Delaunay Refinement by Corner Lopping

Steven E. Pav[1] and Noel J. Walkington[2]

[1] University of California at San Diego, La Jolla, CA. spav@ucsd.edu
[2] Carnegie Mellon University, Pittsburgh, PA. noelw@andrew.cmu.edu

Summary. An algorithm for quality Delaunay meshing of 2D domains with curved boundaries is presented. The algorithm uses Ruppert's "corner lopping" heuristic [1]. In addition to admitting a simple termination proof, the algorithm can accept curved input without any bound on the tangent angle between adjoining curves. In the limit case, where all curves are straight line segments, the algorithm returns a mesh with a minimum angle of $\arcsin\left(1/2\sqrt{2}\right)$, except "near" input corners. Some loss of output quality is experienced with the use of curved input, but this loss is diminished for smaller input curvature.

Key words: unstructured, simplicial, planar, curved boundary, Delaunay, mesh.

1 Introduction

The Delaunay Refinement method is used for quality simplicial mesh generation in two and three dimensions. A Delaunay Refinement algorithm takes an input of points and segments (or curves) and adds Steiner Points to guarantee that the output Delaunay Triangulation conforms to the input and has high quality simplices, as measured by the circumradius to shortest edge length ratio. A Steiner Point is added to "split" an input segment into subsegments if a mesh vertex forms an obtuse angle with the segment. A Steiner Point is added at the circumcenter of a poor triangle in the mesh. Termination of the algorithm is had by proving a lower bound on the distance between Steiner Points, and applying compactness arguments [1, 2].

Ruppert was a pioneer of the Delaunay Refinement method. Ruppert's Algorithm accepts a planar straight line graph, and outputs a Delaunay mesh where no output angle is smaller than a user-chosen parameter, which can be as large as 20.7°. In Ruppert's analysis input segments have to meet at nonacute angles, otherwise his naïve algorithm might not terminate [1].

Ruppert offered two heuristic solutions to this problem. The first, "concentric shell splitting," has been adapted to a working algorithm, and allows better output quality guarantees [3, 2, 4]. In this solution, segments sharing a common endpoint are split at the same distance from the endpoint, *i.e.*, on the same "shell." This simple fix gives a good lower bound, and an input-independent upper bound, on output angles. However, its analysis is involved, and does not generalize naturally to higher dimensions or curved input (due to its reliance on "power of two" arguments). Ruppert's second solution, "corner lopping," is analyzed herein, and admits a simple proof.

Delaunay Refinement for curved input was considered by Boivin and Ollivier-Gooch [5]. Their analysis requires that input segments meet at an angle of at least $\pi/3$. Concentric shell splitting to deal with smaller input angles was mentioned in this context, but not shown to give a working algorithm; this fix clearly would require further modification to the output quality guarantee.

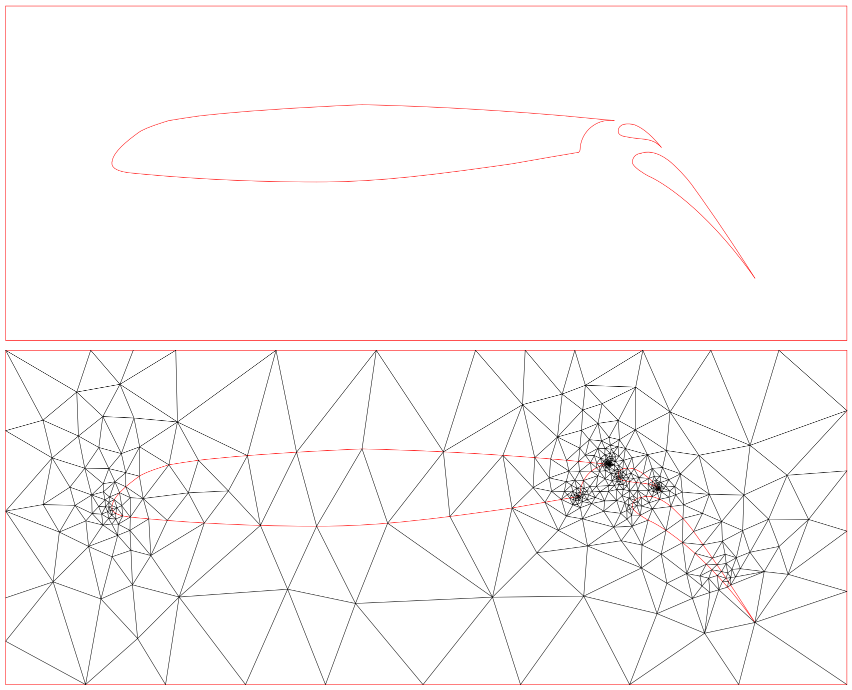


Fig. 1. The outline of a mock air foil and output from the meshing algorithm. Solving a fluid dynamics PDE would probably require further mesh refinement.

The Delaunay Refinement Algorithm has also been generalized to three dimensions. Early analysis required input segments and faces to meet at nonacute angles [6]. As a fix, later work used protective regions around input points and segments [7, 8, 9, 10]. In that way these algorithms resemble corner lopping, which places a protective ball around acute corners in the input. Reverse from what is usually seen, these three-dimensional algorithms do not appear to be the natural generalization of any known two-dimensional algorithm.

The motivations for the present work, then, are:

1. To present an algorithm which accepts straight or curved input without a lower bound on input angle, yet admits a simple termination proof.
2. To find an algorithm which is the "projection" into the plane of recently discovered three-dimensional algorithms, and thereby to gain a better understanding of those algorithms.

2 Preliminaries

The input to the algorithm is assumed to be a PRCC.

Definition 1 (PRCC). *A set of points and a set of non-closed regular curves embedded in $\mathbb{R}^2$, $(\mathcal{P}, \mathcal{C})$, form a* Piecewise Regular Curve Complex *(PRCC) if*

(i) for any curve, $c \in \mathcal{C}$, the endpoints of c are elements of $\mathcal{P}$.

(ii) given two curves, $c_1, c_2 \in \mathcal{C}$, their intersection is either empty or an endpoint (or endpoints) common to both curves.

(iii) given $p \in \mathcal{P}$, $c \in \mathcal{C}$, either p is an endpoint of c, or p is not on c.

The goal of meshing is to produce, from input $(\mathcal{P}, \mathcal{C})$, the Delaunay Triangulation of a set of points, $\mathcal{P}'$, hereafter denoted as $\mathcal{D}(\mathcal{P}')$, such that (i) $\mathcal{P} \subseteq \mathcal{P}'$, (ii) each input curve of $\mathcal{C}$ is approximated by a piecewise linear curve which is the union of edges in $\mathcal{D}(\mathcal{P}')$, (iii) all or most of the triangles of $\mathcal{D}(\mathcal{P}')$ are "high quality." Absent of a specific interpolation problem, triangle "quality" is taken to be inversely proportional to its minimum angle. When guaranteeing a large minimum angle is not possible due to input constraints, an upper bound on the maximum angle of the mesh is often desired [11, 12, 4].

