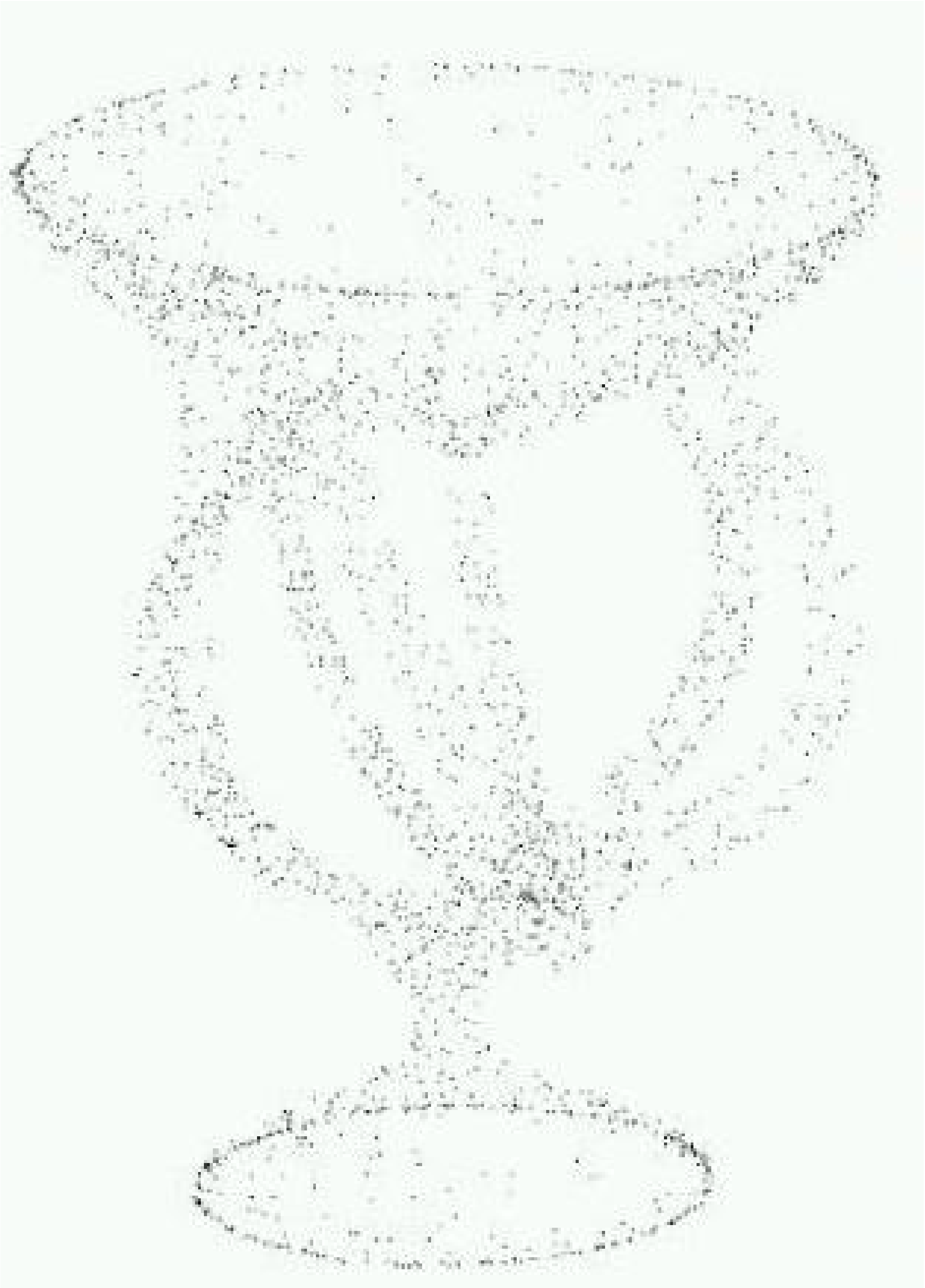
# Delaunay Triangulation: Applications

Reconstruction

Meshing

# Reconstruction

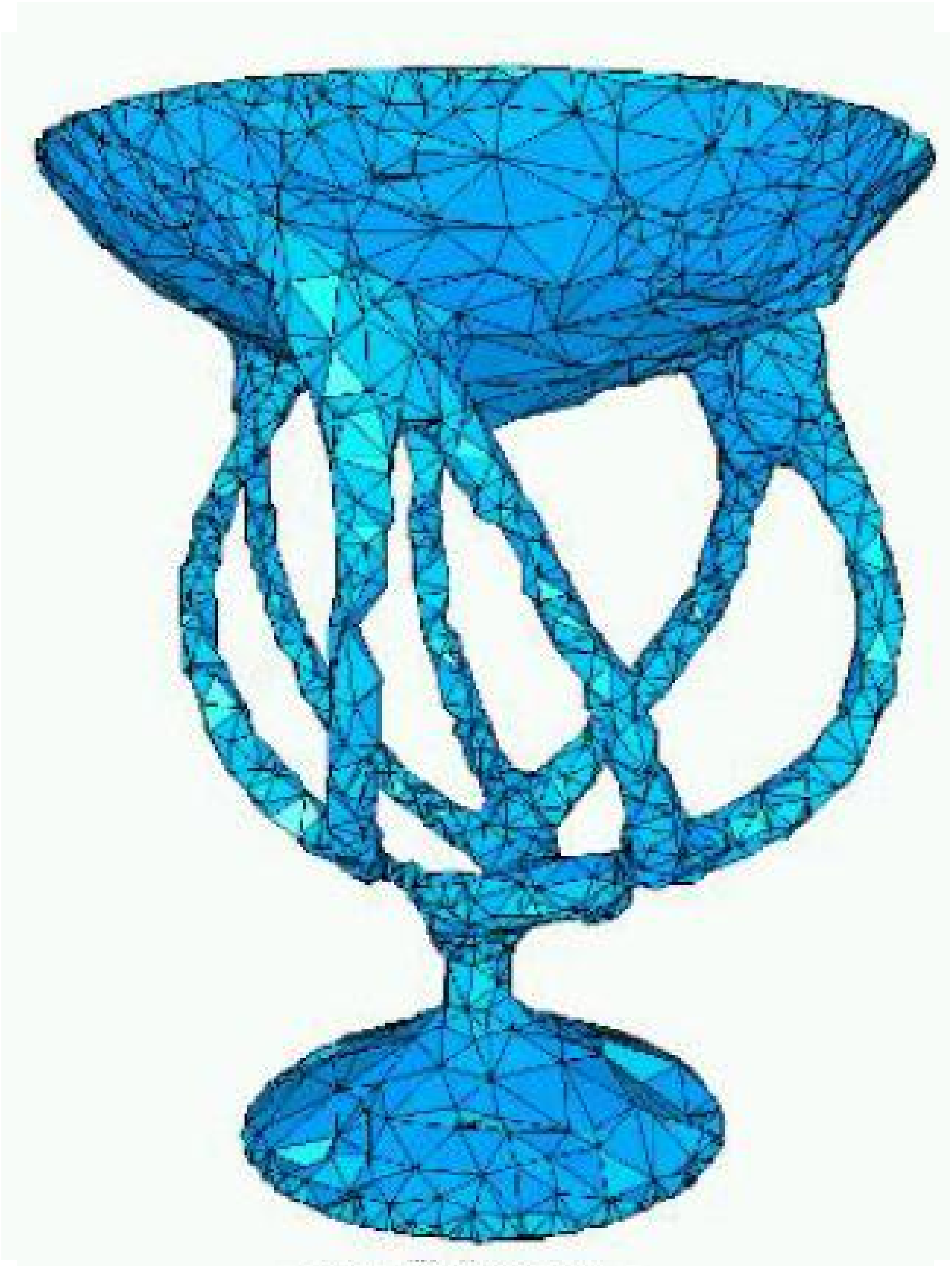
From points



# Reconstruction

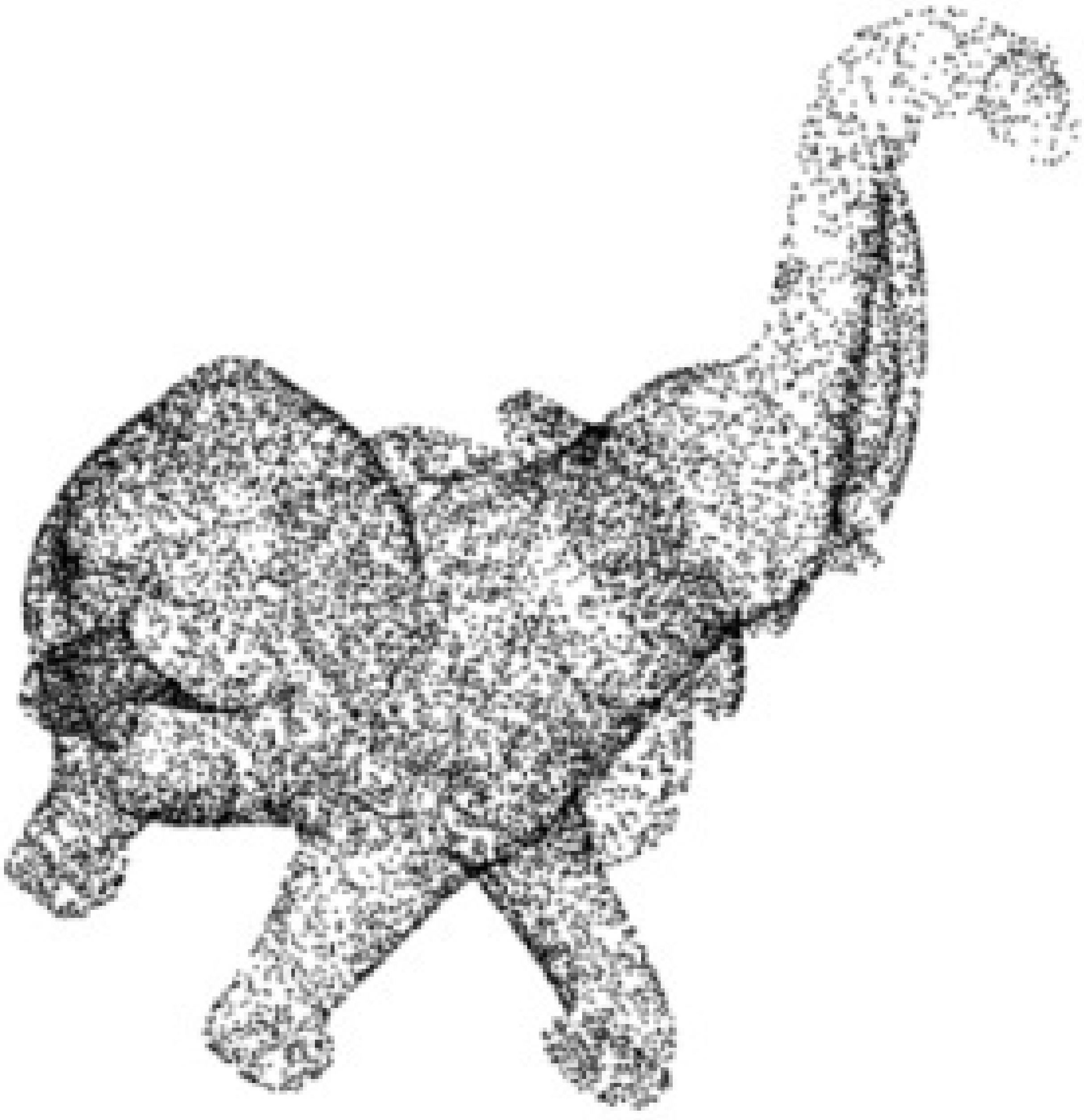
From points

to shape



# Reconstruction

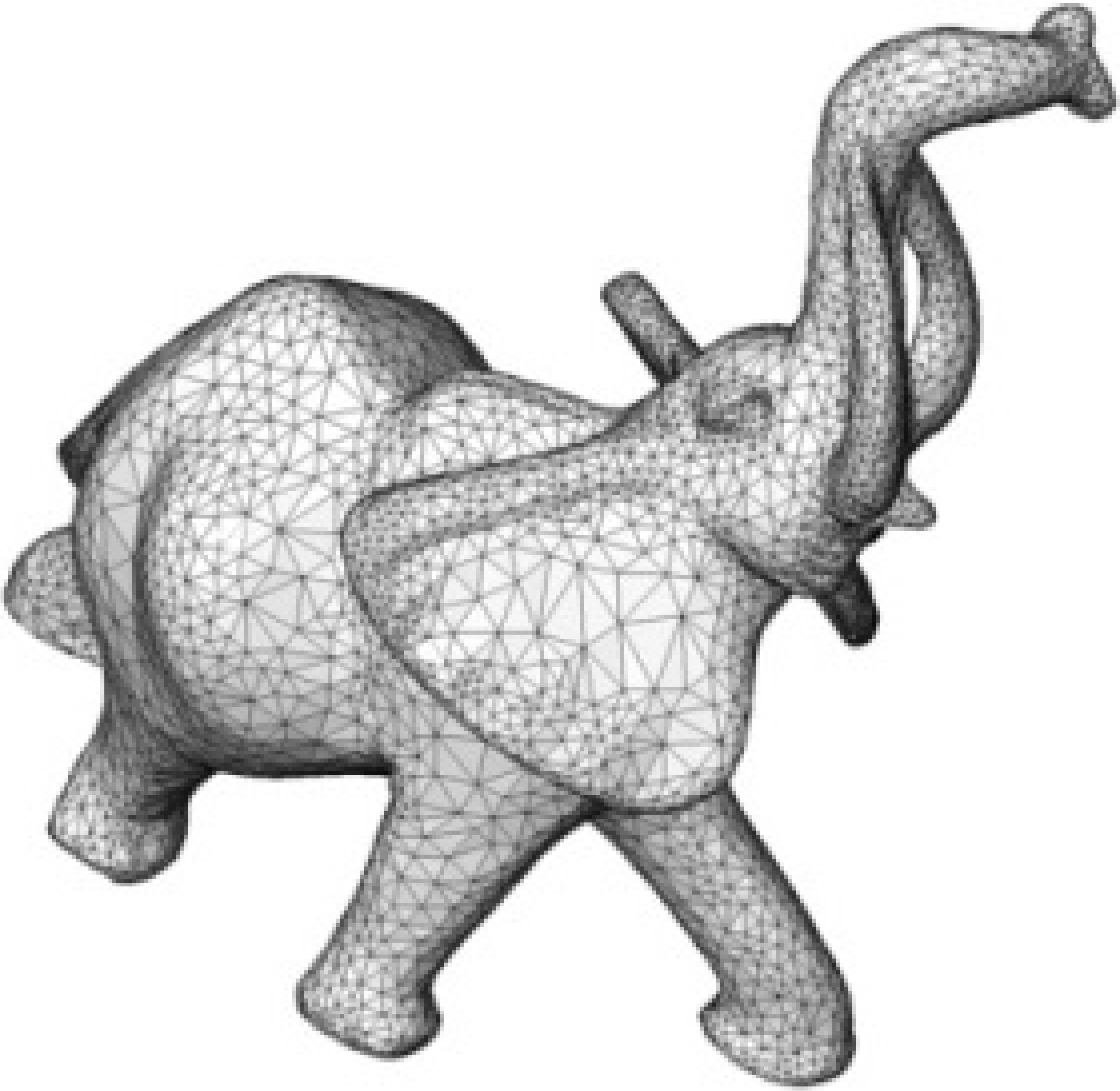
From points



# Reconstruction

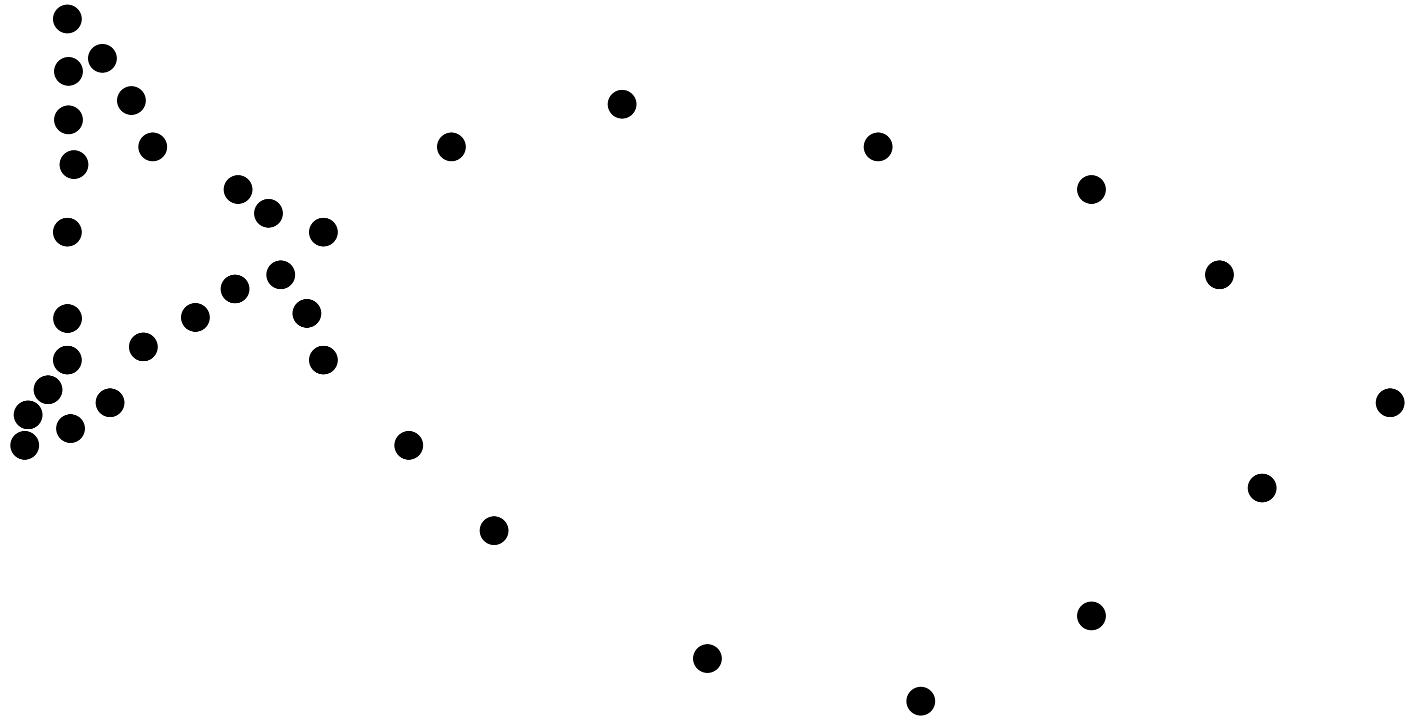
From points

to shape



# Reconstruction

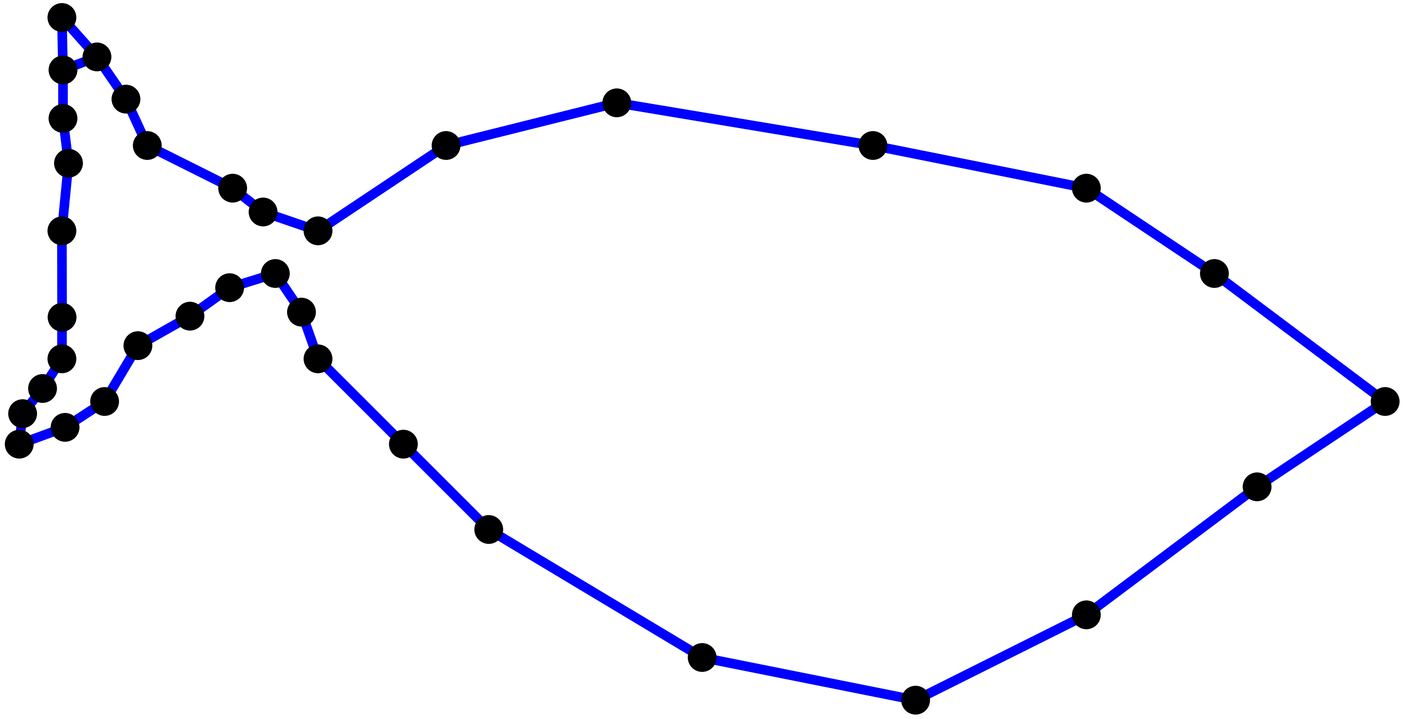
From points



# Reconstruction

From points

to shape



# Reconstruction

Context

Delaunay is a good start (wanted result $\subset$ Delaunay)

Crust 2D Algorithm

0.4 sample $\Rightarrow$ wanted result $\subset$ crust

