1

Delaunay triangulations

The four sections in this chapter focus on Delaunay triangulations for finite point sets in the plane. Section 1.1 introduces the Delaunay triangulation as the dual of the Voronoi diagram. Section 1.2 describes an algorithm that constructs the Delaunay triangulation as a sequence of edge flips. Although the running time of the algorithm is not the best possible, the fact that it halts and is correct allows us to deduce nontrivial structural properties about Delaunay triangulations in the plane. Section 1.3 gives an incremental algorithm whose randomized running time is the best possible. The implementation of a geometric algorithm is generally a challenging task, and the algorithms in Sections 1.2 and 1.3 are no exceptions. Section 1.4 discusses the use of exact arithmetic and symbolic perturbation to implement the numerical aspects with algebraic tools.

1.1 Voronoi and Delaunay

This section introduces Delaunay triangulations as duals of Voronoi diagrams. It discusses the role of general position in the definition and explains some of the basic properties of Delaunay triangulations.

Voronoi diagrams

Given a finite set of points in the plane, the idea is to assign to each point a region of influence in such a way that the regions decompose the plane. To describe a specific way to do that, let $S \subseteq \mathbb{R}^2$ be a set of n points and define the *Voronoi region* of $p \in S$ as the set of points $x \in \mathbb{R}^2$ that are at least as close to p as to any other point in S; that is,

$$V_p = \{x \in \mathbb{R}^2 \mid \|x - p\| \leq \|x - q\|, \ \forall q \in S\}.$$

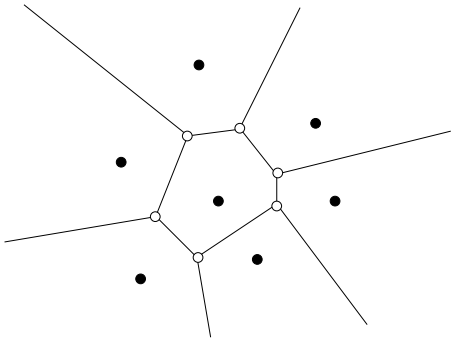

Figure 1.1. Seven points define the same number of Voronoi regions. One of the regions is bounded because the defining point is completely surrounded by the others.

This definition is illustrated in Figure 1.1. Consider the half-plane of points at least as close to p as to q: $H_{pq} = \{x \in \mathbb{R}^2 \mid \|x - p\| \leq \|x - q\|\}$. The Voronoi region of p is the intersection of half-planes H_{pq}, for all $q \in S - \{p\}$. It follows that V_p is a convex polygonal region, possibly unbounded, with at most $n - 1$ edges.

Each point $x \in \mathbb{R}^2$ has at least one nearest point in S, so it lies in at least one Voronoi region. It follows that the Voronoi regions cover the entire plane. Two Voronoi regions lie on opposite sides of the perpendicular bisector separating the two generating points. It follows that Voronoi regions do not share interior points, and if a point x belongs to two Voronoi regions, then it lies on the bisector of the two generators. The Voronoi regions together with their shared edges and vertices form the *Voronoi diagram* of S.

Delaunay triangulation

We get a dual diagram if we draw a straight *Delaunay edge* connecting points $p, q \in S$ if and only if their Voronoi regions intersect along a common line segment; see Figure 1.2. In general, the Delaunay edges decompose the convex hull of S into triangular regions, which are referred to as *Delaunay triangles*.

To count the Delaunay edges we use some results on *planar graphs*, defined by the property that their edges can be drawn in the plane without crossing. It is true that no two Delaunay edges cross each other, but to avoid an argument, we draw each Delaunay edge from one endpoint straight to the midpoint of the shared Voronoi edge and then straight to the other endpoint. Now it is obvious that no two of these edges cross. With the use of Euler's relation, it can be shown that a planar graph with $n \geq 3$ vertices has at most $3n - 6$ edges and at most $2n - 4$ faces. The same bounds hold for the number of Delaunay edges

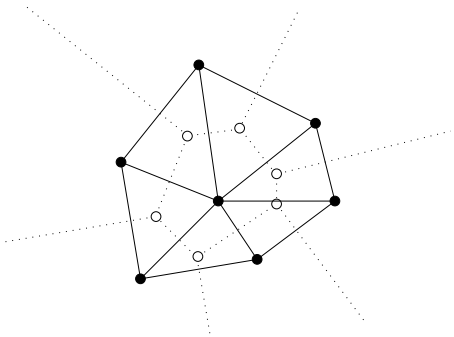

Figure 1.2. The Voronoi edges are dotted and the dual Delaunay edges are solid.

and triangles. There is a bijection between the Voronoi edges and the Delaunay edges, so $3n - 6$ is also an upper bound on the number of Voronoi edges. Similarly, $2n - 4$ is an upper bound on the number of Voronoi vertices.

Degeneracy

There is an ambiguity in the definition of Delaunay triangulation if four or more Voronoi regions meet at a common point u. One such case is shown in Figure 1.3. The points generating the four or more regions all have the same distance from u: they lie on a common circle around u. Probabilistically, the chance of picking even just four points on a circle is zero because the circle defined by the first three points has zero measure in $\mathbb{R}^2$. A common way to say the same thing is that four points on a common circle form a *degeneracy* or a *special case*. An arbitrarily small perturbation suffices to remove the degeneracy and to reduce the special case to the general case.

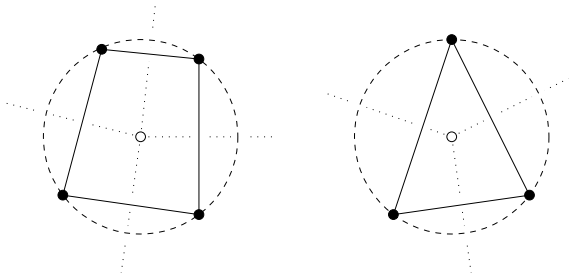

Figure 1.3. To the left, four dotted Voronoi edges meet at a common vertex and the dual Delaunay edges bound a quadrilateral. To the right, we have the general case, where only three Voronoi edges meet at a common vertex and the Delaunay edges bound a triangle.

We will often assume *general position*, which is the absence of any degeneracy. This really means that we delay the treatment of degenerate cases to later. The treatment is eventually done by perturbation, which can be actual or conceptual, or by exhaustive case analysis.

Circles and power

For now we assume general position. For a Delaunay triangle, abc, consider the circumcircle, which is the unique circle passing through a, b, and c. Its center is the corresponding Voronoi vertex, $u = V_a \cap V_b \cap V_c$, and its radius is $\varrho = \|u - a\| = \|u - b\| = \|u - c\|$; see Figure 1.3. We call the circle *empty* because it encloses no point of S. It turns out that empty circles characterize Delaunay triangles.

Circumcircle Claim. Let $S \subset \mathbb{R}^2$ be finite and in general position, and let $a, b, c \in S$ be three points. Then abc is a Delaunay triangle if and only if the circumcircle of abc is empty.

