Voronoi Diagrams and Delaunay Triangulations

O'Rourke, Chapter 5

Outline

- Preliminaries
- Properties and Applications
- Computing the Delaunay Triangulation

Preliminaries

Given a function $f: \mathbb{R}^2 \to \mathbb{R}$, the tangent plane

$$z(x, y) = a \cdot x + b \cdot y + c$$

at $p = (x_0, y_0)$ is the best linear approximation of f.

The values and derivatives match:

$$f(p) = z(p)$$

$$\left.\frac{\partial f}{\partial x}\right|_p = \left.\frac{\partial z}{\partial x}\right|_p \qquad \left.\frac{\partial f}{\partial y}\right|_p = \left.\frac{\partial z}{\partial y}\right|_p$$

$$\Downarrow$$

$$z(x, y) = \left.\frac{\partial f}{\partial x}\right|_p \cdot (x - x_0) + \left.\frac{\partial f}{\partial y}\right|_p \cdot (y - y_0) + f(p)$$

Preliminaries

Definition:

Given a set of points $P = \{p_1, \dots, p_n\}$, $\mathcal{T}(P)$ is a *triangulation* of P if it is a partition of the convex hull of P into disjoint triangles whose vertices are exactly the points of P.

Preliminaries

Claim:

Given a set of points $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$, the number of triangles in a triangulation of P is independent of the triangulation.

Preliminaries

Proof:

Let h be the number of vertices on the hull.

By Euler's formula:

$$V - E + F = 1$$

Each edge not on the hull appears on two triangles:

$$\frac{3F - h}{2} = E - h \quad \Leftrightarrow \quad \frac{3F + h}{2} = E.$$

So by Euler's formula:

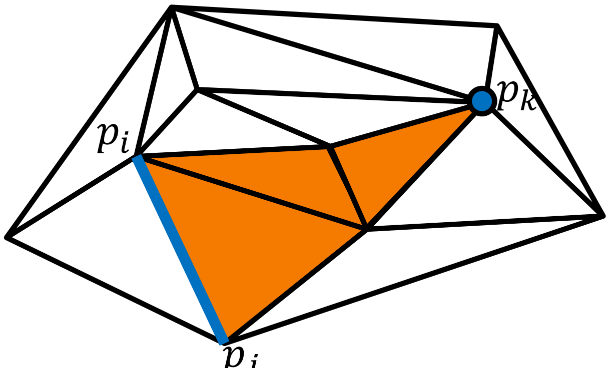
$$V - \frac{3F + h}{2} + F = 1 \quad \Leftrightarrow \quad F = 2V - h - 2.$$

Preliminaries

Claim:

Given a triangulation $\mathcal{T}(P)$, given an edge $\overline{p_i p_j}$ in the triangulation, and given a vertex p_k, we can find a sequence of edge-adjacent triangles $\{t_1, \ldots, t_m\}$ such that:

- $p_i, p_j \in t_1$ and $p \in t_m$.
- if $e \in t_l \cap t_{l+1}$ then t_{l+1} is on the same side of e as p.



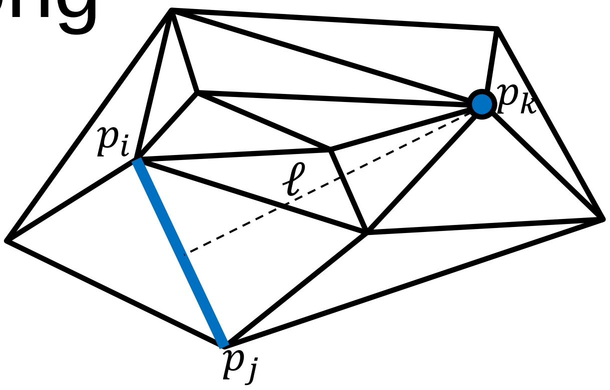
Preliminaries

<u>Proof</u>:

Let ℓ be the line segment from the middle of edge $\overline{p_i p_j}$ to p_k.

Since $\mathcal{T}(P)$ is a triangulation of the convex hull, ℓ is within the triangulation.

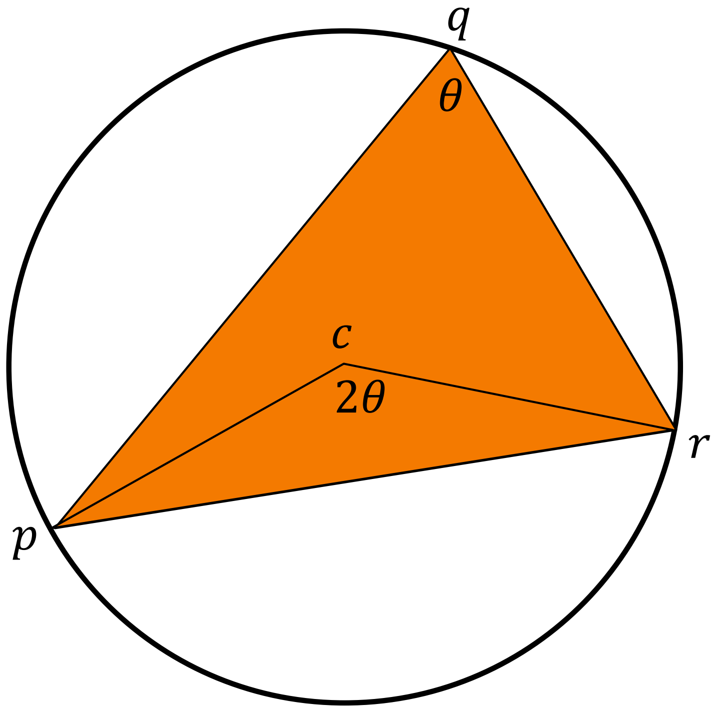
The list of triangles met along the line segment (in order) satisfies the desired properties.



Preliminaries

<u>Inscribed Angle Theorem</u>:

If a triangle Δpqr is inscribed in a circle with center c, $\angle pqr = \frac{1}{2}\angle pcr$.

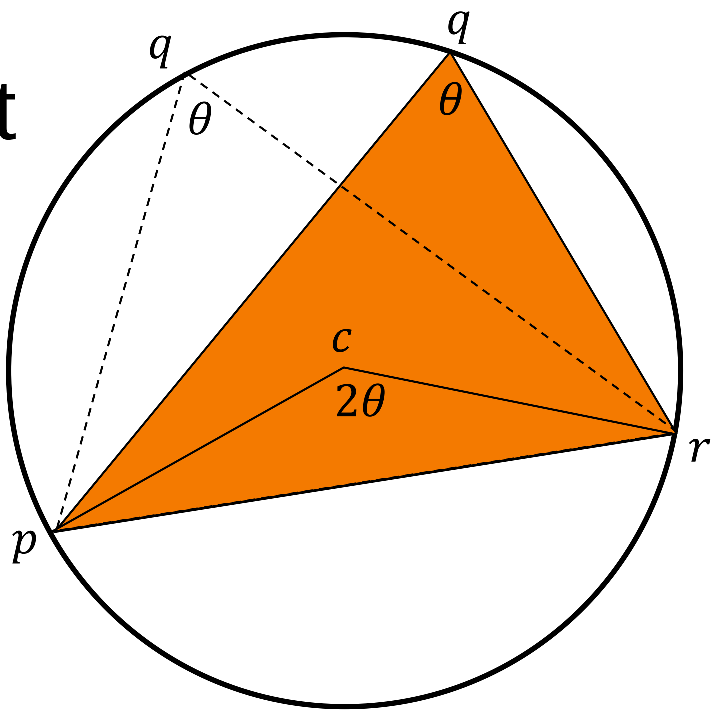


Preliminaries

Inscribed Angle Theorem:

If a triangle Δpqr is inscribed in a circle with center c, $\angle pqr = \frac{1}{2}\angle pcr$.

The angle $\angle pqr$ does not depend on where q is on the circle.