0.25 sample $\Rightarrow$ crust $\subset$ wanted result

3D

# Reconstruction

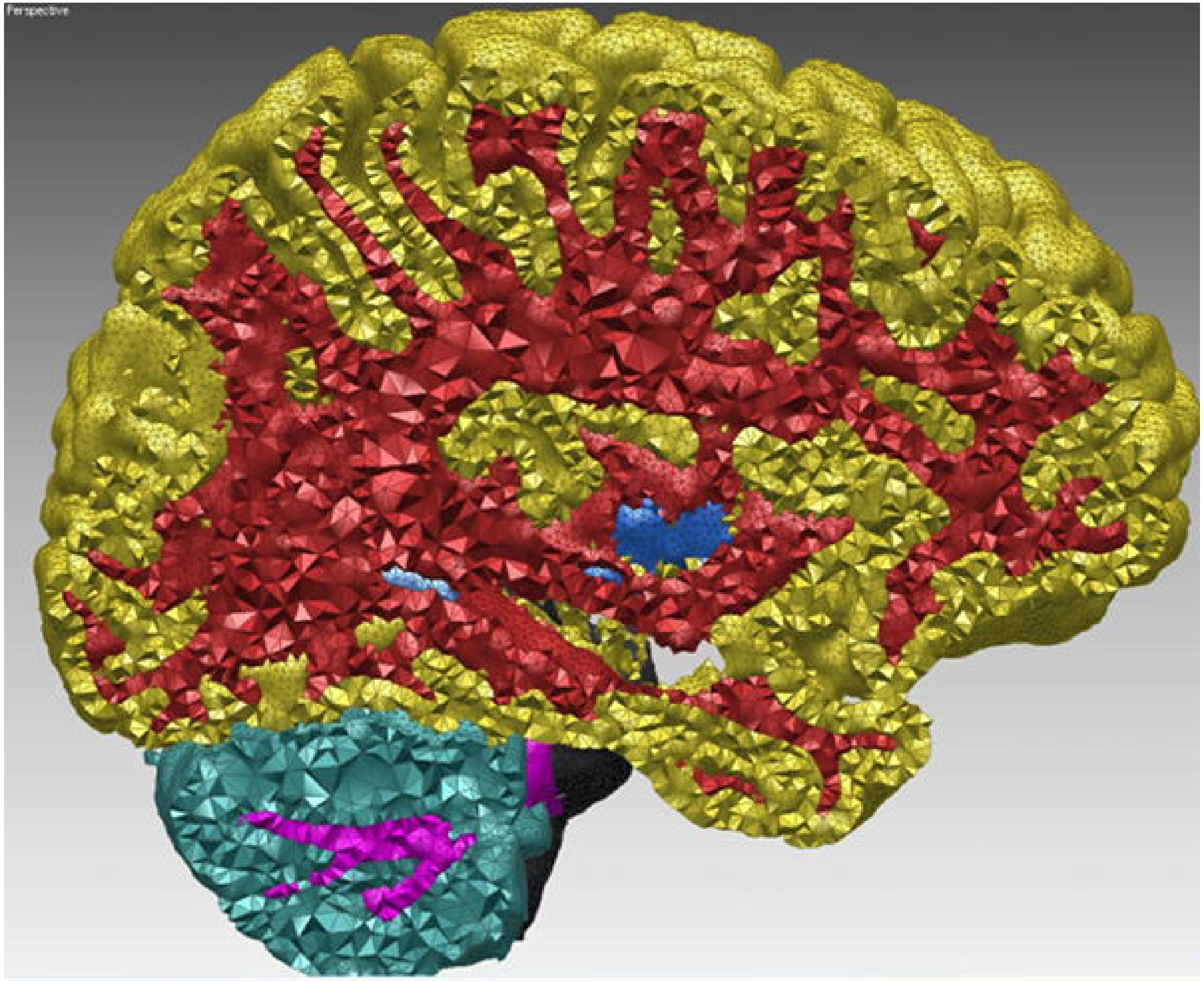
## Context

Sensor $\longrightarrow$ Point set (no structure or unknown)

# Reconstruction

## Context

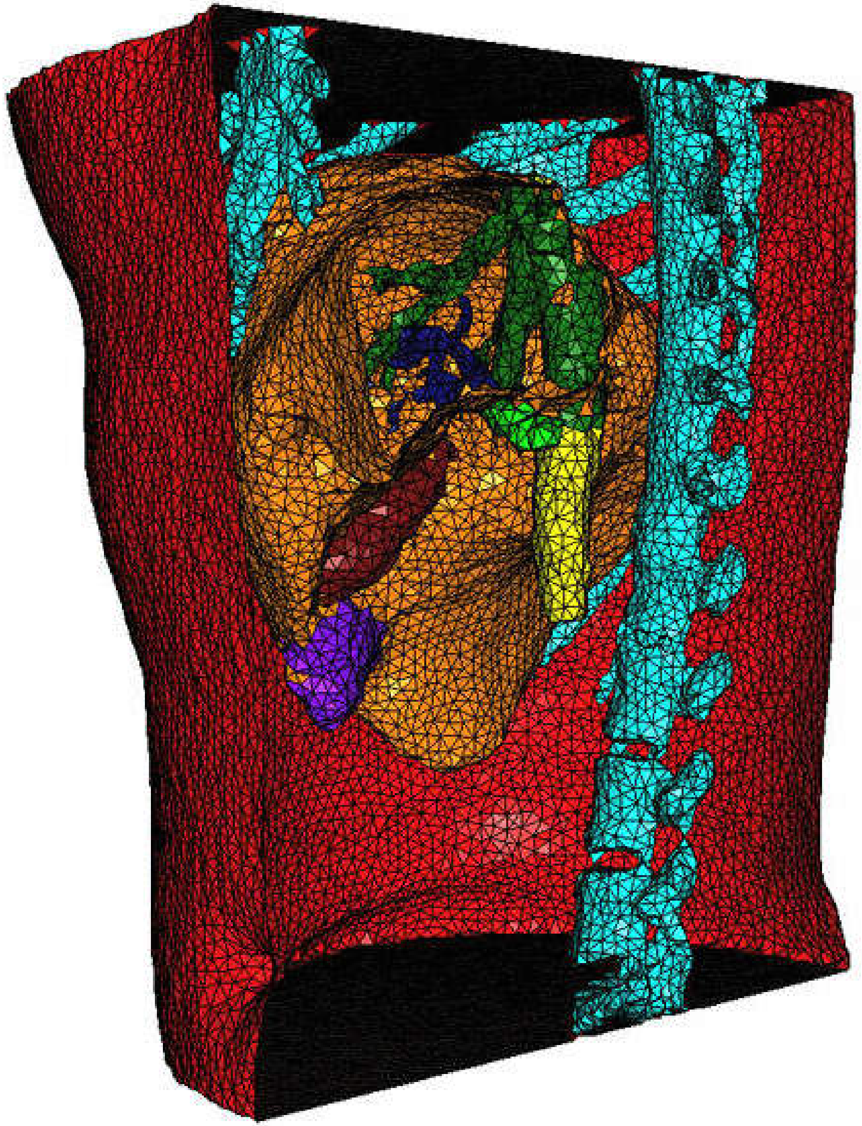
### Medical Images



# Reconstruction

## Context

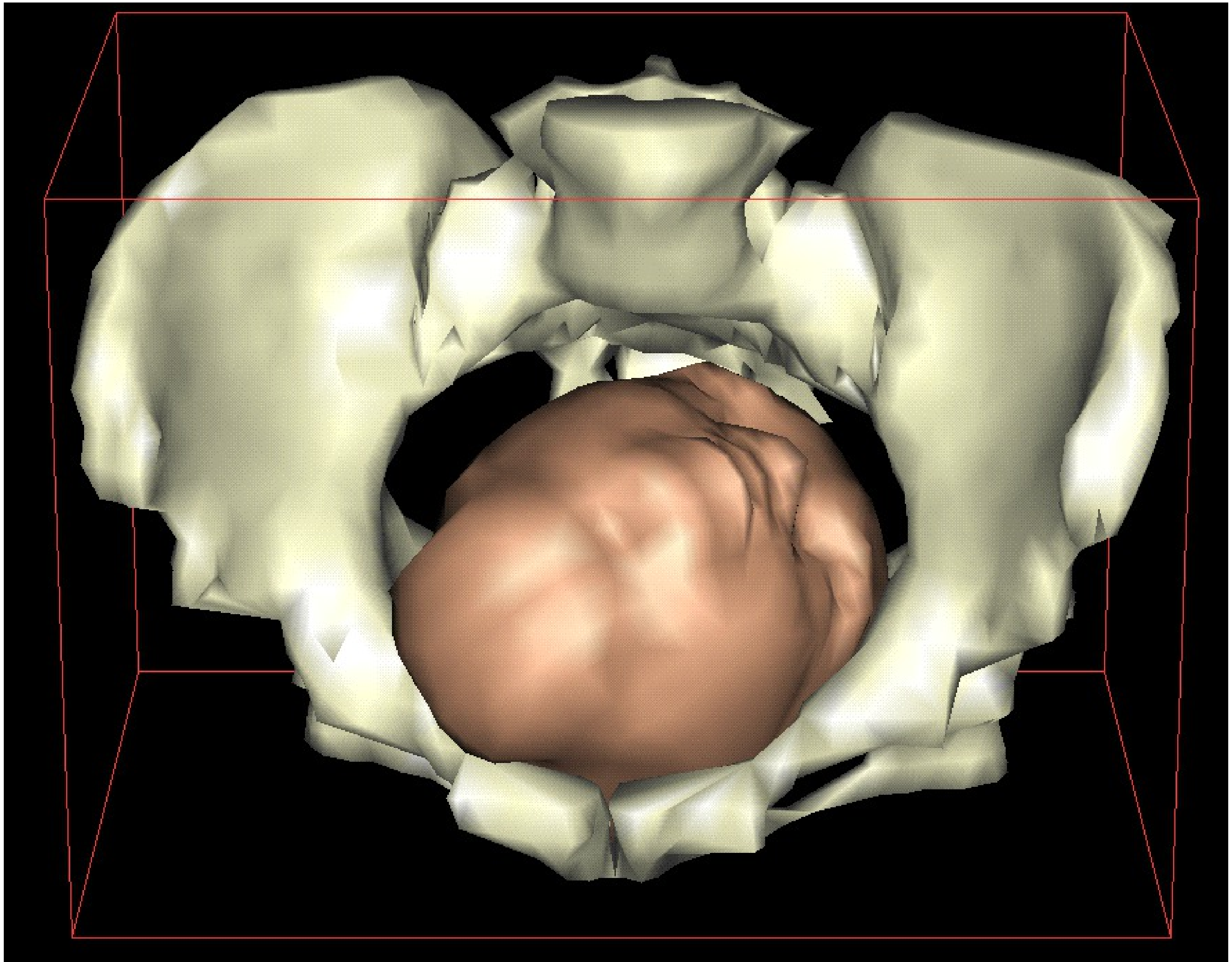
### Medical Images



# Reconstruction

Context

Childbirth simulation



# Reconstruction

## Context

Childbirth simulation

Surgery planning

Radiotherapy planing

Endoscopy simulation

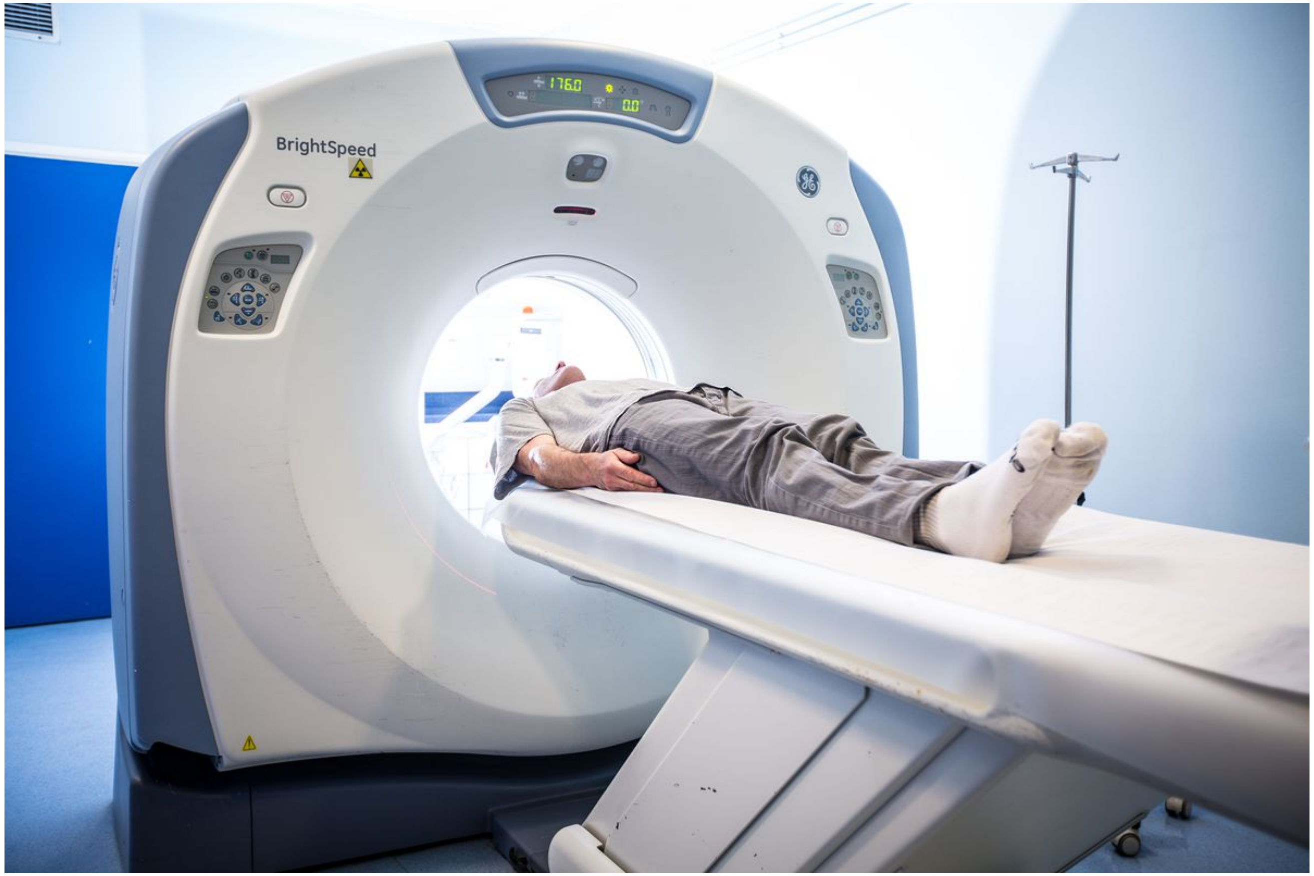
• • •

# Reconstruction

## Context

Sensor → Point set (no structure or unknown)

