SIMPLICIAL MESH GENERATION AND MAINTENANCE

HAI SHAO

Abstract. Following a brief introduction to the general meshing problem, we focus our survey on the formulation and methodology of *simplicial*, or *unstructured*, meshing problems, including state-of-the-art mesh generation and mesh maintenance (adaptation).

1. Introduction

Meshing is a geometric tool that has wide scientific and engineering application in areas like computer aided design (CAD), geographic information systems (GIS), various mathematical modeling and a lot more. For example, simulation of physical processes and phenomena can be modeled by a system of partial differential equations (PDEs) which in turn can be solved using numerical methods such as finite difference method, finite element methods and finite volume methods. All these methods require that a fine enough mesh be generated on the domain where the PDEs are defined, so that the differential equations can be approximated by a *finite* system of algebraic (linear or nonlinear) equations which can be routinely solved in the context of numerical analysis[30].

Two types of issues concerning the meshing are frequently raised but hardly solved. On one hand, the users, such as the engineers, may ask for good meshes to make their simulation work; while on the other hand, the designers, such as the computational geometers, may need standardized specifications of the meshing problems to define their own problems and to make their algorithms (software packages) work for as many cases of applications as possible. The good news is that both groups start to work on these problems together as seen in the recent events in the meshing community[21], even though a lot of progress still needs to be made.

Generally speaking, a mesh is a finite, fine, and possibly approximate representation of a spatial domain, and more specifically, it consists of a set of non-overlapping *elements* whose union conforms the domain. Meshes are classified into two categories[22, 49, 50]: a) *structured meshes*, where all elements are hypercube-like (quadrilaterals in 2D and hexahedra in 3D), and b) *unstructured meshes*, which we call *simplicial meshes* in the paper and are sometimes referred to as *triangulations*, where all elements are simplices. Simplicial meshes have become noticeably more and more dominant in the meshing community mainly because they can automatically incorporate

---

Date: April 30, 1997.

with arbitrary shapes of the domain of interest[26, 50]. Before we look into different ways of formulating and solving the simplicial (unstructured) meshing problems in the following sections, we first take a brief look at the structured meshing problems.

Paradigm for structured mesh generation. First the domain is decomposed into a set of covering blocks, each mapped to a hypercube region defined by a chosen (Cartesian or curvilinear) coordinate system. The mesh of each block, called the submesh, is computed as the preimage of the regular grid structure, axis aligned d-dimensional array of grid points, of the hypercube. The final mesh can be obtained by patching the submeshes at common interfaces of the blocks. The mapping for each block can usually be obtained by solving either an elliptic or a hyperbolic PDE system defined on the hypercube[49, 22]. This is the so-called *structured mesh* generation since the local structure of connectivity is invariant when translating from one mesh point to another.

When solving PDE systems, the regularity of structured grids may lead to nice matrix properties of the approximating algebraic equations, which allows fast linear or nonlinear solvers to be used[22]. However, the pitfalls of the paradigm are a) decomposing the domain Ω into blocks at the beginning and merging the meshes of all the blocks in the end can be done only heuristically, and may be impossible for domains with complicated shapes[7, 50]. The current trend in the meshing community has been toward unstructured meshes[7].

2. Notations, definitions and assumptions

All meshes mentioned in the rest of the paper will be simplicial meshes unless otherwise specified. We will use Ω to denote the domain of interest and d its spatial dimensionality. We will assume that Ω has *one* connected component, since meshes for different connected components can be computed independently. The following terms will be frequently mentioned throughout the paper.

Simplicial complexes[20]**.** A *k-simplex*, σ^k, is the convex hull of a set T of $k+1$ affinely independent points. The *dimension* of a σ^k is $\dim\sigma^k = k$. A simplex by default is a d-simplex. A finite collection of simplices, $\mathcal{K}$, is a *simplicial complex* if the faces of every simplex in $\mathcal{K}$ also belong to $\mathcal{K}$, and the intersection of two simplices in $\mathcal{K}$ is either empty or a face of both. The dimension of $\mathcal{K}$ is $\dim\mathcal{K} = \max_{\sigma\in\mathcal{K}} \dim\sigma$. The *underlying space* of $\mathcal{K}$ is $\bigcup\mathcal{K} = \bigcup_{\sigma\in\mathcal{K}} \sigma$.

Delaunay triangulation and Voronoi diagrams[23]**.** Let $P = \{p_1, \dots, p_n\}$ be a set of points in $\mathbb{R}^d$. The Delaunay triangulation and Voronoi diagram can be defined as dual structures of the point set P.

- For a nonempty subset Q of P, the *Voronoi cell* of Q, $V(Q)$, is the set of all points of $\mathbb{R}^d$ equidistant from all sites in Q and closer to every point of Q than

to any other points not in Q. The *Voronoi diagram* $\mathcal{V}(P)$ is the collection of all nonempty Voronoi cells $V(Q)$, for Q a subset of P.
- If $Q \subset P$ and $V(Q)$ is nonempty, then the *Delaunay cell* $D(Q)$ is the convex hull of Q. The *Delaunay triangulation* $\mathcal{D}(P)$ is the collection of Delaunay cells $D(Q)$ for Q varying over subsets of Q with $V(Q)$ nonempty.

Suppose the points in P are in general positions, *i.e.*, a) no $d+1$ points are affinely dependent, and b) no $d+2$ points are co-spherical. Then $\mathcal{D}(P)$ is a simplicial complex that satisfy the *Delaunay criterion* – that the interior of the circumsphere of a d-simplex in $\mathcal{D}(P)$ contains no points in P.

3. Problem formulations

A meshing problem of a domain Ω consists of two parts in general: mesh generation, mesh maintenance. Each problem is formulated by specifying its preconditions and postconditions.

3.1. **Mesh generation.**

This is to construct a mesh out of some kind of skeleton of the domain Ω.

3.1.1. *Preconditions.* To generate a mesh we need some kind of description of Ω, such as the boundary faces of the domain and the interfaces between different subdomains. A domain on a plane can be generally described by a *planar straight line graphs* (PSLG)[7, 13, 47] for simple polygons, polygons with holes, point sets, *etc.* However, except for some special cases[1], such domain description becomes exponentially complicated in terms of spatial dimensionality. To generalize the notion of PSLG in $\mathbb{R}^d$, we define a *crossing-free flat set* (CFFS)[19], or a *piecewise linear system* as in [40], simply as a set of k-flats, $0 \le k \le (d-1)$, such that the intersection of any two flats is either empty or a common subface of both. A CFFS can be used to describe the boundary faces and interfaces of a domain in $\mathbb{R}^d$. Additional information, such as the orientation of each flat, may be needed to determine the inside or the outside of the domain at an arbitrary location. A CSSF Φ can be nicely described by an *incidence graph* defined as follows.

Definition 3.1 (Edelsbrunner [15]). *A* **incidence graph** $\mathcal{IG}(\Phi)$ *of a face set* Φ *in* $\mathbb{R}^d$ *is a graph where each vertex* $v(\phi)$ *of* $I(\Phi)$ *is associated with a* k*-face* ϕ *of* Φ*. An edge* $(v(\phi_1), v(\phi_2))$ *occurs in* $\mathcal{IG}(\Phi)$ *if and only if* ϕ_1 *is one dimension higher(lower) than* ϕ_2 *and* ϕ_1 *is a superface(subface) of* ϕ_2*.*

---

[1] such as magnetic resonance imaging (MRI), a sequence of cross sections of a 3D objects can be made to provide a set of contours lines in 2D[2].

3.1.2. *Postconditions.* A mesh $\mathcal{M}$ of a domain Ω, described by a CFFS Φ_Ω, is a simplicial complex that *conforms* Φ_Ω in a sense that each flat of Φ_Ω is preserved (and very likely subdivided) in $\mathcal{M}$. Therefore, mesh generation is also referred to as *conforming triangulation*[19]. In some cases, we may have a *constrained triangulation problem* where some flats of Φ_Ω are not allowed to be subdivided. For example in $\mathbb{R}^2$, a *constrained Delaunay triangulation* of a PSLG preserves all the edges of the graph, and can be constructed in $O(n \log n)$ time[13]. In general, constrained triangulation is conceivably much harder than conformal triangulation and will be exempted from discussion from now on.

A *measure set*, denoted by μ, can be used to characterize a mesh, *e.g.*, its quality.

- Global measures
  - Worst *aspect ratio*. The *aspect ratio* of a convex polytope is the ratio between the largest and the smallest separations of parallel supporting hyperplanes[5].
  - Maximum k-angle, $0 \le k \le (d-2)$. A k-face in a d-simplex defines a *d-angle*, and it is measured by the fraction of which a small $(d-k-1)$-sphere centered on and perpendicular to the k-face lies interior to the d-simplex[5]. A 0-angle is a solid angle, and a $(d-2)$-angle is a dihedral.
  - Sum of Euclidean measures of all k-faces, $0 \le k \le d-1$. When $d = 2$, its the weight of the triangulation.
  - Mesh size – number of k-faces, $0 \le k \le d$, $k = 0$ for number of mesh points, and $k = d$ for number of elements.
  - Admission of Delaunay criterion.
- Local measures
  - Density of mesh points.
  - Anisotropy of connectivity between mesh points, characterized by a $d \times d$ *symmetric positive-definite* matrix[12].

Suppose the mesh is used to solve a PDE system defined on Ω. Then most of these quantities will determine the properties of the system of algebraic equations approximating the PDE system. For example, large value of the worst aspect ratio will lead to ill-conditioning of the system of algebraic equations. Due to the perpendicular straight line duality between the Delaunay triangulation and the Voronoi diagram, the Delaunay property of the mesh is strongly desired when using the finite volume method, so that the integral in the dual boxes (Voronoi cells) can be computed properly.

The way that the quantities of the measure set are specified determines what the output mesh should be like. Unfortunately, for a given input Φ_Ω, choices on mesh specifications can be very limited, some of which may even lead to open problems up to the present.