Given two points, p, q let $\overline{pq}$ be the line segment with these points as endpoints, and let $\widetilde{pq}$ denote a curve with p and q as endpoints. Let $|p - q|$ denote the distance between p and q. For a curve c, and a point p, let

$$|p - c| = \min_{x \in c} |p - x|$$

be the distance from p to c. We use $a \vee b$ to denote the maximum of quantities a and b, and $a \wedge b$ to represent the minimum.

The local feature size, first defined by Ruppert [1], is used to prove termination and quality of the output mesh. We take the classical definition:

Definition 2 (Local Feature Size). *Given a PRCC, $(\mathcal{P}, \mathcal{C})$, define $\operatorname{lfs}_i(x)$, for $i = 0, 1$, as the distance from x to two mutually disjoint features of the PRCC of dimension no greater than i. By "feature" we mean a point in $\mathcal{P}$ or a curve in $\mathcal{C}$. Thus, for example, $\operatorname{lfs}_0(x)$ is the distance from x to the second nearest point of $\mathcal{P}$. Let $\operatorname{lfs}(x) = \operatorname{lfs}_1(x)$ be the* local feature size.

Note 1. The following facts about local feature size are immediate: (i) For any x, $\text{lfs}_1(x) \leq \text{lfs}_0(x)$, (ii) $\text{lfs}_i(x)$ is a Lipschitz function with constant 1, *i.e.*, $\text{lfs}_i(p) \leq \text{lfs}_i(q) + |p - q|$, (iii) $\text{lfs}_i(x)$ has a positive minimum value on $\mathbb{R}^2$.

Some authors use "local feature size" to describe a different function, the distance of a point on the input to the medial axis of the input [13]. While both definitions give Lipschitz functions, our definition yields a function defined in all of $\mathbb{R}^2$ with a strictly positive lower bound. The latter fact is important because our local feature size will describe, roughly, the size of triangles we expect to see nearby in an output mesh of the given input.

In the case of straight line input, a lower bound on the angle subtended by input segments is used to show that Steiner Points on input segments are not placed too close together. For PRCC input, we instead use the following.

Definition 3 (Curve Separation). *For the sake of this definition, given curve $c = \widetilde{xy}$, we say that a point z on c is* sufficiently far *from x if $|x - z| \geq \text{lfs}(x)/2C_0$, where $C_0 = 1 + \sqrt{2}$, is a "grading constant" (see Lemma 4).*

Given two curves c_1, c_2 sharing a single endpoint x, then the separation *between them is*

$$\inf_{z_1, z_2} \frac{|z_1 - z_2|}{|z_1 - x| \vee |z_2 - x|},$$

where z_i is a point on c_i that is sufficiently far from x.

If curves c_1, c_2 share both their endpoints, say x and y, then the separation between them is

$$\inf_{z_1, z_2} \frac{|z_1 - z_2|}{|z_1 - x| \vee |z_2 - x|},$$

where z_i is a point on c_i that is sufficiently far from both x and y.

For a given PRCC, let σ be a lower bound on the separation between any pair of curves with at least one common endpoint. We say σ is the "minimum curve separation" of the PRCC.

Given that input curves are continuous and may meet at most at their endpoints, the separation between two curves is strictly positive, and thus σ is positive as well. In the case where all input curves are straight line segments, we have $\sigma = 2\sin(\theta^*/2)$, where $\theta^* \leq \pi/3$ is a lower bound on the angle subtended by the segments.

Following Boivin and Ollivier-Gooch [5], we make the following

Definition 4 (Total Variation of a Curve). *The* total variation *of curve C is $\int |\,\mathrm{d}\theta|$, where θ is the angle subtended by the tangent of the curve. Let τ be an upper bound on the total variation of every curve in an input PRCC.*

For the remainder of this paper we assume $\tau \leq \pi/3$. Such a bound can be achieved by splitting curves in a preprocessing step.

Algorithm 1 Algorithm for meshing via corner lopping.

```
Input: A PRCC, an angle bound, the splitting fraction
Output: A mesh
      CORNERLOP$((\mathcal{P}, \mathcal{C}), \kappa, \beta)$
(1)  Let $\mathcal{P}' \leftarrow \mathcal{P}, \mathcal{C}' \leftarrow \mathcal{C}$.
(2)  Construct $\mathcal{D}(\mathcal{P}')$.
(3)  Use $\mathcal{D}(\mathcal{P}')$ to find $\mathrm{lfs}_0(\cdot)$ for points in $\mathcal{P}$'.
(4)  foreach $p \in \mathcal{P}$
(5)      if $p$ is the corner of an acute angle in $\mathcal{C}'$.
(6)          $d(p) \leftarrow (\sqrt{2} - 1)\, \mathrm{lfs}_0(p)$
(7)          SPLITBALL$(p, 1)$
(8)      else
(9)          Let $d(p) \leftarrow 0$.
(10) while any of these rules can be executed, execute one of them, with the
     rules listed in descending priority:
(11)     if there are $p \in \mathcal{P}, q \in \mathcal{P}'$, with $|p - q| < d(p)$ then SPLITBALL$(p, \beta)$
(12)     if there is $z \in \mathcal{P}', c \in \mathcal{C}'$ such that $z$ encroaches $c$ then SPLITCURVE$(c, z)$
(13)     if there is $\Delta xyz \in \mathcal{D}(\mathcal{P}')$ such that $\angle xyz \leq \kappa$ and the circumcenter of
         $\Delta xyz$ is not closer to some $q \in \mathcal{P}$ than $d(q)$ then SPLITTRI$(\Delta xyz)$
(14) return $\mathcal{D}(\mathcal{P}')$

     SPLITBALL$(p, C)$
(15) Let $d(p) \leftarrow C d(p)$.
(16) foreach $\widetilde{pt} \in \mathcal{C}'$
(17)     Let $x$ be the point on $\widetilde{pt}$ such that $|p - x| = d(p)$
(18)     Add $x$ to $\mathcal{P}'$. Remove $\widetilde{pt}$ from $\mathcal{C}'$, replacing it with $\widetilde{px}, \widetilde{xt}$.
     SPLITCURVE$(\widetilde{xy}, [z])$
(19) if a $z$ is given
(20)     Let $p$ be some point on $\widetilde{xy}$, perhaps based on $z$.
(21) else
(22)     Let $p$ be some point on $\widetilde{xy}$.
(23) Add $p$ to $\mathcal{P}'$, replace $\widetilde{xy}$ in $\mathcal{C}'$ by subcurves $\widetilde{xp}$ and $\widetilde{py}$.
     SPLITTRI$(\Delta xyz)$
(24) Let $p$ be the circumcenter of $\Delta xyz$.
(25) if $p$ encroaches curve $c \in \mathcal{C}'$
(26)     SPLITCURVE$(c)$
(27) else
(28)     Add $p$ to $\mathcal{P}'$.
```

