Acta Numerica (2000), pp. 1–81

Triangulations and meshes in computational geometry

Herbert Edelsbrunner*
Department of Computer Science,
Duke University, Durham, NC 27708
and
Raindrop Geomagic, Research Triangle Park,
North Carolina, NC 27709, USA

The Delaunay triangulation of a finite point set is a central theme in computational geometry. It finds its major application in the generation of meshes used in the simulation of physical processes. This paper connects the predominantly combinatorial work in classical computational geometry with the numerical interest in mesh generation. It focuses on the two- and three-dimensional case and covers results obtained during the twentieth century.

* Research is partially supported by the Army Research Office under grant DAAG55-98-1-0177 and by the National Science Foundation under grants CCR-96-19542 and CCR-97-12088.

CONTENTS

1. Introduction

This is a paper about computational geometry and its connection to science and engineering. We argue that computational geometry draws its motivation from applications to various areas including mesh generation and that it can maintain its livelihood only if it fulfils the promise of advancing these applications in a significant manner.

History

The beginning of computational geometry as an independent intellectual discipline is usually dated around 1975, when Michael Shamos and Dan Hoey proposed algorithmic solutions for a host of basic geometric tasks (Shamos 1975, Shamos and Hoey 1975, 1976). They defined computational geometry as the study of the computational complexity of geometric problems. It is important to notice the implicit but significant shift from a continuous to a discrete conception of geometry. Application areas use geometry to model a presumably continuous reality, while computational complexity relates the finite amount of time it takes to solve a problem with the finite size in which the problem presents itself. Within a few years after its inception, computational geometry developed a strong affinity to discrete geometry as practised by combinatorialists (Erdős 1979, Pach and Agarwal 1995). This affinity was natural and helped the field to mature to a point where it is ready for a reorientation back to its continuous roots.

The intellectual development in computational geometry can be traced fairly well through the series of proceedings documenting the annual Symposium on Computational Geometry, first held in 1985. The breadth of the field is evident from the textbooks, which all take different views and explore different aspects of the field (de Berg, van Kreveld, Overmars and Schwarzkopf 1997, Edelsbrunner 1987, Mulmuley 1994, Klein 1997, Okabe, Boots and Sugihara 1992, O'Rourke 1987, 1994, Preparata and Shamos 1985). We also refer to a recent handbook, which organizes the combined field of discrete and computational geometry in 52 chapters (Goodman and O'Rourke 1997).

Outline

We illustrate the claimed function of computational geometry as a bridge between continuous and discrete methods with a focus on geometric triangulations and in particular Delaunay triangulations. Half the paper studies combinatorial properties of and algorithms for Delaunay triangulations. The other half explores questions that arise in the use of Delaunay triangulations as a representation of pieces of continuous space. To emphasize the shift in focus, we then refer to the triangulation as a mesh, which is the traditional engineering term for space decompositions used in numerical analysis (Bern and Eppstein 1992).

There is an orthogonal way of structuring this paper in two halves. Sections 2 to 8 deal with triangulations in the Euclidean plane, and Sections 9 to 16 study tetrahedrizations in three-dimensional Euclidean space.

In the predominantly discrete block consisting of Sections 2 to 5, we see a progression from geometric/structural to algorithmic considerations, and we see the same in the block consisting of Sections 9 to 12. The move towards a continuous and numerical viewpoint is pursued in the block consisting of Sections 6 to 8 and in the block consisting of Sections 13 to 16.

Style

The style of this paper is representative of the dominant style in computational geometry. Understanding is sought through formulating general claims and proving them. Similarly, algorithms are described in detail and the running time is analysed under worst-case and average assumptions. We make a conscious effort to concentrate on the two- and three-dimensional cases and with a few exceptions avoid discussions of the general d-dimensional case. While identifying properties that hold independent of the particular dimension is generally commendable, it seems counterproductive in the study of meshes whose properties vary significantly with changing dimension.

Each section is designed as a lecture in a graduate course. Whenever there is a choice, we prefer topics that have a general appeal over more specialized ones, and topics that are easy to explain over more complicated ones. Each section ends with bibliographic notes collecting references to the literature and comments on important related developments.

2. 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,

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

This definition is illustrated in Figure 1. Consider the half-plane of points at least as close to p as to q: $H_{pq} = \{x \in \mathbb{R}^2 : \|x - p\| \leq \|x - q\|\}$. 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.

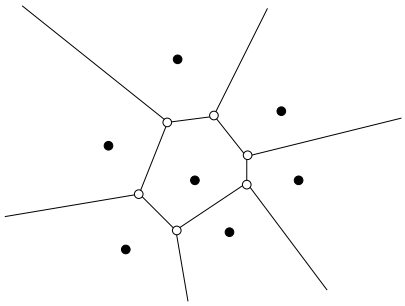

Fig. 1. Seven points define the same number of Voronoi regions. One of the regions is bounded because the defining point is completely surrounded by the others

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 2. In general, the Delaunay edges decompose the convex hull of S into triangular regions, which are referred to as *Delaunay triangles*.

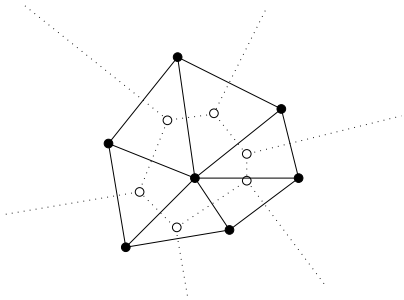

Fig. 2. The Voronoi edges are dotted and the dual Delaunay edges are solid

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 trivial that no two of these edges cross. Using 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 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 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 to the general case.

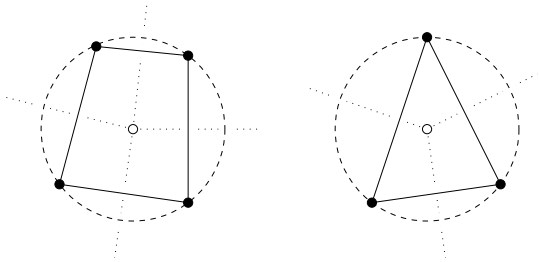

Fig. 3. To the left, four dotted Voronoi edges meet at a common vertex and the dual Delaunay edges bound a quadrilateral. To the right, we have the general case, where only three Voronoi edges meet at a common vertex and the Delaunay edges bound a triangle

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 centre 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 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 $S \subseteq \mathbb{R}^2$ 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 centre u and radius ϱ is

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

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 4 illustrates three different arrangements of two circles and their bisectors of points with equal power from both.

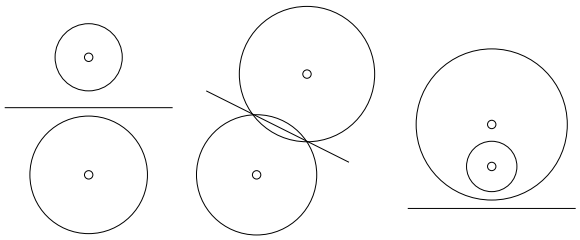

Fig. 4. Three times two circles with bisector. From left to right: two disjoint and non-nested circles, two intersecting circles, two nested circles

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 6. We write $abc \prec def$ if *abc* lies in front of *def*. The set of Delaunay triangles together with $\prec$ forms a relation. 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 5, but they cannot occur if the triangles are defined by empty circumcircles.

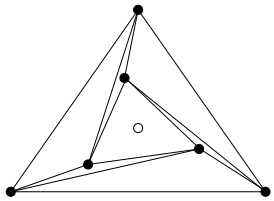

Fig. 5. From the viewpoint in the middle, the three skinny triangles form a cycle in the in-front relation

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

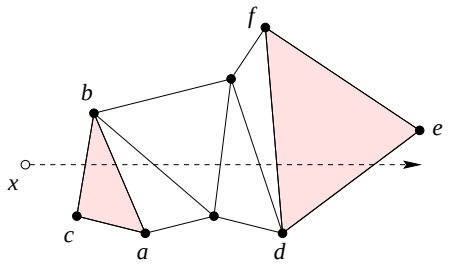

Fig. 6. Triangle abc lies in front of triangle def. If abc and def belong to a Delaunay triangulation, then there is a sequence of triangles between them that all intersect the half-line

Bibliographic notes

Voronoi diagrams are named after the Russian mathematician Georges Voronoi, who published two seminal papers at the beginning of the twentieth century (Voronoi 1907/08). 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 already using Voronoi diagrams in the first half of the seventeenth century. Delaunay triangulations are named after the Russian mathematician Boris Delaunay, who dedicated his paper on empty spheres (Delaunay 1934) to Georges Voronoi. The article by Franz Aurenhammer (1991) offers a nice survey of Voronoi diagrams and their algorithmic applications. The acyclicity of Delaunay triangulations in arbitrary dimensions was proved by Edelsbrunner (1990) and subsequently applied in computer graphics. In particular, the three-dimensional case has been exploited for the visualization of diffuse volumes (Max, Hanrahan and Crawfis 1990, Williams 1992).

3. Edge flipping

This section introduces a local condition for edges, shows it implies 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 centres of these circles are the points on the Voronoi edge $V_a \cap V_b$; see Figure 7. What the Circumcircle Claim is for triangles, the Supporting Circle Claim is for edges.

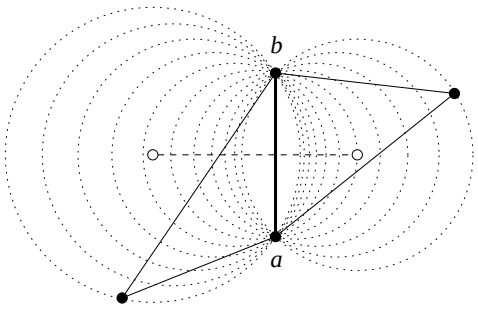

Fig. 7. The Voronoi edge is the dashed line segment of centres of circles passing through the endpoints of ab

Supporting Circle Claim. Let $S \subseteq \mathbb{R}^2$ be finite and in general position and $a, b \in S$. 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 of S, or
(ii) it belongs to two triangles, abc and abd, and d lies outside the circumcircle of abc.

The definition is illustrated in Figure 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.

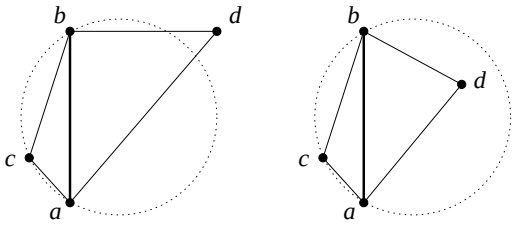

Fig. 8. To the left ab is locally Delaunay and to the right it is not

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 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 lying outside the circumcircle of abc. □

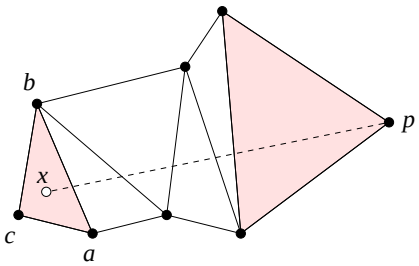

Fig. 9. Sequence of triangles in K that intersect xp

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 10. The picture of a flip looks like a tetrahedron with front and back superimposed. We can use edge flips as elementary operations to convert an

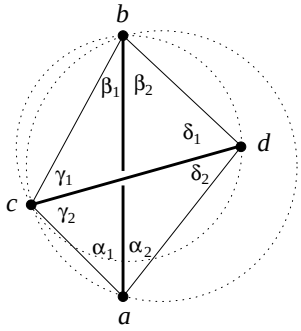

Fig. 10. Flipping ab to cd. If ab is not locally Delaunay then the union of the two triangles is convex and cd is locally Delaunay

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.

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 11.

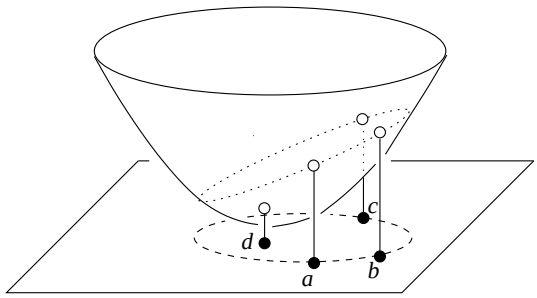

Fig. 11. Points a, b, c lie on the dashed circle in the x_1x_2-plane and d lies inside that circle. The dotted curve is the intersection of the paraboloid with the plane that passes through $\hat{a}, \hat{b}, \hat{c}$. It is an ellipse whose projection is the dashed circle

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

There are cases where the algorithm takes $\Theta(n^2)$ flips to change an initial triangulation to the Delaunay triangulation, and one such case is illustrated

in Figure 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 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.

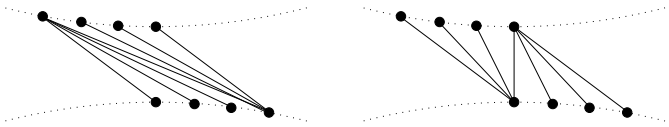

Fig. 12. To the left we see about one-third of the edges in the initial triangulation, and to the right we see the same number of edges in the final Delaunay triangulation

MaxMin Angle property

A flip substitutes two new triangles for two old triangles. It therefore changes six of the angles. In Figure 10, the new angles are $\gamma_1, \delta_1, \beta_1+\beta_2, \gamma_2, \delta_2, \alpha_1+\alpha_2$ and the old angles are $\alpha_1, \beta_1, \gamma_1+\gamma_2, \alpha_2, \beta_2, \delta_1+\delta_2$. 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, $\delta_1 \geq \alpha_1$, $\gamma_2 \geq \beta_2$, $\delta_2 \geq \beta_1$, 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 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 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.

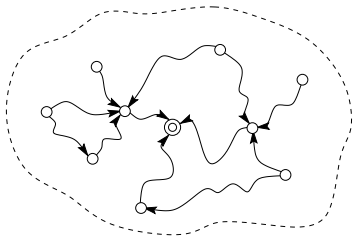

Fig. 13. Sketch of flip-graph. The sink is the Delaunay triangulation. There is a directed path from every node to the Delaunay triangulation

Bibliographic notes

A proof of the Delaunay lemma and its generalization to arbitrary finite dimensions is contained in the original paper by Boris Delaunay (1934). The edge-flip algorithm is due to Charles Lawson (1977). 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, *e.g.*, Schumaker (1987). Unfortunately, the algorithm gets 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 (1978). 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 (Edelsbrunner, Tan and Waupotitsch 1992).

4. 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 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 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.

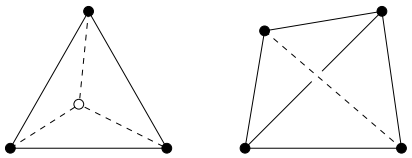

Fig. 14. To the left, the hollow vertex splits the triangle into three. To the right, the dashed diagonal replaces the solid diagonal

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 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 3 less than the number of edges in the final link, which is the same as 3 less than the degree of p_i in D_i.

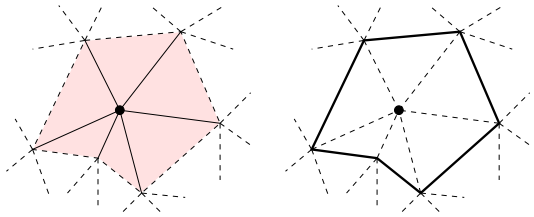

Fig. 15. The star of the solid vertex to the left and the link of the same vertex to the right

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 $p_1, p_2, \ldots, p_n$. 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 the 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

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

The number of flips needed to add all last points is therefore at most $6n - 3n = 3n$. The total number of flips is

$$F(n) \leq n \cdot F(n-1) + 3n \leq 3n \cdot n!.$$

Indeed, if we assume $F(n-1) \leq 3(n-1) \cdot (n-1)!$ we get $n \cdot F(n-1) + 3n = 3(n-1) \cdot n! + 3n \leq 3n \cdot n!$. 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 3, and therefore the expected number of flips to add any point is at most 3.

The history DAG

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 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 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.

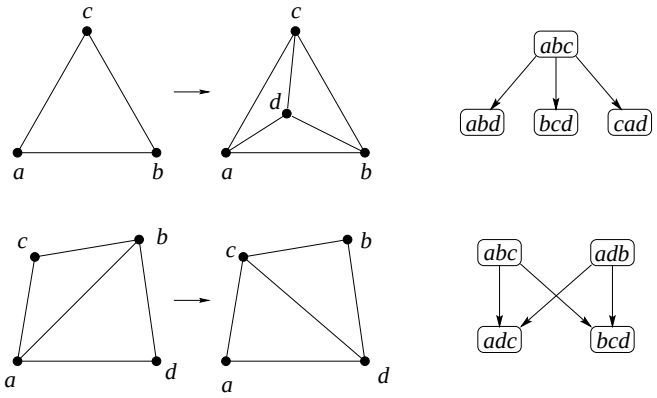

Fig. 16. Splitting a triangle generates a parent with three children. Flipping an edge generates two parents sharing the same two children

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 $0, 1, \ldots, j$ 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 $\sigma_0, \sigma_1, \ldots, \sigma_j$ 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 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 $\operatorname{card} X_j$, for j from 0 to $i - 2$.

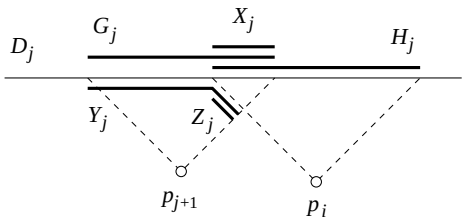

Fig. 17. The intervals represent sets of triangles removed or added when we insert p_{j+1} and/or p_i to D_j

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

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

Expectations

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

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

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

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

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 $\frac{3}{j+1}$ of being in the star of p_i. The expected number of triangles removed by inserting p_{j+1} is at most 4. Because the expectation of a product is the product of expectations, we have $\mathrm{E}[\,z_j\,] \leq \frac{4 \cdot 3}{j+1}$. The

expected length of the search path for p_i is

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

The expected total time spent on searching in the history DAG is $\sum \mathrm{E}[\,x_j\,] \leq c \cdot n \log n$.

