Delaunay Tetrahedralization and Parallel Three Dimensional Mesh Generation

Gary L. Miller Dafna Talmor Shang-Hua Teng

School of Computer Science *
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Department of Mathematics and LCS †
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

We present a new parallel algorithm for computing the Delaunay triangulation (tetrahedralization) of a point set P (in d dimensions for a fixed d) that satisfies natural geometric properties that arise from mesh generation. In particular, we show that if the largest ratio of the circum-radius to the length of smallest edge over all simplexes in the Delaunay triangulation of P, $DT(P)$, is bounded, (called the bounded radius-edge ratio property), then we show that $DT(P)$ has a bounded degree and can be computed in random $O(\log n)$ time using n processors. Our construction uses the observation that $DT(P)$ has a separator of size $O(n^{1-1/d})$ that can be efficiently found by the geometric separator algorithm of Miller, Teng, Thurston, and Vavasis. The bounded radius-edge ratio property is desirable for well-shaped triangular meshes for finite element and finite difference methods. For example, in two dimensions, it implies that the smallest angle in the Delaunay triangulation is bounded from below. We also show how to generate efficiently a size optimal point set with bounded radius-edge ratio property for a discretization description of a continuous domain in parallel. Our algorithm extends the two dimensional point generation algorithm of Miller and Talmor to three dimensions. Our algorithms provide two important subroutines for parallel three dimensional mesh generation.

1 Introduction

An essential step in scientific computing is to find a proper discretization of a continuous domain with a mesh of simple elements. This is the problem of *mesh generation*. In scientific simulation, once we have a discretization or a mesh, differential equations representing physical properties such as flow, waves, and heat distribution are then approximated by linear or non-linear functions that are generated by either finite difference or finite element formulations. However, not all meshes have equal performance in the subsequent numerical solution. Numerical and discretization errors depend on the geometric shape and size of the elements while the computational complexity for finding the numerical solution depends on the number of elements in the mesh and often the overall geometric quality of the mesh as well. Typically, the mesh is unstructured and its elements are triangles in two dimensions and tetrahedra in three dimensions. Such a mesh is called an *unstructured triangular mesh* and is the most versatile type of mesh. The quality of a mesh is often measured by the size and shape, such as the angles and the aspect-ratio, of its triangular elements.

Thus, a mesh generator must determine a partition of the underlining 3-dimensional region into tetrahedron, which in turn determines a collection of bounding triangles, edges, and points. On the other hand, given a set of points P there is a very natural triangulation, namely, the Delaunay Triangulation of P, $DT(P)$. For general points set P in $d \geq 3$ dimensions $DT(P)$ can be very poor for two reasons: the number of simplexes may be exponential in d and the aspect-ratio of the simplexes may be very small. In this paper we first address the question of finding efficient parallel algorithms for computing the Delaunay triangulation of a point set when it is known that the triangulation is small, linear in the size of

* Supported in part by NSF Grant CCR-9016641

† Supported in part by AFOSR F49620-92-J-0125 and DARPA N00014-92-J-1799.

the point set, and a weak condition on the aspect-ratio of the simplexes. We will then present a new method for generating points sets whose *DT* satisfies these two conditions.

Even though our *DT* algorithm should have other application will shall focus on its application to mesh generation. In Figure 1 we show the Delaunay triangulation of our point generation algorithm for the case when there is a singularity in the underling function being approximated. Observe that our *DT* algorithm is able to efficiently handle cases were the point set which may have accumulation points.

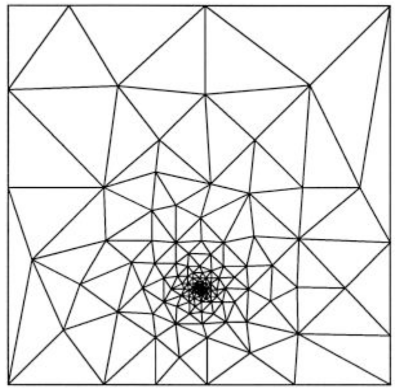

Figure 1: Triangulation of well-spaced point set around a singularity

Various heuristics have been proposed for mesh generation in both two and three dimensional space and are used in practice. Recently, some provably good mesh generators have been developed, most notably the two dimensional Bern-Eppstein-Gilbert quad-tree algorithm and Ruppert's Delaunay refinement algorithm [1, 20]. Three dimensional mesh generation is much harder than two dimensions; only one provably good mesh generator exist and it was developed by Mitchell and Vavasis [18] and uses oct-trees. On the other hand, it becomes increasingly important to have an efficient parallel mesh generator. Efficient parallel mesh partitioning algorithm [16, 10, 11, 21] have been implemented to map unstructured meshes onto parallel machines. Various parallel algorithms have been developed in recent years for finite element and finite difference methods and for solving sparse linear systems, but parallel mesh generation is still less common. It may become a serious bottleneck, especially for parallel adaptive computations.