Scanner



# Reconstruction

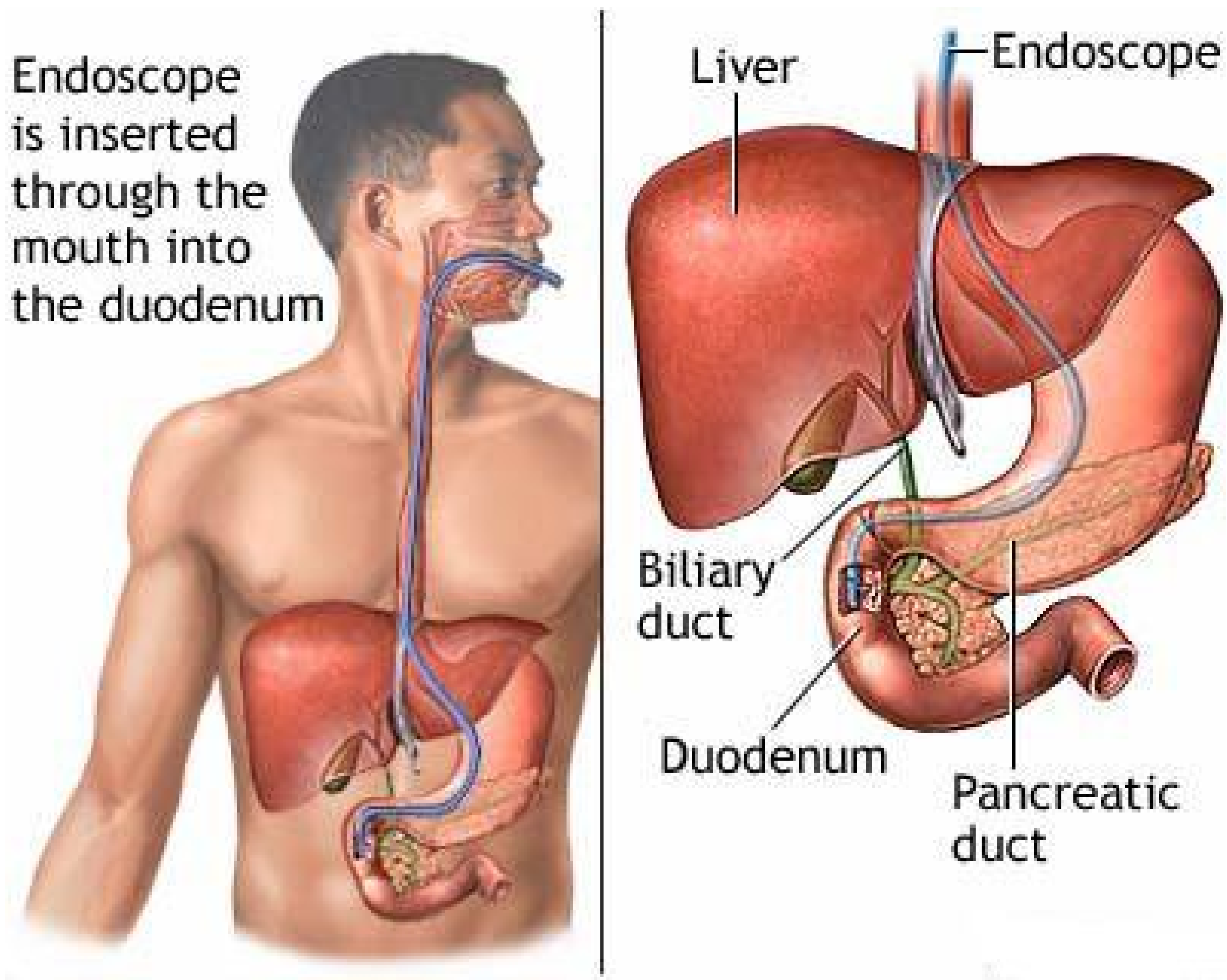
Context

Sensor → Point set (no structure or unknown)

Scanner

Endoscope

• • •



# Reconstruction

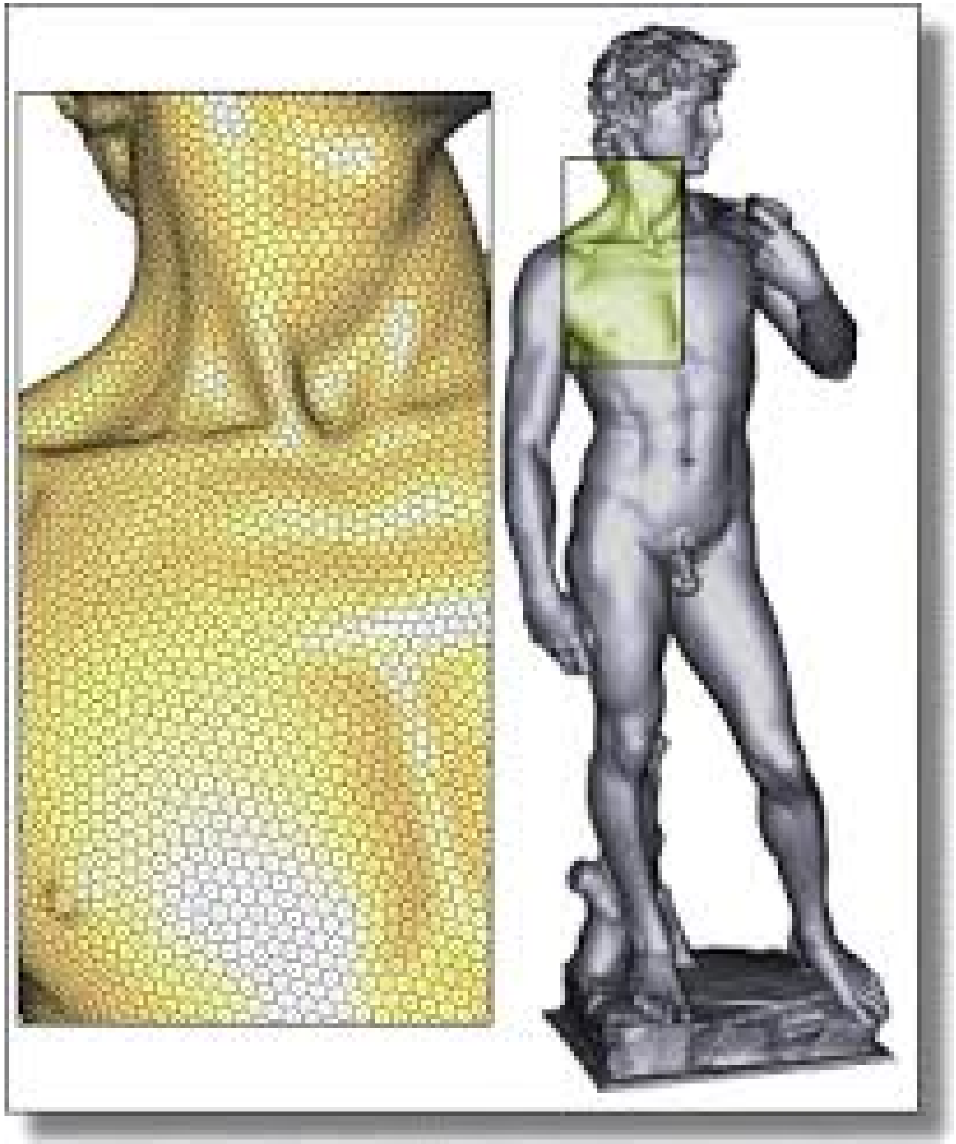
## Context

Cultural heritage

# Reconstruction

Context

Cultural heritage



# Reconstruction

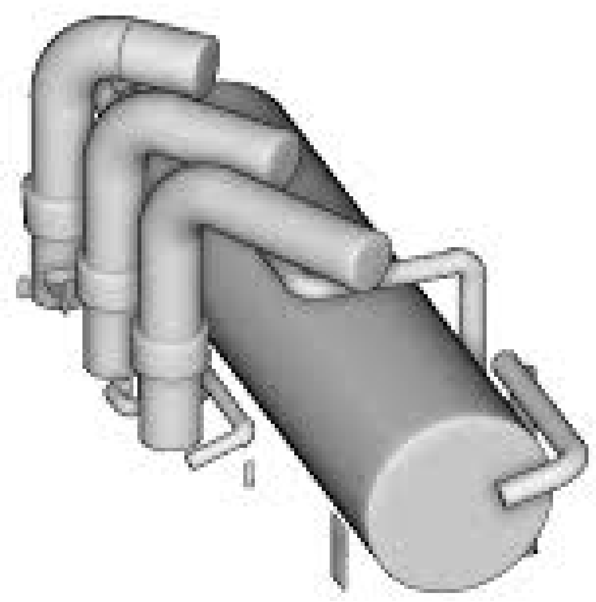
## Context

# Reconstruction

## Context

Reverse engineering



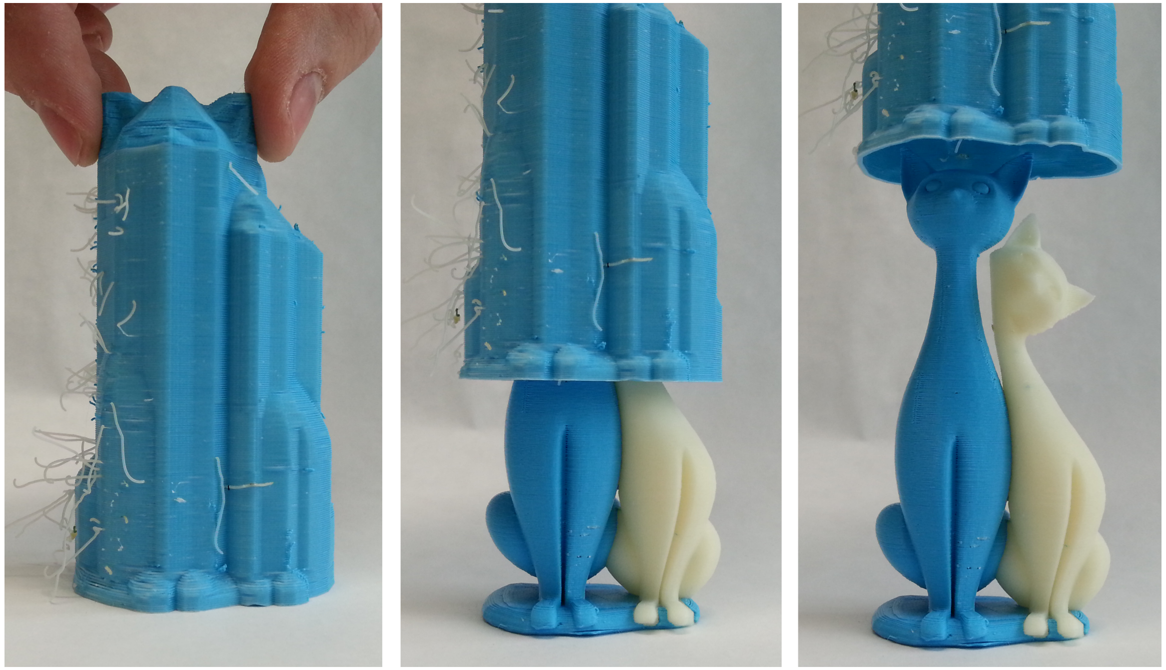
# Reconstruction

## Context

Reverse engineering

Prototyping (3D print)

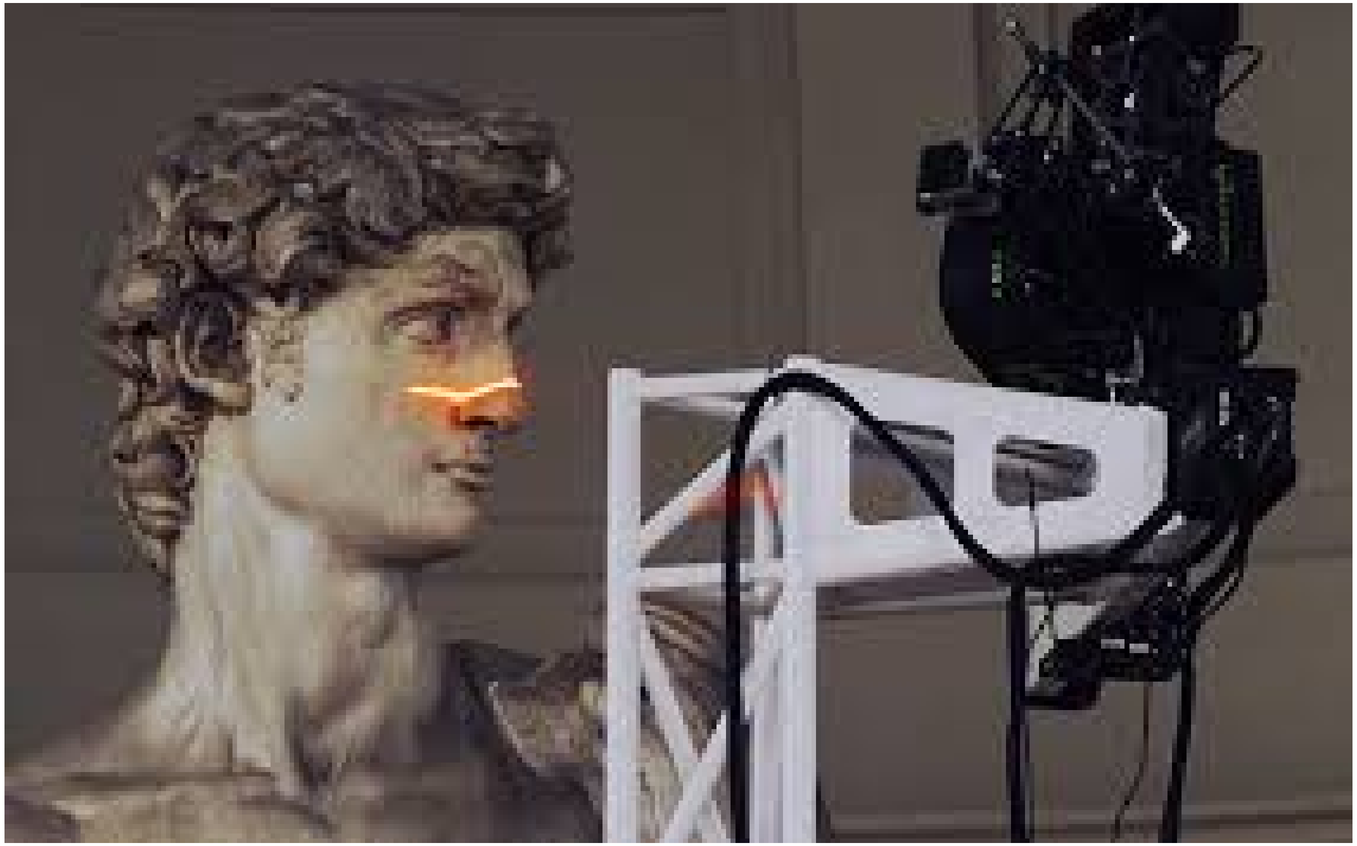
Quality control



# Reconstruction

Context

Sensor → Point set (no structure or unknown)

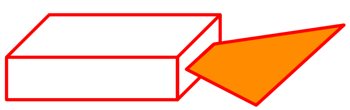


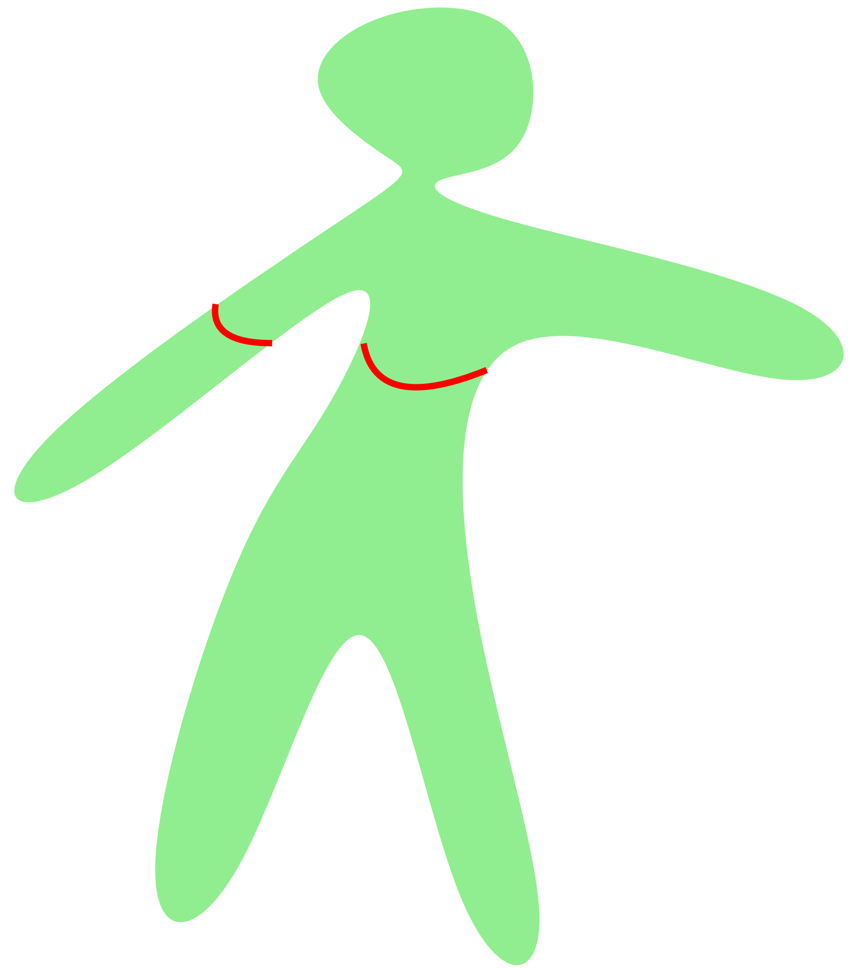
# Reconstruction

Context

Sensor $\longrightarrow$ Point set (no structure or unknown)