3.2. **Mesh maintenance.** G enerally speaking, mesh maintenance is to update the mesh when the preconditions (Φ_Ω) and/or postconditions (μ) for mesh generation alter. Therefore, we can say the the precondition of a mesh maintenance problem is the original mesh $\mathcal{M}(\Phi_\Omega, \mu)$ and the postcondition will be the new objective mesh $\mathcal{M}(\Phi_{\Omega'}, \mu')$. Notice that the domain its can also change in this problem setting. Of course the objective mesh can be generated from scratch just like in a mesh generation problem. However, the amount of work can always be significantly reduced by taking advantage of the information the original mesh provides. The mesh maintenance procedure is also called adaptation, and the combination of mesh generation and adaptation is often referred to as *adaptive mesh generation.* For example in numerical PDE, when part of the domain needs too be refined for progressive accuracy of the numerical solution, the local density of the mesh will need to change.

4. Mesh Generation Methodology

There are many existing methods to generate meshes in fixed or arbitrary dimension which meets certain mesh specifications. Flavors of these methods may vary in wide range from very theoretical, the computer geometer's type, to very practical, the engineer's type. In this section, we will start with describing a general paradigm nad then proceed with reviewing individual mesh generation methods – algorithms, mesh features, limits and advantages, and so on.

4.1. **A general paradigm.**

A mesh generation method for a CFFS Φ_Ω describing a domain Ω in general introduces some additional points, called the *Steiner points*, and triangulates both the input Φ_Ω and the Steiner points. Steiner points and Delaunay trianguation are two commonly used techniques in mesh generation.

4.1.1. *Steiner points.* Steiner points are the vertices in the mesh that are not the vertices of input. Steiner points are introduced to help conform the faces in Φ_Ω and to approach certain mesh specifications, such as higher density, smaller aspect ratio, and so on. Sometimes, Steiner points can help reduce the size of triangulation in higher dimensions.

To the author, Steiner faces may be conceivably useful for mesh generation in higher dimensions. However they have not been mentioned in the current literature.

4.1.2. *Delaunay triangulation.* Delaunay triangulation has many desirable geometric and combinatorial properties that we could always use to generate good meshes. For example, two dimensional Delaunay triangulation of a given point set optimizes various quality measures including the smallest angle, the larger circumscribing circle, the largest minimum enclosing circle, and the integral of the gradient squares[20, 45]. Moreover, the Delaunay triangulation contains the nearest-neighbors graph and the minimal spanning tree[39].

On the other hand, the Delaunay criterion of a triangulation can be satisfied without much overhead. Edelsbrunner and Shah[17] have shown that it takes at most $O(n \log n + n^{\lceil d/2 \rceil})$ number of edge flippings to transform an arbitrary triangulation to a Delaunay triangulation. Construction of Delaunay triangulations (and Voronoi diagrams) for a point set is well studied in computational geometry, and the algorithmic complexity is known to be at most as difficult as that of construction convex hulls in one higher dimension[23]. Commonly used techniques for lower dimensional Delaunay triangulation are divide-and-conquer, plane-sweep and randomized incremental algorithms[23, 32], all of which have similar bounds on running time (or expected running time) of $O(n \log n)$ in $\mathbb{R}^2$ and $O(n^2)$ in $\mathbb{R}^3$, and can be generalized to higher dimensions. However the incremental Delaunay construction is known to be extremely useful for mesh generation, especially adaptive mesh generation. Point insertion to a Delaunay triangulation is considered as a basic operation, which is even named as *Delaunay kernel* by Freitag, Borouchaki and George in [26], in the paradigm of Delaunay mesh generation. Barber, Dopkin and Huhdanpaa[4] have provided the public state-of-the-art implementation of Delaunay triangulation (including convex hull construction) in arbitrary dimensions, known as the `qhull` program.

4.2. **2^d-tree method.** This is one of the earliest proposed and most popularly used methods in mesh generation. We give the following definition of a 2^d-*tree* analogous to those of a *quadtree*[7] in $\mathbb{R}^2$ and an *octree*[41] in $\mathbb{R}^3$.

Definition 4.1. *A* 2^d**-tree** *is a recursive partition of a region into axis-aligned hypercubes, called* **cells**. *One hypercube, the root, covers the entire region. A cell can be divided into* 2^d *child cells by splitting it in halves in all the d dimension. A* 2^d*-tree is said to be* **balanced** *if any edge of all leaf cells is split at most once.*

First, a fine enough 2^d-tree is constructed over the domain Ω described by Φ_Ω such that no leaf cell is *crowded*[7, 41]. Then the leaf cells, either intersecting Φ_Ω or not, are triangulated separately. In this case, the vertices of the leaf cells are used as Steiner points.

In general, a 2^d-tree triangulation produces near optimal number of Steiner points with bounded aspect ratio. Denote $\mathcal{T}$ a 2^d-tree mesh, and $A(\mathcal{T})$ the maximum aspect ratio of all simplices in $\mathcal{T}$.

Theorem 4.2 (Bern, Eppstein and Gilbert [8]). *Let X be a point set in $\mathbb{R}^d$ and $\mathcal{T}$ be a triangulation that respects X. Then there exists a 2^d-tree triangulation that respects X and has bounded aspect ratio and size $O(|\mathcal{T}| \log A(\mathcal{T}))$.*

Moreover, 2^d-tree is an optimal way to reduce the number of simplices of the Delaunay triangulation for a given set of points.

Theorem 4.3 (Bern, Eppstein and Gilbert [8]). *Let X be a point set in $\mathbb{R}^d$. Then there exist a set $Y \supseteq X$ of $O(|X|)$ points for which the d-dimensional Delaunay triangulation contains only $O(|X|)$ simplices.*

One of the problems that the 2^d-tree method may cause is that there may too many right angles in the output mesh. In $\mathbb{R}^2$, the quadtree algorithm by Bern, Eppstein and

Gilbert[8] is able to generate meshes with all angles of the output triangles greater than 20°. By strengthening the balance condition in the quadtrees and replacing quadtree squares by tiles with projections and indentations, Bern et al. can guarantee that all angles measure between 36° and 80°[7]. Neugebauer and Diekmann[43] used a rhomboid structure of the quadtree to generate meshes with angle bounds between 30° and 90°. Conti, Tomizawa and Yoshii[14] used the 2-4-8-*tree*, a variation of the octree where a cell is not always partitioned in all three directions, to generate oriented meshes.

4.3. **Sphere packing method**[40]**.** A *spacing function* f indicating the density of mesh is a positive and α-Lipschitz function defined over the domain Ω, where the Lipschitz condition implies $|f(y) - f(x)| \leq \alpha\|y - x\|$ for all points x and y in the domain. An *f-sphere* at a point $p \in \Omega$ is defined as a sphere with center p and radius $f(p)$. A set of points is a *sphere-packing* with respect to the spacing function f if the interiors of their f-spheres do not intersect. It is *maximal* if P is a maximal such set. A Delaunay triangulation of a maximal sphere-packing can be computed and will have bounded *radius-edge ratio*, the ratio between the radius of the circumscribing sphere and the shortest edge.

The idea for using sphere-packing to construct Delaunay mesh for a CFFS Φ_Ω is first to pack spheres around all faces of Φ_Ω, in the order of increasing dimension, and then to pack spheres around Steiner points to make it maximal. The Delaunay triangulation of vertices in Φ_Ω and the Steiner points will conform all faces in Φ_Ω. A sphere-packing normally generates good quality meshes in 2D but may generates *slivers*, tetrahedra with four vertices lying close to the equator of a sphere, in 3D[39].

4.3.1. *Oversampling and filtering*[39, 40]. It's not obvious how to generate a maximal packing for a given spacing function. In some cases, the maximality is weakened such that the sphere packing only needs to be *ϵ-maxima* ($\epsilon > 0$) where each ϵf-ball contains a point that can not be added to the sphere packing. *Oversampling* samples a point for each ϵf-ball. A *conflict graph* is constructed where the vertices are the oversampling points, and an edge is drawn between two points if interior of their f-spheres intersect. Then *filtering* is to find a maximal independent set. However the quality of the resulting mesh has not been evaluated versus other mesh generators.

4.3.2. *A 2D Delaunay refinement method.* This method by Ruppert[47] generates good quality Delaunay meshes for PSLGs in $\mathbb{R}^2$, and is almost the simplest 2D meshing algorithm in terms of implementation. The method is mainly based on the randomized incremental Delaunay triangulation algorithm by Guibas, Knuth and Sharir[32], and a refinement algorithm by Chew for used for constructing quality constrained Delaunay triangulations[13]. It first generates a Delaunay triangulation for the vertices of the input PSLG and then successively inserts Steiner points and updates the Delaunay triangulation until all the triangles are well-shaped and conform the original PSLG. The edges that form the edges of the PSLG are referred to as *segments* to be distinguished from the other edges of the Delaunay triangulation. The circumcenter of a bad-shaped triangle is added as a Steiner point if it does not lie in the diametral circle of any segments in the current Delaunay triangulation. Otherwise the midpoint of such segment is added.

A directed acyclic graph (DAG) is used to store the history of the Delaunay triangulation[17, 32, 45], such that the running time for point insertion to the Delaunay triangulation is proportional to the logarithm of the size of current Delaunay triangulation on average. And the expected running time for the algorithm is $O((n+k)\log(n+k))$, where n is the size of the PSLG and k is the total number Steiner points. It is shown by Ruppert[47] that the size of the Steiner points are near optimal:

Theorem 4.4 (Ruppert[47])**.** *Given* $\alpha \leq 20^\circ$*, and input* X*, suppose* $\mathcal{T}$ *is any triangulation of* X *with minimum angle bound* α*. There is a constant* C_α *such that if* $\mathcal{T}$ *has* N *triangles, then the Delaunay refinement triangulation* $\mathcal{T}_D$ *has* $N_D \leq C_\alpha \cdot N$ *triangles. Letting* $\mathcal{T}$ *be the triangulation with fewest possible triangles shows that* $\mathcal{T}_D$ *is within a factor* C_α *of optimal.*

Ruppert's Delaunay refinement method can be actually deduced from an (iterative) sphere-packing if we use the notion *Voronoi maximality*[40], another weakened maximality like the ϵ-maximality used in the oversampling and filtering method. We define a sphere-packing to be Voronoi maximal if no circumcenter of a Delaunay ball can be added. How to extend the Delaunay refinement method to higher dimensions is not known. The main difficulty in 3D lies in the fact that the center of minimum containing sphere of a triangle may not stay inside the triangle.

