Off-centers: A new type of Steiner points for computing size-optimal quality-guaranteed Delaunay triangulations*

Alper Üngör

Department of Computer Science
Duke University, Durham, NC 27708, USA
ungor@cs.duke.edu

Abstract. We introduce a new type of Steiner points, called off-centers, as an alternative to circumcenters, to improve the quality of Delaunay triangulations. We propose a new Delaunay refinement algorithm based on iterative insertion of off-centers. We show that this new algorithm has the same quality and size optimality guarantees of the best known refinement algorithms. In practice, however, the new algorithm inserts about 40% fewer Steiner points (hence runs faster) and generates triangulations that have about 30% fewer elements compared with the best previous algorithms.

Keywords. Delaunay refinement, computational geometry, triangulations

1 Introduction

Meshes are heavily used in many applications including engineering simulations, computer-aided design, solid modeling, computer graphics, and scientific visualization. Most of these applications require that the shape of the mesh elements are of good quality and that the size of the mesh is small. An element is said to be good if its aspect ratio (circumradius over inradius) is bounded from above or its smallest angle is bounded from below. Mesh element quality is critical in determining interpolation error in the applications and hence is an important factor in the accuracy of simulations as well as the convergence speed. Mesh size, meaning the number of elements, is also a big factor in the running time of the applications algorithm. Between two meshes with the same quality bound, the one with fewer elements is preferred almost exclusively.

Among several types of domain discretizations, unstructured meshes, in particular Delaunay triangulations, are quite popular due to their theoretical guarantees as well as their practical performance. Earliest algorithms that provide both size optimality and quality guarantee used balanced quadtrees to generate first a nicely spread point set and then the Delaunay triangulation of these

* Research of the author is supported by NSF under grant CCR-00-86013.

points [1]. Subsequently, Delaunay refinement techniques are developed based on an incremental point insertion strategy and provide the same theoretical guarantees [9]. Over the last decade, Delaunay refinement has become much more popular than the quadtree-based algorithms mostly due to its superior performance in generating smaller meshes. Many versions of the Delaunay refinement is suggested in the literature [2, 6–11]. We attribute the large amount of research on Delaunay refinement to its impact on a wide range of applications. It is important to generalize the input domains that the Delaunay refinement works, as well as to improve the performance of the algorithm. Even a small but reasonable reduction in mesh size translates to important savings in the running-time of the subsequent application algorithm.

The first step of a Delaunay refinement algorithm is the construction of a constrained or conforming Delaunay triangulation of the input domain. This initial Delaunay triangulation is likely to have bad elements. Delaunay refinement then iteratively adds new points to the domain to improve the quality of the mesh and to ensure that the mesh conforms to the boundary of the input domain. The points inserted by the Delaunay refinement are called *Steiner points*. A sequential Delaunay refinement algorithm typically adds one new vertex on each iteration. Each new vertex is chosen from a set of candidates — the circumcenters of bad triangles (to improve mesh quality) and the mid-points of input segments (to conform to the domain boundary). Ruppert [9] was the first to show that proper application of Delaunay refinement produces well-shaped meshes in two dimensions whose size is within a constant factor of the best possible. There are efficient implementations [10] as well as three-dimensional extensions of Delaunay refinement [4, 10].

In this paper, we introduce a new type of Steiner points, called *off-centers*, as an alternative to circumcenters and propose a new Delaunay refinement algorithm. We show that this new algorithm has the same theoretical guarantees as the Ruppert's algorithm, and hence, generates quality-guaranteed size-optimal meshes. Moreover, experimental study indicates that our Delaunay refinement algorithm with off-centers inserts about 40% fewer Steiner points than the circumcenter insertion algorithms and results in meshes about 30% smaller in the number of elements. This implies substantial reduction not only in mesh generation time, but also in the running time of the application algorithm. For instance a quadratic-time application algorithm, if ran on the new meshes, would take about half the time it takes on the old meshes.

