Functional Delaunay Refinement

Xiang-Yang Li
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801
xli2@cs.uiuc.edu

Abstract:

Given a complex of vertices, constraining segments (and planar straight-line constraining facets in 3D) and an α-Lipschitz control spacing function $f()$ over the domain, an algorithm presented herein can generate a conforming mesh of Delaunay triangles (tetrahedra in 3D) whose circumradius-to-shortest-edge ratios are no greater than $\sqrt{2}$ (2 in 3D). The triangle (tetrahedron) size is within a constant factor of $f()$. An implementation in 2D demonstrates that the algorithm generates excellent mesh.

Keywords: unstructured mesh generation, delaunay refinement, control spacing, mesh conformity.

1 Introduction

Simplex mesh generation has many applications, including numerical methods such as the finite element method, computer graphics and geographic information system. To ensure accurate result, the simplices of the mesh must be *well-shaped*, in the sense that they have small aspect ratios, i.e., the smallest angles are bounded from below [1, 15]. To get accurate numerical simulation, we also expect to have small element size. On the other hand, we prefer to have larger element size due to the time complexity of the simulation. Hence elements should have properly chosen size and shape that adapt to the complex geometry and solution accuracy.

A control spacing function specifies the desired element size at each point of the domain. Several heuristics and algorithms had been developed to generate a mesh whose element sizes conform well to a given control spacing function. Splitting the longest edge, or subdividing the simplices are the most used heuristics. Shimada [14] used the particle simulation to find a good mesh vertices set, then constructed the final mesh by Delaunay triangulation. There are no any theoretical quality guarantees given by above algorithms. The algorithm by Miller *et al* [11] uses the maximal independent set of a large random sampled points set as the mesh vertices, then triangulate them using the Delaunay method. Li *et al* [9] recently had proposed a new method, called *biting*, which uses the advancing front method to construct a tight sphere packing. It generates a mesh with radius-edge ratio about 1 [9] inside the domain.

Mesh generation algorithms based on Delaunay refinement are effective both in theory and in practice. Paul Chew [4] developed the first Delaunay refinement algorithm for a PSLG (2D) domain. Chew's algorithm generates a uniform mesh whose angles are bounded between 30^o and 120^o. Chew [5] has also proposed two dimensional Delaunay refinement algorithms that produce meshes of well shaped triangles whose sizes are no more than the spacing defined on circumcenters. Jim Ruppert's algorithm [12] generates a well shaped mesh with provable nice gradation on mesh elements size. Shewchuk [13] built upon the algorithmic and analytical framework of Ruppert to design a new tetrahedral Delaunay refinement algorithm. It generates meshes whose tetrahedra have radius-edge ratios (defined shortly) no greater than a bound $B > 2$.

Herein, I build upon the algorithmic framework of Ruppert[12] and Shewchuk [13] to design a new Delaunay refinement algorithm. This algorithm generates meshes whose simplex elements have radius-edge ratio no greater than 2 in 3D and $\sqrt{2}$ in 2D. And the nearest neighbor value, edge length function (defined shortly) of any mesh vertex are within a small constant factor of the given control spacing. My algorithm is distinguished from those of Ruppert and Shewchuk by the ability to handle a given control spacing with the guarantee of good conforming to the control spacing. It is also easy to transform classic Delaunay refinement program to implement our algorithm. The main theoretical deficiency of the algorithm is its assumption that the given control spacing is bounded from above by a constant factor of the nearest neighbor function defined by the Delaunay-refinement-conforming mesh M_c (defined shortly). Notice that it is rare that this assumption can not be satisfied in practice as showed in Section 5.

2 Preliminary

Well-shaped and well-conformed mesh: The aspect ratio of a simplex is often defined as the ratio of radius of the circumsphere to the radius of the inscribed sphere. But unfortunately, it is hard to generate mesh with small aspect ratio in 3D. An alternative is to use the *radius-edge ratio* measurement as defined in [10, 11]. It is the ratio of a simplex's circumradius over the length of its shortest edge, which is the metric that is naturally optimized by Delaunay refinement algorithms [12, 13]. One would like this ratio to be as small as possible. Notice that in 3D, sliver is the only element with small radius-edge ratio but large aspect ratio.