3 The Algorithm

The Delaunay Refinement algorithm we consider maintains sets, $\mathcal{P}'$, and $\mathcal{C}'$, which are initialized, respectively, as $\mathcal{P}$ and $\mathcal{C}$. The algorithm adds *Steiner Points* to $\mathcal{P}'$, then returns $\mathcal{D}(\mathcal{P}')$ on termination. Throughout this work, "curve" or "subcurve" means a member of $\mathcal{C}'$, while "input curve" refers to a member of $\mathcal{C}$. "Input point" refers to a point of $\mathcal{P}$. We will take $\mathrm{lfs}_i(\cdot)$ to be with respect to the PRCC input to the algorithm, $(\mathcal{P}, \mathcal{C})$.

Our algorithm protects regions around the input points and (sub)curves. Following the notation of Ruppert [1], a (sub)curve is said to be "encroached" if there exists a vertex of the mesh inside its diametral circle, where the diametral circle of curve $\widetilde{pq}$ is the circle with $\overline{pq}$ as diameter. We denote the diametral circle of curve c by $\mathcal{C}_d(c)$.

The algorithm assigns to each input point, p, a radius $d(p) \geq 0$, and, by analogy with curves, p is said to be encroached if a vertex of the mesh is in $\mathcal{B}(p)$, which is defined to be the open ball of radius $d(p)$ centered at p.

The algorithm, given as Algorithm 1 on the preceding page, takes the input PRCC and two parameters: κ, the desired angle bound of the output mesh, and $\beta \in \left[\frac{1}{2}, 1\right)$, which is the factor by which a radius $d(p)$ is reduced when an input point p is encroached.

To split the curve $c = \widetilde{xy}$, an intermediate point $p \in c$ is selected and c is replaced by two subcurves $\widetilde{xp}$ and $\widetilde{py}$. In the case c is a segment, traditionally p is chosen as the midpoint of the segment. Without a clear definition of the midpoint of a curve, we assume only that the algorithm satisfies the following assumption regarding the selection of p:

Assumption 1. Assume there are constants $\eta \geq 1$ and $\mu \geq 1$ such that the algorithm is implemented so that
(1) if p is selected to split curve c by a call to SPLITCURVE(c, z), then $|p - \mathcal{C}_d(c)| \geq |p - z|/\eta$.
(2) if p is selected to split curve $c = \widetilde{xy}$ by a call to SPLITCURVE(s), then $|p - \mathcal{C}_d(s)| \geq r/\mu$, where r is the radius of the circumcircle, *i.e.*, $|x - y|/2$.

The need for this assumption is illustrated in Figure 2.

4 Proof of Termination

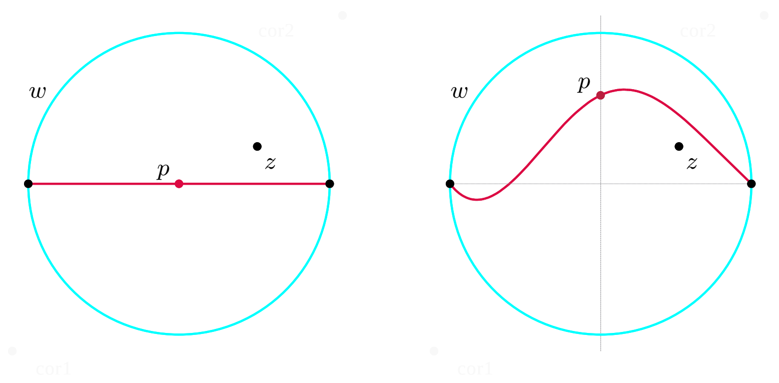
We first consider some facts regarding curved input. The following lemma is a consequence of the Mean Value Theorem:

Lemma 1 (Lens Containment). *Let C be a curve with endpoints x, y, and with total variation less than τ. Suppose z is a point on C distinct from the endpoints. Then*

$$\angle xzy \geq \pi - \tau.$$

This lemma claims that the worst case of a curve of total variation no more than τ is a circular arc. That is, a curve $\widetilde{xy}$ with bounded total variation is contained in a diametral lens of segment $\overline{xy}$. The corollaries claim that by careful choice of p in step 20 and step 22, the algorithm will conform to Assumption 1, with

$$\eta = \frac{1 + \tan(\tau/2)}{1 - \tan(\tau/2)}, \qquad \mu = \frac{1}{1 - \tan(\tau/2)}.$$



(a) Case 1 (b) Case 2

Fig. 2. For the case of segment input, as shown in (a), Assumption 1 can be satisfied with $\eta = \mu = 1$. For general curves, as in (b), a point p selected to split a curve may be near the diametral circle of the curve, *i.e.*, near other Steiner Points which may lie outside the diametral circle.

Corollary 1. *Let $c = \widetilde{xy}$ be a curve with total variation less than τ. Let $\mathcal{C}_d(c)$ be the diametral circumcircle of c, and let z be a point in this circle. Then there is a point p on c such that*

$$|p - \mathcal{C}_d(c)| \geq |p - z| \frac{1 - \tan(\tau/2)}{1 + \tan(\tau/2)}.$$

Corollary 2. *Let $c = \widetilde{xy}$ be a curve with total variation less than τ. Let $\mathcal{C}_d(c)$ be the diametral circumcircle of c. Then there is a point p on c such that*

$$|p - \mathcal{C}_d(c)| \geq r\,(1 - \tan(\tau/2)),$$

where r is the radius of the diametral circle.

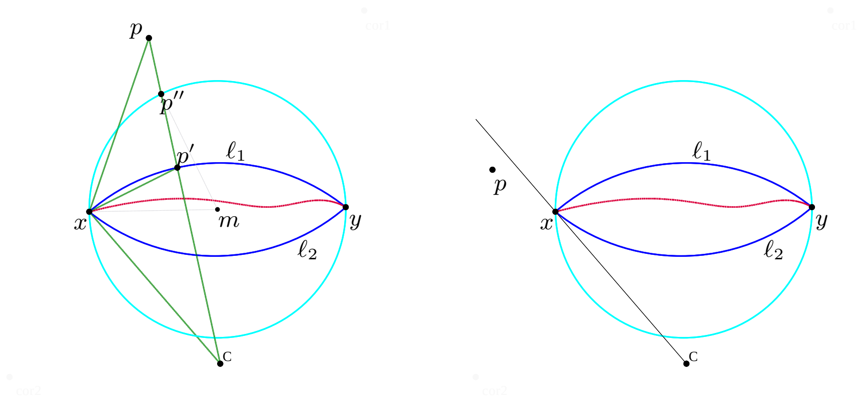
Both the corollaries are proven by taking p to be the intersection of the curve with the perpendicular bisector of segment $\overline{xy}$, as shown in Figure 2(b).