Laser illuminate in a plane





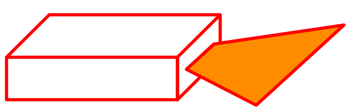
# Reconstruction

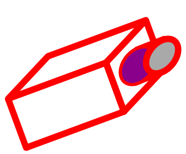
Context

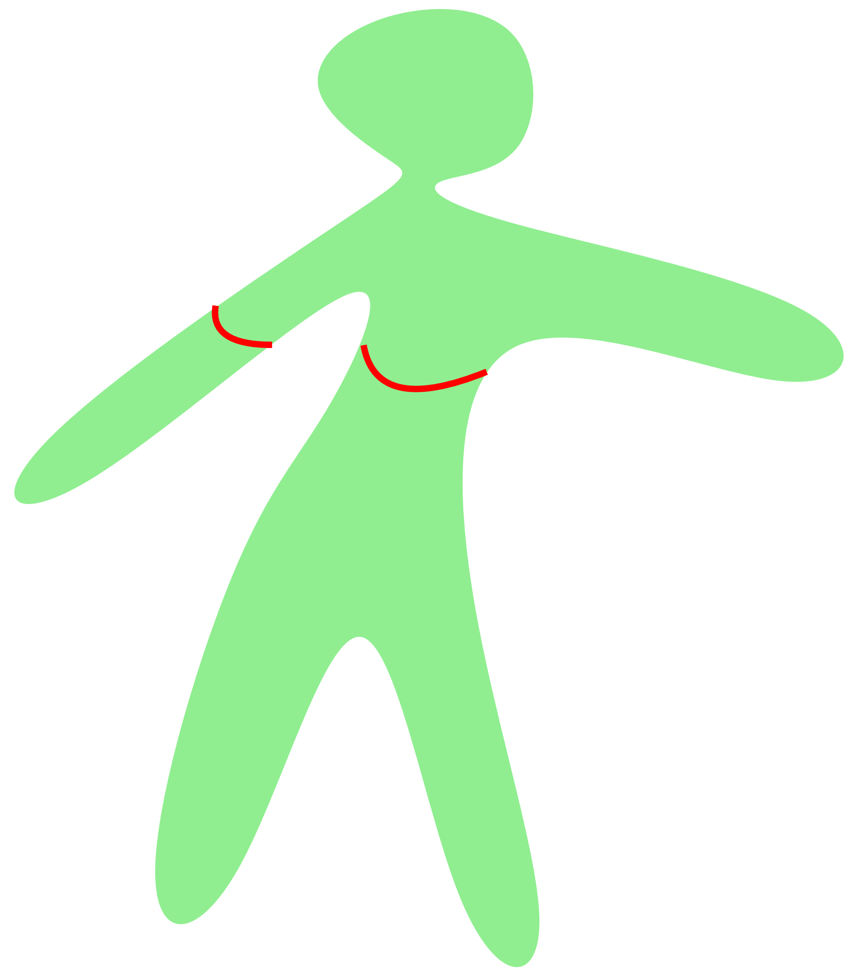
Sensor → Point set (no structure or unknown)

Laser illuminate in a plane

Camera





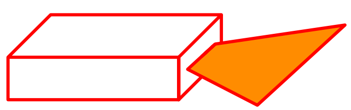


# Reconstruction

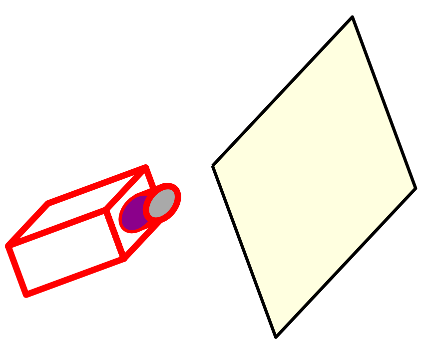
Context

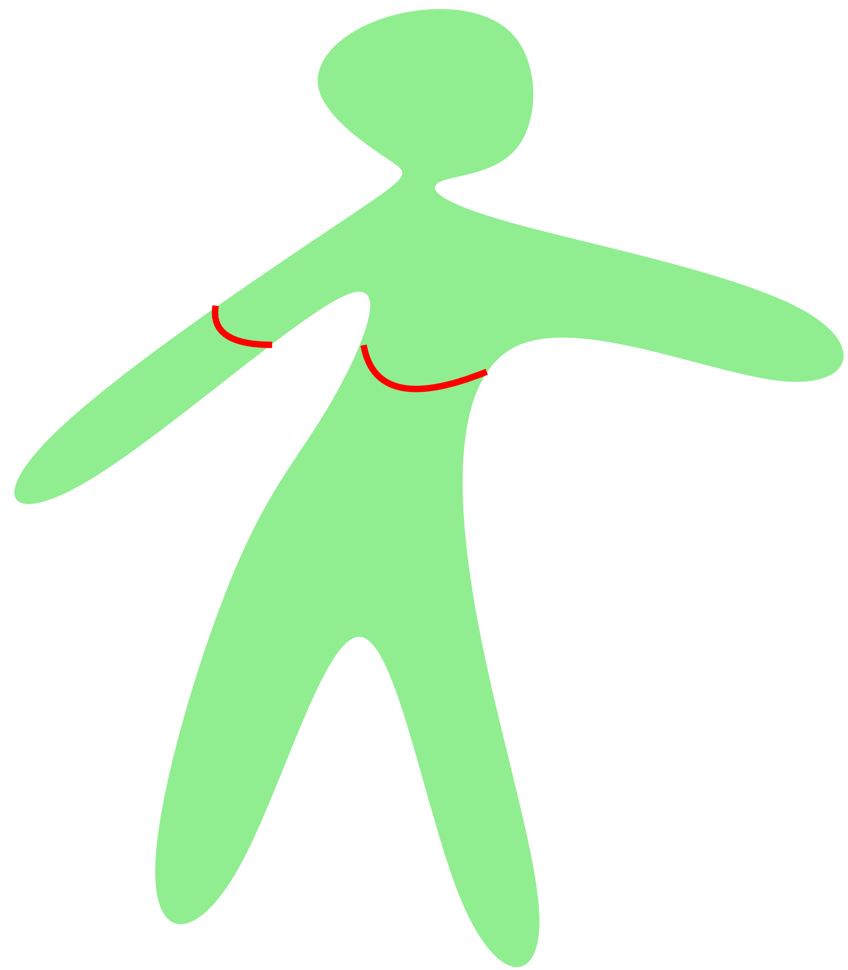
Sensor → Point set (no structure or unknown)

Laser illuminate in a plane



Camera



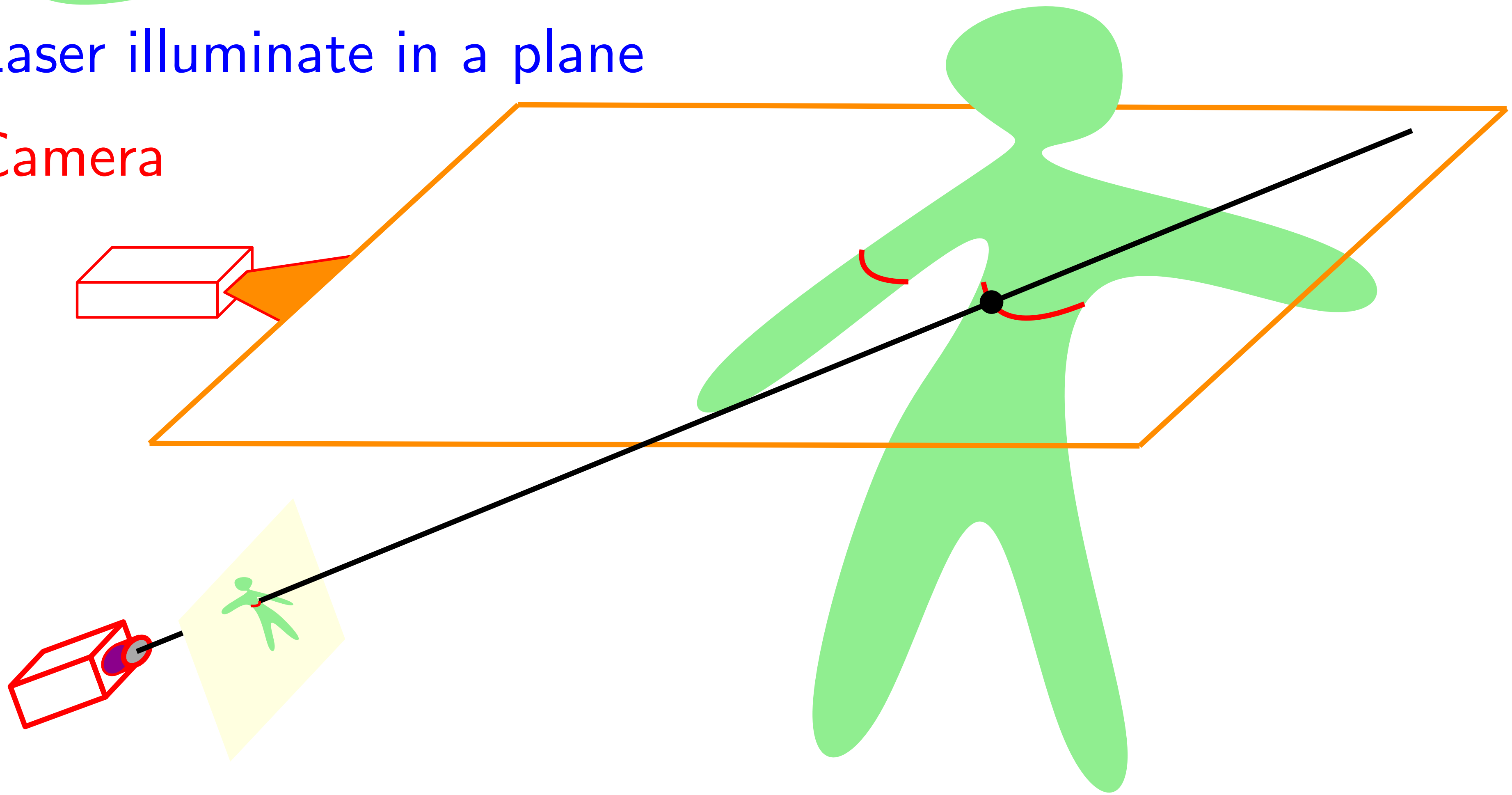
# Reconstruction

## Context

Sensor → Point set (no structure or unknown)

Laser illuminate in a plane

Camera

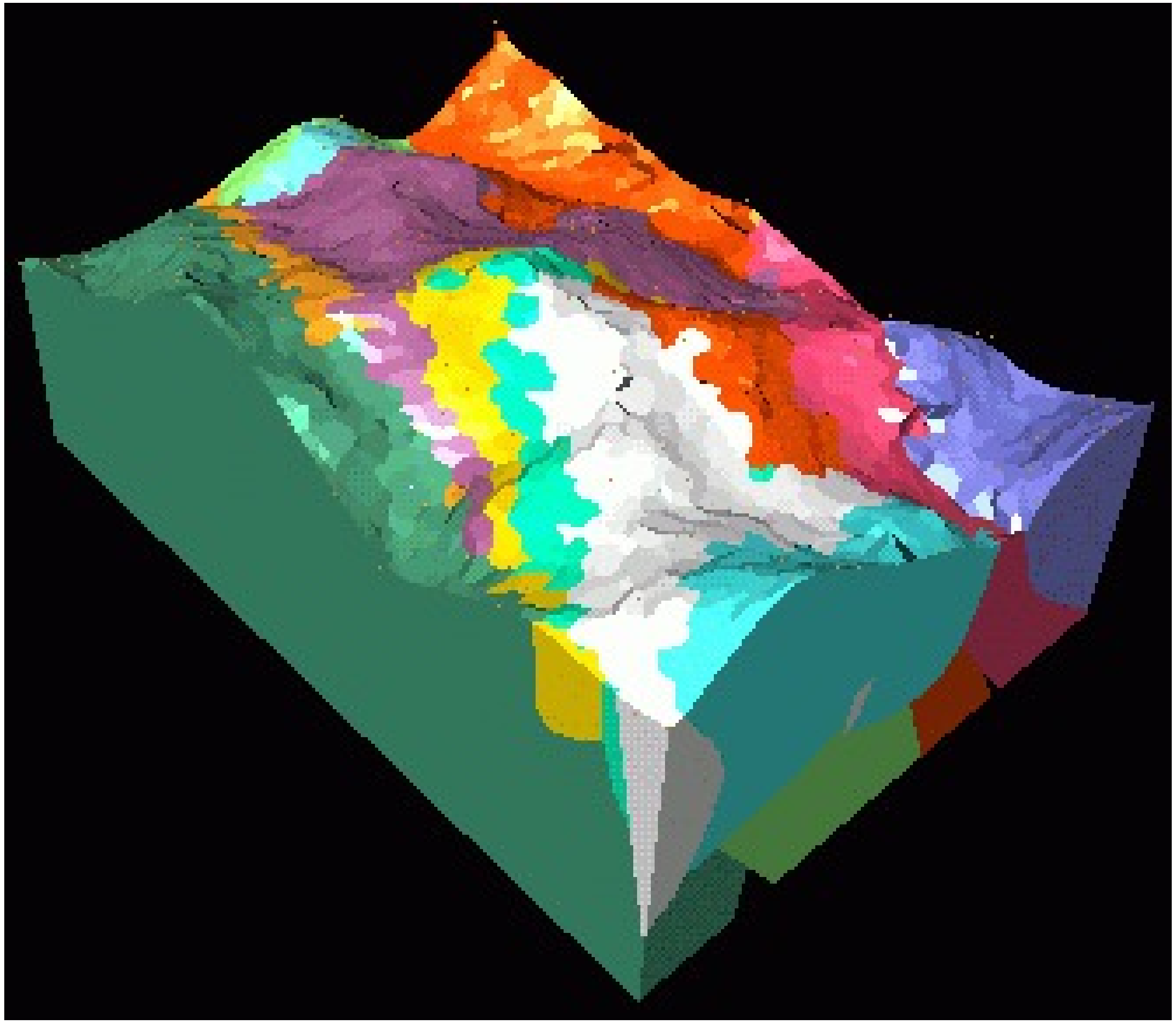


Get 3D position

# Reconstruction

Context

Geology

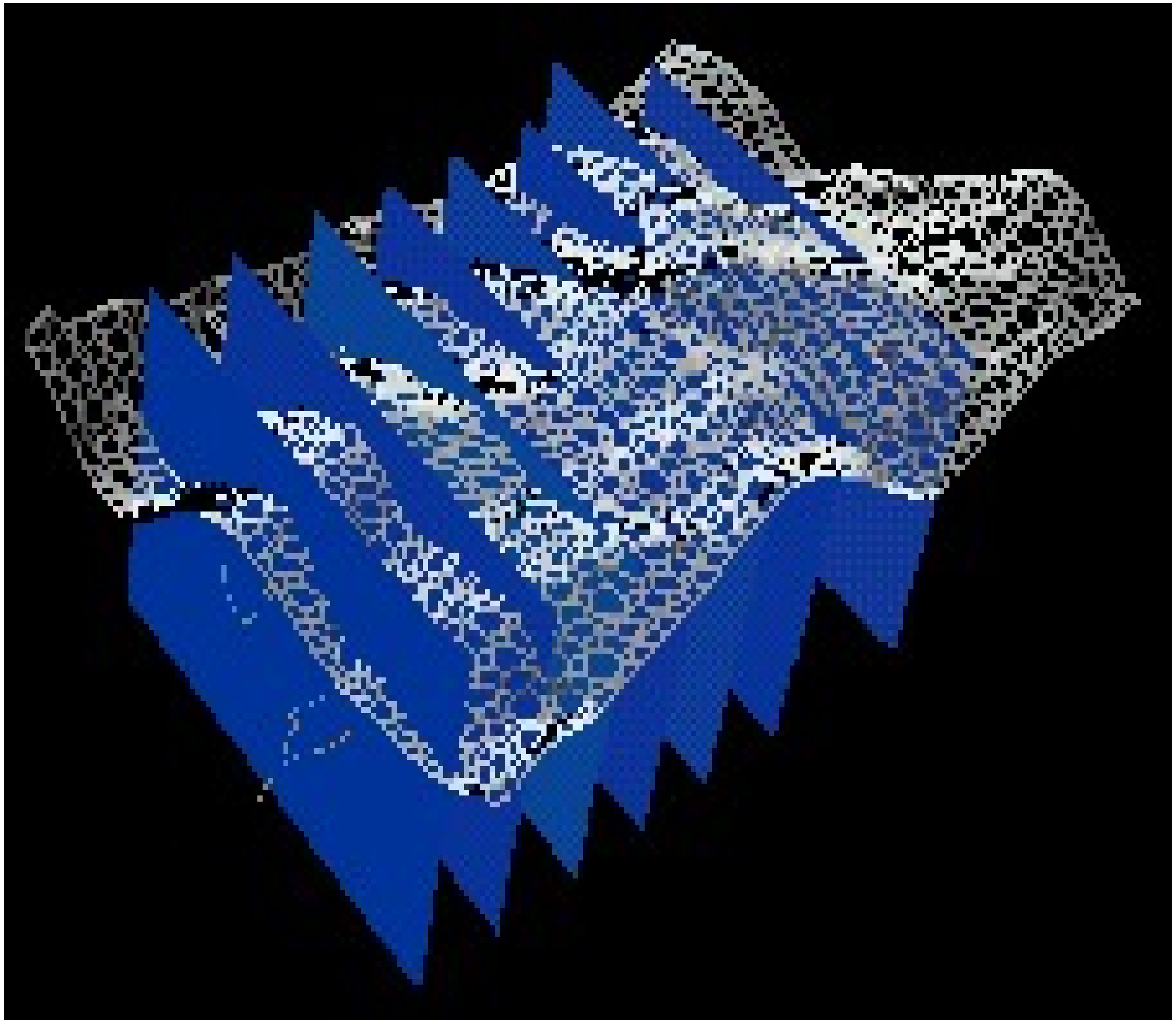


# Reconstruction

Context

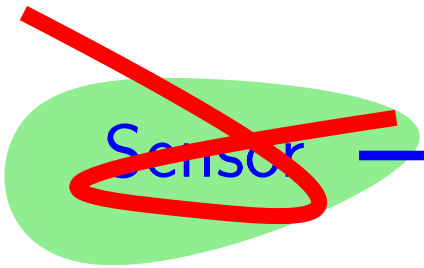
Sensor → Point set (no structure or unknown)

