Recall

**Voronoi diagram**

Given points \( P = \{p_1, \ldots, p_n\} \) in the plane, the **Voronoi region** of \( p_i \) is

\[
V(p_i) = \{ x \in \mathbb{R}^2 : d(x, p_i) \leq d(x, p_j) \forall j \neq i \}
\]

\( p_i \) is called a **site**.

The **Voronoi diagram** \( \mathcal{V}(P) \) consists of all the Voronoi regions.

Given points \( P = \{p_1, \ldots, p_n\} \) in the plane, the **Delaunay triangulation** \( \mathcal{D}(P) \) is a graph with vertices \( p_1, \ldots, p_n \) and edge \((p_i, p_j)\) iff \( V(p_i) \) and \( V(p_j) \) share an edge.

\( \mathcal{D}(P) \) is the **planar dual** of \( \mathcal{V}(P) \)
Recall Delaunay triangulation and empty circle property: \((p,q)\) is an edge of the Delaunay triangulation iff there is an empty circle through \(p\) and \(q\).

An algorithmically more useful characterization:

**Lemma.** A triangulation is Delaunay iff every edge \(e=(p,q)\) is legal.

**Definition.** edge \(e=(p,q)\) is **legal** if either:
- \(e\) is on the convex hull or
- \(e\) is interior with triangles \(pqr\) and \(pqs\), and \(r\) is not in \(\text{Circle}(pqs)\)

**Note:** \(r\) in \(\text{Circle}(p,q,s)\) iff \(s\) in \(\text{Circle}(p,q,r)\)

Note that this is a condition about ALL edges, not a single edge:
edge \(e\) is Delaunay (∃ an empty circle through its endpoints) \(\Rightarrow\) \(e\) is legal \(\Leftrightarrow\)
Ingredients to prove the lemma.

There is one degree of freedom for circles through p and q.

**Thales Theorem.** For pq a chord of a circle, angle psq is constant for s on an arc of the circle. For s inside, the angle is bigger. For s outside the angle is smaller.

Actually, Thales considered pq to be a diameter. The generalization is in Euclid.

Lemma. A triangulation is Delaunay iff every edge $e=(p,q)$ is legal.

Proof. contrapositive

A triangulation is NOT Delaunay iff there is an illegal edge.

If there is an illegal edge $p,q$ has no empty circle.

$	herefore$ not Delaunay.

Then there exists a triangle $pqs$ and site $x$.

Choose $pqs$ and $x$ to maximize $\angle pxq$.

Let $r$ be vertex of triangle on other side from $pqs$.

If $r$ is in Circle $(pqs)$, then $(p,q)$ is illegal.

Note $x$ not in $\triangle pqr$. Suppose $x$ outside edge $rp$.

Consider $pqr$ and point $x$.

$x \in \text{Circle}(pqr)$

and $\angle pxr > \angle pxq$. Contradiction to choice of $pqs$ and $x$.

Where did we use Thales? -- we didn't! though this pf. is straight from [CGAA]. We will use Thales later.
What to do with an illegal edge \((p, q)\)

**Edge Flip**

![Diagram showing edge flip](image)

**Claim:** \((r, s)\) is a legal edge.

Change Circle \((pqrs)\) to Circle \((prs)\);
It shrinks and \(q\) leaves the circle.
Flipping illegal edges makes global improvements in a triangulation:  
the **Angle Vector**.

For any triangulation \( T \) of a set of points, the **angle vector** \( A(T) \) is the list of angles of the triangles sorted min to max.

```
example

\[ A(T) = (45, 45, 60, 60, 60, 90) \]
```

The angle vector always has length \( 3t \) where \( t \) is the number of triangles.

We compare two angle vectors **lexicographically** (dictionary ordering)

```
example

\[ A(T') = (45, 45, 50, 50, 80, 90) \]
```

\[ A(T) = (45, 45, 60, 60, 60, 90) \]

so \( A(T) > A(T') \)
Lemma. Flipping an illegal edge increases the angle vector lexicographically.

Proof.

Before flip: $a, d, h, e, b+c, f+g$ (in some order)

After flip: $f, g, c, b, d+e, a+h$

$c > h$ using Thales on chord $ps$

$b > e$ \hspace{1cm} $qs$

$f > a$, $g > d$

\begin{align*}
\text{min angle before flip: one of } & a, d, h, e \\
\text{because } & b+c > e, f+g > a
\end{align*}

Claim: after flip every angle is $\geq \text{min before flip}$.

\begin{align*}
f > a & \quad g > d \\
c > h & \quad b > e \\
d + e > d & \quad a + h > a
\end{align*}

Note: other angle (outside $psgr$) don't change

\begin{align*}
\text{new min } & \geq \text{old min}
\end{align*}
Thus, flipping illegal edges **always** gets you to the Delaunay triangulation, and the Delaunay triangulation has the lexicographically maximum angle vector.

Consequences:

**Theorem.** The Delaunay triangulation maximizes the minimum angle.

**Algorithm** to find the Delaunay triangulation: find ANY triangulation and then flip illegal edges until there are none left.

How many flips does this take?

We will prove that no edge reappears while flipping illegal edges.

Thus $\# \text{flips} \leq \binom{n}{2} = \# \text{possible edges}$
Lemma. No edge reappears when we flip illegal edges.