The following lemma is needed since we protect curves with the diametral circle of their secant segments, but add "midpoints" on the curve. Thus, in general, the diametral circle of a curve may not wholly contain the diametral circles of its subcurves.

Lemma 2 (Diametral Circle Protection). *Let $c = \widetilde{xy}$ be a curve with total variation less than τ. Let p be a point which does not encroach the diametral circle of c. Letting $|p - c|$ be the distance from p to the curve c, then*

$$\frac{|p - c|}{|p - x| \wedge |p - y|} \geq \frac{1}{\zeta},$$

where $\zeta = \sqrt{2}\sin(\tau/2)/\left(\sqrt{1+\sin\tau}-1\right)$.



(a) Case 1 (b) Case 2

Fig. 3. Two cases in the proof of Lemma 2. In (a), $\overline{wp}$ intersects the arc ℓ_1, while there is no such intersection in (b).

Proof. We assume that $|p-x| \le |p-y|$. Let ℓ_1, ℓ_2 be the two arcs of points subtending angle $\pi - \tau$ with x, y. By Lemma 1, c is between these two arcs. Without loss of generality, assume p is "above" ℓ_1. Let w be the center of arc ℓ_1.

We consider two cases:

1. The first case is if $\overline{pw}$ and ℓ_1 intersect. Let p' be their point of intersection. Let p'' be the intersection of $\overline{pw}$ with the diametral circle of segment $\overline{xy}$, as in Figure 3(a).
   Then, using the sine rule,

$$\frac{|p-c|}{|p-x| \wedge |p-y|} = \frac{|p-c|}{|p-x|} \ge \frac{|p-p'|}{|p-x|} = \frac{\sin \angle p'xp}{\sin \angle pp'x} \ge \frac{\sin \angle p'xp''}{\sin \angle p''p'x} = \frac{|p''-p'|}{|p''-x|}.$$

   Note that since $\angle pp'x$ is obtuse, then $\angle p'xp$ is acute and we can indeed conclude that $\sin \angle p'xp \ge \sin \angle p'xp''$. Let $R = |w-x|$, and let $r = |m-x|$, where m is the midpoint of x and y. Because $\angle xwy = \tau$, then $r = R\sin(\tau/2)$. Let $\phi = \angle p''mx$. Then

$$\begin{aligned} |p''-p'| = |w-p''| - R &= \sqrt{\left(R\cos(\tau/2) + r\sin\phi\right)^2 + \left(r\cos\phi\right)^2} - R \\ &= R\left(\sqrt{1+\sin\tau\sin\phi} - 1\right). \end{aligned}$$

   Then

$$\frac{|p'' - p'|}{|p'' - x|} = \frac{R\sqrt{1 + \sin\tau \sin\phi} - R}{2r \sin(\phi/2)} = \frac{\sqrt{1 + \sin\tau\sin\phi} - 1}{2\sin(\tau/2)\sin(\phi/2)}.$$

This quantity decreases as ϕ increases, thus it takes minimum value when $\phi = \pi/2$. Thus

$$\frac{|p'' - p'|}{|p'' - x|} \geq \frac{\sqrt{1 + \sin\tau} - 1}{\sqrt{2}\sin(\tau/2)}.$$

2. The other case is that $\overline{pw}$ and ℓ_1 do *not* intersect, as shown in Figure 3(b). That is, $\angle mxp > \pi/2 + \tau/2$. Looking at the circle centered at w through ℓ_1, then $|p - c| \geq |p - \ell_1| = |p - x|$. Thus

$$\frac{|p - c|}{|p - x| \wedge |p - y|} \geq 1 > \frac{1}{\sqrt{2}} \geq \frac{\sqrt{1 + \sin\tau} - 1}{\sqrt{2}\sin(\tau/2)}.$$

As expected, this lower bound is "worse," *i.e.*, smaller, for larger τ. However, we can make the rough uniform bound, $\zeta < 2$ for $0 \leq \tau \leq \pi/3$.

For $p \in \mathcal{P}$, let $\mathcal{B}(p)$ be the open ball of radius $d(p)$ centered at p during any step of the execution of Algorithm 1. Let $\partial\mathcal{B}(p)$ be the boundary of $\mathcal{B}(p)$, and let $\overline{\mathcal{B}}(p)$ be the closed ball of radius $d(p)$ centered at p. If $d(p) = 0$, let $\mathcal{B}(p)$ be the empty set and let $\overline{\mathcal{B}}(p)$ be the point p.

After $d(p)$ is set for all $p \in \mathcal{P}$, then for $p, q \in \mathcal{P}$,

$$d(p) + d(q) \leq \left(\sqrt{2} - 1\right)(\mathrm{lfs}_0(p) + \mathrm{lfs}_0(q)) < \frac{\mathrm{lfs}_0(p) + \mathrm{lfs}_0(q)}{2} \leq |p - q|$$

Since $d(p)$ can only shrink for a given $p \in \mathcal{P}$ during the lifetime of the algorithm, we make the following claim.

Claim. For distinct $p, q \in \mathcal{P}$, at all times $\overline{\mathcal{B}}(p) \cap \overline{\mathcal{B}}(q) = \emptyset$.

In the following analysis, we will refer to points of $\mathcal{P}$ as "input;" points which are added to split curves, that is at step 7 or step 23, as "midpoints;" triangle circumcenters added at step 28 are called "circumcenters."

The following two lemmata assert that a ball $\mathcal{B}(p)$ is only split by a midpoint on an input edge disjoint from p, and thus $d(p)$ is always bounded below by a constant times $\mathrm{lfs}(p)$.

Lemma 3. *If a point t is added to $\mathcal{P}'$ such that $t \in \mathcal{B}(p)$ for some $p \in \mathcal{P}$ then t must be a midpoint on a curve of $\mathcal{C}$ disjoint from p.*

Proof. Consider the possible identity of such a t:
- It cannot be that $t \in \mathcal{P}$, as $\overline{\mathcal{B}}(p) \cap \overline{\mathcal{B}}(t) = \emptyset$.
- It cannot be that t was added to $\mathcal{P}'$ in step 7, as if t is on $\partial\mathcal{B}(p)$ then it is not in $\mathcal{B}(p)$, and if $t \in \partial\mathcal{B}(q)$ for some $q \in \mathcal{P}$, then since $\overline{\mathcal{B}}(p)$ does not meet $\overline{\mathcal{B}}(q)$, q is not in $\mathcal{B}(p)$.