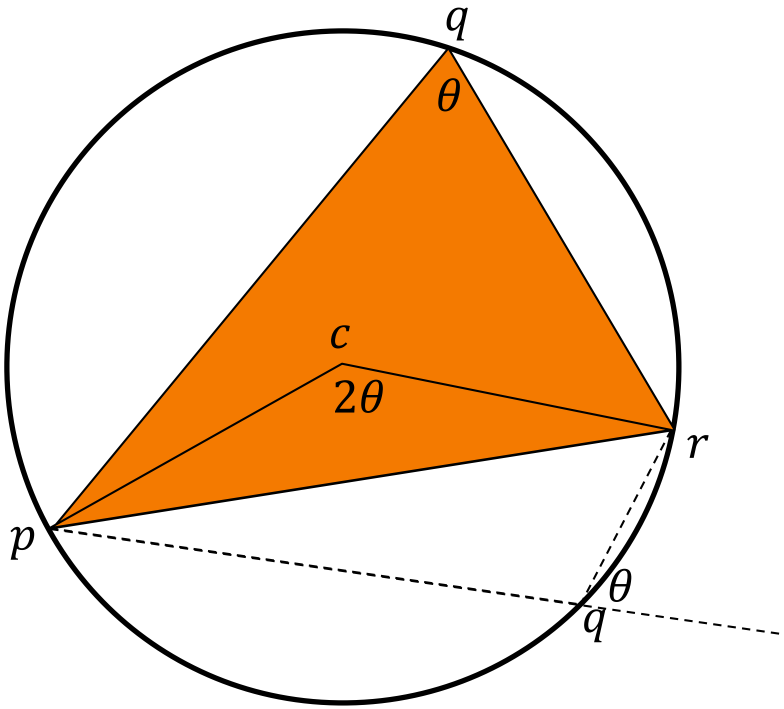


Preliminaries

Inscribed Angle Theorem:

If a triangle Δpqr is inscribed in a circle with center c, $\angle pqr = \frac{1}{2}\angle pcr$.

If the triangle does not contain c, the same is true if we take θ to be the exterior angle at q.



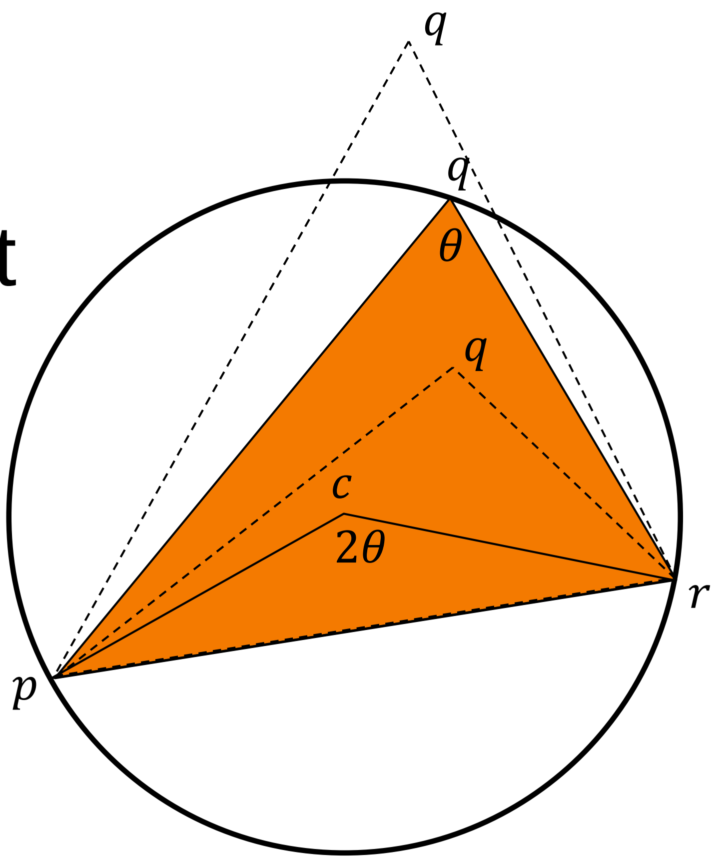
Preliminaries

Inscribed Angle Theorem:

If a triangle Δpqr is inscribed in a circle with center c, $\angle pqr = \frac{1}{2}\angle pcr$.

The angle $\angle pqr$ does not depend on where q is on the circle.

If q is inside/outside the circle the angle is larger/smaller.

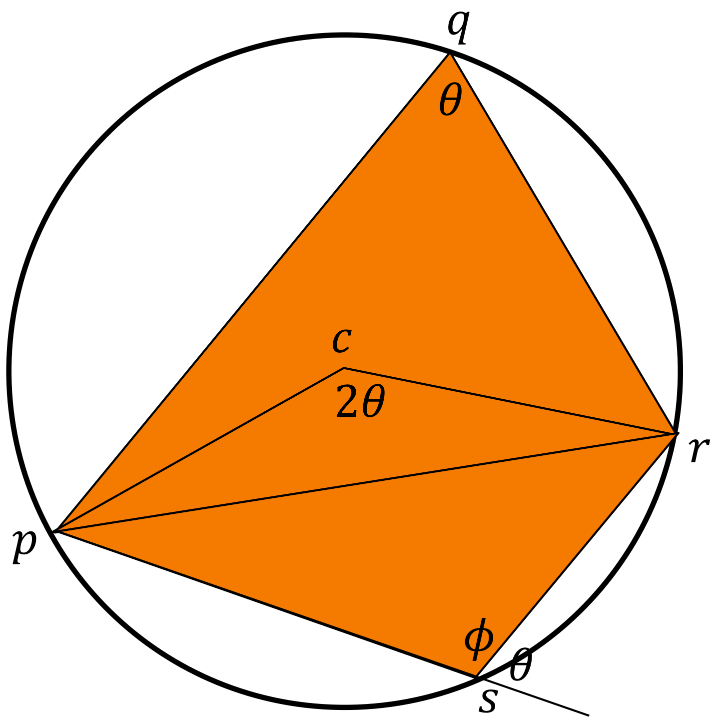


Preliminaries

In Particular:

If Δprs is a triangle with s on the circle and on the other side of the edge $\overline{pr}$ we get:

$$\angle psr = \pi - \angle pqr$$
$$\Downarrow$$
$$\angle psr + \angle pqr = \pi.$$



Outline

- Preliminaries
- Properties and Applications
  - Largest Empty Circle
  - Euclidean Minimal Spanning Tree
  - Locally Delaunay
  - Best Triangulation
- Computing the Delaunay Triangulation

Largest Empty Circle

<u>Claim</u>:

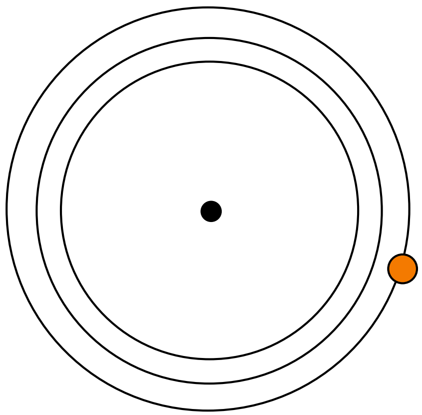
The largest empty (interior) circle, centered within the convex hull of a set of points is either centered at a Voronoi vertex or at the intersection of the Voronoi Diagram and the convex hull.

Largest Empty Circle

Proof:

A maximal circle centered in the interior must be adjacent to a point.

Otherwise, grow the radius to make the circle larger.

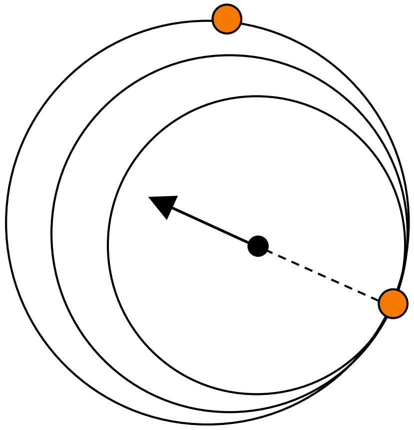


Largest Empty Circle

<u>Proof</u>:

A maximal circle centered in the interior must be adjacent to at least two points.

