Delaunay triangulations & Voronoi diagrams

Vissarion Fisikopoulos

Dept. of Informatics & Telecommunications, University of Athens

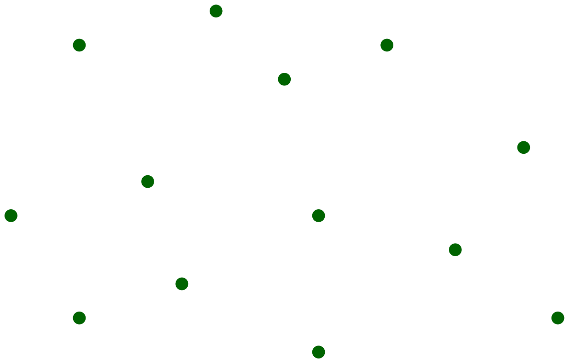
Computational Geometry, Spring 2025

Outline

Outline

A classic example

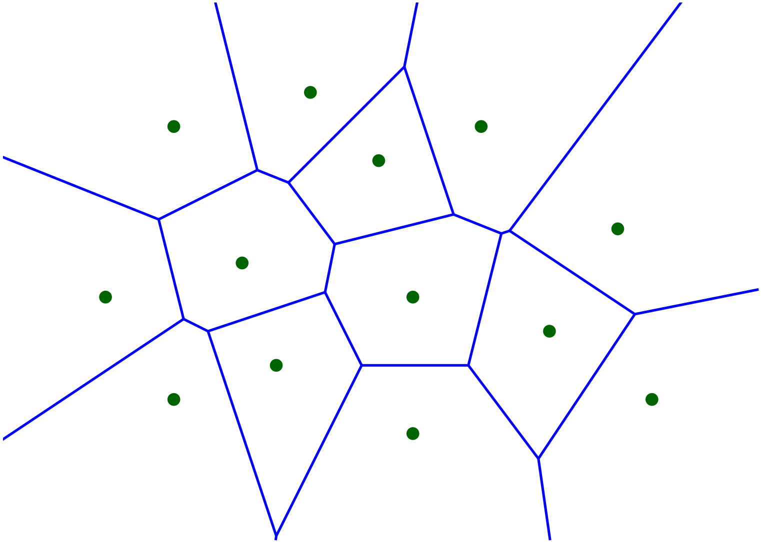
Cites: $P := \{p_1, \ldots, p_n\} \subset \mathbb{R}^2$



A classic example

Cites: $P := \{p_1, \ldots, p_n\} \subset \mathbb{R}^2$

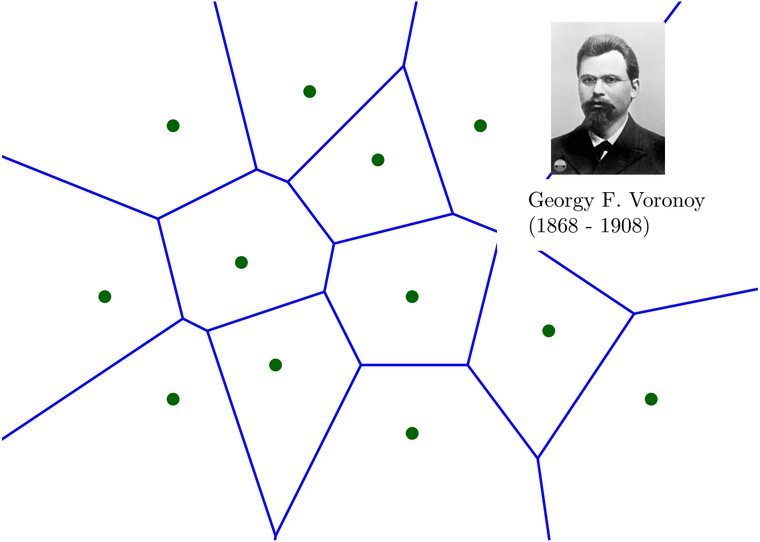
Voronoi cell: $q \in V(p_i) \Leftrightarrow \text{ dist}(q, p_i) \leq \text{ dist}(q, p_j), \forall p_j \in P, j \neq i$



A classic example

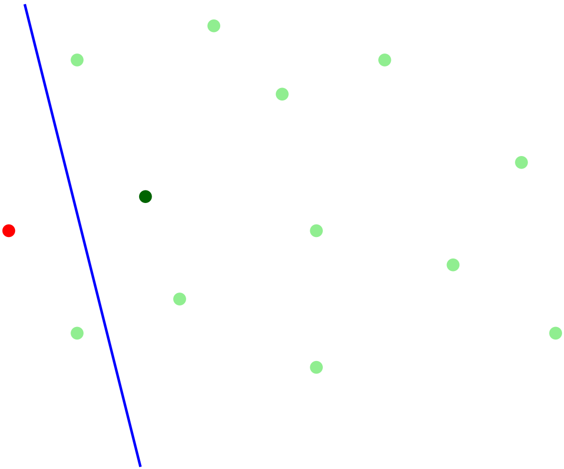
Cites: $P := \{p_1, \ldots, p_n\} \subset \mathbb{R}^2$

