Delaunay Refinement

`Topics:` mesh generation, triangle quality, Delaunay refinement, algorithm, local feature size, constants, invariants, analysis.

Mesh generation. The general objective in mesh generation is to decompose a geometric space into elements. The elements are restricted in type and shape, and the number of elements should not be too big. We discuss a concrete version of the 2-dimensional mesh generation problem.

`Input.` A polygonal region in the plane, possibly with holes and with constraining edges and vertices inside the region.

`Output.` A triangulation of the region whose edges cover all input edges and whose vertices cover all input vertices.

The graph of input vertices and edges is denoted by G, and the output triangulation is denoted by K. It is convenient to enclose G in a bounding box and to triangulate everything inside that box. A triangulation of the input region is obtained by taking a subset of the triangles. Figure 18 shows input and output of a mesh generation problem.

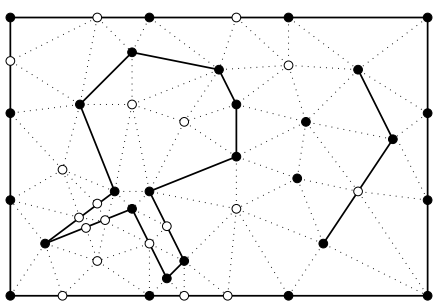

Figure 18: The solid vertices and edges define the input graph, and together with the hollow vertices and dotted edges they define the output triangulation.

Triangle quality. The quality of a triangle abc is expressed by its smallest angle, θ. Two alternative choices would be the largest angle and the aspect ratio. We argue that a good lower bound for the smallest angle implies good bounds for the other two expressions of triangle quality. The largest angle is at most $\pi - 2\theta$, so if the smallest angle is bounded away from zero then the largest angle is bounded away from π. The aspect ratio is the length of the longest edge, which we assume is ac, divided by the distance of b from ac. We have

$$\frac{1}{\sin\theta} \leq \frac{\|a-c\|}{\|b-ac\|} \leq \frac{2}{\sin\theta}.$$

In words, if the smallest angle is bounded away from zero then the aspect ratio can be neither too small not too large.

The goal is to construct K so its smallest angle is no less than some constant, α, and the number of triangles in K is at most some constant times the minimum. We see from the example in Figure 18 that a small angle between two input edges cannot possibly be resolved. A reasonable way to deal with this difficulty is to accept sharp input features as unavoidable and to isolate them so they cause no deterioration of the triangulation nearby. In these notes we assume there are no sharp input features and in particular that all input angles are at least $\frac{\pi}{2}$.

Delaunay refinement. We construct K as the Delaunay triangulation of a set of points that includes all input points. Other points are added one by one to resolve input edges that are not covered and triangles that have too small an angle.

1. Suppose ab is a segment of an edge in G that is not covered by edges of the current Delaunay triangulation. This can only be because some of the vertices lie inside the diameter circle of ab, see Figure 19. We say these vertices *encroach upon* ab, and we use function $\textsc{Split}_1$ to add the midpoint of ab and to repair the Delaunay triangulations with a series of flips.

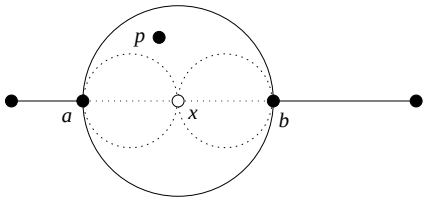

Figure 19: Vertex p encroaches upon segment ab. After adding the midpoint we have too smaller dotted diameter circles both contained in the diameter circle of ab.

2. Suppose a triangle abc in the current Delaunay triangulation K is skinny, that is, it has an angle less than α. We use function $\textsc{Split}_2$ to add the circumcenter as a new vertex, see Figure 20. Since the circumcircle of abc is no longer empty, it is guaranteed to be removed by one of the flips used to repair the Delaunay triangulation.

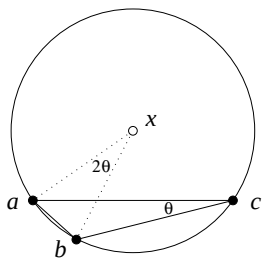

Figure 20: The angle $\angle axb$ is twice the angle $\angle acb$.

Algorithm. The first priority of the algorithm is to cover all input edges and the second is to resolve skinny triangles. Before starting the algorithm we draw the boundary of a 3-by-3 packing of minimum bounding rectangles, and we place G inside the center rectangle. The box is bounded by 12 edges, see Figure 18 where the box is too small but has the right combinatorics. Initially, K is the Delaunay triangulation of the input points, which includes the 12 vertices of B.

```
loop
  while ∃ encroached upon segment ab do
    SPLIT1(ab)
  endwhile;
  if no skinny triangle left then exit endif;
  let abc ∈ K be skinny and x its circumcenter;
  x encroaches upon segments s1, s2, ..., sk;
  if k ≥ 1 then SPLIT1(si) for all i
           else SPLIT2(abc)
  endif
forever.
```

The choice of B implies that no circumcenter x will ever lie outside the box. Initially, all 12 triangles next to the box boundary are non-obtuse so their circumcenters lie inside B. The algorithm maintains the non-obtuseness of triangles next to input edges. Since the circumcircles of Delaunay triangles are empty, this implies that all circumcenters lie inside B.