Otherwise, move out along the ray from the one point to the center while increasing the radius.



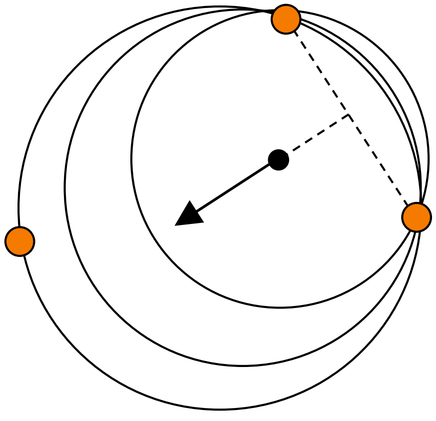
Largest Empty Circle

Proof:

A maximal circle centered in the interior must be adjacent to at least three points.

Otherwise, move out along the bisector along one of the two directions while increasing the radius.

⇒ Maximal circles in the interior are centered on Voronoi vertices.



Largest Empty Circle

Proof:

A maximal circle centered on the hull has to be in the interior of a hull edge.

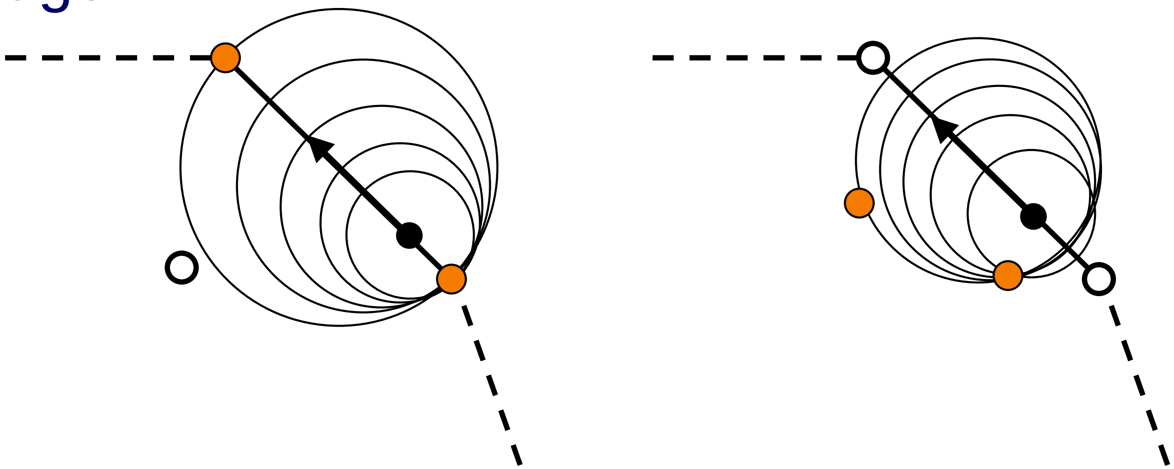
Otherwise, it's on a hull vertex and the radius is zero.

Largest Empty Circle

Proof:

A maximal circle centered on the hull must be adjacent to two points.

- Otherwise, move out along the hull along one of the two directions while increasing the radius
- When you stop, you are on the hull and on a Voronoi edge.



Minimal Spanning Trees

Definition:

Given a connected, undirected graph with weighted edges, the *minimal spanning tree* (*MST*) is the tree with minimal edge length that spans all the points.

Minimal Spanning Trees

Kruskal($G = \left(V, E, \omega\colon E \to \mathbb{R}^{>0}\right)$):

$Q \leftarrow$ SortByDecreasingLength(E , ω)

$C \leftarrow V$

$T \leftarrow \emptyset$

while($|C| > 1$)

$e = (v, w) \leftarrow Q$

if(Disconnected(C , v , w)):

Merge(C , v , w)

$T \leftarrow T \cup \{e\}$

Complexity: $O(|E|)$ using a union-find data-structure.

Euclidean Minimal Spanning Trees

Definition:

Given a set of points $P \subset \mathbb{R}^n$, the *Euclidean minimal spanning tree* (*EMST*) is the minimal spanning tree of the complete graph, with edge weights given by Euclidean distances.

Euclidean Minimal Spanning Trees

Claim:

The EMST is a sub-graph of $\mathcal{D}(P)$.

Implications:

We can find the EMST (in 2D) in $O(n \log n)$ by only running Kruskal's algorithm using the subset of edges in the Delaunay triangulation.

Euclidean Minimal Spanning Trees

<u>Proof</u>:

Assume $\overline{p_i p_j}$ is in the EMST but not in $\mathcal{D}(P)$.

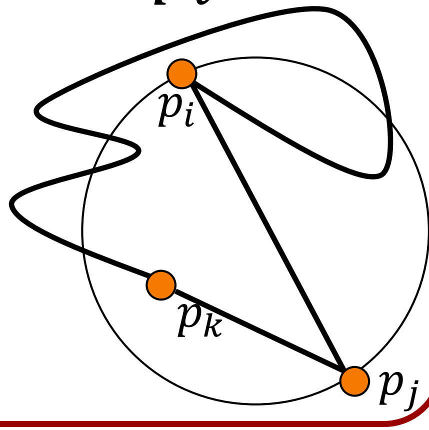
⇒ The circle with p_i and p_j on its <u>diameter</u> contains another point p_k.

⇒ Removing $\overline{p_i p_j}$ disconnects the EMST into two components, one with p_i and the other with p_j.

WLOG, assume p_k is in the component with p_i.

⇒ Adding edge $\overline{p_j p_k}$ reconnects the graph and gives a shorter spanning tree.

⇒ The original tree wasn't a MST.



Locally Delaunay

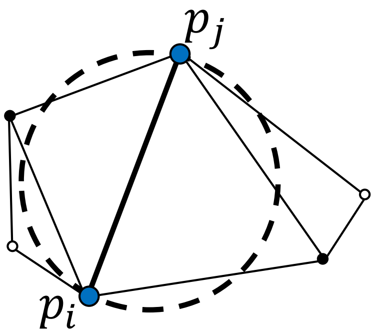
Recall:

Given a set of points $P = \{p_1, \ldots, p_n\}$ we say that an edge $\overline{p_i p_j}$ is Delaunay if there exists a circle with p_i and p_j on its boundary that is empty of other points.

Locally Delaunay

<u>Definition</u>:

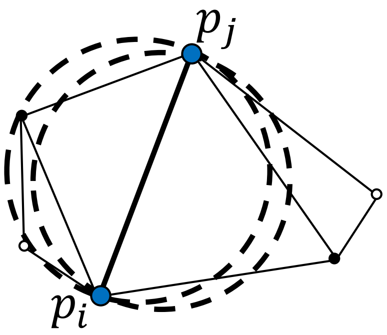
Given a triangulation $\mathcal{T}(P)$, we say that an edge of the triangulation, $\overline{p_i p_j}$, is *locally Delaunay* if there exists a circle with p_i and p_j on its boundary that does not contain the opposite vertices in the adjacent triangles.



Locally Delaunay

Note:

If the edge is locally Delaunay, we can always shift the circle so that it just touches one of the adjacent vertices and does not contain the other.

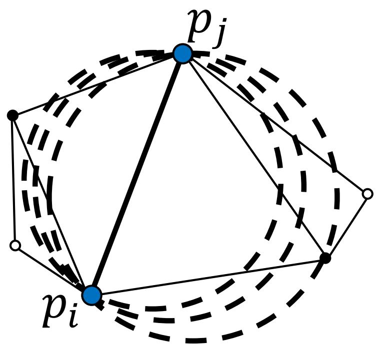


Locally Delaunay

<u>Note</u>:

If the edge is locally Delaunay, we can always shift the circle so that it just touches one of the adjacent vertices and does not contain the other.

⇒ An edge is locally Delaunay if and only if the circumcircle of one adjacent triangle does not contain the opposite vertex in the other.



