, M. J. C., Hecht, F., and Mohammadi, B. New progress in anisotropic grid adaptation for inviscidand viscous flows simulations. Tech. Rep. 2671, INRIA, Oct 1995.

[9] Edelsbrunner, H., Li, X. Y., Miller, G., Stathopoulos, A., Talmor, D., Teng, S. H., ÜNgÖR", A., and Walkington, N. Smoothing and cleaning up slivers. In *ACM Symposium on Theory of Computing (STOC00)* (2000).

[10] Edelsbrunner, H., and Shah, N. R. Incremental topological flipping works for regular triangulations. *Algorithmica 15* (1996).

[11] Freitag, L., and Ollivier-Gooch, C. Tetrahedral mesh improvement using swapping and smoothing. *International Journal of Numerical Methods in Engineering 40* (1997), 3979–4002.

[12] Li, X. Y. Functional delaunay refinement. In *7th International Conference on Numerical Grid Generation in Computational Field Simulations* (2000).

[13] Li, X. Y. *Sliver-free Three Dimensional Delaunay Mesh Generation.* PhD thesis, University of Illinois at Urbana-Champaign, 2000.

[14] Li, X. Y., and Teng, S. H. Sliver-free three dimensional delaunay mesh generation. Submitted for Publication, 2000.

[15] Li, X. Y., Teng, S. H., and Üngör, A. Biting: advancing front meets sphere packing. *Int. Jour. for Numerical Methods in Eng* (2000).

[16] Miller, G. L., Talmor, D., Teng, S. H., and Walkington, N. A delaunay based numerical method for three dimensions: generation, formulation, and partition. In *Proc. 27^{th} Annu. ACM Sympos. Theory Comput.* (1995), pp. 683–692.

[17] Miller, G. L., Talmor, D., Teng, S. H., Walkington, N., and Wang, H. Control volume meshes using sphere packing: generation, refinement, and coarsening. In *5th International Meshing Roundtable* (1996), Sandia National Laboratories, pp. 47–61.

[18] Ruppert, J. A new and simple algorithm for quality 2-dimensional mesh generation. In *Third Annual ACM-SIAM Symposium on Discrete Algorithms* (1992), pp. 83–92.

[19] Shewchuk, J. R. Tetrahedral mesh generation by delaunay refinement. In *14th Annual ACM Symposium on Computational Geometry* (1998), pp. 86–95.

[20] Shimada, K. *Physically-based Mesh Generation:Automated Triangulation of surfaces and Volumes via Bubble Packing.* PhD thesis, MIT, Cambridge, 1993.

[21] Strang, G., and Fix, G. J. *An Analysis of the Finite Element Method.* Prentice-Hall, 1973.

[22] Talmor, D. *Well-Spaced Points for Numerical Methods.* PhD thesis, Carnegie Mellon University, 1997.