2 Preliminaries

In two dimensions, the input domain Ω is represented as a *planar straight line graph* (PSLG) — a proper planar drawing in which each edge is mapped to a straight line segment between its two endpoints [9]. The segments express the *boundaries* of Ω and the endpoints are the *vertices* of Ω. The vertices and boundary segments of Ω will be referred to as the input *features*. A vertex is incident to a segment if it is one of the endpoints of the segment. Two segments are incident if they share a common vertex. In general, if the domain is given as a collection of vertices only, then the boundary of its convex hull is taken to be the boundary of the input.

The *diametral circle* of a segment is the circle whose diameter is the segment. A point is said to *encroach* a segment if it is inside the segment's diametral circle.

Given a domain Ω embedded in $\mathbb{R}^2$, the *local feature size* of each point $x \in \mathbb{R}^2$, denoted by $\mathrm{lfs}_\Omega(x)$, is the radius of the smallest disk centered at x that touches two non-incident input features. This function is proven [9] to have the so-called *Lipschitz property*, i.e., $\mathrm{lfs}_\Omega(x) \leq \mathrm{lfs}_\Omega(y) + |xy|$, for any two points $x, y \in \mathbb{R}^2$.

Let P be a point set in $\mathbb{R}^d$. A simplex τ formed by a subset of P points is a *Delaunay simplex* if there exists a circumsphere of τ whose interior does not contain any points in P. This empty sphere property is often referred to as the *Delaunay property.* The Delaunay triangulation of P, denoted $Del(P)$, is a collection of all Delaunay simplices. If the points are in general position, that is, if no $d+2$ points in P are co-spherical, then $Del(P)$ is a simplicial complex. The Delaunay triangulation of a point set of size n can be constructed in $O(n \log n)$ time in two dimensions [5].

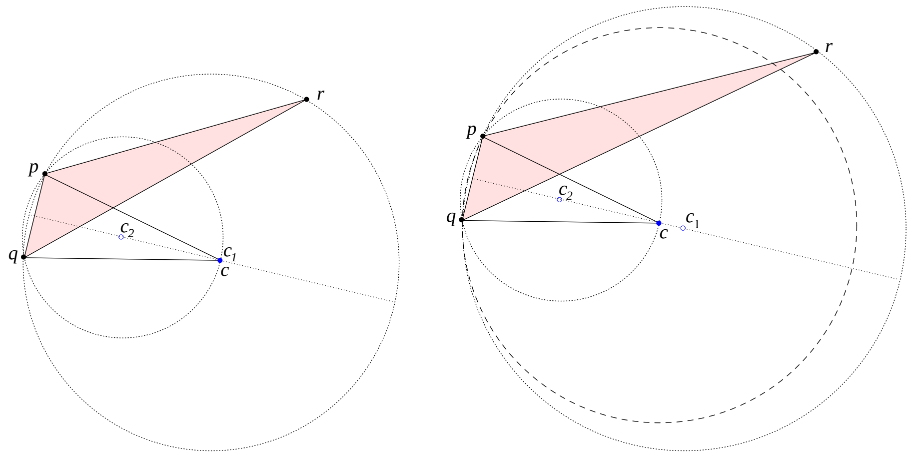
In the design and analysis of the Delaunay refinement algorithms, a common assumption made for the input PSLG is that the input segments do not meet at junctions with small angles. Ruppert [9] assumed, for instance, that the smallest angle between any two incident input segment is at least 90°. A typical Delaunay refinement algorithm may start with the *constrained Delaunay triangulation* [3] of the input vertices and segments or the Delaunay triangulation of the input vertices. In the latter case, the algorithm first splits the segments that are encroached by the other input features. Alternatively, for simplicity, we can assume that no input segment is encroached by other input features. A preprocessing algorithm, which is also parallelizable, to achieve this assumption is given in [11].

