Motivation

- There are many different triangulations for a given point set (exponentially many).
- Not all triangulations are equally “nice”:

Two copies of the same point set ...
Motivation

- There are many different triangulations for a given point set (exponentially many).
- Not all triangulations are equally “nice”:

  - Canonical triangulation
  - Delaunay triangulation
Motivation

- There are many different triangulations for a given point set (exponentially many).
- Not all triangulations are equally “nice”: What is “nice”?

Which triangulation do you like more?
“Nice” Triangulations

• Not all triangulations are equally “nice”: Not all triangulations are equally good for a given application.

⇒ Terrain: We have height information for each point.

Which triangulation is better? Why?
“Nice” Triangulations

- Not all triangulations are equally “nice”: Not all triangulations are equally good for a given application.
  ⇒ Terrain: We have height information for each point.

Interpolation points far away | Interpolation points closer
“Nice” Triangulations

- Not all triangulations are equally “nice”: Not all triangulations are equally good for a given application.

⇒ Terrain: We have height information for each point.
“Nice” Triangulations

- Not all triangulations are equally “nice”: Not all triangulations are equally good for a given application.

⇒ Terrain: We have height information for each point.

Long and skinny triangles

Short and thick triangles
“Nice” Triangulations

- Not all triangulations are equally “nice”: Not all triangulations are equally good for a given application.

⇒ Terrain: We have height information for each point.
“Nice” Triangulations

- Not all triangulations are equally “nice”: Not all triangulations are equally good for a given application.

⇒ Terrain: We have height information for each point.

Goal:
1. introduce Delaunay triangulations
2. prove that they are indeed nice

Small small angles | Large small angles
The Empty Circle Property

“We will now move on to study the ominous and supposedly nice Delaunay triangulations”

**Definition.** A triangulation of a finite point set $S \subset \mathbb{R}^2$ is called a **Delaunay triangulation**, if the **circumcircle of every triangle is empty** (has no points of $S$ in its interior).

**Circumcircle** of a triangle:
The Empty Circle Property

“We will now move on to study the ominous and supposedly nice Delaunay triangulations”

**Definition.** A triangulation of a finite point set \( S \subset \mathbb{R}^2 \) is called a **Delaunay triangulation**, if the **circumcircle of every triangle is empty** (has no points of \( S \) in its interior).

**Example:** Point set from before.
The Empty Circle Property

“We will now move on to study the ominous and supposedly nice Delaunay triangulations”

**Definition.** A triangulation of a finite point set $S \subset \mathbb{R}^2$ is called a **Delaunay triangulation**, if the circumcircle of every triangle is empty (has no points of $S$ in its interior).

**Example:** Point set from before.
The Empty Circle Property

“We will now move on to study the ominous and supposedly nice Delaunay triangulations”

Definition. A triangulation of a finite point set $S \subset \mathbb{R}^2$ is called a **Delaunay triangulation**, if the circumcircle of every triangle is empty (has no points of $S$ in its interior).

Example: Point set from before.
The Empty Circle Property

“We will now move on to study the ominous and supposedly nice Delaunay triangulations”

**Definition.** A triangulation of a finite point set $S \subset \mathbb{R}^2$ is called a **Delaunay triangulation**, if the circumcircle of every triangle is empty (has no points of $S$ in its interior).

**Basics:** Four points in convex position

Two triangulations: non-Delaunay and Delaunay
The Empty Circle Property

“We will now move on to study the ominous and supposedly nice Delaunay triangulations”

**Definition.** A triangulation of a finite point set $S \subset \mathbb{R}^2$ is called a **Delaunay triangulation**, if the **circumcircle of every triangle is empty** (has no points of $S$ in its interior).

**Basics:** Four points in convex position

Two triangulations: Delaunay Delaunay
The Empty Circle Property

“We will now move on to study the ominous and supposedly nice Delaunay triangulations”

**Definition.** A triangulation of a finite point set $S \subseteq \mathbb{R}^2$ is called a **Delaunay triangulation**, if the [circumcircle](https://en.wikipedia.org/wiki/Circle) of every triangle is empty (has no points of $S$ in its interior).

**Basics:** Four points in convex position (!) are not a valid triangle.

Only one triangulation: Delaunay
The Empty Circle Property

“We will now move on to study the ominous and supposedly nice Delaunay triangulations”

Definition. A triangulation of a finite point set $S \subset \mathbb{R}^2$ is called a Delaunay triangulation, if the circumcircle of every triangle is empty (has no points of $S$ in its interior).

Basics: Four points in convex position (?

No triangulation!

Remark. Every finite point set $S \subset \mathbb{R}^2$ with $|S| \geq 3$ and not all points collinear admits a triangulation.
The Empty Circle Property

**Proposition.** Four points in convex position that are not cocircular have exactly one Delaunay triangulation.

**Proof by picture.**

Consider circle $C_1 = abc$, assume $d$ outside $C_1$  
⇒ $b$ outside $C_1' = acd$  
⇒ $c$ inside $C_2 = abd$  
⇒ one Delaunay tr.

