# WHEN AND WHY RUPPERT'S ALGORITHM WORKS

Gary L. Miller[1] Steven E. Pav[2] Noel J. Walkington[2]

[1] *Carnegie Mellon University, Pittsburgh, PA 15213 glmiller@cs.cmu.edu*
[2] *Carnegie Mellon University, Pittsburgh, PA 15213 [spav|noelw]@andrew.cmu.edu*

## ABSTRACT

An "adaptive" variant of Ruppert's Algorithm for producing quality triangular planar meshes is introduced. The algorithm terminates for arbitrary Planar Straight Line Graph (PSLG) input. The algorithm outputs a Delaunay mesh where no triangle has minimum angle smaller than $26.45°$ except "across" from small angles of the input. No angle of the output mesh is smaller than $\arctan\left[(\sin\theta^*)/(2-\cos\theta^*)\right]$ where $\theta^*$ is the minimum input angle. Moreover no angle of the mesh is larger than $137.1°$. The adaptive variant is unnecessary when $\theta^*$ is larger than $36.53°$, and thus Ruppert's Algorithm (with concentric shell splitting) can accept input with minimum angle as small as $36.53°$. An argument is made for why Ruppert's Algorithm can terminate when the minimum output angle is as large as $30°$.

**Keywords: mesh generation, Ruppert's Algorithm, computational geometry, triangular**

## 1. INTRODUCTION

The Delaunay Refinement Algorithm, first described by Ruppert, accepts a set of points and a set of segments, augments the point set with Steiner points, and returns the Delaunay Triangulation of the augmented set. For suitable input, the triangulation conforms to the input, has no angle smaller than some parameterizable $\kappa$ (which is no larger than $\arcsin\frac{1}{2\sqrt{2}} \approx 20.7°$), and exhibits "good grading," *i.e.*, short edges in the triangulation are attributable to nearby input features which are close together. The number of triangles in the output is within a constant of optimal [1].

The algorithm has the advantage of being relatively easy to state and implement, and has been the object of great scrutiny and interest. Since its introduction, the algorithm and the analysis of the algorithm have been improved and modified: the class of known acceptable input has been expanded [2]; a variant algorithm has been developed to handle small input angles [3]; the algorithm has been adapted to accept curved input [4]; it also has been generalized to higher dimensions [2, 5, 6, 7].

Ruppert's original analysis required that no input segments meet at acute angles, and guaranteed that no angle in the output was smaller than a parametrizable $\kappa < \arcsin\frac{1}{2\sqrt{2}}$. As $\kappa \nearrow \arcsin\frac{1}{2\sqrt{2}}$, the proved bound on the number of Steiner Points approaches infinity [1], though this behaviour is *not* seen experimentally; rather, the Delaunay Refinement Algorithm is often run with $\kappa$ as great as $\pi/6$ or greater without diverging. The input condition has been relaxed to a $\pi/3$ lower bound on input angles [3, 5]. The algorithm has been observed to terminate on some input with smaller (in some cases much smaller) input angles.

Shewchuck demonstrated an alteration of the algorithm, the so-called "Terminator," which accepts input with arbitrary minimum angle, $\theta^*$, producing Delaunay meshes with no output angle smaller than $\arcsin\left[\sin\left(\frac{\theta^*}{2}\right)/\sqrt{2}\right]$. This variant is adaptive in the sense that it leaves some small angles in the output mesh, while most angles are larger than $\arcsin\frac{1}{2\sqrt{2}}$. The location of the small output angles cannot be determined very much beyond the statement that they are "near input angles less than $\ldots 60°$." Moreover, the

Support provided in part by National Science Foundation Grants DMS–0208586 and CCR–9902091. This work was also supported by the NSF through the Center for Nonlinear Analysis.

analysis of this scheme comes without grading guarantees, and thus no optimality claim [3].

We here demonstrate an alteration of the algorithm which outputs meshes where all output angles are greater than $\arcsin 2^{-7/6} \approx 26.45^\circ$, except those whose shortest edge is "opposite" an input angle $\theta < 36.53^\circ$; in this case, the output angle is no less than $\arctan\left(\frac{\sin\theta}{2-\cos\theta}\right)$. Moreover, in spite of the potential of arbitrarily small output angles, this algorithm can guarantee that no output angle is *larger* than around $\pi - 2\arcsin\frac{\sqrt{3}-1}{2} \approx 137.1^\circ$. In this sense the algorithm contrasts favorably with the Terminator, which has no upper bound other than the naïve one of $\pi - 2\arcsin\left[\sin\left(\frac{\theta^*}{2}\right)/\sqrt{2}\right]$, which deteriorates when $\theta^*$ is small. Moreover, our algorithm comes with grading and optimality guarantees, and is fairly simple.

In the case where $\theta^* \geq 36.53^\circ$, our analysis shows that the variant algorithm is unnecessary, and that Ruppert's original algorithm with circular shell splitting comes with the same output and optimality guarantees.

In this work we employ the strategy of Shewchuk [2], *i.e.*, termination is proved without showing good grading. This is done since a relatively accessible and complete proof of the "termination-only" result may be given in the limited amount of available space. The proof of good-grading is quite a bit more involved [8].

## 2. THE MESHING PROBLEM

The meshing problem is described in terms of the input to the algorithm and the expected conditions on the output. The input to the mesher is defined as follows:

**Assumption 2.1 (Input).** The input to the meshing problem consists of a finite set of points, $\mathcal{P} \subseteq \mathbb{R}^2$, and a set of segments $\mathcal{S}$ such that

(a) the two endpoints of any segment in $\mathcal{S}$ are in $\mathcal{P}$,
(b) any point of $\mathcal{P}$ intersects a segment of $\mathcal{S}$ only at an endpoint,
(c) two segments of $\mathcal{S}$ meet only at their endpoints, and
(d) the boundary of the convex hull of $\mathcal{P}$ is the union of segments in $\mathcal{S}$.

Let $\Omega$ denote the convex hull of the input, and let $0 < \theta^* \leq \pi/3$ be a lower bound on the angle between any two intersecting segments of the input.

Items (a)-(c) characterize $(\mathcal{P}, \mathcal{S})$ as a Planar Straight Line Graph (PSLG); item (d) can always be satisfied by augmenting an arbitrary PSLG which does not satisfy it with a bounding polygon (typically a rectangle). The restriction that $\theta^* \leq \pi/3$ is merely for convenience; asserting a larger lower bound does not give any better results.

**Assumption 2.2 (Output).** The algorithm outputs sets of points, segments, triangles, $\mathcal{P}', \mathcal{S}', \mathcal{T}'$, respectively, satisfying:

(a) **Complex:** The output collectively forms a simplicial complex, *i.e.*, $\{\emptyset\} \cup \mathcal{P}' \cup \mathcal{S}' \cup \mathcal{T}'$ is closed under taking boundaries, and under intersection.
(b) **Delaunay:** Each triangle of $\mathcal{T}'$ has the Delaunay property with respect to $\mathcal{P}'$.
(c) **Conformality:** $\mathcal{P} \subseteq \mathcal{P}'$, and for every $s \in \mathcal{S}$, $s$ is the union of segments in $\mathcal{S}'$.
(d) **Quality:** There are few or no "poor-quality" triangles in $\mathcal{T}'$.
(e) **Cardinality:** Few Steiner points have been added, *i.e.*, $|\mathcal{P}' \setminus \mathcal{P}|$ is small.

One passable definition of item (d) is that there are some reasonably large constants $0 < \alpha \leq \omega \leq \frac{\pi+\alpha}{4}$ such that for every triangle $t \in \mathcal{T}'$, no angle of $t$ is smaller than $\alpha$ or larger than $\pi - 2\omega$. However, such a guarantee is not consistent with conformality of the triangulation (item (c)) when the input contains angles less than $\alpha$. A weaker definition is that most triangles satisfy the above condition, and those that do not (a) are describably near an input angle of size $\theta$, (b) have no angle smaller than $\theta - \mathcal{O}\left(\theta^2\right)$, and (c) have no angle larger than $\pi - 2\omega$.