Radius-edge ratio of a triangle is the ratio of its circumradius to the length of its shortest side. A triangle is considered *bad* if its radius-edge ratio is larger than a pre-specified constant $\beta \geq \sqrt{2}$. This quality measure is equivalent to other well-known quality measures, such as smallest angle and aspect ratio, in two dimensions [9].

3 Delaunay Refinement with Off-centers

3.1 Off-centers

The line that goes through the midpoint of an edge of a triangle and its circumcenter is called the *bisector* of the edge. Given a bad triangle pqr, suppose that its shortest edge is pq. Let c denote the circumcenter of pqr. We define the *off-center* to be the circumcenter of pqr if the radius-edge-ratio of pqc is smaller than or equal to β (Figure 1 (a)). Otherwise, the *off-center* is the point on the bisector (and inside the circumcircle), which makes the radius-edge ratio of the triangle based on p, q and the off-center itself exactly β (Figure 1 (b)). The circle that is centered at the off-center and goes through the endpoints of the



(a) (b)

Fig. 1. The off-center and the circumcenter of triangle pqr is labeled c and c_1 respectively. The circumcenter of pqc is labeled as c_2. If $|cc_2| \leq \beta|pq|$ then $c = c_1$ (a). Otherwise, $c \neq c_1$ and by construction $|cc_2| = \beta|pq|$ (b). The off-circle of pqr is same as the circumcircle in (a) and shown as dashed circle in (b).

shortest edge is called the *off-circle*. In the first case, off-circle is same as the circumcircle of the triangle. A bad triangle can have two shortest edges. In such cases, the off-center is defined once we arbitrarily choose one of the two edges as the shortest.

Notice in Figure 1 (b) that, if we were to insert the circumcenter c_1, the triangle pqc_1 would still be bad and require another circumcenter insertion. We instead suggest to insert just the off-center c. This, of course, is a simplified picture and the actual behavior of Delaunay refinement is more complicated. Nevertheless, this very observation is the main intuition behind the expectation of smaller size meshes. In other words, around a small feature we create a good element with the longest possible new features.

3.2 Algorithm

At each iteration, we choose a new point for insertion from a set of candidate points. There are two kinds of candidate points: (1) the off-centers of bad triangles, and (2) the midpoints of segments. Let $\dot{\mathcal{C}}$ denote the set of all candidate off-centers that do not encroach any segment. Let $\mathcal{C}$ denote their corresponding off-circles. Similarly, let $\dot{\mathcal{B}}$ denote the set of all candidate off-centers that do encroach some segment. Candidate off-centers of this second type are *rejected* from insertion. Let $\mathcal{B}$ denote their corresponding off-circles. The midpoint of a boundary segment is a *candidate* for insertion if it is encroached by an off-center in $\dot{\mathcal{B}}$. Let $\dot{\mathcal{D}}$ be all midpoint candidates. Then we suggest the following algorithm to incrementally insert the candidate points.

Algorithm 1 DELAUNAY REFINEMENT WITH OFF-CENTERS

Input: A PSLG domain Ω in $\mathbb{R}^2$
  Let T be the Delaunay triangulation of the vertices of Ω.
  Compute $\dot{\mathcal{B}}$, $\dot{\mathcal{C}}$, $\dot{\mathcal{D}}$;
  while $\dot{\mathcal{C}} \cup \dot{\mathcal{D}}$ is not empty **do**
    Choose a point q from $\dot{\mathcal{C}} \cup \dot{\mathcal{D}}$ and insert q into the triangulation. If q is a midpoint of a segment s, replace s with two segments from q to each endpoint of s;
    Update the Delaunay triangulation T and recompute $\dot{\mathcal{B}}$, $\dot{\mathcal{C}}$, $\dot{\mathcal{D}}$.
  end while