It remains to prove that the algorithm halts. Assume for the time being that it does and that n is the number of vertices in the final triangulation. The most expensive step of the algorithm is flipping edges to repair the Delaunay triangulation. The expected linear upper bound on the number of flips in the randomized incremental construction does not apply because points are not added in a random order. The number of flips per new vertex is bounded by the degree, which is less than n. The total amount of time to run the algorithm is therefore in $O(n^2)$.

Local feature size. We can understand the algorithm better if we relate its actions to the *local feature size* defined as a map $f : \mathbb{R}^2 \to \mathbb{R}$. For a point $x \in \mathbb{R}^2$, $f(x)$ is the smallest radius r such that the closed disk with center x and radius r

(i) contains two vertices of G,

(ii) intersects one edge of G and contains one vertex of G that is not endpoint of that edge, or

(iii) intersects two vertex disjoint edges of G,

see Figure 21. Because of (i) we have $f(p) \leq \|a - b\|$ for all vertices $a, b \in G$. The local feature size is con-

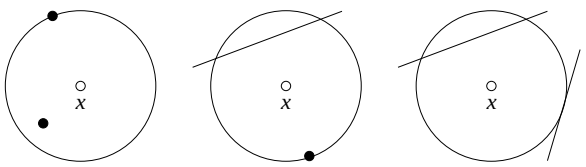

Figure 21: In each case the radius of the circle is the local feature size at x.

tinuous. To see this assume $f(x) < f(y) - c$, for some positive constant c, but $\|x - y\| < c$. Then the disk of x is contained in the interior of the disk of y, which contradicts the definition of $f(y)$. This argument proves a Lipschitz condition, which is stronger than continuity.

CLAIM 8. $|f(x) - f(y)| \leq \|x - y\|$.

Constants. The analysis of the algorithm uses two carefully chosen positive constants C_1 and C_2 such that

$$1 + \sqrt{2}C_2 \quad \leq \quad C_1 \quad \leq \quad \frac{C_2 - 1}{2\sin\alpha}. \qquad (1)$$

The constraints that correspond to the two inequalities are bounded by lines, and we have a solution iff the slope of the first line is greater than that of the second: $1/\sqrt{2} > 2\sin\alpha$. Figure 22 illustrates the two constraints for $\alpha < 20.7°$. In this case the two lines intersect at a point in the positive quadrant. The coordinates of that point are the smallest constants C_1 and C_2 that satisfy (1).

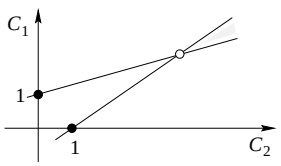

Figure 22: For $\alpha < 20.7°$ the shaded region of pairs (C_1, C_2) that satisfy (1) is non-empty.

Invariants. We show that with the above constants the distance between new and old points cannot be much smaller than the local feature size at the new point.

INVARIANT 9. Let p and x be two vertices such that x is added after p. If x is added by

(A) SPLIT$_1$ then $\|x - p\| \geq f(x)/C_1$,

(B) SPLIT$_2$ then $\|x - p\| \geq f(x)/C_2$.

PROOF. We use induction to prove (B), where x is the circumcenter of a skinny triangle abc. Let $\theta < \alpha$ at c be the smallest angle, see Figure 20. Assume that either a and b both belong to G or that a is added after b. We distinguish three cases depending on how a became to be a vertex. Let L be the length of ab.

`Case 1.` a is a vertex of G. Then b is also a vertex of G and $f(a) \leq L$.

`Case 2.` a is added as circumcenter of a circle with radius r'. Before the addition of a the circumcircle is empty and hence $r' \leq L$. By induction we have $f(a) \leq r' \cdot C_2$ and therefore $f(a) \leq L \cdot C_2$.

`Case 3.` a is added as the midpoint of a segment. Then $f(a) \leq L \cdot C_1$ by the same argument as in Case 2.