4.3.3. *A 2D disk packing method for nonobtuse triangulation.* The algorithm by Bern, Mitchell and Ruppert[9] consists of two stages. The first stage packs the domain with non-overlapping disks, tangent to each other and to sides of the domain. The algorithm then adds edges (radii) between centers of disks and points of tangency on their boundaries, thereby dividing the domain into small polygons. The second stage triangulates the small polygons using Steiner points located only interior to the polygons or on the domain boundary. It can be shown that

Theorem 4.5 (Bern, Mitchell and Ruppert[9])**.** *An* n*-vertex polygonal region can be triangulated with* $O(n)$ *right triangles in time* $On\log n$ *for simple polygons and* $O(n\log^2 n)$ *for polygons with holes.*

The result is significant compared to the previous polynomial size ($O(n^2)$[6]) nonobtuse triangulations

4.4. **Advancing-front method.** This method by Frey, Borouchaki and George[26] is more of an engineering approach. It generates high-quality meshes for many real examples such as geological objects and space surrounding an aircraft. However, no bound on the size or the worst aspect ratio of advancing-front meshes has been proven.

The method first generates a background mesh – a conformal Delaunay triangulation of the CFFS Φ_Ω describing the domain Ω, with Steiner points added only on the faces of Φ_Ω. The set of simplicial faces of the background mesh that form the faces in Φ_Ω is called the initial *front*, denoted by $\mathcal{F}$, A *size field* is defined at a point $p \in \Omega$ as the average length of the edges that is connected with p if p is a vertex of $\mathcal{F}$ or as the linear interpolation of the size field values at all vertices of the simplex containing p. And the distance between two points in Ω is normalized by the size field. The algorithm then successively adds Steiner points which departure from the current font into the domain, updates the Delaunay triangulation and the redefines the front, until the front is empty. At each stage, the frontier separates a collection of well-shaped simplices (toward the boundary of the domain) and a collection of bad-shaped simplices (toward the kernel of th domain), and the Steiner points are generated in the following way:

- For each simplicial face $f \in \mathcal{F}$ find a point that is equidistant from all vertices of f in the sense of normalized distance and with normalized distance α, an adjustable parameter. Let N denote the collection of such points for all faces in $\mathcal{F}$.
- Filter N such no points are too close to the frontier and two points are too close to each other. The filtered points are added as Steiner points.

Throughout the procedure, the well-shaped simplices as well as the bad-shaped ones always form a conformal Delaunay triangulation of the domain. Therefore, it is also called the *advancing-front Delaunay triangulation*, whereas other advancing-front mesh generation is devised by using different kind of heuristics to march the front[36].

5. MESH MAINTENANCE METHODOLOGY

Efficient mesh maintenance is demanded when adaptation becomes a frequent event as in solving numerical PDEs. The mesh needs to be refined or coarsened (or derefined) when the density requirement changes. Mesh smoothing can improve the aspect ratio and eliminate too large or small angles of the elements. Most of the time, these changes only need to happen locally. Therefore, avoiding global behavior in these situations is important to make the methods efficient.

5.1. **Refinement.**

In general, the mesh can be refined by placing more Steiner points inside the domain. A simple and commonly used method is *bisection*[35, 44, 46]. It uses the midpoint of some "long" edges as Steiner points and connect them together and with existing vertices to form a valid triangulation. If the sphere packing method[40] is used to generate the original mesh, then the local refinement can be achieved by decrease the spacing function vale locally while still keeping its α-Lipschitz condition.

5.2. **Coarsening.**

The general method used for coarsening simplicial meshes is that of picking some subset of the original mesh points and triangulating that set to form the coarsening. A commonly used technique is that of the *maximal independent set* (MIS). The mesh is viewed as a graph by its 1-skeleton. An MIS of the graph can be used to form a coarsening. However, repeatedly application of the MIS technique to construct a coarsening hierarchy can degrade the aspect ratio of the meshes[40].

Again, we can use the sphere packing technique[40], where the main idea is to construct a new spacing function as desired corresponding to sparser density of the mesh over the domain. Then a conflict graph can be constructed over the original mesh points using the new spacing function. The the coarser mesh is computed as the Delaunay triangulation of a MIS of the conflict graph.

5.3. **Smoothing.**

Mesh smoothing moves the Steiner points of the mesh while preserving its overall topology. A commonly used technique, *Laplacian smoothing*, sweeps over the mesh, successively moving each point to the *centroid* of its neighbors. This technique has been practically used because of its simplicity. However it is not directly connected to any specific mesh quality criterion; moreover it may not remove slivers from three dimensional meshes, and the result may not even remain a valid triangulation[1, 7].

A second technique, called *mesh relaxation* by Frey and Field[27], flips edges to regularize degrees of vertices. An edge of the triangulation is flipped if the flipping causes sum of its endpoints' degrees decreases by more than two.

Freitag, Jones and Plassmann[24] proposed an alternative to Laplacian smoothing. Rather than using the centroid, their optimization-based method computes for each Steiner point a new placement that maximizes the minimum angle in adjacent triangles. An iterative steepest-descent algorithm is used to solve the optimal placement problem. However the algorithm has not been proved to be correct.

What is the most intriguing and seems to be the most promising method in mesh smoothing is the *quasiconvex programming* by Amenta, Bern and Eppsitein[1]. A quasiconvex programming problem is a generalized linear programming (GLP) problem where all the level sets of the objective functions are convex. More formally, a *nested convex family* is defined as a map $\kappa(t)$ from $\mathbb{R}$ to convex sets in $\mathbb{R}^d$ such that if $a < b$ then $\kappa(a) \subset \kappa(b)$. For a simplex σ of the mesh, a nested convex family can be constructed as $\kappa_\sigma(t) = \{x \mid q(\sigma(x)) \leq t\}$, where $\sigma(p)$ indicates the simplex formed by moving the Steiner point to the position x, and $q(\sigma)$ is a quality measure of σ. Then finding the optimal Steiner point placement is equivalent to finding the optimal quality bound that allows a feasible placement. The set of quality measures that can be used to construct nested convex families are: 1) volume, 2) altitude, 3) containing sphere, 4) dihedrals, 5) solid angles.

Theorem 5.1 (Amenta, Bern and Eppstein[1]). *In any constant dimensions, the Steiner point placement optimizing each of the above criteria, or a weighted combination of criteria, can be computed in linear time by quasiconvex programming.*

6. Conclusion

We have tried to gather together some scattered but up-to-date results of the various simplicial mesh generation and maintenance problems for our future study in this area.

References

[1] N. Amenta, M. Bern and D. Eppstein, "Optimal point placement for mesh smoothing," *Proc. 7th ACM-SIAM Symp. Disc. Algo.*, 528-537, 1997.

[2] C. L. Bajaj, E. J. Coyle and K. N. Lin, "Surface and 3D triangular meshes from planar cross sections", *Proc. 5th International Meshing Roundtable*, 169-178, Sandia National Laboratories, 1996.

[3] R. E. Bank and J. Xu, "An Algorithm for Coarsening Unstructured Meshes," *Numerische Mathematik* 73, 1-36, 1996.

[4] C. B. Barber, D. P. Dobkin and H. Huhdanpaa, "The Quickhull algorithm for convex hulls," *ACM Trans. on Mathematical Software*, Dec 1996.

[5] M. Bern, P. Chew, D. Eppstein and J. Ruppert, "Dihedral bounds for mesh generation in high dimensions," *Proc. 6th ACM-SIAM Symp. Disc. Algo.*, 189-196, 1995.

[6] M. Bern and D. Eppstein, "Polynomial-size nonobtuse triangulation of plygons," *Int. J. Comp. Geometry and Applications*, 2, 241-255, 1992.
triangulation," *Computing in Euclidean geometry*, 47-123, D. Z. Du and F. Hwang ed, World Scientific, Singapore, 1995.

[7] M. Bern and D. Eppstein, "Mesh generation and optimal triangulation," *Computing in Euclidean geometry*, 47-123, D. Z. Du and F. Hwang ed, World Scientific, Singapore, 1995.

[8] M. Bern, D. Eppstein and J. R. Gilbert, "Provably good mesh generation," *FOCS90*, 231-241, IEEE, 1990.

[9] M. Bern, S. Mitchell and J. Ruppert, "Linear-size nonobtuse triangulation of polygons," *Proc. 10th ACM Symp. Comput. Geom.*, 221-230, 1994.

[10] J. D. Boissonnat, A. Cérézo, O. Devillers and M. Teillaud, "Output sensitive construction of the 3D Delaunay triangulation of constrained sets of points," *Technical Report*, Institut National de Recherche en Informatique et Automatique – France, Number RR-1415, 1991.

[11] H. Borouchaki, F. Hecht, E. Saltel and P. L. George, "Reasonably efficient Delaunay based mesh generator in 3 dimensions," *Proceedings 4th International Meshing Roundtable*, 3-14, Sandia National Laboratories, 1995.

[12] F. J. Bossen and P. S. Heckbert, "A pliant method for anisotropic mesh generation", *Proc. 5th International Meshing Roundtable*, 63-76, Sandia National Laboratories, 1996.

[13] L. P. Chew, "Guaranteed-quality triangular meshes," *Technical Report* 89-983, Cornell University, 1989

[14] P. Conti, M. Tomizawa and A. Yoshii, "Generation of oriented three-dimensional Delaunay grids suitable for the control volume integration method,", *International Journal for Numerical Methods in Engineering*, 37, 3211-3227, 1994.