## 3. THE ALGORITHM

We describe a whole class of algorithms, which we collectively refer to as "the" Delaunay Refinement Algorithm. This class contains Ruppert's original formulation [1], as well as the incremental version [5].

We suppose that the algorithm maintains a set of "committed" points, initialized to be the set of input points, $\mathcal{P}$. The algorithm also maintains a set of "current" segments, initialized as the input set, $\mathcal{S}$. The algorithm will "commit" points to the set of committed points. At times the algorithm will choose to "split" a current segment; this is achieved by removing the segment from the set of current segments, adding the two half-length subsegments which comprise the segment to the set of current segments, and committing to the midpoint of the segment. The word "midpoint" should be taken to mean one of these segment midpoints for the remainder of this work, to distinguish them from the other kind of Steiner Point, which will be called "circumcenters."

The algorithm has two high-level operations, and will continue to perform these operations until it can no longer do so, at which time it will output the committed points, the current segments and the Delaunay

Triangulation of the set of committed points. For convenience, we say that a segment is "encroached" by a point $p$ if $p$ is inside the diametral circumball of the segment. Then the two major operations are as follows:

> (CONFORMALITY) If $s$ is a current segment, and there is a committed point that encroaches $s$, then split $s$.
> (QUALITY) If $a, b, c$ are committed points, the circumcircle of the triangle $\Delta abc$ contains no committed point, triangle $\Delta abc$ has an angle smaller than the global *minimum output angle*, $\kappa$, and the triangle's circumcenter, $p$ is in $\Omega$, then attempt to commit $p$. If, however, the point $p$ encroaches any current segment, then do not commit to point $p$, rather in this case split one, some, or all of the current segments which are encroached by $p$.

It should be clear that if the algorithm terminates then every segment of the set $\mathbb{S}$ has been decomposed into current segments, none of which are encroached by committed points, and thus have the Delaunay property with respect to the final point set, and are thus present in the output Delaunay Triangulation. The algorithm clearly never adds any points outside $\Omega$. It is simple to show that if the algorithm terminates, no triangle in the Delaunay Triangulation has an angle smaller than the minimum output angle $\kappa$, though we omit the proof [8].

The Adaptive Delaunay Refinement Algorithm substitutes operation (QUALITY) with the following operation (QUALITY′):

> (QUALITY′) If $a, b, c$ are committed points, the circumcircle of the triangle $\Delta abc$ contains no committed point, $\angle acb < \hat{\kappa}$, the circumcenter, $p$, of the triangle is inside $\Omega$ and either (i) both $a, b$ are midpoints on distinct nondisjoint input segments, sharing input endpoint $x$, and $\angle axb > \pi/3$, or (ii) $a, b$ are not midpoints on adjoining input segments, then attempt to commit $p$. If, however, the point $p$ encroaches any current segment, then do not commit to point $p$, rather in this case split one, some, or all of the current segments which are encroached by $p$.

In summary, the algorithm removes angles smaller than $\hat{\kappa}$ except when the opposite edge spans a small angle in the input, in which case the small output angles are ignored. For this variant we call $\hat{\kappa}$ the *output angle parameter*; the output mesh may well contain angles smaller than $\hat{\kappa}$. We will let $\alpha$ be the minimum angle in the output mesh.

The heuristics involved with determining which operation to perform when and on which segment or poor-quality triangle are not relevant to our discussion. This is not to say that they might not affect ease of implementation, running time, cardinality of the final set of committed points, parallelizability, etc. A common heuristic (and the one chosen by Ruppert and others) is to prefer conformality operations over quality operations, which likely results in a smaller output, and which simplifies detecting that a circumcenter is outside of $\Omega$. A description of a member of this class of algorithms would have to include some discussion of how to figure out which current segments are encroached, which triangles are suitable for removal via the quality operation, how to deal with degeneracy, etc. We do not concern ourselves with these details (though see [9, 10, 11, 5, 12, 2, 13]).

## 3.1 When is Adaptivity Necessary?

We here make the claim that the Delaunay Refinement Algorithm is as good as its adaptive variant when the latter is used with a small output angle parameter $\hat{\kappa}$. The claim is formalized as follows:

*Claim* 3.1. Suppose that we can guarantee that if the Adaptive Delaunay Refinement Algorithm is run with output angle parameter $\hat{\kappa}$, on any appropriate input with minimum input angle $\theta^*$, that (a) the algorithm terminates, (b) no angle of the output mesh is smaller than $\hat{\kappa}$, and (c) no angle is larger than $\pi - 2\omega$.

Then if the Delaunay Refinement Algorithm is run on any appropriate input with minimum input angle $\theta^*$, using output angle parameter $\kappa = \hat{\kappa}$, then (a) the algorithm terminates, (b) no angle of the output mesh is smaller than $\kappa$, and (c) no angle is larger than $\pi - 2\omega$.

*Proof.* The Adaptive Delaunay Refinement Algorithm only attempts to remove a Delaunay triangle if it has minimum angle smaller than $\hat{\kappa}$. Moreover, it produces meshes with no angle smaller than $\hat{\kappa}$. Then the (QUALITY′) operation could be rewritten as follows:

> (QUALITY′) If $a, b, c$ are committed points, the circumcircle of the triangle $\Delta abc$ contains no committed point, $\angle acb < \hat{\kappa}$, and the circumcenter, $p$, of the triangle is inside $\Omega$ then attempt to commit $p$. If, however, the point $p$ encroaches any current segment, then do not commit to point $p$, rather in this case split one, some, or all of the current segments which are encroached by $p$.

This is the same as the operation (QUALITY) of the Delaunay Refinement Algorithm. □

By "appropriate," we refer to the fact that, as stated, both algorithms require some added assumption about edge lengths (*cf.* Assumption 4.2). The restriction can be removed if splitting on concentric shells is used to put input into the required form on an "as-needed" basis, as argued in Section 9.

Thus we will first examine the adaptive variant, then use the results to analyze the regular Delaunay Refinement Algorithm.

The analysis that follows should be read with a tacit understanding that it can be applied to the Delaunay Refinement Algorithm as well, if $\kappa$ is set propertly. For example, it will be shown that if an input with $\theta^* \approx 36.53°$ conforms to Assumption 4.2, then the Adaptive Delaunay Refinement Algorithm with $\hat{\kappa} = 26.45°$ will terminate leaving no angle in the output mesh smaller than $\hat{\kappa}$, and no angle larger than $\pi - 2\hat{\kappa}$. Then we can immediately claim that the Delaunay Refinement Algorithm (*i.e.*, Ruppert's Algorithm) with $\kappa = 26.45°$ will also terminate on the same input, and with the same grading guarantees.

So the adaptive variant is only necessary if $\theta^*$ is small, say smaller than about $36.53°$. When $\theta^*$ *is* small, the adaptive variant will remove small angles where this is possible, *i.e.*, away from small input angles.

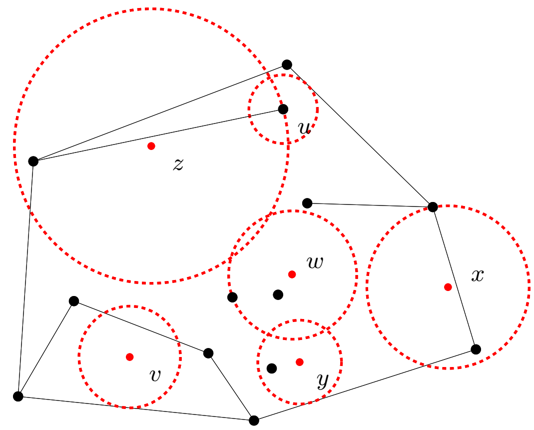



**Figure 1**: For a number of points in the plane, the local feature size with respect to the given input is shown. About each of the points $u, v, w, x, y, z$ is a circle whose radius is the local feature size of the center point. The point $u$ is an input point.