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

Bibliographic notes

The randomized incremental algorithm of this section is due to Guibas, Knuth and Sharir (1992). It has been generalized to three and higher dimensions by Edelsbrunner and Shah (1996). All this is based on earlier work on randomized algorithms and in particular on the methods developed by Clarkson and Shor (1989). 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 (1993).

5. 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 $a = (\alpha_1, \alpha_2)$, $b = (\beta_1, \beta_2)$, $c = (\gamma_1, \gamma_2)$ be three points in the plane. We consider a, b, c degenerate if they lie on a common line. This includes the case where 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 non-degenerate 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 $a_0 = (0,0)$, $b_0 = (1,0)$, $c_0 = (0,1)$. 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 determinant. □

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 $\alpha_3 = \alpha_1^2 + \alpha_2^2$, *etc.* 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 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 non-degenerate 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 $d_0 = (\frac{1}{2}, \frac{1}{2})$ and $a_0 = (0,0), b_0 = (1,0), c_0 = (0,1)$ as before. Point d_0 lies at the centre 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. □

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 18. Each manifold

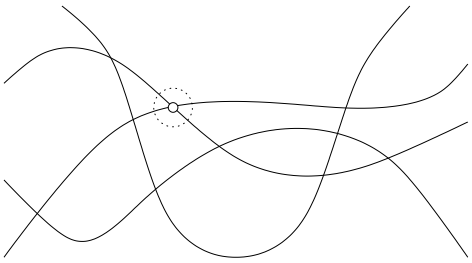

Fig. 18. Schematic picture of the union of $(2n-1)$-dimensional manifolds in $2n$-dimensional space. The marked point lies on two manifolds and thus has two degenerate subconfigurations. The dotted circle bounds a neighbourhood, and most points in that neighbourhood are non-degenerate

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 neighbourhood 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 neighbourhoods thus implies that there are arbitrarily close non-degenerate perturbations of S.

Perturbation

We construct a non-degenerate perturbation of S 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

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

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 non-degenerate.
II. $Z(\varepsilon)$ retains all non-degenerate 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

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

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,

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

where

$$\begin{aligned} \Delta \quad &= \quad \begin{bmatrix} 1 & \zeta_{2i-1} & \zeta_{2i} \\ 1 & \zeta_{2j-1} & \zeta_{2j} \\ 1 & \zeta_{2k-1} & \zeta_{2k} \end{bmatrix}, \\ \Delta_1 \quad &= \quad \begin{bmatrix} 1 & \zeta_{2j} \\ 1 & \zeta_{2k} \end{bmatrix}, \end{aligned}$$

$$\Delta_2 = \begin{bmatrix} 1 & \zeta_{2j-1} \\ 1 & \zeta_{2k-1} \end{bmatrix},$$
$$\Delta_3 = \begin{bmatrix} 1 & \zeta_{2i} \\ 1 & \zeta_{2k} \end{bmatrix}.$$

Property I is satisfied because the fifth term is non-zero, and its influence on the sign of the determinant cannot be cancelled 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 need 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 (Danzig 1963). It reappeared in computational geometry with the work of four independent

groups of authors. Edelsbrunner and Mücke (1990) develop SoS, which is the method described in this section. Yap (1990) studies the class of perturbations obtained with different orderings of infinitesimals. Emiris and Canny (1995) introduce perturbations along straight lines. Michelucci (1995) 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 (1998) addresses this issue, offers a unified view of symbolic perturbation, and discusses limitations of the method. Fortune and Van Wyk (1996) describe a floating point filter that reduces the overhead needed for exact computation.

6. Constrained triangulation

This section studies triangulations in the plane constrained by edges specified as part of the input. We show that there is a unique constrained triangulation that is closest, in some sense, to the (unconstrained) Delaunay triangulation.

Constraining line segments

The preceding sections constructed triangulations for a given set of points. The input now consists of a finite set of points, $S \subseteq \mathbb{R}^2$, together with a finite set of line segments, L, each connecting two points in S. We require that any two line segments are either disjoint or meet at most in a common endpoint. A *constrained triangulation* of S and L is a triangulation of S that contains all line segments of L as edges. Figure 19 illustrates that we can construct a constrained triangulation by adding straight edges connecting points in S as long as they have no interior points in common with previous edges.

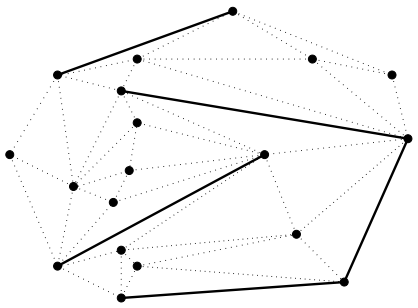

Fig. 19. Given the points and solid edges, we form a constrained triangulation by adding as many dotted edges as possible without creating improper intersections

Plane-sweep algorithm

The idea of organizing the actions of the algorithm around a line sweeping over the plane leads to an efficient way of constructing constrained triangulations. We use a vertical line that sweeps over the plane from left to right, as shown in Figure 20. The algorithm uses two data structures. The *schedule*, X, orders events in time. The *cross-section*, Y, stores the line segments in L that currently intersect the sweep-line. The algorithm is defined by the following invariant.

(I) At any moment in time, the partial triangulation contains all edges in L, a maximal number of edges connecting points to the left of the sweep-line, and no other edges.

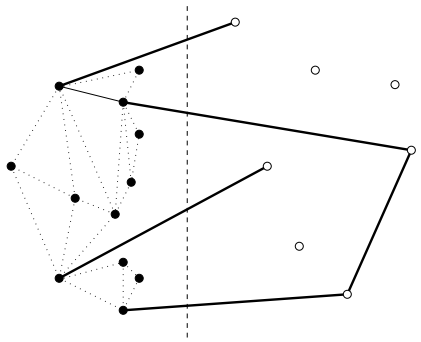

Fig. 20. Snapshot of plane-sweep constructing a constrained triangulation

Invariant (I) implies that between the left endpoints of two constraining line segments adjacent along the sweep-line we have a convex chain of edges in the partial triangulation. To ensure that new edges can each be added in constant time, the algorithm remembers the rightmost vertex in each chain. If the point p encountered next by the sweep-line falls inside one of the intervals along the sweep-line, the algorithm connects p to the corresponding rightmost vertex. It then proceeds in a clockwise and an anticlockwise order along the convex chain. Each step either adds a new edge or it ends the walk. If p is the right endpoint of a line segment then it separates two intervals along the sweep-line, and the algorithm does the same kind of walking twice, once for each interval.

The schedule is constructed by sorting the points in S from left to right, which can be done in time $O(n \log n)$, where $n = \operatorname{card} S$. The cross-section is maintained as a dictionary, which supports search, insertion, deletion all in time $O(\log n)$. There is a search for each point in S and an insertion–deletion pair for each line segment in L, taking total time $O(n \log n)$. Fewer

than $3n$ edges are added to the triangulation, each in constant time. The plane-sweep algorithm thus constructs a constrained triangulation of S and L in time $O(n \log n)$.

Constrained Delaunay triangulations

The triangulations constructed by plane-sweep usually have many small and large angles. We use a notion of visibility between points to introduce a constrained triangulation that avoids small angles to the extent possible.

Points $x, y \in \mathbb{R}^2$ are *visible* from each other if xy contains no point of S in its interior and it shares no interior point with a constraining line segment. Formally, $\operatorname{int} xy \cap S = \emptyset$ and $\operatorname{int} xy \cap uv = \emptyset$ for all $uv \in L$. Assume general position. An edge ab, with $a, b \in S$, belongs to the *constrained Delaunay triangulation* of S and L if

(i) $ab \in L$, or
(ii) a and b are visible from each other and there is a circle passing through a and b such that each point inside this circle is invisible from every point $x \in \operatorname{int} ab$.

We say the circle in (ii) *witnesses* the membership of ab in the constrained Delaunay triangulation. Figure 21 illustrates this definition. Note if $L = \emptyset$ then the constrained Delaunay triangulation of S and L is the Delaunay triangulation of S. More generally, it is however unclear that what we defined is indeed a triangulation. For example, why is it true that no two edges satisfying (i) or (ii) cross?

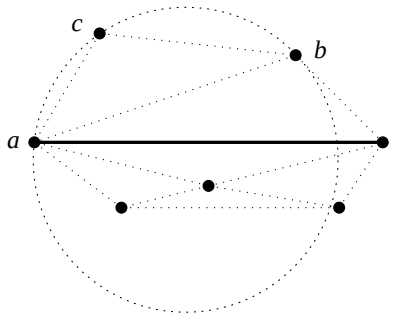

Fig. 21. Constrained Delaunay triangulation for seven points and one constraining line segment. The circumcircle of abc encloses only points that are invisible from all points of $\operatorname{int} ab$

Edge flipping

We introduce a generalized concept of being locally Delaunay, and use it to prove that the above definition makes sense. Let K be any constrained

triangulation of S and L. An edge $ab \in K$ is *locally Delaunay* if $ab \in L$, or ab is a convex hull edge, or d lies outside the circumcircle of abc, where $abc, abd \in K$.

Constrained Delaunay Lemma. If every edge of K is locally Delaunay then K is the constrained Delaunay triangulation of S and L.

Proof. We show that every edge in K satisfies (i) or (ii) and therefore belongs to the constrained Delaunay triangulation. The claim follows because every additional edge crosses at least one edge of K and therefore of the constrained Delaunay triangulation.

Let ab be an edge and p a vertex in K. Assume $ab \notin L$, for else ab belongs to the constrained Delaunay triangulation for trivial reasons. Assume also that ab is not a convex hull edge, for else we can easily find a circle passing through a and b such that p lies outside the circle. Hence, ab belongs to two triangles, and we let abc be the one separated from p by the line passing through ab. We need to prove that if p is visible from a point $x \in \text{int } ab$ then it lies outside the circumcircle of abc. Consider the sequence of edges in K crossing xp. Since x and p are visible from each other, all these edges are not in L. We can therefore apply the argument of the proof of the original Delaunay lemma, which is illustrated in Figure 9. □

This result suggests we use the edge-flipping algorithm to construct the constrained Delaunay triangulation. The only difference to the original edge-flipping algorithm is that edges in L are not flipped, since they are locally Delaunay by definition. As before, the algorithm halts in time $O(n^2)$ after fewer than $\binom{n}{2}$ flips. The analysis of angle changes during an edge flip presented in Section 3 implies that the MaxMin Angle Lemma also holds in the constrained case.

Constrained MaxMin Angle Lemma. Among all constrained triangulations of S and L, the constrained Delaunay triangulation maximizes the minimum angle.

Extended Voronoi diagrams

Just as for ordinary Delaunay triangulations, every constrained Delaunay triangulation has a dual Voronoi diagram, but in a surface that is more complicated than the Euclidean plane. Imagine $\mathbb{R}^2$ is a sheet of paper, Σ_0, with the points of S and the line segments in L drawn on it. For each $\ell_i \in L$, we cut Σ_0 open along ℓ_i and glue another sheet Σ_i, which is also cut open along ℓ_i. The gluing is done around ℓ_i such that every traveller who crosses ℓ_i switches from Σ_0 to Σ_i and *vice versa*. A cross-section of the particular gluing necessary to achieve that effect is illustrated in Figure 22. It is not possible to do this without self-intersections in $\mathbb{R}^3$, but in $\mathbb{R}^4$ there is already

sufficient space to embed the resulting surface. Call Σ_0 the *primary sheet*, and after the gluing is done we have $m = \operatorname{card} L$ *secondary sheets* Σ_i for $1 \leq i \leq m$. Each secondary sheet is attached to Σ_0, but not connected to any of the other secondary sheets. For each point $x \in \mathbb{R}^2$, we now have $m + 1$ copies $x_i \in \Sigma_i$, one on each sheet.

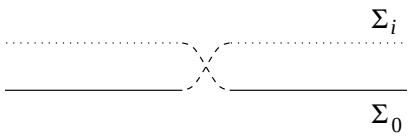

Fig. 22. The gap in Σ_0 represents the cut along ℓ_i. The secondary sheet Σ_i is glued to Σ_0 so that each path crossing ℓ_i switches sheets

We know what it means for two points on the primary sheet to be visible from each other. For other pairs we need a more general definition. For $i \neq 0$, points $x_0 \in \Sigma_0$ and $y_i \in \Sigma_i$ are *visible* if xy crosses ℓ_i, and ℓ_i is the first constraining line segment crossed if we traverse xy in the direction from x to y. The *distance* between points x_0 and y_i is

$$d(x_0, y_i) \quad = \quad \begin{cases} \|x - y\|, & \text{if } x_0, y_i \text{ are visible,} \\ \infty, & \text{otherwise.} \end{cases}$$

The new distance function is used to define the *extended Voronoi diagram*, which is illustrated in Figure 23. A circle that witnesses the membership of an edge ab in the constrained Delaunay triangulation has its centre on the primary or on a secondary sheet. In either case, that centre is closer to a and b than to any other point in S. This implies that the Voronoi regions of a and b meet along a non-empty common portion of their boundary. Conversely, every point on an edge of the extended Voronoi diagram is the centre of a circle witnessing the membership of the corresponding edge in the constrained Delaunay triangulation.

Bibliographic notes

The idea of using plane-sweep for solving two-dimensional geometric problems is almost as old as the field of computational geometry itself. It was propagated as a general algorithmic paradigm by Nievergelt and Preparata (1982). Constrained Delaunay triangulations were independently discovered by Lee and Lin (1986) and by Paul Chew (1987). Extended Voronoi diagrams are due to Raimund Seidel (1988), who used them to construct constrained Delaunay triangulations in worst-case time $O(n \log n)$.

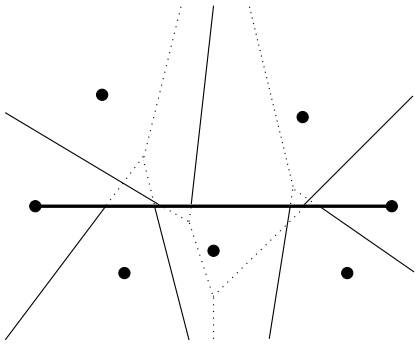

Fig. 23. Extended Voronoi diagram dual to the constrained Delaunay triangulation in Figure 21. There is only one secondary sheet glued to the primary one. The solid Voronoi edges lie in the primary sheet and the dotted ones in the secondary sheet

7. Delaunay refinement

This section demonstrates the use of Delaunay triangulations in constructing triangle meshes in the plane. The idea is to add new vertices until the triangulation forms a satisfying mesh. Constraining edges are covered by Delaunay edges, although forcing them into the triangulation as we did in Section 6 would also be possible.

The meshing problem

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 two-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 24 shows input and output for a particular mesh generation problem.

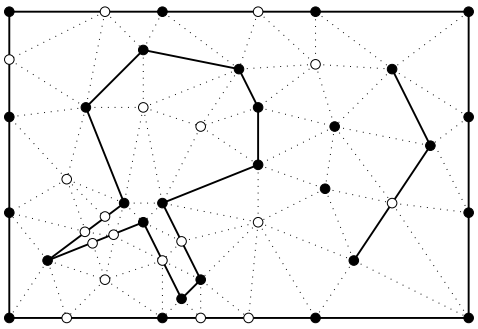

Fig. 24. The solid vertices and edges define the input graph, and together with the hollow vertices and dotted edges they define the output triangulation

Triangle quality

The quality of a triangle abc is measured 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 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 converse is not true. The aspect ratio is the length of the longest edge, which we assume is ac, divided by the distance of b from ac; see Figure 25. Suppose the smallest angle occurs at a. Then $\|b - x\| = \|b - a\| \cdot \sin\theta$, where x is the orthogonal projection of b onto ac. The edge ab is at least as long as cb, and therefore $\|b - a\| \geq \|c - a\|/2$. It follows that

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

In words, the aspect ratio is linearly related to one over the smallest angle. If θ is bounded away from zero then the aspect ratio is bounded from above by some constant, and *vice versa*.

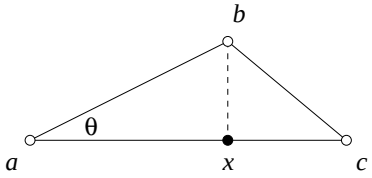

Fig. 25. Triangle with base ac, height bx, and minimum angle θ

The goal is to construct K so its smallest angle is no less than some constant, and the number of triangles in K is at most some constant times the minimum. We see from the example in Figure 24 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 this section, we assume that 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, as in Figure 26. We say these vertices *encroach upon* ab, and we use function SPLIT_1 to add the midpoint of ab and to repair the Delaunay triangulation with a series of edge flips.

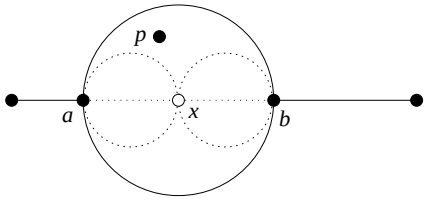

Fig. 26. Vertex p encroaches upon segment ab. After adding the midpoint, we have two smaller diameter circles, both contained in the diameter circle of ab

(2) Suppose a triangle abc in the current Delaunay triangulation K is skinny, that is, it has an angle less than the required lower bound. We use function SPLIT_2 to add the circumcentre as a new vertex, such as point x in Figure 27. Since its circumcircle is no longer empty, triangle abc is guaranteed to be removed by one of the edge flips used to repair the Delaunay triangulation.