Voronoi cell: $q \in V(p_i) \Leftrightarrow \text{dist}(q, p_i) \leq \text{dist}(q, p_j),\ \forall p_j \in P, j \neq i$

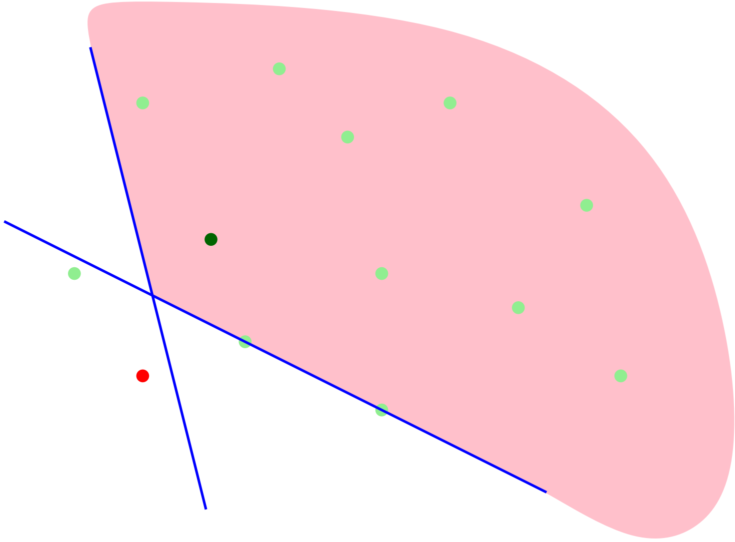


Georgy F. Voronoy
(1868 - 1908)