## 4. PRELIMINARIES

Some preliminary definitions and results are essential to the exposition. First there is the matter of terminology: if $p$ is a committed point that was the midpoint of a segment, we say this segment is the "parent" segment (or parent subsegment) of $p$; the "radius" of a segment is half its length, while the radius associated with a midpoint is the radius of its parent segment; any segment derived from a segment $s \in \mathcal{S}$ by splitting is a "subsegment" of (or on) $s$; segments in $\mathcal{S}$ which share an endpoint are nondisjoint; distinct nondisjoint segments are said to be "adjoining."

Throughout this work, we let $|x - y|$ denote the Euclidian distance between points $x$ and $y$. For a segment $S$, we let $|S|$ denote the length of the segment. Local feature size is defined in terms of the input, and is the classical definition due to Ruppert:

**Definition 4.1 (Local Feature Size).** For a point $x \in \mathbb{R}^2$, the *local feature size* at $x$, relative to an input PSLG, $(\mathcal{P}, \mathcal{S})$, is the minimum $r$ such that a closed ball of radius $r$ centered at $x$ intersects at least two disjoint features of $\mathcal{P} \cup \mathcal{S}$. The local feature size is a Lipschitz function, *i.e.*, $\text{lfs}(x) \leq |x - y| + \text{lfs}(y)$.

This definition is illustrated in Figure 1.

For the proof we require an extra condition on the input:

**Assumption 4.2.** In addition to those of Assumption 2.1 we make the following assumption:

(a) If $S_1, S_2$ are two adjoining input segments that meet at angle other than $\pi$, then they have the same length modulo a power of two, that is $\frac{|S_1|}{|S_2|} = 2^k$ for some integer $k$.

It is simple to show that this assumption can be satisfied by the addition of no more than $2|\mathcal{S}|$ augmenting points, effectively redefining the input [8]. Later we will argue that Ruppert's strategy of splitting on concentric circular shells obviates this additional assumption [1].

## 5. MIDPOINT-MIDPOINT INTERACTIONS

Ruppert noted that one way his algorithm could fail was due to infinite cascades of segment midpoints each encroaching on an adjoining subsegment; the prescribed cure was concentric shell splitting [1], which puts input into a form which satisfies Assumption 4.2 on an as-needed basis. To simplify the proof, we assume the input satisfies this assumption up-front, then ease the restriction later. In this section we show how this assumption can prevent infinite cascades of midpoints.

The classic result on Ruppert's Algorithm for input satisfying a $\pi/3$ angle condition can be proven with the following purely geometric lemma [8].

**Lemma 5.1.** *Given two rays, $R$ and $R'$ from a point $x$ with angle $\theta$ between them, suppose there is a ball of radius $r$ with center $p$ on ray $R$ such that the ball does not contain $x$ but does contain a point $q$ of $R'$. Then if $\pi/4 \leq \theta < \pi/2$,*

$$\frac{|q - x|}{|p - x|} \leq \frac{|q - x|}{r} < \frac{|q - x|}{|p - q|} \leq 2\cos\theta.$$

Given the $\pi/3$ angle condition, the right hand side of the inequality in the lemma is no greater than 1. Roughly this guarantees that radii do not "dwindle," or in terms of Shewchuk's dataflow diagrams, the midpoint-midpoint loop does not admit a decrease in insertion radius [2].

The following lemma makes the same guarantees, but for input which satisfy Assumption 4.2. The lemma explicitly states that the radii are non-dwindling, though note these are actual segment radii, not Shewchuk's insertion radii, which is also known as nearest neighbor distance. Using the non-dwindling property of segment radii, we will prove termination of the algorithm by demonstrating a lower bound on a segment's radius at time of splitting.

This lemma takes care of the case where a midpoint encroaches a segment on a nondisjoint input feature. In the following sections, we consider another way in which a midpoint can trigger such a segment split, namely via sequences of triangle circumcenters.

**Lemma 5.2.** *Suppose that the input conforms to Assumption 4.2. Let p be the midpoint of a segment which is encroached by a committed point, q, on an adjoining input segment. Let $r_p$ be the radius associated with p, and $r_q$ that of q. Then $r_q \le r_p$, and moreover,*

$$|p - q| \ge 2r_q \sin\frac{\theta}{2},$$

*where $\theta$ is the angle between the two input segments.*

*Proof.* Let $(x, y), (x, z)$ be the two input segments containing, respectively, $p, q$. Let $(a, b)$ be the subsegment of which $p$ is midpoint. Let $(c, d)$ be that for which $q$ is midpoint. Assume that $a$ is closer to $x$ than $b$ is, and assume $c$ is closer to $x$ than $d$ is. It may be the case that $x = a$, or $x = c$.

It is easy to show that, $\log_2 \frac{|x-y|}{|a-b|}$, and $\log_2 \frac{|x-z|}{|c-d|}$ are nonnegative integers. By Assumption 4.2, and since $\theta \ne \pi$, $\log_2 \frac{|x-y|}{|x-z|}$ is an integer. Thus $\log_2 \frac{|a-b|}{|c-d|} = \log_2 \frac{r_p}{r_q} = j$ is also an integer. We wish to show that it is nonnegative.

A geometric argument gives $|x - a| < |x - q| < |x - b|$, so that $|x - a| < |x - c| + r_q < |x - a| + 2r_p$. It then can be shown that $k = \frac{|x-a|}{|a-b|} = \frac{|x-a|}{2r_p}$ is a nonnegative integer, as is, *mutatis mutandis*, $l = \frac{|x-c|}{2r_q}$. Thus

$$2kr_p < (2l + 1)r_q < 2(k + 1)r_p, \quad \text{or}$$
$$2^{j+1}k < (2l + 1) < 2^{j+1}(k + 1), \quad \text{and so}$$
$$\frac{2l + 1}{2^{j+1}} - 1 < k < \frac{2l + 1}{2^{j+1}}.$$

If $j$ is a negative integer, then $2^{j+1}$ is a power of two no greater than 1; in particular it divides any integer, thus $\frac{2l+1}{2^{j+1}} = m$ is an integer. This gives the contradiction that $m - 1 < k < m$ for integer $m, k$. Thus $j$ is a nonnegative integer, or $r_p \ge r_q$.

For the second part, we first show that $|p - q| \ge 2(|x - q| \wedge |x - p|) \sin\frac{\theta}{2}$. We consider the case where $|x - q| \le |x - p|$; the other case follows *mutatis mutandis*.

Let $L = \frac{|x-p|}{|x-q|} \ge 1$. Using the cosine rule on $\Delta xpq$,

$$\begin{aligned} |p - q|^2 &= |x - p|^2 + |x - q|^2 - 2|x - p||x - q|\cos\theta. \\ &= (1 + L^2)|x - q|^2 - 2L|x - q|^2\cos\theta \\ &\ge 2L|x - q|^2 - 2L|x - q|^2\cos\theta \\ &= 2L|x - q|^2(1 - \cos\theta), \end{aligned}$$

where we have used that $1 + L^2 \ge 2L$. Using $L \ge 1$, we obtain $\frac{|p-q|}{|x-p|} \ge \sqrt{2(1 - \cos\theta)}$. It is a simple exercise to show that $2\sin\frac{\theta}{2} = \sqrt{2(1 - \cos\theta)}$ for $\theta \in [0, \pi]$.

Now, clearly $|x - p| \ge r_p \ge r_q$, and $|x - q| \ge r_q$, so the result $|p - q| \ge 2r_q \sin\frac{\theta}{2}$ holds, as desired. □

## 6. CIRCUMCENTER SEQUENCES

We now consider sequences of triangle circumcenter additions.