Geology



# Reconstruction

Context

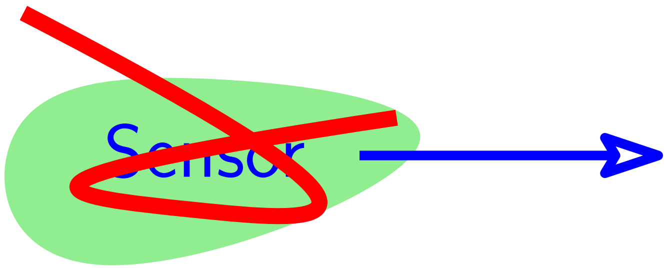



→ Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

# Reconstruction

Context

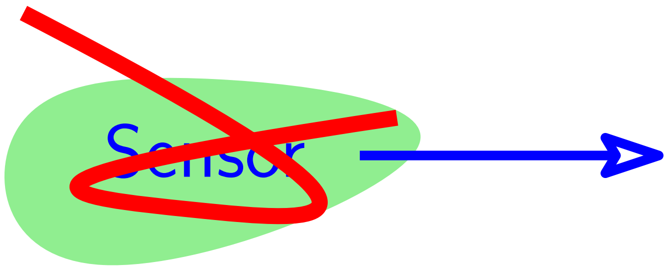



Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

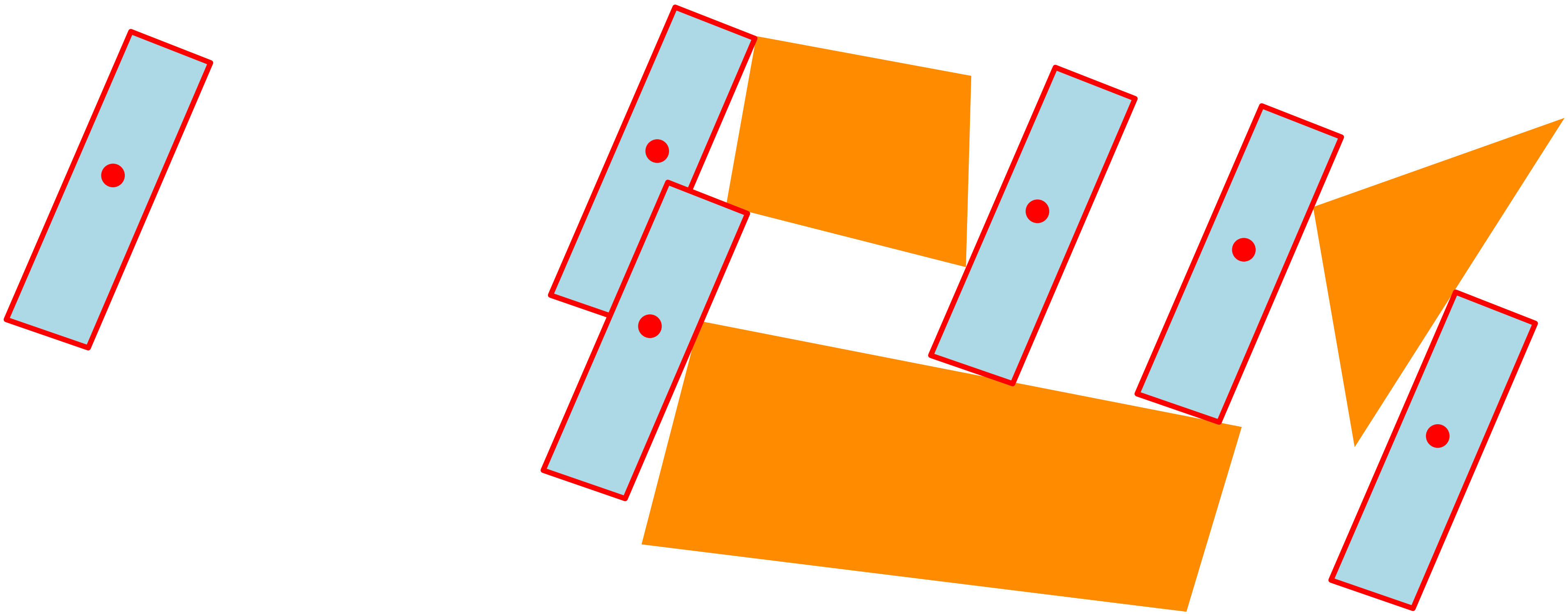
# Reconstruction

Context




Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

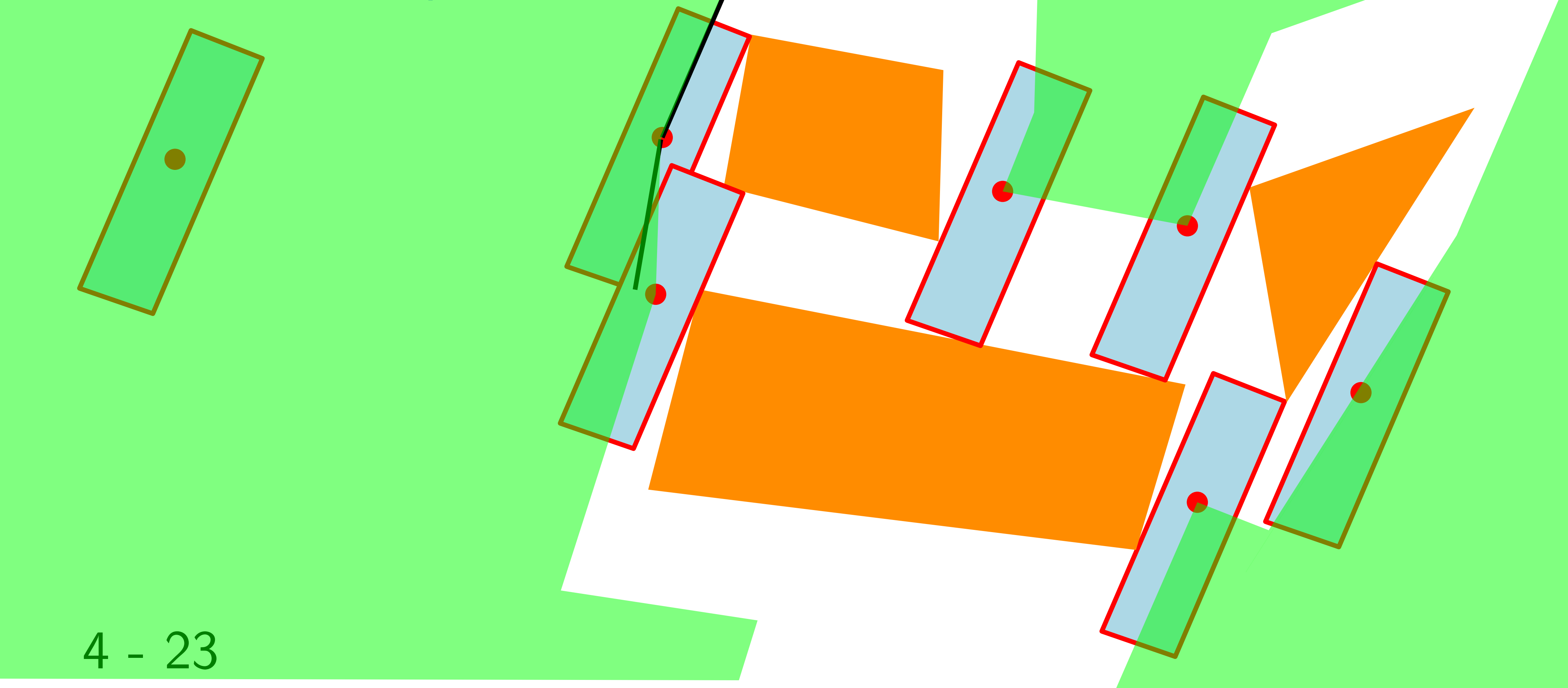


# Reconstruction

Context

~~Sensor~~ → Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

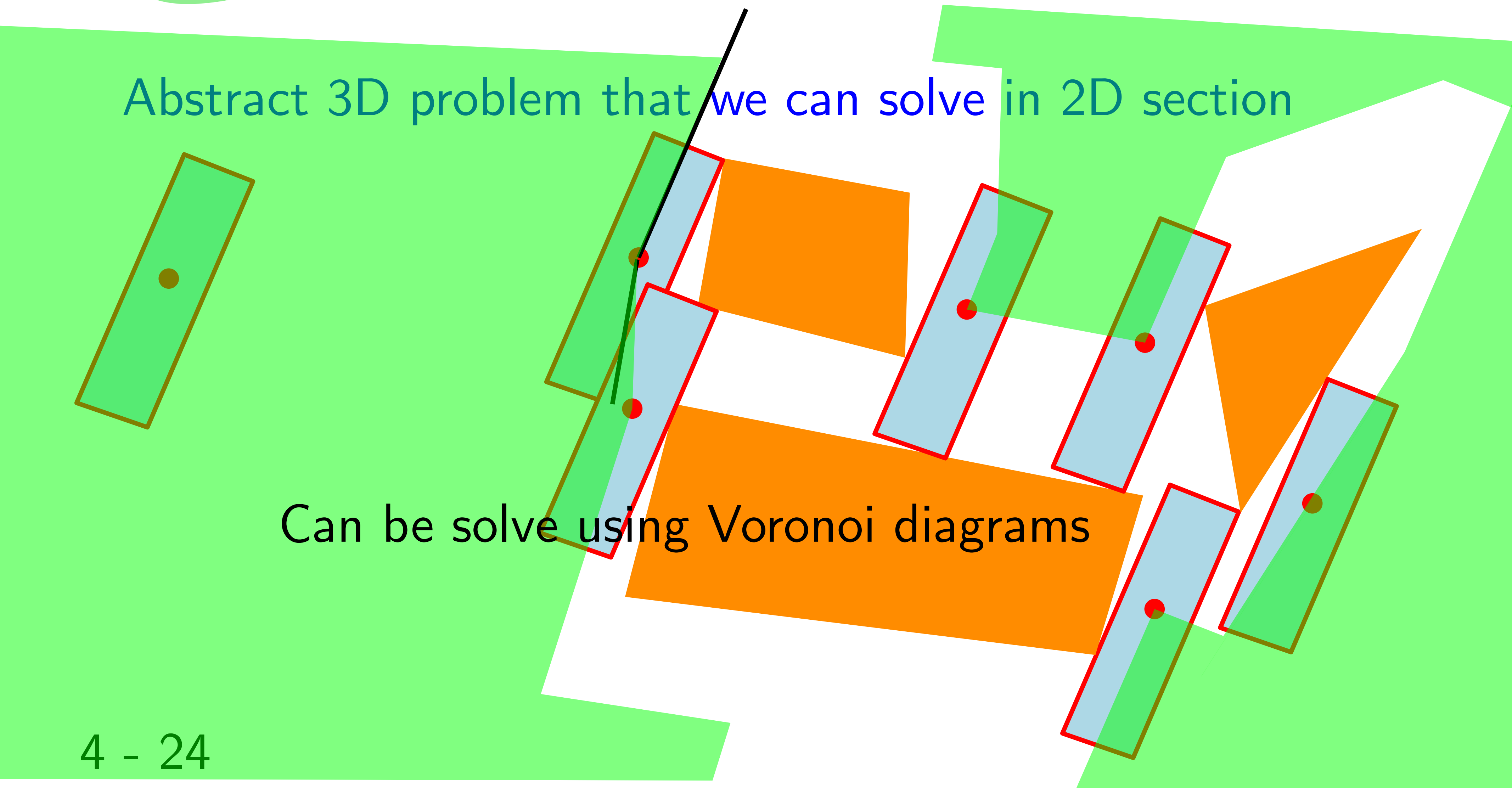


# Reconstruction

Context

~~Sensor~~ → Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section



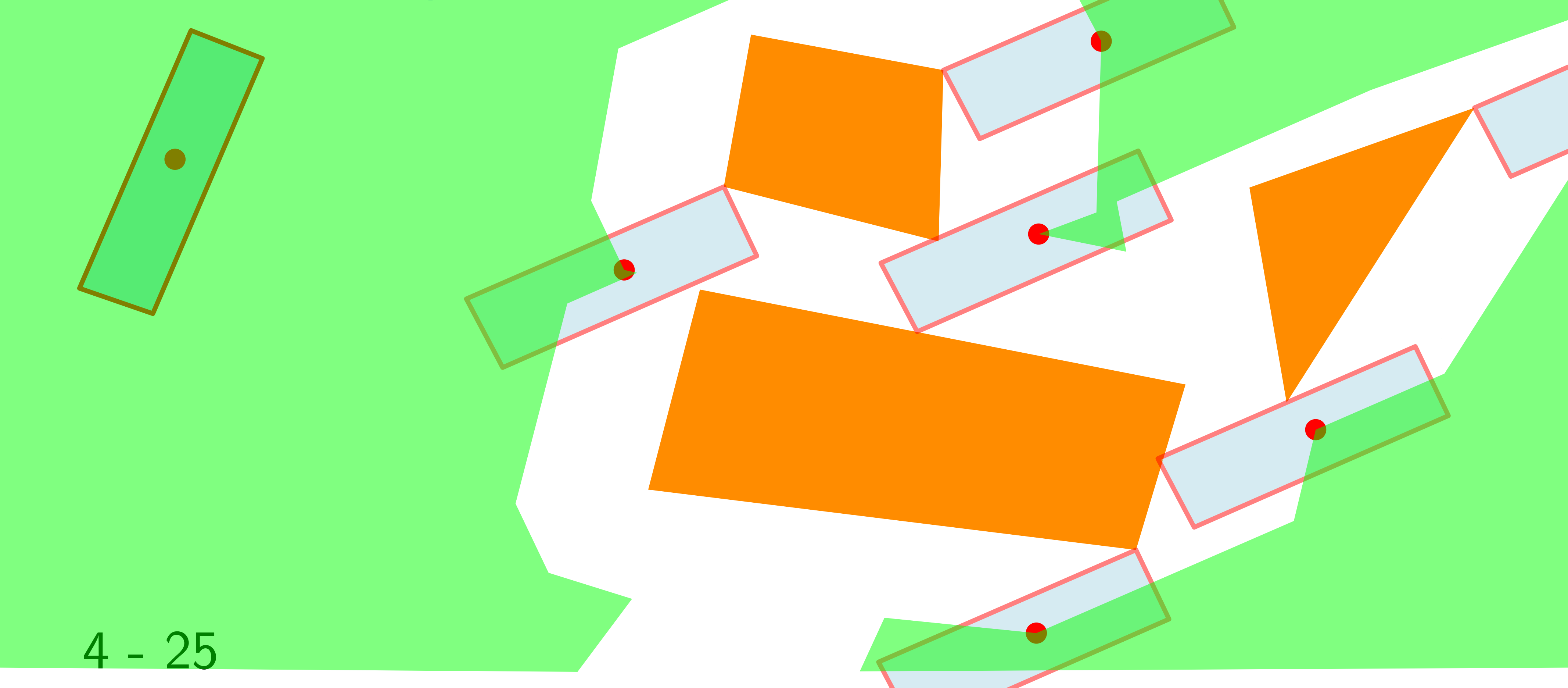
Can be solve using Voronoi diagrams