It is not entirely straightforward to see that this is true, at least not at the moment. Instead of proving the Circumcircle Claim, we focus our attention on a new concept of distance from a circle. The *power* of a point $x \in \mathbb{R}^2$ from a circle U with center u and radius ϱ is

$$\pi_U(x) = \|x - u\|^2 - \varrho^2.$$

If x lies outside the circle, then $\pi_U(x)$ is the square length of a tangent line segment connecting x with U. In any case, the power is positive outside the circle, zero on the circle, and negative inside the circle. We sometimes think of a circle as a weighted point and of the power as a weighted distance to that point. Given two circles, the set of points with equal power from both is a line. Figure 1.4 illustrates three different arrangements of two circles and their bisectors of points with equal power from both.

Acyclicity

We use the notion of power to prove an acyclicity result for Delaunay triangles. Let $x \in \mathbb{R}^2$ be an arbitrary but fixed viewpoint. We say a triangle abc *lies in front of* another triangle def if there is a half-line starting at x that first passes through abc and then through def; see Figure 1.6. We write $abc \prec def$ if abc lies in front of def. The set of Delaunay triangles together with $\prec$ forms a relation.

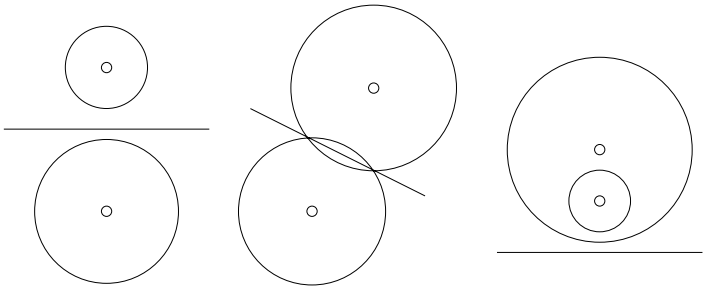

Figure 1.4. Three times two circles with bisector. From left to right: two disjoint and nonnested circles; two intersecting circles; and two nested circles.

General relations have cycles, which are sequences $\tau_0 \prec \tau_1 \prec \cdots \prec \tau_k \prec \tau_0$. Such cycles can also occur in general triangulations, as illustrated in Figure 1.5, but they cannot occur if the triangles are defined by empty circumcircles.

Acyclicity Lemma. The in-front relation for the set of Delaunay triangles defined by a finite set $S \subseteq \mathbb{R}^2$ is acyclic.

Proof. We show that $abc \prec def$ implies that the power of x from the circumcircle of abc is less than the power from the circumcircle of def. Define $abc = \tau_0$ and write $\pi_0(x)$ for the power of x from the circumcircle of abc. Similarly define $def = \tau_k$ and $\pi_k(x)$. Because S is finite, we can choose a half-line that starts at x, passes through abc and def, and contains no point of S. It intersects a sequence of Delaunay triangles:

$$abc = \tau_0 \prec \tau_1 \prec \cdots \prec \tau_k = def.$$

For any two consecutive triangles, the bisector of the two circumcircles contains the common edge. Because the third point of τ_{i+1} lies outside the circumcircle of τ_i, we have $\pi_i(x) < \pi_{i+1}(x)$ for $0 \leq i \leq k-1$. Hence $\pi_0(x) < \pi_k(x)$. The

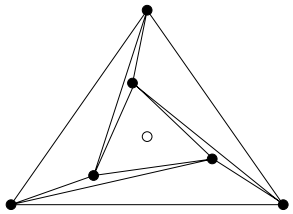

Figure 1.5. From the viewpoint in the middle, the three skinny triangles form a cycle in the in-front relation.

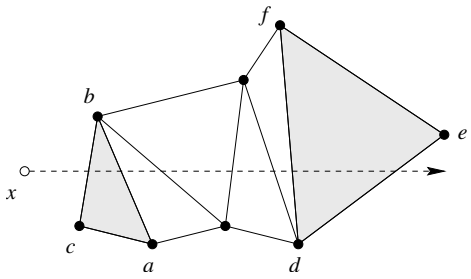

Figure 1.6. Triangle abc lies in front of triangle def. If abc and def belong to a Delaunay triangulation, then there is a sequence of triangles between them that all intersect the half-line.

acyclicity of the relation follows because real numbers cannot increase along a cycle. □

Bibliographic notes

Voronoi diagrams are named after the Ukrainian mathematician Georges Voronoi, who published two seminal papers at the beginning of the twentieth century [5]. The same concept was discussed about half a century earlier by P. G. L. Dirichlet, and there are unpublished notes by René Descartes suggesting that he was using Voronoi diagrams in the first half of the seventeenth century. Delaunay triangulations are named after the Russian mathematician Boris Delaunay (also Delone), who dedicated his paper on empty spheres [2] to Georges Voronoi. The article by Franz Aurenhammer [1] offers a nice survey of Voronoi diagrams and their algorithmic applications. The acyclicity of Delaunay triangulations in arbitrary dimensions was proved by Edelsbrunner [3] and subsequently applied in computer graphics. In particular, the three-dimensional case has been exploited for the visualization of diffuse volumes [4, 6].

[1] F. Aurenhammer. Voronoi diagrams – a study of a fundamental geometric data structure. *ACM Comput. Surveys* **23** (1991), 345–405.
[2] B. Delaunay. Sur la sphère vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* **7** (1934), 793–800.
[3] H. Edelsbrunner. An acyclicity theorem for cell complexes in d dimensions. *Combinatorica* **10** (1990), 251–260.
[4] N. Max, P. Hanrahan, and R. Crawfis. Area and volume coherence for efficient visualization of 3D scalar functions. *Comput. Graphics* **24** (1990), 27–33.
[5] G. Voronoi. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. *J. Reine Angew. Math.* **133** (1907), 97–178, and **134** (1908), 198–287.
[6] P. L. Williams. Visibility ordering meshed polyhedra. *ACM Trans. Graphics* **11** (1992), 103–126.

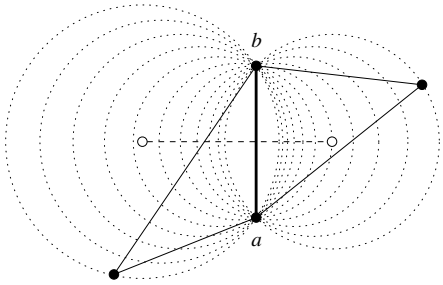

Figure 1.7. The Voronoi edge is the dashed line segment of centers of circles passing through the endpoints of ab.

1.2 Edge flipping

This section introduces a local condition for edges, shows it implies that a triangulation is Delaunay, and derives an algorithm based on edge flipping. The correctness of the algorithm implies that, among all triangulations of a given point set, the Delaunay triangulation maximizes the smallest angle.