4 Termination and Size Optimality

When analyzing his algorithm, Ruppert [9] used the Delaunay property on the bad triangles, that is, their circumcircles are empty of other points. Unfortunately, the off-circles are not necessarily empty of other points. There is a small crescent-shape possibly non-empty region of each off-circle outside the corresponding circumcircle. This raises a challenge in our analysis. One easy way around this is to use a special insertion order among the off-centers. For instance, it is relatively easy to prove that the off-circle of the bad triangle that has the shortest edge is empty of all other points. Alternatively, an ordering that favors the bad triangles with the smallest circumradius serves for the same purpose. We could use one of these ordering strategies and apply the same arguments given in [9]. However, for the sake of a generic result, we opt for an arbitrary order in the analysis of our off-center insertion algorithm.

We prove that the meshes generated by the off-center insertion algorithm is size optimal using the same machinery as Ruppert [9]. Moreover, we adapt the terminology introduced in [10] which includes a clearer rewrite of Ruppert's results. We first prove that the edge length function is within a constant factor of the local feature size. Then, we conclude that the output mesh is size-optimal within a constant.

Let *insertion length* of a vertex u, denoted r_u, be the length of the shortest edge incident to u right after u is inserted (or were to be inserted if u is encroaching). If u is an input vertex its insertion length is the shortest edge incident to u in the initial Delaunay triangulation of the input. Also, for each Steiner vertex u, we define a *parent vertex*, denoted $\hat{u}$, as the most recently inserted endpoint of the shortest edge of the bad triangle responsible of the insertion of u. This definition applies also for vertices that are considered but not actually inserted due to encroachment.

Lemma 1. *Let pqr be a bad triangle with off-center u. Then, $r_u \geq C_0|u\hat{u}|$, for some constant C_0. Moreover, $r_u \geq \beta r_{\hat{u}}$.*

Proof. Without loss of generality, let pq be the shortest edge of pqr and $\hat{u} = p$. Consider the following two cases:

- *u is the circumcenter of pqr:* By the Delaunay property, $r_u \geq |u\hat{u}|$, that is $C_0 = 1$. Moreover, since the triangle pqr is bad, $|u\hat{u}|/|pq| \geq \beta$. The distance from p to q is at least $r_p = r_{\hat{u}}$. Hence, $r_u \geq \beta r_{\hat{u}}$.
- *u is not the circumcenter of pqr:* Let m be the midpoint of the segment pq and c_2 be the circumcenter of pqu. See Figure 1. The intersection of the off-circle and the circumcircle is empty by the Delaunay property. So, as a conservative bound, r_u is at least $|um|$. By construction, $\angle pum = \arcsin(\frac{1}{2\beta})/2$. Also, on the right triangle pum, $cos(\angle pum) = |um|/|u\hat{u}|$. Since $\beta \geq \sqrt{2}$, $|um| \geq |u\hat{u}| \cos(\arcsin(\frac{1}{2\sqrt{2}})/2)$. So, $C_0 = \cos(\arcsin(\frac{1}{2\sqrt{2}})/2) \approx 0.98$. Moreover,

$$\begin{aligned} r_u \geq |um| \geq |uc_2| & \qquad \text{(because } \angle pc_2q < 90^\circ\text{)} \\ = \beta|pq| & \qquad \text{(by construction)} \\ \geq \beta r_{\hat{u}} & \qquad \square \end{aligned}$$

Lemma 2. *For each vertex u, either $r_u \geq \mathrm{lfs}_\Omega(u)$ or $r_u \geq C_1 r_{\hat{u}}$, for some constant C_1.*

Proof. We consider the following cases:

- *u is not a Steiner vertex:* Then, its nearest neighbor in the initial triangulation is at most $\mathrm{lfs}_\Omega(u)$ away, hence $r_u \geq \mathrm{lfs}_\Omega(u)$.
- *u is an off-center Steiner vertex:* Then, by Lemma 1 we know that $r_u \geq \beta r_{\hat{u}}$, that is $C_1 = 1$.
- *u is midpoint of an encroached subsegment s:* If $\hat{u}$ is an input vertex, or is a Steiner vertex on a segment then $r_u \geq \mathrm{lfs}_\Omega(u)$. Otherwise, $\hat{u}$ is an encroaching rejected circumcenter. Let v be the nearest endpoint of s from $\hat{u}$. By definition, $r_{\hat{u}}$ is at most $|\hat{u}v|$. Moreover, since $\hat{u}$ is inside the diametral circle of s, $|\hat{u}v| \leq \sqrt{2} r_u$. Therefore, $r_u \geq r_{\hat{u}}/\sqrt{2}$, that is $C_1 = 1/\sqrt{2}$. $\square$

Theorem 1. *The* Delaunay Refinement with Off-centers *terminates.*

Proof. Let $\underline{\mathrm{lfs}}$ be the smallest distance between two non-incident features of the input PSLG. We prove, by contradiction that there are no edges shorter than $\underline{\mathrm{lfs}}$ introduced during the refinement. Suppose e is the first edge that is shorter than $\underline{\mathrm{lfs}}$. Then, at least one end-point of e is a Steiner vertex. Let v be the most recently inserted endpoint of e. Let $\hat{\hat{v}}$ be the grandparent of v.

- If v is the off-center of a bad triangle, then by Lemma 2, $r_v \geq \beta r_{\hat{v}}$.
- If v is the midpoint of an encroached segment then there are two sub-cases. If $\hat{v}$ is the off-center of a bad triangle, then by Lemma 2, $r_v \geq r_{\hat{v}}/\sqrt{2} \geq \beta r_{\hat{\hat{v}}}/\sqrt{2} \geq r_{\hat{\hat{v}}}$. Otherwise, $\hat{v}$ is on a non-incident segment because of the PSLG input assumption. Then, clearly $r_v \geq \underline{\mathrm{lfs}}$.

In all cases, $r_v \geq r_u$ for some ancestor u of v. If $r_v < \underline{\mathrm{lfs}}$, then $r_u < \underline{\mathrm{lfs}}$, contradicting the assumption that e was the first such edge. Hence, the termination of the algorithm follows. This also implies that there are no bad triangles in the output mesh. $\square$

For each vertex u, let D_u be the ratio of $\mathrm{lfs}_\Omega(u)$ over r_u.

Lemma 3. *If $r_u \geq r_{\hat{u}}/C_2$ for some constant C_2, then $D_u \leq 1/C_0 + C_2 D_{\hat{u}}$.*

Proof.
$$\begin{aligned} D_u = \mathrm{lfs}_\Omega(u)/r_u &\leq (\mathrm{lfs}_\Omega(\hat{u}) + |u\hat{u}|)/r_u && \text{(By Lipschitz property)} \\ &\leq (D_{\hat{u}} r_{\hat{u}} + r_u/C_0)/r_u && \text{(By definition and Lemma 1)} \\ &\leq (D_{\hat{u}} C_2 r_u + r_u/C_0) r_u \\ &= C_2 D_{\hat{u}} + 1/C_0 \end{aligned}$$
□

Lemma 4. *There exist fixed constants $C_T \geq 1$ and $C_S \geq 1$ such that, for each vertex u, $D_u \leq C_T$ if u is a Steiner or rejected off-center vertex and $D_u \leq C_S$ if u is a midpoint Steiner vertex.*

Proof. We prove the lemma by induction.
Basis: If $\hat{u}$ is an input vertex or on a segment, then $D_{\hat{u}} = \mathrm{lfs}_\Omega(\hat{u})/r_{\hat{u}} \leq 1$.
Induction hypothesis: Lemma holds for vertex $\hat{u}$. So, $D_{\hat{u}} \leq \max\{C_T, C_S\}$.
Induction: Now we make a case analysis:

- If u is an off-center of a bad triangle, then by Lemma 3 (where $C_2 = 1/\beta$ by Lemma 1) and the induction hypothesis, $D_u \leq \frac{1}{C_0} + \max\{C_T, C_S\}/\beta$. This implies that $D_u \leq C_T$ if
$$C_T \geq \frac{1}{C_0} + \max\{C_T, C_S\}/\beta \tag{1}$$
- Otherwise, u is a midpoint of a subsegment s. If parent is an input vertex or on another segment, lemma holds by the basis of the induction. If $\hat{u}$ is a rejected off-center of a bad triangle, then by Lemma 2, $r_u \geq r_{\hat{u}}/\sqrt{2}$. So, by Lemma 3 (where $C_2 = \sqrt{2}$) and the induction hypothesis, $D_u \leq \frac{1}{C_0} + \sqrt{2} C_T$. This implies that $D_u \leq C_S$ if
$$C_S \geq \frac{1}{C_0} + \sqrt{2} C_T \tag{2}$$

We choose $C_S = \frac{\beta(\sqrt{2}+1)}{C_0(\beta-\sqrt{2})}$ and $C_T = \frac{\beta+1}{C_0(\beta-\sqrt{2})}$, to satisfy both Inequalities (1) and (2). Hence the lemma holds. □

Lemma 5. *For each vertex u of the output mesh, its nearest neighbor vertex v is at a distance at least $C_3 \mathrm{lfs}_\Omega(u)$ for some constant C_3.*

Proof. By Lemma 4, $\mathrm{lfs}_\Omega(u)/r_u \leq C_S$, for any vertex u. If u was inserted after v, then $|uv|$ is at least r_u. Hence, $|uv| \geq r_u \geq \mathrm{lfs}_\Omega(u)/C_S$, and the lemma holds. If v was inserted after u, then by Lemma 4 $|uv| \geq r_v \geq \mathrm{lfs}_\Omega(v)/C_S$. By Lipschitz property, $|uv| \geq (\mathrm{lfs}_\Omega(u) - |uv|)/C_S$. Hence, $|uv| \geq \mathrm{lfs}_\Omega(u)/(C_S + 1)$, that is, $C_3 = 1/(C_S + 1)$. □

Local feature size for an output mesh M (which is a PSLG) is well-defined and denoted by $\mathrm{lfs}_M()$. Previous lemma essentially states that $\mathrm{lfs}_M(x) \geq \mathrm{lfs}_\Omega(x)$, $\forall x \in M$. We next state a theorem proven by Ruppert [9], which together with Lemma 5 leads to Theorem 3, the main result of this section.

Theorem 2 ([9]). *Suppose a triangulation M with radius-edge ratio bound β has the property that there is some constant C_4 such that $\mathrm{lfs}_M(p) \geq \mathrm{lfs}_\Omega(p)/C_4$, $\forall p \in \mathbb{R}^2$. Then, the size of T is less than C_5 times the size of any triangulation of the input Ω with bounded radius-edge ratio β, where $C_5 = O(C_4^2\beta)$.*

Theorem 3. *The* Delaunay Refinement with Off-centers *algorithm generates a size-optimal mesh.*

5 Experiments

Implementing the Delaunay refinement with off-centers is as simple as replacing the circumcenter procedure in classical Delaunay refinement implementations with a new off-center procedure. Computing off-centers and circumcenters are very similar and take roughly the same time. Hence, savings in the number of Steiner points reported below also reflects the amount mesh generation time.