Recently, new methods [2, 7, 8] had been proposed to theoretically remove the slivers inside. Based on the radius-edge ratio quality measure, we say a mesh M is *ρ-well-shaped* if the maximum radius-edge ratio over all of its elements is at most ρ.

The spacing function $f()$ can be derived from the geometry condition such as local feature size [12], or from an *a priori* error analysis, or an *a posteriori* error analysis based on an initial numerical simulation. Generally, $f()$ is the combination of all above. Given a mesh M, we capture the size of the elements that contain a point $\boldsymbol{x} \in \Omega$ as follows. For point $\boldsymbol{x} \in \Omega$, $el_M(x)$ is the length of the longest edges of all mesh elements that contain $\boldsymbol{x}$; $nn_M(\boldsymbol{x})$ is the distance of $\boldsymbol{x}$ to the second nearest mesh vertex in M. Notice that $\boldsymbol{x}$ is the closest one if itself is mesh vertex.

In the ideal mesh M, the spacing function derived at any vertex of M should be within a constant factor of $f()$. The smaller the constant, the better the mesh conforms to $f()$. Thus, we define the conformity of the mesh as following to capture how well the mesh conforms to the given control spacing $f()$.

Definition 2.1 **[Conformity]** *For mesh vertex $\boldsymbol{x}$, $c(\boldsymbol{x}) = \min(\frac{nn(\boldsymbol{x})}{f(\boldsymbol{x})}, \frac{f(\boldsymbol{x})}{nn(\boldsymbol{x})})$ is the conformity of $\boldsymbol{x}$ to spacing function $f()$.*

Delaunay refinement: The main ingredient of Chew's [3, 6], Ruppert's [12], and Shewchuk's [13] Delaunay refinement algorithm is the insertion of a vertex at the circumcenter of a triangle or tetrahedron with poor quality. The following concepts are used in classic Delaunay refinement to protect the domain boundary. The *diametric sphere* of a subsegment is the smallest sphere that encloses it. A subsegment is said to be *encroached* if its diametric sphere contains a vertex other than its endpoints [12]. Any encroached subsegment is split into two subsegments by inserting its midpoint; see Figure 1 (c). The *equatorial sphere* of a triangular subfacet is the smallest sphere that passes through the three vertices of the subfacet. A subfacet is encroached if a noncoplanar vertex lies inside or on its equatorial sphere [13]. Each encroached subfacet is normally split by inserting its circumcenter; see Figure 1 (b). However, if the new vertex would encroach upon any subsegment, it is not inserted; instead, all the subsegments it would encroach upon are split.

Notice that a subsegment/subfacet may be encroached by a point p whether or not p actually appears in the mesh. Encroached subsegments are given priority over encroached subfacets, which have priority over skinny tetrahedra. These encroachment rules are intended to recover missing segments and facets, and to ensure that all vertex insertions are valid. The first time the mesh reaches this state (all subsegments/subfacets are not encroached), we call the mesh *the-Delaunay-refinement-conforming mesh*, denoted by M_c. We assume that there is a positive constant D such that for any mesh vertex $\boldsymbol{x} \in M_c$, $nn_{M_c}(\boldsymbol{x}) \geq Df(\boldsymbol{x})$, where $nn_{M_c}(\boldsymbol{x})$ is the nearest neighbor function defined by mesh M_c.

3 Algorithm Outline

As in Delaunay refinement algorithm, we have to apply some criteria to measure the elements quality of the mesh. Notice, given a control spacing $f()$, a mesh element is good if both the radius-edge ratio is bounded from above and the conformity of its vertices is bounded from below. We use the following definition to distinguish the good elements from the bad elements in our algorithm.

Definition 3.1 (B-Bad Element) *Assume $f()$ is α-Lipschitz, a simplex is B-bad element if $\frac{R}{f(\boldsymbol{c})} > B$, where $\alpha B < 1$ and $\mathbf{c}$ is the element's circumcenter. The ratio $\frac{R}{f(\boldsymbol{c})}$ is called the* radius-center-spacing ratio.