On the theoretical side, Bern, Eppstein, and Teng [2] developed the first parallel algorithm for quality mesh generation in two dimensions. Although, their approach can be extended to three dimensions by parallelizing Mitchell and Vavasis oct-tree algorithm [18], the constant in mesh size may be fairly large. Its performance in practice still needs to be seen. It is desirable to have a practical parallel mesh generator especially for three dimensions.

As mention above we present a new scheme for parallel three dimensional mesh generation. Our scheme generates a mesh in two steps: point set generation and Delaunay tetrahedralization. We generalize the Poisson point process of Miller and Talmor [14] to three dimensions. The point set generated has the properties that (1) the local density of the points approximate the local feature size of the input and hence is optimal up to to a constant. (2) Its Delaunay tetrahedralization has a bounded ratio radius property. Namely, we define the *circum-radius* of a simplex to be the radius the of uniquely defined sphere passing through the vertices of the simplex. The *radius-edge ratio* of the simplex is the ratio of the circum-radius over the length of the shortest side.

We then show that the bounded radius-edge ratio property implies that the Delaunay triangulation in d-dimensions satisfies the density property [17, 16], an important geometric property that is true for well shaped finite difference meshes. The *density* of the graph with a given embedding is the largest ratio of the length of longest edge incident to it to its distance to the nearest neighbor over all nodes of the graph. A graph with bounded density always has finite degree and is a special case of the overlap graph defined by the nearest neighborhood system [15]. This observation enables us to use the geometric sphere separator decomposition to find the Delaunay triangulation.

Algorithmically, the basic idea then is divide and conquer. First, the divide and conquer algorithm is used to find the nearest neighbor system in parallel $O(\log n)$ time using n processors [9]. The observation is that the DT is a subgraph of the overlap graph. We then use the divide and conquer approach to find the DT edges and faces. The proof of the correctness is not direct and relies on some nice geometric results which are interesting on their own right. Our algorithm can be applied to the two dimensional algorithm of Miller and Talmor [14], removing the need for a parallel three dimensional convex hull algorithm. Our new scheme is uniform and data parallel, in part, because the geometric partitioner is data parallel in nature.

Geometrically and algorithmically, we show that if the DT of a point set has overlap properties (which include density property and bounded radius-edge radio property as a special case), then the DT has linear size and can be constructed in $O(n \log n)$ time and $O(\log n)$ parallel time using n processors in 2 and 3 dimensions, and that for the density property and random set points similar bounds exist in any dimension.

2 Definitions

In the following definition, the term "triangle" is used as a short hand notation for "simplex on d+1 vertices". **Delaunay Triangulation** Given a set S of N points in R^d, a subset $S' \subseteq S$ of size $|S'| = d + 1$ is a Delaunay triangle if the sphere defined by S' contains no points from S in its interior.

If the set S contains no $d + 2$ points which are co-spherical, then the set of Delaunay triangles defines a unique triangulation of S. For more discussion, see [19, 7].

Definition 2.1 (Separators) *A subset of vertices C of a graph G with n vertices is an $f(n)$-**separator** that δ-**splits** if $|C| \leq f(n)$ and the vertices of $G - C$ can be partitioned into two sets A and B such that there are no edges from A to B, and $|A|$, $|B| \leq \delta n$, where f is a function and $0 < \delta < 1$.*

We now give a brief review of the results of sphere separators of overlap graphs. One important notation is the neighborhood system [16]. Let $P = \{p_1, \ldots, p_n\}$ be points in $\mathbf{R}^d$. A *k-ply neighborhood system* for P is a set, $\{B_1, \ldots, B_n\}$, of closed balls such that (1) B_i is centered at p_i and (2) no point $p \in \mathbf{R}^d$ is strictly interior to more than k balls from B.

Given a neighborhood system $\{B_1, \ldots, B_n\}$ description of a point set P, we define a family of geometric graphs called *overlap graphs*.