**Definition 6.1.** A *circumcenter sequence* is a sequence of points, $\{b_i\}_{i=0}^{l-1}$ such that for $i = 1, 2, \ldots, l-1$, $b_i$ is the circumcenter of a triangle in which $b_{i-1}$ is the more recently committed endpoint of an edge opposite an angle less than $\hat{\kappa}$. The point $b_0$ may be an input point or segment midpoint.

For $i = 0, 1, \ldots, l-2$, let $a_i$ be the *other* endpoint of the short edge of which $b_i$ is the more recently committed endpoint. In the case where $a_0, b_0$ are both input points, they are committed simultaneously; we imagine a total order on input points which determines the tie. Both $a_0, b_0$ may be midpoints on distinct, nondisjoint input segments. In this case we assume that the triangle with circumcenter $b_1$ was removed by a (QUALITY′) operation because of a small angle opposite $a_0, b_0$. In particular this means that we assume the angle subtended by the input segments containing $a_0, b_0$ is at least $\pi/3$ in this case.

When talking about such sequences, for $i = 1, 2, \ldots, l-1$, let $\tilde{r}_i$ be the circumradius of the triangle associated with $b_i$. Note that $\tilde{r}_i = |b_i - b_{i-1}| = |b_i - a_{i-1}|$, and that $|a_i - b_i| \ge \tilde{r}_i$. We let $\tilde{r}_0 = |b_0 - a_0|$, *i.e.*, the length of the first short edge.

Note that for a circumcenter sequence, $\{b_i\}_{i=0}^{l-1}$, the points $b_1, b_2, \ldots, b_{l-2}$ are circumcenters which have been committed, $b_{l-1}$ is a circumcenter, though it may be rejected, and $b_0$ may be any type of point. If $b$ is a triangle circumcenter, there is always a circumcenter sequence ending with $b$, although it may be a trivial sequence of two elements. Any circumcenter sequence whose first element, $b_0$, is a triangle circumcenter may be extended to a maximal sequence whose first element is either a segment midpoint or an input point.

The following geometric lemma is the key result which allows us to make the $\arcsin 2^{-7/6}$ output guarantee. It states that only circumcenter sequences longer than a certain length can "turn" around a 180° feature.

**Lemma 6.2.** *Let $S_1, S_2$ be two segments with disjoint interiors on a common line, $L$. Assume that $|S_2| \leq |S_1|$, i.e., $S_2$ is no longer than $S_1$. Let $b_0$ be the midpoint of $S_1$, and let $a_0$ be some other point. Let $\{b_i\}_{i=1}^{l-1}$ be a circumcenter sequence such that $b_{l-1}$ is inside the diametral circle of $S_2$, and such that $b_1$ is the circumcenter of a triangle with edge $(a_0, b_0)$ opposite an angle smaller than $\hat{\kappa}$. Then $l \geq 4$.*

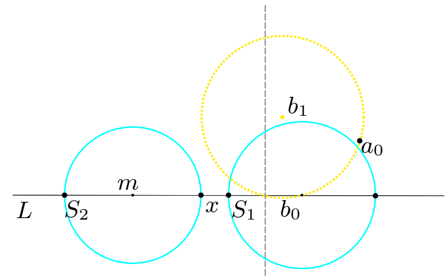
Note that unlike in the regular terminology of circumcenter sequences, this lemma makes no assumptions about which of $a_0, b_0$ was committed first. This is why we have chosen to index the circumcenter sequence from $i = 1$ instead of the usual $i = 0$.

*Proof.* The basic argument is sketched in Figure 2. The point $b_1$ is the circumcenter of a triangle whose circumcircle does not contain the point $x$, which is the endpoint of $S_1$ closer to $S_2$. However, this circumcircle has $b_0$ on it, so $b_1$ must be in the closed halfspace defined by the bisector of $x$ and $b_0$ and which does not contain $x$, as shown in Figure 2(a). Thus $b_1$ cannot be in the diametral circle of $S_2$, which is in the open halfspace on the other side of this bisector. Now let $G$ be the bisector of points $b_1$ and $x$. Point $b_2$ is the center of a circle which does not contain $x$, but has $b_1$ on its boundary, since $b_1$ is one of the vertices of the triangle which $b_2$ is added to remove. Thus $b_2$ must be either on the line $G$, or in the open halfspace defined by $G$ that is closer to the point $b_1$. In Figure 2(b), this is the halfspace to the upper right of $G$.

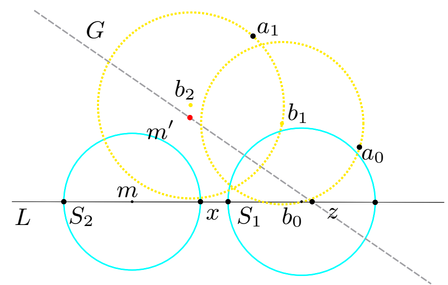
It then suffices to show that the closure of the diametral ball of $S_2$ is contained in the other open halfspace defined by $G$, and thus $b_2$ cannot encroach $S_2$.

Let $z$ be the intersection of $L$ and $G$; take $m$ to be the midpoint of $S_2$, and $m'$ is its projection onto $G$. Let $x'$ be the projection of $x$ onto $G$. Let $y$ be the projection of $b_1$ onto $L$. See Figure 3. The point $x$ is clearly between $m$ and $z$, otherwise $x$ would be in the halfspace closer to $b_1$ than to $x$, a contradiction. Thus $|m - z| = |m - x| + |x - z|$.

By congruency of the three triangles of Figure 3, $\frac{|m-m'|}{|m-z|} = \frac{|x-x'|}{|x-z|} = \frac{|x-y|}{|x-b_1|}$.




(a) $b_1$ does not encroach $S_2$.




(b) $b_2$ does not encroach $S_2$.

**Figure 2**: The head of a circumcenter sequence is shown; the point $b_1$ must be to the right of the bisector of $b_0$ and $x$, and so it cannot encroach $S_2$, which is on the other side of this bisector, as shown in (a). In (b) the bisector of $b_1$ and the point $x$ is shown. Since $b_2$ cannot be closer to $x$ than to $b_1$, and since the diametral circle of $S_2$ is on the opposite side the bisector, $b_2$ cannot encroach $S_2$. In this case, $a_0$ is shown to be outside the diametral circle of $S_1$. This is not a necessary hypothesis for this lemma.

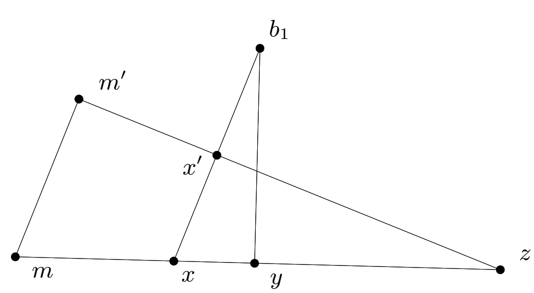



**Figure 3**: The geometric heart of the argument is shown, with three congruent triangles, $\Delta mm'z, \Delta xx'z, \Delta xyb_1$.

Let $r = \frac{|S_2|}{2} \leq \frac{|S_1|}{2}$, by assumption. Since $S_1, S_2$ have disjoint interiors, $|m - x| \geq r$. Then $|m - z| \geq r +$

$|x - z|$, so

$$\begin{aligned} |m - m'| &= \frac{|x - x'|\,|m - z|}{|x - z|}, \\ &\geq \frac{|x - x'|\,(r + |x - z|)}{|x - z|}, \\ &\geq \frac{|x - x'|}{|x - z|} r + |x - x'| \\ &= \frac{|x - y|}{|x - b_1|} r + |x - x'|. \end{aligned}$$

As noted above, $b_1$ is to the right of the bisector of $x$ and $b_0$, so $|x - y| \geq \frac{|x - b_0|}{2} = \frac{|S_1|}{4} \geq \frac{r}{2}$. Note also that $|x - b_1| = 2\,|x - x'|$. Then

$$|m - m'| \geq \frac{r^2}{4\,|x - x'|} + |x - x'|.$$

The right hand side is minimized when $|x - x'| = \frac{r}{2}$, where the right hand side has value $r$. Note, however, that $|x - x'| \geq \frac{\tilde{r}_1}{2} \geq \frac{1}{2\sin\hat{\kappa}} \frac{|S_1|}{4} > \frac{r}{2}$, so the right hand side will be strictly larger than $r$.