Case $d$ inside $C_1$ is symmetric
The Empty Circle Property

**Proposition.** Four points in convex position that are not cocircular have exactly one Delaunay triangulation.

**Proof by picture.**

Consider circle $C_1 = abc$, assume $d$ outside $C_1$

$\Rightarrow b$ outside $C_1' = acd$

$\Rightarrow c$ inside $C_2 = abd$

$\Rightarrow$ one Delaunay

Case $d$ inside $C_1$

is symmetric

Case with 3 collinear points: similar

4 collinear points: also cocircular

\[\square\]
The Empty Circle Property

**Proposition.** Four points in convex position that are not cocircular have exactly one Delaunay triangulation.

**Proof by picture.**

non-convex point set:

Only one triangulation: Delaunay
The Empty Circle Property

**Proposition.** Four points in convex position that are not cocircular have exactly one Delaunay triangulation.

**Corollary.** Any set of four points (not all of them collinear) has at least one Delaunay triangulation.
The Lawson Flip Algorithm

Does every point set (not all points collinear) have a Delaunay triangulation?

- **True** for four points. Only one situation where not every triangulation is Delaunay:

  ![Diagram](image)

  The flip from non-Delaunay to Delaunay is called **Lawson Flip**.

- A sub-triangulation of four points in a triangulation (of more than four points) is called **locally Delaunay**, if it is a Delaunay triangulation of those four points.
The Lawson Flip Algorithm

Does every point set (not all points collinear) have a Delaunay triangulation?

The Lawson Flip algorithm.

1. Compute some arbitrary triangulation.
2. While there exists a subtriangulation of four points that is not locally Delaunay, perform a Lawson flip.

We will show:

- The Lawson Flip algorithm terminates and
- results in a Delaunay triangulation.

Corollary.

Locally Delaunay everywhere = Globally Delaunay
The Parabolic Lifting Map

A projection from the plane to the unit paraboloid

The parabolic lifting map is defined by the function

\[ \ell : \mathbb{R}^2 \to \mathbb{R}^3, \quad \ell(x, y) = (x, y, x^2 + y^2). \]
The Parabolic Lifting Map

A little bit of algebra ...

- Any two points $a, b \in \mathbb{R}^2$ are collinear.
- When are three points $a, b, c \in \mathbb{R}^2$ are collinear?

$$D := \begin{vmatrix} x_a & y_a & 1 \\ x_b & y_b & 1 \\ x_c & y_c & 1 \end{vmatrix} = 0$$

- What if $D \neq 0$?
  Then $c$ lies to the left or right of the line through $a, b$. 
The Parabolic Lifting Map

* A little bit of algebra ...

- Any three points $a, b, c \in \mathbb{R}^3$ are coplanar.
- Four points $a, b, c, d \in \mathbb{R}^3$ are coplanar, if
  \[ D := \begin{vmatrix} x_a & y_a & z_a & 1 \\ x_b & y_b & z_b & 1 \\ x_c & y_c & z_c & 1 \\ x_d & y_d & z_d & 1 \end{vmatrix} = 0 \]

- What if $D \neq 0$?
  Then $d$ lies above or below the plane through $a, b, c$. 
The Parabolic Lifting Map

A little bit of algebra...

- Any three points $a, b, c \in \mathbb{R}^2$ are cocircular.
- Four points $a, b, c, d \in \mathbb{R}^2$ are cocircular if

$$D := \begin{vmatrix}
  x_a & y_a & x_a^2 + y_a^2 & 1 \\
  x_b & y_b & x_b^2 + y_b^2 & 1 \\
  x_c & y_c & x_c^2 + y_c^2 & 1 \\
  x_d & y_d & x_d^2 + y_d^2 & 1 \\
\end{vmatrix} = 0$$

- What if $D \neq 0$?
  Then $d$ lies inside or outside the circle through $a, b, c$. 
The Parabolic Lifting Map

Let’s compare once more!

- Any three points \( a, b, c \in \mathbb{R}^3 \) are coplanar.
- Four points \( a, b, c, d \in \mathbb{R}^3 \) are coplanar, if

\[
D := \begin{vmatrix}
    x_a & y_a & z_a & 1 \\
    x_b & y_b & z_b & 1 \\
    x_c & y_c & z_c & 1 \\
    x_d & y_d & z_d & 1
\end{vmatrix} = 0
\]

- What if \( D \neq 0 \)?
  Then \( d \) lies above or below the plane through \( a, b, c \).
The Parabolic Lifting Map

Let’s compare once more!