For later convenience, hereafter, let R be the circumradius; L be the longest edge length; l be the shortest edge length of an element. Then we give the formal description of our functional Delaunay refinement *FDR* method as follows.

Algorithm: FUNCTIONAL-DELAUNAY-REFINEMENT(B)

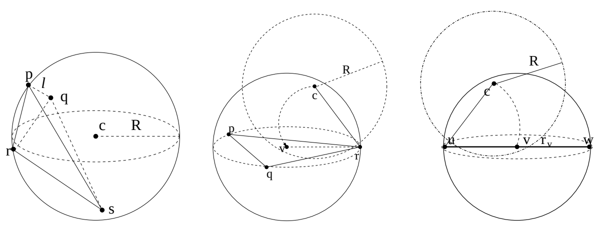
1. [**Boundary Edge Encroach**] Split any encroached boundary subsegment by adding its midpoint;
2. [**Boundary Face Encroach**] Split any encroached boundary subfacet by adding its circumcenter. However, if the new point would encroach subsegments, apply rule 1 to these subsegments instead;
3. [**Remove Bad Element**] Split any B-bad element simplex by adding its circumcenter. However, if the circumcenter would encroach any subsegment or subfacet, then apply rules 1 and/or 2 instead.

The main idea of the algorithm is as follows. Let $B(\boldsymbol{x}, r)$ denote the sphere centered at point $\boldsymbol{x}$ with radius r. We call sphere $B(\boldsymbol{x}, \beta f(\boldsymbol{x}))$ the *protection sphere* of mesh vertex $\boldsymbol{x}$. By carefully selecting B and β, the B-bad element definition makes sure that the above algorithm will not introduce any overlap among all protection spheres at mesh vertices. Then by a simple volume argument, we know that the algorithm is guaranteed to terminate. After the algorithm terminates, the resulted mesh elements are well shaped and well conformed, if B and β are selected properly.

4 Proof of Termination and Quality of FDR

In this section, we show that the algorithm is guaranteed to terminate, the radius edge ratio of element is bounded from above and the element size conforms well to the given function $f()$.

For any noninput vertex $\boldsymbol{v}$ (whether inserted or rejected), let parent $p(\boldsymbol{v})$ be the vertex "responsible" for the insertion of $\boldsymbol{v}$. For convenience, we often use $\boldsymbol{c}$ to denote $p(\boldsymbol{v})$. See Figure 1. Recall that after the Delaunay-refinement-conforming mesh is constructed, the circumcenter of B-bad element is "responsible" or the parent of "responsible" for all point insertions. With each vertex $\boldsymbol{v}$, also associate an insertion radius $r_{\boldsymbol{v}}$ equal to the length of the shortest edge connected to $\boldsymbol{v}$ immediately after $\boldsymbol{v}$ is introduced into the tetrahedralization. Notice that $\boldsymbol{v}$ may not have to be inserted into the mesh actually: if it encroaches some subfacets or subsegments, then it is rejected.



(a) (b) (c)

Figure 1: Three cases of inserting points: (a) inserting the circumcenter; (b) split an encroached boundary facet; (c) split an encroached segment.

Proof of termination is based on the following lemma proved in [9].

Lemma 4.1 *For any points $\boldsymbol{x}$ and $\boldsymbol{y}$, the spheres $B(\boldsymbol{x}, \beta f(\boldsymbol{y}))$ and $B(\boldsymbol{y}, \beta f(\boldsymbol{y}))$ will not overlap if $||\boldsymbol{x}-\boldsymbol{y}|| \geq \frac{2\beta}{1-\alpha\beta}\min(f(\boldsymbol{x}), f(\boldsymbol{y}))$.*

The following theorem proves the termination guarantee, if B and β are selected properly.