- Suppose t is the midpoint of some curve, c, which was added in step 23 or step 26. Then c cannot have been encroached by some $q \in \mathcal{B}(p)$ with $q \neq p$ as then the rule SPLITBALL would have been preferred to SPLITCURVE or SPLITTRI. Thus c cannot have endpoint p, as otherwise its diametral circle would be contained in $\overline{\mathcal{B}}(p)$, and thus it could not be encroached by such a q. That is, c must be disjoint from p.
- It cannot be that t was the circumcenter of a triangle, as this kind of circumcenter is explicitly prohibited from being added to $\mathcal{P}'$.

Lemma 4. *During execution of Algorithm 1, if $p \in \mathcal{P}$, and $d(p) > 0$ then*

$$\operatorname{lfs}(p) \leq C_0 d(p),$$

where $C_0 = \left(1 + \sqrt{2}\right) \approx 2.41$.

Proof. A nonzero $d(p)$ is set at two places in the algorithm, step 6 and step 15:

(step 6) In this case, because $\operatorname{lfs}(p) \leq \operatorname{lfs}_0(p)$,

$$\operatorname{lfs}(p) \leq \operatorname{lfs}_0(p) = \frac{1}{\sqrt{2}-1} d(p) = C_0 d(p).$$

(step 15) In this case, by Lemma 3, the input point is encroached by a point t on a curve disjoint from p. Then before $d(p)$ is reduced to $\beta d(p)$ it is the case that $\operatorname{lfs}(p) \leq |p - t| \leq d(p)$. Considering the new value of $d(p)$, this is $\operatorname{lfs}(p) \leq d(p)/\beta \leq 2d(p) < C_0 d(p)$.

Thus the radius $d(p)$ associated with an input point is never too much smaller than suggested by the local feature size, which is determined by the input. This implies that for a given input, SPLITBALL(p, C) is called a finite number of times. Next we show that this lower bound on $d(p)$ gives a bound on the distance between midpoints on nondisjoint input curves.

Lemma 5. *Suppose p, q are midpoints in $\mathcal{P}'$. Furthermore assume the midpoints are on distinct nondisjoint curves of $\mathcal{C}$. Then*

$$\operatorname{lfs}(p) \leq \frac{1 + C_0}{\sigma} |p - q|,$$

where $C_0 = \left(1 + \sqrt{2}\right)$ is the constant from Lemma 4.

Proof. Let the two curves share input point x. By the Lipschitz property, then using Lemma 4

$$\operatorname{lfs}(p) \leq |p - x| + \operatorname{lfs}(x) \leq |p - x| + C_0 d(x).$$

Since p is a point on an input curve with endpoint x, it is the case that $d(x) \leq |p - x|$. Thus $\operatorname{lfs}(p) \leq (1 + C_0)|p - x|$. Since q is also on a curve with endpoint x, then $d(x) \leq |q - x|$. Moreover, using Lemma 4, we can claim that

$$|p-x| \wedge |q-x| \geq d(x) \geq \frac{\operatorname{lfs}(x)}{C_0} > \frac{\operatorname{lfs}(x)}{2C_0}$$

The, by the definition of minimum curve separation (Definition 3), $|p-q| \geq |p-x|\,\sigma$, and thus

$$\operatorname{lfs}(p) \leq (1+C_0)\,|p-x| \leq \frac{1+C_0}{\sigma}\,|p-q|\,.$$

The following theorem asserts that the spacing between points in $\mathcal{P}'$ is bounded by the local feature size of the input, *i.e.*, the output of the algorithm is "well graded." Moreover, this theorem proves termination of the algorithm, by a ball-packing argument [1, 2].

Theorem 1. *Suppose p is to be added to $\mathcal{P}'$ during the execution of Algorithm 1. Suppose the algorithm is implemented such that Assumption 1 is satisfied. Let q be a point in $\mathcal{P}'$ at the time p is to be added. Then there are constants C_1, C_2 such that*

$$\operatorname{lfs}(p) \leq \begin{cases} C_1\,|p-q| & \textit{if } p \textit{ is a curve midpoint,} \\ C_2\,|p-q| & \textit{if } p \textit{ is a triangle circumcenter.} \end{cases}$$

Proof. We consider the cases:

1. Suppose p is a midpoint on input curve c. If q is an input point disjoint from c, or a midpoint on a curve disjoint from c, then $\operatorname{lfs}(p) \leq |p-q|$, and it suffices to take $1 \leq C_1$. If q is an input endpoint of c, then since $d(q) \leq |p-q|$, by Lemma 4, $\operatorname{lfs}(p) \leq |p-q| + \operatorname{lfs}(q) \leq |p-q| + C_0 d(q) = (1+C_0)\,|p-q|$. Thus $1+C_0 \leq C_1$ suffices.
   If q is a midpoint on a curve nondisjoint from c, then by Lemma 5, it suffices to take

   $$\boxed{\frac{1+C_0}{\sigma} \leq C_1}$$

   Thus we can assume q is a circumcenter, or another midpoint on c. Now consider the subcases for the identity of p:
   a) Suppose p is a point on input curve $c = \widetilde{xy}$, added to $\mathcal{P}'$ in step 7. Since no circumcenters have been added to $\mathcal{P}'$ when step 7 is executed, q must be a midpoint on c. Since at most two midpoints are added to c during the grooming phase, p, q are associated with the endpoints, x, y respectively. That is $|p-x| = d(x)$, and $|q-y| = d(y)$. Then

   $$\begin{aligned} |p-q| = |x-y| - d(x) - d(y) &\geq \frac{1}{\sqrt{2}-1}\left(d(x) \vee d(y)\right) - 2\left(d(x) \vee d(y)\right) \\ &= \frac{3-2\sqrt{2}}{\sqrt{2}-1}\left(d(x) \vee d(y)\right). \end{aligned}$$

   Because x, y are input points

$$\begin{aligned}\operatorname{lfs}(p) &\le |p-x| \vee |p-y| = d(x) \vee (|p-q| + d(y)) \\ &\le \left(1 + \frac{\sqrt{2}-1}{3-2\sqrt{2}}\right) |p-q| = (1+C_0)\,|p-q|\end{aligned}$$

Thus $1 + C_0 \le C_1$ suffices.