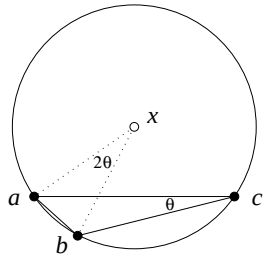

Fig. 27. The angle $\angle axb$ is twice the angle $\angle acb$

Algorithm

The first priority of the algorithm is to cover input edges, and its second priority is to resolve skinny triangles. Before starting the algorithm, we place G inside a rectangular box B. The purpose of the box is to contain the points added by the algorithm and thus prevent the perpetual growth of the meshed region. To be specific, we take B three times the size of the minimum enclosing rectangle of G. Box B has space for nine copies of the rectangle, and we place G inside the centre copy. Each side of B is decomposed into three equally long edges. Refer to Figure 24, where for aesthetic reasons the box is drawn smaller than required but with the right combinatorics. Initially, K is the Delaunay triangulation of the input points, which includes the 12 vertices along the boundary of B.

```
loop
  while ∃ encroached segment ab do
    SPLIT1(ab)
  endwhile;
  if no skinny triangle left then exit endif;
  let abc ∈ K be skinny and x its circumcentre;
  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 circumcentre x will ever lie outside the box. This is because the initial 12 or fewer triangles next to the box boundary have non-obtuse angles opposite to boundary edges. Since the circumcircles of Delaunay triangles are empty, this implies that all circumcentres lie inside B. The algorithm maintains the non-obtuseness of angles opposing input edges and thus limits circumcentres to lie inside B.

Preliminary analysis

The behaviour of the algorithm is expressed by the points it adds as vertices to the mesh. We already know that all points lie on the boundary or inside the box B, which has finite area. If we can prove that no two points are less than a positive constant 2ε apart, then this implies that the algorithm halts after adding finitely many points. To be specific, let w be the width and h the height of B. The area of the box obtained by extending B by ε on each side is $A = (w+2\varepsilon)(h+2\varepsilon)$. The number of points inside the box is $n \leq A/\varepsilon^2\pi$. This is because the disks with radius ε centred at the vertices of the mesh have pairwise disjoint interiors, and they are all contained in the extended box. This type of area argument is common in meshing and related

to packing, as illustrated in Figure 28. The existence of a positive ε will be established in Section 8. The analysis there will refine the area argument by varying the sizes of disks with their location inside the meshing region.

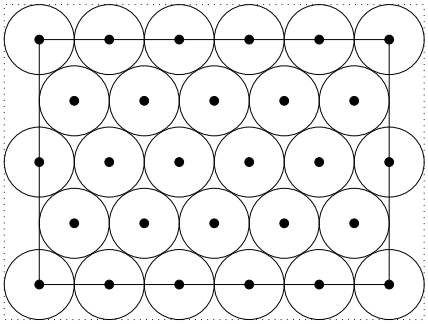

Fig. 28. The centres of the disk are contained in the inner box, and the disks are contained in the box enlarged by the disk radius in all four directions

In terms of running time, the most expensive activity is edge flipping used to repair the Delaunay triangulation. The expected linear bound on the number proved in Section 4 does not apply because points are not added in a random order. The total number of flips is less than $\binom{n}{2}$. This implies an upper bound of $O(n^2)$ on the running time, as long as the cost for adding a new vertex is at most $O(n)$.

Bibliographic notes

The algorithm described in this section is due to Jim Ruppert (1995). Experiments suggest it achieves best results if the skinny triangles are removed in order of non-decreasing smallest angle. A predecessor of Ruppert's algorithm is the version of the Delaunay refinement method by Paul Chew (1989). That algorithm is also described in Chew (1993), where it is generalized to surfaces in three-dimensional space. The main contribution of Ruppert is a detailed analysis of the Delaunay refinement method. The gained insights are powerful enough to permit modifications of the general method that guarantee a close to optimum mesh.

8. Local feature size

This section analyses the Delaunay refinement algorithm of Section 7. It proves an upper bound on the number of triangles generated by the algorithm and an asymptotically matching lower bound on the number of triangles that must be generated.

Local feature size

We understand the Delaunay refinement algorithm through relating 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 centre 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.

The three cases are illustrated in Figure 29. Because of (i) we have $f(a) \leq \|a - b\|$ for all vertices $a \neq b$ in G. The local feature size satisfies a one-sided Lipschitz inequality, which implies continuity.

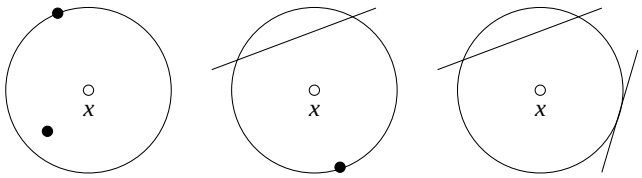

Fig. 29. In each case, the radius of the circle is the local feature size at x

Lipschitz Condition. $|f(x) - f(y)| \leq \|x - y\|$.

Proof. To get a contradiction, assume there are points x, y with $f(x) < f(y) - \|x - y\|$. The disk with radius $f(x)$ around x is contained in the interior of the disk with radius $f(y)$ around y. We can thus shrink the disk of y while maintaining its non-empty intersection with two disjoint vertices or edges of G. This contradicts the definition of $f(y)$. □

Constants

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

$$1 + \sqrt{2}C_2 \quad \leq \quad C_1 \quad \leq \quad \frac{C_2 - 1}{2 \sin \alpha},$$

where α is the lower bound on angles enforced by the Delaunay refinement algorithm. The constraints that correspond to the two inequalities are bounded by lines, and we have a solution if and only if the slope of the first line is greater than that of the second, $1/\sqrt{2} > 2 \sin \alpha$. Figure 30 illustrates the two constraints for $\alpha < \arcsin \frac{1}{2\sqrt{2}} = 20.7\ldots^\circ$. The two lines intersect at a point in the positive quadrant, and the coordinates of that point are the smallest constants C_1 and C_2 that satisfy the inequalities.

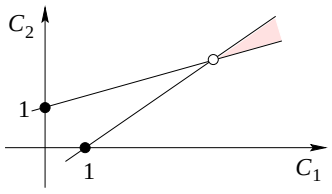

Fig. 30. Each line bounds a half-plane of points (C_1, C_2) that satisfy one inequality. The shaded wedge contains all points that satisfy both inequalities

Invariants

The algorithm starts with the vertices of G and generates all other vertices in sequence. We show that, when a new vertex is added, its distance to already present vertices is not much smaller than the local feature size.

Invariants. Let p and x be two vertices such that x was added after p. If x was 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 first prove (B). In this case, point x is the circumcentre of a skinny triangle abc. Let $\theta < \alpha$ at c be the smallest angle in abc, as in Figure 27. Assume that either a and b both belong to G or that a was 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 was added as the circumcentre of a circle with radius r'. Prior to the addition of a this circle was 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 was added as the midpoint of a segment. Then $f(a) \leq L \cdot C_1$, again by induction.

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 $L = 2r \sin\theta$ from Figure 27 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 $C_2 \geq 1 + 2C_1 \cdot \sin\alpha$ we get

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

as required.

We use a similar argument to prove (A). In this case, x is the midpoint of a segment ab. Let $r = \|x - a\| = \|x - b\|$ be the radius of the smallest circle passing through a and b, and let p be a vertex that encroaches upon ab, as in Figure 26. Consider first the case where p lies on an input edge that shares no endpoint with the input edge of 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 circumcentre. Let p be this circumcentre 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 induction 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. □

Upper bound

Invariants (A) and (B) guarantee 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. Recall that K is the final triangulation generated by the Delaunay refinement algorithm.

Smallest Gap Lemma. $\|a - b\| \geq \frac{f(a)}{1+C_1}$ for all vertices $a, b \in K$.

Proof. If b precedes a then $\|a - b\| \geq f(a)/C_1 \geq f(a)/(1+C_1)$. Otherwise, we have $\|b - a\| \geq f(b)/C_1$ and therefore

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

as claimed. □

Since vertices cannot get arbitrarily close to each other, we can use an area argument to show that the algorithm halts after adding a finite number of vertices. We relate the number of vertices to the integral of $1/f^2(x)$. Recall that B is the bounding box used in the construction of K.

Upper Bound Lemma. The number of vertices in K is at most some constant times $\int_B \mathrm{d}x/f^2(x)$.

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

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

This is a constant times the number of vertices. □

Two geometric results

We prepare the lower bound argument with two geometric results on triangles with angles no smaller than some constant $\alpha > 0$. Two edges of such a triangle abc cannot be too different in length, and specifically, $\frac{\|a-c\|}{\|a-b\|} \leq \varrho = 1/\sin\frac{\alpha}{2}$. If we have a chain of triangles connected through shared edges, the length ratio cannot exceed ϱ^t, where t is the number of triangles. Two edges sharing a common vertex are connected by the chain of triangles around that vertex. That chain cannot be longer than $\frac{2\pi}{\alpha}$, simply because we cannot pack more angles into 2π.

Length Ratio Lemma. The length ratio between two edges sharing a common vertex is at most $\varrho^{2\pi/\alpha}$.

The second result concerns covering a triangle with four disks, one each around the three vertices and the circumcentre. For each vertex we take a disk with radius c_0 times the length of the shortest edge. For the circumcentre we take a disk with radius $1 - c_2$ times the circumradius. For a general triangle, we can keep c_0 fixed and force c_2 as close to zero as we like, just by decreasing the angle. If angles cannot be arbitrarily small, then c_2 can also be bounded away from zero.

Triangle Cover Lemma. For each constant $c_0 > 0$ there is a constant $c_2 > 0$ such that the four disks cover the triangle.

Proof. Refer to Figure 31. Let R be the circumradius and ab be the shortest of the three edges. Its length is $\|a - b\| \geq 2R \cdot \sin\frac{\alpha}{2}$. The disk around a covers all points at distance at most $c_0 \cdot \|a - b\|$ from a, and we assume without loss of generality that $c_0 < \frac{1}{2}$. The distance between the circumcentre, z,

and the point $y \in ab$ at distance $c_0 \cdot \|a-b\|$ from a is

$$\begin{aligned} \|y-z\| &< \sqrt{R^2 - c_0^2 \|a-b\|^2} \\ &\leq \sqrt{R^2 \cdot \left(1 - 4c_0^2 \cdot \sin^2 \frac{\alpha}{2}\right)} \\ &< R \cdot \left(1 - 2c_0^2 \cdot \sin^2 \frac{\alpha}{2}\right). \end{aligned}$$

All other points on triangle edges not covered by disks around a, b, c are at most that distance from z. Since c_0 and α are positive constants, $c_2 = 2c_0^2 \cdot \sin^2 \frac{\alpha}{2}$ is also a positive constant. □

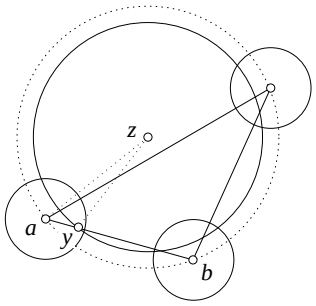

Fig. 31. The disks constructed for a triangle and its three vertices cover the triangle

Lower bound

The reason for picking the disk of radius $(1-c_2)R$ around the circumcentre is that for a point x inside this disk the local feature size cannot be arbitrarily small. In particular, it cannot be smaller than the distance from the circumcircle times the cosine of half the smallest angle, $f(x) \geq c_2 R \cdot \cos \frac{\alpha}{2}$. To get a similar result for disks around vertices, let L be the length of the shortest edge incident to a vertex a. The local feature size of a is at least $L \cdot \sin \alpha$. By choosing $c_0 = \frac{\sin \alpha}{2}$ we get $f(a) \geq 2c_0 L$ and therefore $f(x) \geq f(a) - \|a-x\| \geq c_0 L$ for every point x inside the disk with radius $c_0 L$ around a.

We use these observations to show that any algorithm that constructs triangles with angles no smaller than some constant $\alpha > 0$ generates at least some constant times the integral of $1/f^2(x)$ many vertices. It follows that the algorithm in Section 7 constructs meshes with asymptotically minimum size.

Lower Bound Lemma. If K is a triangle mesh of G with all angles larger than α, then the number of vertices is at least some constant times $\int_B \mathrm{d}x/f^2(x)$.

Proof. Around each vertex $a \in K$ draw a disk with radius equal to $\frac{\sin\alpha}{2}$ times the length of the shortest incident edge. Let $c_0 = \frac{\sin\alpha}{2}\varrho^{\pi/\alpha}$ and use the Triangle Cover Lemma to pick a matching constant $c_2 > 0$. For each triangle $abc \in K$ draw the disk with radius $1 - c_2$ times the circumradius around the circumcentre. Each triangle is covered by its four disks, which implies that the mesh is covered by the collection of disks.

For each disk D_i in the collection, let f_i be the minimum local feature size at any point $x \in D_i$. By what we said earlier, that minimum is at least some constant fraction of the radius of D_i, $f_i \geq r_i/C$. Given that the disks cover the mesh we have

$$\begin{aligned}\int_B \frac{\mathrm{d}x}{f^2(x)} &\leq \sum_i \int_{D_i} \frac{\mathrm{d}x}{f^2(x)} \\ &\leq \sum_i \frac{r_i^2\pi}{f_i^2} \\ &\leq \sum_i C^2\pi.\end{aligned}$$

The number of triangles is less than twice the number of vertices, which we denote as n. Hence,

$$n \geq \sum_i \frac{1}{3} \geq \frac{1}{3C^2\pi}\int_B \frac{\mathrm{d}x}{f^2(x)},$$

as claimed. □

Bibliographic notes

The idea of using the local feature size function in the analysis of the Delaunay refinement algorithm is due to Jim Ruppert. The details of the analysis left out in the journal publication Ruppert (1995) can be found in the technical report Ruppert (1992). Bern, Eppstein and Gilbert (1994) show that the same technical result (constant minimum angle and constant times minimum number of triangles) can also be achieved using quad-trees. Experimentally, the approach with Delaunay triangulations seems to generate meshes with fewer and nicer triangles. One reason for the better performance might be the absence of any directional bias from Delaunay triangulations.

9. Lifting and polarity

The Delaunay tetrahedrization of a finite set of points in $\mathbb{R}^3$ is dual to the Voronoi diagram of the same set. This section introduces both concepts and shows how they can be obtained as projections of the boundary of convex polyhedra.

Voronoi diagrams

The *Voronoi region* of a point p in a finite collection $S \subseteq \mathbb{R}^3$ is the set of points at least as close to p as to any other point in S,

$$V_p = \{x \in \mathbb{R}^3 : \|x - p\| \leq \|x - q\|, \ \forall q \in S\}.$$

Each inequality defines a closed half-space, and V_p is the intersection of a finite collection of such half-spaces. In other words, V_p is a convex polyhedron, maybe like the one shown in Figure 32. In the generic case, every vertex of V_p belongs to only three facets and three edges of the polyhedron. If V_p is bounded then it is the convex hull of its vertices. It is also possible that V_p is unbounded. This is the case if and only if there is a plane through p with all points of S on or on one side of the plane.

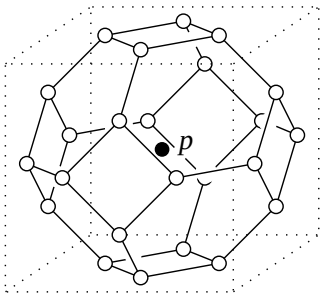

Fig. 32. The Voronoi polyhedron of a point in a body-centred cube lattice. The relevant neighbours of the cube centre p are the eight corners of the cube and the centres of the six adjacent cubes

The Voronoi regions together with their shared facets, edges, vertices form the *Voronoi diagram* of S. A point x that belongs to k Voronoi regions is equally far from the k generating points. It follows that the k points lie on a common sphere. If the points are in general position then $k \leq 4$. A Voronoi vertex x belongs to at least four Voronoi regions, and assuming general position it belongs to exactly four regions.

Delaunay tetrahedrization

We obtain the *Delaunay tetrahedrization* by taking the dual of the Voronoi diagram. The Delaunay vertices are the points in S. The Delaunay edges connect generators of Voronoi regions that share a common facet. The Delaunay facets connect generators of Voronoi regions that share a common edge. Assuming general position, each edge is shared by three Voronoi regions and the Delaunay facets are triangles. The Delaunay polyhedra

connect generators of Voronoi regions that share a common vertex. Assuming general position, each vertex is shared by four Voronoi regions and the Delaunay polyhedra are tetrahedra. Consider point p in Figure 32. Its Voronoi polyhedron has 14 facets, 36 edges, and 24 vertices. It follows that p belongs to 14 Delaunay edges, 36 Delaunay triangles, and 24 Delaunay tetrahedra, as illustrated in Figure 33.

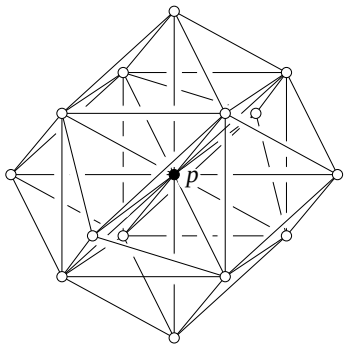

Fig. 33. The Delaunay neighbourhood of a point in a body-centred cube lattice

Assuming general position of the points in S, the Delaunay tetrahedrization is a collection of simplices. To prove that it is a simplicial complex, we still need to show that the simplices avoid improper intersections. We do this by introducing geometric transformations that relate Voronoi diagrams and Delaunay tetrahedrizations in $\mathbb{R}^3$ with boundary complexes of convex polyhedra in $\mathbb{R}^4$.

Distance maps

The square distance from $p \in S$ is the map $\pi_p : \mathbb{R}^3 \to \mathbb{R}$ defined by $\pi_p(x) = \|x - p\|^2$. Its graph is a paraboloid of revolution in $\mathbb{R}^4$. We simplify notation by supressing the difference between a function and its graph. Figure 34 illustrates this idea in one lower dimension. Take the collection of all square distance functions defined by points in S. The pointwise minimum is the map $\pi_S : \mathbb{R}^3 \to \mathbb{R}$ defined by

$$\pi_S(x) \quad = \quad \min\{\pi_p(x) : p \in S\}.$$

Its graph is the lower envelope of the collection of paraboloids. By definition of Voronoi region, $\pi_S(x) = \pi_p(x)$ if and only if $x \in V_p$. We can therefore think of V_p as the projection of the portion of the lower envelope contributed by the paraboloid π_p.

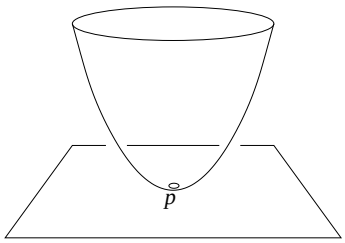

Fig. 34. The graph of the square
distance function of a point p in the plane

Linearization

All square distance functions have the same quadratic term, which is $\|x\|^2$. If we subtract that term we get linear functions, namely

$$\begin{aligned} f_p(x) &= \pi_p(x) - \|x\|^2 \\ &= (x-p)^T \cdot (x-p) - x^T \cdot x \\ &= -2p^T \cdot x + \|p\|^2. \end{aligned}$$

The graph of f_p is a hyperplane in $\mathbb{R}^4$. The same transformation warps the hyperplane $x_4 = 0$ to the upside-down paraboloid Π defined as the graph of the map defined by $\Pi(x) = -\|x\|^2$. Figure 35 shows the result of the transformation applied to the plane and paraboloid in Figure 34. We can apply the transformation to the entire collection of paraboloids at once. Each point in $\mathbb{R}^4$ travels vertically, that is, parallel to the x_4-axis. The travelled distance is the square distance to the x_4-axis. Paraboloids go to hyperplanes, intersections of paraboloids go to intersections of hyperplanes, and the lower envelope of the paraboloids goes to the lower envelope of the hyperplanes.

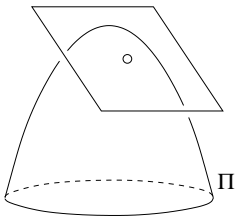

Fig. 35. The plane in Figure 34
becomes an upside-down paraboloid,
and the paraboloid becomes a plane

Replace each hyperplane by the closed half-space bounded from above by the hyperplane. The intersection of the half-spaces is a convex polyhedron F in $\mathbb{R}^4$, and the lower envelope of the hyperplanes is the boundary of F. It is

a complex of convex faces of dimension 3, 2, 1, 0. Since the transformation moves points vertically, the projection onto $x_4 = 0$ of the lower envelope of paraboloids and the lower envelope of hyperplanes are the same. In particular, the projection of each three-dimensional face of F is a Voronoi region, and the projection of the entire boundary complex is the Voronoi diagram.

Polarity

We still need to describe what all this has to do with the Delaunay tetrahedrization of S. Instead of addressing this question directly, we first study the relationship between non-vertical hyperplanes and their polar points in $\mathbb{R}^4$.

A non-vertical hyperplane is the graph of a linear function $f : \mathbb{R}^3 \to \mathbb{R}$, which can generally be defined by a point $p \in \mathbb{R}^3$ and a scalar $c \in \mathbb{R}$, that is,

$$f(x) \quad = \quad -2p^T \cdot x + \|p\|^2 - c.$$

The hyperplane parallel to f and tangent to Π is defined by the equation $-2p^T \cdot x + \|p\|^2$. The vertical distance between the two hyperplanes is $|c|$. The *polar point* of f is $g = f^* = (p, -\|p\|^2 + c)$. The vertical distance between g and f is $2|c|$, and the parallel tangent hyperplane lies right in the middle between g and f. Furthermore, the vertical line through g also passes through the point where the tangent hyperplane touches Π. It follows that $g \in \Pi$ if and only if f is tangent to Π. Figure 36 shows a few examples of hyperplanes and their polar points in $\mathbb{R}^2$. Since hyperplanes are non-vertical, the points lying *above*, *on*, *below* are unambiguously defined. Let f_1, f_2 be two non-vertical hyperplanes and g_1, g_2 their polar points.

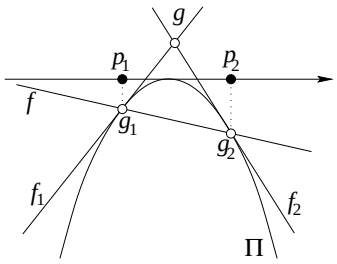