# Reconstruction

## Context

~~Sensor~~ → Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

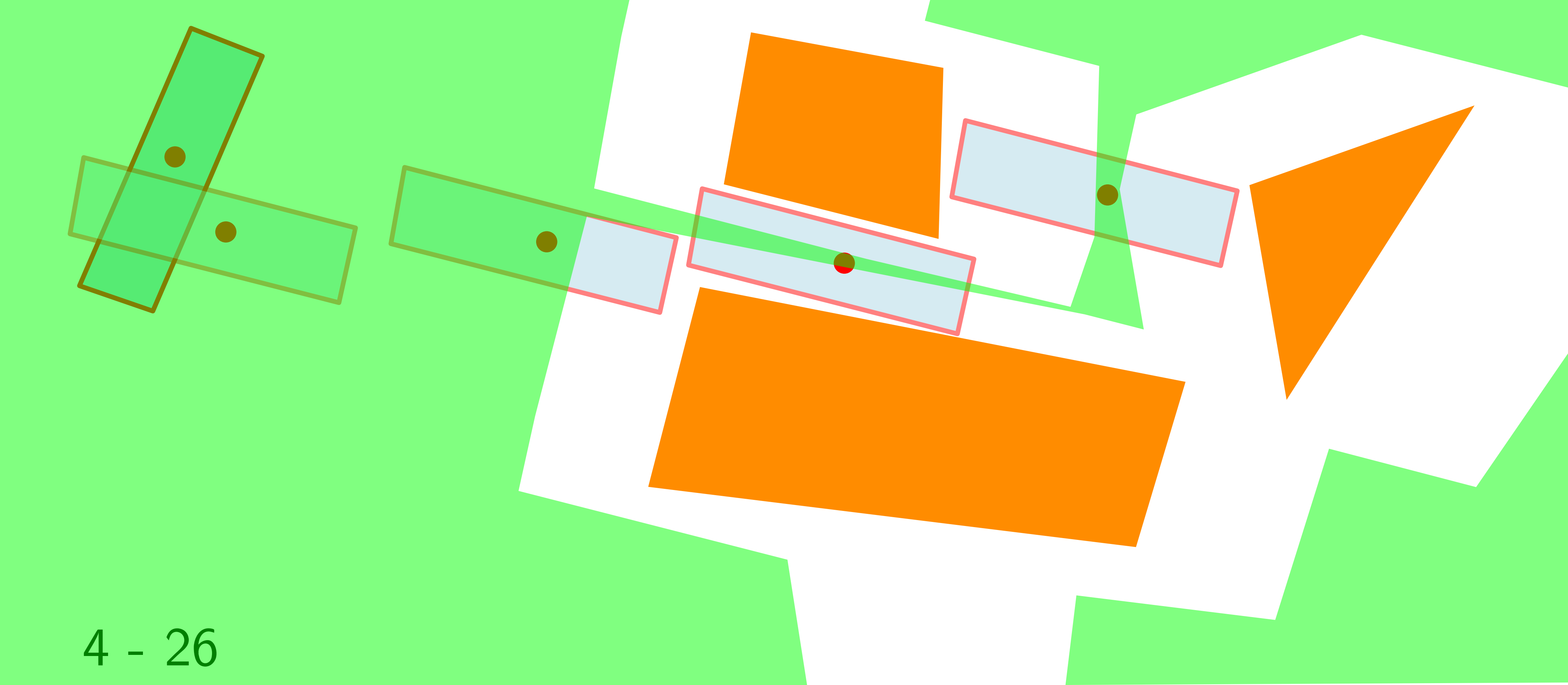


# Reconstruction

Context

~~Sensor~~ → Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

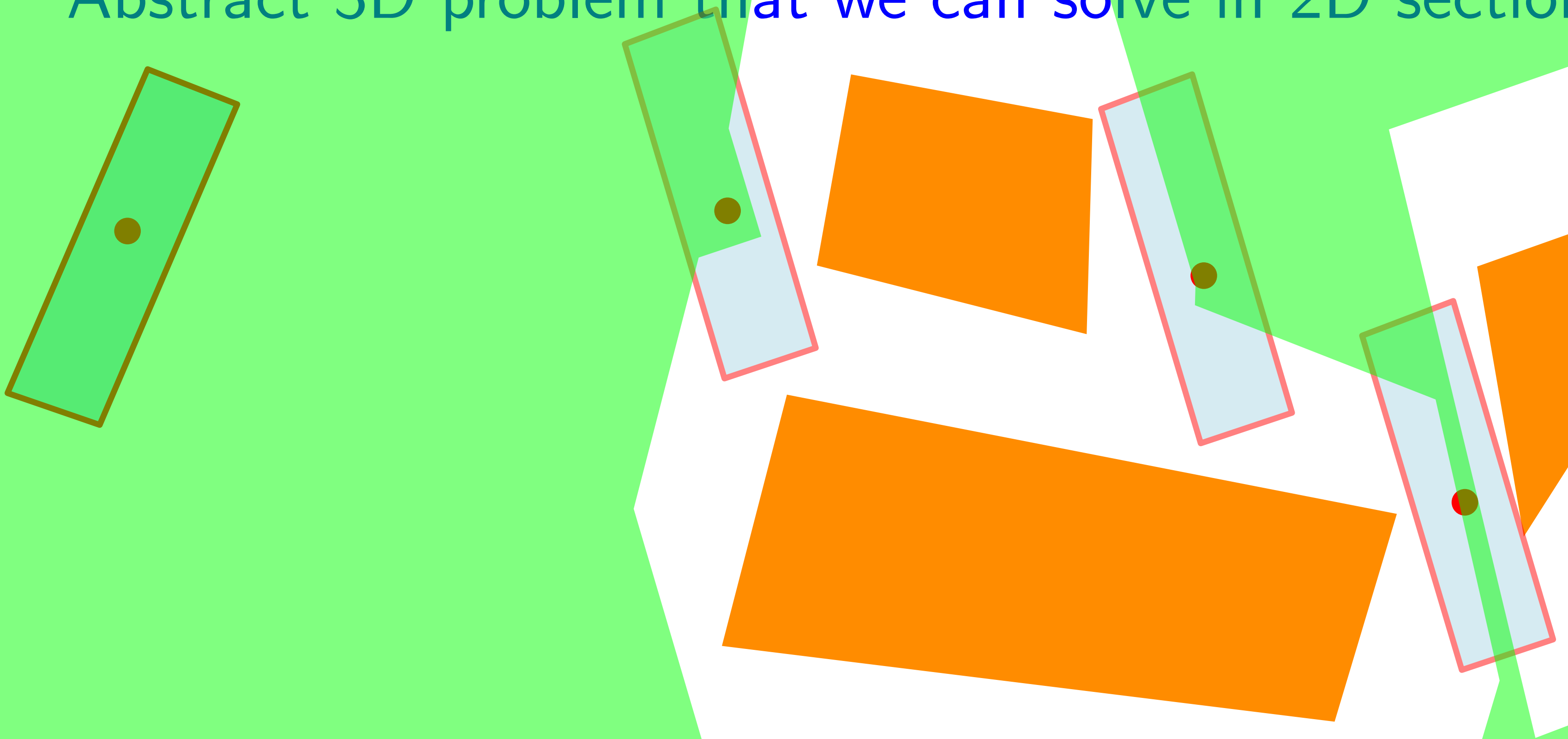


# Reconstruction

Context

~~Sensor~~ → Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

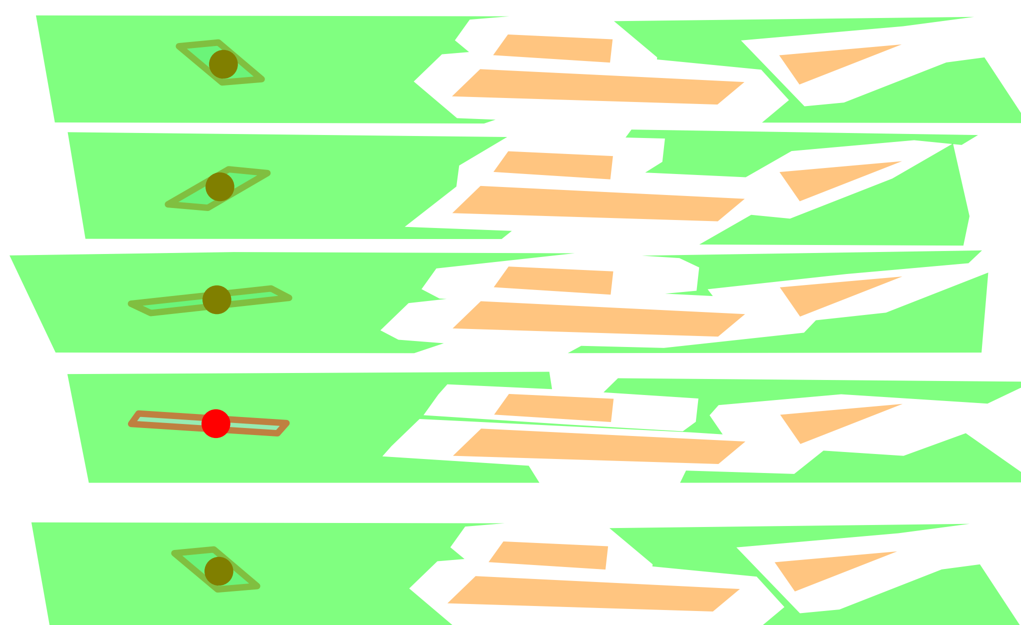


# Reconstruction

Context

~~Sensor~~ → Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

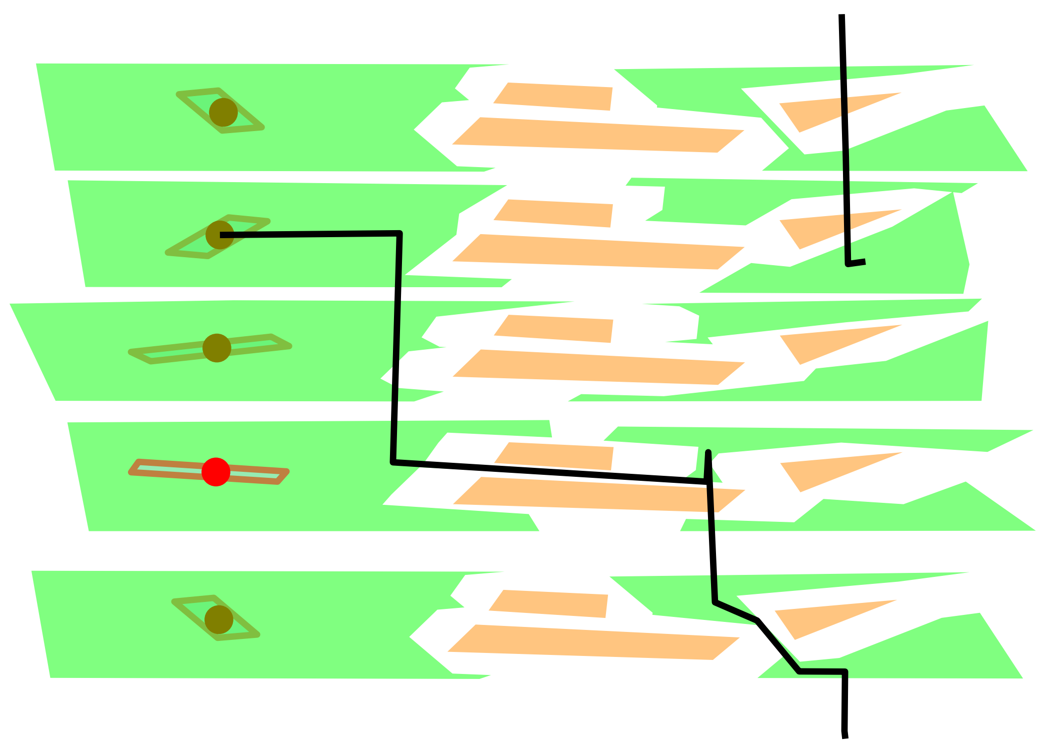


# Reconstruction

Context

~~Sensor~~ → Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section

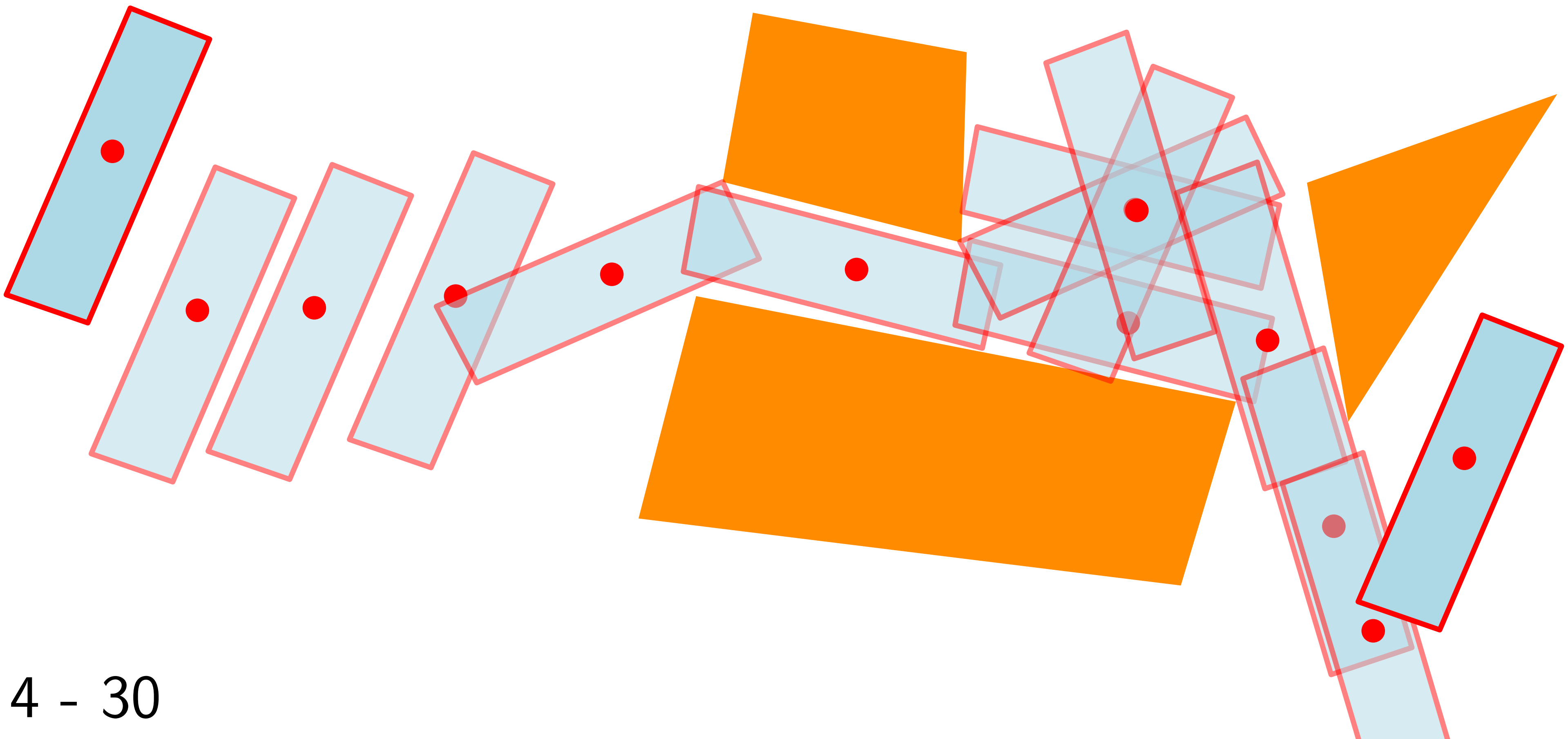


# Reconstruction

Context

~~Sensor~~ → Point set (no structure or unknown)