Theorem 4.2 [Terminate Theorem] *If $B \geq \frac{4\beta}{1-(3+2\sqrt{2})\alpha\beta}$, then introduced protection sphere is not overlapped with existed protection spheres.*

Proof: We prove it by analyzing the three cases of inserting points to a mesh:

1. The circumcenter $\mathbf{c}$ of a B-bad element is inserted. See Figure 1 (a). Then from lemma 4.1, we know that if $B \geq \frac{2\beta}{1-\alpha\beta}$, then the

protection sphere $B(\boldsymbol{c}, f(\boldsymbol{c}))$ will not intersect with any other existed protection spheres, because $nn(\boldsymbol{c}) = R \geq Bf(\boldsymbol{c}) \geq \frac{2\beta}{1-\alpha\beta} f(\boldsymbol{c})$.

2. The circumcenter $\boldsymbol{v}$ of a boundary facet τ is inserted because the circumcenter $\boldsymbol{c}$ of a B-bad element encroaches τ. See Figure 1 (b). Notice that $r_{\boldsymbol{c}} \geq Bf(\boldsymbol{c})$.

   If point $\boldsymbol{v}$ is contained in sphere $B(\boldsymbol{c}, \beta f(\boldsymbol{c}))$ then $r_{\boldsymbol{v}} \geq \frac{\sqrt{2}}{2} r_{\boldsymbol{c}}$ (see [13]) and $f(\boldsymbol{v}) \leq f(\boldsymbol{c}) + \alpha r_{\boldsymbol{v}} \leq (\frac{\sqrt{2}}{B} + \alpha) r_{\boldsymbol{v}}$. It follows that $r_{\boldsymbol{v}} \geq \frac{B}{\sqrt{2}+\alpha B} f(\boldsymbol{v})$. Recall that before $\boldsymbol{c}$ is introduced (it is actually rejected), the equatorial sphere $B(\boldsymbol{v}, r_{\boldsymbol{v}})$ is empty. To make sure that the protection sphere $B(\boldsymbol{v}, \beta f(\boldsymbol{v}))$ will not overlap with any existed protection sphere, $r_{\boldsymbol{v}} \geq \frac{2\beta}{1-\alpha\beta} f(\boldsymbol{v})$ is a sufficient condition. Then we need $\frac{B}{\sqrt{2}+\alpha B} \geq \frac{2\beta}{1-\alpha\beta}$. Which implies that we need $B \geq \frac{2\sqrt{2}\beta}{1-3\alpha\beta}$.
   If point $\boldsymbol{v}$ is not contained in sphere $B(\boldsymbol{c}, \beta f(\boldsymbol{c}))$ then $r_{\boldsymbol{v}} \geq r_{\boldsymbol{c}}$ and $f(\boldsymbol{c}) \geq f(\boldsymbol{v}) - \alpha r_{\boldsymbol{v}}$. Then we have $r_{\boldsymbol{v}} \geq Bf(\boldsymbol{c}) \geq Bf(\boldsymbol{v}) - \alpha B r_{\boldsymbol{v}}$. It follows that $r_{\boldsymbol{v}} \geq \frac{B}{1+\alpha B} f(\boldsymbol{v})$. Similarly, we have to make sure that $r_{\boldsymbol{v}} \geq \frac{2\beta}{1-\alpha\beta} f(\boldsymbol{v})$. Then we need $\frac{B}{1+\alpha B} \geq \frac{2\beta}{1-\alpha\beta}$. Which implies that we need $B \geq \frac{2\beta}{1-3\alpha\beta}$.
   Combine above two subcases,
   $$B \geq \frac{2\sqrt{2}\beta}{1-3\alpha\beta}$$
   is a sufficient condition that the new inserted protection sphere will not overlap with any existed sphere. Notice that in both subcases, we have $r_{\boldsymbol{v}} \geq \frac{B}{\sqrt{2}+\alpha B} f(\boldsymbol{v})$.