Empty circles

Recall the Circumcircle Claim, which says that three points $a, b, c \in S$ are vertices of a Delaunay triangle if and only if the circle that passes through a, b, c is empty. A Delaunay edge, ab, belongs to one or two Delaunay triangles. In either case, there is a pencil of empty circles passing through a and b. The centers of these circles are the points on the Voronoi edge $V_a \cap V_b$; see Figure 1.7. What the Circumcircle Claim is for triangles, the Supporting Circle Claim is for edges.

Supporting Circle Claim. Let $S \subseteq \mathbb{R}^2$ be finite and in general position and let $a, b \in S$. Then ab is a Delaunay edge if and only if there is an empty circle that passes through a and b.

Delaunay lemma

By a *triangulation* we mean a collection of triangles together with their edges and vertices. A triangulation K *triangulates* S if the triangles decompose the convex hull of S and the set of vertices is S. An edge $ab \in K$ is *locally Delaunay* if

(i) it belongs to only one triangle and therefore bounds the convex hull, or
(ii) it belongs to two triangles, abc and abd, and d lies outside the circumcircle of abc.

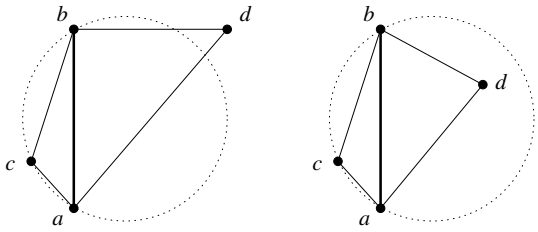

Figure 1.8. To the left ab is locally Delaunay and to the right it is not.

The definition is illustrated in Figure 1.8. A locally Delaunay edge is not necessarily an edge of the Delaunay triangulation, and it is fairly easy to construct such an example. However, if *every* edge is locally Delaunay, then we can show that *all* are Delaunay edges.

Delaunay Lemma. If every edge of K is locally Delaunay, then K is the Delaunay triangulation of S.

Proof. Consider a triangle $abc \in K$ and a vertex $p \in K$ different from a, b, c. We show that p lies outside the circumcircle of abc. Because this is then true for every p, the circumcircle of abc is empty, and because this is then true for every triangle abc, K is the Delaunay triangulation of S. Choose a point x inside abc such that the line segment from x to p contains no vertex other than p. Let $abc = \tau_0, \tau_1, \ldots, \tau_k$ be the sequence of triangles that intersect xp, as in Figure 1.9. We write $\pi_i(p)$ for the power of p to the circumcircle of τ_i, as before. Since the edges along xp are all locally Delaunay, we have $\pi_0(p) > \pi_1(p) > \cdots > \pi_k(p)$. Since p is one of the vertices of the last triangle, we have $\pi_k(p) = 0$. Therefore $\pi_0(p) > 0$, which is equivalent to p's lying outside the circumcircle of abc. □

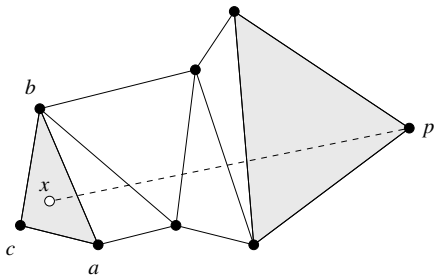

Figure 1.9. Sequence of triangles in K that intersect xp.

Edge-flip algorithm

If ab belongs to two triangles, abc and abd, whose union is a convex quadrangle, then we can *flip* ab to cd. Formally, this means we remove ab, abc, abd from the triangulation and we add cd, acd, bcd to the triangulation, as in Figure 1.10. The picture of a flip looks like a tetrahedron with the front and back superimposed. We can use edge flips as elementary operations to convert an arbitrary triangulation K to the Delaunay triangulation. The algorithm uses a stack and maintains the invariant that unless an edge is locally Delaunay, it resides on the stack. To avoid duplicates, we mark edges stored on the stack. Initially, all edges are marked and pushed on the stack.

```
while stack is non-empty do
  pop ab from stack and unmark it;
  if ab not locally Delaunay then
    flip ab to cd;
    for xy ∈ {ac, cb, bd, da} do
      if xy not marked then
        mark xy and push it on stack
      endif
    endfor
  endif
endwhile.
```

Let n be the number of points. The amount of memory used by the algorithm is $O(n)$ because there are at most $3n - 6$ edges, and the stack contains at most one copy of each edge. At the time the algorithm terminates, every edge is locally Delaunay. By the Delaunay Lemma, the triangulation is therefore the Delaunay triangulation of the point set.

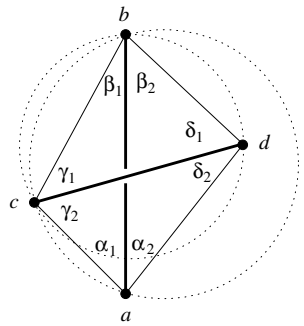

Figure 1.10. Flipping ab to cd. If ab is not locally Delaunay, then the union of the two triangles is convex and cd is locally Delaunay.

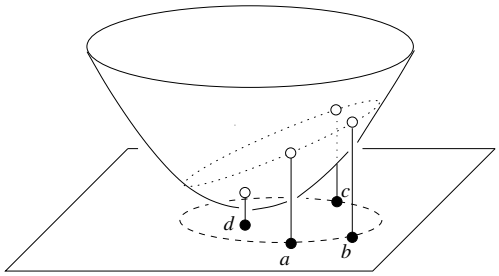

Figure 1.11. Points a, b, c lie on the dashed circle in the x_1x_2-plane and d lies inside that circle. The dotted curve is the intersection of the paraboloid with the plane that passes through $\hat{a}$, $\hat{b}$, $\hat{c}$. It is an ellipse whose projection is the dashed circle.

Circle and plane

Before proving the algorithm terminates, we interpret a flip as a tetrahedron in three-dimensional space. Let $\hat{a}$, $\hat{b}$, $\hat{c}$, $\hat{d}$ be the vertical projections of points a, b, c, d in the x_1x_2-plane onto the paraboloid defined as the graph of Π: $x_3 = x_1^2 + x_2^2$; see Figure 1.11.

Lifted Circle Claim. Point d lies inside the circumcircle of abc if and only if point $\hat{d}$ lies vertically below the plane passing through $\hat{a}$, $\hat{b}$, $\hat{c}$.