[15] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer-Verlag, 1987.

[16] H. Edelsbrunner, "Algebraic decomposition of non-convex polyhedra," *Technical Report*, Department of Computer Science, University of Illinois at Urbana-Champaign, Number 1928, 1995.

[17] H. Edelsbrunner and N. R. Shah, "Incremental topological flipping works for regular triangulations," *Proc. 8th ACM Symp. Comput. Geom.*, 43-52, 1992.

[18] H. Edelsbrunner and N. R. Shah, "Triangulating topological spaces," *Proc. 10th ACM Symp. Comput. Geom.*, 285-292, 1994.

[19] H. Edelsbrunner and T. S. Tan, "An upper bound for conforming Delaunay triangulations," *Proc. 8th ACM Symp. Comput. Geom.*, 53-62, 1992.

[20] H. Edelsbrunner and R. Waupotitsch, "Adaptive simplicial grids from cross-sections of monotone complexes," *Technical Report*, Department of Computer Science, University of Illinois at Urbana-Champaign, Number 1864, 1994.

[21] D. Field, "Give me a good mesh," *Proc. 5th International Meshing Roundtable*, pp.3, Sandia National Laboratories, 1996.

[22] M. Filipiak, "Mesh generation," *Technology Watch Report*, Edinburgh Parallel Computing Centre, 1996

[23] S. Fortune, "Voronoi diagrams and Delaunay triangulations," *Computing in Euclidean geometry*, 225-265, D. Z. Du and F. Hwang ed, World Scientific, Singapore, 1995.

[24] L. A. Freitag, M. T. Jones and P. E. Plassmann, "An efficient parallel algorithm for mesh smoothing," *Proc. 4th International Meshing Roundtable*, 47-58 Sandia National Laboratories, 1995.

[25] L. A. Freitag and C. Ollivier-Gooch, "A comparison of tetrahedral mesh improvement techniques," *Proc. 5th International Meshing Roundtable*, 87-106, Sandia National Laboratories, 1996.

[26] P. J. Frey, H. Borouchaki and P. L. George. "Delaunay tetrahedralization using an advancing-front approach", *Proc. 5th International Meshing Roundtable*, 31-46, Sandia National Laboratories, 1996.

[27] W. H. Frey and D. A. Field, "Mesh relaxation: a new technique fro improving triangulations," *International Journal for Numerical Methods in Engineering*, 31, 1121-1133, 1991.

[28] J. Galtier and P. L. George, "Prepartitioning as a way to mesh subdomains in parallel," *Proc. 5th International Meshing Roundtable*, 107-122, Sandia National Laboratories, 1996.

[29] R. Garimella, and M. S. Shephard, "Tetrahedral mesh generation with multiple elements through the thickness," *Proceedings 4th International Meshing Roundtable*, 321-333, Sandia National Laboratories, 1995.

[30] P. L. George, *Automatic Mesh Generation – Application to Finite Element Methods*, John Wiley & Sons, 1991.

[31] C. S. Gitlin and C. R. Johnson, "MeshView: a tool for exploring 3D unstructured tetrahedral meshes," *Proc. 5th International Meshing Roundtable*, 333-345, Sandia National Laboratories, 1996.

[32] L. J. Guibas, D. E. Knuth and M. Sharir, "Randomized incremental construction of Delaunay and Voronoi diagrams," *Algorithmica*, 7, 381-413, 1992.

[33] B. Joe, "Construction of three-dimensional Delaunay triangulation using local transformations," *Computer Aided Geometric Design*, 8, 123-142, North-Holland, 1991.

[34] M. T. Jones and P. E. Plassmann, "Parallel algorithms for adaptive mesh refinement," *SIAM J. Sci. Comput.*, 18(3), 686-708, 1997.

[35] A. Liu and B. Joe, "Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision," *Mathematics of Computation*, 65(215), 1183-1200, 1996.

[36] D. J. Marvriplis, "Unstructured mesh generation and adaptivity," *ICASE Report*, NASA Langley Research Center, Number 95-26, 1995.

[37] T. Midtbø, "Spatial modeling by Delaunay networks of two and three dimensions," *Dr.Ing. Thesis*, Department of Surveying and Mapping, Norwegian University of Science and Technology, 1993.

[38] G. L. Miller, D. Talmor and S. H. Teng, "Optimal good-aspect-ratio coarsening for unstructured meshes," *Proc. 7th ACM-SIAM Symp. Disc. Algo.*, 538-547, 1997.

[39] G. L. Miller, D. Talmor, S. H. Teng and N. Walkington, "A Delaunay based numerical method for three dimensions: generation, Formulation, and Partition," *FOCS95*, 683-692, IEEE, 1995.

[40] G. L. Miller, D. Talmor, S. H. Teng, N. Walkington and H. Wang, "Control volume meshes using sphere packing: generation, refinement and coarsening," *Proc. 5th International Meshing Roundtable*, 47-62, Sandia National Laboratories, 1996.

[41] S. Mitchell and S. Vavasis, "Quality mesh generation in three dimensions," *Proc. 8th ACM Symp. Comput. Geom.*, 212-221, 1992.

[42] S. Mitchell and S. Vavasis, "An aspect ratio bound for triangulating a d-grid cut by a hyperplane," *Proc. 12th ACM Symp. Comput. Geom.*, 48-57, 1996

[43] F. Neugebauer and R. Diekmann, "Improved mesh generation: not simple but good," *Proc. 5th International Meshing Roundtable*, 257-272, Sandia National Laboratories, 1996.

[44] A. Plaza and G. F. Carey, "About local refinement of tetrahedral grids based on bisection," *Proc. 5th International Meshing Roundtable*, 123-136, Sandia National Laboratories, 1996.

[45] T. T. Rajan, "Optimality of the Delaunay triangulation in $\mathbb{R}^d$," *Proc. 7th ACM Symp. Comput. Geom.*, 357-363, 1991.

[46] M. C. Rivara and P. Inostroza, "A discussion on mixed (longest-side midpoint insertion) Delaunay techniques for the triangulation refinement problem," *Proceedings 4th International Meshing Roundtable*, Sandia National Laboratories, 1995.

[47] J. Ruppert, "A Delaunay refinement algorithm for quality 2-dimensional mesh generation," *J. Algo.*, 18(3), 548-585, 1995.

[48] J. A. Schmidt, "Mesh generation with applications in computational electrophysiology," *Ph.D. Thesis*, Department of Electrical Engineering, Duke University, 1993.

[49] J. F. Thompson and N. P. Weatherill, "Structured and unstructured grid generation," *High-Performance Computing in Biomedical Research*, 63-111, CRC Press, 1993.

[50] N. P. Weatherill and O. Hassan, "Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints," *International Journal for Numerical Methods in Engineering*, 37, 2005-2039, John Wiley & Sons, 1994

SIMPLICIAL MESH GENERATION AND MAINTENANCE

HAI SHAO

ABSTRACT. Following a brief introduction to the general meshing problem, we focus our survey on the formulation and methodology of *simplicial*, or *unstructured*, meshing problems, including state-of-the-art mesh generation and mesh maintenance (adaptation).

1. INTRODUCTION

Meshing is a geometric tool that has wide scientific and engineering application in areas like computer aided design (CAD), geographic information systems (GIS), various mathematical modeling, and a lot more. For example, simulation of physical processes and phenomena can be modeled by a system of partial differential equations (PDEs), which in turn can be solved using numerical methods, such as finite difference method, finite element methods, and finite volume methods. All these methods require that a fine enough mesh be generated on the domain where the PDEs are defined, so that the differential equations can be approximated by a *finite* system of algebraic (linear or nonlinear) equations which can be routinely solved in the context of numerical analysis [30].

Two types of issues concerning the meshing are frequently raised but hardly solved. On one hand, the users, such as the engineers, may ask for good meshes to make their simulation work; while on the other hand, the designers, such as the computational geometers, may need standardized specifications of the meshing problems to define their own problems and to make their algorithms (software packages) work for as many cases of applications as possible. The good news is that both groups start to work on these problems together, as seen in the recent events in the meshing community [21], even though a lot of progress still needs to be made.

Generally speaking, a mesh is a finite, fine, and possibly approximate representation of a spatial domain, and more specifically, it consists of a set of non-overlapping *elements* whose union conforms the domain. Meshes are classified into two categories [22, 51, 52]: a) *structured meshes*, where all elements are on a d-dimensional grid (quadrilaterals in 2D and hexahedra in 3D), and b) *unstructured meshes*, which we call *simplicial meshes* in the paper and are sometimes referred to as *triangulations*, where all elements are simplices. Simplicial meshes have become noticeably more and more dominant in the meshing community mainly because they can automatically incorporate with arbitrary shapes of the domain of interest [26, 52]. Before we look into different ways of formulating and solving the simplicial (unstructured) meshing problems in the following sections, we first take a brief look at the structured meshing problems.

Date: April 30, 1997.

Paradigm for structured mesh generation. First the domain is decomposed into a set of covering blocks, each mapped to a hypercube region defined by a chosen (Cartesian or curvilinear) coordinate system. The mesh of each block, called the submesh, is computed as the preimage of the regular grid structure, axis aligned d-dimensional array of grid points, of the hypercube. The final mesh can be obtained by patching the submeshes at common interfaces of the blocks. The mapping for each block can usually be obtained by solving either an elliptic or a hyperbolic PDE system defined on the hypercube [51, 22]. This is the so-called *structured mesh* generation since the local structure of connectivity is invariant when translating from one mesh point to another.

When solving PDE systems, the regularity of structured grids may lead to nice matrix properties of the approximating algebraic equations, which allows fast linear or nonlinear solvers to be used [22]. However, the pitfalls of the paradigm are a) decomposing the domain into blocks at the beginning and merging the meshes of all the blocks in the end can be done only heuristically, and may be impossible for domains with complicated shapes [7, 52]. The current trend in the meshing community has been toward unstructured meshes [7].