- Any three points \( a, b, c \in \mathbb{R}^2 \) are cocircular.
- Four points \( a, b, c, d \in \mathbb{R}^2 \) are cocircular if

\[
D := \begin{vmatrix}
  x_a & y_a & x_a^2 + y_a^2 & 1 \\
  x_b & y_b & x_b^2 + y_b^2 & 1 \\
  x_c & y_c & x_c^2 + y_c^2 & 1 \\
  x_d & y_d & x_d^2 + y_d^2 & 1
\end{vmatrix} = 0
\]

- What if \( D \neq 0 \)?
  Then \( d \) lies inside or outside the circle through \( a, b, c \).
A projection from the plane to the unit paraboloid

The parabolic lifting map is defined by the function
\[ \ell : \mathbb{R}^2 \rightarrow \mathbb{R}^3, \quad \ell(x, y) = (x, y, x^2 + y^2). \]

- cocircular points in the plane are coplanar in the lifting
- \( d \) lies inside a circle \( C \subset \mathbb{R}^2 \) \( \iff \)
  \[ \ell(d) \text{ lies below the plane through } \ell(C) \subset \mathbb{R}^3. \]
Termination of the Lawson Flip Algorithm

Consider again a **Lawson flip** $ac \rightarrow bd$:
Termination of the Lawson Flip Algorithm

Consider again a Lawson flip $ac \rightarrow bd$:

$bd$ lies completely in a circle $C$ through $a$ and $c$.

$\Rightarrow \ell(b)\ell(d)$ lies completely below the plane through $\ell(C)$.

$\Rightarrow$ after the flip $\ell(a)\ell(c)$ lies above the lifted triangulation.

$\Rightarrow$ An edge that is flipped away never comes back.

$\Rightarrow$ Each of the $\binom{n}{2}$ edges is flipped at most once.

$\Rightarrow$ The algorithm terminates after at most $\binom{n}{2} - 3$ flips.
Correctness of the Lawson Flip Algorithm

Locally Delaunay everywhere = Globally Delaunay

**We know:** After termination of the algorithm every subtriangulation of four points is locally Delaunay.

**We want:** After termination of the algorithm we in fact have a Delaunay triangulation.

**Proof.** Assume the result $T(S)$ contains a triangle $\triangle$ with a point $p \in S$ strictly inside $C(\triangle)$.

Chose a pair $(\triangle, p)$ with minimum distance $d$ between $p$ and $\triangle$.

Consider the edge $e$ of $\triangle$ closest to $p$ and the second triangle $\triangle'$ at $e$.

The circumcircle $C(\triangle')$ contains $p$ and $p$ is closer to $\triangle'$ than to $\triangle$. 
Next

How many Delaunay triangulations can a point set have?

- Definition of the **Delaunay graph**
- A central property of this graph

Why are we actually interested in Delaunay triangulations?

- **Angles** in Delaunay triangulations
- Some further properties of Delaunay triangulations

What if we need a triangulation with some non-Delaunay edges, for example to model a river in a terrain?

- **Constrained** Delaunay triangulations
The Delaunay Graph

How many Delaunay triangulations can a point set have?

Definition. The **Delaunay graph** of a point set \( S \subset \mathbb{R}^2 \) contains as edges all line segments \( ab, \ a, b \in S \), that are contained in every Delaunay triangulation of \( S \).

Lemma. An edge \( ab, \ a, b \in S \) is in the Delaunay graph of \( S \) if and only if there exists a circle \( C \) through \( a, b \) such that all other points of \( S \) lie strictly outside \( C \).
The Delaunay Graph

**Lemma.** An edge $ab$, $a, b \in S$, is in the Delaunay graph $D$ of $S$ if and only if there exists a circle $C$ through $a, b$ such that all other points of $S$ lie strictly outside $C$.

**Proof ⇒**
Consider edge $ab$ of $D(S)$ + Delaunay triangulation $T$ of $S$.
$\Rightarrow \exists \Delta = abc$ in $T$, $C(\Delta)$ has no points of $S$ in its interior.
Assume $\exists p \in S$ on $\partial C(\Delta)$ s.t. $cp$ intersects $ab$.
$\Rightarrow \exists$ triangle $\Delta' = abc'$ in $T$ with $c' \in \partial C(\Delta)$.
$\Rightarrow$ Flipping $ab$ to $cc'$ gives Delaunay triangulation $T'$
$\Leftrightarrow$ to $\exists p \in S$ on $\partial C'(\Delta)$.
$\Rightarrow$ all points outside $C'(\Delta)$.
The Delaunay Graph

**Lemma.** An edge $ab$, $a, b \in S$, is in the Delaunay graph $\mathcal{D}$ of $S$ if and only if there exists a circle $C$ through $a, b$ such that all other points of $S$ lie strictly outside $C$.

**Proof ⇒**
Consider edge $ab$ of $\mathcal{D}(S) +$ Delaunay triangulation $T$ of $S$.

$\Rightarrow \exists \Delta = abc$ in $T$, $C(\Delta)$ has no points of $S$ in its interior.

Assume $\exists p \in S$ on $\partial C(\Delta)$ s.t. $cp$ intersects $ab$.

$\Rightarrow \exists$ triangle $\Delta' = abc'$ in $T$ with $c' \in \partial C(\Delta)$.

$\Rightarrow$ Flipping $ab$ to $cc'$ gives Delaunay triangulation $T'$

$\Leftrightarrow \exists p \in S$ on $\partial C'(\Delta)$.

$\Rightarrow$ all points outside $C'(\Delta)$.