That is, $|m - m'| > r$, and thus the distance from $m$ to $G$, which is $|m - m'|$, is greater than the radius of the diametral circle of $S_2$. Then the closed diametral circle of $S_2$ is contained in the open halfspace opposite $b_1$, as desired. □

This lemma allows us to prove a better output angle for the Delaunay Refinement Algorithm. Previous proofs required $2\sin\hat{\kappa} \leq \frac{1}{\sqrt{2}}$; by the lemma, the following proof only requires that $(2\sin\hat{\kappa})^3 \leq \frac{1}{\sqrt{2}}$. A better output angle could be guaranteed if the lemma could be improved; this would have to be via some alternation of the algorithm, as the example of Figure 4 shows the lemma cannot be extended in the naïve setting. We return to this matter later.

Since $\hat{\kappa} < \pi/6$, we can establish a geometric series which gives the following lemma and its corollary. The corollary describes how a segment midpoint which is not caused by a midpoint encroaching the segment is caused by some other midpoint or input point.

**Lemma 6.3.** *Suppose $\{b_i\}_{i=0}^{l-1}$ is a circumcenter sequence. For $i > 0$, let $\tilde{r}_i$ be the circumradius associated with $b_i$. Then for $i = 1, 2, \ldots, l-1$,*

- *$\tilde{r}_{i-1} < 2\tilde{r}_i \sin\hat{\kappa}$ and therefore $\tilde{r}_i < (2\sin\hat{\kappa})^{l-1-i}\tilde{r}_{l-1}$, and*
- *$|b_{l-1} - b_i| < \frac{\tilde{r}_{l-1}}{1-2\sin\hat{\kappa}}$, and $|b_{l-1} - a_i| < \frac{\tilde{r}_{l-1}}{1-2\sin\hat{\kappa}}$.*

*Proof.* By definition, $b_i$ is the circumcenter of a triangle of radius $\tilde{r}_i$, which has a short edge no shorter than $\tilde{r}_{i-1}$ opposite an angle less than $\hat{\kappa}$. By the sine rule, then $2\tilde{r}_i \sin\hat{\kappa} > \tilde{r}_{i-1}$.

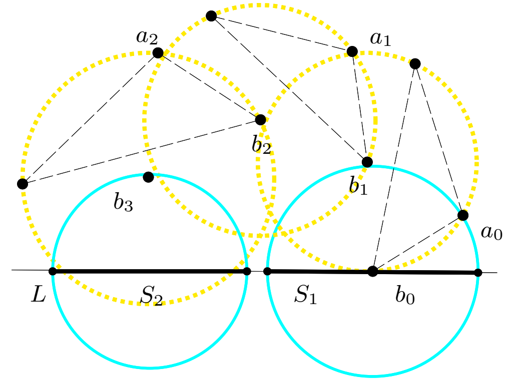



**Figure 4**: A circumcenter sequence, $\{b_i\}_{i=0}^{3}$, is displayed, which shows that Lemma 6.2 cannot be extended. The segments $S_1, S_2$ are shown, with their diametral circles. The points $b_1, b_2, b_3$ are circumcenters of triangles (shown) with an angle smaller than $\pi/6$. The point $b_3$ encroaches $S_2$.

Using this repeatedly gives $\tilde{r}_i < (2\sin\hat{\kappa})^{l-i}\tilde{r}_{l-1}$. Since $2\sin\hat{\kappa} < 1$, we may bound the distance from $b_i$ to $b_{l-1}$ by the geometric series, as follows:

$$\begin{aligned} |b_{l-1} - b_i| &\leq |b_{l-1} - b_{l-2}| + |b_{l-2} - b_{l-3}| + \ldots \\ &\quad + |b_{i+1} - b_i|, \\ &\leq \tilde{r}_{l-1} + \tilde{r}_{l-2} + \ldots + \tilde{r}_{i+1}, \\ &< \tilde{r}_{l-1} + (2\sin\hat{\kappa})\tilde{r}_{l-1} + \ldots \\ &\quad + (2\sin\hat{\kappa})^{l-i-2}\tilde{r}_{l-1}, \\ &< \frac{1}{1 - 2\sin\hat{\kappa}}\tilde{r}_{l-1}. \end{aligned}$$

The bound for $|b_{l-1} - a_i|$ follows since $|b_{i+1} - a_i| = |b_{i+1} - b_i| = \tilde{r}_{i+1}$, and the above analysis suffices. □

**Corollary 6.4.** *Suppose that segment $s_p$ with midpoint $p$ and radius $r$ was split, but the segment was not encroached by a committed point. Then there is some maximal circumcenter sequence $\{b_i\}_{i=0}^{l-1}$ such that $b_{l-1}$ "yielded" to $p$, causing it to be committed. Moreover, $\tilde{r}_i < (2\sin\hat{\kappa})^{l-1-i}\sqrt{2}r_p$, $|p - b_i| \leq \eta r_p$, and $|p - a_i| \leq \eta r_p$, for $i = 0, 1, \ldots, l-1$, with $\eta = 1 + \frac{\sqrt{2}}{1-2\sin\hat{\kappa}}$.*

*Proof.* Since $b_{l-1}$ was the center of an empty circumcircle, but encroached $s_p$, then $\tilde{r}_{l-1} \leq \sqrt{2}r_p$. Using the lemma gives the desired bound on $\tilde{r}_i$. By the lemma, and since $\hat{\kappa} < \pi/6$, $\tilde{r}_i \leq \tilde{r}_{l-1}$. Then

$$\begin{aligned} |p - b_i| &\leq |p - b_{l-1}| + |b_{l-1} - b_i| \leq r_p + \frac{\tilde{r}_{l-1}}{1 - 2\sin\hat{\kappa}} \\ &\leq \left(1 + \frac{\sqrt{2}}{1 - 2\sin\hat{\kappa}}\right) r_p = \eta r_p. \end{aligned}$$

The bound on $|p - a_i|$ follows, *mutatis mutandis*. □

## 7. PROVING TERMINATION

We prove termination not by showing that output mesh edges are well-graded, rather by showing that the algorithm can create no mesh edge smaller than dictated by the minimum local feature size of the input. Towards this end we define

$$\mathrm{lfs}_{min} = \min \{ \mathrm{lfs}(x) \mid x \in \Omega \} .$$

**Theorem 7.1 (Radius Bounds).** *Suppose that the input to the Adaptive Delaunay Refinement Algorithm conforms to Assumption 4.2. Suppose that $\hat{\kappa} \leq \arcsin 2^{-7/6}$. Then there is a constant, $\mu$, depending on $\theta^*$ and $\hat{\kappa}$ such that if $p$ is the midpoint of a segment, $s$, of radius $r$ that is committed by the algorithm, then $\mathrm{lfs}_{min} \leq \mu r$.*

*Proof.* We consider why the segment was split. If there was an input point or a point on a disjoint input sequence that encroached $s$, then clearly $\mathrm{lfs}(p) \leq r$, so it suffices to take $\mu \geq 1$.

Suppose a midpoint $q$ on a nondisjoint input sequence encroached $s$. Using this result inductively we know that $\mathrm{lfs}_{min} \leq \mu r_q$, where $r_q$ is the radius associated with $q$. By Lemma 5.2, $r_q \leq r$, which suffices.