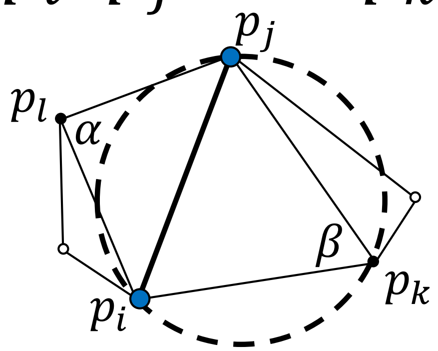
Locally Delaunay

Note:

An edge is locally Delaunay, if and only if the sum of the opposite angles satisfies:

$$\alpha + \beta \leq \pi.$$

If p_l were on the circumcircle through p_i, p_j, and p_k, then we would have $\alpha + \beta = \pi$.



Moving p_l outside the circle reduces α.

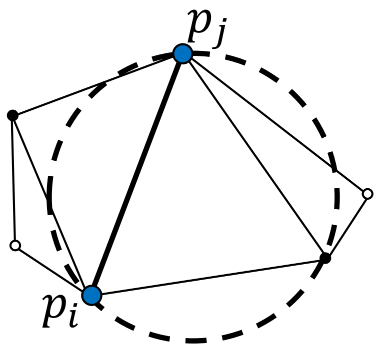
Locally Delaunay

Claim:

A triangulation $\mathcal{T}(P)$ is Delaunay if and only if it is locally Delaunay.

Implications:

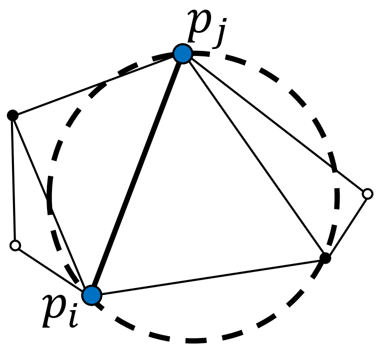
We can test if a triangulation is Delaunay in linear time by testing if each edge is locally Delaunay.



Locally Delaunay

Proof ($\Rightarrow$):

Trivial.

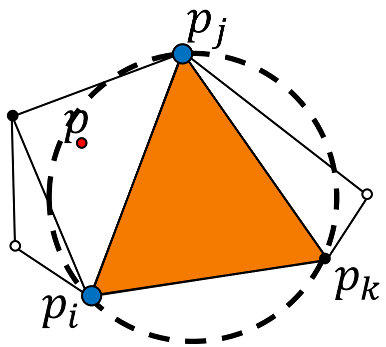


Locally Delaunay

Proof ($\Leftarrow$) [By Induction]:

Assume it is not Delaunay.

$\Rightarrow$ There exists a triangle $\Delta p_i p_j p_k \in \mathcal{T}(P)$ and a point $p \in P$ that is inside the circumcircle of $\Delta p_i p_j p_k$.

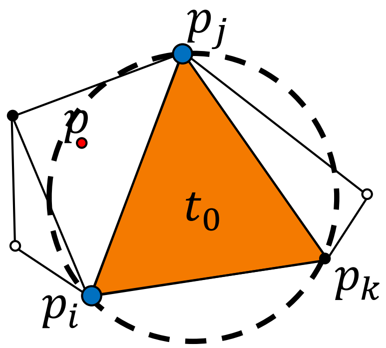


Locally Delaunay

Proof (⇐) [By Induction]:

Choose edge-adjacent triangles $\{t_0, \ldots, t_m\}$ s.t.:

- $t_0 = \Delta p_i p_j p_k$ and $p \in t_m$.
- if $e \in t_l \cap t_{l+1}$ then t_{l+1} is on the same side of e as p.



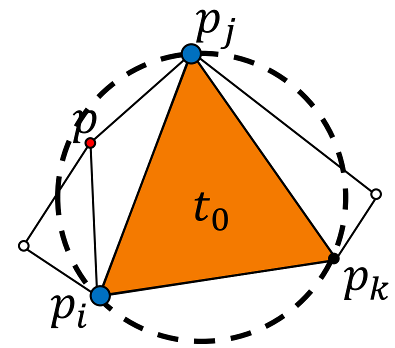
Locally Delaunay

Proof (⇐) [By Induction]:

Choose edge-adjacent triangles $\{t_0, \ldots, t_m\}$ s.t.:

- $t_0 = \Delta p_i p_j p_k$ and $p \in t_m$.
- if $e \in t_l \cap t_{l+1}$ then t_{l+1} is on the same side of e as p.

If $m = 1$ then $\mathcal{T}(P)$ is not locally Delaunay.



Locally Delaunay

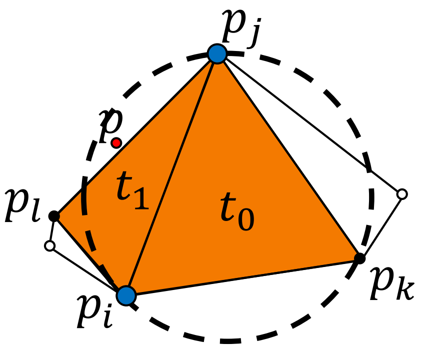
<u>Proof (⇐) [By Induction]</u>:

Choose edge-adjacent triangles $\{t_0, \dots, t_m\}$ s.t.:

- $t_0 = \Delta p_i p_j p_k$ and $p \in t_m$.
- if $e \in t_l \cap t_{l+1}$ then t_{l+1} is on the same side of e as p.

If $m = 1$ then $\mathcal{T}(P)$ is not locally Delaunay.

Set $t_1 = \Delta p_i p_j p_l$.



Locally Delaunay

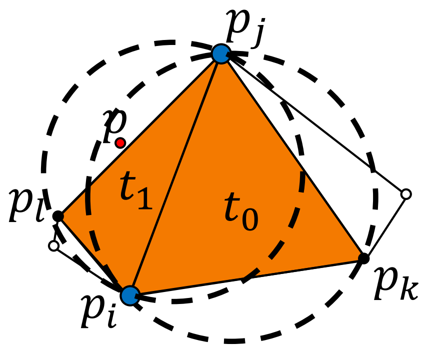
Proof (⇐) [By Induction]:

Choose edge-adjacent triangles $\{t_0, \ldots, t_m\}$ s.t.:

- $t_0 = \Delta p_i p_j p_k$ and $p \in t_m$.
- if $e \in t_l \cap t_{l+1}$ then t_{l+1} is on the same side of e as p.

If $m = 1$ then $\mathcal{T}(P)$ is not locally Delaunay.

Set $t_1 = \Delta p_i p_j p_l$.



Since p_l is outside the circumcircle of t_1 and on the same side as p, the circumcircle of t_2 contains the part of the circumcircle of t_1 that is outside t_1 and contains p.

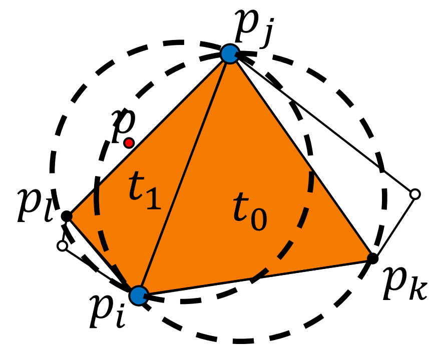
Locally Delaunay

Proof (⇐) [By Induction]:

We can repeat with triangle $\Delta p_i p_j p_l$,
but now the sequence of triangles is one shorter.

If $m = 1$ then $\mathcal{T}(P)$ is not locally Delaunay.

Set $t_1 = \Delta p_i p_j p_l$.