$\Rightarrow \exists$ circle $C'$ through $a, b$ with all other points outside $C'$.
The Delaunay Graph

**Lemma.** An edge $ab$, $a, b \in S$, is in the Delaunay graph $\mathcal{D}$ of $S$ if and only if there exists a circle $C$ through $a, b$ such that all other points of $S$ lie strictly outside $C$.

**Proof**

Consider edge $ab$ with empty circle $C$.
Assume $\exists$ Delaunay triangulation $T$ without edge $ab$.
$\Rightarrow T$ contains at least one edge that intersects $ab$.
Consider edge $cd$ of $T$ that intersects $ab$ closest to $b$.
$\Rightarrow cd$ together with $b$ forms a triangle $\Delta = bcd$ of $T$.

Circle $C'$ through $abc$: $d$ outside
Circle $C''$ through $bcd$: $a$ inside

$\Rightarrow C'' = C(\Delta)$ not empty
to $T$ Delaunay triangulation.
The Delaunay Graph

**Lemma.** An edge $ab$, $a, b \in S$, is in the Delaunay graph $\mathcal{D}$ of $S$ if and only if there exists a circle $C$ through $a, b$ such that all other points of $S$ lie strictly outside $C$.

**Corollary.** Every point set $S \subset \mathbb{R}^2$ with no four cocircular points has a unique Delaunay triangulation.

**Remark.** There exist point sets $S \subset \mathbb{R}^2$ with exponentially many different Delaunay triangulations.
Angles in Delaunay Triangulations

Why are we actually interested in Delaunay triangulations?

Recall. We compared the canonical triangulation with the Delaunay triangulation for some point set.

Closeby interpolation points
Short and thick triangles
Large small angles in triangles

Question: Is this true in general or just for that point set?
Angles in Delaunay Triangulations

Why are we actually interested in Delaunay triangulations?

Recall. Given a point set $S \subset \mathbb{R}^2$, every triangulation of $S$ has the same number $m$ of triangles.

Definition. Given a triangulation $T$ of a point set $S \subset \mathbb{R}^2$, the angle vector $A(T) = (\alpha_1, \alpha_2, \ldots, \alpha_{3m})$ of $T$ is the sorted sequence of the interior angles of $T$.

$$\alpha_1 \leq \alpha_2 \leq \ldots \leq \alpha_{3m}$$
Angles in Delaunay Triangulations

Why are we actually interested in Delaunay triangulations?

Recall. Given a point set \( S \subset \mathbb{R}^2 \), every triangulation of \( S \) has the same number \( m \) of triangles.

Definition. Given a triangulation \( T \) of a point set \( S \subset \mathbb{R}^2 \), the angle vector \( A(T) = (\alpha_1, \alpha_2, \ldots, \alpha_{3m}) \) of \( T \) is the sorted sequence of the interior angles of \( T \).

Consider a second triangulation \( T' \) of \( S \). We say that \( A(T) < A(T') \) if \( A(T) \) is lexicographically smaller than \( A(T') \).

\[
\exists 1 \leq i \leq 3m : \alpha_j = \alpha'_j \ \forall \ 1 \leq j < i \ \text{and} \ \alpha_i < \alpha'_i.
\]
Angles in Delaunay Triangulations

Why are we actually interested in Delaunay triangulations?

Recall. Given a point set $S \subset \mathbb{R}^2$, every triangulation of $S$ has the same number $m$ of triangles.

Definition. Given a triangulation $T$ of a point set $S \subset \mathbb{R}^2$, the angle vector $A(T) = (\alpha_1, \alpha_2, \ldots, \alpha_{3m})$ of $T$ is the sorted sequence of the interior angles of $T$.

Consider a second triangulation $T'$ of $S$. We say that $A(T) < A(T')$ if $A(T)$ is lexicographically smaller than $A(T')$.

Theorem. For any point set $S \subset \mathbb{R}^2$ with no four cocircular points, the Delaunay triangulation strictly maximizes the angle vector among all triangulations of $S$. 
Angles in Delaunay Triangulations

Why are we actually interested in Delaunay triangulations?

Recall. Given a point set \( S \subset \mathbb{R}^2 \), every triangulation of \( S \) has the same number \( m \) of triangles.

Definition. Given a triangulation \( T \) of a point set \( S \subset \mathbb{R}^2 \), the angle vector \( A(T) = (\alpha_1, \alpha_2, \ldots, \alpha_{3m}) \) of \( T \) is the sorted sequence of the interior angles of \( T \).

Consider a second triangulation \( T' \) of \( S \). We say that \( A(T) < A(T') \) if \( A(T) \) is lexicographically smaller than \( A(T') \).

Theorem. For any point set \( S \subset \mathbb{R}^2 \) with no four cocircular points, the Delaunay triangulation strictly maximizes the angle vector among all triangulations of \( S \). \[ A(T) < A(D) \] for all triangulations \( T \neq D \) of \( S \).
Angles in Delaunay Triangulations

**Theorem.** For any point set $S \subset \mathbb{R}^2$ with no four cocircular points, the Delaunay triangulation strictly maximizes the angle vector among all triangulations of $S$.