Definition 2.2 *Let $\alpha \geq 1$ and let $B = \{B_1, \ldots, B_n\}$ be a k-ply neighborhood system for $P = \{p_1, \ldots, p_n\}$. The (α, k)-**overlap graph** for the k-ply neighborhood system $\{B_1, \ldots, B_n\}$ is the undirected graph with vertices $V = \{1, \ldots, n\}$ and edges $E = \{(i,j) : (B_i \cap (\alpha \cdot B_j) \neq \emptyset)$ **and** $((\alpha \cdot B_i) \cap B_j \neq \emptyset)\}$.*

An important property of overlap graphs, as shown by Miller, Teng, Thurston and Vavasis [15] is that they have small cost sphere separators, i.e., there is a sphere S that divides B into three subsets: B_I, B_E and B_O such that (1) balls from B_I are completely in the interior of S and balls from B_E are in the exterior of S, (2) there exists a constant $1/2 < \delta < 1$ such that $|B_I|$, $|B_E| \leq \delta n$; (3) there are no edges in the overlap graph that connect any ball from B_I with any ball in B_E. (4) $|B_O| = O(\alpha k^{1/d} n^{1-1/d})$. Furthermore, such a separator can be found in random linear time sequentially and in random constant time, using n processors.

A special case of the overlap graph is the density graph (first introduced by Miller and Vavasis [17]). The density condition of an embedding is important for finite difference methods. Let G be an undirected graph and let π be an embedding of its nodes in $\mathbb{R}^d$. We say π is an embedding of G of *density* α if the following inequality holds for all vertices v in G. Let u be the closest node to v. Let w be the farthest node from v that is connected to v by an edge. Then

$$\frac{||\pi(w) - \pi(v)||}{||\pi(u) - \pi(v)||} \leq \alpha.$$

In general, G is an *α-density graph* in $\mathbb{R}^d$ if there exist an embedding of G in $\mathbf{R}^d$ with density α. We will show later that there is a $\Delta(\alpha, d)$ depending only on α and d such that the maximum degree of an α-density graph is bounded by $\Delta(\alpha, d)$.

3 Bounded Radius-Edge Ratio Delaunay Triangulations

Geometrically, the problem of mesh generation is to find a "good" point set (that respects the input domain) which has a triangulation that satisfies certain geometric conditions. Thus mesh generation, explicitly or implicitly, consists of two steps: point generation and triangulation. Because Delaunay triangulations enjoy several important optimality properties, e.g., maximizing the smallest angle in two dimensions, it has been used to triangulate the point set to obtain the final mesh.

We will now establish a connection between some local geometric properties of *DT(P)* with the quality of *P* as a density embedding of *DT(P)*. Such a connection is a key ingredient to our parallel Delaunay triangulation algorithm.

Definition 3.1 *A Delaunay triangulation has the* **bounded radius-edge ratio property** *with parameter $C \geq 1$ if the largest ratio of the circum-radius to the smallest edge over all of its triangles (or polygon) is bounded by C.*

In two dimensions, if a Delaunay triangulation has bounded radius-edge ratio property with parameter C then its smallest angle is at least $sin^{-1}(1/(2C))$. In three dimensions, Delaunay based mesh generators all implicitly maintain bounded radius-edge ratio. In fact, no stronger condition is known for such meshes. The problem of generating a point set in 3-dimension with a Delaunay triangulation of a bounded minimal dihedral angle [6] is still of interest. (Mitchell and Vavasis's construction is not Delaunay based, and places mesh points on corners of rectangular boxes, so small perturbations of the nodes introduces many slivers) Therefore, the bounded radius-edge ratio is a natural condition for mesh generation.

Let P be a point set in $\mathbb{R}^d$. We now show that if *DT(P)* has a bounded radius-edge ratio, then its 1-dimensional skeleton is a density graph, and hence has a bounded degree and a small sphere separator. In the next section, we will use this result to develop an efficient parallel algorithm for finding the Delaunay triangulation. Note that for general triangulations, the bounded radius-edge ratio does not imply the density condition.

Theorem 3.2 *Let P be a point set in $\mathbf{R}^d$. If DT(P) satisfies the bounded radius-edge ratio property with parameter C, then there is a constant α dependent only on d and C such that P is an α-density embedding of DT(P).*

We will prove this theorem in two steps. We examine the set of all tetrahedra incident to a point $p \in P$. Each tetrahedron defines a circumsphere. We assign to each tetrahedron a vector from p to the center of its circumsphere. This vector is referred as the *radius vector* of the tetrahedron. In the first step, we show that tetrahedra whose radius vectors form a small angle have edge lengths with a maximal ratio of C_1. In the second step, we use this fact to show the graph is a density graph for some α which depends on C and C_1. Finally, we show that if *DT(P)* is a density graph then its degree is bounded by some constant dependent only on C and d.