Fig. 36. Points g_1, g_2, g are polar to the lines (hyperplanes) f_1, f_2, f. Lines f_1, f_2 are warped images of the distance square functions of the points p_1, p_2 on the real line

Order Reversal Claim. Point g_1 lies above, on, below hyperplane f_2 if and only if point g_2 lies above, on, below hyperplane f_1.

Proof. Let $g_i = (p_i, -\|p_i\|^2 + c_i)$ for $i = 1, 2$. The algebraic expression for g_1 above f_2 is

$$-\|p_1\|^2 + c_1 \quad > \quad -2p_2^T \cdot p_1 + \|p_2\|^2 - c_2.$$

We move terms left and right and use the fact that vector products are commutative to get

$$-\|p_2\|^2 + c_2 \quad > \quad -2p_1^T \cdot p_2 + \|p_1\|^2 - c_1.$$

This is the algebraic expression for g_2 above f_1. The arguments for point g_1 lying on and below hyperplane f_2 are the same. □

Polar polyhedron

We are now ready to construct the Delaunay tetrahedrization as the projection of the boundary complex of a convex polyhedron in $\mathbb{R}^4$. For each point $p \in S$, let $g_p = (p, -\|p\|^2)$ be the polar point of the corresponding hyperplane. All points g_p lie on the upside-down paraboloid Π, as shown in Figure 37. For a non-vertical hyperplane f, we consider the closed half-space bounded from above by f. Let G be the intersection of all such half-spaces that contain all points g_p. G is a convex polyhedron in $\mathbb{R}^4$. Its boundary consists of the upper portion of the convex hull boundary plus the silhouette extended to infinity in the $-x_4$ direction. The Order Reversal Claim implies the following correspondence between G and F. A hyperplane *supports* G if it has non-empty intersection with the boundary and empty intersection with the interior.

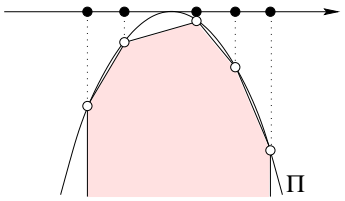

Fig. 37. The boundary complex of the shaded polyhedron projects onto the Delaunay tetrahedrization of the set of solid points

Support Claim. A hyperplane f supports G if and only if the polar point $g = f^*$ lies in the boundary of F.

Imagine exploring G by rolling the supporting hyperplane along its boundary. The dual image of this picture is the polar point moving inside the boundary of F. For each k-dimensional face of G we get a $(3-k)$-dimensional face of F and *vice versa*. An exception is the set of vertical faces of G, which

do not correspond to any faces of F, except possibly to faces stipulated at infinity. The relationship between the two boundary complexes is the same as that between the Delaunay tetrahedrization and the Voronoi diagram. The isomorphism between the boundary complex of F and the Voronoi diagram implies the isomorphism between the boundary complex of G (excluding vertical faces) and the Delaunay tetrahedrization. Since the vertices of G project onto points in S, it follows that the boundary complex of G projects onto the Delaunay tetrahedrization of S. This finally implies that there are no improper intersections between Delaunay simplices. The Delaunay tetrahedrization of a set S of finitely many points in general position is indeed a simplicial complex.

Bibliographic notes

Voronoi diagrams and Delaunay triangulation are named after Georges Voronoi (1907/08) and Boris Delaunay (1934). The concepts themselves are older and can be traced back to prominent mathematicians of earlier centuries, including Friedrich Gauß and René Descartes. The connection to convex polytopes has also been known for a long time. The combinatorial theory of convex polytopes is a well developed field within mathematics. We refer to the texts by Branko Grünbaum (1967) and by Günter Ziegler (1995) for excellent sources of the accumulated knowledge in that subject.

10. Weighted distance

The correspondence between Voronoi diagrams and convex polyhedra hints at a generalization of Voronoi and Delaunay diagrams forming a richer class of objects. This section describes this generalization using points with real weights. Within this larger class of diagrams we find a symmetry between Voronoi and Delaunay diagrams absent in the smaller class of unweighted diagrams.

Commuting diagram

Figure 38 illustrates the correspondence between Voronoi diagrams and Delaunay tetrahedrizations in $\mathbb{R}^3$ and convex polyhedra in $\mathbb{R}^4$, as worked out in Section 9. V and D are dual to each other. F is obtained from V through linearization of distance functions, and V is formed by the projections of the boundary complex of F. F and G are polar to each other. G is the convex hull of the points projected onto Π (extended to infinity along the $-x_4$-direction), and D is the projection of the boundary complex of G.

We call G an *inscribed* polyhedron because each vertex lies on the upside-down paraboloid Π. Similarly, we call F a *circumscribed* polyhedron because each hyperplane spanned by a 3-face is tangent to the Π. Being inscribed or

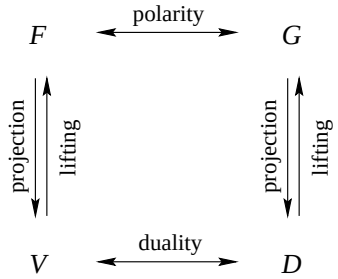

Fig. 38. Relationship between Voronoi diagram, V, Delaunay tetrahedrization, D, and convex polyhedra, F and G

circumscribed is a rather special property. We use weights to generalize the concepts of Voronoi diagrams and Delaunay tetrahedrization in a way that effectively frees the polyhedra from being inscribed or circumscribed. For technical reasons, we still require that every vertical line intersects F in a half-line and G either in a half-line or the empty set. This is an insubstantial although sometimes inconvenient restriction.

Weighted points

We prepare the definition of weighted Delaunay tetrahedrization by introducing points with real weights. It is convenient to write the weight of a point as the square of a non-negative real or a non-negative multiple of the imaginary unit. We think of the weighted point $\hat{p} = (p, P^2) \in \mathbb{R}^3 \times \mathbb{R}$ as the sphere with centre $p \in \mathbb{R}^3$ and radius P. The *power* or *weighted distance function* of $\hat{p}$ is the map $\pi_{\hat{p}} : \mathbb{R}^3 \to \mathbb{R}$ defined by

$$\pi_{\hat{p}}(x) \quad = \quad \|x - p\|^2 - P^2.$$

It is positive for points x outside the sphere, zero for points on the sphere, and negative for points inside the sphere. The various cases permit intuitive geometric interpretations of weighted distance. For example, for positive P^2 and x outside the sphere, it is the square length of a tangent line segment connecting x with a point on the sphere. This is illustrated in Figure 39. What is it if x lies inside the sphere? In Section 2, we have seen that the set of points with equal weighted distance from two circles is a line. Similarly, the set of points with equal weighted distance from two spheres in $\mathbb{R}^3$ is a plane. If the two spheres intersect then the plane passes through the intersection circle, and if the two spheres are disjoint and lie side by side then the plane separates the two spheres.

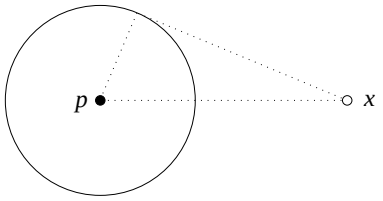

Fig. 39. The segment px, the tangent segment from x to the circle, and the connecting radius form a right-angled triangle

Orthogonality

Given two spheres or weighted points $\hat{p} = (p, P^2)$ and $\hat{q} = (q, Q^2)$, we generalize weighted distance to the symmetric form

$$\pi_{\hat{p},\hat{q}} = \|p - q\| - P^2 - Q^2.$$

For $Q^2 = 0$, this is the weighted distance from q to $\hat{p}$, and for $P^2 = 0$, this is the weighted distance from p to $\hat{q}$. We call $\hat{p}$ and $\hat{q}$ *orthogonal* if $\pi_{\hat{p},\hat{q}} = 0$. Indeed, if $P^2, Q^2 > 0$ then $\pi_{\hat{p},\hat{q}} = 0$ if and only if the two spheres meet in a circle and the two tangent planes at every point of this circle form a right angle. Orthogonality is the key concept in generalizing Delaunay to weighted Delaunay tetrahedrizations. We call $\hat{p}$ and $\hat{q}$ *further than orthogonal* if $\pi_{\hat{p},\hat{q}} > 0$.

Let us contemplate for a brief moment how weights affect the lifting process. The graph of the weighted distance function is a paraboloid whose zero-set, $\pi_{\hat{p}}^{-1}(0)$, is the sphere $\hat{p}$. We can linearize as before and get a hyperplane defined by

$$\begin{aligned} f_{\hat{p}}(x) &= \pi_{\hat{p}}(x) - \|x\|^2 \\ &= -2p^T \cdot x + \|p\|^2 - P^2. \end{aligned}$$

We can also polarize and get

$$g_{\hat{p}} = (p, -\|p\|^2 + P^2).$$

Orthogonality between two spheres now translates to a point-hyperplane incidence.

Orthogonality Claim. Spheres $\hat{p}$ and $\hat{q}$ are orthogonal if and only if point $g_{\hat{p}}$ lies on the hyperplane $f_{\hat{q}}$.

Proof. The algebraic expression for $g_{\hat{p}} \in f_{\hat{q}}$ is

$$-2q^T \cdot p + \|q\|^2 - Q^2 = -\|p\|^2 + P^2.$$

This is equivalent to

$$(p-q)^T \cdot (p-q) - P^2 - Q^2 \quad = \quad 0,$$

which is equivalent to $\pi_{\hat{p},\hat{q}} = 0$. □

Weighted Delaunay tetrahedrization

Let S be a finite set of spheres. Depending on the application, we think of an element of S as a point in $\mathbb{R}^3$ or a weighted point in $\mathbb{R}^3 \times \mathbb{R}$. The weighted distance can be used to construct the *weighted Voronoi diagram*, and the *weighted Delaunay tetrahedrization* is dual to that diagram, as usual. Instead of going through the technical formalism of the construction, which is pretty much the same as for unweighted points, we illustrate the concept in Figure 40. For unweighted points, a tetrahedron belongs to the Delaunay tetrahedrization if and only if the circumsphere passing through the four vertices is empty. For weighted points, the circumsphere is replaced by the *orthosphere*, which is the unique sphere orthogonal to all four spheres whose centres are the vertices of the tetrahedron. Its centre is the Voronoi vertex shared by the four Voronoi regions, and its weight is the common weighted distance of that vertex from the four spheres. We summarize by generalizing the Circumcircle Claim of Section 2 to three dimensions and to the weighted case.

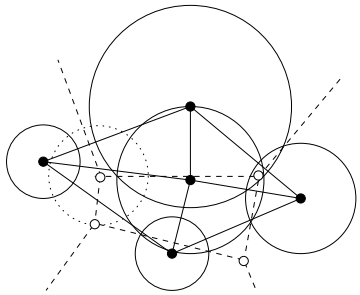

Fig. 40. Dashed weighted Voronoi diagram and solid weighted Delaunay triangulation of five weighted points in the plane. Each Voronoi vertex is the centre of a circle orthogonal to the generating circles of the regions that meet at that vertex. Only one such circle is shown

Orthosphere Claim. A tetrahedron belongs to the weighted Delaunay tetrahedrization if and only if the orthosphere of the four spheres is further than orthogonal from all other sphere in the set.

A sphere in S is *redundant* if its Voronoi region is empty. By definition, the centre of a sphere is a vertex of the weighted Delaunay triangulation

if and only if it is non-redundant. All extreme points are non-redundant, which implies that the underlying space is the convex hull of S, as in the unweighted case.

Local convexity

Recall the Delaunay lemma of Section 3, which states that a triangulation of a finite set in $\mathbb{R}^2$ is the Delaunay triangulation if and only if every one of its edges is locally Delaunay. This result generalizes to three (and higher) dimensions and to the weighted case. For the purpose of this discussion, we define a *tetrahedrization* of S as a simplicial complex K whose underlying space is $\operatorname{conv} S$ and whose vertex set is a subset of S. A triangle abc in K is *locally convex* if

(i) it belongs to only one tetrahedron and therefore bounds the convex hull of S, or
(ii) it belongs to two tetrahedra, $abcd$ and $abce$, and $\hat{e}$ is further than orthogonal from the orthosphere of $abcd$.

If all triangles in K are locally convex, then after lifting we get the boundary complex of a convex polyhedron. This is consistent with the right side of the commuting diagram in Figure 38. However, to be sure this polyhedron is G, we also require that no lifted point lies vertically below the boundary.

Local Convexity Lemma. If $\operatorname{Vert} K$ contains all non-redundant weighted points and every triangle is locally convex, then K is the weighted Delaunay tetrahedrization of S.

The proof is rather similar to that of the Delaunay lemma in Section 3 and does not need to be repeated. Similarly, we can extend the Acyclicity Lemma of Section 2 to three (and higher) dimensions and to the weighted case. Details should be clear and are omitted.

Bibliographic notes

Weighted Voronoi diagrams are possibly as old as unweighted ones. Some of the earliest references appear in the context of quadratic forms, which arise in the study of the geometry of numbers (Gruber and Lekkerkerker 1987). These forms are naturally related to weighted as opposed to unweighted diagrams. Examples of such work are the papers by Dirichlet (1850) and Voronoi (1907/08). Weighted Delaunay triangulations and their generalizations to three and higher dimensions seem less natural and have a shorter history. Nevertheless, they have already acquired at least three different names, namely regular triangulations (Billera and Sturmfels 1992) and coherent triangulations (Gelfand, Kapranov and Zelevinsky 1994) besides the one used in this paper.

11. Flipping

The goal of this section is to generalize the idea of edge flipping to three and higher dimensions. We begin with two classic theorems in convex geometry. Helly's theorem talks about the intersection structure of convex sets. It can be proved using Radon's theorem, which talks about partitions of finite point sets and is directly related to flips in d dimensions. We then define flips and discuss structural issues that arise in $\mathbb{R}^3$.

Radon's theorem

This is a result on $n \geq d+2$ points in $\mathbb{R}^d$. The case of $n = 4$ points in $\mathbb{R}^2$ is related to edge flipping in the plane.

Radon's Theorem. Every collection S of $n \geq d+2$ points in $\mathbb{R}^d$ has a partition $S = A \dot{\cup} B$ with $\operatorname{conv} A \cap \operatorname{conv} B \neq \emptyset$.

Proof. Since there are more than $d+1$ points, they are affinely dependent. Hence there are coefficients λ_i, not all zero, with $\sum \lambda_i p_i = 0$ and $\sum \lambda_i = 0$. Let I be the set of indices i with $\lambda_i > 0$, and let J contain all other indices. Note that $c = \sum_{i \in I} \lambda_i = -\sum_{j \in J} \lambda_j > 0$, and also

$$x \;=\; \frac{1}{c} \cdot \sum_{i \in I} \lambda_i p_i \;=\; -\frac{1}{c} \cdot \sum_{j \in J} \lambda_j p_j .$$

Let A be the collection of points p_i with $i \in I$ and let B contain all other points. Point x is a convex combination of the points in A as well as of the points in B. Equivalently, $x \in \operatorname{conv} A \cap \operatorname{conv} B$. □

A $(d+1)$-dimensional simplex has $d+2$ vertices and a face for every subset of the vertices. If we project its boundary complex onto $\mathbb{R}^d$ we get a simplex for every subset of at most $d+1$ vertices. By Radon's theorem, at least two of these simplices have an improper intersection. This intersection comes from projecting the two sides of the simplex boundary on top of each other.

Helly's theorem

This is a result on $n \geq d+2$ convex sets in $\mathbb{R}^d$. For $d = 1$ it states that if every pair of a collection of $n \geq 2$ closed intervals has a non-empty intersection then the entire collection has a non-empty common intersection. This is true because the premise implies that the rightmost left endpoint is to the left or equal to the leftmost right endpoint. The interval between these two endpoints belongs to every interval in the collection.

Helly's Theorem. If every $d+1$ sets in a collection of $n \geq d+2$ closed convex sets in $\mathbb{R}^d$ have a non-empty common intersection, then the entire collection has a non-empty intersection.

Proof. Assume inductively that the claim holds for $n-1$ closed convex sets. For each C_i in the collection of n sets, let p_i be a point in the common intersection of the other $n-1$ sets. Let S be the collection of points p_i. By Radon's theorem, there is a partition $S = A \dot{\cup} B$ and a point $x \in \operatorname{conv} A \cap \operatorname{conv} B$. By construction, $\operatorname{conv} A$ is contained in all sets C_j with $p_j \in B$, and symmetrically, $\operatorname{conv} B$ is contained in all sets C_i with $p_i \in A$. Hence, x is contained in every set of the collection. □

Flipside of a simplex

Consider the case $d = 2$. The projection of a 3-simplex (tetrahedron) onto $\mathbb{R}^2$ is either a convex quadrangle or a triangle. In the former case the two diagonals cross, and in the latter case one vertex lies in the triangle spanned by the other three. Both cases are illustrated in Figure 41. The direction of projection defines an *upper* and a *lower* side of the tetrahedron boundary, and the two sides meet along the *silhouette*. Let $\alpha = \operatorname{conv} A$ and $\beta = \operatorname{conv} B$ be the two faces whose projections have an improper intersection. They lie on opposite sides, and we assume that α belongs to the upper and β to the lower side. The quadrangle case defines an edge flip, which replaces the projection of the upper by the projection of the lower side, or *vice versa*. We also call this a 2-*to*-2 *flip* because it replaces 2 old by 2 new triangles. The triangle case defines a new type of flip, which we refer to as a 1-*to*-3 or a 3-*to*-1 *flip* depending on whether a new vertex is added or an old vertex is removed.

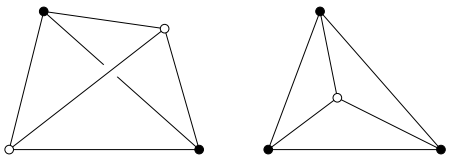

Fig. 41. The two generic projections of a tetrahedron onto the plane

How do these considerations generalize to the case $d = 3$? As illustrated in Figure 42, the projection of a 4-simplex onto $\mathbb{R}^3$ is either a double pyramid or a tetrahedron. In the double pyramid case, α is an edge and β is a triangle. There are three tetrahedra that share α and they form the upper side of the 4-simplex. The remaining two tetrahedra share β and form the lower side. The 3-*to*-2 *flip* replaces the projection of the upper side by the projection of the lower side, and the 2-*to*-3 *flip* does it the other way round. In the tetrahedron case, α is one vertex and β is the tetrahedron spanned by the other four vertices. The 1-*to*-4 *flip* adds α, effectively replacing β by four tetrahedra, and the 4-*to*-1 *flip* removes α.

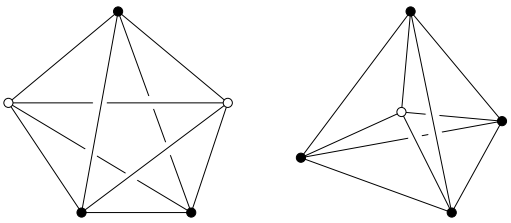

Fig. 42. The two generic projections
of a 4-simplex onto 3-dimensional space

Transformability

In using flips to construct a Delaunay tetrahedrization in $\mathbb{R}^3$, we encounter cases where we would like to flip but we cannot. This happens only for 2-to-3 flips. Let *abcd* and *bcde* share the triangle *bcd*. If the edge *ae* crosses *bcd* we can replace *abcd*, *bcde* by *baec*, *caed*, *daeb*, which is a 2-to-3 flip. However, if the edge *ae* misses *bcd*, as illustrated in Figure 43 where *ae* passes behind *bd*, we cannot add *ae* because it might cross other triangles in the current tetrahedrization. In this case, the union of the two tetrahedra is non-convex. Assume without loss of generality that *bd* is the non-convex edge. There are two cases. If *bd* belongs to only three tetrahedra then the third one is *abde*, and we can replace *abdc*, *cbde*, *ebda* by *bace*, *aced*. This is a 3-to-2 flip. However, if *bd* belongs to four or more tetrahedra then we are stuck and cannot remove the triangle *bcd*. This is the *non-transformable* case.

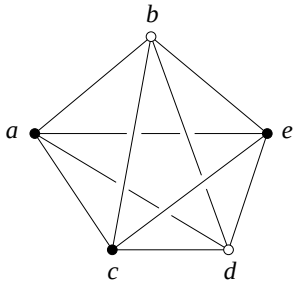

Fig. 43. The edge *ae* does not pass through
the triangle *bcd* but rather behind the edge *bd*

The reason for studying flips is of course the interest in an algorithm that constructs a weighted Delaunay tetrahedrization by flipping. The occurrence of non-transformable cases does not imply that all hope is lost. It might still be possible to flip elsewhere in a way that resolves non-transformable cases by changing their local neighbourhood. But this requires further analysis.

Bibliographic notes

Radon's theorem is a by-product of the attempt by Johann Radon (1921) to prove Helly's theorem, communicated to him by Eduard Helly (1923). The two theorems are equivalent and form a cornerstone of modern convex geometry. Helly was missing as a prisoner of war in Russia, so Radon published his theorem and proof. After returning from Russia, Helly published his theorem and his own proof, which is inductive in the size of the collection *and* the dimension. Years later, Helly generalized his theorem to a topological setting where convexity is replaced by requirements of connectivity (Helly 1930). The concept of an edge flip was generalized to three and higher dimensions by Lawson (1986) without, however, realizing the connection to Radon's theorem.

12. Incremental algorithm

This section generalizes the algorithm of Section 4 to three dimensions and to the weighted case. The algorithm is incremental and adds a point in a sequence of flips. We describe the algorithm, prove its correctness, and discuss its running time.

Algorithm

Let S be a finite set of weighted points in $\mathbb{R}^3$. We denote the points by $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n$ and assume they are in general position. To reduce the number of cases, we let $wxyz$ be a sufficiently large tetrahedron. In particular, we assume $wxyz$ contains all points of S in its interior. Define $S_i = \{w, x, y, z, \hat{p}_1, \hat{p}_2, \ldots, \hat{p}_i\}$ for $0 \leq i \leq n$, and let D_i be the weighted Delaunay tetrahedrization of S_i. The algorithm starts with D_0 and adds the weighted points in order. Adding $\hat{p}_1$ is done in a sequence of flips.