b) Suppose p is added in step 18 when the ball around input point x is reduced. Let $\mathcal{B}_-(x)$ be ball around x before $d(x)$ is reduced, *i.e.*, the ball of radius $d_-(x) = d(x)/\beta$, where $d(x)$ reflects the value after step 15 has been executed. Then we have $|p-x| = d(x)$, and the distance from x to the boundary of $\mathcal{B}_-(x)$ is $(1-\beta)d(x)/\beta$.
By Lemma 3, $d(x)$ is only reduced if there is some t on a curve disjoint from x with $t \in \mathcal{B}_-(x)$. In this case $\operatorname{lfs}(x) \le |x-t| \le d_-(x) = d(x)/\beta$.
We entertain the two possible locations of q. The first is that q is outside $\overline{\mathcal{B}}_-(x)$, and thus $|p-q| \ge (1-\beta)d(x)/\beta$. Then

$$\begin{aligned}\operatorname{lfs}(p) \le |p-x| + \operatorname{lfs}(x) \le (1 + 1/\beta)\, d(x) &= \left(\frac{1+\beta}{1-\beta}\right)\left(\frac{1-\beta}{\beta}\right) d(x) \\ &\le \frac{1+\beta}{1-\beta}\,|p-q|.\end{aligned}$$

Thus we must take

$$\boxed{\frac{1+\beta}{1-\beta} \le C_1}$$

The second possibility is that q is inside $\overline{\mathcal{B}}_-(x)$. We assumed q is a circumcenter or on the input curve c also containing p. However, by design of the algorithm, q cannot be a circumcenter in $\mathcal{P}'$ *and* be inside $\overline{\mathcal{B}}_-(x)$. The remaining possibility is that q be x itself. In this case $\operatorname{lfs}(p) \le |p-x| + \operatorname{lfs}(x) \le |p-x| + C_0 d(x) = (1+C_0)\,|p-x|$, and so the requirement $1 + C_0 \le C_1$ suffices.

c) Suppose p is added to $\mathcal{P}'$ by a call to SPLITCURVE(c', z). If z is an input point or midpoint on an input curve distinct from c, then by arguments above

$$\operatorname{lfs}(p) \le \frac{1+C_0}{\sigma}\,|p-z|.$$

If z is a circumcenter, then at the time it was added it did not encroach a supercurve of c', as otherwise it would have been rejected. By Lemma 2, since p is a point on that supercurve, $|z-p| \ge r_z/\zeta$, where r_z is the radius of the triangle that z killed. Using this result inductively when z was added, $\operatorname{lfs}(p) \le |p-z| + \operatorname{lfs}(z) \le |p-z| + C_2 r_z \le (1+\zeta C_2)\,|p-z|$.
Similarly, if q is any other point of $\mathcal{P}'$ which encroached c', then by the arguments of the previous paragraph,

$$\operatorname{lfs}(p) \le \max\left\{\frac{1+C_0}{\sigma}, 1+\zeta C_2\right\}|p-q|.$$

Now, since Assumption 1 is satisfied, $|p-q| \le \eta\,|p-\mathcal{C}_d(s')|$, and so it suffices to take

$$\boxed{\eta\frac{1+C_0}{\sigma} \le C_1} \quad \text{and} \quad \boxed{\eta\,(1+\zeta C_2) \le C_1}$$

If q is a point of $\mathcal{P}'$ which does not encroach c' then $|p-\mathcal{C}_d(c')| \le |p-q|$, and the above bounds on C_1 suffice.

d) Suppose p is added to $\mathcal{P}'$ by a call to SPLITCURVE(s') at step 26. Then there was some circumcenter x which encroached s', but was rejected. Inductively $\operatorname{lfs}(x) \le C_2\,|x-t|$, where t is either endpoint of s'. Letting r be the radius of s', *i.e.*, half its length, this gives $\operatorname{lfs}(x) \le \sqrt{2}C_2 r$, by the classical argument [1]. Then $\operatorname{lfs}(p) \le |p-x| + \operatorname{lfs}(x) \le \left(1+\sqrt{2}C_2\right) r$. Since SPLITTRI was being executed, s was not encroached, thus q cannot encroach s', or $|p-\mathcal{C}_d(s')| \le |p-q|$. Using Assumption 1, then $\operatorname{lfs}(p) \le \left(1+\sqrt{2}C_2\right)\mu\,|p-\mathcal{C}_d(s)|$, and so it suffices to take

$$\boxed{\mu\left(1+\sqrt{2}C_2\right) \le C_1}$$

2. Suppose p is the circumcenter of a triangle $\Delta xyz \in \mathcal{D}(\mathcal{P}')$, with $\angle xyz \le \kappa$. Without loss of generality, assume z was added to $\mathcal{P}'$ after x. If x, z are points of $\mathcal{P}$, then $\operatorname{lfs}(z) \le |x-z|$. Otherwise, if z is not an input point then, inductively, $\operatorname{lfs}(z) \le (C_1 \vee C_2)\,|x-z|$. By the sine rule, $|x-z| = 2\,|p-z|\sin\theta \le 2\,|p-z|\sin\kappa$. Thus

$$\operatorname{lfs}(p) \le |p-z| + \operatorname{lfs}(z) \le (1+2\sin\kappa\,[C_1 \vee C_2])\,|p-z|.$$

Because the triangle is Delaunay, it must be that $|p-z| \le |p-q|$. Thus it suffices to take

$$\boxed{1+2\sin\kappa\,[C_1 \vee C_2] \le C_2}$$

Then the following choices of constants work:

$$\boxed{\begin{aligned} C_1 &= \max\left\{\frac{1+\beta}{1-\beta}, \frac{\eta\,(1+C_0)}{\sigma}, \frac{\eta\,(1+\zeta)}{1-2\zeta\eta\sin\kappa}, \frac{\mu\left(1+\sqrt{2}\right)}{1-2\mu\sqrt{2}\sin\kappa}\right\}, \\ C_2 &= 1+2C_1\sin\kappa, \end{aligned}}$$

as long as $\kappa < \arcsin\left(1/2\mu\sqrt{2}\right) \wedge \arcsin\left(1/2\zeta\eta\right)$.

Corollary 3. *If* $\kappa < \arcsin\left(1/2\mu\sqrt{2}\right) \wedge \arcsin\left(1/2\zeta\eta\right)$, *the algorithm will terminate.*

Lemma 6. *Let* $(\mathcal{P}', \mathcal{C}')$ *be the point and curve sets at the termination of Algorithm 1. Then*

1. *For every $c \in \mathcal{C}$ there are $c_0, c_1, \ldots, c_n \in \mathcal{C}'$ such that c is the union of the c_i, and each of the c_i has empty diametral circle with respect to $\mathcal{P}'$, and thus each c_i corresponds to an edge in $\mathcal{D}(\mathcal{P}')$.*
2. *If $\Delta xyz \in \mathcal{D}(\mathcal{P}')$, then either the minimum angle of the triangle is no less than κ, or there is a point $p \in \mathcal{P}$ with*

$$\max(|x-p|, |y-p|, |z-p|) \leq 2d(p).$$

Note that, if midpoints are chosen properly (*cf.* Corollary 1 and Corollary 2), then as $\tau \to 0$, that $\zeta \to \sqrt{2}$, and η and μ go to 1, and thus the bound on κ goes to $\arcsin\left(1/2\sqrt{2}\right)$. This is the classical output bound of the segment input case [2, 1].