The following lemma shows that if the angle between two radius vectors is small, than the ratio $\frac{\max(R_1,R_2)}{\min(R_1,R_2)} \leq C_1$, for some constant C_1 dependent only on C and d, where R_1 and R_2 are the lengths of the two radius vectors, respectively.

Lemma 3.3 *There is a constant α_0 and a constant C_1 dependent only on C, d such that if the angle between two radius vectors from p is smaller than α_0, then $\frac{R}{r} \leq C_1$, where R is the larger radius and r is the smaller radius.*

Proof: We depict the case in Figure 3. Let α be the angle between the two radius vectors, i.e., the angle defined by O_1, p, and O_2, where O_1 and O_2 are two circum-centers, and we assume that the length r of the radius vector defined by p and O_1 is no longer than the length R of the radius vector defined by p and O_2. Let β be the angle defined by p, O_2 and O_1. Let L be the length of the diameter of the sphere defined by the intersection of the two circum-spheres S_1 and S_2 associated with O_1 and O_2, respectively.

Without loss of generality, we assume that $r = 1$. By an elementary geometric argument, we have $\beta = atan\frac{\sin\alpha}{R-\cos\alpha}$ and $L = 2\sin(\alpha+\beta)$. Notice that as α approaches to 0, β approaches to $\alpha/(R-r)$.

Because S_1 and S_2 are Delaunay tetrahedra, there is no point from P in the interior of O_2. So the length of the longest side of the tetrahedron of O_1 is at most L which is equal to $2\sin(\alpha+\beta)$. Because the radius-edge ratio of the Delaunay triangulation is bounded by C, we know $L > 1/C$. Since L is bounded from below, R can not grow arbitrarily and there exists a constant α_0 and C_1 dependent only on C, d such that if $\alpha < \alpha_0$ then $R/r = R \leq C_1$. □

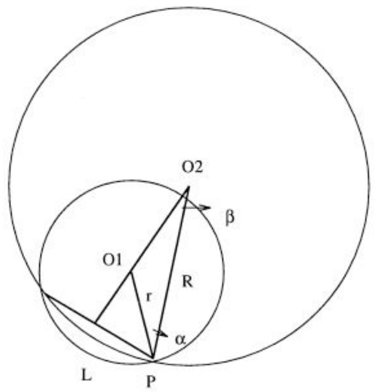

Figure 2: Projection of two intersecting spheres on the plane defined by their radius vectors from P

Lemma 3.4 *If the ratio of the length of vector radii of two tetrahedra in a Delaunay triangulation with radius-edge ratio C is bounded by C_1, than the ratio of the largest edge E and the smallest edge e in the union of the edges is bounded by $2C_1C$.*

Proof: We break up the proof into the following set of cases:

- If the shortest edge and the longest edge are in the same tetrahedron, then the ratio is bounded by $2C$ since $\frac{E}{e} = \frac{E}{r}\frac{r}{e}$.
- If the shortest edge e is in the tetrahedron with smaller vector radius and the longest edge E is in the another tetrahedron, then $\frac{|E|}{|e|} = \frac{E}{R}\frac{R}{r}\frac{r}{e} \leq 2CC_1$
- If the longest edge E is in the tetrahedron with smaller vector radius and the shortest edge e is in another tetrahedron, then

Therefore, $\frac{E}{e} \leq 2C_1C$. □

Two tetrahedra are *neighboring* if they share a common edge. The following lemma bounds the ratio of the length of the longest edge to the shortest edge in the union of two neighboring tetrahedra.

Lemma 3.5 *If e and E belong to neighboring tetrahedra in a Delaunay triangulation with radius-edge ratio C, then the ratio of their length is $4C^2$.*

Proof: If ge is an edge common to the two polygons, then $\frac{E}{e} = \frac{E}{ge}\frac{ge}{e} \leq 4C^2$. □

From now on, we use $D = \max(2CC_1, 4C^2)$. Now, we show the graph is a density graph. Assume S is a very small sphere centered around a point $p \in P$. We cover S by a collection of circular patches such that their cone angle is less than α_0. The following lemma is a folklore,

Lemma 3.6 *There is a constant K dependent only on α_0 and d such that there is a cover of the unit sphere in $\mathbb{R}^d$ with no more than K circular patches whose angle is equal to α_0.*

Now we cover S according to Lemma 3.6. Each radius vector intersects the sphere in at least one cone patch (the patches are not necessarily disjoint, so it could intersect more than one patch). Assign to each radius vector a label which corresponds to one of the patches it intersects. If two vector radii have the same label, then by Lemmas 3.4 and 3.5, the maximal ratio of the edges belonging to the two tetrahedra is bounded by D.