```
for i = 1 to n do
  find pqrs ∈ D_{i-1} that contains p_i;
  if p̂_i is non-redundant among p̂, q̂, r̂, ŝ then
    add p̂_i with a 1-to-4 flip
  endif;
  while ∃ triangle bcd not locally convex do
    flip bcd
  endwhile
endfor.
```

The algorithm maintains a tetrahedrization, which we denote as K. Sometimes, K is a weighted Delaunay tetrahedrization of a subset of the points, but often it is not. Consider flipping the triangle bcd in K. Let $abcd$ and $bcde$ be the two tetrahedra that share bcd. If their union is convex, then flipping

bcd means a 2-to-3 flip that replaces bcd by edge ae together with triangles aeb, aec, aed. Otherwise, we consider the subcomplex induced by a, b, c, d, e. It consists of the simplices in K spanned by subsets of the five points. If the underlying space of the induced subcomplex is non-convex then bcd cannot be flipped. If the underlying space is convex then it is either a double pyramid or a tetrahedron. In the former case, flipping means a 3-to-2 flip. In the latter case, flipping means a 4-to-1 flip, which effectively removes a vertex. The various types of flips are illustrated in Figure 44.

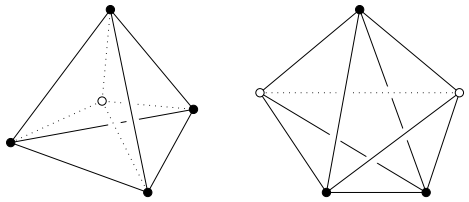

Fig. 44. To the left, a 1-to-4 or a 4-to-1 flip depending on whether the hollow vertex is added or removed. To the right, a 2-to-3 or a 3-to-2 flip depending on whether the dotted edge is added or removed

Stack of triangles

Flipping is done in a sequence controlled by a stack. At any moment, the stack contains all triangles in the link of p_i that are not locally convex. It may also contain other triangles in the link, but it contains each triangle at most once. Initially, the stack consists of the four triangles of $pqrs$. Flipping continues until the stack is empty.