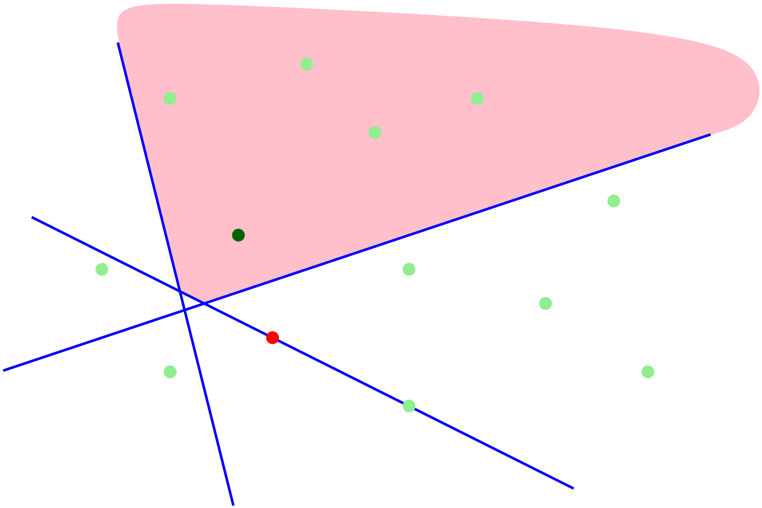
Faces of Voronoi diagram



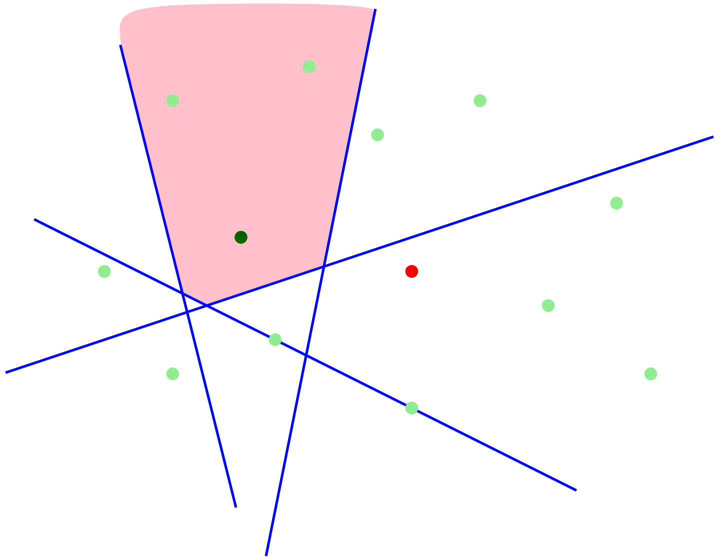
Faces of Voronoi diagram



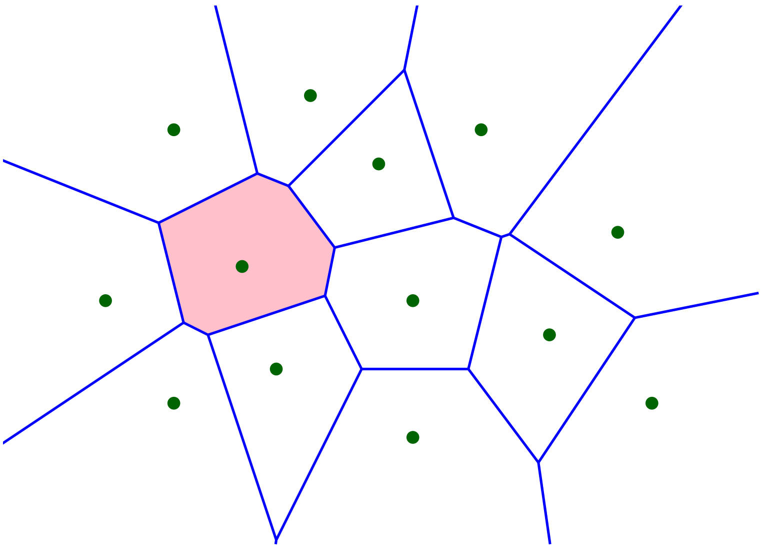
Faces of Voronoi diagram



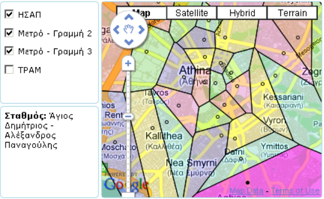
Faces of Voronoi diagram



Faces of Voronoi diagram



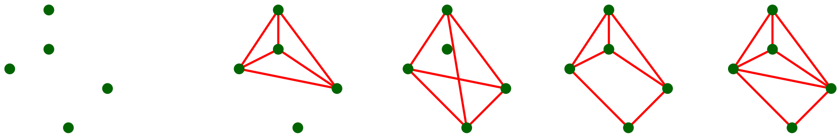
Voronoi diagrams



Triangulation

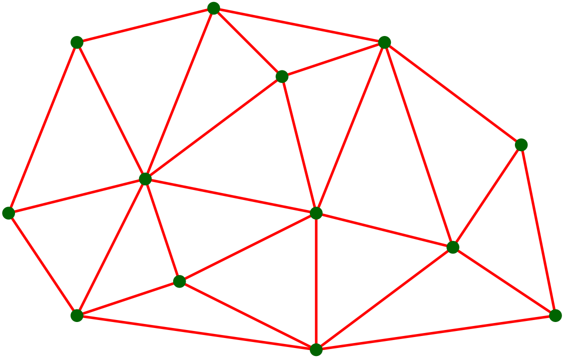
A triangulation of a point set $P \subset \mathbb{R}^2$ is a collection of subsets of P called cells s.t.

- The cells cover the convex hull of P
- Every pair of cells intersect at a (possibly empty) common face
- All cells are triangles



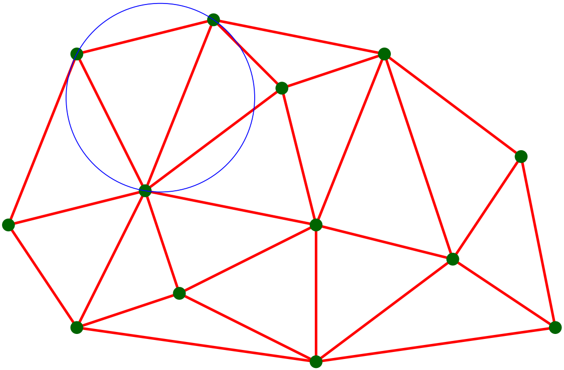
Delaunay Triangulation

The Delaunay triangulation is the unique triangulation where no point in P lies inside the circumcircle of any triangle.



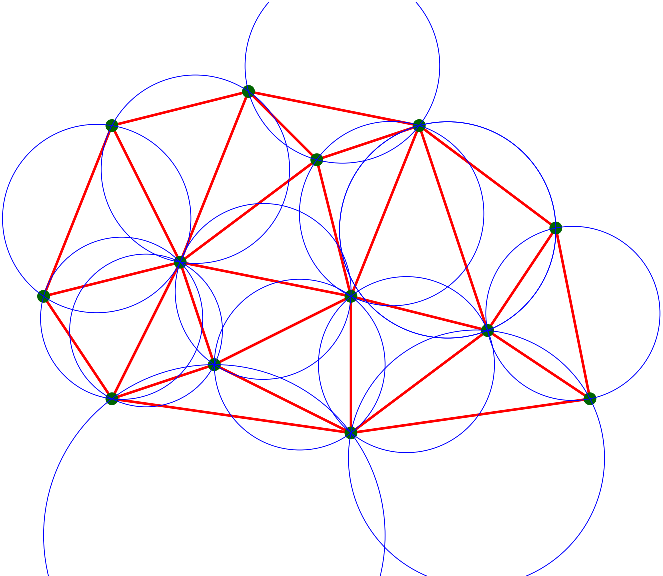
Delaunay Triangulation

The Delaunay triangulation is the unique triangulation where no point in P lies inside the circumcircle of any triangle.