Let e be the shortest edge out of P. Let E be another edge out of P. There is a path between e and E through edges that belong to neighboring tetrahedra incident to P. In each transition of the path, the edge lengths can grow by at most a factor of D.

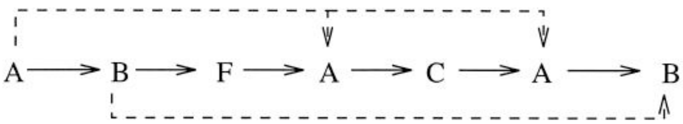

Figure 3: Contracting the dotted edges results in a finite path, where the ratio between neighboring elements is bounded by a constant.

We assign a label to each edge in the path. The label indicates the patch that the edge's radius vector intersects. If a label appears more than once in the path, we can "erase" all labels between last and first appearance of the label, and instead use the ratio information forced by the label, which is D. This process would leave us with a constant number of edges (because no label repeats), and therefore a constant bound on the ratio of e to E. Therefore, the graph is a density graph and P is such a density embedding.

The following lemma shows that if a graph is a density graph then its degree is bounded.

Lemma 3.7 *There is a constant D_G dependent only on α and d such that if a graph is a density graph with parameter α then its degree is bounded by D_G.*

Proof: Let's look at all the neighboring nodes of P. They are contained in some sphere with radius αe around P, where e is the smallest edge from P. Let Q be one of P's neighbors, then Q has an edge of length at least e, so Q's nearest neighbor is no closer than $\frac{e}{\alpha}$. Therefore, for each neighbor Q of P, we define a sphere around it of radius $\frac{e}{2\alpha}$. Those spheres cannot intersect, and there can be at most a finite number of them, so the graph is of finite degree. □

4 Parallel Delaunay Triangulation

We now describe our parallel algorithm for constructing the Delaunay triangulation of a point set that satisfies the bounded radius-edge ratio property. Our construction has two major steps: (1) Finding a supergraph of *DT(P)* such that the supergraph is also of bounded degree; (2) Recognizing the edges of the supergraph that belong to *DT(P)*. For better illustration, we start with the second step first and then show how to find such a supergraph.

4.1 Convex Hulls and Delaunay Triangulations

The following lemma shows how to find the Delaunay triangulation when a supergraph of the Delaunay triangulation is given.

Lemma 4.1 *Let P be a point set in $\mathbf{R}^d$ and assume that DT(P) has degree bounded by D_1. Let G be a supergraph of DT(P) of maximum degree $D_2 \geq D_1$. Then we can compute DT(P) from G in $O(T_{CH,d}(D_2))$ using n processors, where $T_{CH,d}(m)$ is the sequential time for finding the convex hull of m points in d dimensions.*

Therefore, given G with a constant degree bound, *DT(P)* can be found in $O(1)$ time using n processors.

In the proof of Lemma 4.1 we exploit the geometric relationship between Delaunay triangulations and convex hulls. For each point p in $\mathbf{R}^d$, let *lift(p)* = $(p, ||p||^2)$, where $||p||$ is the norm of vector given by p. Geometrically, *lift* maps point p vertically onto the paraboloid $x_{d+1} = \sum_{i=1}^{d} x_i^2$ (see Figure 4.1).

Brown [4] and Edelsbrunner and Seidel [8] proved the following result.

Lemma 4.2 *Suppose $P = \{p_1, ..., p_n\}$ is a point set in $\mathbb{R}^d$. Let Q = lift(P). Then DT(P) is isomorphic to the lower convex hull of Q.*

It follows directly from the lemma above that the problem of finding the Delaunay triangulation in d dimensions can be reduced to the problem of finding the convex hull in $(d + 1)$ dimensions. Instead of using Lemma 4.2 directly, we use it to relate a Delaunay triangulation in d dimensions to a set of small convex hull problems in d dimensions.

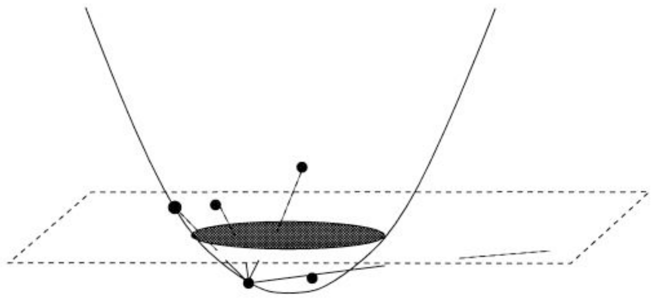

Figure 4: Lift transformation and local convex hull

It follows from Lemma 4.2, that (p_i, p_j) is an edge in *DT(P)* only if (q_i, q_j) is on the convex hull (lower hull) of Q. One way to recognize the set of edges with endpoint p_i that belong to *DT(P)* is to recognize the set of edges with endpoint q_i that belong to the convex hull of Q.