```
while stack is non-empty do
  pop bcd from stack;
  if bcd ∈ K and bcd is not locally convex
              and bcd is transformable then
    apply a 2-to-3, 3-to-2, or 4-to-1 flip;
    push new link triangles on stack
  endif
endwhile.
```

Why can we restrict our attention to triangles in the link of p_i? Outside the link, K is equal to D_{i-1}, hence all triangles are locally convex. A triangle inside the link connects p_i with an edge cd in the link. Let xp_icd and p_icdy be the two tetrahedra sharing p_icd. If their union is convex, we can remove p_icd by a 2-to-3 flip. This creates a new tetrahedron $acde$ not incident to p_i, which contradicts that D_{i-1} is a weighted Delaunay tetrahedrization.

If their union is non-convex, the triangles xcd and cdy in the link are also not locally convex.

Correctness

Let K be the tetrahedrization at some moment in time after adding $\hat{p}_i$ when it is not yet the weighted Delaunay tetrahedrization of S_i. It suffices to show that K has at least one link triangle that is not locally convex and transformable. To get a contradiction, we suppose all triangles that are not locally convex are non-transformable. Let L be the set of tetrahedra in $K - \operatorname{St} p_i$ that have at least one triangle in the link. These tetrahedra form a spiky sphere around p_i, not unlike the spiky circle in Figure 45. Let $L' \subseteq L$ contain all tetrahedra whose triangles in the link are not locally convex. By assumption, $L' \neq \emptyset$. For each tetrahedron in L, consider the orthosphere $\hat{z}$ and the weighted distance $\pi_{\hat{p}_i,\hat{z}}$. Let $abcd \in L$ be the tetrahedron whose orthosphere minimizes that function. We have $abcd \in L'$, or equivalently $\pi_{\hat{p}_i,\hat{z}} < 0$, for else the triangle bcd in the link would be locally convex, and so would every other link triangle.

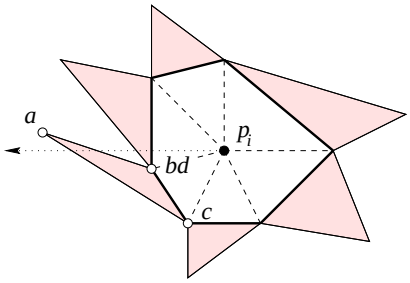

Fig. 45. The bold edges belong to the link of p_i and the shaded triangles belong to L

We argue that bcd is transformable. To get a contradiction assume it is not. Let bd be a non-convex edge of the union of $abcd$ and $bcdp_i$, and let $abdx$ be the tetrahedron on the other side of abd. If bd is the only non-convex edge then $x \neq p_i$, for else bcd would be transformable. Otherwise, there is another non-convex edge, say bc. Let $abcy$ be the tetrahedron on the other side of abc. If $x = y = p_i$ we again have a contradiction because this would imply that bcd is transformable. We may therefore assume that $x \neq p_i$. Equivalently, abd is not in the link of $\hat{p}_i$. Consider a half-line that starts at p_i and passes through an interior point of abd. After crossing the link, the half-line goes through a tetrahedron of L before it encounters $abcd$. This is illustrated in Figure 45. Outside the link, we have a genuine weighted Delaunay tetrahedrization, namely a portion of D_{i-1}. For tetrahedra in D_{i-1}, the weighted distance of $\hat{p}_i$ from their orthospheres increases along

the half-line, which contradicts the minimality assumption in the choice of *abcd*. This finally proves that flipping continues until D_i is reached.

Number of flips

To upper-bound the number of flips in the worst case, we interpret that algorithm as gluing 4-simplices to a three-dimensional surface consisting of tetrahedra in $\mathbb{R}^4$. Each flip corresponds to a 4-simplex. It either removes or introduces one or four edges. Once an edge is removed it cannot be introduced again. This implies that the total number of flips is less than $2\binom{n}{2} < n^2$. Modulo implementation details, we thus have an algorithm that constructs the Delaunay tetrahedrization of n points in $\mathbb{R}^3$ in $O(n^2)$ time. The size of the final Delaunay tetrahedrization is therefore at most some constant times n^2.

There are sets of n points in $\mathbb{R}^3$ with at least some constant times n^2 Delaunay tetrahedra. Take, for example, two skew lines and place $\frac{n}{2}$ unweighted points on each line, as shown in Figure 46. Consider two contiguous point on one line together with two contiguous points on the other line. The sphere passing through the four points is empty, which implies that the four points span a Delaunay tetrahedron. The total number of such tetrahedra is roughly $\frac{n^2}{4}$. However, for point sets that seem to occur in practice, the number of Delaunay tetrahedra is typically less than some constant times n. Examples of such sets are dense packing of spheres common in molecular modelling, and well-spaced sets as produced by three-dimensional mesh generation software.

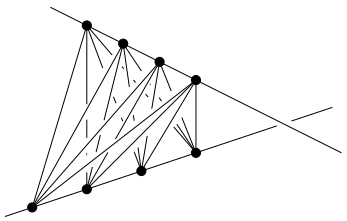

Fig. 46. A tetrahedral mesh whose edge skeleton contains a complete bipartite graph

Expected running time

It is a good idea to first compute a random permutation of the points so that the construction proceeds in a random order. However, because the size of the tetrahedrization can vary between linearly and quadratically many simplices, the analysis is more involved than in two dimensions. We cannot even claim that the expected running time is at most $\log_2 n$ times the size of

the final tetrahedrization. Indeed, this is false because there exist point sets with linear size Delaunay tetrahedrizations that reach quadratic intermediate size with positive constant probability. Nevertheless, such a claim holds if we further relativize the statement by drawing points from a fixed distribution. Suppose the expected size of the Delaunay tetrahedrization of k points chosen randomly from the distribution is $O(f(k))$. If $f(k) = \Omega(k^{1+\varepsilon})$, for some constant $\varepsilon > 0$, then the expected running time is $O(f(n))$, and otherwise it is $O(f(n) \log n)$. The argument is similar to the one presented in Section 4 and details are omitted.

Bibliographic notes

Algorithms that construct a Delaunay tetrahedrization in $\mathbb{R}^3$ through flips have first been considered by Barry Joe. In the paper Joe (1989) he gives an example where the non-transformable cases form a deadlock situation and flipping does not lead to the Delaunay tetrahedrization. In Joe (1991) he shows that flipping succeeds if the points are added one at a time. The proof of Joe's result in this section is taken from Edelsbrunner and Shah (1996), where the same is shown for weighted Delaunay tetrahedrization in $\mathbb{R}^d$.

13. Meshing polyhedra

In this paper, meshing a spatial domain means decomposing a polyhedron into tetrahedra that form a simplicial complex. This section introduces polyhedra and studies the problem of how many tetrahedra are needed to mesh them.

Polyhedra and faces

A *polyhedron* is the union of convex polyhedra, $P = \bigcup_{i \in I} \bigcap H_i$, where I is a finite index set and each H_i is a finite set of closed half-spaces. For example the polyhedron in Figure 47 can be specified as the union of four convex polyhedra. As we can see, faces are not necessarily simply connected. We use a definition that permits faces even to be disconnected.

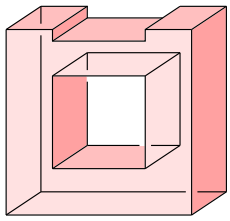

Fig. 47. A non-convex polyhedron

Let b be the open ball with unit radius centred at the origin of $\mathbb{R}^3$. For a point x we consider a sufficiently small neighbourhood, $N_\varepsilon(x) = (x + \varepsilon \cdot b) \cap P$. The *face figure* of x is the enlarged version of this neighbourhood within the polyhedron, $x + \bigcup_{\lambda>0} \lambda \cdot (N_\varepsilon(x) - x)$. A *face* of P is the closure of a maximal collection of points with identical face figure. To distinguish the faces of P from the edges and triangles of the Delaunay tetrahedrization to be constructed, we call 1- and 2-faces of P *segments* and *facets*. Observe that the polyhedron in Figure 47 has 24 vertices, 30 segments, 11 facets, and two 3-faces, namely the inside with face figure $\mathbb{R}^3$ and the outside with empty face figure. Six of the segments and three of the facets are non-connected. Two of the facets are connected but not simply connected, namely the front and the back facets.

Tetrahedrizations

A *tetrahedrization* of P is a simplicial complex K whose simplices decompose P. Since simplicial complexes are finite by definition, only bounded polyhedra have tetrahedrizations. A tetrahedrization of P triangulates every facet and every segment by a subcomplex each. Every vertex of P is necessarily also a vertex of K.

We will see shortly that every bounded polyhedron has a tetrahedrization. Interestingly, there are polyhedra whose tetrahedrizations have necessarily more vertices than the polyhedra. The smallest such example is the Schönhardt polyhedron shown in Figure 48. It can be obtained from a triangular prism by a slight rotation of one triangular facet relative to the other. The six vertices of the polyhedron span $\binom{6}{4} = 15$ tetrahedra, which we classify into three types exemplified by $abcA$, $abAB$, $bcCA$. All three tetrahedra share bA as an edge. But this edge lies outside the Schönhardt polyhedron, which implies that none of the 15 tetrahedra is contained in the polyhedron. The Schönhardt polyhedron can therefore not be tetrahedrized using tetrahedra spanned by its vertices. There are of course other tetrahedrizations. The simplest uses a vertex z in the centre and cones from z to the 6 vertices, 12 edges, 8 triangles in the boundary.

Fencing off

We give a constructive proof that every polyhedron P has a tetrahedrization. For simplicity we assume that P is everywhere three-dimensional. Equivalently, P is the closure of its interior, $P = \operatorname{cl} \operatorname{int} P$. It is convenient to place P in space such that no facet lies in a vertical plane and no segment is contained in a vertical line. Call two points $x, y \in P$ *vertically visible* if x, y lie on a common vertical line and the edge xy is contained in P. The *fence* of a segment consists of all points $x \in P$ vertically visible from some

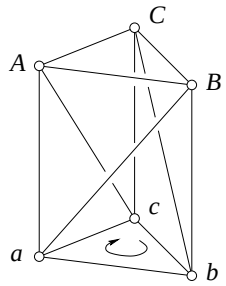

Fig. 48. The Schönhardt polyhedron.
The edges aB, bC, cA are non-convex

point y of the segment. The tetrahedrization is constructed in three steps, the first of which is illustrated in Figure 49.

`Step 1.` Erect the fence of each segment. The fences decompose P into vertical cylinders, each bounded by a top and a bottom facet and a circle of fence pieces called *walls*.

`Step 2.` Triangulate the bottom facet of every cylinder and erect fences from the new segments, effectively decomposing P into triangular cylinders.

`Step 3.` Decompose each wall into triangles and finally tetrahedrize each cylinder by constructing cones from an interior point to the boundary.

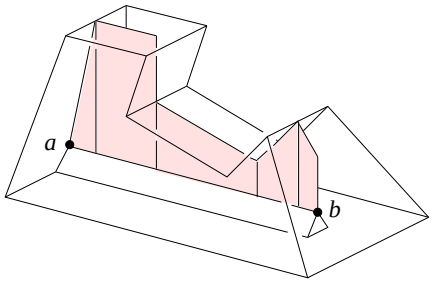

Fig. 49. The fence of the segment ab consists
of five walls, each a triangle or a quadrangle

Upper bound

We analyse the tetrahedrization obtained by erecting fences and prove that the final number of tetrahedra is at most some constant times the square of the number of segments.

Upper Bound Claim. The three steps tetrahedrize a bounded polyhedron with m segments using fewer than $28m^2$ tetrahedra.

Proof. Fences erected in `Step 1` may meet in vertical edges. Each intersection corresponds to a crossing between vertical projections of segments. The total number of crossings is at most $\binom{m}{2}$. Each segment creates a fence, and each crossing involving this segment may cut one wall of the fence into two. The total number of walls is therefore no more than $m + 2\binom{m}{2} = m^2$. A cylinder bounded by k walls is decomposed into $k-2$ triangular cylinders separated from each other by $k-3$ new walls. `Step 2` thus increases the total number of walls to less than $3m^2$. The total number of cylinders at this stage is less than $2m^2$. Each wall is a triangle or a quadrangle, and it may be divided into two by the piece of the segment that defines it. `Step 2` therefore triagulates each wall using four or fewer triangles, and it tetrahedrizes each cylinder using 14 or fewer tetrahedra. The final tetrahedrization thus contains fewer than $28m^2$ tetrahedra. □

Saddle surface

We prepare a matching lower bound by studying the *hyperbolic paraboloid* specified by the equation $x_3 = x_1 \cdot x_2$. Figure 50 illustrates the paraboloid by showing its intersection with the vertical planes $\pm x_1 \pm x_2 = 1$. A general

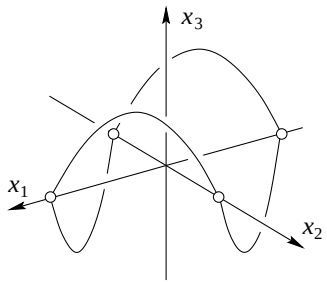

Fig. 50. Hyperbolic paraboloid indicated through its intersection with vertical walls

line in the x_1x_2-plane is specified by $ax_1 + bx_2 + c = 0$. To determine the intersection of the paraboloid with the vertical plane through that line, we can either express x_1 in terms of x_2 or *vice versa*,

$$\begin{aligned} x_3 &= -\frac{b}{a}x_2^2 - \frac{c}{a}x_2, \\ x_3 &= -\frac{a}{b}x_2^2 - \frac{c}{b}x_2. \end{aligned}$$

For $a \cdot b \neq 0$ we get a parabola. For $a = 0$ we get a line for every value of $\frac{c}{b}$, and we sample this family at integer values. Similarly, we sample the 1-parameter family of lines we get for $b = 0$ at integer values of $\frac{c}{a}$. Figure 51 shows a small portion of the two families in top view. If two points x and y lie on the paraboloid then the segment between them lies on the surface

if and only if the vertical projections of x, y onto the x_1x_2-plane line on a common horizontal or vertical line. If the line has positive slope then the segment lies below the surface, and if the line has negative slope then it lies above the surface.

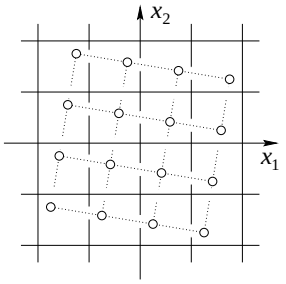

Fig. 51. Top view of hyperbolic paraboloid. We see samples of the two ruling families of lines and dotted edges connecting points sampled on the surface

Lower bound construction

We build a polyhedron Q out of a cube by cutting deep wedges, each close to a line of the two ruling families. The construction is illustrated in Figure 52. Assuming we have n cuts from the top and n from the bottom, we have $m = 14n + 8$ segments forming the polyhedron.

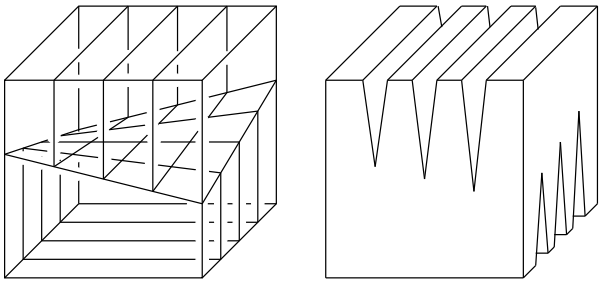

Fig. 52. Polyhedron Q with two families of cuts almost meeting along the saddle surface

Lower Bound Claim. Every tetrahedrization of Q consists of at least $(n+1)^2$ tetrahedra.

Proof. Consider the checkerboard produced by the $2n + 4$ lines on the saddle surface that mark the ends of the $2n$ cuts and the intersection with the boundary of the cube. Choose a point in each square of the checkerboard producing the slightly tilted square grid pattern of Figure 51. The edges

connecting any two points intersect at least one of the wedges, provided the sharp ends of the wedges reach sufficiently close to the saddle surface. It follows that in any tetrahedrization of Q, the $(n+1)^2$ points lie inside pairwise different tetrahedra. □

Bibliographic notes

The definition of a polyhedron as the union of intersections of closed half-spaces is taken from Hadwiger (1957). The definition of a face is taken from Edelsbrunner (1995) and should be contrasted with that suggestion in Grünbaum and Shephard (1994). The Schönhardt polyhedron was named after E. Schönhardt who described the polyhedron in 1928 (Schönhardt 1928). The same construction was mentioned 17 years earlier in a paper by Lennes (1911). Ruppert and Seidel (1992) build on this construction, and show that deciding whether or not a polyhedron can be tetrahedrized without adding new vertices is NP-complete. The quadratic upper and lower bounds for tetrahedrizing polyhedra are taken from a paper by Bernard Chazelle (1984).

14. Tetrahedral shape

This section looks at the various shapes tetrahedra can assume. For the time being, good shape quality is defined as having a small circumradius over shortest edge length ratio. We will see later that meshes of tetrahedra with small ratio also have nice combinatorial properties, such as constant size vertex stars.

Classifying tetrahedra

The classification of tetrahedra into shape types is a fuzzy undertaking. We normalize by scaling tetrahedra to unit diameter. A normalized tetrahedron has small volume either because its vertices are close to a line, or, if that is not the case, its vertices are close to a plane. In the first case, the tetrahedron is *skinny*, and we distinguish five types depending on how its vertices cluster along the line. Up to symmetry, the possibilities are 1-1-1-1, 1-1-2, 1-2-1, 1-3, 2-2, as shown from left to right in Figure 53. A *flat* tetrahedron has small volume but is not skinny. We have four types depending on whether two vertices are close to each other, three vertices lie close to a line, the orthogonal projection of the tetrahedron onto the close plane is a triangle, or the projection is a quadrangle. All four types are shown from left to right in Figure 54.

Circumradius over shortest edge length

A tetrahedron $abcd$ has a unique circumsphere. Let $R = R(abcd)$ be that radius and $L = L(abcd)$ the length of the shortest edge. We measure the

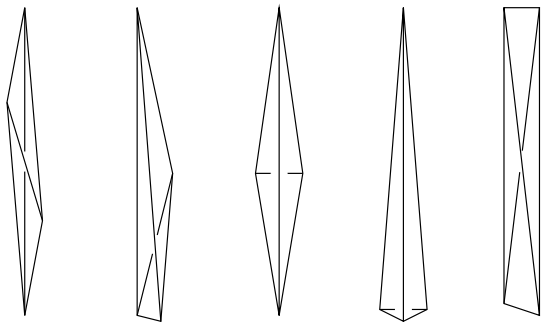

Fig. 53. Five fuzzy types of skinny tetrahedra

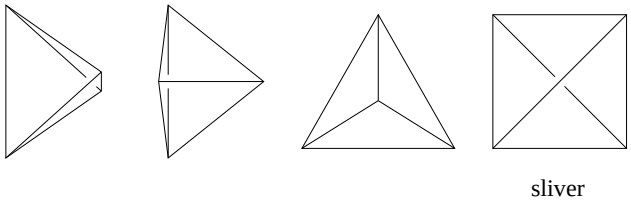

Fig. 54. Four fuzzy types of flat tetrahedra

quality of the tetrahedron shape by taking the ratio, that is,

$$\varrho \;=\; \varrho(abcd) \;=\; \frac{R}{L}.$$

We also define ϱ for triangles, taking the radius of the circumcircle over the length of the shortest edge. Observe that the ratio of a tetrahedron is always larger than or equal to the ratio of each of its triangles.

A triangle abc minimizes the ratio if and only if it is equilateral, in which case the circumcentre is also the barycentre,

$$y \;=\; \frac{1}{3}\cdot(a+b+c) \;=\; \frac{2}{3}\cdot x + \frac{1}{3}\cdot c,$$

where $x = \frac{1}{2}\cdot(a+b)$. Normalization implies that the three edges have length 1. The ratio is therefore equal to the circumradius, which is

$$\begin{aligned}\|c-y\| \;&=\; \frac{2}{3}\cdot\|c-x\| \;=\; \frac{2}{3}\cdot\sqrt{1-\frac{1}{4}}\\ &=\; \frac{\sqrt{3}}{3} \;=\; 0.577\ldots\end{aligned}$$

A tetrahedron $abcd$ minimizes the ratio if and only if it is regular, in which case the circumcentre is again the barycentre,

$$z \;=\; \frac{1}{4}\cdot(a+b+c+d) \;=\; \frac{3}{4}\cdot y + \frac{1}{4}\cdot d.$$

Normalization implies that the six edges have length 1. The ratio is therefore equal to the circumradius, which is

$$\begin{aligned} \|d-z\| &= \frac{3}{4}\cdot\|d-y\| = \frac{3}{4}\cdot\sqrt{1-\frac{3}{9}} \\ &= \frac{\sqrt{6}}{4} = 0.612\ldots \end{aligned}$$

Both calculations are illustrated in Figure 55.

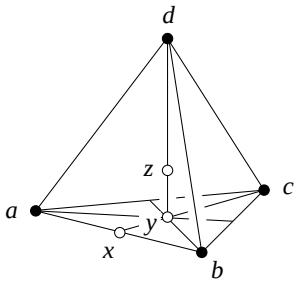

Fig. 55. A regular tetrahedron and the barycentres of an edge, a triangle, the tetrahedron

A *skinny* triangle has small area. It has either a short edge or a large circumradius. In either case, its ratio is large. A skinny tetrahedron has skinny triangles, hence its ratio is large. A flat triangle that is not a sliver has either a short edge or a large circumradius and thus a large ratio. The only remaining small volume tetrahedron is the sliver, and it can have ϱ as small as $\frac{\sqrt{2}}{2} = 0.707\ldots$ or even a tiny amount smaller.