Proof. Recall connection between Delaunay triangulation and convex hull

Flip illegal edge \((a, c)\) to \((d, b)\).

d inside \(\text{Circle}(abc)\) \(\Rightarrow\) \(d\) below plane \((\hat{a}, \hat{b}, \hat{c})\)

\(\hat{a}, \hat{b}, \hat{c}\) form a tetrahedron in \(\mathbb{R}^3\)

Before flip have top 2 triangles \(\hat{a}, \hat{b}, \hat{c}\) and \(\hat{a}, \hat{b}, \hat{c}\)

After flip have bottom 2 triangles \(\hat{a}, \hat{b}, \hat{c}\) and \(\hat{a}, \hat{b}, \hat{c}\)

Segment \(\hat{a}, \hat{c}\) disappears forever above \(\text{CH}\) of raised points.
[Randomized] Incremental Delaunay triangulation algorithm.

Add points one by one, maintaining the Delaunay triangulation. To add a new point p:

Find the current triangle ABC containing p.

Join p to A, B, C

Then flip illegal edges until there are none left.

Issues and details:

1. what if p is outside the current convex hull?
2. how to limit testing for illegal edges
3. how to find the triangle containing p
[Randomized] Incremental Delaunay triangulation algorithm.

Issues and details.

1. what if p is outside the current convex hull?

Initialize by adding a very large triangle \( q_0, q_1, q_2 \) outside all points.

Condition:
- every circle through 3 original points does not contain any \( q_i \).
[Randomized] Incremental Delaunay triangulation algorithm.

Issues and details.

2. how to limit testing for illegal edges after adding point p

Call Test(A,B), Test(B,C), Test(C,A)

where Test(U,V) is a recursive routine
to fix edge UV in triangle UVp

Test(U,V) — if UV is illegal i.e. W is inside Circle(Cuvp)
where W is apex of triangle on UV

- flip UV to pW
- test(U,W), Test(W,V)
Changes produced by this Test update:

Some region is retriangulated via a star at p.
All the new edges are incident to p.

Correctness. Why is this limited Test and retriangulation sufficient?

- all the tests and flips we do are correct
- the only issue is that we do not test all the edges to check if they are illegal
Correctness. Why is this limited Test and retriangulation sufficient?

**Claim.** Edges not incident to $p$ are legal.

**Lemma.** Any edge we add (incident to $p$) is legal. In fact, Delaunay.

- Original edges
- Edge created by flip

- Black edges — nothing changed
- Orange edges — we tested
- Blue edges?

**Example:**

- $ABC$ is Delaunay because $\text{Circle}(ABC)$ was empty & can shrink to empty circle through $e.g. Ap$

- We flipped because $p$ in $\text{Circle}(uvw)$ $p$ is the only site in there, shrink to get empty circle through $pW$.

$PW$ is Delaunay.
[Randomized] Incremental Delaunay triangulation algorithm.

Analysis of expected run time when points are inserted in random order. Note: we are still ignoring how to find which triangle contains p (and its runtime).

Lemma. The expected time to insert one point is $O(1)$.

Proof. Time spent to do tests and updates when inserting $p_i$

\[
\begin{align*}
\text{Time} &= O(\#\text{edges incident to } p_i \text{ after the updates}) \\
&= O(\text{degree of } p_i) \\
\end{align*}
\]

So we want expected degree of $p_i$ in $\mathcal{D}(\mathcal{E}[p_0 \dots p_3])$

\[
= \text{avg. degree in } \mathcal{D} = \text{a planar graph} -- \text{so } O(1).
\]

This shows that expected number of triangles created over the course of the algorithm is $O(n)$. 
[Randomized] Incremental Delaunay triangulation algorithm.

**Final issue:** How to find the triangle containing \( p \).
The method is easy, the analysis is not.
Note: it is this part of the algorithm that causes the \( O(n \log n) \) expected behaviour.

The idea is like Kirkpatrick’s Point Location.
Maintain the history of triangles and changes to them. Then “trace” point \( p_i \) through the changes.

Two possible updates to triangles:

1. Old \( \Delta \) points to 3 new \( \Delta \)s
   - Keep all this info.
   - **Claim** Expected size \( O(n) \)

2. Each old \( \Delta \) points to 2 new \( \Delta \)s
   - because expected # triangles is \( O(n) \)
[Randomized] Incremental Delaunay triangulation algorithm.

Final issue: How to find the triangle containing p.

How to “trace” p:

- initially (with one big triangle) p is in the big triangle
- at each update, the triangle containing p points to 2 or 3 new ones — check which one contains p

This completes the description of the algorithm.
Analysis of expected work to trace $p_i$

Can prove it is $O(\log i)$. Then total expected work to add all points is

$$O\left(\sum_i \log i\right) = O(n \log n)$$

First idea: charge work of tracing $p_i$ to each triangle $T$ in the sequence that contains $p_i$

Better idea: charge work to Delaunay triangles that appear in the sequence.

Can show that the expected work for triangles of $D\left(\{p_1, \ldots, p_j\}\right)$ is $O\left(\frac{3}{j}\right)$

$$O\left(\sum_{j=1}^{i-1} \frac{3}{j}\right) = O(\log i) \quad \text{Harmonic series}$$

There is a lovely backwards analysis involved. For details, see [CGAA].
What primitive operations are needed for this algorithm?

Given 4 points, A, B, C, D, is D inside Circle(A,B,C)?

Use the mapping from last day

\[(x, y) \rightarrow (x, y, z = x^2 + y^2)\]

Then the test becomes: is D below the plane through A, B, C?

This is a Sidedness test in 3D, and can be decided with a few multiplications, additions, subtractions.
Summary

- a randomized incremental algorithm for the Delaunay triangulation

- the idea of flipping illegal edges to get to the Delaunay triangulation

- the Delaunay triangulation maximizes the angle vector

References

- [CGAA] Chapter 9.

- [Zurich notes] Chapter 5.