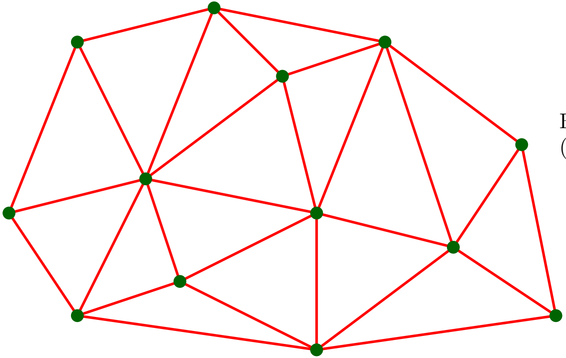
Delaunay Triangulation

The Delaunay triangulation is the unique triangulation where no point in P lies inside the circumcircle of any triangle.



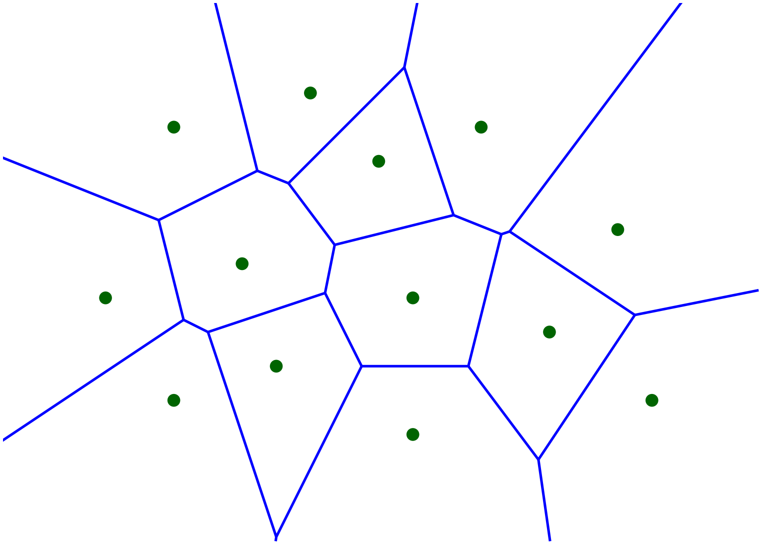
Delaunay Triangulation

The Delaunay triangulation is the unique triangulation where no point in P lies inside the circumcircle of any triangle.

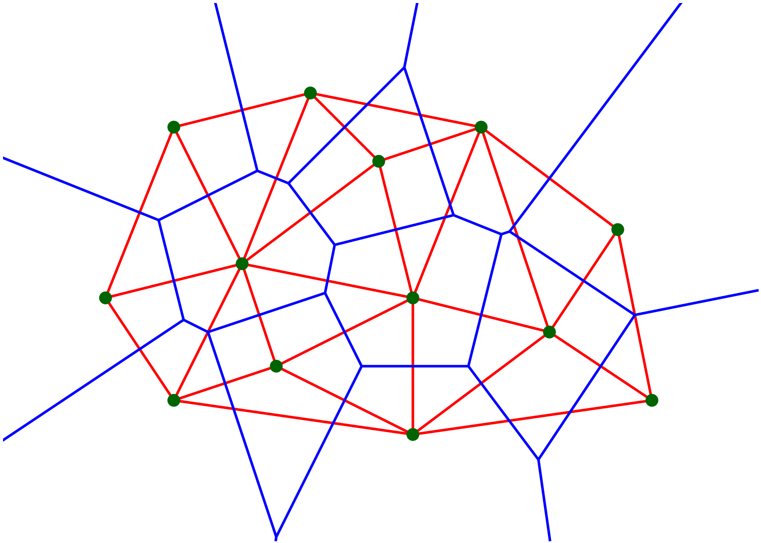


Boris N. Delaunay (1890 - 1980)

Delaunay Triangulation: dual of Voronoi diagram



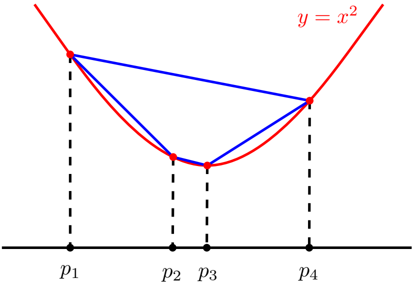
Delaunay Triangulation: dual of Voronoi diagram



Delaunay triangualtion: projection from parabola

Definition/Construction of Delaunay triangulation:

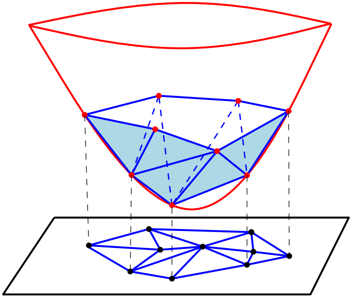
- Lift input points $p = (x) \in \mathbb{R}$ to $\widehat{p} = (x, x^2) \in \mathbb{R}^2$
- Compute the convex hull of the lifted points
- Project the lower hull to $\mathbb{R}$

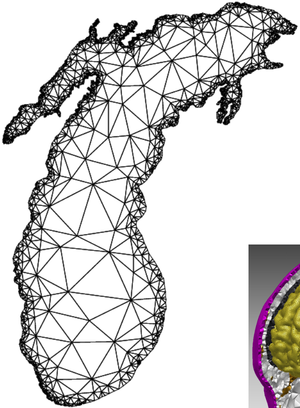


Delaunay triangualtion: going a bit higher...