**Proof.**
If every Lawson flip increases the angle vector we are done: Every triangulation $T$ of $S$ can be transformed to the Delaunay triangulation $DT$ via Lawson flips.
Angles in Delaunay Triangulations

**Theorem.** For any point set $S \subset \mathbb{R}^2$ with no four cocircular points, the Delaunay triangulation strictly maximizes the angle vector among all triangulations of $S$.

**Proof.**
Four cocircular points.

**Claim:**
Angles in Delaunay Triangulations

**Theorem.** For any point set $S \subset \mathbb{R}^2$ with no four cocircular points, the Delaunay triangulation strictly maximizes the angle vector among all triangulations of $S$.

**Proof.**

Four cocircular points.

Claim: $\bar{\alpha} > \alpha$, $\bar{\beta} > \beta$, $\gamma < \gamma$, $\delta < \delta$
Angles in Delaunay Triangulations

**Theorem.** For any point set $S \subset \mathbb{R}^2$ with no four cocircular points, the Delaunay triangulation strictly maximizes the angle vector among all triangulations of $S$.

**Proof.**
Lawson flip $bd \rightarrow ac$, $T \rightarrow T'$. Four non-cocircular points.

Angles before flip:

- $\bar{\alpha} + \bar{\beta}$, $\delta$, $\gamma$, $\alpha$, $\gamma + \delta$, $\beta$

Angles after flip:

- $\bar{\beta}$, $\delta + \alpha$, $\gamma$, $\bar{\alpha}$, $\delta$, $\beta + \gamma$

$\Rightarrow$ for every angle after there is at least one smaller angle before.

$\Rightarrow A(T) < A(T')$
Angles in Delaunay Triangulations

**Theorem.** For any point set $S \subset \mathbb{R}^2$ with no four cocircular points, the Delaunay triangulation strictly maximizes the angle vector among all triangulations of $S$.

**Proof.**
Four cocircular points.

**Claim:**

**Proof of Claim.**
Angles in Delaunay Triangulations

**Theorem.** For any point set $S \subset \mathbb{R}^2$ with no four cocircular points, the Delaunay triangulation strictly maximizes the angle vector among all triangulations of $S$.

**Proof.**

Four cocircular points.

Proof of Claim.

Claim:
**Theorem.** For any point set $S \subset \mathbb{R}^2$ with no four cocircular points, the Delaunay triangulation strictly maximizes the angle vector among all triangulations of $S$.

**Proof.**
Four cocircular points.

**Claim:**

**Proof of Claim.**
Case 1: $m$ outside triangle $abc$

- **triangle $mbc$:**
  $$\pi = \delta + 2\beta$$
- **triangle $abc$:**
  $$\pi = 2\alpha + 2\beta - 2\gamma$$
- **$mbc$:**
  $$\delta = \pi - 2\beta$$
- **$abc$:**
  $$2\beta = \pi - 2\alpha + 2\gamma$$
- **$\phi = \alpha - \gamma \Rightarrow \delta = 2\phi$**
Angles in Delaunay Triangulations

**Theorem.** For any point set $S \subset \mathbb{R}^2$ with no four cocircular points, the Delaunay triangulation strictly maximizes the angle vector among all triangulations of $S$.

**Proof.**

Four cocircular points.

**Claim:**

**Proof of Claim.**

Case 2: $m$ inside triangle $abc$

- **Triangle $mbc$:**
  \[
  \pi = \delta + 2\beta
  \]
- **Triangle $abc$:**
  \[
  \pi = 2\alpha + 2\beta + 2\gamma
  \]
- **$mbc$:**
  \[
  \delta = \pi - 2\beta
  \]
- **$abc$:**
  \[
  2\beta = \pi - 2\alpha - 2\gamma
  \]
- \[
  \phi = \alpha + \gamma \Rightarrow \delta = 2\phi
  \]
Angles in Delaunay Triangulations

**Theorem.** For any point set $S \subset \mathbb{R}^2$ with no four cocircular points, the Delaunay triangulation strictly maximizes the angle vector among all triangulations of $S$.

**Question.** What if the point set has cocircular points?

Angle vectors are in general different!
Angles in Delaunay Triangulations

**Theorem.** For any point set $S \subset \mathbb{R}^2$ with no four cocircular points, the Delaunay triangulation strictly maximizes the angle vector among all triangulations of $S$.

**Theorem.** For any point set $S \subset \mathbb{R}^2$ (not all points coll.), every Delaunay triangulation maximizes the smallest angle among all triangulations of $S$.

**Proof.**
- Take an arbitrary Delaunay triangulation $DT$ of $S$.
- Slightly perturb $S$ to eliminate cocircularities.
- Apply Lawson flips to get the unique Delaunay tr. $D^*$.
- Perform the same flip sequence in $S$ (unperturbed).