Since $1 \leq C_2 \leq C_1$ we have $f(a) \leq L \cdot C_1$ in all three cases. Let $r = \|x - a\|$ be the radius of the circumcircle of abc. Using the Lipschitz condition and $\sin\theta = \frac{L}{2r}$ from Figure 20 we get

$$\begin{aligned} f(x) &\leq f(a) + r \\ &\leq L \cdot C_1 + r \\ &\leq 2r \cdot \sin\theta \cdot C_1 + r. \end{aligned}$$

Since $\theta < \alpha$ and $C_2 \geq 1 + 2C_1 \sin\alpha$ we get

$$r \;\geq\; \frac{f(x)}{1 + 2C_1\sin\alpha} \;\geq\; \frac{f(x)}{C_2},$$

as required. We use a similar argument to prove (A), where x is the midpoint of a segment ab. Let p be a vertex that encroaches upon ab, see Figure 19. Consider first the case where p lies on a segment that shares no endpoint with ab. Then $f(x) \leq r$ by condition (iii) of the definition of local feature size. Consider second the case where the splitting of ab is triggered by rejecting the addition of a circumcenter. Let p be this circumcenter and let r' be the radius of its circle. Since p lies inside the diameter circle of ab we have $r' \leq \sqrt{2}r$. Using the Lipschitz condition and the argument for (B) we get

$$\begin{aligned} f(x) &\leq f(p) + r \\ &\leq r' \cdot C_2 + r \\ &\leq \sqrt{2}r \cdot C_2 + r. \end{aligned}$$

Using $C_1 \geq 1 + \sqrt{2}C_2$ we get

$$r \;\geq\; \frac{f(x)}{1 + \sqrt{2}C_2} \;\geq\; \frac{f(x)}{C_1},$$

as required. □

Analysis. Invariant 9 guarantees that vertices added to the triangulation cannot get arbitrarily close to preceding vertices. We show that this implies that they cannot get close to succeeding vertices either.

LEMMA 10. $\|a - b\| \geq \frac{f(a)}{1+C_1}$ for all vertices a, b.

PROOF. If b precedes a then

$$\|a - b\| \;\geq\; \frac{f(a)}{C_1} \;\geq\; \frac{f(a)}{1 + C_1}.$$

Otherwise, we have $\|b - a\| \geq f(b)/C_1$ and therefore

$$f(a) \;\leq\; f(b) + \|a - b\| \;\leq\; \|a - b\| \cdot (1 + C_1).$$

□

A simple packing argument implies that the algorithm halts after adding a finite number of vertices. Let K be the final triangulation. We relate the number of triangles to the integral of $1/f^2(x)$. Recall that B is the bounding box used in the construction of K.

LEMMA 11. The number of vertices in K is at most some constant times

$$\int_B \frac{1}{f^2(x)} dx.$$

PROOF. For each vertex a of K let D_a be the disk with center a and radius $r_a = f(a)/(2 + 2C_1)$. By Lemma 10, the disks are pairwise disjoint. At least one quarter of each disk lies inside B. Therefore

$$\begin{aligned} \int_B \frac{dx}{f^2(x)} &\geq \frac{1}{4} \cdot \sum_a \int_{D_a} \frac{dx}{f^2(x)} \\ &\geq \frac{1}{4} \cdot \sum_a \frac{r_a^2 \pi}{(f(a) + r_a)^2} \\ &\geq \frac{1}{4} \cdot \sum_a \frac{\pi}{(3 + 2C_1)^2}, \end{aligned}$$

because $(f(a) + r_a)^2 / r_a^2 \leq 1 + (f(a)/r_a)^2 = 3 + 2C_1$. This is a constant times the number of vertices. □

It can also be shown that every triangulation solving the mesh generation problem for G has at least some constant times $\int dx/f^2(x)$ triangles. This finally shows that the size of K is at most some constant times the minimum.

Bibliographic notes. The algorithm in these notes is due to Jim Ruppert. It achieves best results if the skinny triangles are removed in order of non-decreasing smallest angle. The details left out of the journal publication [4] can be found in the technical report [3]. Bern, Eppstein and Gilbert [1] show that the same technical result (constant minimum angle and proportional to minimum size) can also be achieved using quad trees. However, the constants are considerably worse than the ones obtained using Delaunay refinement. Another predecessor of Ruppert's algorithm is the Delaunay refinement algorithm by Paul Chew [2] that achieves a constant minimum angle but with uniform density over the entire region. What distinguishes Ruppert's from Chew's algorithm is the idea of local feature size and that it relates to the density of the triangulation.

[1] M. BERN, D. EPPSTEIN AND J. GILBERT. Provably good mesh generation. *J. Comput. Syst. Sci.* **48** (1994), 384–409.

[2] P. CHEW. Guaranteed-quality triangular meshes. Report TR-98-983, Comput. Sci. Dept., Cornell Univ., Ithaca, New York, 1989.

[3] J. RUPPERT. A new and simple algorithm for quality 2-dimensional mesh generation. Report UCB/CSD 92/694, Comput. Sci. Div., Univ. California, Berkeley, California, 1992.

[4] J. RUPPERT. A Delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms* **18** (1995), 548–585.