Suppose that $s$ was not encroached by an input point or midpoint, rather it was split when a circumcenter "yielded" to the segment split. Consider a maximal circumcenter sequence, $\{b_i\}_{i=0}^{l-1}$ ending in the circumcenter $b_{l-1}$ which yielded to the split of $s$. By maximality, $b_0$ is not a circumcenter. Consider its identity.

If $b_0$ is an input point or a midpoint on an input feature disjoint from the segment containing $s$, then $\mathrm{lfs}(p) \leq |p - b_0| \leq \eta r$, by Corollary 6.4. Thus it suffices to take $\eta \leq \mu$.

The only remaining possibility is that $b_0$ is a midpoint on an input feature nondisjoint from the one containing $s$. Let $r_b$ be the radius associated with $b_0$. This radius may be larger or smaller than $\tilde{r}_0 = |b_0 - a_0|$. We consider the possibilities:

- Suppose $r_b \leq \tilde{r}_0$. Using this result inductively we have $\mathrm{lfs}_{min} \leq \mu r_b$. By Corollary 6.4, $\tilde{r}_0 \leq (2 \sin \hat{\kappa})^{l-1} \sqrt{2} r$. If $b_0$ is a midpoint on the same input segment as $p$ or on a distinct input segment subtending an angle other than $\pi$, then by Assumption 4.2, $\log_2 \frac{r}{r_b}$ is an integer. But since $r_b \leq \sqrt{2} r$, it must be a nonnegative one, thus $r \geq r_b$, so $\mathrm{lfs}_{min} \leq \mu r$. The only alternative is $b_0$ is a midpoint on a distinct input segment subtending angle $\pi$ with the one containing $p$. Then either $r_b \leq r$, in which case immediately $\mathrm{lfs}_{min} \leq \mu r$, or $r < r_b$, in which case by Lemma 6.2, $l \geq 4$, so $r_b \leq (2 \sin \hat{\kappa})^3 \sqrt{2} r$. This yields a contradiction when $\hat{\kappa} \leq \arcsin 2^{-7/6}$, as assumed.
- Suppose $r_b > \tilde{r}_0$. This means that $a_0$ encroached the diametral circle of the subsegment associated with $b_0$, and thus, since $b_0$ was committed after $a_0$, $a_0$ is not a circumcenter.
  If $a_0$ is an input point or on an input segment disjoint from the one containing $b_0$, then $\mathrm{lfs}_{min} \leq |a_0 - b_0| = \tilde{r}_0$, so it suffices to take $\mu \geq 1$.
  The alternative is that $a_0$ is a midpoint on an input segment adjoining the one containing $b_0$. By the definition of the (QUALITY′) operation and circumcenter sequences, it must be the case that $\theta$, the angle between the two input segments is as least $\pi/3$. Using Lemma 5.2, we know that $\tilde{r}_0 = |a_0 - b_0| \geq r_a$, where $r_a$ is the radius associated with $a_0$.
  If the input segment containing $a_0$ is disjoint from the one containing $p$, then using Corollary 6.4 again it suffices to take $\eta \leq \mu$.
  Otherwise arguments as above show that $r \geq r_a$, and using this result inductively suffices.

In all it suffices to take $\mu = \eta = 1 + \frac{\sqrt{2}}{1 - 2 \sin \hat{\kappa}}$. □

The following corollary gives termination:

**Corollary 7.2.** *Suppose the Adaptive Delaunay Refinement Algorithm considers committing point $p$. Let $q$ be the closest point that has already been committed. Then $\mathrm{lfs}_{min} \leq \frac{\mu}{2 \sin \frac{\theta^*}{2}} |p - q|$.*

*Proof.* Consider the identity of $p$.

- Suppose $p$ is a midpoint, and let $r$ be the associated radius. If $r \leq |p - q|$, then the theorem gives $\mathrm{lfs}_{min} \leq \mu |p - q|$. If, however, $r > |p - q|$, then $q$ encroaches the subsegment of $p$, so it cannot be a circumcenter (which would have yielded). If $q$ is an input point or on a disjoint input feature then $\mathrm{lfs}_{min} \leq |p - q|$, which suffices. Otherwise $q$ is a midpoint on a nondisjoint input segment. Then, using, Lemma 5.2, $|p - q| \geq 2 r_q \sin \frac{\theta^*}{2}$, where $r_q$ is the radius associated with $q$. Using the theorem on $q$, we have $\mathrm{lfs}_{min} \leq \mu r_q$, which gives the desired result.
- Suppose $p$ is a circumcenter with associated radius $r$. Then $r = |p - q|$, since the triangle is Delaunay. Then $p$ can be considered the last circumcenter in a circumcenter sequence, and by Lemma 6.3 $r > \tilde{r}_0$. Then using this corollary inductively on the point $b_0$, the first point of the circumcenter sequence, gives the desired result.

□

Note that this proof entirely ignores the issue of grading. The skeptic might object that all the edges in the final mesh could have size $\Theta(\text{lfs}_{min})$. However, the algorithm actually does exhibit good grading; the proof is too involved for presentation in this forum [8].

The uniform grading constant does not diverge as $\hat{\kappa}$ reaches its limit value of $\arcsin 2^{-7/6}$, but does diverge as $\hat{\kappa}$ approaches $\pi/6$. Note that the limitation $\hat{\kappa} < \arcsin 2^{-7/6}$ comes from the case of collinear subsegments connected by a circumcenter sequence; in this situation Lemma 6.2 gives a lower bound on the length of the circumcenter sequence. A greater lower bound would relax the restriction on $\hat{\kappa}$, but this is not theoretically possible without changing the algorithm, as shown by the counterexample of Figure 4.

This does illustrate, however, why the Adaptive Delaunay Refinement Algorithm *might* work with $\hat{\kappa}$ as large as $30^\circ$ on a given input: constructing a counterexample such as Figure 4 where collinear subsegments are connected by a circumcenter sequence is difficult work. Moreover, such counterexamples require a few committed points noncollinear with the subsegments, points which have to be perfectly aligned to make the counterexample work. Thus it seems unlikely that one could construct a counterexample where setting $\hat{\kappa} = 30^\circ$ could cause the algorithm to fall into an infinite loop; such a counterexample would likely have to exhibit a structure which is scaled and repeated by repeated action of circumcenter sequences between collinear subsegments.

## 8. OUTPUT QUALITY

Recall that the Adaptive Delaunay Refinement Algorithm ignores angles smaller than the parameter $\hat{\kappa}$. We will show that small output angles are not too much smaller than a nearby small input angle. The following simple geometric claim gives the output quality guarantee; the idea is to use it with facts about midpoints, the definition of (QUALITY′), and the Delaunay property to get the bound on output angles. We omit the proofs due to space constraints.

**Lemma 8.1.** *Let $x, s, q$ be three distinct noncollinear points. Let $p$ be a point on the open line segment from $x$ to $s$. Suppose that $|p - s| \leq |x - p| \leq |x - q|$. Let $\theta = \angle pxq$, and $\phi = \angle psq$. Then*

$$\phi \geq \arctan\left(\frac{\sin\theta}{2 - \cos\theta}\right).$$

*Claim* 8.2 *(Edge-Apex Rule).* Given a triangle $\Delta pqr$ in the Delaunay Triangulation of a set of points, $\mathcal{P}$, with $L$ the line through $p, q$, then $\angle prq \geq \angle pr'q$ for every $r' \in \mathcal{P}$ that is on the same side of $L$ as $p$, with equality only holding in the case of degeneracy.

We can now state the output guarantee.

