Recall: Algorithm for Delaunay triangulations

Edge Flip of illegal edge

(p, q) is an illegal edge.  
(r, s) is a legal edge.

Lemma. If an edge is Delaunay then it is legal. Converse not true, but if ALL edges are legal then the triangulation is Delaunay.
[Randomized] Incremental Del. triang. alg. \(O(n \log n)\)

Add points one by one
To add point \(X\):

Find current triangle \(ABC\) containing \(X\)

\[\begin{array}{c}
\text{A} \\
\text{X} \\
\text{C}
\end{array} \quad \Rightarrow \quad \begin{array}{c}
\text{A} \\
\text{X} \\
\text{C}
\end{array} \quad \begin{array}{c}
\text{B} \\
\text{X}
\end{array} \quad \begin{array}{c}
\text{A} \\
\text{B} \\
\text{C}
\end{array}\]

Join \(X\) to \(A, B, C\)

Then flip edges until Delaunay

Note: edges \(AX, BX, CX\) are Delaunay because there was an empty circle \((A,B,C)\) — can shrink to empty circle thru \(AX\), etc.
issues, details

1. What if $X$ is outside current CH?
   Start by adding a very large $\triangle$ around all points. Large enough s.t.
   1. $q_0$ outside every circle through $3$ input points
   2. $q_i$ outside every circle through $3$ input points

- then find Del. triang.
- then delete $q_0$, $q_i$, $q_2$

Claim: We get Del. triang. of original points.
How to limit testing for illegal edges:

Test \((A, B, x)\)  Test \((B, C, x)\)  Test \((C, A, x)\)

where

Test \((u, v, x)\) — recursive subroutine
- if \(uv\) not on \(CH\)
  - \(w\) — vertex of other \(\Delta\) on \(uv\)
  - if \(x\) in Circle \((uvw)\)
    - swap \(uv\) for \(xw\)
    - Test \((u, w, x)\)  Test \((v, w, x)\)
Changes produced by this test update:

Recall

All the new edges will be incident to X.
**Triangulations.** Recall what we’ve seen:

- Delaunay triangulation of point set in $\mathbb{R}^d$, $O(n \log n)$ algorithm in $\mathbb{R}^2$.

- $O(n)$ algorithm to triangulate any polygon in $\mathbb{R}^2$ (hard)

[Diagram of two triangulations of a point set]

[Link: https://en.wikipedia.org/wiki/Point_set_triangulation]
Triangulations. Applications and criteria

- angle criteria - for meshing

- length criteria: minimum weight triangulation

- constrained triangulations (when certain edge must be included)

- meshing - triangulations with Steiner points

- flip distance

- morphing

- curve and surface reconstruction
Angle conditions for triangulations

The motivation is meshing for finite element methods (more on this later) where small and large angles are bad.

![bad triangles]

Problems:

1. given a point set, find a triangulation that maximizes the min. angle
   The Delaunay triangulation does this.

2. given a point set, find a triangulation that minimizes the max. angle

EX. Show that these two can be different.
There is a poly time algorithm to find a triangulation that minimizes the maximum angle. *Note: I mis-stated this as NP-hard last day.*

It uses a solution for the case of triangulating a polygon (via dynamic programming).

Length conditions: Minimum weight triangulation

Given a point set find a triangulation that minimizes the sum of the lengths of the edges.

Solved by dynamic programming for triangulations of a simple polygon.

For point sets, proved NP-hard in 2008 (had been open since 1979).
Note: not known to be in NP because of square root computations.


Approximations

- approximation ratio of Delaunay triangulation: Theta(n)
- approximation ratio of greedy triangulation (add edges in order of weight): Theta(sqrt(n))

- quasi-poly time approximation scheme:

the Mulzer-Rote result shows unlikely to be FPTAS

but approximation ratio $1 + \varepsilon$ in quasi-polynomial time $\exp(O((\log n)^t))$

heuristics based on polygon result (find edges that will be in any min weight triang. If they form polygons, ok.)
Constrained triangulations

Given points $P$ in $\mathbb{R}^2$ and some non-crossing edges $F$, add more edges to get a triangulation optimizing some criterion.

This generalizes polygon triangulation (though note that the above problem asks to triangulate the inside AND the outside of a polygon).
The **Constrained Delaunay triangulation (CDT)** consists of triangles $abc$ not crossed by any edge of $F$ s.t. $\text{Circle}(a,b,c)$ contains no point of $P$ visible from inside triangle $abc$.

Must be proved that these triangles form a triangulation.
Examples of Constrained Delaunay triangulations
Most results for Delaunay triangulations carry over to Constrained Delaunay triangulations.

The edge empty circle condition carries over:
(a,b) is an edge of the Constrained Delaunay triangulation iff no edge of F crosses (a,b) and there is a circle through a and b that does not contain any point p in P visible to a point on edge (a,b).

Edge flipping carries over.

There is an $O(n \log n)$ time algorithm to compute the Constrained Delaunay triangulation.

The Constrained Delaunay triangulation maximizes the min angle (among all constrained triangulations).
Triangulations for finite element methods

Example problem: find how a solid body deforms under stress. Requires solving partial differential equations, which is done by approximating on a “mesh” (often a triangulation)

Meshing. Given a region of $\mathbb{R}^2$ with a polygonal boundary, subdivide it into disjoint triangles meeting edge-to-edge and conforming to the boundary, i.e. every boundary edge is a union of triangle edges. Use “nicely shaped” triangles.

Note: can add new points called “Steiner points”
**Meshing.** Given a region of $\mathbb{R}^2$ with a polygonal boundary, subdivide it into disjoint triangles meeting edge-to-edge and *conforming* to the boundary, i.e. every boundary edge is a union of triangle edges. Use “nicely shaped” triangles.

We concentrate on unstructured meshes.


[PDF] berkeley.edu

Theoretically Guaranteed Delaunay Mesh Generation—In Practice (slides)

[PDF] (color, 2003k, 106 pages) - EECS at UC Berkeley
bad for finite element methods:

small angle
"needle"

large angle
"cap"

in 3D also
"sliver"

4 points
almost
on equator
of sphere

Also want as few triangles as possible, but this conflicts with angle constraints.
Well–Shaped Elements vs. Few Elements
somewhat contradictory goals

These meshes generated by Ruppert’s Delaunay refinement algorithm.

Jonathan Shewchuk
Delaunay refinement algorithm

Start with Delaunay triangulation and add more points to improve angles.

kill skinny triangle by adding point v at center of circumcircle

A related puzzle problem (Martin Gardner)

Given a square or an obtuse triangle, dissect into smallest number of acute triangles.

Solution with 14 triangles
A related puzzle problem (Martin Gardner)

Given a square or an obtuse triangle, dissect into smallest number of acute triangles.

\[
\text{min is 8}
\]

\[
\text{min is 7}
\]