Definition/Construction of Delaunay triangulation:

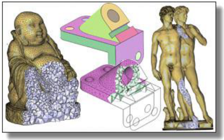
- Lift input points $p = (x, y) \in \mathbb{R}^2$ to $\widehat{p} = (x, x^2 + y^2) \in \mathbb{R}^3$
- Compute the convex hull of the lifted points
- Project the lower hull to $\mathbb{R}$

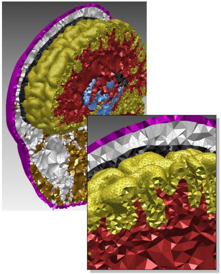


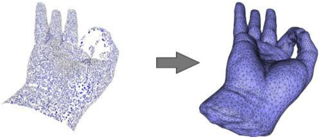


Applications

Nearest Neighbors
Reconstruction
Meshing

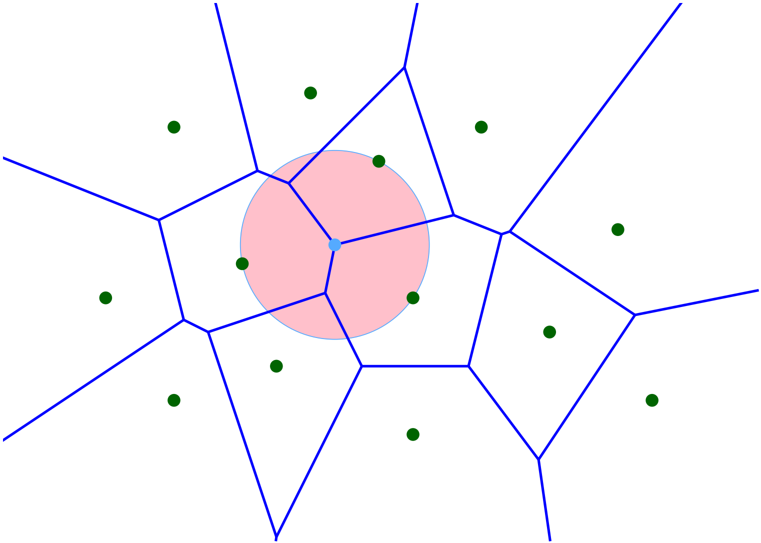




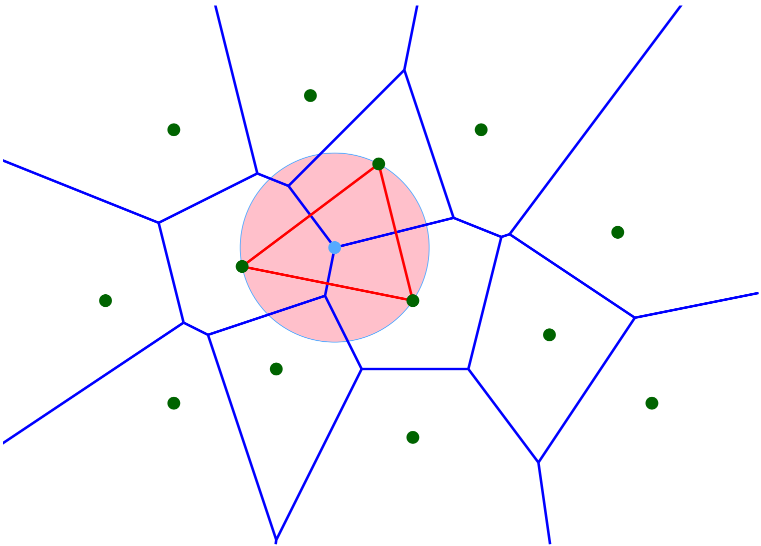


Outline

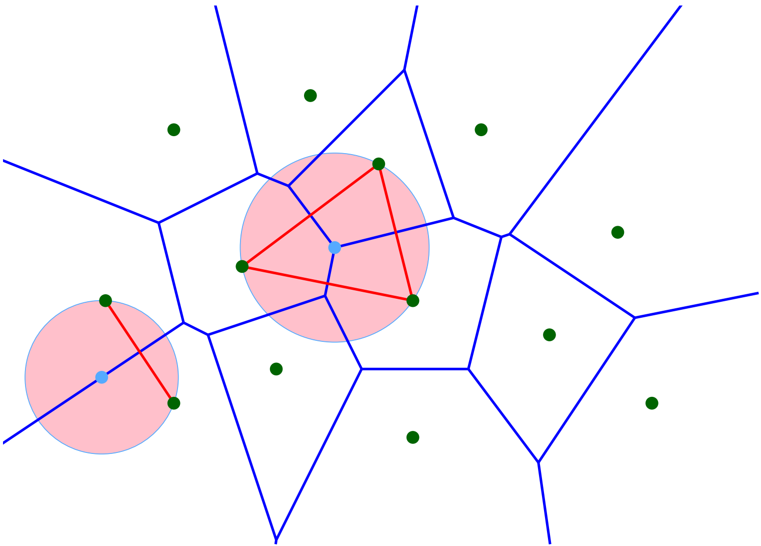
Main Delaunay property: empty sphere



Main Delaunay property: empty sphere



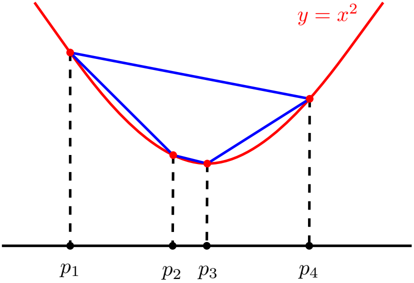
Main Delaunay property: empty sphere



Main Delaunay property: 1 picture proof

Thm (in $\mathbb{R}$): $S(p_1, p_2)$ is a Delaunay segment $\Leftrightarrow$ its interior contains no p_i.

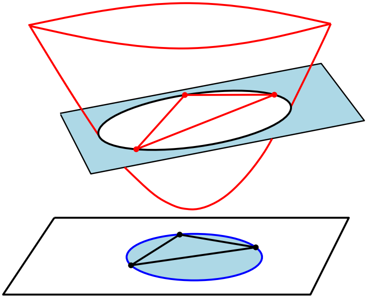
Proof. Delaunay segment $\Leftrightarrow$ $(\widehat{p_1}, \widehat{p_2})$ edge of the Lower Hull
$\Leftrightarrow$ no $\widehat{p_i}$ "below" $(\widehat{p_1}, \widehat{p_2})$ on the parabola
$\Leftrightarrow$ no p_i inside the segment (p_1, p_2).



Main Delaunay property: 1 picture proof

Thm (in $\mathbb{R}^2$): $T(p_1, p_2, p_3)$ is a Delaunay triangle $\Leftrightarrow$ the interior of the circle through p_1, p_2, p_3 (enclosing circle) contains no p_i.

Proof. Circle(p_1, p_2, p_3) contains no p_i in interior