3. The middle point $\boldsymbol{v}$ of a boundary segment is inserted because the circumcenter $\boldsymbol{c}$ of either a B-bad element or an encroached facet encroaches the segment. If the parent point $\boldsymbol{c}$ of $\boldsymbol{v}$ is the circumcenter of a B-bad element, then similar to previous case 2, a sufficient condition is $B \geq \frac{2\sqrt{2}\beta}{1-3\alpha\beta}$. If the parent point $\boldsymbol{c}$ of $\boldsymbol{v}$ is the circumcenter of another encroached facet (but it is rejected), then from the result of previous case, we know that $r_{\boldsymbol{c}} \geq \frac{B}{\sqrt{2}+\alpha B} f(\boldsymbol{c})$. Then again similar to the proof of previous case 2, a sufficient condition to avoid overlapping is $\frac{B'}{\sqrt{2}+\alpha B'} \geq \frac{2\beta}{1-\alpha\beta}$, where $B' = \frac{B}{\sqrt{2}+\alpha B}$. It implies that $B \geq \frac{4\beta}{1-(3+2\sqrt{2})\alpha\beta}$.
   Combining both subcases, a sufficient condition is
   $$B \geq \frac{4\beta}{1-(3+2\sqrt{2})\alpha\beta}.$$

Then the sufficient conditions (of three cases) to avoid overlapping are $B \geq \frac{2\beta}{1-\alpha\beta}$, $B \geq \frac{2\sqrt{2}\beta}{1-3\alpha\beta}$ and $B \geq \frac{4\beta}{1-(3+2\sqrt{2})\alpha\beta}$. Consequently, the theorem follows from the fact that, if

$$B \geq \frac{4\beta}{1-(3+2\sqrt{2})\alpha\beta},$$

all sufficient conditions are satisfied. □

In other words, if the protection sphere is defined by the constant $\beta = \min(\frac{B}{4+(3+2\sqrt{2})\alpha B}, \frac{D}{2+\alpha D})$, the functional Delaunay refinement algorithm is guaranteed to terminate by a simple volume argument. Here we assume that there exists $\epsilon > 0$ such that $f(\boldsymbol{x}) \geq \epsilon$ for all $\boldsymbol{x}$. Then after the algorithm terminates, we know that $\frac{R}{f(\boldsymbol{c})} \leq B$ for all mesh simplices. The following theorem guarantees a good radius-edge ratio for all mesh simplices.

Theorem 4.3 **[Radius-Edge Ratio]** *If $B < \frac{1}{\alpha}$, for all elements*

$$\frac{R}{l} \leq \frac{B}{2\beta(1-\alpha B)}.$$

Proof: For any point $\boldsymbol{v}$ on the circumsphere $B(\boldsymbol{c}, R)$ of tetrahedron $\boldsymbol{pqrs}$, we have $f(\boldsymbol{v}) \geq f(\boldsymbol{c}) - \alpha R \geq (1-\alpha B)f(\boldsymbol{c})$. Then the length of the shortest edge $l \geq 2\beta f(v) \geq 2\beta(1-\alpha B)f(c)$. Recall $R \leq Bf(c)$. Thus $\frac{R}{l} \leq \frac{B}{2\beta(1-\alpha B)}$, if $B < \frac{1}{\alpha}$. □

Assume B is selected such that $\frac{B}{4+(3+2\sqrt{2})\alpha B} \leq \frac{D}{2+\alpha D}$. Then the radius-edge ratio of mesh elements generated by functional Delaunay refinement is at most $\frac{2+(1.5+\sqrt{2})\alpha B}{1-\alpha B}$. It shows that the theoretic guarantee of the radius-edge ratios of the mesh elements generated are close to 2 (by setting B almost 0). The fact that all protection spheres do not overlap implies the following statement about the nearest neighbor of every mesh vertex.