Thus the user can select κ arbitrarily close to $\arcsin\left(1/2\sqrt{2}\right)$ by subdividing his/her input to make τ sufficiently small. It is straightforward to automate this process for commonly used curves [5]. However, this preprocessing step is likely unnecessary; rather if κ is set to the desired level, curves with too large variation will automatically be split. An argument of this type was used by the authors in the analysis of a three-dimensional algorithm [10].

From Theorem 1, and the definition of σ, it should be clear that the algorithm can accept input with curves that meet at zero tangent angle. See Figure 4 for an example.

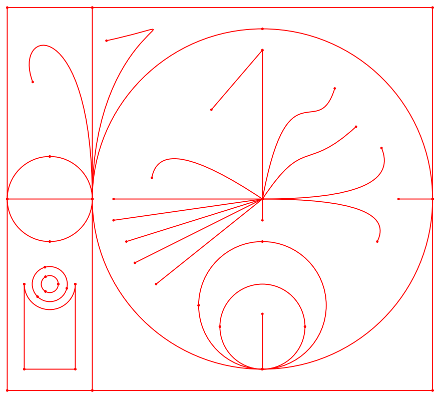
5 Results

A prototype of the algorithm was implemented using CGAL [14]. The code accepts straight segments, circular arcs, and quadratic and cubic curves. Curves are automatically split to meet a user-specified curvature bound, τ.

The input in Figure 4 contains several curves which meet at a zero tangent angle. Where input curves meet with small curve separation, the output mesh contains a large number of large number of triangles, as predicted by the reliance of the grading constants on $1/\sigma$. This is shown in detail in Figure 5, which shows the region just above where the two circles meet a line segment and two cubic curves, all at zero tangent angle. There is small curve separation near the tangency point, which results in a number of small angle triangles.

6 Discussion and Improvements

The presence of large angles in the output meshes is an obvious annoyance, for which a fix has been discovered: The fix involves protecting the ball $\mathcal{B}(p)$ with "pseudo-input" arcs, and enforcing the empty diametral circle property for these arcs. The arcs need to be split to ensure a bound of $\pi/3$ on total curvature, which introduces pseudo-midpoints. When a ball $\mathcal{B}(p)$ is split, the pseudo-input arcs are removed from consideration, while the pseudo-midpoints remain in the mesh.


(a) Input

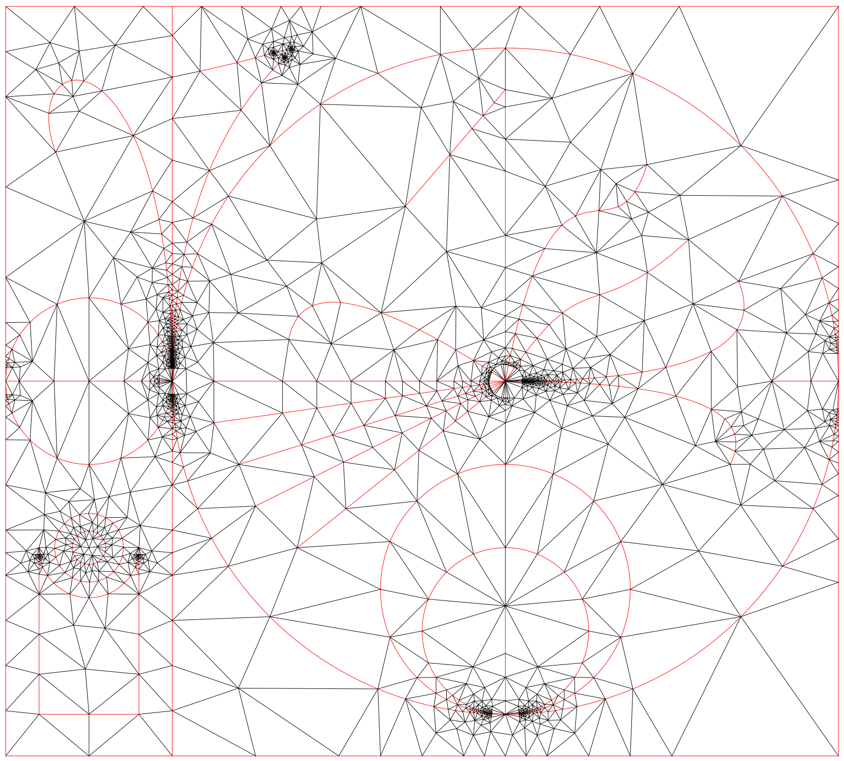

(b) Output

Fig. 4. Results of the prototype code applied to an example PRCC. The input was first subdivided with Steiner Points to insure $\tau = 0.5$. The code was executed with $\beta = 0.95$ and $\kappa = \arcsin\left(1/2\sqrt{2}\right)$. The minimum output angle, however, was only about 0.29°, due to the presence of small angles in the input. The mesh contains 2503 points.