Lemma 4.3 *Suppose we take a hyper-plane H in* $\mathbf{R}^{d+1}$ *close enough to* q_i *to separate* q_i *and* $Q - \{q_i\}$. *Let* q'_j *be the intersection of* q_iq_j *and H. Then* q_iq_j *is an edge on the convex hull of Q iff* q'_j *is on the convex hull of* $\{q'_j : j \neq i\}$.

Lemma 4.3 yields another way to find *DT(P)*: Lift P to the paraboloid to obtain Q and solve the n convex hull problems (one for each point in Q) in d dimensions. The convex hull problem for q_i determines the set of convex hull edges of Q with q_i as an endpoint, and hence, the set of edges of the Delaunay Triangulation of P with p_i as an endpoint.

Now suppose G is a supergraph of *DT(p)*. To determine the set of edges with endpoint p_i of *DT(P)*, we simply lift the graph neighbors of p_i and perform a d dimensional convex hull construction (as in Lemma 4.3). We can perform such local operation independently for all points in parallel. Therefore, if the maximum degree of G is D_2, we can compute *DT(P)* from G in $O(T_{CH,d}(D_2))$ using n processors, completing the proof of Lemma 4.1.

4.2 Density graphs as supergraphs for DT

We first define the α-*density graph* of a point set P, denoted by $DG_\alpha(P)$. Let B_i be the nearest neighbor ball of p_i (i.e., the ball whose center is p_i and whose radius is equal to the distance between p_i to its nearest neighbor in P). The α-density graph of P is the restriction of the α-overlap graph (See section 2) for this neighborhood system to a density graph — that is, all edges that are longer than α times the nearest neighbor are removed from the α-overlap graph.

Notice that the α-density graph of P is the supergraph of any α-density embedding of a graph that uses P as its vertices. Therefore, if *DT(P)* satisfies the bounded radius-edge ratio property then by Theorem 3.2 there exists a constant α, depending only on d and the radius ratio, such that *DT(P)* is a subgraph of $DG_\alpha(P)$. Notice also that $DG_\alpha(P)$ has bounded degree as well.

Our strategy for computing *DT(P)* when P has the bounded radius-edge ratio property is to first compute $DG_\alpha(P)$ and then find the set of edges of $DG_\alpha(P)$ that belongs to *DT(P)*. In the next section, we show that the first step can be performed efficiently in parallel.

Theorem 4.4 *Let P be a point set in three dimensions. If DT(P) has bounded radius-edge ratio property, then DT(P) can be found in* $O(n \log n)$ *time sequentially and in randomized parallel* $O(\log n)$ *time using n processors on an EREW PRAM with unit time scan primitive. The result can be generalized to any fixed dimension.*

4.3 Constructing α-density graphs in parallel

The basic approach is the geometric separator based divide and conquer. $DG_\alpha(P)$ has a sphere separator of cost $O(\alpha n^{1-1/d})$ [15] and we can find a small sphere separator of $DG_\alpha(P)$ without first constructing $DG_\alpha(P)$.

The first step is to find the neighborhood system $B = \{B_1, B_2, ..., B_n\}$ of P. We can use the algorithm of Frieze, Miller and Teng [9] or the algorithm of Callahan [5]. Both algorithms take $O(\log n)$ time, using n processors. The first algorithm is randomized but as it itself uses the geometric separator based divide-and-conquer construction it may be more suitable from the software development viewpoint.

Once the nearest neighborhood system is found, we apply the following divide and conquer algorithm:

Algorithm α-density graph construction:

1. If the size of the point set is less than some constant, construct the α-density graph directly.
2. Using the geometric sphere separator algorithm [15] find a sphere separator S which divides B into three subsets: B_I, B_E and B_O, where (1) Vertices corresponding to B_O consist a separator of the α-density graph of P that divides P into two three sets P_I and P_E and P_O, respectively. (2) $|B_O| \leq O(\alpha n^{1-1/d})$ and (3) $|B_I|, |B_E| \leq (1-(d+1)/(d+2)+\epsilon)n$.
3. Recursively find the α-density graph G_I of $P_I \cup P_O$;
4. Recursively find the α-density graph G_E of $P_E \cup P_O$;
5. Return the graph G defined by the union (after deleting the multiple edges) of G_I and G_E.