Proof. Let U be the circumcircle of abc and H the plane passing through $\hat{a}$, $\hat{b}$, $\hat{c}$. We first show that U is the vertical projection of $H \cap \text{gf}\,\Pi$. Transform the entire space by mapping every point (x_1, x_2, x_3) to $(x_1, x_2, x_3 - x_1^2 - x_2^2)$. Points $\hat{a}$, $\hat{b}$, $\hat{c}$, $\hat{d}$ are mapped back to a, b, c, d, and the paraboloid Π becomes the x_1x_2-plane. The plane H becomes a paraboloid that passes through a, b, c. It intersects the x_1x_2-plane in the circumcircle of abc. Plane H partitions gf Π into a patch below H, a curve in H, and a patch above H. The curve in H is projected onto the circumcircle of abc, and the patch below H is projected onto the open disk inside the circle. It follows that $\hat{d}$ belongs to the patch below H if and only if d lies inside the circumcircle of abc. □

Running time

Flipping ab to cd is like gluing the tetrahedron $\hat{a}\hat{b}\hat{c}\hat{d}$ from below to $\hat{a}\hat{b}\hat{c}$ and $\hat{a}\hat{b}\hat{d}$. The algorithm can be understood as gluing a sequence of tetrahedra. Once we glue $\hat{a}\hat{b}\hat{c}\hat{d}$, we cannot glue another tetrahedron right below $\hat{a}\hat{b}$. In other words, once we flip ab we cannot introduce ab again by some other flip. This implies there are at most as many flips as there are edges connecting

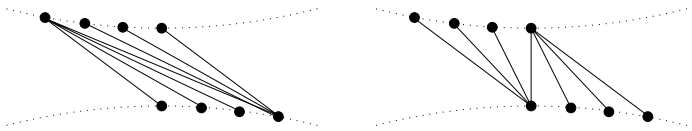

Figure 1.12. To the left we see about one third of the edges in the initial triangulation, and to the right we see the same number of edges in the final Delaunay triangulation.

n points, namely $\binom{n}{2}$. Each flip takes constant time; hence the total running time is $O(n^2)$.

There are cases in which the algorithm takes $\Theta(n^2)$ flips to change an initial triangulation to the Delaunay triangulation, and one such case is illustrated in Figure 1.12. Take a convex upper and a concave lower curve and place m points on each, such that the upper points lie to the left of the lower points. The edges connecting the two curves in the initial and the Delaunay triangulation are shown in Figure 1.12. For each point, count the positions it is away from the middle, and for each edge charge the minimum of the two numbers obtained for its endpoints. In the initial triangulation, the total charge is about m^2, and in the Delaunay triangulation, the total charge is zero. Each flip moves an endpoint by at most one position and therefore decreases the charge by at most one. A lower bound of about m^2 for the number of flips follows.

MaxMin angle property

A flip substitutes two new triangles for two old triangles. It therefore changes six of the angles. In Figure 1.10, the new angles are γ_1, δ_1, $\beta_1 + \beta_2$, γ_2, δ_2, and $\alpha_1 + \alpha_2$, and the old angles are α_1, β_1, $\gamma_1 + \gamma_2$, α_2, β_2, and $\delta_1 + \delta_2$. We claim that for each of the six new angles there is an old angle that is at least as small. Indeed, $\gamma_1 \geq \alpha_2$ because both angles are opposite the same edge, namely bd, and a lies outside the circle passing through b, c, d. Similarly, $\delta_1 \geq \alpha_1$, $\gamma_2 \geq \beta_2$, and $\delta_2 \geq \beta_1$, and for trivial reasons $\beta_1 + \beta_2 \geq \beta_1$ and $\alpha_1 + \alpha_2 \geq \alpha_1$. It follows that a flip does not decrease the smallest angle in a triangulation. Since we can go from any triangulation K of S to the Delaunay triangulation, this implies that the smallest angle in K is no larger than the smallest angle in the Delaunay triangulation.

MaxMin Angle Lemma. Among all triangulations of a finite set $S \subseteq \mathbb{R}^2$, the Delaunay triangulation maximizes the minimum angle.

Figure 1.13 illustrates the above proof of the MaxMin Angle Lemma by sketching what we call the *flip graph* of S. Each triangulation is a node, and there is a

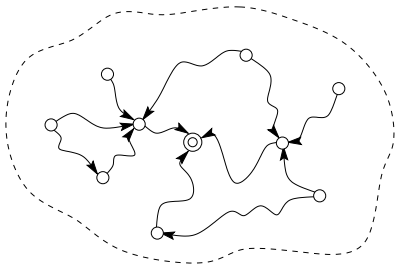

Figure 1.13. Sketch of flip graph. The sink is the Delaunay triangulation. There is a directed path from every node to the Delaunay triangulation.

directed arc from node μ to node ν if there is a flip that changes the triangulation μ to ν. The direction of the arc corresponds to our requirement that the flip substitutes a locally Delaunay edge for one that is not locally Delaunay. The running time analysis implies that the flip graph is acyclic and that its undirected version is connected. If we allow flips in either direction, we can go from any triangulation of S to any other triangulation in less than n^2 flips.

Bibliographic notes

A proof of the Delaunay Lemma and its generalization to arbitrary finite dimensions is contained in the original paper by Boris Delaunay [1]. The edge-flip algorithm is from Charles Lawson [3]. The algorithm does not generalize to three or higher dimensions. For planar triangulations, the edge-flip operation is widely used to improve local quality measures; see, for example, Schumaker [4]. Unfortunately, the algorithms get caught in local optima for almost all interesting measures. The observation that the Delaunay triangulation maximizes the smallest angle was first made by Robin Sibson [5]. Minimizing the largest angle seems more difficult, and the only known polynomial time algorithm uses edge insertions, which are somewhat more powerful than edge flips [2].

[1] B. Delaunay. Sur la sphère vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk* **7** (1934), 793–800.

[2] H. Edelsbrunner, T. S. Tan, and R. Waupotitsch. An $O(n^2 \log n)$ time algorithm for the minmax angle triangulation. *SIAM J. Sci. Stat. Comput.* **13** (1992), 994–1008.

[3] C. L. Lawson. Software for C^1 surface interpolation. In *Mathematical Software III*, Academic Press, New York, 1977, 161–194.

[4] L. L. Schumaker. Triangulation methods. In *Topics in Multivariate Approximation*, C. K. Choi, L. L. Schumaker, and F. I. Utreras (eds.), Academic Press, New York, 1987, 219–232.

[5] R. Sibson. Locally equiangular triangulations. *Comput. J.* **21** (1978), 243–245.

1.3 Randomized construction

The algorithm in this section constructs Delaunay triangulations incrementally, using edge flips and randomization. After explaining the algorithm, we present a detailed analysis of the expected amount of resources it requires.

Incremental algorithm

We obtain a fast algorithm for constructing Delaunay triangulations if we interleave flipping edges with adding points. Denote the points in $S \subseteq \mathbb{R}^2$ as $p_1, p_2, \ldots, p_n$ and assume general position. When we add a point to the triangulation, it can either lie inside or outside the convex hull of the preceding points. To reduce the outside case to the inside case, we start with a triangulation D_0 that consists of a single and sufficiently large triangle xyz. Define $S_i = \{x, y, z, p_1, p_2, \ldots, p_i\}$, and let D_i be the Delaunay triangulation of S_i. The algorithm is a `for`-loop adding the points in sequence. After adding a point, it uses edge flips to satisfy the Delaunay Lemma before the next point is added.

```
for i = 1 to n do
  find τ_{i−1} ∈ D_{i−1} containing p_i;
  add p_i by splitting τ_{i−1} into three;
  while ∃ab not locally Delaunay do
    flip ab to other diagonal cd
  endwhile
endfor.
```

The two elementary operations used by the algorithm are shown in Figure 1.14. Both pictures can be interpreted as the projection of a tetrahedron, though from different angles. For this reason, the addition of a point inside a triangle is sometimes called a 1-to-3 flip, while an edge flip is sometimes also called a 2-to-2 flip.