$\Rightarrow$ All those flips are between cocircular points.
$\Rightarrow$ The value of the smallest angle never changes.
Properties of Delaunay Triangulations

*Delaunay triangulations are “nice”*

- Every Delaunay triangulation of a point set $S$ maximizes the smallest angle among all triangulations of $S$.
- If the Delaunay triangulation of $S$ is unique, it strictly maximizes the angle vector among all triangulations of $S$.
- Further, every Delaunay triangulation of a point set $S$ minimizes the largest circumcircle and maximizes the mean inradius among all triangulations of $S$.
- Hence, Delaunay triangulations tend to
  - avoid narrow triangles
  - avoid numerical problems
  - allow fast convergence in FEM
Constrained Delaunay Triangulations

What if we need a triangulation with some non-Delaunay edges, for example to model a river in a terrain?

Idea. Given a point set $S \subset \mathbb{R}^2$ and a plane geometric graph $G$ on $S$, we want to complete $G$ to a triangulation of $S$ that is as close to a Delaunay triangulation as possible.
Constrained Delaunay Triangulations

*What if we need a triangulation with some non-Delaunay edges, for example to model a river in a terrain?*

**Definition.** Given a point set $S \subset \mathbb{R}^2$ and a plane straight-line graph $G$ on $S$, a triangulation of $S$ that contains all edges of $G$ is a **constrained Delaunay triangulation** of $S$ with respect to $G$ if the circumcircle of each triangle $\Delta$ contains no points that are visible from the interior of $\Delta$.

Every straight-line connection between a point of $S$ inside the circumcircle of $\Delta$ and a point inside $\Delta$ intersects an edge of $G$. 
Constrained Delaunay Triangulations

What if we need a triangulation with some non-Delaunay edges, for example to model a river in a terrain?
Constrained Delaunay Triangulations

What if we need a triangulation with some non-Delaunay edges, for example to model a river in a terrain?
Constrained Delaunay Triangulations

What if we need a triangulation with some non-Delaunay edges, for example to model a river in a terrain?

Definition. Given a point set $S \subseteq \mathbb{R}^2$ and a plane straight-line graph $G$ on $S$, a triangulation of $S$ that contains all edges of $G$ is a constrained Delaunay triangulation of $S$ with respect to $G$ if the circumcircle of each triangle $\triangle$ contains no points that are visible from the interior of $\triangle$.

Theorem. For any point set $S \subseteq \mathbb{R}^2$ and any plane straight-line graph $G$ on $S$, there exists a constrained Delaunay triangulation of $S$ with respect to $G$. 
Some Exercises

1. For every $n \geq 3$ give an example of an $n$-point set in which every Delaunay triangulation of the point set contains a vertex of degree $n - 1$.

2. A Minimum spanning tree (MST) of a point set $S \subset \mathbb{R}^2$ is a spanning tree that minimizes the sum of the edge lengths among all spanning trees of $S$.

Prove or disprove the following statements:

(a) Every MST is a plane graph.
(b) Every MST of $S$ contains an edge between some pair of points of $S$ with minimum distance.
(c) Every MST of $S$ contains an edge between every pair of points of $S$ with minimum distance.
(d) Every Delaunay triangulation of $S$ contains an MST of $S$. 

Some Exercises

3. Prove or disprove that every triangulation with minimum total edge length is a Delaunay triangulation.

4. Given a triangulation $T(S)$ of a set $S$ of $n$ points in general position and an extreme / interior point $p \in S$ with $d(p) < n - 1$, is there always a flip that increases $d(p)$?

5. Given a triangulation $T(S)$ of a set $S$ of $n$ points in general position and an extreme / interior point $p \in S$ with $d(p) > 3$, is there always a flip that decreases $d(p)$?

6. How fast can the canonical triangulation of an $x$-sorted of a set $S$ of $n$ points in general position be computed? Design a worst-case optimal algorithm.
Constructing Delaunay Triangulations

How fast can we compute a Delaunay triangulation?

Up to now: The Lawson Flip algorithm. $O(n^2)$ flips
1. Compute some arbitrary triangulation.
2. While there exists a subtriangulation of four points that is not locally Delaunay, perform a Lawson flip.

Open: How and how fast can we find if and where to flip?

Alternative Idea: Insert points one after another.
- Maintain Delaunay triangulation $D(R)$ of the point set $R \cup S$ inserted so far.
- When adding $s$, update $D(R)$ to obtain $D(R \cup s)$.
- Start with a (large) triangle of three extra points that contains all points of $S$ (and end with removing the triangle) $\Rightarrow$ No special cases, all insertions interior.
Adding a point to a Delaunay Triangulation

**Steps for point insertion** (point $s$ interior of $D(R)$):

1. Find the triangle $\Delta$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\Delta$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

**Example:**

![Example diagram of a Delaunay triangulation with a point added and Lawson flips applied]
Adding a point to a Delaunay Triangulation

**Steps for point insertion** (point \( s \) interior of \( D(R) \)):

1. Find the triangle \( \Delta \) of \( D(R) \) that contains \( s \).
2. Connect \( s \) to all vertices of \( \Delta \) to obtain a triangulation \( T(R \cup \{s\}) \) from \( D(R) \).
3. Make Lawson flips to transform \( T(R \cup \{s\}) \) to a Delaunay triangulation \( D(R \cup \{s\}) \).