Ratio property

A mesh of tetrahedra has the *ratio property for* ϱ_0 if $\varrho \leq \varrho_0$ for all tetrahedra. We assume that every triangle in the mesh is the face of a tetrahedron in the mesh. It follows that $\varrho \leq \varrho_0$ also for every triangle. We prove two elementary facts about edge lengths in a mesh K that has the ratio property for a constant ϱ_0.

Claim A. If abc is a triangle in K then

$$\frac{1}{2\varrho_0}\cdot\|a-b\| \leq \|a-c\| \leq 2\varrho_0\cdot\|a-b\|.$$

Proof. The length of an edge is at most twice the circumradius, $\|a-b\| \leq 2Y$. By assumption, $\|a-b\| \geq Y/\varrho_0$. The same inequalities hold for $\|a-c\|$, which implies the claim. □

Next we show that, if K has the ratio property and it is a Delaunay tetrahedrization, then edges that share a common endpoint and form a small angle cannot have very different lengths. For this to hold, it is not necessary that the two edges belong to a common triangle. Define

$$\eta_0 = \arctan 2\left(\varrho_0 - \sqrt{\varrho_0^2 - 1/4}\right).$$

Since ϱ_0 is a constant, so is η_0.

Claim B. If the angle between ab and ap is less than η_0 then

$$\frac{1}{2} \cdot \|a-b\| \;<\; \|a-p\| \;<\; 2 \cdot \|a-b\|.$$

Proof. Consider the circumsphere of a tetrahedron that contains ab as an edge, and let $\hat{y} = (y, Y^2)$ be the circle in which the plane passing through a, b, p intersects the sphere. We use Figure 56 as an illustration throughout

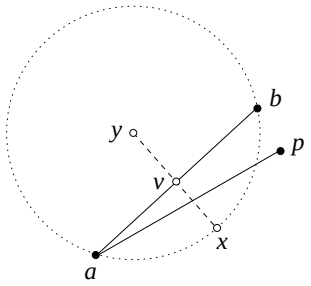

Fig. 56. Section through a circumsphere of a Delaunay tetrahedron with edge ab

the proof. Let v be the midpoint of ab, and let x be the point on the circle such that y, v, x lie in this sequence on a common line. We have $Y \le \varrho_0 \cdot \|a-b\|$ by assumption. The distance between x and v is

$$\begin{aligned} \|x-v\| &= Y - \sqrt{Y^2 - \|a-b\|^2/4} \\ &\ge \left(\varrho_0 - \sqrt{\varrho_0^2 - 1/4}\right) \cdot \|a-b\|, \end{aligned}$$

because the difference between Y and $\sqrt{Y^2 - C}$ decreases with increasing Y. The angle between ab and ax is

$$\begin{aligned} \angle bax &= \arctan \frac{2\|x-v\|}{\|a-b\|} \\ &\ge \arctan 2\left(\varrho_0 - \sqrt{\varrho_0^2 - 1/4}\right) \\ &= \eta_0. \end{aligned}$$

The claimed lower bound follows because the circle forces ap to be at least as long as ax, which is longer than half of ab. The claimed upper bound on the length of ap follows by a symmetric argument that reverses the roles of b and p. □

Length variation

We use Claims A and B to show that the length variation of edges with a common endpoint a in K is bounded by some constant. As before, we assume K has the ratio property and is a Delaunay tetrahedrization. Define $m_0 = 2/(1 - \cos \frac{\eta_0}{4})$ and $\nu_0 = 2^{2m_0-1} \cdot \varrho_0^{m_0-1}$. Since ϱ_0 and η_0 are constants, so are m_0 and ν_0.

Length Variation Lemma. If ab, ap are edges in K then

$$\frac{1}{\nu_0} \cdot \|a - b\| \ < \ \|a - p\| \ < \ \nu_0 \cdot \|a - b\|.$$

Proof. Let Σ be the sphere of directions around a. We form a maximal packing of circular caps, each with angle $\eta_0/4$. This means if y is the centre and x a boundary point of a cap then $4\angle xay = \eta_0$. The area of each cap is $(1 - \cos \frac{\eta_0}{4})/2$ times the area of Σ, which implies that there are at most m_0 caps.

By increasing the caps to radius $\eta_0/2$ we change the maximal packing into a covering of Σ. For each edge ab in the star of a, let $b' \in \Sigma$ be the radial projection of b. Similarly, for each triangle abc consider the arc on Σ that is the radial projection of bc. The points and arcs form a planar graph. Let ab be the longest and ap the shortest edge in the star of a. We walk in the graph from b' to p'. This path leads from cap to cap, and we record the sequence ignoring detours that return to previously visited caps. The sequence consists of at most m_0 caps. Let us track the edge length during the walk. As long as we stay within a cap, Claim B implies the length decreases by less than a factor $\frac{1}{2}$. If we step from one cap to the next, Claim A implies the length decreases by at most a factor $\frac{1}{2\varrho_0}$. Hence $\|a - p\| > \frac{1}{\nu_0} \cdot \|a - b\|$. The upper bound follows by a symmetric argument that exchanges b and p. □

Constant degree

A straightforward volume argument together with the Length Variation Lemma implies that each vertex in K belongs to at most some constant number of edges. Define $\delta_0 = (2\nu_0^2 + 1)^3$, which is a constant.

Degree Lemma. Every vertex a in K belongs to at most δ_0 edges.

Proof. Let ab be the longest and ap the shortest edge in the star of a. Assume without loss of generality that $\|a-p\| = 1$. Let c be a neighbour of a and let d be a neighbour of c. We have $\|a-c\| \geq 1$ by assumption and $\|c-d\| \geq \frac{1}{\nu_0}$ by the Length Variation Lemma. For each neighbour c of a let Γ_c be the open ball with centre c and radius $\frac{1}{2\nu_0}$. The balls are pairwise disjoint and fit inside the ball Γ with centre a and radius $\|a-b\| + \frac{1}{2\nu_0}$. The volume of Γ is

$$\begin{aligned} \operatorname{vol}\Gamma &= \frac{4\pi}{3}\left(\|a-b\| + \frac{1}{2\nu_0}\right)^3 \\ &\leq \frac{4\pi}{3}\left(\frac{2\nu_0^2+1}{2\nu_0}\right)^3 \\ &= (2\nu_0^2+1)^3 \cdot \operatorname{vol}\Gamma_c. \end{aligned}$$

In words, at most $\delta_0 = (2\nu_0^2+1)^3$ neighbour balls fit into Γ. This implies that δ_0 is an upper bound on the number of neighbours of a. □

The constant δ_0 in the Degree Lemma is miserably large. The main reason is that the constant ν_0 in the Length Variation Lemma is miserably large. It would be nice to find a possibly more direct proof of that lemma and bring the constant down to reasonable size.

Bibliographic notes

The idea of measuring the quality of a tetrahedron by its circumradius over shortest edge length ratio is due to Miller and co-authors (Miller, Talmor, Teng and Walkington 1995). The proofs of the Length Variation and Degree Lemmas are taken from the same source. Further results on meshes of tetrahedra that have the ratio property can be found in the doctoral thesis by Talmor (1997).

15. Delaunay refinement

This section generalizes the Delaunay refinement algorithm of Section 7 from two to three dimensions. The additional dimension complicates matters. In particular, special care must be taken to avoid infinite loops bouncing back and forth between refining segments and facets of the input polyhedron.

Refinement algorithm

For technical reasons, we restrict ourselves to bounded polyhedra P without interior angles smaller than $\frac{\pi}{2}$. The condition applies to angles between two segments, between a segment and a facet, and between two facets. The polyhedron in Figure 47 satisfies the condition, but the polyhedron in Figure 48 does not. The goal is to construct a Delaunay tetrahedrization D

with a subcomplex $K \subseteq D$ that subdivides P and has the ratio property for a constant ϱ_0. The first step of the algorithm computes D as the Delaunay tetrahedrization of the set of vertices of P. Unless we are lucky, there will be segments that are not covered by edges of D, and there will be facets that are not covered by triangles of D. To recover these segments and facets, we add new points and update the Delaunay tetrahedrization using the incremental algorithm of Section 12. The points are added using the three rules given below.

We need some definitions. A segment of P is decomposed into *subsegments* by vertices of the Delaunay tetrahedrization that lie on the segment, and a facet is decomposed into (triangular) *subfacets* by the Delaunay triangulation of the vertices on the facet and its boundary. A vertex *encroaches upon* a subsegment if it is enclosed by the diameter sphere of that subsegment, and it *encroaches upon* a subfacet if it is enclosed by the equator sphere of that subfacet. Both spheres are the smallest that pass through all vertices of the subsegment and the subfacet.

`Rule 1.` If a subsegment is encroached upon, we split it by adding the midpoint as a new vertex to the Delaunay tetrahedrization. The new subsegments may or may not be encroached upon, and splitting continues until none of the subsegments is encroached upon.

`Rule 2.` If a subfacet is encroached upon, we split it by adding the circumcentre x as a new vertex to the Delaunay tetrahedrization. However, if x encroaches upon one or more subsegments then we do not add x and instead split the subsegments.

`Rule 3.` If a tetrahedron inside P has circumradius over shortest edge length ratio $R/L > \varrho_0$ then we split the tetrahedron by adding the circumcentre x as a new vertex to the Delaunay tetrahedrization. However, if x encroaches upon any subsegments or subfacets, we do not add x and instead split the subsegments and subfacets.

`Rule 1` takes priority over `Rule 2`, and `Rule 2` takes priority over `Rule 3`. At the time we add a point on a facet, the prioritization guarantees that the boundary segments of the facet are subdivided by edges of the Delaunay tetrahedrization. Similarly, at the time we add a point in the interior of P, the boundary of P is subdivided by triangles in the Delaunay tetrahedrization. A point considered for addition to the Delaunay tetrahedrization has a *type*, which is the number of the rule that considers it or equivalently the dimension of the simplex it splits. Points of type 1 split subsegments and are always added once they are considered. Points of type 2 and 3 may be added or rejected.

Local density

Just as in two dimensions, the *local feature size* is crucial to understanding the Delaunay refinement algorithm. It is the function $f : \mathbb{R}^3 \to \mathbb{R}$ with $f(x)$ the radius of the smallest closed ball with centre x that intersects at least two disjoint faces of P. Note that f is bounded away from zero by some positive constant. It is easy to show that f satisfies the Lipschitz condition

$$f(x) \quad \leq \quad f(y) + \|x - y\|.$$

This implies that f is continuous over $\mathbb{R}^3$, but more than that, the condition says that f varies only slowly with x.

The local feature size is related to the *insertion radius* r_x of a point x, which is the length of the shortest Delaunay edge with endpoint x immediately after adding x. If x is a vertex of P then r_x is the distance to the nearest other vertex of P. If x is type 1 or 2 then r_x is the distance to the nearest encroaching vertex. If that encroaching vertex does not exist because it was rejected, then r_x is either half the length of the subsegment if x is type 1, or it is the circumradius of the subfacet if x is type 2. Finally, r_x is the circumradius of the tetrahedron it splits if x is type 3. We also define the insertion radius for a point that is considered for addition but rejected, because it encroaches upon subsegments or subfacets. This is done by hypothetically adding the point and taking the length of the shortest edge in the hypothetical star.

Radii and parents

Points are added in a sequence, and for each new point there are predecessors that we can make responsible for the addition. If x is type 1 or 2 then we define the responsible *parent* $p = p_x$ as the encroaching point that triggers the event. The point p may be a Delaunay vertex or a rejected circumcentre. If there are several encroaching points then p is the one closest to x. If x is type 3 then p is the most recently added endpoint of the shortest edge of the tetrahedron x splits.

Radius Claim. Let x be a vertex of D and p its parent, if it exists. Then $r_x \geq f(x)$ or $r_x \geq c \cdot r_p$, where $c = 1/\sqrt{2}$ if x is type 1 or 2 and $c = \varrho_0$ if x is type 3.

Proof. If x is a vertex of P then $f(x)$ is less than or equal to the distance to the nearest other vertex. This distance is $r_x \geq f(x)$. For the rest of the proof assume x is not a vertex of P. It therefore has a parent $p = p_x$. First consider the case where p is a vertex of P. If x is type 1 or 2, it lies in a segment or facet of P, and p is not contained in that segment or facet. Hence $r_x = \|x - p\| \geq f(x)$. If x is type 3 then the tetrahedron split by x has at least two vertices in P. Hence $r_x = \|x - p\| \geq f(x)$ as before. Secondly,

consider the case where p is not a vertex of P. If x is type 1 or 2 then p was rejected for triggering the insertion of x. Since p encroaches upon the subsegment or subfacet split by x, its distance to the closest vertex of that subsegment or subfacet is at most $\sqrt{2}$ times the distance of x from that same vertex. Hence $r_x \geq r_p/\sqrt{2}$. Finally, if x is type 3 then $r_p \leq L$, where L is the length of the shortest edge of the tetrahedron split by x. The algorithm splits that tetrahedron only if $R > L\varrho_0$. Hence $r_x = R > L\varrho_0 \geq \varrho_0 r_p$. □

Termination

The Radius Claim limits how quickly the insertion radius can decrease. We aim at choosing the only independent constant, which is ϱ_0, such that the insertion radii are bounded from below by a positive constant. Once this is achieved, we can prove termination of the algorithm using a standard packing argument. Figure 57 illustrates the possible parent–child relations between the three types of points added by the algorithm. We follow an arc of the digraph whenever the insertion radius of a point x is less than $f(x)$. The arc is labelled by the smallest possible factor relating the insertion radius of x to that of its parent. Note that there is no arc from type 1 to type 2 and there are no loops from type 1 back to type 1 and from type 2 back to type 2. This is because the angle constraint on the input polyhedron prevents parent–child relations for points on segments and facets with non-empty intersection. If there is a relation between points on segments and facets with empty intersection then $r_x \geq f(x)$ and there is no need to follow an arc in the digraph.

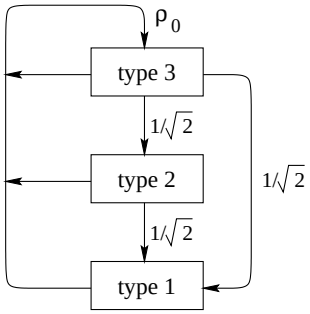

Fig. 57. The directed arcs indicate possible parent–child relations, and their labels give the worst case factors relating insertion radii

Observe that every cycle in the digraph contains the arc labelled ϱ_0 leading into type 3. We choose $\varrho_0 \geq 2$ to guarantee that the products of arc labels for all cycles are 1 or larger. The smallest product of any path in the digraph is therefore $\frac{1}{2}$. In cases where r_x is not at least $f(x)$, there exist ancestors q

with $r_x \geq r_q/2$ and $r_q \geq f(q)$. Since $f(q)$ is bounded away from zero by some positive constant, we conclude that the insertion radii cannot get arbitrarily small. It follows that the Delaunay refinement algorithm terminates. For $\varrho_0 < 2$ there are cases where the algorithm does not terminate.

Graded meshes

With little additional effort we can show that for ϱ_0 strictly larger than 2, insertion radii are directly related to local feature size, and not just indirectly through chains of ancestors. We begin with a relation between the local feature size over insertion radius ratio of a vertex and of its parent.

Ratio Claim. Let x be a Delaunay vertex with parent p and assume $r_x \geq c \cdot r_p$. Then

$$\frac{f(x)}{r_x} \leq 1 + \frac{f(p)}{c \cdot r_p}.$$

Proof. We have $r_x = \|x - p\|$ if p is a Delaunay vertex and $r_x \geq \|x - p\|$ if p is a rejected midpoint or circumcentre. Starting with the Lipschitz condition we get

$$\begin{aligned} f(x) &\leq f(p) + \|x - p\| \\ &\leq \frac{f(p)}{c \cdot r_p} \cdot r_x + r_x, \end{aligned}$$

and the result follows after dividing by r_x. □

To prepare the next step we assume $\varrho_0 > 2$ and define constants

$$\begin{aligned} C_1 &= \frac{(3+\sqrt{2}) \cdot \varrho_0}{\varrho_0 - 2}, \\ C_2 &= \frac{(1+\sqrt{2}) \cdot \varrho_0 + \sqrt{2}}{\varrho_0 - 2}, \\ C_3 &= \frac{\varrho_0 + 1 + \sqrt{2}}{\varrho_0 - 2}. \end{aligned}$$

Note that $C_1 > C_2 > C_3 > 1$.

Invariant. If x is a type i vertex in the Delaunay tetrahedrization, for $1 \leq i \leq 3$, then $r_x \geq f(x)/C_i$.

Proof. If the parent p of x is a vertex of the input polyhedron P then $r_x \geq f(x)$ and we are done. Otherwise, assume inductively that the claimed inequality holds for vertex p. We finish the proof by case analysis. If x is type 3 then $c = \varrho_0$ and $r_x \geq \varrho_0 \cdot r_p$ by the Radius Claim. By induction we

get $f(p) \leq C_1 r_p$, no matter what type p is. Using the Ratio Claim we get

$$\frac{f(x)}{r_x} \;\leq\; 1 + \frac{C_1}{\varrho_0} \;=\; C_3.$$

If x is type 2 then $c = \frac{1}{\sqrt{2}}$. We have $r_x \geq f(x)$ unless p is type 3, and therefore $f(p) \leq C_3 r_p$ by inductive assumption. Then $r_x \geq r_p/\sqrt{2}$ by the Radius Claim, and

$$\frac{f(x)}{r_x} \;\leq\; 1 + \sqrt{2} \cdot C_3 \;=\; C_2$$

by the Ratio Claim. If x is type 1 then $c = \frac{1}{\sqrt{2}}$. We have $r_x \geq f(x)$ unless p is type 2 or 3, and therefore $f(x) \leq C_2 r_p$ by inductive assumption. Then $r_x \leq 1 + r_p/\sqrt{2}$ by the Radius Claim, and

$$\frac{f(x)}{r_x} \;\leq\; 1 + \sqrt{2} \cdot C_2 \;=\; C_1$$

by the Ratio Claim. □

Because C_1 is the largest of the three constants, we can simplify the Invariant to $r_x \geq f(x)/C_1$ for every Delaunay vertex x. From this we conclude

$$\|x - y\| \;\geq\; \frac{f(x)}{1 + C_1}$$

for any two vertices x, y in the Delaunay tetrahedrization, using the argument in the proof of the Smallest Gap Lemma in Section 8.

Bibliographic notes

The bulk of the material in this section is taken from a paper by Jonathan Shewchuk (1998). In that paper, the assumed input is a so-called piecewise linear complex as defined by Miller et al. (1996). This is a 3-face of a polyhedron together with its faces, which is slightly more general than a three-dimensional polyhedron.

16. Sliver exudation

The sliver is the only type of small volume tetrahedron whose circumradius over shortest edge length ratio does not grow with decreasing volume. Experimental studies indicate that slivers frequently exist right between other well-shaped tetrahedra inside Delaunay tetrahedrizations. This section explains how point weights can be used to remove slivers.

Periodic meshes

Suppose S is a finite set of points in $\mathbb{R}^3$ whose Delaunay tetrahedrization has the ratio property for a constant ϱ_0. The goal is to prove that there are weights we can assign to the points such that the weighted Delaunay tetrahedrization is free of slivers. This cannot be true in full generality, for if S consists of only four points forming a sliver then no weight assignment can make that sliver disappear. We avoid this and similar boundary effects by replacing the finite by a periodic set $S = P + \mathbb{Z}^3$, where P is a finite set of points in the half-open unit cube $[0,1)^3$ and $\mathbb{Z}^3$ is the three-dimensional integer grid. The periodic set S contains all points $p + \mathbf{v}$, where $p \in P$ and $\mathbf{v}$ is an integer vector. Like S, the Delaunay tetrahedrization D of S is periodic. Specifically, for every tetrahedron $\tau \in D$, the shifted copies $\tau + \mathbb{Z}^3$ are also in D. This idea is illustrated for a periodic set generated by four points in the half-open unit square in Figure 58.