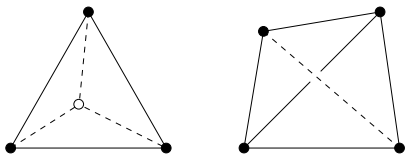

Figure 1.14. To the left, the hollow vertex splits the triangle into three. To the right, the dashed diagonal replaces the solid diagonal.

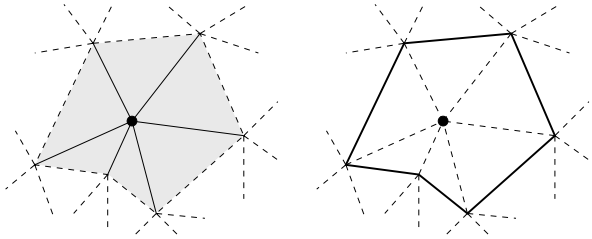

Figure 1.15. The star of the solid vertex is shown on the left and the link of the same vertex is shown on the right.

Growing star

Note that every new triangle in D_i has p_i as one of its vertices. Indeed, abc is a triangle in D_i if and only if $a, b, c \in S_i$ and the circumcircle is empty of points in S_i. But if p_i is not one of the vertices, then $a, b, c \in S_{i-1}$, and if the circumcircle is empty of points in S_i, then it is also empty of points in S_{i-1}. So abc is also a triangle in D_{i-1}. This implies that all flips during the insertion of p_i occur right around p_i.

We need some definitions. The *star* of p_i consists of all triangles that contain p_i. The *link* of p_i consists of all edges of triangles in the star that are disjoint from p_i. Both concepts are illustrated in Figure 1.15. Right after p_i is added, the link consists of three edges, namely the edges of the triangle that contains p_i. These edges are marked and pushed on the stack to start the edge-flipping `while`-loop. Each flip replaces a link edge by an edge with endpoint p_i. At the same time, it removes one triangle in the star and one outside the star and it adds the two triangles that cover the same quadrangle to the star. The net effect is one more triangle in the star. The number of edge flips is therefore three less than the number of edges in the final link, which is the same as three less than the degree of p_i in D_i.

Number of flips

We temporarily ignore the time needed to find the triangles τ_{i-1}. The rest of the time is proportional to the number of flips needed to add p_1 to p_n. We assume $p_1, p_2, \ldots, p_n$ is a randomly chosen input sequence. Random does not mean arbitrary but rather that every permutation of n points is equally likely. The expected number of flips is the total number of flips needed to construct the Delaunay triangulation for all $n!$ input permutations divided by $n!$.

Consider inserting the last point, p_n. The sum of degrees of all possible last points is the same as the sum of degrees of all points p_i in D_n. The latter is

equal to twice the number of edges and therefore

$$\sum_{i=1}^{n} \deg p_i \leq 6n.$$

The number of flips needed to add all last points is therefore at most $6n - 3n = 3n$. Each last point is added $(n-1)!$ times. The total number of flips is therefore

$$\begin{aligned} F(n) &\leq n \cdot F(n-1) + 3n! \\ &\leq 3n \cdot n!. \end{aligned}$$

Indeed, if we assume $F(n-1) \leq 3(n-1) \cdot (n-1)!$, we get $n \cdot F(n-1) + 3n! \leq 3(n-1) \cdot n! + 3n! = 3n \cdot n!$. The expected number of edge flips needed for n points is therefore at most $3n$.

There is a simple way to say the same thing. The expected number of flips for the last point is at most three, and therefore the expected number of flips to add any point is at most three.

The history

We use the evolution of the Delaunay triangulation to find the triangle τ_{i-1} that contains point p_i. Instead of deleting a triangle when it is split or flipped away, we just make it the parent of the new triangles. Figure 1.16 shows the two operations to the left and the corresponding parent–child relations to the right. Each time we split or flip, we add triangles or nodes to the growing data

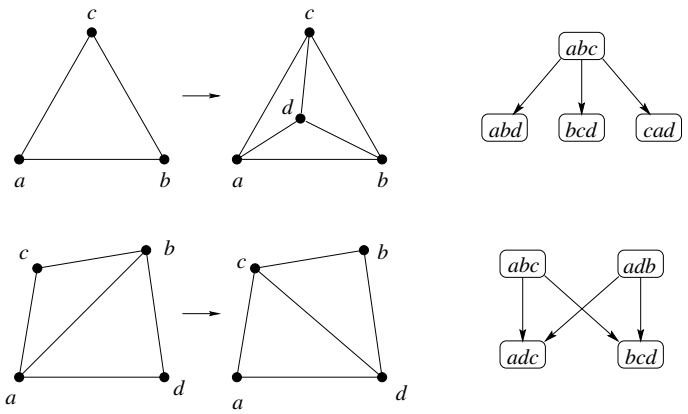

Figure 1.16. Splitting a triangle generates a parent with three children. Flipping an edge generates two parents sharing the same two children.

structure that records the history of the construction. The evolution from D_0 to D_n consists of n splits and an expected number of at most $3n$ flips. The resulting directed acyclic graph, or DAG for short, therefore has an expected size of at most $1 + 3n + 2 \cdot 3n = 9n + 1$ nodes. It has a unique source, the triangle xyz, and its sinks are the triangles in D_n.

Searching and charging

Consider adding the point p_i. To find the triangle $\tau_{i-1} \in D_{i-1}$, we search a path of triangles in the history DAG that all contain p_i. The path begins as xyz and ends at τ_{i-1}. The history DAG of D_{i-1} consists of i layers. Layers 0 to j represent the DAG of D_j. Its sinks are the triangles in D_j, and we let $\sigma_j \in D_j$ be the triangle that contains p_i. Triangles σ_0 to σ_j form a not necessarily contiguous subsequence of nodes along the search path. It is quite possible that some of the triangles σ are the same. Let G_j be the set of triangles removed from D_j during the insertion of p_{j+1}, and let H_j be the set of triangles removed from D_j during the hypothetical and independent insertion of p_i into D_j. The two sets are schematically sketched as intervals along the real line representing the Delaunay triangulation in Figure 1.17. We have $\sigma_j = \sigma_{j+1}$ if G_j and H_j are disjoint. Suppose $\sigma_j \neq \sigma_{j+1}$. Then $X_j = G_j \cap H_j \neq \emptyset$, and all triangles on the portion of the path from σ_j to σ_{j+1} are generated by flips that remove triangles in X_j. The cost for searching with p_i is therefore at most proportional to the sum of card X_j, for j from 0 to $i - 2$.