$\Leftrightarrow$ plane of lifted $\widehat{p}_1, \widehat{p}_2, \widehat{p}_3$ leaves all lifted $\widehat{p}_i$ on same halfspace
$\Leftrightarrow$ CCW$(\widehat{p}_1, \widehat{p}_2, \widehat{p}_3, \widehat{p}_i)$ of same sign for all i.
Suffices to prove: p_i lies on Circle(p_1, p_2, p_3)
$\Leftrightarrow$ $\widehat{p}_i$ lies on plane of $\widehat{p}_1, \widehat{p}_2, \widehat{p}_3$ $\Leftrightarrow$ CCW$(\widehat{p}_1, \widehat{p}_2, \widehat{p}_3, \widehat{p}_i) = 0$.



Predicate InCircle

Given points p, q, r, $s \in \mathbb{R}^2$, point $s = (s_x, s_y)$ lies inside the circle through p, q, r $\Leftrightarrow$

$$\det \begin{pmatrix} p_x & p_y & p_x^2 + p_y^2 & 1 \\ q_x & q_y & q_x^2 + q_y^2 & 1 \\ r_x & r_y & r_x^2 + r_y^2 & 1 \\ s_x & s_y & s_x^2 + s_y^2 & 1 \end{pmatrix} > 0,$$

assuming p, q, r in clockwise order (otherwise det < 0).

Lemma. InCircle$(p, q, r, s) = 0 \Leftrightarrow \exists$ circle through p, q, r, s.
Proof. InCircle$(p, q, r, s) = 0 \Leftrightarrow CCW(\widehat{p}, \widehat{q}, \widehat{r}, \widehat{s}) = 0$

Triangulations of planar point sets

Thm. Let P be set of n points in $\mathbb{R}^2$, not all collinear, $k = \#$points on boundary of $\mathrm{CH}(P)$. Any triangulation of P has $2n - 2 - k$ triangles and $3n - 3 - k$ edges.

Proof. Hint: Euler

Triangulations of planar point sets

Thm. Let P be set of n points in $\mathbb{R}^2$, not all collinear, $k = \#$points on boundary of CH(P). Any triangulation of P has $2n-2-k$ triangles and $3n-3-k$ edges.