Lemma 4.5 *The graph G computed by the algorithm above is the α-density graph of P and the algorithm above runs in randomized $O(n \log n)$ time sequentially and in random $O(\log n)$ parallel time using n processors.*

Proof: The key observation is that we can find a small vertex separator of a density graph from the neighborhood system, without having to construct the density graph. □

4.4 Remarks

If the Delaunay Triangulation is known to be a subgraph of a k-overlap graph, we can find the supergraph efficiently, and as before, solve n convex-hull problems in d dimensions (rather than the traditional reduction to $d+1$ dimensions). But, where as for the density graph case the subproblems were of size bounded by a constant, here some subproblem might be of size as large as n. Nevertheless, since the total number of edges in a k-overlap graph is $O(n)$, the sum of the subproblems sizes is linear. This gives a reduction from the Delaunay triangulation in d dimensions to a convex hull in the same dimension, since the following holds: $\sum_i T_{CH,d}(s_i) < T_{CH,d}(\sum_i s_i) = T_{CH,d}(n)$.

4.5 Random Points

We now show that our parallel Delaunay triangulation algorithm can be used for a "random" point set P of n points in d dimensions. Formally, we use the homogeneous Poisson point process of intensity one as our probabilistic model. This standard model is characterized by the property that the number of points in a region is a random variable that depends only on the d-dimensional volume of the region [13, 12, 3, 22]. In this model,

- The probability of exactly k points appearing in any region of volume V is $e^{-V}V^k/k!$.
- The conditional distribution of points in any region given that exactly k points fall in the region is joint uniform.

We now show that if P is a set of n random points in $\mathbb{R}^d$, then $DT(P)$ can be found in $(\log^{d/2} n)$ time, using n processors. We use the following observation [22] that there is a constant c dependent only on d such that with high probability (e.g., $1 - 1/n$), $DT(P)$ is a subgraph of the $c \log n$-nearest neighborhood graph of P.

To find $DT(P)$ in parallel, we first apply the parallel k-nearest neighbor algorithm of [9] to find the $c \log n$-nearest neighborhood graph of P in $O(\log n)$ parallel random time using n processors. Then we apply Lemma 4.1 to find the $DT(P)$, where we need to solve n convex hull problem of size $O(\log n)$ in d dimensions.

5 Point Set Generation

We now present an efficient parallel algorithm for generating a point set for a simple input domain, with the following properties: (1) The point set density is bounded by the local feature size. This is the optimal point density for well shaped meshes [18]. (2)The Delaunay triangulation of the point set has the bounded radius-edge ratio property.

The local feature size assigns to each point in the domain (not only input points) its distance to the second closest input feature. Thus, it captures the local input density. The function is also a Lipschitz function with a constant 1.

Definition 5.1 *A function f is Lipschitz with constant C if for any two points x, y in the domain* $|f(x) - f(y)| \leq C||x - y||$

Definition 5.2 *A point set S is f-spaced if:*

- *There exists a Lipschitz function f such that for any two points $P, Q \in S$ we have* $||P - Q|| \leq \min(f(P), f(Q))$.
- *The set of point S is maximal with respect to the above function f — that is, no points can be added to the set S without violating the first property.*

Theorem 5.3 *An f-spaced point set, with f α-Lipschitz where $\alpha < 1$, has a bounded radius-edge ratio Delaunay triangulation, with constant $\frac{1}{1-\alpha}$.*

Proof: Let's look at a Delaunay triangle, with its accompanying circumscribed sphere of radius R. Let the smallest edge of the triangle have length l. Therefore, one of the triangle points P has that $f(P) \leq l$. The value of the function f at the center of the sphere is therefore smaller then $l + \alpha R$. Now, if $l + \alpha R < R$ then we could add the center to S, which contradicts maximality. Therefore, $R \leq l + \alpha R$, or, $\frac{R}{l} \leq \frac{1}{1-\alpha}$. $\square$.

A similar theorem is true for an f-spaced point set which is an approximation to the maximal. We now show how to generate this point set. We use Mitchell and Vavasis oct-tree algorithm as our start point. The cell size in Mitchell and Vavasis oct-tree is an approximation to the local feature size, but the constant can be fairly big. Parallelization of the algorithm cause an increase of that constant [2]. The idea is to use the oct-tree as an initial approximation for the points density, and then to perform smoothing and filtering to obtain a density closer to optimal.