We write X_j in terms of other sets. These sets represent what happens if we again hypothetically first insert p_i into D_j and then insert p_{j+1} into the Delaunay triangulation of $S_j \cup \{p_i\}$. Let Y_j be the set of triangles removed during the insertion of p_{j+1}, and let $Z_j \subseteq Y_j$ be the subset of triangles that do not belong to D_j. Each triangle in Z_j is created during the insertion of p_i, so p_i must be one of its vertices. We have

$$X_j = G_j - (Y_j - Z_j).$$

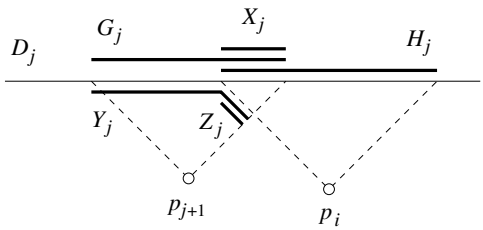

Figure 1.17. The intervals represent sets of triangles removed or added when we insert p_{j+1} and/or p_i to D_j.

Expectations

We bound the expected search time by bounding the expected total size of the X_j. Write cardinalities by using corresponding lower-case letters. Because $Z_j \subseteq Y_j$ and $Y_j - Z_j \subseteq G_j$, we have

$$x_j = g_j - y_j + z_j.$$

The expected values of g_j and y_{j-1} are the same, because both count triangles removed by inserting a random jth point. Because the expectation of a sum is the sum of expectations, we have

$$\begin{aligned} E\left[\sum_{j=0}^{i-2} x_j\right] &= \sum_{j=0}^{i-2} E[g_j] - E[y_j] + E[z_j] \\ &= E[g_0 - g_{i-1}] + \sum_{j=0}^{i-2} E[z_j]. \end{aligned}$$

To compute the expected value of z_j, we use the fact that among $j+2$ points, every pair is equally likely to be p_{j+1} and p_i. For example, if p_{j+1} and p_i are not connected by an edge in the Delaunay triangulation of $S_j \cup \{p_{j+1}, p_i\}$, then $Z_j = \emptyset$. In general, a triangle in the Delaunay triangulation of $S_j \cup \{p_i\}$ has probability at most $3/(j+1)$ of being in the star of p_i. The expected number of triangles removed by inserting p_{j+1} is at most four. Because the expectation of a product is the product of expectations, we have $E[z_j] \leq (4 \cdot 3)/(j+1)$. The expected length of the search path for p_i is

$$\sum_{j=0}^{i-2} E[x_j] \leq \sum_{j=0}^{i-2} \frac{12}{j+1} \leq 1 + 12 \ln(i-1).$$

The expected total time spent on searching in the history DAG is $\sum_i \sum_j E[x_j] \leq c \cdot n \log n$.

To summarize, the randomized incremental algorithm constructs the Delaunay triangulation of n points in $\mathbb{R}^2$ in the expected time $O(n \log n)$ and with the expected amount of memory $O(n)$.

Bibliographic notes

The randomized incremental algorithm of this section is from Guibas, Knuth, and Sharir [3]. It has been generalized to three and higher dimensions by Edelsbrunner and Shah [2]. All this is based on earlier work on randomized algorithms and in particular on the methods developed by Clarkson and Shor [1]. The arguments used to bound the expected number of flips and the expected search time are examples of the backwards analysis introduced by Raimund Seidel [4].

[1] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry. *Discrete Comput. Geom.* **4** (1989), 387–421.
[2] H. Edelsbrunner and N. R. Shah. Incremental topological flipping works for regular triangulations. *Algorithmica* **15** (1996), 223–241.
[3] L. J. Guibas, D. E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica* **7** (1992), 381–413.
[4] R. Seidel. Backwards analysis of randomized geometric algorithms. In *New Trends in Discrete and Computational Geometry*, J. Pach (ed.), Springer-Verlag, Berlin, 1993, 37–67.

1.4 Symbolic perturbation

The computational technique of symbolically perturbing a geometric input justifies the mathematically convenient assumption of general position. This section describes a particular perturbation known as SoS, or Simulation of Simplicity.

Orientation test

Let $a = (\alpha_1, \alpha_2)$, $b = (\beta_1, \beta_2)$, and $c = (\gamma_1, \gamma_2)$ be three points in the plane. We consider a, b, c degenerate if they lie on a common line. This includes the case in which two or all three points are the same. In the degenerate case, point c is an affine combination of a and b; that is, $c = \lambda_1 a + \lambda_2 b$ with $\lambda_1 + \lambda_2 = 1$. Such λ_1, λ_2 exist if and only if the determinant of

$$\Delta = \begin{bmatrix} 1 & \alpha_1 & \alpha_2 \\ 1 & \beta_1 & \beta_2 \\ 1 & \gamma_1 & \gamma_2 \end{bmatrix}$$

vanishes. In the nondegenerate case, the sequence a, b, c either forms a left or a right turn. We can again use the determinant of Δ to decide which it is.

Orientation Claim. The sequence a, b, c forms a left turn if and only if $\det \Delta > 0$, and it forms a right turn if and only if $\det \Delta < 0$.

Proof. We first check the claim for $a_0 = (0, 0)$, $b_0 = (1, 0)$, and $c_0 = (0, 1)$. It is geometrically obvious that a_0, b_0, c_0 form a left turn, and indeed

$$\det \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 1.$$

We can continuously move a_0, b_0, c_0 to any other left-turn a, b, c without ever having three collinear points. Since the determinant changes continuously with the coordinates, it remains positive during the entire motion and is therefore positive at a, b, c. Symmetry implies that all right turns have negative determinants. □

In-circle test

The in-circle test is formulated for four points a, b, c, d in the plane. We consider a, b, c, d degenerate if a, b, c lie on a common line or a, b, c, d lie on a common circle. We already know how to test for points on a common line. To test for points on a common circle, we recall the definition of lifted points, $\hat{a} = (\alpha_1, \alpha_2, \alpha_3)$ with $\alpha_3 = \alpha_1^2 + \alpha_2^2$, and so on. Points a, b, c, d lie on a common circle if and only if $\hat{a}, \hat{b}, \hat{c}, \hat{d}$ lie on a common plane in $\mathbb{R}^3$; see Figure 1.11. In other words, $\hat{d}$ is an affine combination of $\hat{a}, \hat{b}, \hat{c}$, which is equivalent to

$$\Gamma = \begin{bmatrix} 1 & \alpha_1 & \alpha_2 & \alpha_3 \\ 1 & \beta_1 & \beta_2 & \beta_3 \\ 1 & \gamma_1 & \gamma_2 & \gamma_3 \\ 1 & \delta_1 & \delta_2 & \delta_3 \end{bmatrix}$$

having zero determinant. In the nondegenerate case, d either lies inside or outside the circle defined by a, b, c. We can use the determinants of Δ and Γ to decide which it is. Note that permuting a, b, c can change the sign of $\det \Gamma$ without changing the geometric configuration. Since the signs of $\det \Gamma$ and $\det \Delta$ change simultaneously, we can counteract by multiplying the two.