**Example:**
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):

1. Find the triangle $\Delta$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\Delta$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Example:
Adding a point to a Delaunay Triangulation

**Steps for point insertion** (point $s$ interior of $D(R)$):

1. Find the triangle $\triangle$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\triangle$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

**Example:**

![Diagram of Delaunay Triangulation with point insertion](image)
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):

1. Find the triangle $\Delta$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\Delta$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Example:
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):

1. Find the triangle $\Delta$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\Delta$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Example:
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):
1. Find the triangle $\Delta$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\Delta$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Example:
Adding a point to a Delaunay Triangulation

Steps for point insertion (point \( s \) interior of \( D(R) \)):
1. Find the triangle \( \Delta \) of \( D(R) \) that contains \( s \).
2. Connect \( s \) to all vertices of \( \Delta \) to obtain a triangulation \( T(R \cup \{s\}) \) from \( D(R) \).
3. Make Lawson flips to transform \( T(R \cup \{s\}) \) to a Delaunay triangulation \( D(R \cup \{s\}) \).

Example:
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):
1. Find the triangle $\Delta$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\Delta$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Example:
Adding a point to a Delaunay Triangulation

**Steps for point insertion** (point $s$ interior of $D(R)$):

1. Find the triangle $\triangle$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\triangle$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

**Example:**

![Diagram of Delaunay triangulation with point $s$ added and Lawson flips highlighted.]
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):
1. Find the triangle $\triangle$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\triangle$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Example:
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):
1. Find the triangle $\Delta$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\Delta$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Lemma. Every new edge is incident to $s$.

Example:
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):
1. Find the triangle $\triangle$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\triangle$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Lemma. Every new edge is incident to $s$.

Corollary. Locating the edges that need to be flipped can be done in constant time per edge.
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):
1. Find the triangle $\Delta$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\Delta$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Lemma. Every new edge is in the Delaunay graph of $R \cup \{s\}$.

Proof.
Edges added in Step 2:
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):
1. Find the triangle $\Delta$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\Delta$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Lemma. Every new edge is in the Delaunay graph of $R \cup \{s\}$.

Proof.
Edges added in Step 2:
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):

1. Find the triangle $\Delta$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\Delta$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Lemma. Every new edge is in the Delaunay graph of $R \cup \{s\}$.

Proof.

Edges added in Step 2:
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):
1. Find the triangle $\Delta$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\Delta$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Lemma. Every new edge is in the Delaunay graph of $R \cup \{s\}$.

Proof.
Edges added in Step 3:
Adding a point to a Delaunay Triangulation

Steps for point insertion (point $s$ interior of $D(R)$):
1. Find the triangle $\triangle$ of $D(R)$ that contains $s$.
2. Connect $s$ to all vertices of $\triangle$ to obtain a triangulation $T(R \cup \{s\})$ from $D(R)$.
3. Make Lawson flips to transform $T(R \cup \{s\})$ to a Delaunay triangulation $D(R \cup \{s\})$.

Lemma. Every new edge is in the Delaunay graph of $R \cup \{s\}$.

Corollary. The number of flips for adding a new point $s$ to $D(R)$ is $d(s) - 3$, where $d(s)$ is the degree of $s$ in the resulting Delaunay triangulation $D(R \cup \{s\})$.

Next: How can we efficiently solve Step 1?
History Graph

For locating $s$ in $D(R)$, we maintain a **history graph**:

- Its vertices are all triangles that have ever been created during the incremental construction of $D(R)$.
- It contains a **directed edge** from $\Delta$ to $\Delta'$ whenever $\Delta$ has been destroyed during the insertion of $\Delta'$. 
Incremental Construction

How much in total?

Incrementally constructing a Delaunay triangulation $D(S)$:

- We add an initial triangle of extra points
  History graph: first vertex $O(n)$
- For each point of $S$ we ...
  1. find the triangle $\Delta$ of $D(R)$ that contains $s$.
     History graph: find unique path to leaf. per step $\Theta(1)$
  2. Connect $s$ to all vertices of $\Delta$ to obtain a
     triangulation $T(R \cup \{s\})$ from $D(R)$. $\Theta(1)$
     History graph: add 3 vertices + edges, $d_{out}(\Delta) = 3$.
  3. Make Lawson flips to transform $T(R \cup \{s\})$ to a
     Delaunay triangulation $D(R \cup \{s\})$. per flip $\Theta(1)$
     History graph: add 2 vertices + 4 edges per flip,
     $d_{out} = 2$ for every destroyed triangle.
- We remove the initial triangle and all incident edges. $O(n)$
Incremental Construction: Analysis

**Problem 1.** How many flips / fliptests per inserted point?

- $d(s) - 3$ flips if $s$ has degree $d(s)$ in $D(R \cup \{s\})$.
- 3 tests after inserting, $\leq 2$ additional tests per flip.
- Trick: insert points in randomized order.

$\implies$ Expected degree: $E(d(s)) < 6$.

$\implies$ Expected number of flips and tests for flips: $\Theta(1)$.

$\implies$ Expected number of triangles created in total during the construction: $\Theta(n)$.

$\implies$ Expected size of the history graph: $\Theta(n)$.