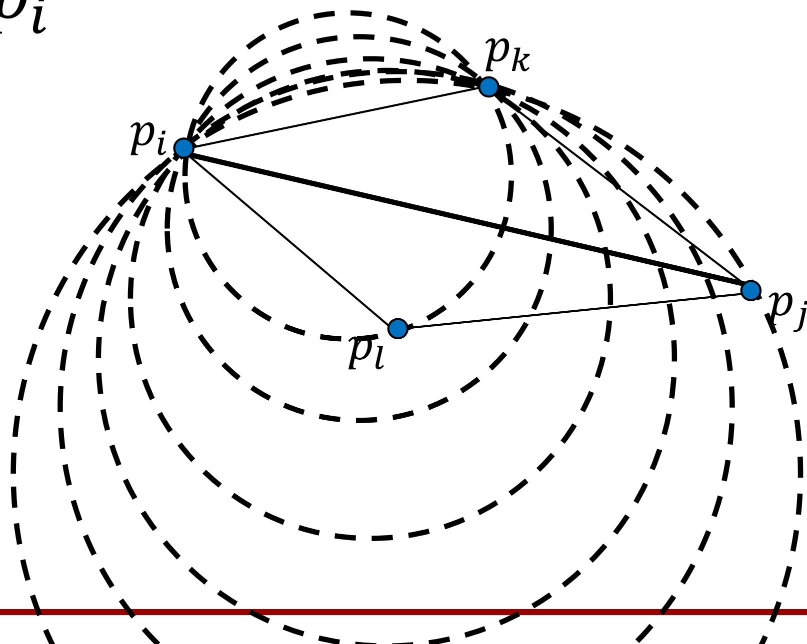
Since p_l is outside the circumcircle of t_1 and on the same side as p, the circumcircle of t_2 contains the part of the circumcircle of t_1 that is outside t_1 and contains p.

Locally Delaunay

<u>Note</u>:

If an edge $\overline{p_i p_j}$ of a triangulation is not locally Delaunay, the circle through p_i, p_j, and an opposite vertex p_k, must contain the other vertex p_l.

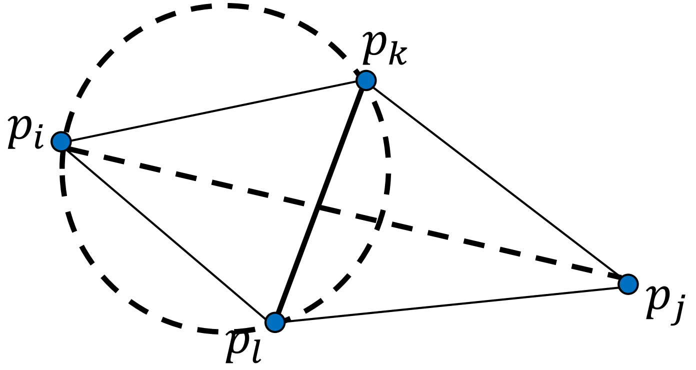
⇒ We can pin the circle at p_i and p_k and shrink it until it contains p_l.



Locally Delaunay

Note:

If an edge $\overline{p_i p_j}$ of a triangulation is not locally Delaunay, the circle through p_i, p_j, and an opposite vertex p_k, must contain the other vertex p_l.



⇒ We can pin the circle at p_i and p_k and shrink it until it contains p_l.

⇒ p_j is not inside the circle.

⇒ $\overline{p_l p_k}$ is locally Delaunay.

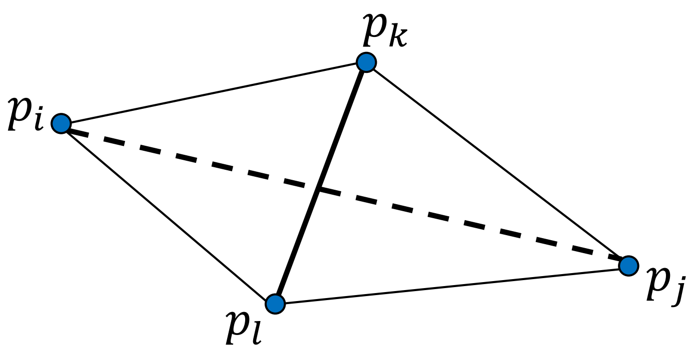
Locally Delaunay

Note:

If an edge $\overline{p_i p_j}$ of a triangulation is not locally Delaunay, the circle through p_i, p_j, and an opposite vertex p_k, must contain the other vertex p_l.

$\Rightarrow$ We can pin the circle at p_i and p_k and shrink it until it contains p_l.

$\Rightarrow$ p_j is not inside the circle.



We can perform an *edge-flip* to change a non-locally Delaunay edge into a locally Delaunay edge.

Locally Delaunay

Equivalently:

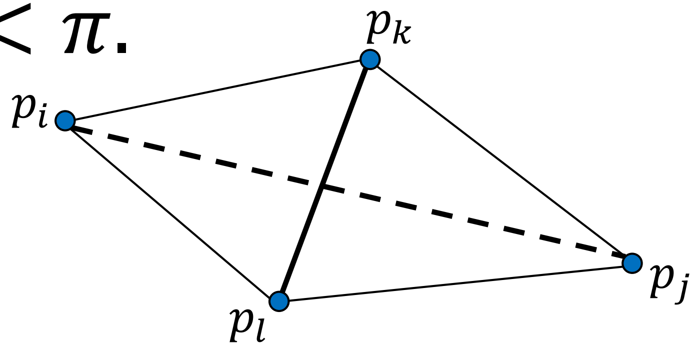
An edge $\overline{p_i p_j}$ is not locally Delaunay iff.:

$$\angle p_i p_k p_j + \angle p_i p_l p_j > \pi.$$

But the sum of the angles of a quad is 2π so:

$$\angle p_l p_i p_k + \angle p_l p_j p_k < \pi.$$

So the flipped edge $\overline{p_l p_k}$ must be Delaunay.