In-circle Claim. Point d lies inside the circle passing through a, b, c if and only if $\det \Delta \cdot \det \Gamma < 0$, and d lies outside the circle if and only if $\det \Delta \cdot \det \Gamma > 0$.

Proof. We first check the claim for $d_0 = (1/2, 1/2)$ and $a_0 = (0, 0)$, $b_0 = (1, 0)$, and $c_0 = (0, 1)$ as before. Point d_0 lies at the center and therefore inside the circle passing through a_0, b_0, c_0. The determinant of Δ is 1, and that of Γ is

$$\det \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix} = -\frac{1}{2},$$

so their product is negative. As in the proof of the Orientation Claim, we derive the general result from the special one by continuity. Specifically, every configuration a, b, c, d, where d lies inside the circle of a, b, c, can be obtained from a_0, b_0, c_0, d_0 by continuous motion, avoiding all degeneracies. The signs of the two determinants remain the same throughout the motion, and so does their product. This implies the claim for negative products, and symmetry implies the claim for positive products. □

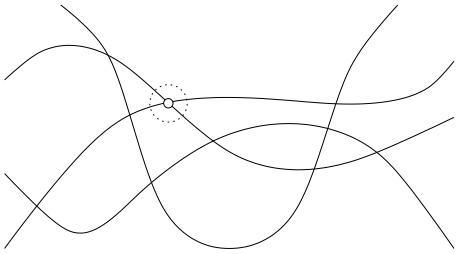

Figure 1.18. Schematic picture of the union of $(2n-1)$-dimensional manifolds in $2n$-dimensional space. The marked point lies on two manifolds and thus has two degenerate subconfigurations. The dotted circle bounds a neighbourhood, and most points in that neighbourhood are non-degenerate.

Algebraic framework

Let us now take a more abstract and algebraic view of degeneracy as a geometric phenomenon. For expository reasons, we restrict ourselves to orientation tests in the plane. Let S be a collection of n points, denoted as $p_i = (\phi_{i,1}, \phi_{i,2})$, for $1 \leq i \leq n$. By listing the $2n$ coordinates in a single sequence, we think of S as a single point in $2n$-dimensional space. Specifically, S is mapped to $Z = (\zeta_1, \zeta_2, \zeta_3, \ldots, \zeta_{2n}) \in \mathbb{R}^{2n}$, where $\zeta_{2i-1} = \phi_{i,1}$ and $\zeta_{2i} = \phi_{i,2}$, for $1 \leq i \leq n$. Point Z is degenerate if and only if

$$\det \begin{bmatrix} 1 & \zeta_{2i-1} & \zeta_{2i} \\ 1 & \zeta_{2j-1} & \zeta_{2j} \\ 1 & \zeta_{2k-1} & \zeta_{2k} \end{bmatrix} = 0$$

for some $1 \leq i < j < k \leq n$. The equation identifies a differentiable $(2n-1)$-dimensional manifold in $\mathbb{R}^{2n}$. There are $\binom{n}{3}$ such manifolds, $\mathbb{M}_\ell$, and Z is degenerate if and only if $Z \in \bigcup_\ell \mathbb{M}_\ell$, as sketched in Figure 1.18. Each manifold has dimension one less than the ambient space and hence measure zero in $\mathbb{R}^{2n}$. We have a finite union of measure zero sets, which still has measure zero. In other words, most points in an open neighborhood of $Z \in \mathbb{R}^{2n}$ are non-degenerate. A point nearby Z is often called a perturbation of Z or S. The result on neighborhoods thus implies that there are arbitrarily close nondegenerate perturbations of S.

Perturbation

We construct a nondegenerate perturbation of S by using positive parameters $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_{2n}$. These parameters will be chosen anywhere between arbitrarily and sufficiently small, and we may think of them as infinitesimals. They will

also be chosen sufficiently different, and we will see shortly what this means. Let $Z \in \mathbb{R}^{2n}$, and for every $\varepsilon > 0$ define

$$Z(\varepsilon) = (\zeta_1 + \varepsilon_1, \zeta_2 + \varepsilon_2, \ldots, \zeta_{2n} + \varepsilon_{2n}),$$

where $\varepsilon_i = f_i(\varepsilon)$ with $f_i : \mathbb{R} \to \mathbb{R}$ continuous and $f_i(0) = 0$. If the ε_i are sufficiently different, we get the following three properties provided $\varepsilon > 0$ is sufficiently small.

I. $Z(\varepsilon)$ is nondegenerate.
II. $Z(\varepsilon)$ retains all nondegenerate properties of Z.
III. The computational overhead for simulating $Z(\varepsilon)$ is negligible.

For example, if $\varepsilon_i = \varepsilon^{2^i}$ then $\varepsilon_1 \gg \varepsilon_2 \gg \cdots \gg \varepsilon_{2n}$ and we can do all computations simply by comparing indices without ever computing a feasible ε. We demonstrate this by explicitly computing the orientation of the points p_i, p_j, p_k after perturbation. By definition, that orientation is the sign of the determinant of

$$\Delta(\varepsilon) = \begin{bmatrix} 1 & \zeta_{2i-1} + \varepsilon_{2i-1} & \zeta_{2i} + \varepsilon_{2i} \\ 1 & \zeta_{2j-1} + \varepsilon_{2j-1} & \zeta_{2j} + \varepsilon_{2j} \\ 1 & \zeta_{2k-1} + \varepsilon_{2k-1} & \zeta_{2k} + \varepsilon_{2k} \end{bmatrix}.$$

Note that $\Delta(\varepsilon)$ is a polynomial in ε. The terms with smaller power are more significant than those with larger power. We assume $i < j < k$ and list the terms of $\Delta(\varepsilon)$ in the order of decreasing significance; that is,

$$\begin{aligned} \det \Delta(\varepsilon) = \det \Delta &- \det \Delta_1 \cdot \varepsilon^{2^{2i-1}} \\ &+ \det \Delta_2 \cdot \varepsilon^{2^{2i}} + \det \Delta_3 \cdot \varepsilon^{2^{2j-1}} \\ &- 1 \cdot \varepsilon^{2^{2j-1}} \varepsilon^{2^{2i}} \pm \ldots, \end{aligned}$$

where

$$\Delta = \begin{bmatrix} 1 & \zeta_{2i-1} & \zeta_{2i} \\ 1 & \zeta_{2j-1} & \zeta_{2j} \\ 1 & \zeta_{2k-1} & \zeta_{2k} \end{bmatrix},$$

$$\Delta_1 = \begin{bmatrix} 1 & \zeta_{2j} \\ 1 & \zeta_{2k} \end{bmatrix},$$

$$\Delta_2 = \begin{bmatrix} 1 & \zeta_{2j-1} \\ 1 & \zeta_{2k-1} \end{bmatrix},$$

$$\Delta_3 = \begin{bmatrix} 1 & \zeta_{2i} \\ 1 & \zeta_{2k} \end{bmatrix}.$$

Property I is satisfied because the fifth term is nonzero, and its influence on the sign of the determinant cannot be canceled by subsequent terms. Property II is satisfied because the sign of the perturbed determinant is the same as that of the unperturbed one, unless the latter vanishes.