**Lemma 8.3.** *Suppose the Adaptive Delaunay Refinement Algorithm terminates for a given input. Let $\Delta pqr$ be a triangle in the output triangulation. Then either*

*(a) The angle $\angle prq > \hat{\kappa}$, or*
*(b) the points $p$ and $q$ are midpoints on adjoining input segments which meet at angle $\theta < \pi/3$ and*

$$\angle prq \geq \arctan\left(\frac{\sin\theta}{2 - \cos\theta}\right).$$

*Consequently no angle in the output mesh is smaller than* $\min\left\{\hat{\kappa}, \arctan\left(\frac{\sin\theta^*}{2-\cos\theta^*}\right)\right\}$.

*Proof.* Supposing that $\angle prq \leq \hat{\kappa}$, by the definition of the Adaptive Delaunay Refinement Algorithm, it must be that $p, q$ are midpoints on an adjoining input segment, meeting at an angle, $\theta$, less than $\pi/3$. Let $x$ be the input point common to these segments. Without loss of generality, assume that $|x - p| \leq |x - q|$. The midpoint $p$ is the endpoint of two subsegments of this input segment; let the one farther from $x$ be $(p, s)$. By Claim **??**, $|p - s| \leq |p - x|$. Then by Lemma 8.1, $\angle psq \geq \arctan\left(\frac{\sin\theta}{2-\cos\theta}\right)$. Letting $L$ be the line through $p, q$, consider the location of $r$:

- Suppose $r$ is on the same side of $L$ as $x$. By Claim 8.2, $\angle prq \geq \angle pxq = \theta > \arctan\left(\frac{\sin\theta}{2-\cos\theta}\right)$.
- If $r$ is on the same side of $L$ as $s$, by Claim 8.2, $\angle prq \geq \angle psq \geq \arctan\left(\frac{\sin\theta}{2-\cos\theta}\right)$.

□

We note briefly that $\arctan[(\sin\theta)/(2-\cos\theta)] = \theta + \mathcal{O}(\theta^2)$, which makes this lower bound much better than that of $\arcsin\left[\sin\left(\frac{\theta}{2}\right)/\sqrt{2}\right] = \frac{\theta}{2\sqrt{2}} + \mathcal{O}(\theta^2)$ achieved by Shewchuk's Terminator [3].

The following corollary gives an *upper* bound on output angles that depends on the output angle parameter, $\hat{\kappa}$, but *not* on the minimum output angle. Given $\hat{\kappa} = \arcsin 2^{-7/6} \approx 26.45^\circ$, it guarantees no output angle is bigger than about $\pi - 2\arcsin\frac{\sqrt{3}-1}{2} \approx 137.1^\circ$. The (omitted) proof relies on the location of small output angles and uses the fact that diametral circles of subsegments are not encroached in the final mesh.

**Corollary 8.4.** *If $\Delta pqr$ is a triangle in the output triangulation produced by the Adaptive Delaunay Refinement Algorithm, then*

$$\angle pqr \leq \max\left\{\pi - 2\hat{\kappa}, \pi - 2\arcsin\frac{\sqrt{3}-1}{2}\right\}.$$

## 9. ADAPTIVE MIDPOINT SPLITTING

Our analysis so far has required that input meet Assumption 4.2. This assumption can be satisfied by first adding no more than $2|\mathbb{S}|$ augmenting points, effectively redefining the input. While this can be done while only suffering a constant increase in the cardinality of the final point set, this increase may be unacceptably large [8]. Ruppert's original heuristic for dealing with midpoint-midpoint interactions can remove the additional restriction on input while still giving good point set sizes in practice.

Ruppert's strategy of splitting on concentric circular shells [1] proceeds as follows: The first time an input segment is split, it is split by a point at its midpoint, creating two subsegments each with one input point associated. When one of these subsegments is split, it is split by a point $p$ closest to the midpoint of the subsegment such that $|p - x|$ is a power of two (in some global unit), where $x$ is the input point associated with the subsegment. All further subsegment splits are committed at midpoints.

We will refer to these first three points on any segment as "off-center" points, even though they could be at the midpoint of the involved subsegment. It is simple to show that $\text{lfs}_{min}$ is no greater than three times the length of the shortest subsegment created by an off-center split under this strategy. This follows since $\text{lfs}_{min}$ is no greater than half the length of any input segment, and the fact that the off-center split must occur in the middle third of the subsegment.

Then Theorem 7.1 can be reproven for the Adaptive Delaunay Refinement Algorithm with concentric shell splitting for arbitrary input satisfying Assumption 2.1. The basic strategy is that if any of the midpoints involved in the proof are actually off-center points, they can be shown to be not far away by Corollary 6.4, and then the Lipschitz property of local feature size suffices; in the end game none of the involved midpoints are off-center, and the input locally conforms to Assumption 4.2, so the previous arguments may be used.

For the analysis to be valid, it is necessary that the algorithm treat off-center points as input points, not as midpoints. This makes a difference because the adaptive variant of the Delaunay Refinement Algorithm regards triangles differently if the shortest edge has midpoints as endpoints.

In light of the discussion in Subsection 3.1, we can make the following

*Claim* 9.1. Suppose an input conforming to Assumption 2.1 if given to Ruppert's Algorithm with concentric shell splitting. Then if $\kappa < 26.45^\circ \vee \arctan\left[(\sin\theta^*)/(2 - \cos\theta^*)\right]$, the algorithm will terminate with no output angle smaller than $\kappa$.

## 10. RESULTS

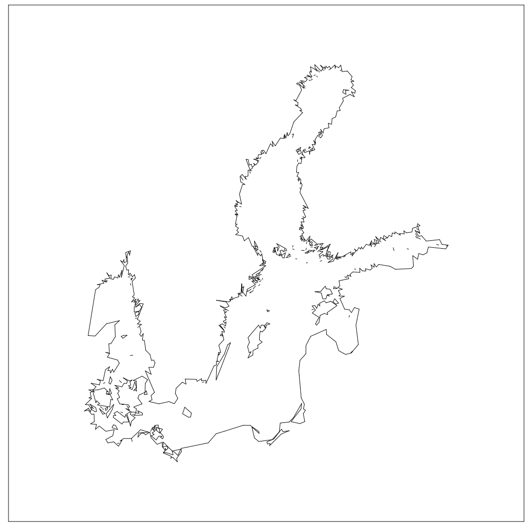


**Figure 5**: The Baltic Sea input data. The input consists of 1401 points and 1301 line segments. There are a number of small angles and small segments present. The minimum angle, $\theta^*$ is approximately $0.052^\circ$.

The Adaptive Delaunay Refinement Algorithm with splitting on concentric shells was implemented. The code was tested on the Baltic Sea, as shown in Figure 5, with $\hat{\kappa} \approx \arcsin 2^{-7/6}$. The input has a number of small angles, the smallest being around $0.052^\circ$.

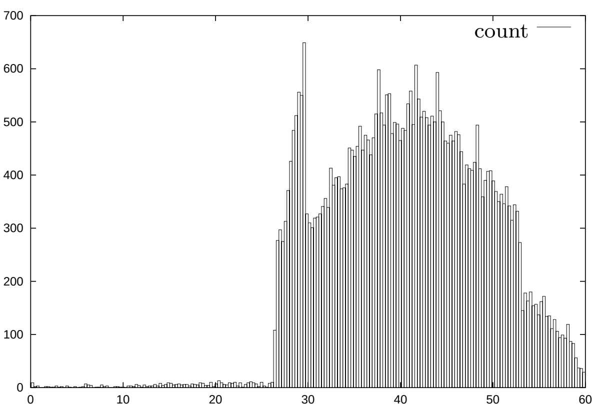
The output is shown in Figure 7, and is a mesh on 21704 vertices. The minimum and maximum angle histograms are shown in Figure 6. The minimum angle histogram shows that a small number of triangles have minimum angle less than $26.45^\circ$; these are all small input angles or "across" from small input angles, in accordance with Lemma 8.3.