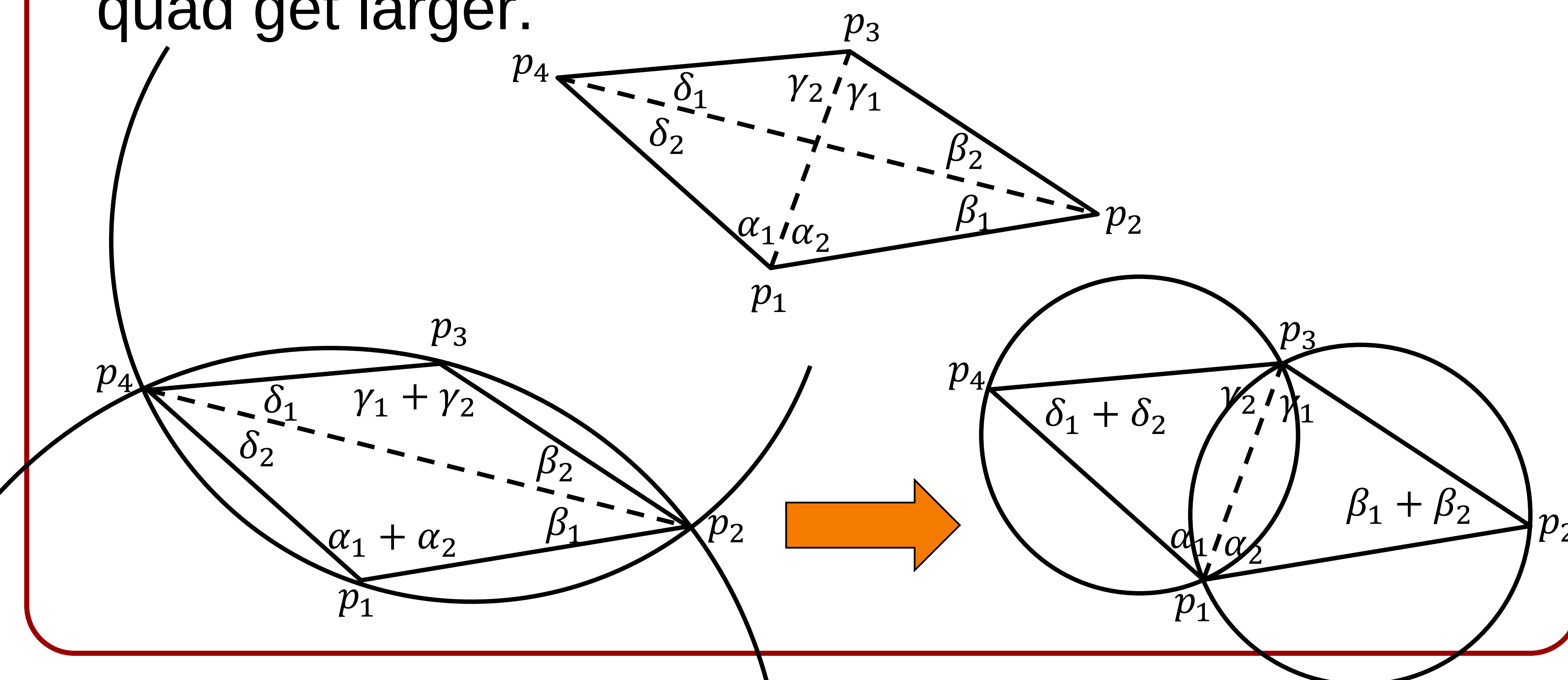


We can perform an *edge-flip* to change a non-locally Delaunay edge into a locally Delaunay edge.

Locally Delaunay

Claim:

If we edge-flip a non-locally Delaunay edge into a locally Delaunay edge the smallest angle in the quad get larger.

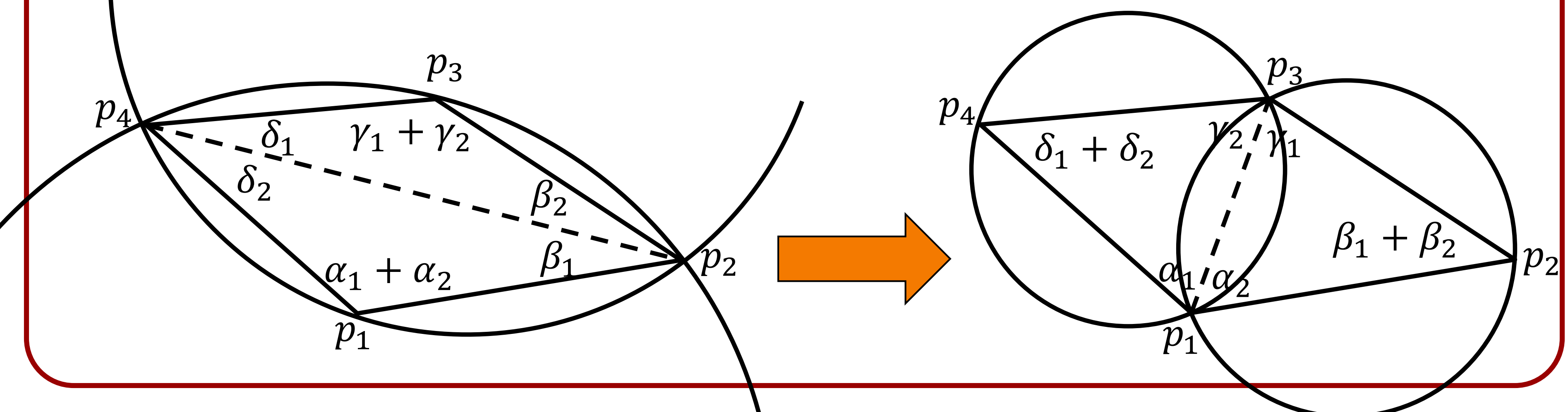


Locally Delaunay

Claim:

If we edge-flip a non-locally Delaunay edge into a locally Delaunay edge the smallest angle in the quad get larger.

- Trivially:
  - $\delta_1 + \delta_2 > \delta_1$
  - $\beta_1 + \beta_2 > \beta_1$

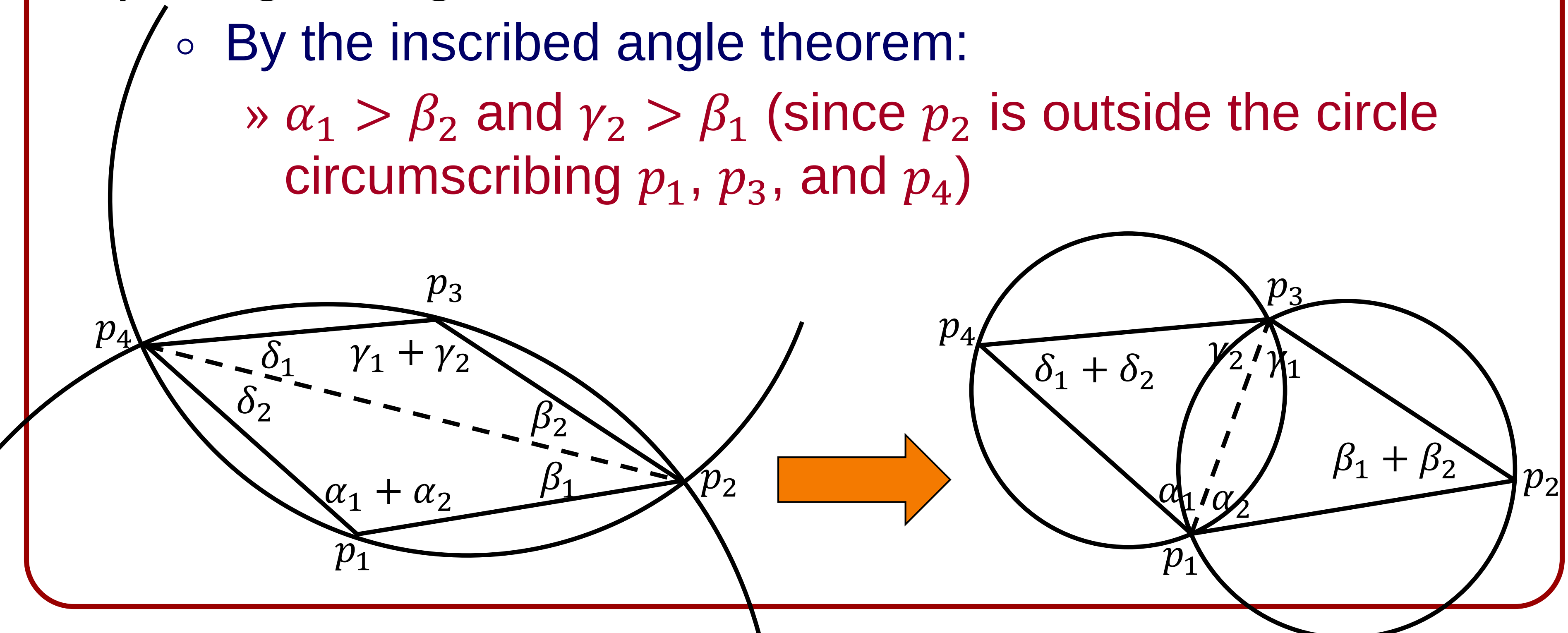


Locally Delaunay

Claim:

If we edge-flip a non-locally Delaunay edge into a locally Delaunay edge the smallest angle in the quad get larger.