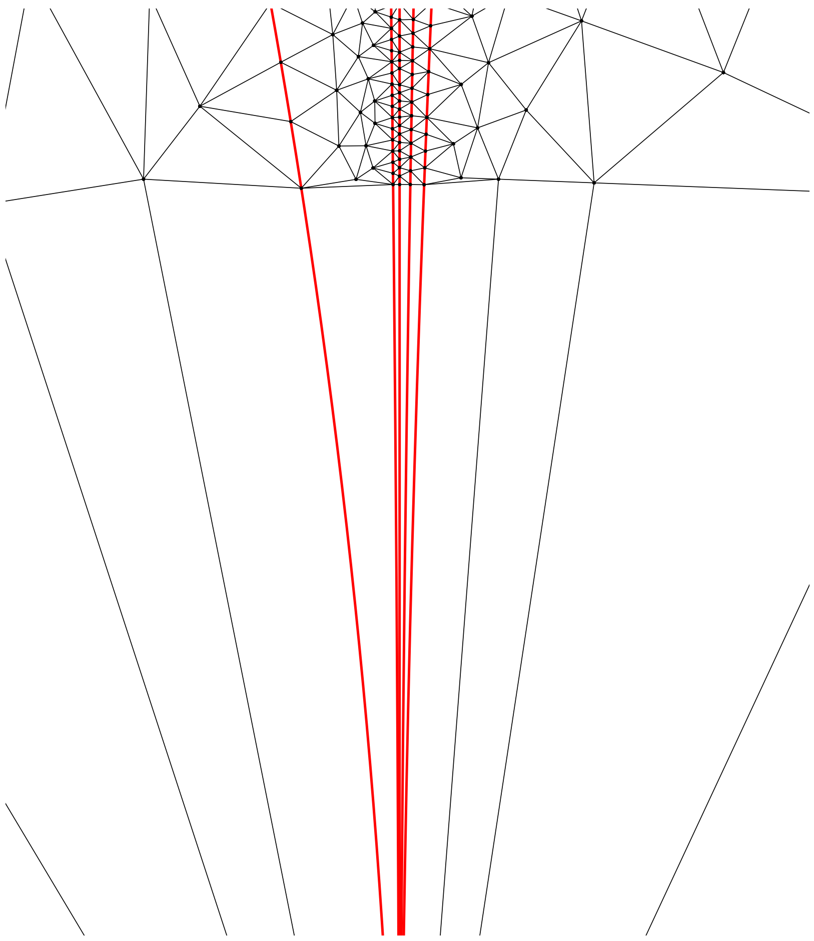


Fig. 5. Detail from Figure 4, showing the region near where the two larger circles touch a line segment and two cubic curves. All five curves meet at a zero tangent angle. The input is shown in bold red lines. Note the presence of triangles with large angles facing small angle triangles emanating from the point of tangency (which is not shown).

Then when a ball $\mathcal{B}(p)$ is split, check the pseudo-midpoints it would generate. If one of these has nearest neighbor closer than $d(p)/2$, split the ball again. If one of the pseudo-midpoints would encroach a segment, split the segment if its diameter is less than $d(p)$, otherwise split the ball again. If the ball need not be split again, add the pseudo-input points to $\mathcal{P}'$.

Then protect the pseudo-input segments as real segments. In particular, if a circumcenter encroaches one, split the pseudo-input segment instead of adding the circumcenter. The only other modification is to, of course, not attempt to kill a poor quality triangle which is wholly contained in a ball $\mathcal{B}(p)$.

The changes in the proof are minimal: C_0 needs to be increased to 3, the proof of Lemma 4 has to include the new reasons a ball $\mathcal{B}(p)$ might be split, and minor changes need to be made in Theorem 1.

This modification is not considered in the main of the paper because of the added complication of the description of the algorithm and its proof, because of the increase in output cardinality (and the following chorus of "Your mesh has too many points!"), and because it has not yet been implemented.

The use of pseudo-input arcs would make the algorithm similar to the initialization part of the three-dimensional Delaunay Refinement Algorithm of these authors, wherein the boundaries of input faces are protected by portions of arcs [10]. The use of an implicit boundary to protect $\mathcal{B}(p)$ in Algorithm 1, as opposed to an explicit circle of pseudo-segments, is more akin to the three-dimensional algorithm of Cheng *et al.* [9]. The presence of large angles in the meshes produced by this two-dimensional algorithm raises doubts about the three-dimensional algorithms which appear to generalize it: do they not also produce simplices with large (solid) angles, even in the absence of small angles, dihedral and otherwise, in the input? It is possible this problem is avoided by the more aggressive approach of Cheng and Poon, who place explicit spherical patches around input facet boundaries to protect small dihedral angles [15]. Among these three algorithms, only that of Cheng *et al.* appears to have been implemented, so they cannot be compared directly. It seems that an aggressive protection strategy may be necessary in three dimensions, but this creates meshes with large cardinalities, and papers with impenetrable technicalities.

The ambiguity in the choice of a midpoint by SPLITCURVE, *i.e.*, that it follows Assumption 1, allows room for heuristic modifications to the algorithm. In particular, it may not always be desireable to split a curve $\widetilde{xy}$ by the point where it intersects the bisector of x and y. In the case where point z encroaches $\widetilde{xy}$, one may instead choose the intersection of the angle bisector of $\angle xzy$ with the curve as the splitting point. Using this heuristic on the input shown in Figure 4 results in a mesh with approximately 10% more points. However, in some situations this heuristic can slightly improve the maximum output angle of the mesh.

The choice of β appears to affect mesh cardinality for some input. For the input of Figure 4, the use of $\beta = 0.5$ results in a mesh with approximately 32% more points, compared to the use of $\beta = 0.95$. Note that the term $(1 + \beta)/(1 - \beta)$ which bounds C_1 from below is modest for $\beta \approx \frac{1}{2}$. That is, it should be smaller than the other terms which bound C_1, so there appears to be little lost by increasing β, as was seen in this case. By no means, however, is there a clear relationship between β and mesh cardinality.

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. 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`
3. 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
4. Miller G.L., Pav S.E., Walkington N.J. "When and Why Delaunay Refinement Algorithms Work." *Internat. J. Comput. Geom. Appl.*, vol. 15, no. 1, 25–54, 2005. Special Issue: Selected Papers from the 12th International Meshing Roundtable, 2003
5. Boivin C., Ollivier-Gooch C.F. "Guaranteed-Quality Triangular Mesh Generation For Domains with Curved Boundaries." *Internat. J. Numer. Methods Eng.*, vol. 55, no. 10, 1185–1213, 2002
6. Miller G.L., Pav S.E., Walkington N.J. "Fully Incremental 3D Delaunay Mesh Generation." *Proceedings of the 11th International Meshing Roundtable*, pp. 75–86. Sandia National Laboratory, September 2002
7. Murphy M., Mount D.M., Gable C.W. "A point-placement strategy for conforming Delaunay tetrahedralization." *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pp. 67–74. Society for Industrial and Applied Mathematics, 2000
8. Cohen-Steiner D., de Verdiere E.C., Yvinec M. "Conforming Delaunay Triangulations in 3D." CS 4345, INRIA, 2001
9. Cheng S.W., Dey T.K., Ramos E.A., Ray T. "Quality meshing for polyhedra with small angles." *SCG '04: Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pp. 290–299. ACM Press, 2004
10. Pav S.E., Walkington N.J. "Robust Three Dimensional Delaunay Refinement." *Proceedings of the 13th International Meshing Roundtable*, pp. 145–156. Sandia National Laboratory, September 2004
11. Babuška I., Aziz A.K. "On the angle condition in the finite element method." *SIAM J. Numer. Anal.*, vol. 13, no. 2, 214–226, 1976
12. Ciarlet P.G. *The Finite Element Method for Elliptic Problems*. North–Holland, 1978
13. Amenta N., Choi S., Kolluri R.K. "The power crust." *SMA '01: Proceedings of the sixth ACM symposium on Solid modeling and applications*, pp. 249–266. ACM Press, New York, NY, USA, 2001
14. Fabri A. "CGAL - The Computational Geometry Algorithm Library." *Proceedings of the 10th International Meshing Roundtable*, pp. 137–142. Sandia National Laboratory, October 2001
15. 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