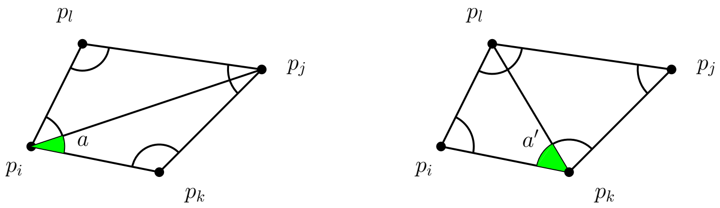
Proof.

- f: #facets (except ∞)
- e: #edges
- n: #vertices

1. Euler: $f - e + n = 1$
2. Triangulation: $3f + k = 2e$

Delaunay maximizes the smallest angle

Let T be a triangulation with m triangles.
Sort the $3m$ angles: $a_1 \leqslant a_2 \leqslant \cdots \leqslant a_{3m}$. $T_a := \{a_1, a_2, \ldots, a_{3m}\}$.
Edge $e = (p_i, p_j)$ is illegal $\Leftrightarrow \min_{1 \leqslant i \leqslant 6} a_i < \min_{1 \leqslant i \leqslant 6} a'_i$.



T' obtained from T by flipping illegal e, then $T'_a >_{lex} T_a$.

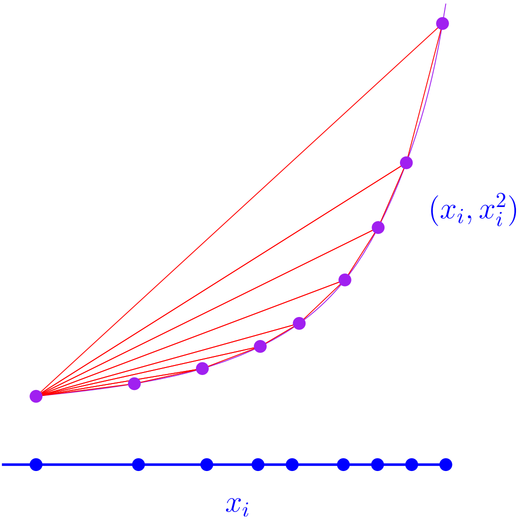
Flips yield triangualtion without illegal edges.
The algorithm terminates (angles decrease), but is too slow.

Outline

1. Definition & Examples
2. Properties
3. Algorithms

Lower bound

$\Omega(n \log n)$ by reduction from sorting



Delaunay triangulation

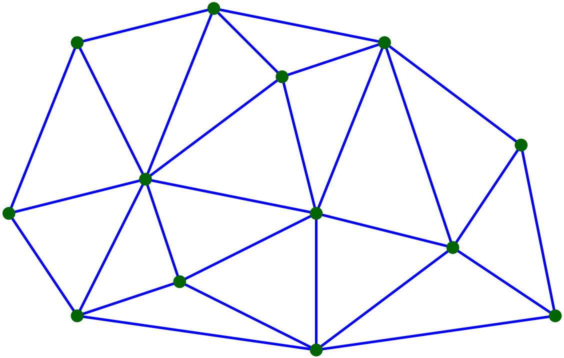
Theorem. Let P be a set of points $\in \mathbb{R}^2$. A triangulation $\mathcal{T}$ of P has no illegal edge $\Leftrightarrow$ $\mathcal{T}$ is a Delaunay triangulation of P.

Cor. Constructing the Delaunay triangulation is a fast (optimal) way of maximizing the min angle.

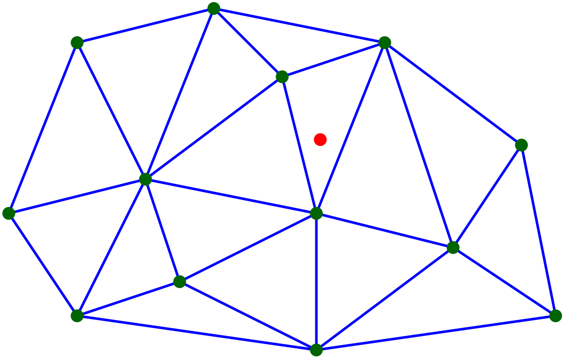
Algorithms in $\mathbb{R}^2$.

– Lift, CH3, project the lower hull: $O(n \log n)$
– Incremental algorithm: $O(n \log n)$ exp., $O(n^2)$ worst
– Construct the Voronoi diagram (sweep): $O(n \log n)$
– Divide + Conquer: $O(n \log n)$