Implementation

In order to show Property III, we give an implementation of the test for $Z(\varepsilon)$. First we sort the indices such that $i < j < k$, and we count the number of transpositions. Then we determine whether the three perturbed points form a left or a right turn by computing determinants of the four submatrices listed above.

```
boolean LEFTTURN(integer  i, j, k):
  assert i < j < k;
  case det Δ ≠ 0: return det Δ > 0;
  case det Δ1 ≠ 0: return det Δ1 < 0;
  case det Δ2 ≠ 0: return det Δ2 > 0;
  case det Δ3 ≠ 0: return det Δ3 > 0;
  otherwise: return FALSE.
```

If the number of transpositions needed to sort i, j, k is odd, then the sorting reverses the sign, and we correct the reversal by reversing the result of the Function LEFTTURN.

As an important detail, we note that signs of determinants have to be computed exactly. With normal floating point arithmetic, this is generally not possible. We must therefore resort to exact arithmetic methods using long integer or other representations of coordinates. These methods are typically more costly than floating point arithmetic, but differences vary widely among different computer hardware. A pragmatic compromise uses floating point arithmetic together with error analysis. After computing the determinant with floating point arithmetic, we check whether the absolute value is large enough for its sign to be guaranteed. Only if that guarantee cannot be obtained do we repeat the computation in exact arithmetic.

Bibliographic notes

The idea of using symbolic perturbation for computational reasons is already present in the work of George Danzig on linear programming [1]. It reappeared in computational geometry with the work of four independent groups of authors. Edelsbrunner and Mücke [2] develop SoS, which is the method described in

this section. Yap [7] studies the class of perturbations obtained with different orderings of infinitesimals. Emiris and Canny [3] introduce perturbations along straight lines. Michelucci [5] exploits randomness in the design of perturbations.

Symbolic perturbations as a general computational technique within computational geometry remains a controversial subject. It succeeds in extending partially to completely correct software for some but not all geometric problems. Seidel [6] addresses this issue, offers a unified view of symbolic perturbation, and discusses limitations of the method. Fortune and Van Wyk [4] describe a floating point filter that reduces the overhead needed for exact computation.

[1] G. B. Danzig. *Linear Programming and Extensions.* Princeton Univ. Press, Princeton, New Jersey, 1963.
[2] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. Graphics* **9** (1990), 66–104.
[3] I. Emiris and J. Canny. A general approach to removing geometric degeneracies. *SIAM J. Comput.* **24** (1995), 650–664.
[4] S. Fortune and C. J. Van Wyk. Static analysis yields efficient exact integer arithmetic for computational geometry. *ACM Trans. Graphics* **15** (1996), 223–248.
[5] D. Michelucci. An ε-arithmetic for removing degeneracies. In "Proc. IEEE Sympos. Comput. Arithmetic," 1995.
[6] R. Seidel. The nature and meaning of perturbations in geometric computing. *Discrete Comput. Geom.* **19** (1998), 1–18.
[7] C. K. Yap. Symbolic treatment of geometric degeneracies. *J. Symbolic Comput.* **10** (1990), 349–370.

Exercise collection

The credit assignment reflects a subjective assessment of difficulty. A typical question can be answered by using knowledge of the material combined with some thought and analysis.

1. **Section of triangulation** (two credits). Let K be a triangulation of a set of n points in the plane. Let ℓ be a line that avoids all points. Prove that ℓ intersects at most $2n - 4$ edges of K and that this upper bound is tight for every $n \geq 3$.
2. **Minimum spanning tree** (one credit). The notion of a minimum spanning tree can be extended from weighted graphs to a geometric setting in which the nodes are points in the plane. Take the complete graph of the set of nodes and define the length of an edge as the Euclidean distance between its endpoints. A minimum spanning tree of that graph is a *Euclidean minimum spanning tree* of the point set. Prove that all edges of every Euclidean minimum spanning tree belong to the Delaunay triangulation of the same point set.

3. **Sorted angle vector** (one credit). Let K be a triangulation of a finite set in the plane. Let t be the number of triangles and consider the sorted vector of angles,
$$\mathbf{v}(K) = (\alpha_1 \leq \alpha_2 \leq \cdots \leq \alpha_{3t}).$$
Prove that $\mathbf{v}(K) = \mathbf{v}(D)$ or $\mathbf{v}(K)$ is lexicographically smaller than $\mathbf{v}(D)$, where D is the Delaunay triangulation of the points.
4. **Minmax circumcircle** (two credits). Let K be a triangulation of a finite set in the plane and let $\varrho(K)$ be the radius of the largest circumcircle of any triangle in K. Prove $\varrho(K) \geq \varrho(D)$, where D is the Delaunay triangulation of the set.
5. **Random permutation** (one credit). Show that the following algorithm constructs a random permutation of the integers 1 to n.

```
for i = 1 to n do
  Z[i] = i; choose random index 1 ≤ j ≤ i;
  swap Z[i] and Z[j]
endfor.
```

6. **Furthest-point Voronoi diagram** (one credit). Let $S \subseteq \mathbb{R}^2$ be finite. The *furthest-point Voronoi region* of a point $p \in S$ consists of all points at least as far from p as from any other point in S,
$$F_p = \{x \in \mathbb{R}^2 \mid \|x - p\| \geq \|x - q\|, \forall q \in S\}.$$
   (i) Prove $F_p \neq \emptyset$ if and only if p lies on the boundary of the convex hull of S.
   (ii) Draw the furthest-point Voronoi regions of about ten points in the plane, together with the dual furthest-point Delaunay triangulation.
7. **Line segment intersection** (two credits). Let a, b, x, y be points in $\mathbb{R}^2$. They are in general position if no three are collinear.
   (i) Assume general position and write a boolean function that decides whether the line segments ab and xy cross or are disjoint.
   (ii) What are the degenerate cases, and how does your function deal with them?
8. **Enumerating degeneracies** (one credit). Let a, b, c, d be points in $\mathbb{R}^3$. The *orientation* of the sequence is the sign of
$$\det \begin{bmatrix} 1 & \alpha_1 & \alpha_2 & \alpha_3 \\ 1 & \beta_1 & \beta_2 & \beta_3 \\ 1 & \gamma_1 & \gamma_2 & \gamma_3 \\ 1 & \delta_1 & \delta_2 & \delta_3 \end{bmatrix}.$$

Simulation of simplicity expands the determinant into a polynomial $P(\varepsilon)$, and the orientation is decided by finding the sign of P for sufficiently small $\varepsilon > 0$.

(i) List the terms of the polynomial in the order of decreasing significance.

(ii) The perturbation classifies and disambiguates the various degenerate cases that occur. Each class corresponds to a prefix of the polynomial that is identically zero. Describe each class in words or figures.