2. NOTATIONS, DEFINITIONS, AND ASSUMPTIONS

We first define *simplices* and *simplicial complexes* in $\mathbb{R}^d$ as follows.

Definition 2.1 (Edelsbrunner [20]). *A* **k-simplex**, *σ^k, $0 \leq k \leq d$, is the convex hull of a set T of $k+1$ affinely independent points. The* **dimension** *of a σ^k is $dim\sigma^k = k$. A simplex by default is a d-simplex. A finite collection of simplices, $\mathcal{K}$, is a* **simplicial complex** *if the faces (fixed-dimensional subregions) of every simplex in $\mathcal{K}$ also belong to $\mathcal{K}$, and the intersection of two simplices in $\mathcal{K}$ is either empty or a face of both. The dimension of $\mathcal{K}$ is $dim\mathcal{K} = \max_{\sigma \in \mathcal{K}} dim\sigma$. The* **underlying space** *of $\mathcal{K}$ is $\bigcup \mathcal{K} = \bigcup_{\sigma \in \mathcal{K}} \sigma$.*

We will use Ω to denote the (polygonal) domain of interest in $\mathbb{R}^d$. By a *simplicial mesh*, or just *mesh* in short, we mean a simplicial complex that exactly spans Ω. We will assume that Ω is *connected*, since meshes for different connected components can be computed independently.

Voronoi diagrams and *Delaunay triangulation* are also important geometric concepts which will be frequently brought up throughout the paper. Let $P = \{p_1, \dots, p_n\}$ be a set of points in $\mathbb{R}^d$. The Voronoi diagram and Delaunay triangulation can be defined as dual structures on a point set P as follows.

Definition 2.2 ([15, 23, 43, 46]). *The* **Voronoi region** *$V(p)$ of a point $p \in P$ is the region consisting of the points of $\mathbb{R}^d$ whose nearest neighbor in P is p. The* **Voronoi diagram** *$\mathcal{V}(P)$ is the partition of $\mathbb{R}^d$ into Voronoi regions $V(p)$, for all $p \in P$. The* **Delaunay triangulation** *$\mathcal{D}(P)$ is defined as the dual of $\mathcal{V}(P)$, such that $\{s,t\}$ is an edge of $\mathcal{D}(P)$, $s,t \in P$, if and only if $V(s)$ and $V(t)$ are adjacent.*

Suppose the points in P are in general positions, *i.e.*, a) no $d+1$ points are affinely dependent, and b) no $d+2$ points are co-spherical. Then $\mathcal{D}(P)$ is a simplicial complex that satisfy the *Delaunay criterion* – that the interior of the circumsphere of a d-simplex in $\mathcal{D}(P)$ contains no points in P.

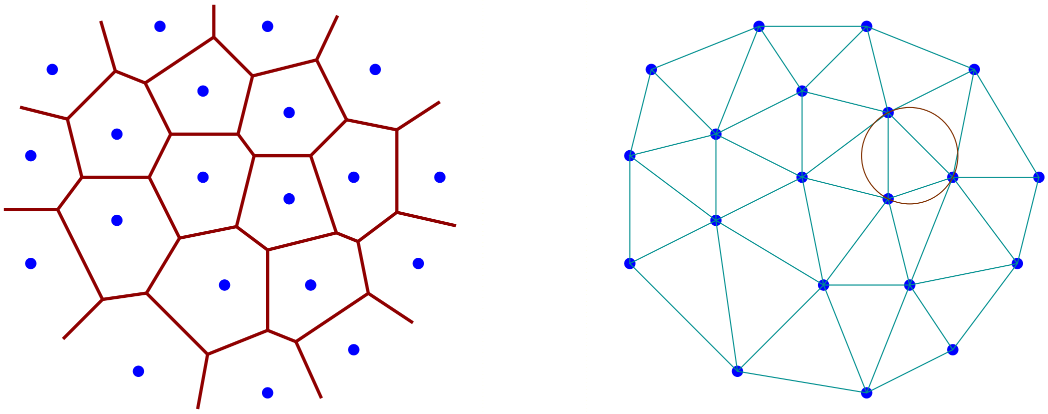

FIGURE 1. The Voronoi diagram and Delaunay triangulation of a set of points in $\mathbb{R}^2$

3. PROBLEM FORMULATIONS

A meshing problem of a domain Ω consists of two parts in general: mesh generation, mesh maintenance. Each part of the problem is formulated by specifying its preconditions and postconditions.

3.1. **Mesh generation.**

This is to construct a mesh based on a "skeleton" of the domain Ω of interest.

3.1.1. *Preconditions.* To generate a mesh we need some kind of description of Ω, such as the boundary faces of the domain and the interfaces between different subdomains. A domain on a plane can be generally described by a *planar straight line graphs* (PSLG) [7, 13, 49] for simple polygons, polygons with holes, point sets, *etc.* However, except for some special cases[1], such domain description becomes exponentially complicated in terms of spatial dimensionality. To generalize the notion of PSLG in $\mathbb{R}^d$, we define a *crossing-free flat set* (CFFS) [19], or a *piecewise linear system* as in [40], simply as a set of k-flats, $0 \leq k \leq (d-1)$, such that the intersection of any two flats is either empty or a common subface of both. A CFFS can be used to describe the boundary faces and interfaces of a domain in $\mathbb{R}^d$. Additional information, such as the orientation of each flat, may be needed to determine the inside or the outside of the domain at an arbitrary location. An elegant way for describing a CFFS Φ is to use the concept of an *incidence graph* defined as follows.

Definition 3.1 (Edelsbrunner [15]). *A* **incidence graph** $\mathcal{IG}(\Phi)$ *of a face set* Φ *in* $\mathbb{R}^d$ *is a graph where each vertex* $v(\phi)$ *of* $I(\Phi)$ *is associated with a* k*-face* ϕ *of* Φ*. An edge* $(v(\phi_1), v(\phi_2))$ *occurs in* $\mathcal{IG}(\Phi)$ *if and only if* ϕ_1 *is one dimension higher(lower) than* ϕ_2 *and* ϕ_1 *is a superface(subface) of* ϕ_2*.*

---

[1] such as magnetic resonance imaging (MRI), a sequence of cross sections of a 3D objects can be made to provide a set of contours lines in 2D [2].

3.1.2. *Postconditions.* A mesh $\mathcal{M}$ of a domain Ω, described by a CFFS Φ_Ω, is a simplicial complex that *conforms* Φ_Ω in a sense that each flat of Φ_Ω is preserved (and very likely subdivided) in $\mathcal{M}$. Therefore, mesh generation is also referred to as *conforming triangulation* [19]. In some cases, we may have a *constrained triangulation problem* where some flats of Φ_Ω are not allowed to be subdivided. For example in $\mathbb{R}^2$, a *constrained Delaunay triangulation* of a PSLG preserves all the edges of the graph, and can be constructed in $O(n \log n)$ time [13]. In general, constrained triangulation is conceivably much harder than conformal triangulation and will be exempted from further discussion.

A *measure set*, denoted by μ, is used to characterize a mesh, *e.g.*, its quality.

- Global measures
  - Worst *aspect ratio.* The *aspect ratio* of a convex polytope is the ratio between the largest and the smallest separations of parallel supporting hyperplanes [5].
  - Maximum k-angle, $0 \leq k \leq (d-2)$. A k-face in a d-simplex defines a *k-angle*, and it is measured by the fraction of which a small $(d-k-1)$-sphere centered on and perpendicular to the k-face lies interior to the d-simplex [5]. A 0-angle is a *solid angle*, and a $(d-2)$-angle is a *dihedral.*
  - Sum of Euclidean measures of all k-faces, $0 \leq k \leq d-1$. When $d = 2$, its the weight of the triangulation.
  - Mesh size or number of k-faces, $0 \leq k \leq d$, $k = 0$ for number of mesh points, and $k = d$ for number of elements.
  - Admission of Delaunay criterion.
- Local measures
  - Density of mesh points.
  - Anisotropy of connectivity between mesh points, characterized by a $d \times d$ *symmetric positive-definite* matrix [12].

Suppose the mesh is used to solve a PDE system defined on Ω. Then most of these quantities will determine the properties of the system of algebraic equations approximating the PDE system. For example, large value of the worst aspect ratio will lead to ill-conditioning of the system of algebraic equations. The Delaunay property of the mesh may be strongly desired for the finite volume method, since the perpendicular straight line duality between the Delaunay triangulation and the Voronoi diagram implies that the normal of a Voronoi facet is represented by a Delaunay edge, the dual of the facet, and thus the surface integral of a Voronoi cell is easily computed in terms of the mesh point coordinates as compared to the surface integral over barycentric cells which has to be used on non-Delaunay meshes.

To summarize, the input description Φ_Ω together with the output specification μ completely determines the mesh $\mathcal{M}(\Phi_\Omega, \mu)$. Unfortunately, for a given input Φ_Ω, choices over μ are so limited that many of these have led to open problems up to the present.

3.2. **Mesh maintenance.** Generally speaking, the goal of mesh maintenance is to efficiently update the mesh when the specifications on the input (Φ_Ω) and/or the output (μ) of a mesh alter. Therefore, we can say that the precondition of a mesh maintenance problem is the original mesh $\mathcal{M}(\Phi_\Omega, \mu)$ while the postcondition is the new objective mesh $\mathcal{M}(\Phi_{\Omega'}, \mu')$. Notice that the domain itself also can change in this problem setting. A naive approach will be to generate the objective mesh from scratch just like in a mesh generation problem, whereas different techniques can be developed for the sake of efficiency by making use of the information on the original mesh. The mesh maintenance procedure is also called adaptation, and the combination of mesh generation and adaptation is often referred to as *adaptive mesh generation.* For example in numerical PDE problems, when part of the domain needs too be refined for progressive accuracy of the numerical solution, the local density of the mesh will need to change.