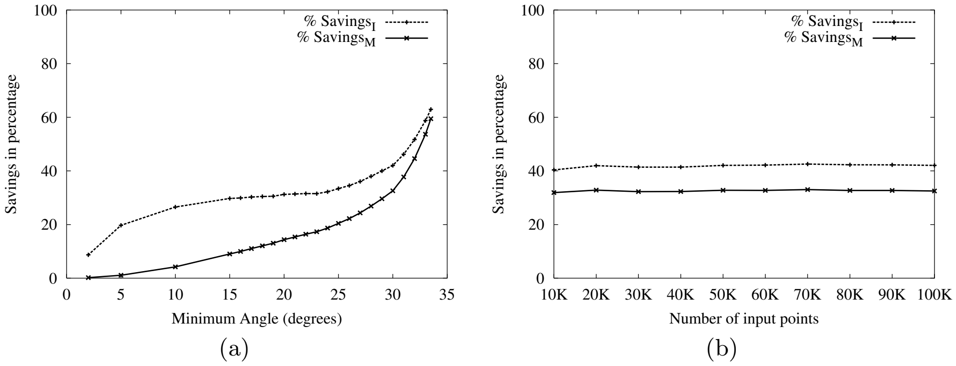


Fig. 2. (a) Percentage savings when the number of input points is 10K and the minimum angle threshold samples the interval [2°-34°]. (b) Percentage savings when the minimum angle threshold is 30° and the number of input points samples [10K-100K].

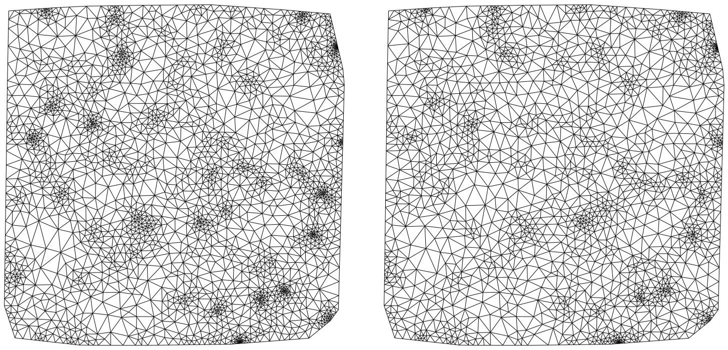
Earlier experiments with circumcenter insertion method indicates that the insertion order has an impact on the output mesh size. For instance, inserting the circumcenter of worst triangles first tends to result in smaller meshes. In this study, for fairness of comparison, we chose the ordering strategy that performs the best for the circumcenter insertion and use the same for the off-center insertion. For Delaunay refinement with circumcenters we used the CMU software `triangle`[1] [10], which is reported to have over thousand users.

[1] Available at `http://www-2.cs.cmu.edu/~quake/triangle.html`

Figure 2 illustrates a summary of our experimental results on randomly generated point sets. Let S_c and S_o be the number of Steiner points inserted by the circumcenter and the off-center insertion methods, respectively. Also, let M_c and M_o be the number of elements generated by the circumcenter and the off-center insertion methods, respectively. We report the following two measures:

$$Savings_I = \frac{S_c - S_o}{S_c}, \qquad Savings_M = \frac{M_c - M_o}{M_c}$$

Percentage savings both in the number of Steiner points and in the mesh size increases as the user specified minimum angle threshold gets higher (Figure 2 (a)). We also observed that for a given threshold angle, the savings remain consistent as we change the input size (Figure 2 (b)). For a visual comparison of the off-center and the circumcenter insertion algorithms see Figure 3 where the input is a randomly generated point set.



(a) (b)

Fig. 3. Input consists of 500 points. Smallest angle in both meshes is 29°. Circumcenter insertion adds 2399 Steiner points resulting a mesh with 4579 triangles (a). Off-center insertion adds 1843 Steiner points resulting a mesh with 3479 triangles (b).

6 Discussions

By definition, the off-center of some triangles is same as their circumcenter. The off-center and the circumcenter insertion algorithms are likely to generate very similar (sometimes the same) meshes when the initial triangulation is reasonably good to begin with. In most applications, however, tiny angles are ubiquitous in the initial Delaunay triangulation. Figure 4 demonstrates the output of the two algorithms in one such case. In this example, our off-center insertion algorithm gives a mesh that is a factor six smaller than the output of `triangle`. We also observed many other examples, where the off-center insertion algorithm terminates (computing a quality-bounded mesh) and `triangle` does not.

