28. Delaunay triangulations

We consider decompositions of a convex polygonal region into triangles, using a prescribed set of points as vertices of the triangles.

- We motivate the issue through numerical integration (kept to its simplest form).
- In any given example, many different such "triangulations" exist, related to each other by sequences of local transformations.
- The Delaunay condition describes triangulations that avoid thin triangles.
- Going back to applications, we explain how one can use this to think of finite point sets in the plane as tracing out a geometric shape.

(28a) Numerical integration. Take a function of one variable, $f(x)$. We want an approximate formula for $\int_a^b f(x)\,dx$, based only on knowing finitely many values $f(x_1),\dots,f(x_n)$, where $a = x_1 < x_2 < \cdots < x_n = b$. The simplest solution is the *trapezoid rule*

$$\int_a^b f(x)\,dx \approx \tfrac{1}{2}(f(x_2)+f(x_1))(x_2-x_1) + \tfrac{1}{2}(f(x_3)+f(x_2))(x_3-x_2) + \cdots . \tag{28.1}$$

You have probably seen this before, at least in the case where $x_1,\dots,x_n$ are equally spaced. Now, let's look at the corresponding problem for functions of two variables.

(28.2) We are given points $v_1 = (x_1,y_1),\dots,v_n = (x_n,y_n)$; no two are equal, and they do not all lie on the same straight line.

Definition 28.1. *The* convex hull *of* $(v_1,\dots,v_n)$ *is the smallest convex polygon P which contains all those points. (The vertices of P will be a subset of the v_i.)*

For functions $f(x,y)$ defined on the convex hull P, we want an approximate formula for $\int_P f$ in terms of the values $f(v_i)$. It's easy to find such a formula if, say, P is a rectangle and the v_i form a grid; but that may not be true in applications. One way to approach this is to decompose P into triangles. More precisely:

Definition 28.2. *A triangulation of P, with vertices $(v_1,\dots,v_n)$, is a decomposition into non-overlapping triangles, such that all the v_i, and no other points, appear as vertices of those triangles.*

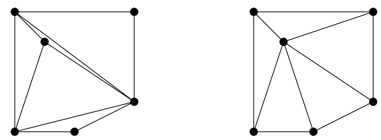
Given such a triangulation, the analog of the trapezoid rule is:

$$\int_P f(x,y)\,dx\,dy \approx \sum_T \mathrm{area}(T)(\text{average value of } f \text{ at the three vertices of } T). \tag{28.3}$$

Here, the sum is over the triangles in the triangulation. The vertices of each triangle belong to our $(v_1,\dots,v_n)$, so the overall formula is a weighted sum of $f(v_i)$. Different choices of triangulations give different approximate answers, some better than others.

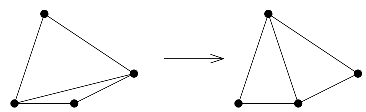
Example 28.3. *Take the points* $(x, y) = (0, 0)$, $(2, 0)$, $(4, 1)$, $(0, 4)$, $(1, 3)$, $(4, 4)$. *In the picture below, the triangulation on the left has triangles that are very long and thin, and we suspect that it's not a good choice. The triangulation on the right looks better in that respect:*

(28.4)



(28b) Different triangulations. We can change a triangulation by a *flip*, applied to a pair of neighbouring triangles which form a convex quadrilateral:

(28.5)

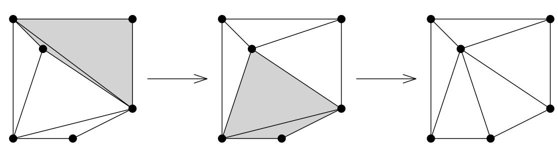


Some topological facts:

- Any finite set of points admits a triangulation (boring).
- Any two triangulations of the same set of points have the same number of triangles (interesting).
- Any two triangulations of the same point set can be related by a sequence of flips (even more interesting).

Example 28.4. *One can get from one triangulation in* (28.4) *to the other by two flips:*

(28.6)



Definition 28.5. *A triangulation is* Delaunay *if, when we take the circumcircle of any triangle in it (the unique circle going through its vertices), no point of our set lies inside that circle. To clarify: "inside" means in the interior. It is ok for a Delaunay triangulation if more points of our set lie on the circumcircle itself.*

The key property is:

Theorem 28.6. *For every finite set of points as in* (28.2)*, there is a Delaunay triangulation.*

For instance, the triangulation on the right in (28.4) is Delaunay, but that on the left isn't. Moreover, there is an algorithm which, starting from any triangulation, produces a Delaunay triangulation in finitely many flip steps. Namely, suppose that we have two adjacent triangles which together form a convex quadrilateral, and which by themselves (as a triangulation of that quadrilateral, forgetting all the other points) fail to obey the Delaunay condition. Then, we flip; and repeat that until that's no longer possible. Why does this work, and not, for instance, cycle endlessly?

LEMMA 28.7. *Suppose we have two adjacent triangles which form a convex quadrilateral and, by themselves, are not Delaunay. Apply a flip. Then, the new triangulation gives an approximate formula for* $\int_P x^2 + y^2$ *which is less than that for the original triangulation.*

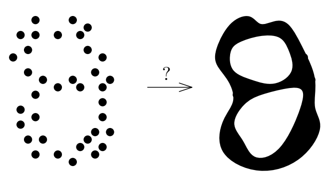
This is a small nifty piece of geometry, which we won't explain here. Given that, the flip algorithm can never cycle back to a previous choice of triangulation; and because there are only finitely many possible triangulation of our given point set, it must eventually end in a situation where no further such moves are available. This means that for any two adjacent triangles which form a convex quadrilateral, Delaunay holds. By a further geometric argument, it then follows that the entire triangulation is Delaunay. Next, what can we say abou how many Delaunay triangulations a fixed set of points can have?

THEOREM 28.8. *Suppose that T is a triangle whose vertices belong to our point set, and with the following property (which is stronger than what's in the definition of Delaunary triangulation): all the other points in our finite set lie outside (in the exterior of) the circumcircle of the triangle. Then, T occurs in every possible Delaunay triangulation.*

In particular, if no four points in our set lie on the same circle, the Delaunay triangulation is unique (because then, the Theorem applies to any triangle in it).

(28c) The topology of data. Suppose that we have a finite set of points in the plane, which are the result of some measurement or sampling process. I would like assign an overall shape to this "point cloud", as if looking at it with my glasses off:

(28.7)



There are many ways of doing this, all depending on a choice of scale $\sigma > 0$ to do the blurring. Let's say that we want a computational (polygonal) flavour. Here's a particularly simple approach. First, form the Delaunay triangulation (let's assume for simplicity that there's a unique one). Draw the original point set, together with all the edges in the triangulation which are of length $< \sigma$, and finally those triangles from our triangulations all of whose edges have length $< \sigma$. Let's call the union of all that the *shape complex* of the point set, at scale σ. If we take σ small (smaller than the distance between any two points), the shape complex just consists of the original points. If we take σ large (larger than any of the distances), we are being told to add all edges and all triangles, so the outcome is just the convex hull P. Obviously, the right choice of scale (somewhere between those two extremes) is important, in order for the outcome to be meaningful.

Example 28.9. *Take our running example of a Delaunay triangulation, and form the shape complex at various scales:*

(28.8)