4. Mesh Generation Methodology

There are many existing methods to generate meshes in fixed or arbitrary dimension that meet certain mesh specifications. Flavors of these methods may vary in wide range from very theoretical, the computer geometer's type, to very practical, the engineer's type. In this section, we will start with describing a general paradigm nad then proceed with reviewing individual mesh generation methods – algorithms, mesh features, limits and advantages, and so on.

4.1. **A general paradigm.** A mesh generation method for a CFFS Φ_Ω describing a domain Ω in general introduces some additional points, called the *Steiner points*, and triangulates both the input Φ_Ω and the Steiner points. Steiner points and Delaunay triangulation are two commonly used techniques in mesh generation.

4.1.1. *Steiner points.* Steiner points are the vertices in the mesh that are not the vertices of input. Steiner points are introduced to help conform the faces in Φ_Ω and to approach certain mesh specifications, such as higher density, smaller aspect ratio, and so on. Sometimes, Steiner points can help reduce the size of triangulation in higher dimensions.

To the author, Steiner faces may be conceivably useful for mesh generation in higher dimensions. However they have not been mentioned in the current literature.

4.1.2. *Delaunay triangulation.* Even when the Delaunay property of the mesh is not required from the application, Delaunay triangulation is often used as an auxiliary structure (an intermediate or background mesh) and/or a primary structure (the objective mesh) because its nice geometric properties and ease of construction. For example, two dimensional Delaunay triangulation of a given point set optimizes various quality measures including the smallest angle, the larger circumscribing circle, the largest minimum enclosing circle, and the integral of the gradient squares [20, 47]. Moreover, the Delaunay triangulation contains the nearest-neighbors graph and the minimal spanning tree [39].

On the other hand, the Delaunay criterion of a triangulation can be satisfied without much overhead. Edelsbrunner and Shah [17] have shown that it takes at most $O(n \log n + n^{\lceil d/2 \rceil})$ number of edge flippings to transform an arbitrary triangulation to a Delaunay triangulation. Construction of Delaunay triangulations (and Voronoi diagrams) for a point set is well studied in computational geometry, and the algorithmic complexity is known to be at most as difficult as that of construction convex hulls in one higher dimension [23]. Commonly used techniques for lower dimensional Delaunay triangulation are divide-and-conquer, plane-sweep and randomized incremental algorithms [23, 32], all of which have similar bounds on running time (or expected running time) of $O(n \log n)$ in $\mathbb{R}^2$ and $O(n^2)$

in $\mathbb{R}^3$, and can be generalized to higher dimensions. However the incremental Delaunay construction is known to be extremely useful for mesh generation, especially for adaptive mesh generation. Point insertion to a Delaunay triangulation is considered as a basic operation, and is named as *Delaunay kernel* by Freitag, Borouchaki and George in [26], in the paradigm of Delaunay mesh generation. Barber, Dobkin and Huhdanpaa [4] have provided the public state-of-the-art implementation of Delaunay triangulation (including convex hull construction) in arbitrary dimensions, known as the `qhull` program.

4.2. **2^d-tree method.** This is one of the earliest proposed and most popularly used methods in mesh generation. We give the following definition of a 2^d-*tree* analogous to those of a *quadtree* [7] in $\mathbb{R}^2$ and an *octree* [41] in $\mathbb{R}^3$.

Definition 4.1. *A* 2^d**-tree** *is a recursive partition of a region into axis-aligned hypercubes, called* **cells***. One hypercube, the root, covers the entire region. A cell can be divided into* 2^d *child cells by splitting it in halves in all the* d *dimension. A* 2^d*-tree is said to be* **balanced** *if any edge of all leaf cells is split at most once.*

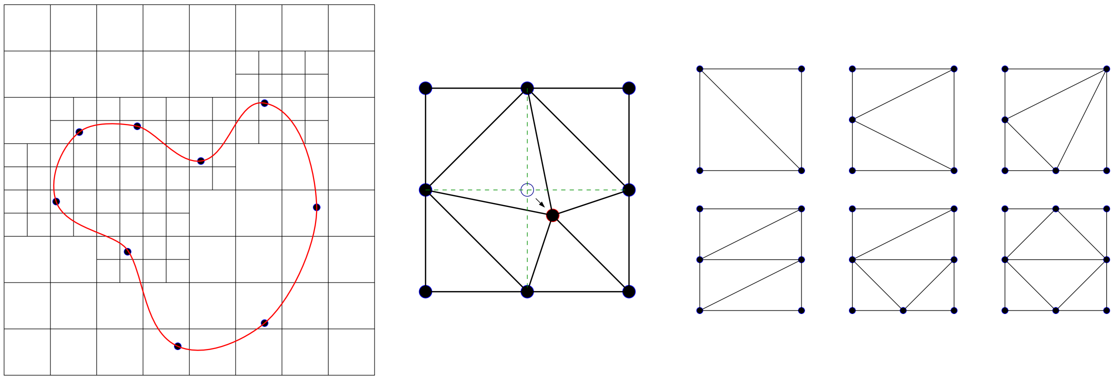

FIGURE 2. A balanced quadtree and triangulation of its leaf cells

First, a fine enough balanced 2^d-tree is constructed over the domain Ω described by Φ_Ω such that no leaf cell is *crowded* [7, 41]. Then the leaf cells, either intersecting Φ_Ω or not, are triangulated separately. In this case, the vertices of the leaf cells are used as Steiner points.

In general, a 2^d-tree triangulation produces near optimal number of Steiner points with bounded aspect ratio.

Theorem 4.2 (Bern, Eppstein and Gilbert [8])**.** *Let* X *be a point set in* $\mathbb{R}^d$*,* $\mathcal{T}$ *be an arbitrary triangulation that respects* X*, and* $A(\mathcal{T})$ *be the maximum aspect ratio of all simplices in* $\mathcal{T}$*. Then there exists a* 2^d*-tree triangulation that respects* X *and has size* $O(|\mathcal{T}|\log A(\mathcal{T}))$ *with bounded aspect ratio.*

Moreover, 2^d-tree is an optimal way to reduce the number of simplices of the Delaunay triangulation for a given set of points.

Theorem 4.3 (Bern, Eppstein and Gilbert [8]). *Let X be a point set in $\mathbb{R}^d$. Then there exist a set $Y \supseteq X$ of $O(|X|)$ points for which the d-dimensional Delaunay triangulation contains only $O(|X|)$ simplices.*

In $\mathbb{R}^2$, the quadtree algorithm by Bern, Eppstein and Gilbert [8] is able to generate meshes with all angles of the output triangles greater than 20°, and furthermore by strengthening the balance condition in the quadtrees and replacing quadtree squares by tiles with projections and indentations, it can guarantee that all angles measure between 36° and 80° [7]. One property of the path simplices is that all dihedrals are bounded above by 90°. In $\mathbb{R}^d$, a *path simplex* is defined as a simplex which is isometric to

$$\{(x_1, \dots, x_d) \mid 0 \le c_1 x_1 \le \cdots \le c_d x_d \le 1\}$$

for some set of positive constants $c_1, \dots, c_d$. Bern, Chew, Eppstein and Ruppert [5] have shown that

Theorem 4.4. *Any point set in $\mathbb{R}^d$ can be triangulated with $O(n^{\lceil d/2 \rceil})$ path simplices.*

A lower bound of $\Omega(n \log A)$, with A the maximum aspect ratio of a simplex in the Delaunay triangulation, is given as the complexity of the triangulation with all dihedrals are no larger than $90° - \epsilon$ for some $\epsilon > 0$. However it is unknown whether a 2^d-tree-based method can be used to achieve an $O(n \log A)$ upper bound for acute-dihedral-triangulation.

Variants of the 2^d-tree method have been found useful in mesh generation. Neugebauer and Diekmann [44] used a rhomboid structure of the quadtree to generate meshes with angle bounds between 30° and 90°. Conti, Tomizawa and Yoshii [14] used the 2-4-8-*tree*, a variation of the octree where a cell is not always partitioned in all three directions, to generate oriented meshes.

4.3. **Sphere packing method** [40].

A *spacing function* f indicating the density of mesh is a positive and α-Lipschitz function defined over the domain Ω, where the Lipschitz condition implies $|f(y) - f(x)| \le \alpha \|y - x\|$ for all points x and y in the domain. An *f-sphere* at a point $p \in \Omega$ is defined as a sphere with center p and radius $f(p)$. A set of points is a *sphere-packing* with respect to the spacing function f if the interiors of their f-spheres do not intersect. It is *maximal* if P is a maximal such set. A Delaunay triangulation of a maximal sphere-packing can be computed and will have bounded *radius-edge ratio*, the ratio between the radius of the circumscribing sphere and the shortest edge.

The idea for using sphere-packing to construct Delaunay mesh for a CFFS Φ_Ω is first to pack spheres around all faces of Φ_Ω, in the order of increasing dimension, and then to pack spheres around Steiner points to make it maximal. The Delaunay triangulation of vertices in Φ_Ω and the Steiner points will conform all faces in Φ_Ω. A sphere-packing normally generates good quality meshes in 2D but may generates *slivers*, tetrahedra with four vertices lying close to the equator of a sphere, in 3D [39].

4.3.1. *Oversampling and filtering* [39, 40].

It's not obvious how to generate a maximal packing for a given spacing function. In some cases, the maximality is weakened such that the sphere packing only needs to be *ϵ-maxima* ($\epsilon > 0$) where each ϵf-ball contains a point that can not be added to the sphere packing. *Oversampling* samples a point for each ϵf-ball. A *conflict graph* is constructed where the vertices are the oversampling points, and an edge is drawn between two points if interior of their f-spheres intersect. Then *filtering* is to find a maximal independent set. However the quality of the resulting mesh has not been evaluated versus other mesh generators.