Theorem 4.4 **[Nearest-neighbor]** *For every mesh vertex $\boldsymbol{p}$,*

$$nn(\boldsymbol{p}) \geq 2\beta/(1+\alpha\beta)f(\boldsymbol{p}).$$

Proof: For mesh vertices $\boldsymbol{p}, \boldsymbol{q}$, the spheres $B(\boldsymbol{p}, \beta f(\boldsymbol{p}))$ and $B(\boldsymbol{q}, \beta f(\boldsymbol{q}))$ will not overlap. Then $||\boldsymbol{p}-\boldsymbol{q}|| \geq \beta(f(\boldsymbol{p}) + f(\boldsymbol{q}))$. The theorem follows from $f(\boldsymbol{q}) \geq f(\boldsymbol{p}) - \alpha||\boldsymbol{p}-\boldsymbol{q}||$. □

Similarly, if $\frac{B}{4+(3+2\sqrt{2})\alpha B} \leq \frac{D}{2+\alpha D}$, for mesh vertex $\boldsymbol{p}$, we have

$$nn(\boldsymbol{p}) \geq \frac{2B}{4+(5+2\sqrt{2})\alpha B} f(\boldsymbol{p}).$$

The following theorem shows that the value $el()$ at any mesh vertex is also bounded from above respecting to its control spacing function.

Theorem 4.5 **[Edge-length]** *For vertex* $\boldsymbol{p}$, $el(\boldsymbol{p}) \leq \frac{2B}{1+\alpha B} f(\boldsymbol{p})$.

Proof: For any mesh vertex $\boldsymbol{p}$, let $\boldsymbol{pqrs}$ be a tetrahedron incident on $\boldsymbol{p}$ with the longest edge $\boldsymbol{pq}$. Let R be the circumradius of $\boldsymbol{pqrs}$. Then $el(\boldsymbol{p}) = ||\boldsymbol{p} - \boldsymbol{q}|| \leq 2R \leq 2Bf(\boldsymbol{c})$. Notice that $f(\boldsymbol{p}) \geq f(\boldsymbol{c}) + \alpha R \geq (1 + \alpha B) f(\boldsymbol{c})$. Thus we have $el(\boldsymbol{p}) \leq \frac{2B}{1+\alpha B} f(\boldsymbol{p})$. □

Notice the above Theorems 4.4 and 4.3 implies that

$$\frac{2\beta}{1+\alpha\beta} \leq \frac{nn(\boldsymbol{p})}{f(\boldsymbol{p})} \leq \frac{el(\boldsymbol{p})}{f(\boldsymbol{p})} \leq \frac{2B}{1+\alpha B}.$$

The 2D, we use constant $\beta = \min(\frac{B}{2\sqrt{2}+3\alpha B}, \frac{D}{2+\alpha D})$ to define the protection circles. If $\beta = \frac{B}{2\sqrt{2}+3\alpha B}$ and $B < \frac{1}{\alpha}$, then $\frac{R}{l} \leq \frac{2\sqrt{2}+3\alpha B}{2(1-\alpha B)}$. Then the minimal angle θ satisfies $\sin(\theta) \geq \frac{1-\alpha B}{2\sqrt{2}+3\alpha B}$.

5 Experiments and Discussions

We had conducted some experiments of our algorithm on two dimensional domain. Each of the meshes illustrated was generated by enforcing a lower bound on the radius-center-spacing ratio, rather than the radius-edge ratio. See definition 3.1. However, the implementation gives priority to triangles with high radius-center-spacing ratio. The experiments show that the quality of meshes generated is much better than the theoretic guarantee. The angle quality and the control spacing conformity is plotted in the following Figure 3. We observed that using different B does not change much on the angle distribution. The protection sphere does not appear in implementation.

Recall that we assume there is a positive constant D such that $nn_{M_c}(\boldsymbol{x}) \geq Df(\boldsymbol{x})$ for every vertex $\boldsymbol{x} \in M_c$. We show that it is reasonable in the following sense. First, to make it possible to generate a well-shaped and well-conformed mesh, $f()$ should be less than a constant factor of the local feature size $lfs()$, let us say $lfs(\boldsymbol{x}) \geq \delta \cdot f(\boldsymbol{x})$. Following the result by Ruppert [12] and Shewchuk [13], there is a constant ω such that $nn_{M_c}(\boldsymbol{x}) \geq \omega \cdot lfs(\boldsymbol{x})$, if the input domain has no acute angles. Then we have $nn_{M_c}(\boldsymbol{x}) \geq \omega\delta \cdot f(\boldsymbol{x})$. In other words, $D = \omega\delta$ satisfies our assumption.