- By the inscribed angle theorem:
  - $\alpha_1 > \beta_2$ and $\gamma_2 > \beta_1$ (since p_2 is outside the circle circumscribing p_1, p_3, and p_4)

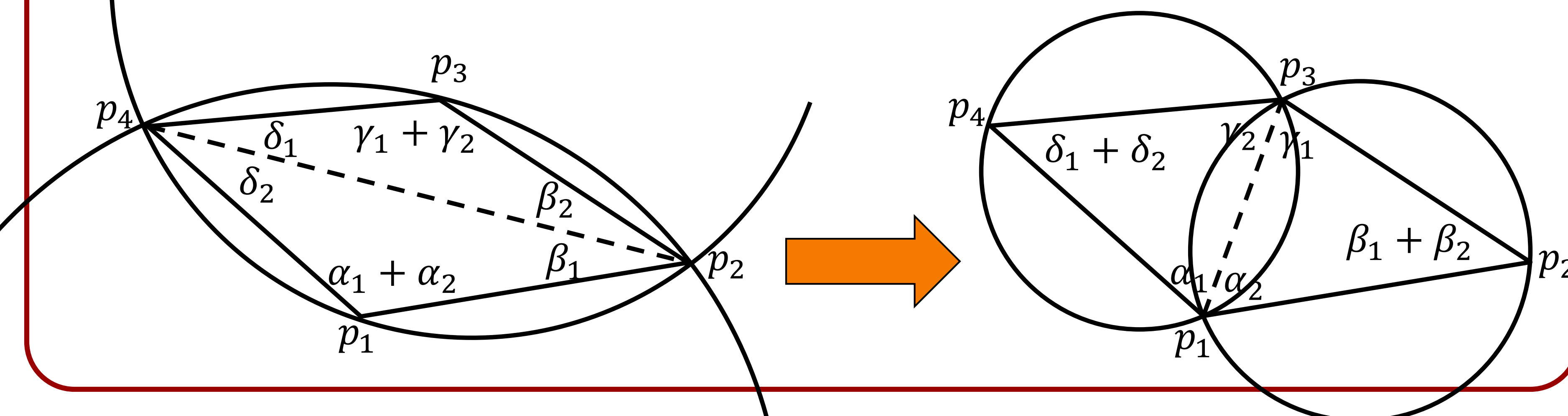


Locally Delaunay

<u>Claim</u>:

If we edge-flip a non-locally Delaunay edge into a locally Delaunay edge the smallest angle in the quad get larger.

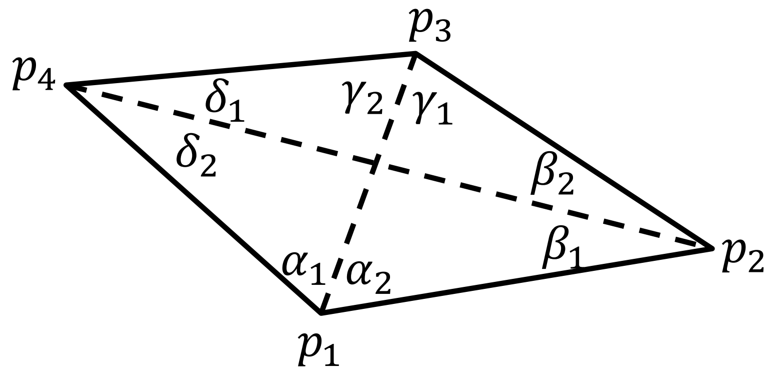
- By the inscribed angle theorem:
  - $\alpha_2 > \delta_1$ and $\gamma_1 > \delta_2$ (since p_4 is outside the circle circumscribing p_1, p_2, and p_3)



Locally Delaunay

Note:

If we edge-flip a non-locally Delaunay edge into a locally Delaunay edge all angles of the triangulation exterior to the quad are unchanged.



Best Triangulation

Definition:

Given a set of points $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$ and given a triangulation $\mathcal{T}$ of P define the *angle vector* of the triangulation, $\vec{\alpha}^{\mathcal{T}} \in (0, \pi)^{3T}$, to be the sorted angles of the triangles in the triangulation:

$$\alpha_i^{\mathcal{T}} \leq \alpha_{i+1}^{\mathcal{T}}.$$

We can define an ordering on triangulations of P by saying that for triangulations $\mathcal{S}$ and $\mathcal{T}$, $\mathcal{S} > \mathcal{T}$ if $\mathcal{S}$ is larger than $\mathcal{T}$, lexicographically.

Best Triangulation

Claim:

Given a set of points $P = \{p_1, \ldots, p_n\}$ the Delaunay triangulation, $\mathcal{D}$, is maximal over all triangulations:

$$\mathcal{D} \geq \mathcal{T}$$

for all triangulations $\mathcal{T}$ of P.

Best Triangulation

Proof:

Suppose that the maximal triangulation $\mathcal{T}$ is not Delaunay.

$\Rightarrow$ There is an edge that is not locally Delaunay.

$\Rightarrow$ Flipping the edge will increase the angles interior to the quad.

$\Rightarrow$ The new triangulation will be larger than $\mathcal{T}$.

$\Rightarrow$ $\mathcal{T}$ was not maximal.

Outline

- Preliminaries
- Properties and Applications
- Computing the Delaunay Triangulation
  - Edge-Flipping
  - Reduction to Convex Hulls

Edge-Flipping

DelaunayTriangulation($P \subset \mathbb{R}^2$):

$\mathcal{T} \leftarrow$ Triangulate(P)

$Q \leftarrow \emptyset$

for $e \in E(\mathcal{T})$

if(!LocallyDelaunay(e)) $Q \leftarrow Q \cup \{e\}$

while(NotEmpty(Q))

$e \leftarrow$ Pop(Q)

if(!LocallyDelaunay(e)

Flip(e)

for $e' \in$Neighbor(e)

if(!LocallyDelaunay(e')) $Q \leftarrow Q \cup \{e'\}$

Edge-Flipping

DelaunayTriangulation($P \subset \mathbb{R}^2$):

$\mathcal{T} \leftarrow$ Triangulate(P)

Q

f

if(

while

$e \leftarrow$ Po

if(!Lo

Flip(e)

for $e' \in$Neighbor(e)

if(!LocallyDelaunay(e')) $Q \leftarrow Q \cup \{e'\}$

This requires being able to generate some initial (non-Delaunay) triangulation quickly.

This is guaranteed to converge since each iteration increases the angle vector.

Can show that this never requires more than $O(n^2)$ flips.

Edelsbrunner & Seidel

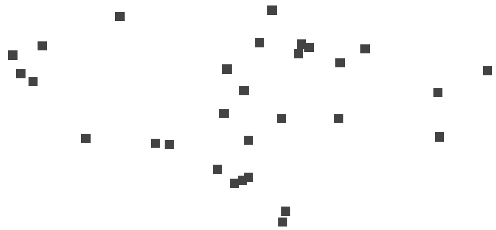
DelaunayTriangulation($P \subset \mathbb{R}^n$):

$Q \leftarrow \{q \in \mathbb{R}^{n+1} | q = (p, \|p\|^2)\}$

$C \leftarrow$ ConvexHull(Q)

$D \leftarrow$ ProjectLowerTriangles(C)

return D



Edelsbrunner & Seidel

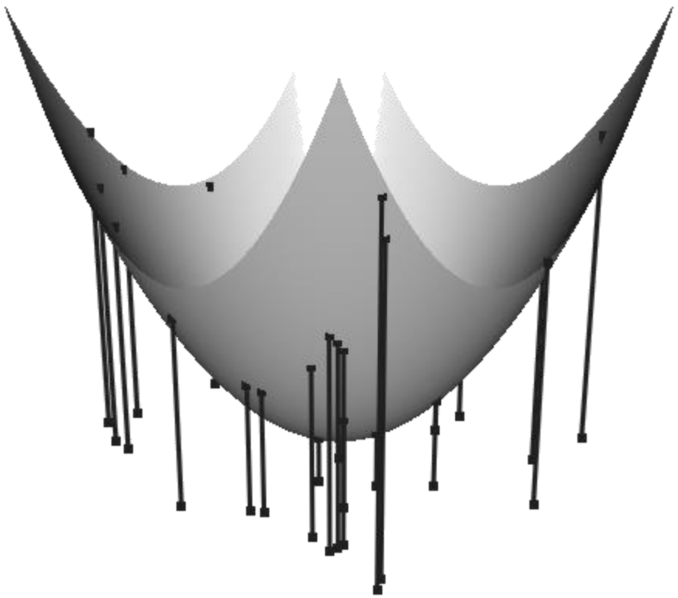
DelaunayTriangulation($P \subset \mathbb{R}^n$):