### References

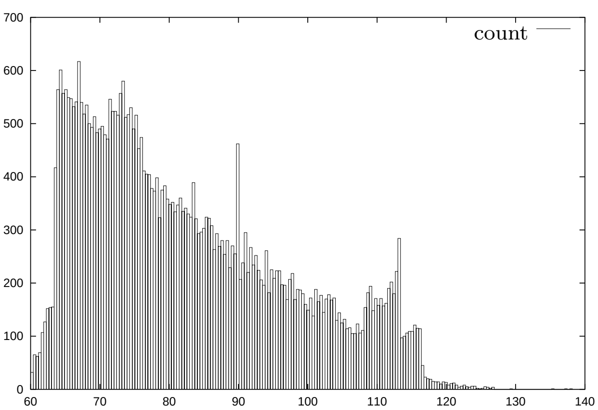

[1] Ruppert J. "A Delaunay refinement algorithm for quality 2-dimensional mesh generation." *J. Algorithms*, vol. 18, no. 3, 548–585, 1995. Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) (Austin, TX, 1993)

[2] Shewchuk J.R. *Delaunay Refinement Mesh Generation*. Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 1997. Available as Technical Report CMU-CS-97-137

[3] Shewchuk J.R. "Delaunay refinement algorithms for triangular mesh generation." *Comput. Geom.*, vol. 22, no. 1-3, 21–74, 2002. 16th ACM Symposium on Computational Geometry (Hong Kong, 2000)




(a) Minimum Angle




(b) Maximum Angle

**Figure 6**: The minimum- and maximum angle histograms are shown, respectively, in (a) and (b). In this figure triangles are counted, not angles, thus the total count is the number of triangles (in this case $43357$), and not three times that number. In (a), those triangles with minimum angle smaller than $\hat{\kappa} \approx 26.45^\circ$ are due to small input angles, in accordance with Lemma 8.3. The lack of large angles is guaranteed by Corollary 8.4.


[4] Boivin C., Ollivier-Gooch C.F. "Guaranteed-Quality Triangular Mesh Generation For Domains with Curved Boundaries." *International Journal for Numerical Methods in Engineering*, vol. 55, no. 10, 1185–1213, 2002

[5] Miller G.L., Pav S.E., Walkington N.J. "An Incremental Delaunay Meshing Algorithm." Tech. Rep. 02-CNA-011, Center for Nonlinear Analysis, Carnegie Mellon University, 2002. URL `http://www.math.cmu.edu/cna`

[6] Si H. "Tetgen. A 3D Delaunay Tetrahedral Mesh Generator. v.1.2 Users Manual.", Dec. 2002. URL `http://www.wias-berlin.de/publications/technicalreports/4`

[7] Cheng S.W., Poon S.H. "Graded conforming Delaunay tetrahedralization with bounded radius-edge ratio." *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (Baltimore, MD, 2003)*, pp. 295–304. ACM, New York, 2003

[8] Pav S.E. *Delaunay Refinement Algorithms.* Ph.D. thesis, Department of Mathematics, Carnegie Mellon University, Pittsburgh, Pennsylvania, May 2003. URL `http://www.andrew.cmu.edu/~spav/work`

[9] de Berg M., van Kreveld M., Overmars M., Schwarzkopf O. *Computational geometry.* Springer-Verlag, Berlin, revised edn., 2000. Algorithms and applications

[10] Teillaud M. *Towards dynamic randomized algorithms in computational geometry.* Springer-Verlag, Berlin, 1993

[11] Miller G.L. "A Timing Analysis of a Delaunay Refinement Mesh Generation Algorithm.", December 2002. In preparation

[12] Nanevski A., Blelloch G., Harper R. "Automatic Generation of Staged Geometric Predicates." *International Conference on Functional Programming*, pp. 217–228. Florence, Italy, September 2001

[13] Mücke E.P. "A robust implementation for three-dimensional Delaunay triangulations." *Internat. J. Comput. Geom. Appl.*, vol. 8, no. 2, 255–276, 1998

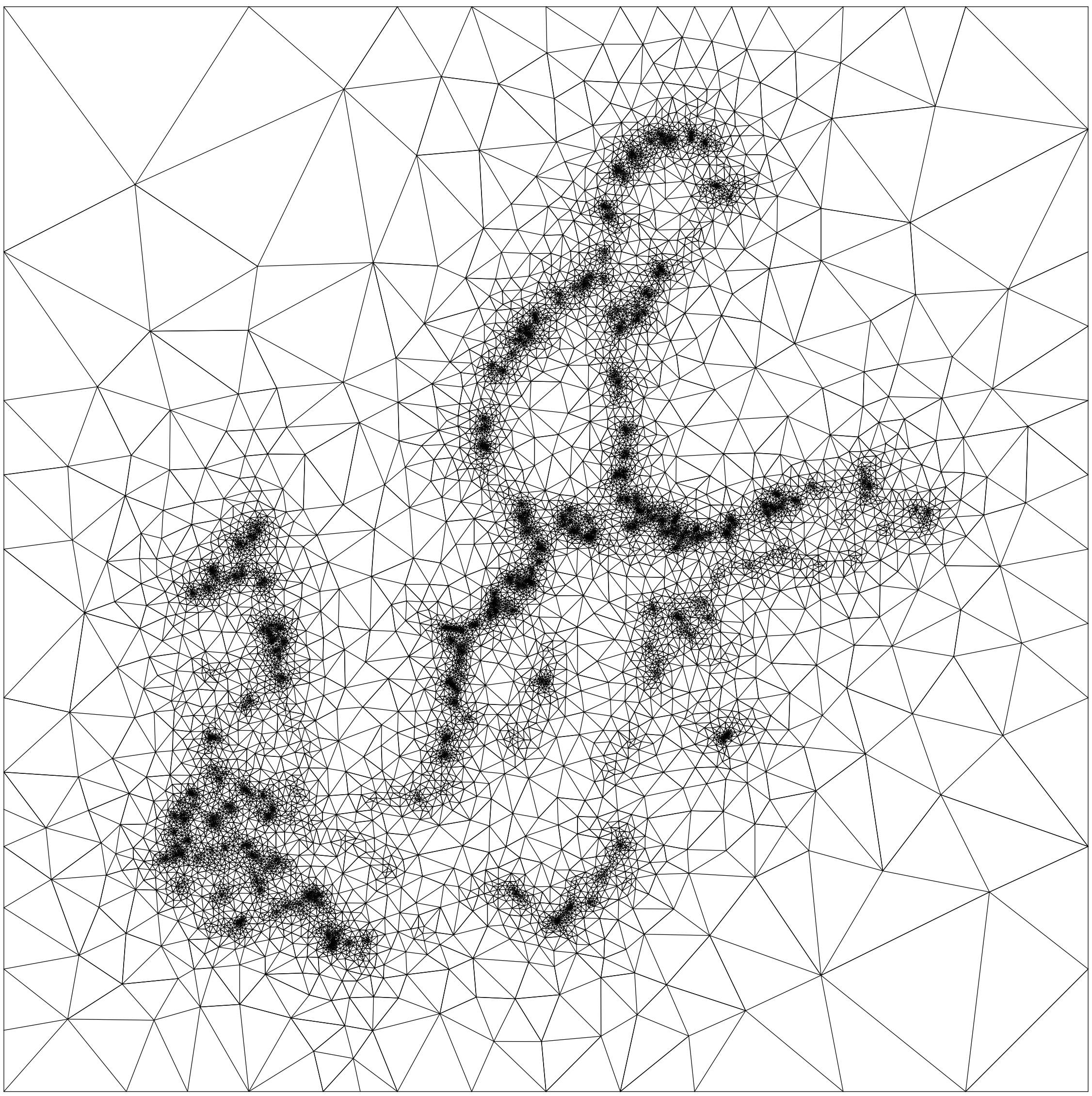


**Figure 7**: The output mesh of the Baltic Sea input (Figure 5) with $\hat{\kappa} \approx \arcsin 2^{-7/6}$ is shown.