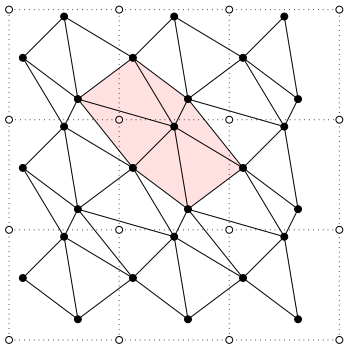

Fig. 58. Periodic tiling of the plane. The shaded triangles form a domain whose shifted copies tile the entire plane

Weight assignment

A *weight assignment* is a function $\omega : P \to \mathbb{R}$. The resulting set of spheres is denoted as $S_\omega = \{(a, \omega(p)) : p \in P, a \in p + \mathbb{Z}^3\}$. Depending on ω, a point p may or may not be a vertex of the weighted Delaunay triangulation of S_ω, which we denote as D_ω. Let $N(p)$ be the minimum distance to any other point in S. To prevent points from becoming redundant, we limit ourselves to *mild* weight assignments that satisfy $0 \leq \omega(p) < \frac{1}{3}N(p)$ for all $p \in P$. Every sphere in S_ω has a real radius and every pair is disjoint and not nested. It follows that none of the points is redundant. Another benefit of a mild weight assignment is that it does not drastically change the shape of triangles and tetrahedra. In particular, D_ω has the ratio property for a

constant ϱ_1 that only depends on ϱ_0. It follows that the area of each triangle is bounded from below by some constant times the square of its circumcircle. The same is not true for volumes of tetrahedra, which is why eliminating slivers is difficult.

A crucial step towards eliminating slivers is a generalization of the Degree Lemma of Section 14. Let K be the set of simplices that occur in weighted Delaunay tetrahedrizations for mild weight assignments of S. In other words, $K = \bigcup_\omega D_\omega$, which is a three-dimensional simplicial complex but not necessarily geometrically realized in $\mathbb{R}^3$. The vertex set of K is $\operatorname{Vert} K = S$, and the *degree* of a vertex is the number of edges in K that share the vertex.

Weighted Degree Lemma. There exists a constant δ_1 depending only on ϱ_0 such that the degree of every vertex in K is at most δ_1.

The proof is fairly tedious and partially a repeat of the proofs of the Length Variation and Degree Lemmas of Section 14. It is therefore omitted.

Slicing orthogonal spheres

We need an elementary fact about spheres (a, A^2) and (z, Z^2) that are orthogonal, that is, $\|a - z\|^2 = A^2 + Z^2$. A plane intersects the two spheres in two circles, which may have real or imaginary radii.

Slicing Lemma. A plane passing through a intersects the two spheres in two orthogonal circles.

Proof. Let $(x, X^2), (y, Y^2)$ be the circles where the plane intersects the two spheres. We have $x = a$, $X^2 = A^2$, and $Y^2 = Z^2 - \|z - y\|^2$. Hence

$$\begin{aligned} \|x - y\|^2 &= \|x - z\|^2 - \|z - y\|^2 \\ &= (A^2 + Z^2) - (Z^2 - Y^2) \\ &= X^2 + Y^2. \end{aligned}$$

In words, the two circles are also orthogonal. □

As an application of the Slicing Lemma consider three spheres and the plane that passes through their centres, as in Figure 59. The plane intersects the three spheres in three circles, and there is a unique circle orthogonal to all three. The Slicing Lemma implies that every sphere orthogonal to all three spheres intersects the plane in this same circle.

Variation of orthoradius

Another crucial step towards eliminating slivers is the stability analysis of their orthospheres. We will see that a small weight change can increase the size of the orthosphere dramatically. This is useful because a tetrahedron in

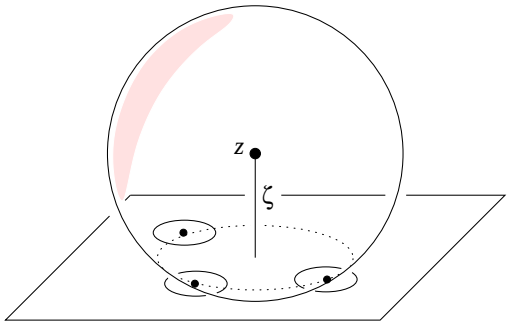

Fig. 59. Slice through three spheres and another sphere orthogonal to the first three

D_ω cannot have a large orthosphere, for else that orthosphere would be closer than orthogonal to some weighted point. We later exploit this observation and change weights to increase orthospheres of slivers.

Let us analyse how the radius of the orthosphere of four spheres changes as we manipulate the weight of one of the sphere. Let (y, Y^2) be the smallest sphere orthogonal to the first three spheres, let (p, P^2) be the fourth sphere, and let (z, Z^2) be the orthosphere of all four spheres, as illustrated in Figure 60. Let ζ and ϕ be the distances of z and p from the plane h that passes through the centres of the first three spheres. With varying P^2, the centre of the orthosphere moves along the line that meets h orthogonally at y. The distance of z from h is a function of the weight of p, $\zeta : \mathbb{R} \to \mathbb{R}$.

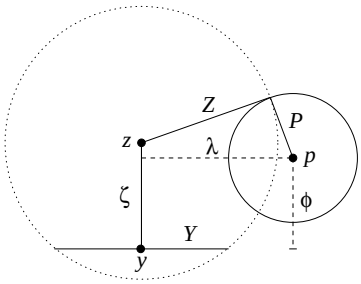

Fig. 60. The orthocentre z moves downward as the weight of p increases

Distance Variation Lemma. $\zeta(P^2) = \zeta(0) - \frac{P^2}{2\phi}$.

Proof. Let λ be the distance from p to the line along which z moves. We have $Z^2 + P^2 = (\zeta(P^2) - \phi)^2 + \lambda^2$. The weight of the orthosphere is

$Z^2 = \zeta(P^2)^2 + Y^2$. Hence

$$\begin{aligned} \zeta(P^2)^2 &= Z^2 - Y^2 \\ &= (\zeta(P^2) - \phi)^2 + \lambda^2 - P^2 - Y^2. \end{aligned}$$

After cancelling $\zeta(P^2)^2$ we get

$$\zeta(P^2) = \frac{\phi^2 + \lambda^2 - Y^2}{2\phi} - \frac{P^2}{2\phi}.$$

The first term on the right-hand side is $\zeta(0)$. □

The term $P^2/2\phi$ is the displacement of the orthocentre that occurs as we change the weight of p from 0 to P^2. For slivers, the value of ϕ is small which implies that the displacement is large.

Sliver theorem

We finally show that there is a mild weight assignment that removes all slivers. The proof is constructive and assigns weights in sequence to the points in P. To quantify the property of being a sliver, we define $\xi = V/L$, where V is the volume and L is the length of the shortest edge of the tetrahedron. Only slivers can have bounded R/L as well as small ξ. Note that the volume of the tetrahedron indicated in Figure 60 is one-third the area of the base triangle times ϕ. As mentioned above, the area of the base triangle is some positive constant fraction Y^2. Similarly, L is some positive constant fraction of Y, which implies that ξ is some positive constant fraction of $Y\phi$.

Sliver Theorem. There are constants $\varrho_1, \xi_0 > 0$ and a mild weight assignment ω, such that the weighted Delaunay tetrahedrization has the ratio property for ϱ_1 and $\xi > \xi_0$ for all its tetrahedra.

Proof. We focus on proving $\xi > \xi_0$ for all tetrahedra in D_ω. Assume without loss of generality that the distance from a point p to its nearest neighbour in S is $N(p) = 1$. The weight assigned to p can be anywhere in the interval $[0, \frac{1}{3}]$. According to the Weighted Degree Lemma, there is only a constant number of tetrahedra that can possibly be in the star of p. Each such tetrahedron can exist in D_ω only if its orthosphere is not too big. In other words, the tetrahedron can only exist if $\omega(p)$ is chosen inside some subinterval of $[0, \frac{1}{3}]$. The Distance Variation Lemma implies that the length of this subinterval decreases linearly with ϕ and therefore linearly with ξ. We can choose ξ_0 small enough such that the constant number of subintervals cannot possibly cover $[0, \frac{1}{3}]$. By the pigeonhole principle, there is a value $\omega(p) \in [0, \frac{1}{3}]$ that excludes all slivers from the star of p. □

Removing slivers

The proof of the Sliver Theorem suggests an algorithm that assigns weights to individual points in an arbitrary sequence. For each point $p \in P$, the algorithm considers the interval of possible weights and the subintervals in which tetrahedra in K can occur in the weighted Delaunay tetrahedrization. We could consider all tetrahedra in the star of p in K, but it is more convenient to consider only the subset in the 1-parameter family of weighted Delaunay tetrahedrizations generated by continuously increasing the weight of p from 0 through $\frac{1}{3}N(p)$. For each such tetrahedron, we get the ξ value and a subinterval during which it exists in D_ω. Figure 61 draws each tetrahedron as a horizontal line segment in the $\omega\xi$-plane. The lower envelope of the line segments is the function that maps the weight of p to the worst ξ value of any tetrahedron in its star. The algorithm finds the weight where that function has a maximum and assigns it to p. Since there is only a constant number of tetrahedra to be considered, this can be accomplished in constant time. The overall running time of the algorithm is therefore $O(n)$, where $n = \operatorname{card} P$.

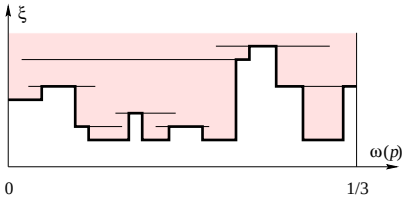

Fig. 61. Each tetrahedron in the star is represented by a horizontal line segment

A source of possible worry is that, after we have fixed the weight of p we may modify the weight of a neighbour q of p. Modifying the weight of q may change the star of p. However, all new tetrahedra in the star of p also belong to the star of q and thus cannot have arbitrarily small ξ values. We thus do not have to reconsider p, and $O(n)$ time indeed suffices. The Sliver Theorem guarantees the algorithm is successful as quantified by the positive constant ξ_0. While the algorithm does not find the globally optimum weight assignment, it finds the optimum for each point individually, assuming fixed weights of other points. It might therefore achieve a minimum ξ value that is much better than the rather pessimistic estimate for ξ_0 guaranteed by the Sliver Theorem.

Bibliographic notes

The material of this section is taken from the sliver exudation paper by Cheng et al. (1999). The occurrence of slivers as a menace in three-dimensional Delaunay tetrahedrizations was reported by Cavendish, Field and

Frey (1985). Besides the sliver exudation method described in this section, there are two other methods that provably remove slivers. The first by Chew (1997) adds points and uses randomness to avoid creating new slivers. The second by Edelsbrunner et al. (1999) moves points and relies on the ratio property of the Delaunay tetrahedrization, as in the weight assignment method of this section.

REFERENCES

F. Aurenhammer (1991), Voronoi diagrams: a study of a fundamental geometric data structure, *ACM Comput. Surveys* **23**, 345–405.

M. Bern and D. Eppstein (1992), Mesh generation and optimal triangulations, in *Computing in Euclidean Geometry*, Vol. 1 (D.-Z. Du and F. K. Hwang, eds), World Scientific, Singapore, pp. 23–90.

M. Bern, D. Eppstein and J. Gilbert (1994), Provably good mesh generation, *J. Comput. Syst. Sci.* **48**, 384–409.

L. J. Billera and B. Sturmfels (1992), Fiber polytopes, *Ann. Math.* **135**, 527–549.

J. C. Cavendish, D. A. Field and W. H. Frey (1985), An approach to automatic three-dimensional finite element mesh generation, *Internat. J. Numer. Methods Engrg* **21**, 329–347.

B. Chazelle (1984), Convex partitions of polyhedra: a lower bound and worst case algorithm, *SIAM J. Comput.* **13**, 488–507.

S.-W. Cheng, T. K. Dey, H. Edelsbrunner, M. A. Facello and S.-H. Teng (1999), Sliver exudation, in *Proc. 15th Ann. Sympos. Comput. Geom., 1999*, ACM, pp. 1–14.

L. P. Chew (1987), Constrained Delaunay triangulations, in *Proc. 3rd Ann. Sympos. Comput. Geom., 1987*, ACM, pp. 215–222.

L. P. Chew (1989), Guaranteed-quality triangular meshes, Report TR-98-983, Comput. Sci. Dept., Cornell Univ., Ithaca, NY.

L. P. Chew (1993), Guaranteed-quality mesh generation for curved surfaces, in *Proc. 9th Ann. Sympos. Comput. Geom., 1993*, ACM, pp. 274–280.

L. P. Chew (1997), Guaranteed-quality Delaunay meshing in 3D, in *Proc. 13th Ann. Sympos. Comput. Geom., 1997*, ACM, pp. 391–393.

K. L. Clarkson and P. W. Shor (1989), Applications of random sampling in computational geometry, *Discrete Comput. Geom.* **4**, 387–421.

G. B. Danzig (1963), *Linear Programming and Extensions*, Princeton Univ. Press, Princeton, NJ.

M. de Berg, M. van Kreveld, M. Overmars and O. Schwarzkopf (1997), *Computational Geometry: Algorithms and Applications*, Springer, Berlin.

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

P. G. L. Dirichlet (1850), Über die Reduktion der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen, *J. Reine Angew. Math.* **40**, 209–227.

H. Edelsbrunner (1987), *Algorithms in Combinatorial Geometry*, Springer, Heidelberg.

H. Edelsbrunner (1990), An acyclicity theorem for cell complexes in d dimensions, *Combinatorica* **10**, 251–260.

H. Edelsbrunner (1995), Algebraic decomposition of non-convex polyhedra, in *Proc. 36th Ann. IEEE Sympos. Found. Comput. Sci. 1995*, pp. 248–257.

H. Edelsbrunner, X.-Y. Li, G. L. Miller, A. Stathopoulos, D. Talmor, S.-H. Teng, A. Üngör and N. Walkington (1999), Smoothing cleans up slivers, Manuscript.

H. Edelsbrunner and E. P. Mücke (1990), Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms, *ACM Trans. Graphics* **9**, 66–104.

H. Edelsbrunner and N. R. Shah (1996), Incremental topological flipping works for regular triangulations, *Algorithmica* **15**, 223–241.

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

I. Emiris and J. Canny (1995), A general approach to removing geometric degeneracies, *SIAM J. Comput.* **24**, 650–664.

P. Erdős (1979), Combinatorial problems in geometry and number theory, *Proc. Sympos. Pure Math.* **34**, 149–162.

S. Fortune and C. J. Van Wyk (1996), Static analysis yields efficient exact integer arithmetic for computational geometry, *ACM Trans. Graphics* **15**, 223–248.

I. M. Gelfand, M. M. Kapranov and A. V. Zelevinsky (1994), *Discriminants, Resultants and Multidimensional Determinants*, Birkhäuser, Boston.

J. E. Goodman and J. O'Rourke, eds (1997) *Handbook of Discrete and Computational Geometry*, CRC Press, Boca Raton, FL.

P. M. Gruber and C. G. Lekkerkerker (1987), *Geometry of Numbers*, 2nd edn, North-Holland, Amsterdam.

B. Grünbaum (1967), *Convex Polytopes*, Wiley, London.

B. Grünbaum and G. C. Shephard (1994), A new look at Euler's theorem for polyhedra, *Amer. Math. Monthly* **101**, 109–128.

L. J. Guibas, D. E. Knuth and M. Sharir (1992), Randomized incremental construction of Delaunay and Voronoi diagrams, *Algorithmica* **7**, 381–413.

H. Hadwiger (1957), *Vorlesungen über Inhalt, Oberfläche und Isoperimetrie*, Springer, Berlin.

E. Helly (1923), Über Mengen konvexer Körper mit gemeinschaftlichen Punkten, *Jahresber. Deutsch. Math.-Verein.* **32**, 175–176.

E. Helly (1930), Über Systeme von abgeschlossenen Mengen mit gemeinschaftlichen Punkten, *Monatsh. Math. Physik* **37**, 281–302.

B. Joe (1989), Three-dimensional triangulations from local transformations, *SIAM J. Sci. Statist. Comput.* **10**, 718–741.

B. Joe (1991), Construction of three-dimensional Delaunay triangulations from local transformations, *Comput. Aided Geom. Design* **8**, 123–142.

R. Klein (1997), *Algorithmische Geometrie*, Addison-Wesley, Bonn.

C. L. Lawson (1977), Software for C^1 surface interpolation, in *Mathematical Software III*, Academic Press, New York, pp. 161–194.

C. L. Lawson (1986), Properties of n-dimensional triangulations, *Computer Aided Geometric Design* **3**, 231–246.

D. T. Lee and A. K. Lin (1986), Generalized Delaunay triangulations for planar graphs, *Discrete Comput. Geom.* **1**, 201–217.

N. J. Lennes (1911), Theorems on the simple finite polygon and polyhedron, *Amer. J. Math.* **33**, 37–62.

N. Max, P. Hanrahan and R. Crawfis (1990), Area and volume coherence for efficient visualization of 3D scalar functions, in *San Diego Workshop on Volume Visualization*, published in *Computer Graphics* **24**, 27–33.

D. Michelucci (1995), An ε-arithmetic for removing degeneracies, in *Proc. IEEE Sympos. Comput. Arithmetic, 1995.*

G. L. Miller, D. Talmor, S.-H. Teng and N. Walkington (1995), A Delaunay based numerical method for three dimensions: generation, formulation, and partition, in *Proc. 27th Ann. ACM Sympos. Theory Comput., 1995*, pp. 683–692.

G. L. Miller, D. Talmor, S.-H. Teng, N. Walkington and H. Wang (1996), Control volume meshes using sphere packing: generation, refinement and coarsening, in *Proc. 5th Internat. Meshing Roundtable, 1996*, pp. 47–61.

K. Mulmuley (1994), *Computational Geometry: An Introduction Through Randomized Algorithms*, Prentice-Hall, Englewood Cliffs, NJ.

J. Nievergelt and F. P. Preparata (1982), Plane-sweep algorithms for intersecting geometric figures, *Comm. ACM* **25**, 739–747.

A. Okabe, B. Boots and K. Sugihara (1992), *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, Wiley, Chichester.

J. O'Rourke (1987), *Art Gallery Theorems and Algorithms*, Oxford Univ. Press, New York.

J. O'Rourke (1994), *Computational Geometry in C*, Cambridge Univ. Press, Cambridge.

J. Pach and P. K. Agarwal (1995), *Combinatorial Geometry*, Wiley-Interscience, New York.

F. P. Preparata and M. I. Shamos (1985), *Computational Geometry: An Introduction*, Springer, New York.

J. Radon (1921), Mengen konvexer Körper, die einen gemeinschaftlichen Punkt enthalten, *Math. Ann.* **83**, 113–115.

J. Ruppert (1992), A new and simple algorithm for quality 2-dimensional mesh generation, Report UCB/CSD 92/694, Comput. Sci. Div., Univ. California, Berkeley, CA.

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

J. Ruppert and R. Seidel (1992), On the difficulty of triangulating three-dimensional non-convex polyhedra, *Discrete Comput. Geom.* **7**, 227–254.

E. Schönhardt (1928), Über die Zerlegung von Dreieckspolyedern in Tetraeder, *Math. Ann.* **98**, 309–312.

L. L. Schumaker (1987), Triangulation methods, in *Topics in Multivariate Approximation*, (C. K. Choi, L. L. Schumaker and F. I. Utreras, eds), Academic Press, pp. 219–232.

R. Seidel (1988), Constrained Delaunay triangulations and Voronoi diagrams with obstacles, in *1978–1988 Ten Years IIG*, pp. 178–191.

R. Seidel (1993), Backwards analysis of randomized geometric algorithms, in *New Trends in Discrete and Computational Geometry*, (J. Pach, ed.), Springer, Berlin, pp. 37–67.

R. Seidel (1998), The nature and meaning of perturbations in geometric computing, *Discrete Comput. Geom.* **19**, 1–18.

M. I. Shamos (1975), Geometric complexity, in *Proc. 7th Ann. ACM Sympos. Theory Comput., 1975*, pp. 224–233.

M. I. Shamos and D. Hoey (1975), Closest-point problems, in *Proc. 16th Ann. IEEE Sympos. Found. Comput. Sci., 1975*, pp. 151–162.

M. I. Shamos and D. Hoey (1976), Geometric intersection problems, in *Proc. 17th Ann. IEEE Sympos. Found. Comput. Sci., 1976*, pp. 208–215.

J. R. Shewchuk (1998), Tetrahedral mesh generation by Delaunay refinement, in *Proc. 14th Ann. Sympos. Comput. Geom., 1998*, ACM, pp. 86–95.

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

D. Talmor (1997), Well-spaced points for numerical methods, Report CMU-CS-97-164, Dept. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, PA.

G. Voronoi (1907/08), Nouvelles applications des paramètres continus à la théorie des formes quadratiques, *J. Reine Angew. Math.* **133** (1907), 97–178, and **134** (1908), 198–287.

P. L. Williams (1992), Visibility ordering meshed polyhedra, *ACM Trans. Graphics* **11**, 103–126.

C. K. Yap (1990), Symbolic treatment of geometric degeneracies, *J. Symbolic Comput.* **10**, 349–370.

G. M. Ziegler (1995), *Lectures on Polytopes*, Springer, New York.