Abstract 3D problem that we can solve in 2D section



# Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

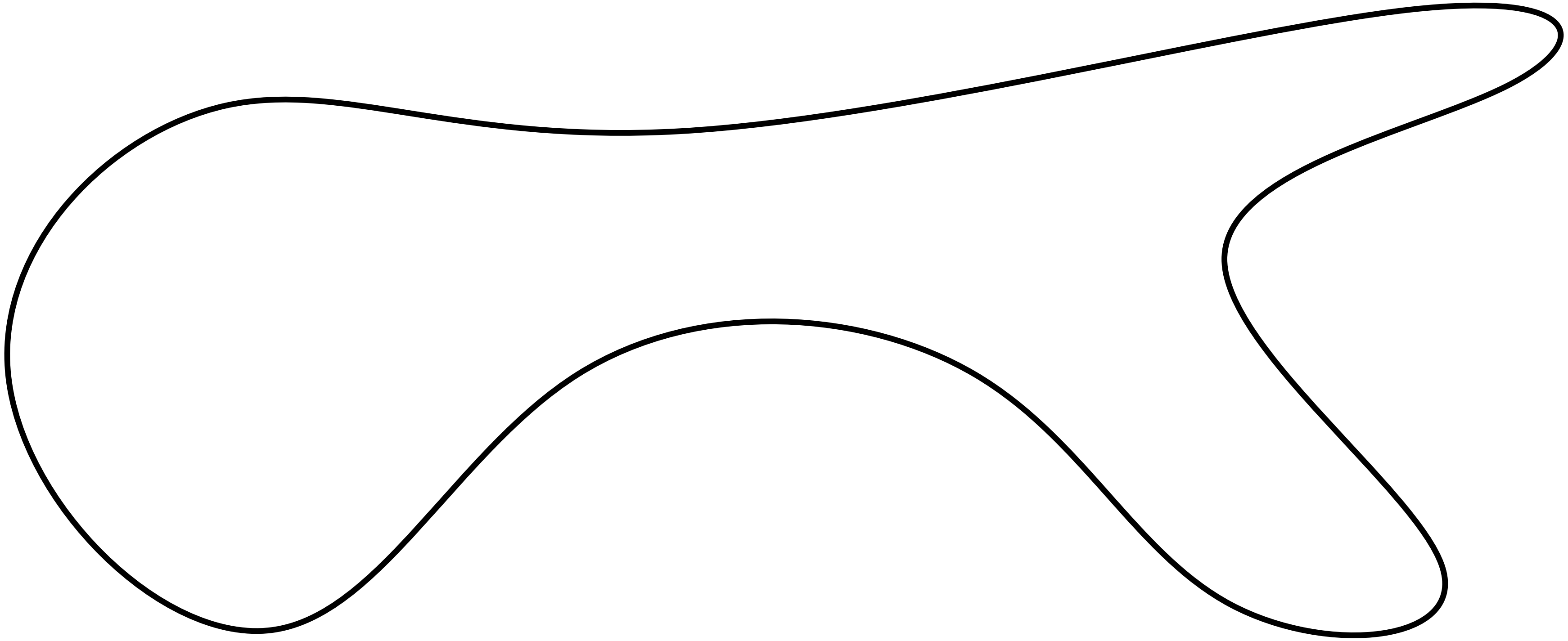
Locus of center of bitangent spheres

# Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

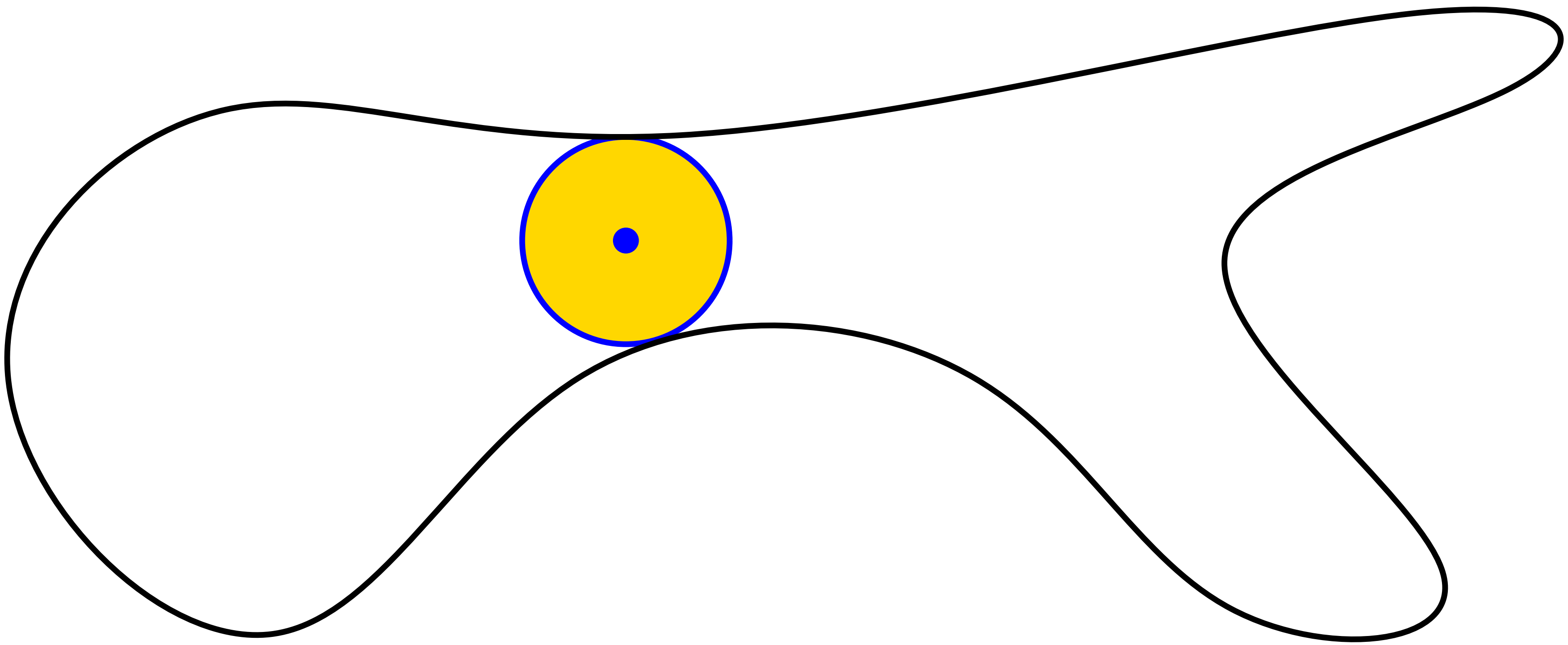


# Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

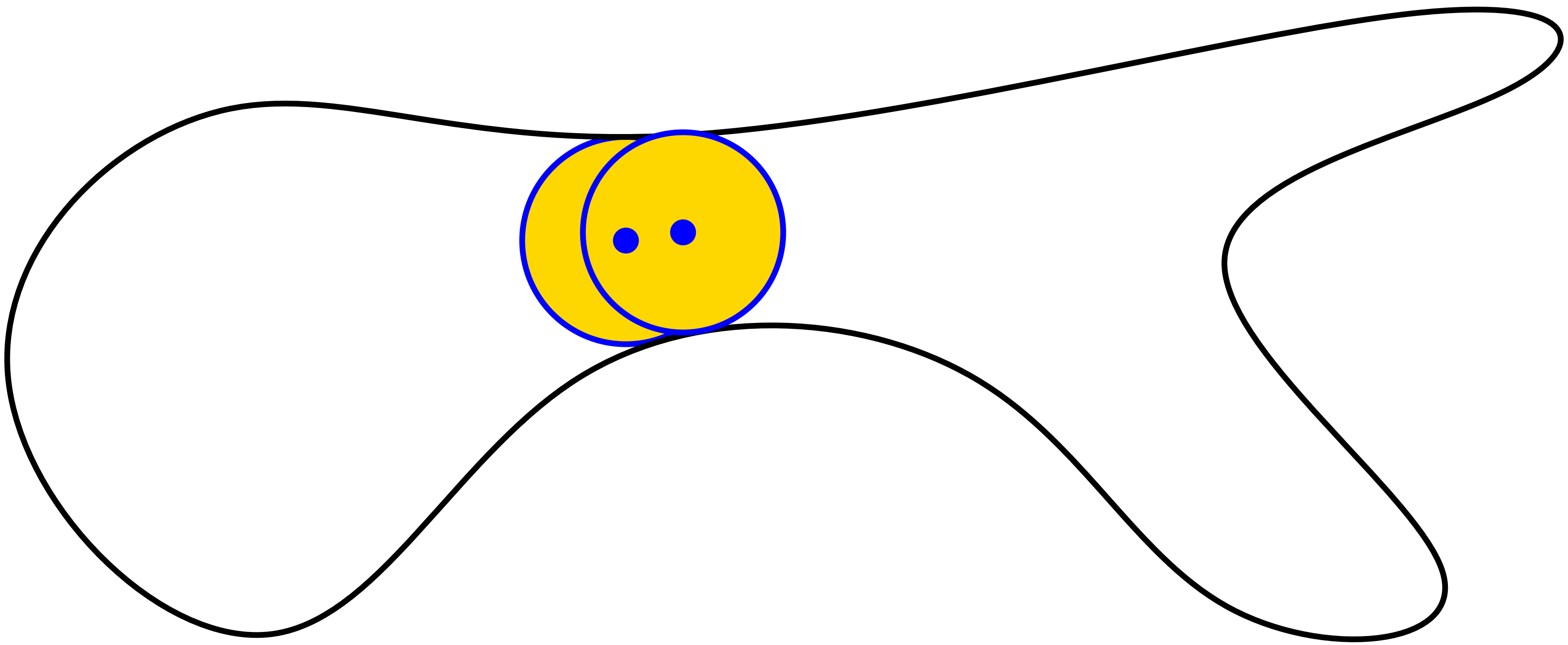


# Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

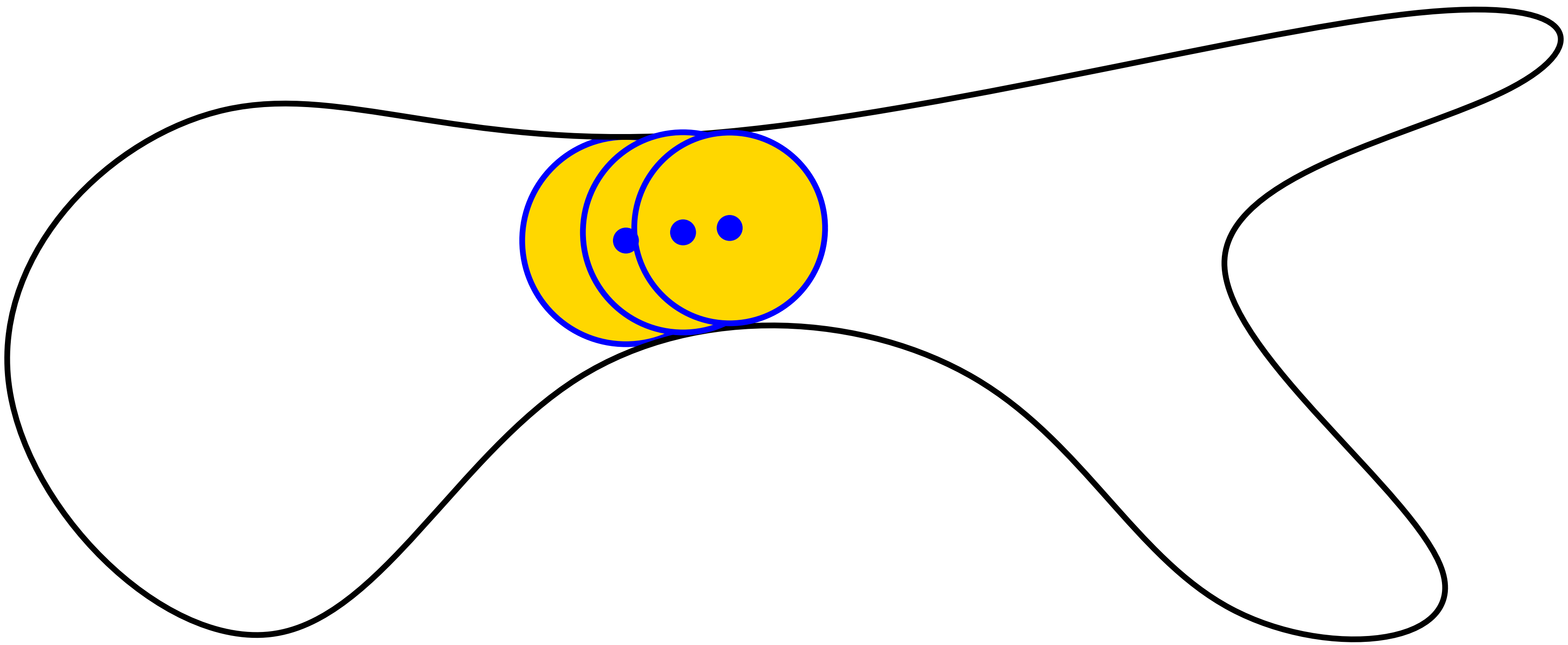


# Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

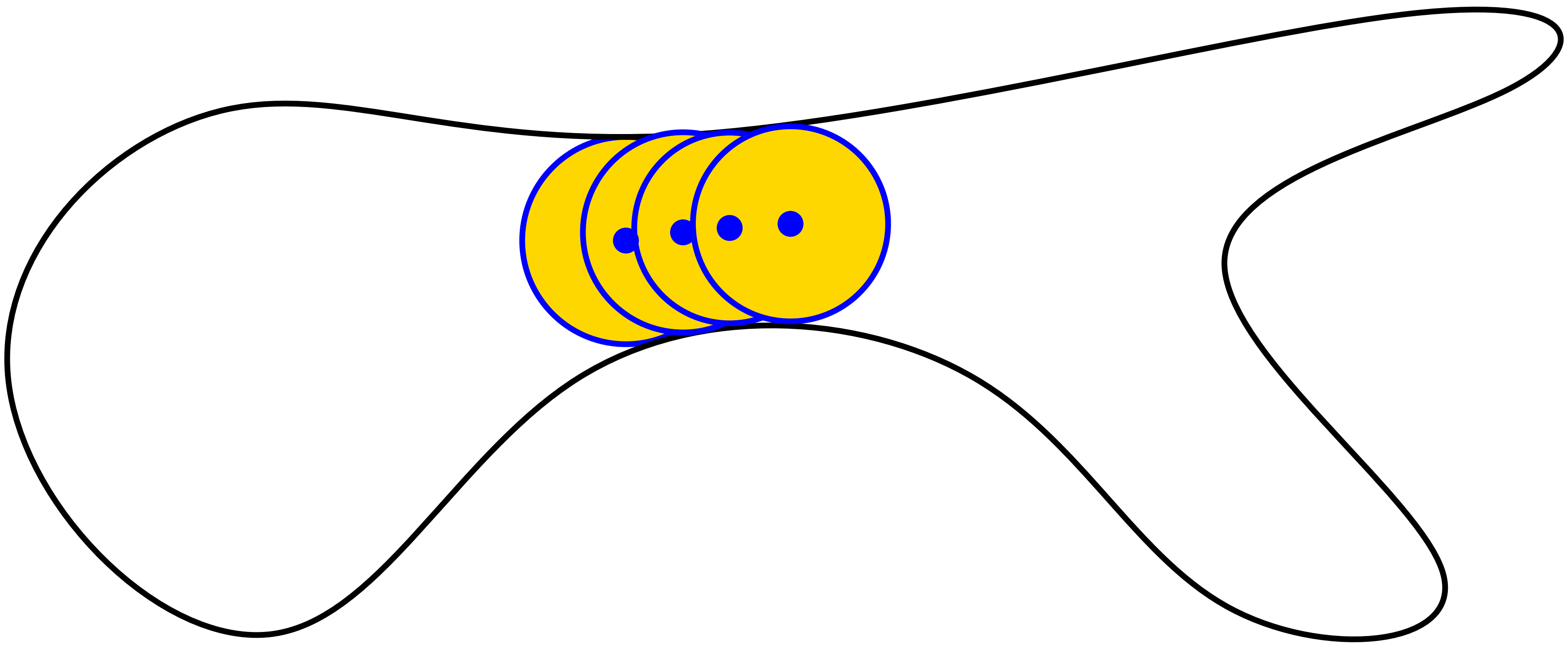


# Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

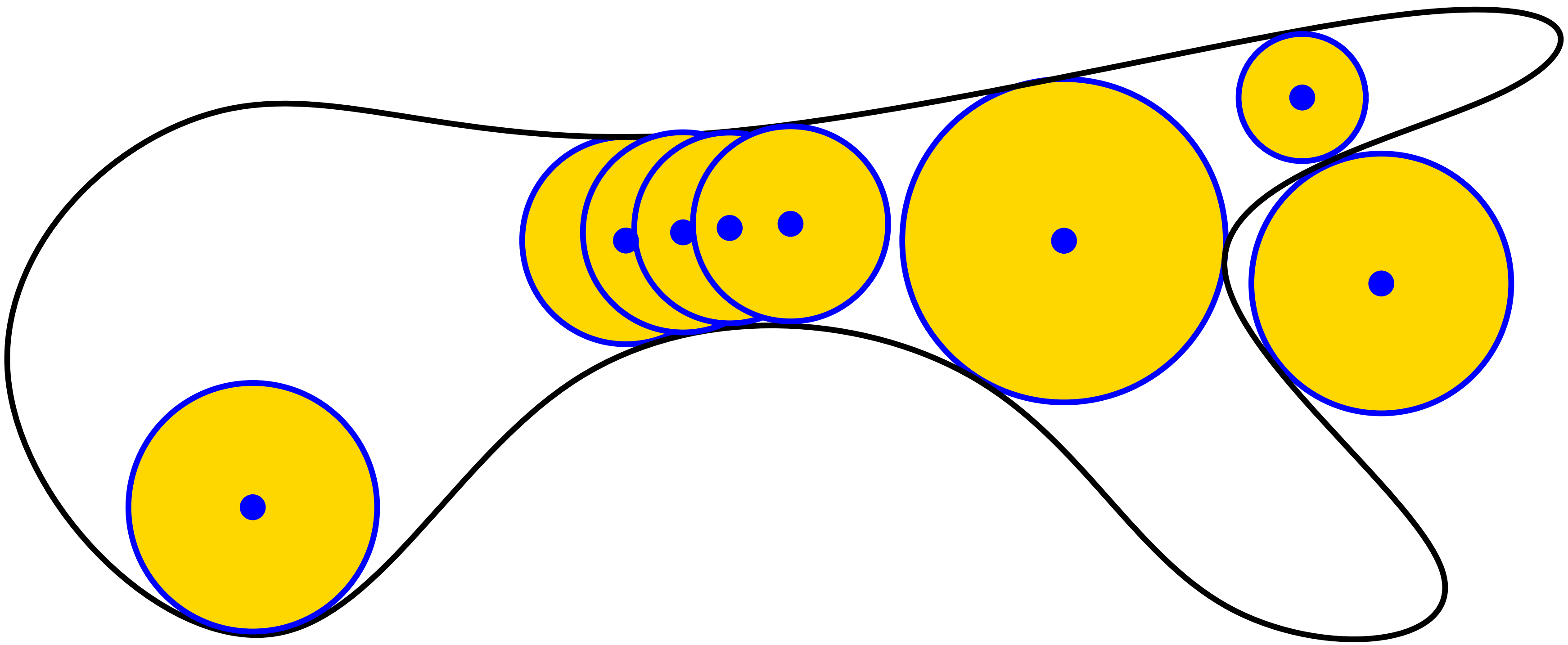


# Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

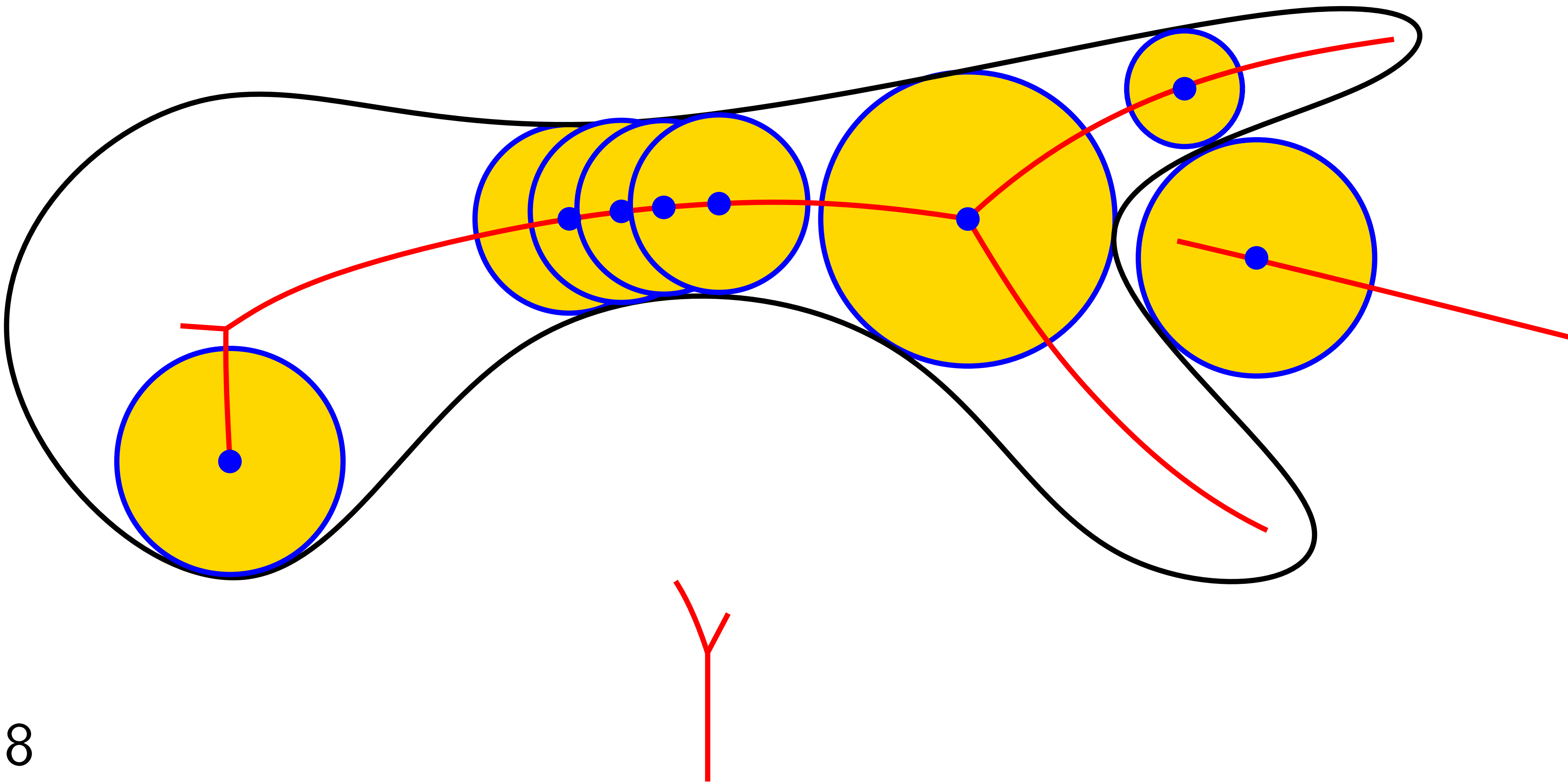


# Reconstruction

Delaunay is a good start

Medial axis of a curve (surface in 3D)