References

[1] Babuška, I., and Aziz, A. K. On the angle condition in the finite element method. *SIAM J. Numer. Anal. 13(2)* (1976), 214–226.

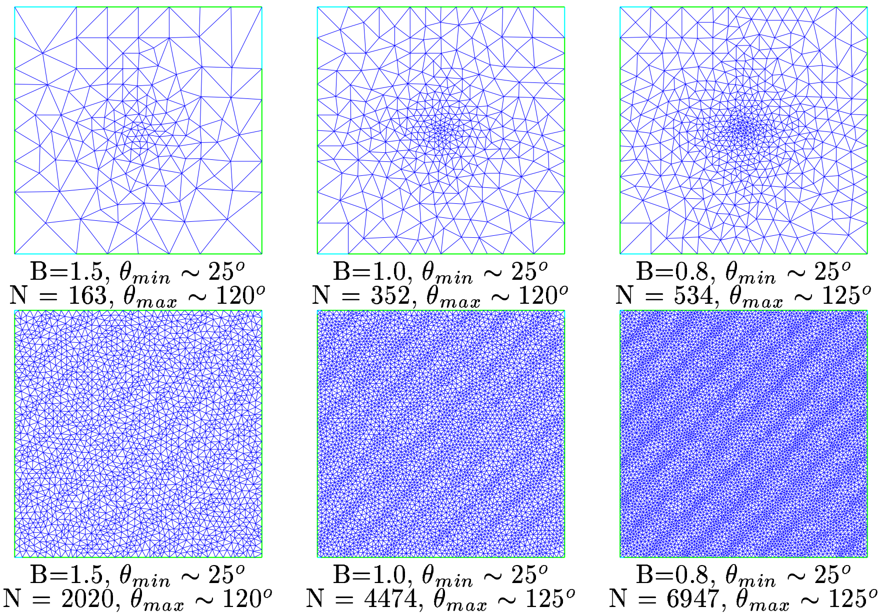


Figure 2: Meshes created on a $[0, 10] \times [0, 10]$ square. N is the number of mesh vertices; θ_{min} and θ_{max} are the minimal and maximal angle. The spacing function is defined as, $(||x - 5|| + ||y - 5||) * 0.1 + 0.1$ for the up figures; $(||\sin(x + y)|| + ||\cos(x + y)||) * 0.1$ for low figures.

[2] 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.

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

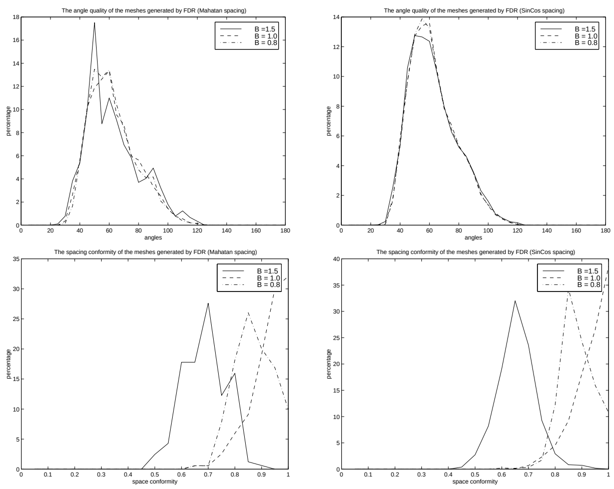
[4] Chew, L. P. Guaranteed-quality triangular meshes. Tech. Rep. TR-89-983, Department of Computer Science, Cornell University, 1989.

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

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

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

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



(a) up meshes (b) low meshes

Figure 3: The mesh angle quality and spacing conformity.

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

[10] 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.

[11] Miller, G. L., Talmor, D., Teng, S. H., and Walkington, N. On the radius–edge condition in the control volume method. *SIAM J. on Numerical Analysis* (1998). accepted and to appear.

[12] 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.

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

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

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