This new insertion scheme also leads to a parallel Delaunay refinement algorithm that takes only $O(\log(L/h))$ iterations to generate quality-guaranteed

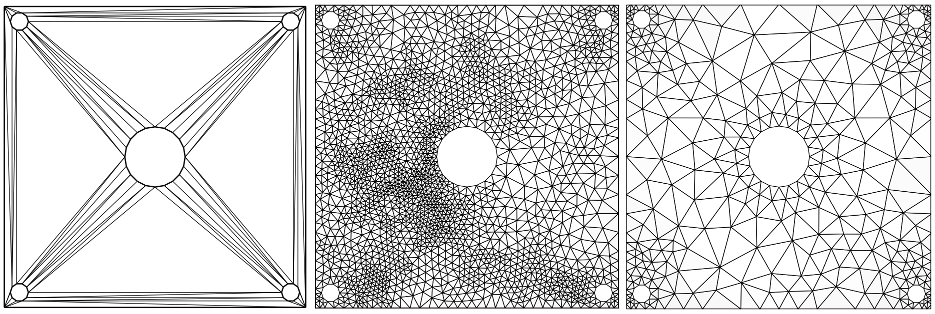


Fig. 4. Input PSLG is a plate with five holes described by 64 points and 64 segments. Smallest angle in the initial triangulation (a) is about 1°. Smallest angle in both output triangulations is 34°. Circumcenter insertion (`triangle` software) introduces 1984 Steiner points resulting a mesh with 3910 triangles (b). Off-center insertion introduces only 305 Steiner points resulting a mesh with 601 triangles (c).

size-optimal meshes, where L is the diameter of the domain, and h is the smallest edge length in the initial triangulation. This is an improvement over the previously best known equivalent algorithm that runs $O(\log^2(L/h))$ iterations [11]. Due to space limitations, we do not include the description and the analysis of the new parallel off-center insertion algorithm in this publication. Furthermore, we plan to extend the off-center algorithm to three dimensions and explore its benefits both in theoretical and practical fronts.

References

1. M. Bern, D. Eppstein, and J. Gilbert. Provably good mesh generation. *J. Comp. System Sciences* 48:384–409, 1994.
2. L.P. Chew. Guaranteed-quality triangular meshes. TR-89-983, Cornell Univ., 1989.
3. L.P. Chew. Constrained Delaunay triangulations. *Algorithmica* 4:97–108, 1989.
4. T.K. Dey, C. L. Bajaj, and K. Sugihara. On good triangulations in three dimensions. *Int. J. Computational Geometry & Applications* 2(1):75–95, 1992.
5. H. Edelsbrunner. *Geometry and Topology for Mesh Generation*. Cambridge Univ. Press, 2001.
6. H. Edelsbrunner and D. Guoy. Sink insertion for mesh improvement. *Proc. 17th ACM Symp. Comp. Geometry*, 115–123, 2001.
7. G.L. Miller. A time efficient Delaunay refinement algorithm. *Proc. ACM-SIAM Symp. on Disc. Algorithms*, (to appear), 2004.
8. G.L. Miller, S. Pav, and N. Walkington. When and why Ruppert's algorithm works. *Proc. 12th Int. Meshing Roundtable*, 91–102, 2003.
9. J. Ruppert. A new and simple algorithm for quality 2-dimensional mesh generation. *Proc. 4th ACM-SIAM Symp. on Disc. Algorithms*, 83–92, 1993.
10. J.R. Shewchuk. *Delaunay Refinement Mesh Generation*. Ph.D. thesis, Carnegie Mellon University, 1997.
11. D.A. Spielman, S.-H. Teng, and A. Üngör. Parallel Delaunay refinement: Algorithms and analyses. *Proc. 11th Int. Meshing Roundtable*, 205–217, 2002.