4.3.2. *A 2D Delaunay refinement method.* Ruppert [49] gives a remarkably simple algorithm to triangulate PSLGs in $\mathbb{R}^2$. This algorithm is based on the randomized-incremental algorithm by Guibas, Knuth, and Sharir [32], and the refinement algorithm by Chew used for constructing quality constrained Delaunay triangulations [13]. It first generates a Delaunay triangulation for the vertices of the input PSLG and then successively inserts Steiner points and updates the Delaunay triangulation until all the triangles are well-shaped and conform the original PSLG. The edges that form the edges of the PSLG are referred to as *segments* to be distinguished from the other edges of the Delaunay triangulation. The circumcenter of a bad-shaped triangle is added as a Steiner point if it does not lie in the diametral circle of any segments in the current Delaunay triangulation. Otherwise the midpoint of such segment is added.

A directed acyclic graph (DAG) is used to store the history of the Delaunay triangulation [17, 32, 47], such that the running time for point insertion to the Delaunay triangulation is proportional to the logarithm of the size of current Delaunay triangulation on average. And the expected running time for the algorithm is $O((n+k)\log(n+k))$, where n is the size of the PSLG and k is the total number Steiner points. A mesh generated by this method is always a Delaunay triangulation and the size of its Steiner points is near optimal:

Theorem 4.5 (Ruppert [49]). *Given $\alpha \leq 20°$, and input X, suppose $\mathcal{T}$ is any triangulation of X with minimum angle bound α. There is a constant C_α such that if $\mathcal{T}$ has N triangles, then the Delaunay refinement triangulation $\mathcal{T}_D$ has $N_D \leq C_\alpha \cdot N$ triangles. Letting $\mathcal{T}$ be the triangulation with fewest possible triangles shows that $\mathcal{T}_D$ is within a factor C_α of optimal.*

Ruppert's Delaunay refinement method can be actually deduced from an iterative sphere-packing if we use the notion *Voronoi maximality* [40], another weakened maximality like the ϵ-maximality used in the oversampling and filtering method. We define a sphere-packing to be Voronoi maximal if no circumcenter of a Delaunay ball can be added. How to extend the Delaunay refinement method to higher dimensions is still an open problem. The main difficulty in 3D lies in the fact that the center of minimum containing sphere of a triangle may not stay inside the triangle.

4.3.3. *A 2D disk packing method for nonobtuse triangulation.* The algorithm by Bern, Mitchell and Ruppert [9] consists of two stages. The first stage packs the domain with non-overlapping disks, tangent to each other and to sides of the domain. The algorithm then adds edges (radii) between centers of disks and points of tangency on their boundaries, thereby dividing the domain into small polygons. The second stage triangulates the small polygons using Steiner points located only interior to the polygons or on the domain boundary. It can be shown that

Theorem 4.6 (Bern, Mitchell and Ruppert [9]). *Let Ω be a polygonal region with n vertices. Ω can be triangulated with $O(n)$ right triangles. The running time is $O(n \log n)$ for simple polygons and $O(n \log^2 n)$ for polygons with holes.*

The result is significant compared to the previous polynomial size ($O(n^2)$ [6]) nonobtuse triangulations. However, it is not known how well one can do to triangulate a PSLG into nonobtuse triangles.

Open Problem 4.7. *Is there an algorithm for polynomial-complexity nonobtuse triangulation of a PSLG?*

4.4. **Advancing-front method.** This method by Frey, Borouchaki, and George [26] is more of an engineering approach. It generates high-quality meshes for many real examples such as geological objects and space surrounding an aircraft. However, no bound on the size or the worst aspect ratio of advancing-front meshes has been proven.

The method first generates a background mesh – a conformal Delaunay triangulation of the CFFS Φ_Ω describing the domain Ω, with Steiner points added only on the faces of Φ_Ω. The set of simplicial faces of the background mesh that form the faces in Φ_Ω is called the initial *front*, denoted by $\mathcal{F}$, A *size field* is defined at a point $p \in \Omega$ as the average length of the edges that is connected with p if p is a vertex of $\mathcal{F}$ or as the linear interpolation of the size field values at all vertices of the simplex containing p. And the distance between two points in Ω is normalized by the size field. The algorithm then successively adds Steiner points which departure from the current font into the domain, updates the Delaunay triangulation and the redefines the front, until the front is empty. At each stage, the frontier separates a collection of well-shaped simplices (toward the boundary of the domain) and a collection of bad-shaped simplices (toward the kernel of th domain), and the Steiner points are generated in the following way:

- For each simplicial face $f \in \mathcal{F}$ find a point that is equidistant from all vertices of f in the sense of normalized distance and with normalized distance α, an adjustable parameter. Let N denote the collection of such points for all faces in $\mathcal{F}$.
- Filter N such no points are too close to the frontier and two points are too close to each other. The filtered points are added as Steiner points.

Throughout the procedure, the well-shaped simplices as well as the bad-shaped ones always form a conformal Delaunay triangulation of the domain. Therefore, it is also called the *advancing-front Delaunay triangulation*, whereas other advancing-front mesh generation is devised by using different kind of heuristics to march the front [36].

5. Mesh Maintenance Methodology

Efficient mesh maintenance is demanded when change of mesh conditions becomes a frequent query in applications such as solving numerical PDEs. The mesh needs to be refined or coarsened (or derefined) when the density requirement changes. Mesh smoothing can improve the aspect ratio and eliminate too large or small angles of the elements. Most of the time, these changes only need to happen locally. Therefore, avoiding global behavior in these situations is important to make the methods efficient.

5.1. **Refinement.** In general, the mesh can be refined by placing more Steiner points inside the domain. A simple and commonly used method is *bisection* [35, 45, 48]. It uses the midpoint of some "long" edges as Steiner points and connect them together and with existing vertices to form a valid triangulation. Liu and Joe [35] have shown that there exist a minimum angle bound for the bisection refinement in 3D. If a sphere packing method [40] is used to generate the original mesh, then the local refinement can be achieved by decrease the spacing function vale locally while still keeping its α-Lipschitz condition.

5.2. **Coarsening.** The general method used for coarsening simplicial meshes is that of picking some subset of the original mesh points and triangulating the subset. A commonly used technique is that of the *maximal independent set* (MIS), where a mesh is viewed as a graph by its 1-skeleton, and an MIS of the graph can be used to form a coarsening. However, repeatedly application of the MIS technique to construct a coarsening hierarchy can degrade the aspect ratio of the meshes [40].

Again, the sphere packing technique used for mesh generation also can be used for mesh coarsening [40]. Here the goal is to produce a hierarchy $\mathcal{M}_1, \ldots, \mathcal{M}_k$ of respectively coarser good aspect ratio meshes given an initial fine simplicial mesh $\mathcal{M}_0$. The main idea is to construct a spacing function f_i at each stage, construct a conflict graph $CG_i = (P_0, E_i)$ with P_0 the mesh points in $\mathcal{M}_0$ and

$$E = \{(p, q) \mid \|q - p\| < f_i(p) + f_i(q)\},$$

compute P_i as a completion of P_{i-1} into an MIS of CG_i and $\mathcal{M}_i$ as the Delaunay triangulation of P_i. The value of the initial spacing function f_0 at a point is computed as half its distance to the nearest neighbor in P_0, so that f_0 is 0.5-Lipschitz. A successive spacing function f_i is generated by taking the maximum of f_0 and a constant function, the value of which indicates the degree of coarsening, so that f_i remains 0.5-Lipschitz, and therefore guarantees the quality of the mesh coarsened at that stage.

5.3. **Smoothing.** Mesh smoothing moves the Steiner points of the mesh while preserving its overall topology. A commonly used technique, *Laplacian smoothing*, sweeps over the mesh, successively moving each point to the *centroid* of its neighbors. This technique has been practically used because of its simplicity. However it is not directly connected to any specific mesh quality criterion; moreover it may not remove slivers from three dimensional meshes, and the result may not even remain a valid triangulation [1, 7].

A second technique, called *mesh relaxation* by Frey and Field [27], flips edges to regularize degrees of vertices. An edge of the triangulation is flipped if the flipping causes sum of its endpoints' degrees decreases by more than two.

Freitag, Jones and Plassmann [24] proposed an alternative to Laplacian smoothing. Rather than using the centroid, their optimization-based method computes for each Steiner point a new placement that maximizes the minimum angle in adjacent triangles. An iterative steepest-descent algorithm is used to solve the optimal placement problem. However the algorithm has not been proved to be correct.

What is the most intriguing and seems to be the most promising method in mesh smoothing is the *quasiconvex programming* by Amenta, Bern and Eppstein [1]. A quasiconvex programming problem is a generalized linear programming (GLP) problem where all the level sets of the objective functions are convex. More formally, a *nested convex family* is defined as a map $\kappa(t)$ from $\mathbb{R}$ to convex sets in $\mathbb{R}^d$ such that if $a < b$ then $\kappa(a) \subset \kappa(b)$. For a simplex σ of the mesh, a nested convex family can be constructed as $\kappa_\sigma(t) = \{x \mid q(\sigma(x)) \le t\}$, where $\sigma(p)$ indicates the simplex formed by moving the Steiner point to the position x, and $q(\sigma)$ is a quality measure of σ. Then finding the optimal Steiner point placement is equivalent to finding the optimal quality bound that allows a feasible placement. The set of quality measures that can be used to construct nested convex families are: 1) volume, 2) altitude, 3) containing sphere, 4) dihedrals, 5) solid angles.

Theorem 5.1 (Amenta, Bern and Eppstein [1]). *In any constant dimensions, the Steiner point placement optimizing each of the above criteria, or a weighted combination of criteria, can be computed in linear time by quasiconvex programming.*

However, further study is needed to find out to what extent theoretical generalized linear programming algorithms can serve as practical methods for mesh smoothing problems.

6. Conclusion

We have tried to gather together some scattered but up-to-date results as well as open problems of the various simplicial mesh generation and maintenance problems for our future study in this area.

References

[1] N. Amenta, M. Bern and D. Eppstein, "Optimal point placement for mesh smoothing," *Proc. 7th ACM-SIAM Symp. Disc. Algo.*, 528-537, 1997.