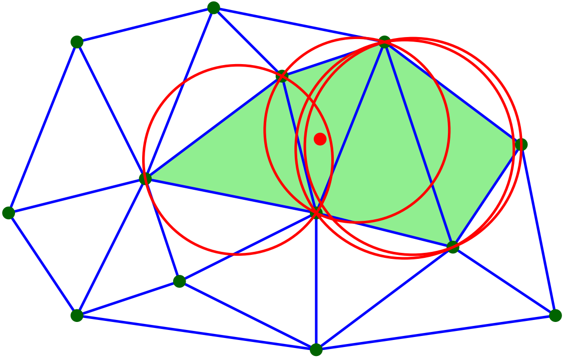
Incremental Delaunay



Incremental Delaunay

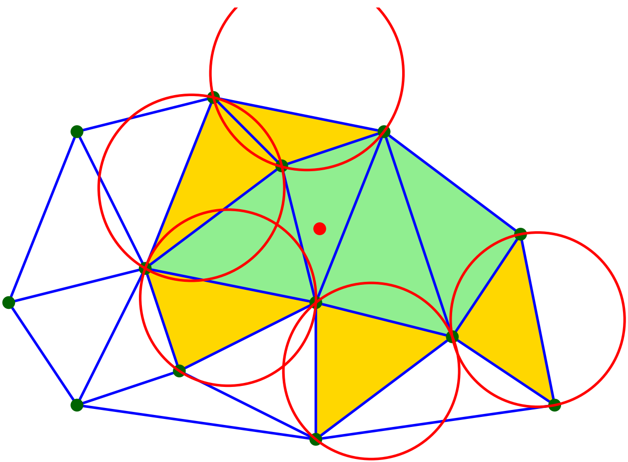


Incremental Delaunay

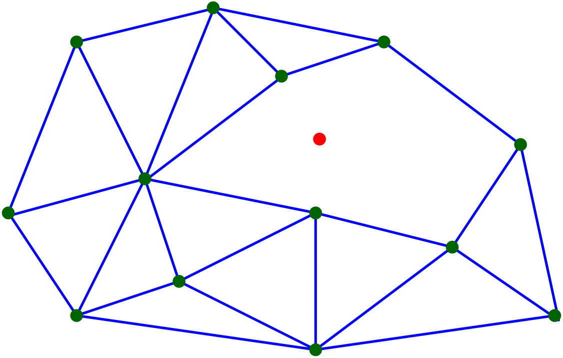


Find triangles in conflict

Incremental Delaunay

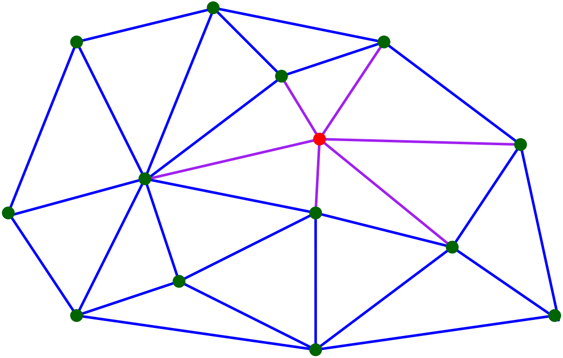


Incremental Delaunay



Delete triangles in conflict

Incremental Delaunay



Triangulate hole

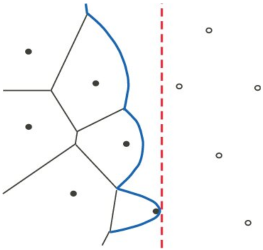
Fortune's Algorithm for Voronoi Diagram

Key Idea:

- Constructs the Voronoi diagram in $O(n \log n)$ time.
- Uses a sweep line (moving left-right) and a beach line (a sequence of parabolic arcs).

Data Structures:

- **Event (priority) Queue:** Stores site events (new point) and circle events (Voronoi vertex formation).
- **Beach Line:** A balanced binary tree maintaining active arcs.



Algorithm Steps

Step 1: Process Site Events

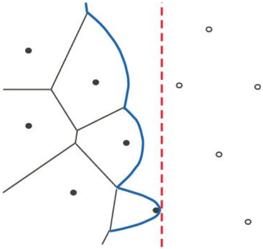
- When encountering a new point, a new parabola is created.
- The beach line updates to reflect the new parabolic region.

Step 2: Process Circle Events

- When three arcs meet, a Voronoi vertex is formed.
- The middle arc disappears, and the diagram updates.

Step 3: Maintain Beach Line

- The beach line evolves dynamically as new points appear.
- Stored in a balanced tree for efficient updates.



Algorithm Steps

Step 1: Process Site Events

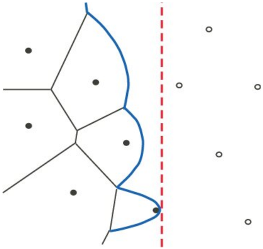
- When encountering a new point, a new parabola is created.
- The beach line updates to reflect the new parabolic region.

Step 2: Process Circle Events

- When three arcs meet, a Voronoi vertex is formed.
- The middle arc disappears, and the diagram updates.

Step 3: Maintain Beach Line

- The beach line evolves dynamically as new points appear.
- Stored in a balanced tree for efficient updates.



Algorithm Steps

Step 1: Process Site Events

- When encountering a new point, a new parabola is created.
- The beach line updates to reflect the new parabolic region.

Step 2: Process Circle Events

- When three arcs meet, a Voronoi vertex is formed.
- The middle arc disappears, and the diagram updates.

Step 3: Maintain Beach Line

- The beach line evolves dynamically as new points appear.
- Stored in a balanced tree for efficient updates.