Algorithm sketch

- Apply the three dimensional oct-tree algorithm to approximate the local feature size.
- In each cell, lay a constant number of points, and derive a better bound of the local feature size by searching constant number of nearby cells.
- Create a graph over the nodes, by connecting two nodes if the distance between them is larger than some constant times the local feature size of either.
- Obtain a maximal independent set of the graph.

This is a simple sketch of the algorithm. In the presence of boundary faces and edges, a 2-dimensional version of the algorithm will have to be run first on the edges and faces of the input.

6 Final Remarks

We have given a simple and efficient parallel algorithm to compute the Delaunay triangulation in d dimensions, for point sets with bounded radius-edge ratio, and for random point sets. This is the first parallel construction for more than 2 dimension we know of. We have also demonstrated that point sets with these properties play a crucial part in the important problem of mesh generation and therefore that the bounded radius-edge ratio is a natural restriction. We have shown as well that this condition implies that the Delaunay graph is a density graph. Our point generation algorithm and our Delaunay triangulation algorithm provide two important subroutines for parallel three dimensional mesh generation.

References

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

[2] M. Bern, D. Eppstein, and S.-H. Teng. Parallel construction of quadtrees and quality triangulations. In *Workshop on Algorithms and Data Structures*, 1993.

[3] M. Bern, D. Eppstein, and F. Yao. The expected extremes in a delaunay triangulation. *International Journal of Computational Geometry & Applications*, 1(1):79–91, 1991.

[4] K. Q. Brown. Geometric transforms for fast geometric algorithms. Ph.D dissertation, Carnegie-Mellon University, 1980.

[5] Paul B. Callahan and S. Rao Kosaraju. A Decomposition of Multi-Dimensional Point-Sets with Applications to k-Nearest-Neighbors and n-Body Potential Fields. Procedings of the 24th ACM Symposium on Theory of Computing, 1992, 546-556.

[6] T. K. Dey, C. L. Bajaj and K. Sugihara. On Good Triangulations in Three Dimensions. *Proceedings ACM symposium on solid modeling and CAD/CAM applications*, 1991, 431-441.

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

[8] H. Edelsbrunner and R. Seidel. Voronoi diagrams and arrangements. *Discrete. Comput. Geome.*,1 (1986) 25-44.

[9] A. M. Frieze, G. L. Miller, and S.-H. Teng. Separator based divide and conquer in computational geometry. In *Proceedings of the 1992 ACM Symposium on Parallel Algorithms and Architectures*, pages 420–430, 1992.

[10] J. R. Gilbert, G. L. Miller, and S.-H. Teng. Geometric mesh partitioning and nested dissection. In *12th Householder Symposium on Numerical Algebra, Plenary talk*, June 1993.

[11] J. R. Gilbert, G. L. Miller, and S.-H. Teng. Geometric mesh partitioning: Implementation and experiments. Technical report, Xerox Palo Alto Research Center, to appear 1993.

[12] R. M. Karp and J. M. Steele. *Probabilistic analysis of heuristics*. Wiley, New York, 1985.

[13] R. E. Miles. On the homogeneous planar poisson point process. *Mathematical Biosciences*, 6:85–127, 1970.

[14] Gary L. Miller and Dafna Talmor. Mesh generation using well spaced point sets. Technical Report, Carnegie Mellon University, CMU-CS 94-143

[15] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. Finite element meshes and geometric separators. to appear 1994.

[16] G. L. Miller, S.-H. Teng, W. Thurston, and S. A. Vavasis. Automatic mesh partitioning. In A. George, J. Gilbert, and J. Liu, editors, *Sparse Matrix Computations: Graph Theory Issues and Algorithms*, IMA Volumes in Mathematics and its Applications. Springer-Verlag, 1993.

[17] Gary L. Miller and Stephen A. Vavasis. Density graphs and separators. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 331–336, 1991.

[18] S. A. Mitchell and S. A. Vavasis. Quality mesh generation in three dimensions. In *Proceedings of the ACM Computational Geometry Conference*, pages 212–221, 1992. Also appeared as Cornell C.S. TR 92-1267.

[19] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry: An Introduction*. Texts and Monographs in Computer Science. Springer-Verlag, 1985.

[20] J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. In *4th Symp. Discrete Algorithms*, pages 83–92. SIAM, 1993.

[21] H. D. Simon. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering*, 2(2/3):135–148, 1991.

[22] S.-H. Teng and F. F. Yao. k-nearest-neighbor clustering and percolation theory. In *International Symposium on Algorithms and Computation*, page to appear, 1993.