[2] C. L. Bajaj, E. J. Coyle and K. N. Lin, "Surface and 3D triangular meshes from planar cross sections", *Proc. 5th International Meshing Roundtable*, 169-178, Sandia National Laboratories, 1996.

[3] R. E. Bank and J. Xu, "An Algorithm for Coarsening Unstructured Meshes," *Numerische Mathematik* 73, 1-36, 1996.

[4] C. B. Barber, D. P. Dobkin and H. Huhdanpaa, "The Quickhull algorithm for convex hulls," *ACM Trans. on Mathematical Software*, Dec 1996.

[5] M. Bern, P. Chew, D. Eppstein and J. Ruppert, "Dihedral bounds for mesh generation in high dimensions," *Proc. 6th ACM-SIAM Symp. Disc. Algo.*, 189-196, 1995.

[6] M. Bern and D. Eppstein, "Polynomial-size nonobtuse triangulation of polygons," *Int. J. Comp. Geometry and Applications*, 2, 241-255, 1992.

[7] M. Bern and D. Eppstein, "Mesh generation and optimal triangulation," *Computing in Euclidean geometry*, 47-123, D. Z. Du and F. Hwang ed, World Scientific, Singapore, 1995.

[8] M. Bern, D. Eppstein and J. R. Gilbert, "Provably good mesh generation," *Proceedings 31st Symp. Foundations of Computer Science*, 231-241, IEEE, 1990.

[9] M. Bern, S. Mitchell and J. Ruppert, "Linear-size nonobtuse triangulation of polygons," *Proc. 10th ACM Symp. Comput. Geom.*, 221-230, 1994.

[10] J. D. Boissonnat, A. Cérézo, O. Devillers and M. Teillaud, "Output sensitive construction of the 3D Delaunay triangulation of constrained sets of points," *Technical Report*, Institut National de Recherche en Informatique et Automatique – France, Number RR-1415, 1991.

[11] H. Borouchaki, F. Hecht, E. Saltel and P. L. George, "Reasonably efficient Delaunay based mesh generator in 3 dimensions," *Proceedings 4th International Meshing Roundtable*, 3-14, Sandia National Laboratories, 1995.

[12] F. J. Bossen and P. S. Heckbert, "A pliant method for anisotropic mesh generation", *Proc. 5th International Meshing Roundtable*, 63-76, Sandia National Laboratories, 1996.

[13] L. P. Chew, "Guaranteed-quality triangular meshes," *Technical Report* 89-983, Cornell University, 1989

[14] P. Conti, M. Tomizawa and A. Yoshii, "Generation of oriented three-dimensional Delaunay grids suitable for the control volume integration method,", *International Journal for Numerical Methods in Engineering*, 37, 3211-3227, 1994.

[15] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer-Verlag, 1987.

[16] H. Edelsbrunner, "Algebraic decomposition of non-convex polyhedra," *Technical Report*, Department of Computer Science, University of Illinois at Urbana-Champaign, Number 1928, 1995.

[17] H. Edelsbrunner and N. R. Shah, “Incremental topological flipping works for regular triangulations,” *Proc. 8th ACM Symp. Comput. Geom.*, 43-52, 1992.

[18] H. Edelsbrunner and N. R. Shah, “Triangulating topological spaces,” *Proc. 10th ACM Symp. Comput. Geom.*, 285-292, 1994.

[19] H. Edelsbrunner and T. S. Tan, “An upper bound for conforming Delaunay triangulations,” *Proc. 8th ACM Symp. Comput. Geom.*, 53-62, 1992.

[20] H. Edelsbrunner and R. Waupotitsch, “Adaptive simplicial grids from cross-sections of monotone complexes,” *Technical Report*, Department of Computer Science, University of Illinois at Urbana-Champaign, Number 1864, 1994.

[21] D. Field, “Give me a good mesh,” *Proc. 5th International Meshing Roundtable*, pp.3, Sandia National Laboratories, 1996.

[22] M. Filipiak, “Mesh generation,” *Technology Watch Report*, Edinburgh Parallel Computing Centre, 1996

[23] S. Fortune, “Voronoi diagrams and Delaunay triangulations,” *Computing in Euclidean geometry*, 225-265, D. Z. Du and F. Hwang ed, World Scientific, Singapore, 1995.

[24] L. A. Freitag, M. T. Jones and P. E. Plassmann, “An efficient parallel algorithm for mesh smoothing,” *Proc. 4th International Meshing Roundtable*, 47-58 Sandia National Laboratories, 1995.

[25] L. A. Freitag and C. Ollivier-Gooch, “A comparison of tetrahedral mesh improvement techniques,” *Proc. 5th International Meshing Roundtable*, 87-106, Sandia National Laboratories, 1996.

[26] P. J. Frey, H. Borouchaki and P. L. George. “Delaunay tetrahedralization using an advancing-front approach”, *Proc. 5th International Meshing Roundtable*, 31-46, Sandia National Laboratories, 1996.

[27] W. H. Frey and D. A. Field, “Mesh relaxation: a new technique fro improving triangulations,” *International Journal for Numerical Methods in Engineering*, 31, 1121-1133, 1991.

[28] J. Galtier and P. L. George, “Prepartitioning as a way to mesh subdomains in parallel,” *Proc. 5th International Meshing Roundtable*, 107-122, Sandia National Laboratories, 1996.

[29] R. Garimella, and M. S. Shephard, “Tetrahedral mesh generation with multiple elements through the thickness,” *Proceedings 4th International Meshing Roundtable*, 321-333, Sandia National Laboratories, 1995.

[30] P. L. George, *Automatic Mesh Generation – Application to Finite Element Methods*, John Wiley & Sons, 1991.

[31] C. S. Gitlin and C. R. Johnson, “MeshView: a tool for exploring 3D unstructured tetrahedral meshes,” *Proc. 5th International Meshing Roundtable*, 333-345, Sandia National Laboratories, 1996.

[32] L. J. Guibas, D. E. Knuth and M. Sharir, “Randomized incremental construction of Delaunay and Voronoi diagrams,” *Algorithmica*, 7, 381-413, 1992.

[33] B. Joe, “Construction of three-dimensional Delaunay triangulation using local transformations,” *Computer Aided Geometric Design*, 8, 123-142, North-Holland, 1991.

[34] M. T. Jones and P. E. Plassmann, “Parallel algorithms for adaptive mesh refinement,” *SIAM J. Sci. Comput.*, 18(3), 686-708, 1997.

[35] A. Liu and B. Joe, “Quality local refinement of tetrahedral meshes based on 8-subtetrahedron subdivision,” *Mathematics of Computation*, 65(215), 1183-1200, 1996.

[36] D. J. Marvriplis, “Unstructured mesh generation and adaptivity,” *ICASE Report*, NASA Langley Research Center, Number 95-26, 1995.

[37] T. Midtbø, “Spatial modeling by Delaunay networks of two and three dimensions,” *Dr.Ing. Thesis*, Department of Surveying and Mapping, Norwegian University of Science and Technology, 1993.

[38] G. L. Miller, D. Talmor and S. H. Teng, “Optimal good-aspect-ratio coarsening for unstructured meshes,” *Proc. 7th ACM-SIAM Symp. Disc. Algo.*, 538-547, 1997.

[39] G. L. Miller, D. Talmor, S. H. Teng and N. Walkington, “A Delaunay based numerical method for three dimensions: generation, Formulation, and Partition,” *Proceedings 36th Symp. Foundations of Computer Science*, 683-692, IEEE, 1995.

[40] G. L. Miller, D. Talmor, S. H. Teng, N. Walkington and H. Wang, “Control volume meshes using sphere packing: generation, refinement and coarsening,” *Proc. 5th International Meshing Roundtable*, 47-62, Sandia National Laboratories, 1996.

[41] S. Mitchell and S. Vavasis, “Quality mesh generation in three dimensions,” *Proc. 8th ACM Symp. Comput. Geom.*, 212-221, 1992.

[42] S. Mitchell and S. Vavasis, "An aspect ratio bound for triangulating a d-grid cut by a hyperplane," *Proc. 12th ACM Symp. Comput. Geom.*, 48-57, 1996

[43] K. Mulmuley, *Computational Geometry – An Introduction through Randomized Algorithms*, Pretince Hall, 1994.

[44] F. Neugebauer and R. Diekmann, "Improved mesh generation: not simple but good," *Proc. 5th International Meshing Roundtable*, 257-272, Sandia National Laboratories, 1996.

[45] A. Plaza and G. F. Carey, "About local refinement of tetrahedral grids based on bisection," *Proc. 5th International Meshing Roundtable*, 123-136, Sandia National Laboratories, 1996.

[46] F. P. Preparata and M. I. Shamos, *Computational Geometry*, Springer-Verlag, 1985.

[47] V. T. Rajan, "Optimality of the Delaunay triangulation in $\mathbb{R}^d$," *Proc. 7th ACM Symp. Comput. Geom.*, 357-363, 1991.

[48] M. C. Rivara and P. Inostroza, "A discussion on mixed (longest-side midpoint insertion) Delaunay techniques for the triangulation refinement problem," *Proceedings 4th International Meshing Roundtable*, Sandia National Laboratories, 1995.

[49] J. Ruppert, "A Delaunay refinement algorithm for quality 2-dimensional mesh generation," *J. Algo.*, 18(3), 548-585, 1995.

[50] J. A. Schmidt, "Mesh generation with applications in computational electrophysiology," *Ph.D. Thesis*, Department of Electrical Engineering, Duke University, 1993.

[51] J. F. Thompson and N. P. Weatherill, "Structured and unstructured grid generation," *High-Performance Computing in Biomedical Research*, 63-111, CRC Press, 1993.

[52] N. P. Weatherill and O. Hassan, "Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints," *International Journal for Numerical Methods in Engineering*, 37, 2005-2039, John Wiley & Sons, 1994