$\implies$ **Expected total storage:** $\Theta(n)$. 
Incremental Construction: Analysis

**Problem 2.** How much time for locating the next point?

- **History graph:** every vertex has outdegree $\leq 3$
  
  $\Rightarrow$ Search time per step: $\Theta(1)$.

- Assume $s$ is inserted as $i^{th}$ point and that $p$ was inserted before $s$ as $j^{th}$ point ($j < i$) into $D(R)$.
  
  $\Rightarrow$ Probability that the triangle for $s$ in $D(R)$ is different from the triangle $\Delta$ for $s$ in $D(R \cup \{p\})$:
  
  Probability that $p$ is a vertex of $\Delta$ in $D(R \cup \{p\})$.

  $\Rightarrow$ Triangle for $s$ has changed with probability $3/j$ during insertion of $p$.

  $\Rightarrow$ Expected number of triangle changes $\leq 2 \sum_{j < i} \frac{3}{j} = O(\log i)$

  $\Rightarrow$ **Expected total running time:** $O(n \log n)$. 

Incremental Construction: Analysis

**Theorem.** A Delaunay triangulation of any \( n \)-point set in \( \mathbb{R}^2 \) can be constructed incrementally in expected time \( O(n \log n) \) and with expected storage \( \Theta(n) \).

**Question:** What is the worst-case runtime and space of this incremental algorithm?

**Remark.** It has been shown that any incremental construction algorithm for Delaunay triangulations needs \( \Omega(n^2) \) time in the worst case.

**Question:** What is a lower bound for the worst-case runtime of any Delaunay triangulation algorithm?
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... 
... a similar approach works for Delaunay triangulations!
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... … a similar approach works for Delaunay triangulations!
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... … a similar approach works for Delaunay triangulations!
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... ... a similar approach works for Delaunay triangulations!
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... … a similar approach works for Delaunay triangulations!
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ...

... a similar approach works for Delaunay triangulations!
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ...

... a similar approach works for Delaunay triangulations!
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... … a similar approach works for Delaunay triangulations!
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... … a similar approach works for Delaunay triangulations!
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... 
... a similar approach works for Delaunay triangulations!
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... 
... a similar approach works for Delaunay triangulations!
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... ... a similar approach works for Delaunay triangulations!

Merge:  ● create triangles with empty circumcircles
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... 
... a similar approach works for Delaunay triangulations!

Merge: • create triangles with empty circumcircles
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... ... a similar approach works for Delaunay triangulations!

Merge: • create triangles with empty circumcircles
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... ... a similar approach works for Delaunay triangulations!

Merge: create triangles with empty circumcircles
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... ... a similar approach works for Delaunay triangulations!

Merge:
- create triangles with empty circumcircles
- remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... … a similar approach works for Delaunay triangulations!

Merge:
- create triangles with empty circumcircles
- remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... ... a similar approach works for Delaunay triangulations!

Merge:
- create triangles with empty circumcircles
- remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... ... a similar approach works for Delaunay triangulations!

Merge:
- create triangles with empty circumcircles
- remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ...  
... a similar approach works for Delaunay triangulations!

Merge: • create triangles with empty circumcircles
      • remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... ... a similar approach works for Delaunay triangulations!

Merge:
- create triangles with empty circumcircles
- remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... ... a similar approach works for Delaunay triangulations!

Merge:
- create triangles with empty circumcircles
- remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... ... a similar approach works for Delaunay triangulations!

Merge:
- create triangles with empty circumcircles
- remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... … a similar approach works for Delaunay triangulations!

Merge:
- create triangles with empty circumcircles
- remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... 
... a similar approach works for Delaunay triangulations!

Merge: • create triangles with empty circumcircles
• remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... … a similar approach works for Delaunay triangulations!

Merge:  
- create triangles with empty circumcircles  
- remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ...
... a similar approach works for Delaunay triangulations!

Merge: • create triangles with empty circumcircles
      • remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... 
... a similar approach works for Delaunay triangulations!

Merge:
- create triangles with empty circumcircles
- remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... 
... a similar approach works for Delaunay triangulations!

Merge:  
• create triangles with empty circumcircles
• remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... … a similar approach works for Delaunay triangulations!

Merge: • create triangles with empty circumcircles
      • remove conflicting edges
Divide and Conquer

Recall the algorithm for convex hulls from the first lecture ... 
... a similar approach works for Delaunay triangulations!

Merge: • create triangles with empty circumcircles
      • remove conflicting edges

It can be shown that merging can be done in linear time.

**Theorem.** A Delaunay triangulation of any $n$-point set in $\mathbb{R}^2$ can be constructed in $O(n \log n)$ time.
Conclusion

Delaunay triangulations can be computed efficiently

- Incremental construction in expected time $O(n \log n)$
- Divide and conquer with deterministic time $O(n \log n)$
- More constructions later this course.

Delaunay triangulations are “nice”

- Maximize the smallest angle or even the angle vector
- Minimize the largest circumcircle
- Maximize the mean inradius
- Tend to avoid narrow triangles
- More properties? See exercises!