$\boldsymbol{Q} \leftarrow \{\boldsymbol{q} \in \mathbb{R}^{\boldsymbol{n}+\mathbf{1}} | \boldsymbol{q} = (\boldsymbol{p}, \|\boldsymbol{p}\|^2)\}$

$C \leftarrow$ ConvexHull(Q)

$D \leftarrow$ ProjectLowerTriangles(C)

return D



Edelsbrunner & Seidel

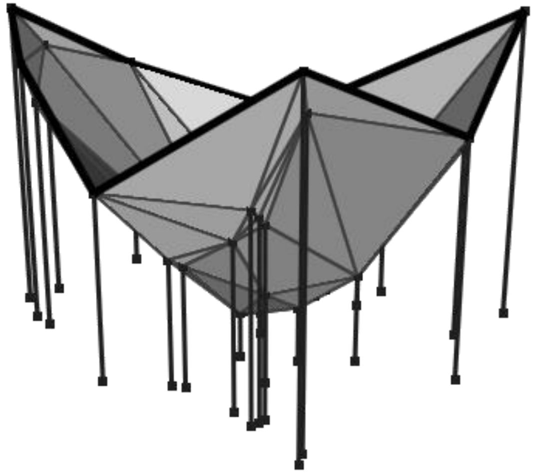
DelaunayTriangulation($P \subset \mathbb{R}^n$):

$Q \leftarrow \{q \in \mathbb{R}^{n+1} | q = (p, \|p\|^2)\}$

$C \leftarrow$ **ConvexHull(Q)**

$D \leftarrow$ ProjectLowerTriangles(C)

return D



Edelsbrunner & Seidel

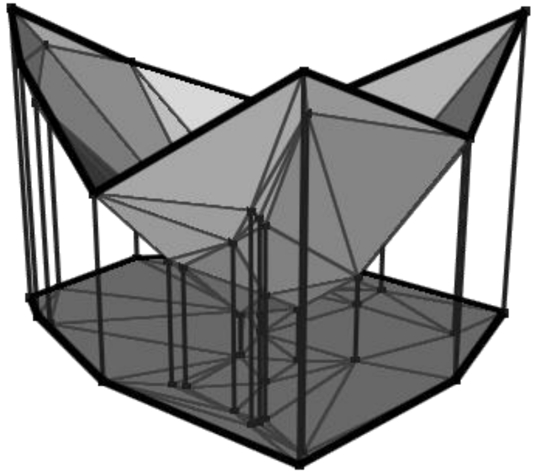
DelaunayTriangulation($P \subset \mathbb{R}^n$):

$Q \leftarrow \{q \in \mathbb{R}^{n+1} | q = (p, ||p||^2)\}$

$C \leftarrow$ ConvexHull(Q)

$D \leftarrow$ ProjectLowerTriangles(C)

return D



Edelsbrunner & Seidel

DelaunayTriangulation($P \subset \mathbb{R}^n$):

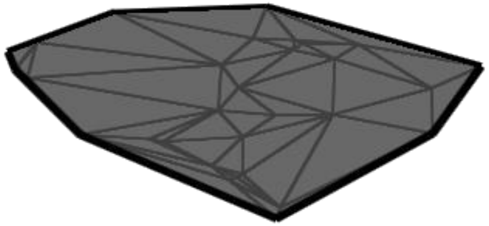
$Q \leftarrow \{q \in \mathbb{R}^{n+1} | q = (p, \|p\|^2)\}$

$C \leftarrow$ ConvexHull(Q)

$D \leftarrow$ ProjectLowerTriangles(C)

return D

Note:
Since all points end up on the hull, an output-sensitive convex hull algorithm does not help.



Edelsbrunner & Seidel

Correctness:

- Since the paraboloid is convex all points in Q end up on the lower hull of Q.
- The projection of the hull of Q is the hull of P.
- The projection of two edges on the convex hull can only intersect if one is on the top half and the other is on the bottom.

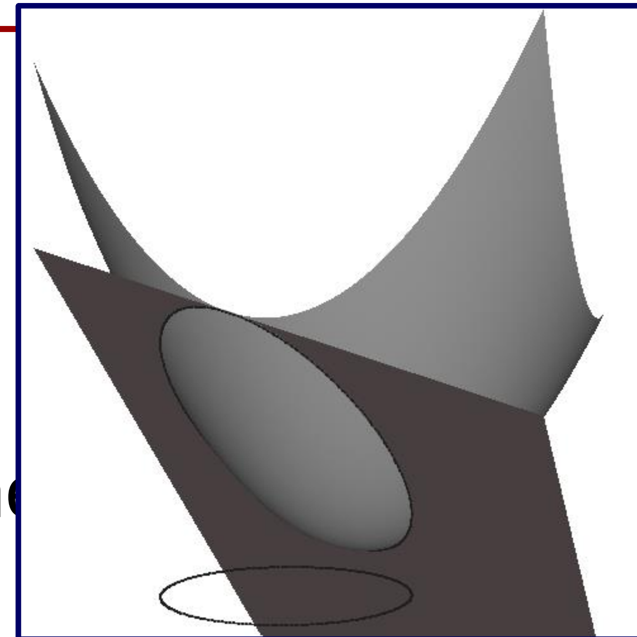
$\Rightarrow$ The projection is a triangulation of P.

Computation

Proof:

- Given a point $(a, b, a^2 + b^2)$ on the paraboloid, the tangent plane is given by:
$$z(x, y) = 2ax + 2by - (a^2 + b^2)$$
- Shifting the plane up by r^2 we get the plane:
$$z^{\uparrow}(x, y) = 2ax + 2by - (a^2 + b^2) + r^2$$
- The shifted plane intersects the paraboloid at:
$$z^{\uparrow}(x, y) = x^2 + y^2$$
$$\Leftrightarrow 2ax + 2by - (a^2 + b^2) + r^2 = x^2 + y^2$$
$$\Leftrightarrow (x - a)^2 + (y - b)^2 = r^2$$

Computation



Proof:

- Given a point $(a, b, a^2 + b^2)$ on the
the tangent plane is given by:
$$z(x, y) = 2ax + 2by - (a^2 + b^2)$$

The projection of the points of intersection onto the 2D plane is a circle with radius r around (a,b).

- Shi
$$z^{\uparrow}(x, y) = 2ax + 2by - (a^2 + b^2) + r^2$$

- The shifted plane intersects the paraboloid at:
$$z^{\uparrow}(x, y) = x^2 + y^2$$
$$\Leftrightarrow 2ax + 2by - (a^2 + b^2) + r^2 = x^2 + y^2$$
$$\Leftrightarrow (x - a)^2 + (y - b)^2 = r^2$$

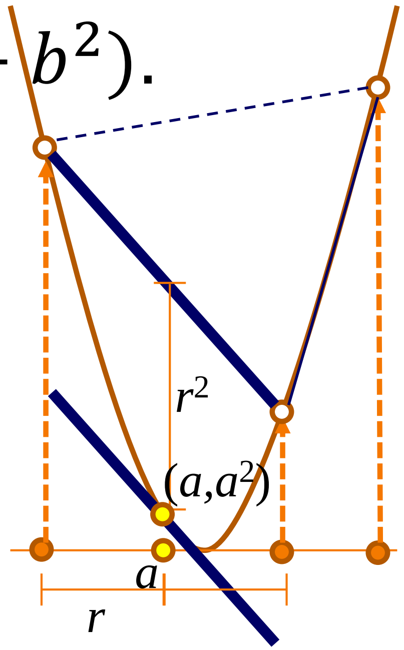
Computation

Proof:

If we have a triangle on the lower convex hull, we can pass a plane through the three vertices.

We can drop the plane by some r^2 so that it is tangent to the paraboloid at $(a, b, a^2 + b^2)$.

Then the projected vertices of the triangle must lie on a circle of radius r around the point (a, b).



Computation

Proof:

Since the original plane was on the lower hull, all other points must be above.

We can raise the plane until it intersects another point.

The distance from the projection of the point onto the 2D to (a, b) must be larger than r.

The circle of radius r around (a, b) contains no other points.