Locus of center of bitangent spheres

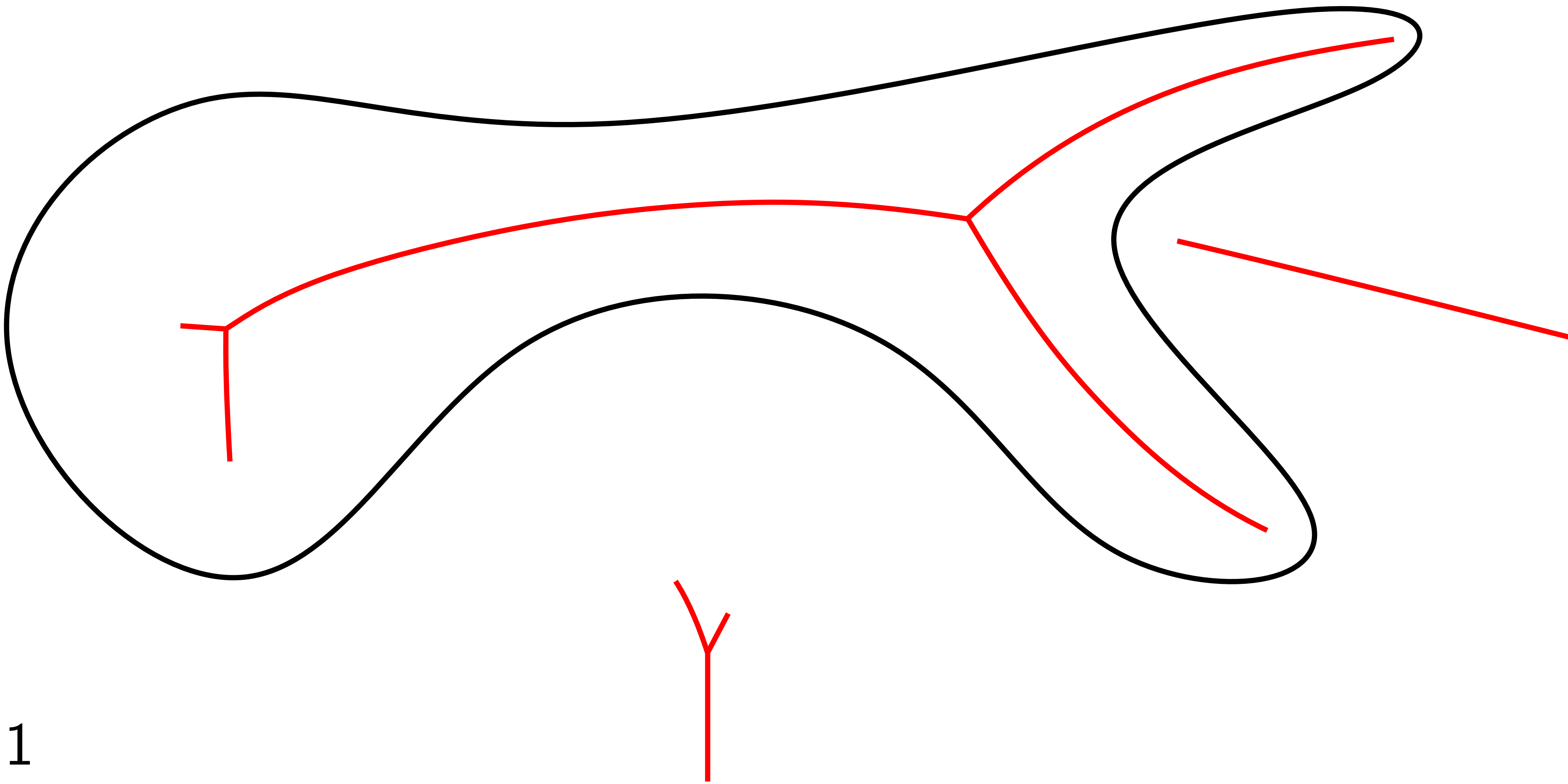


# Reconstruction

## Delaunay is a good start

$\epsilon$-sample of a curve

Local feature size:

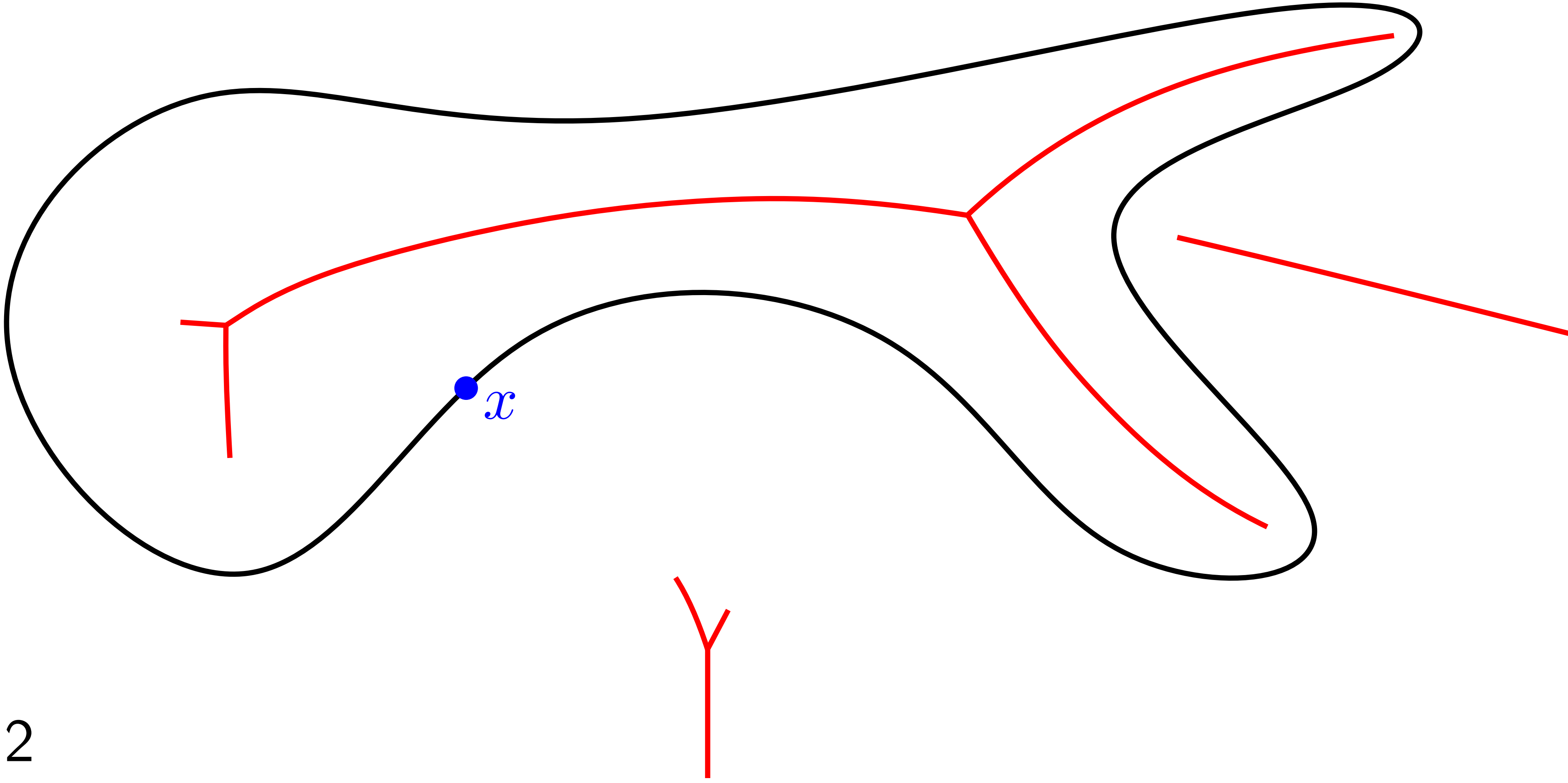


# Reconstruction

Delaunay is a good start

$\epsilon$-sample of a curve

Local feature size: $\mathsf{lfs}(x) =$

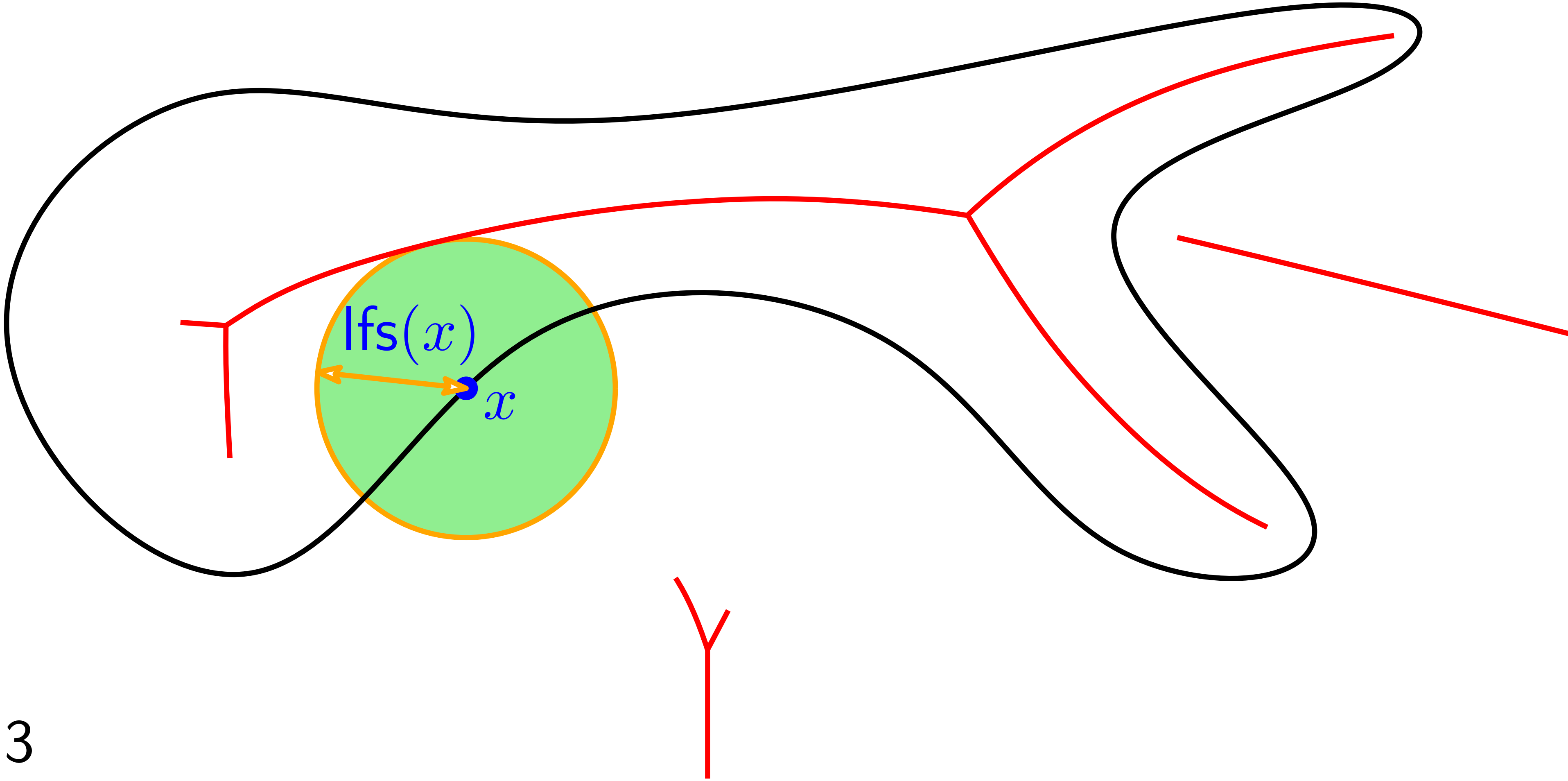


# Reconstruction

Delaunay is a good start

$\epsilon$-sample of a curve

Local feature size: $\mathsf{lfs}(x) =$ distance($x$, medial axis)



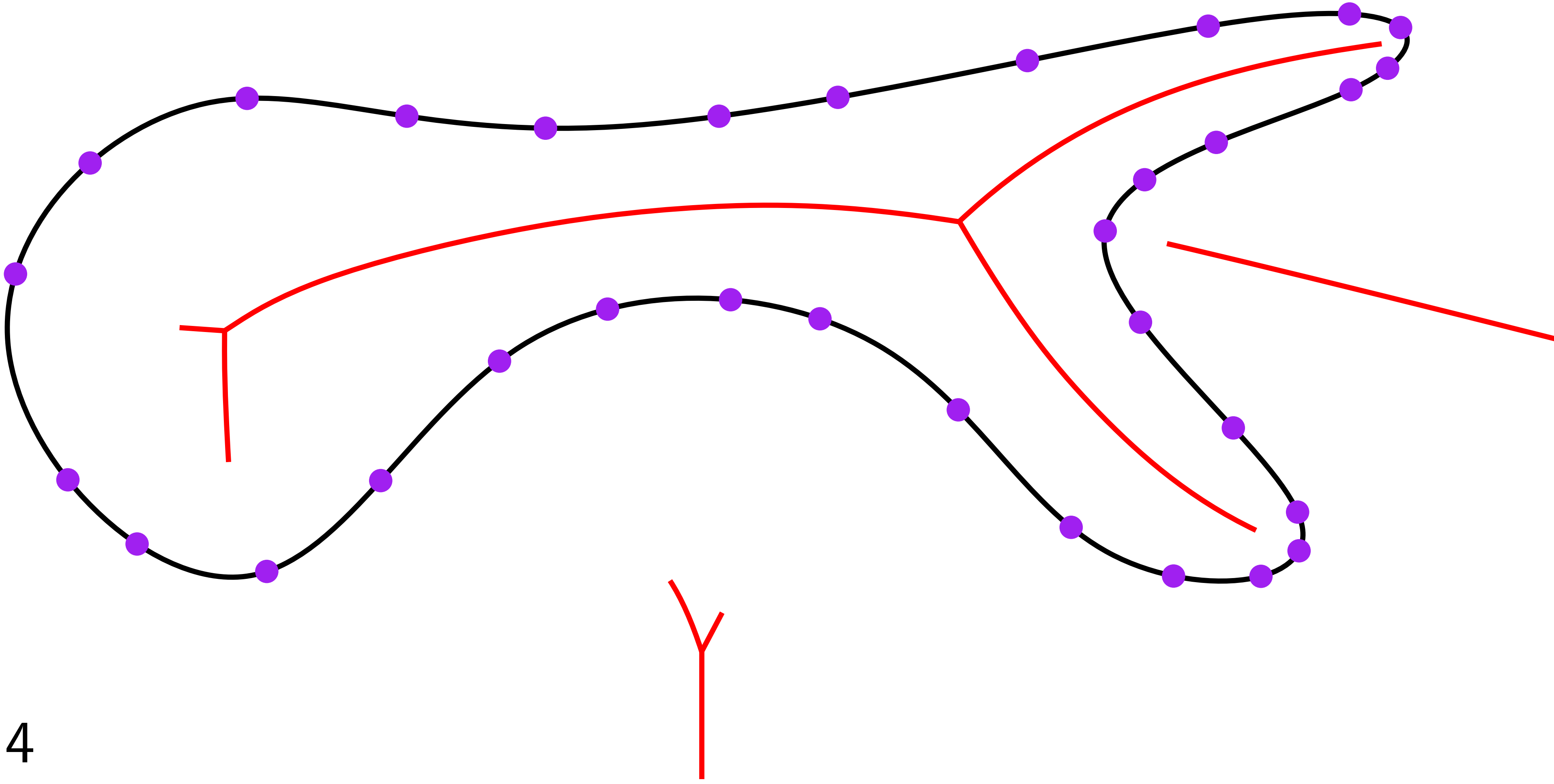
# Reconstruction

Delaunay is a good start

Sample is an

$\epsilon$-sample of a curve

Local feature size:



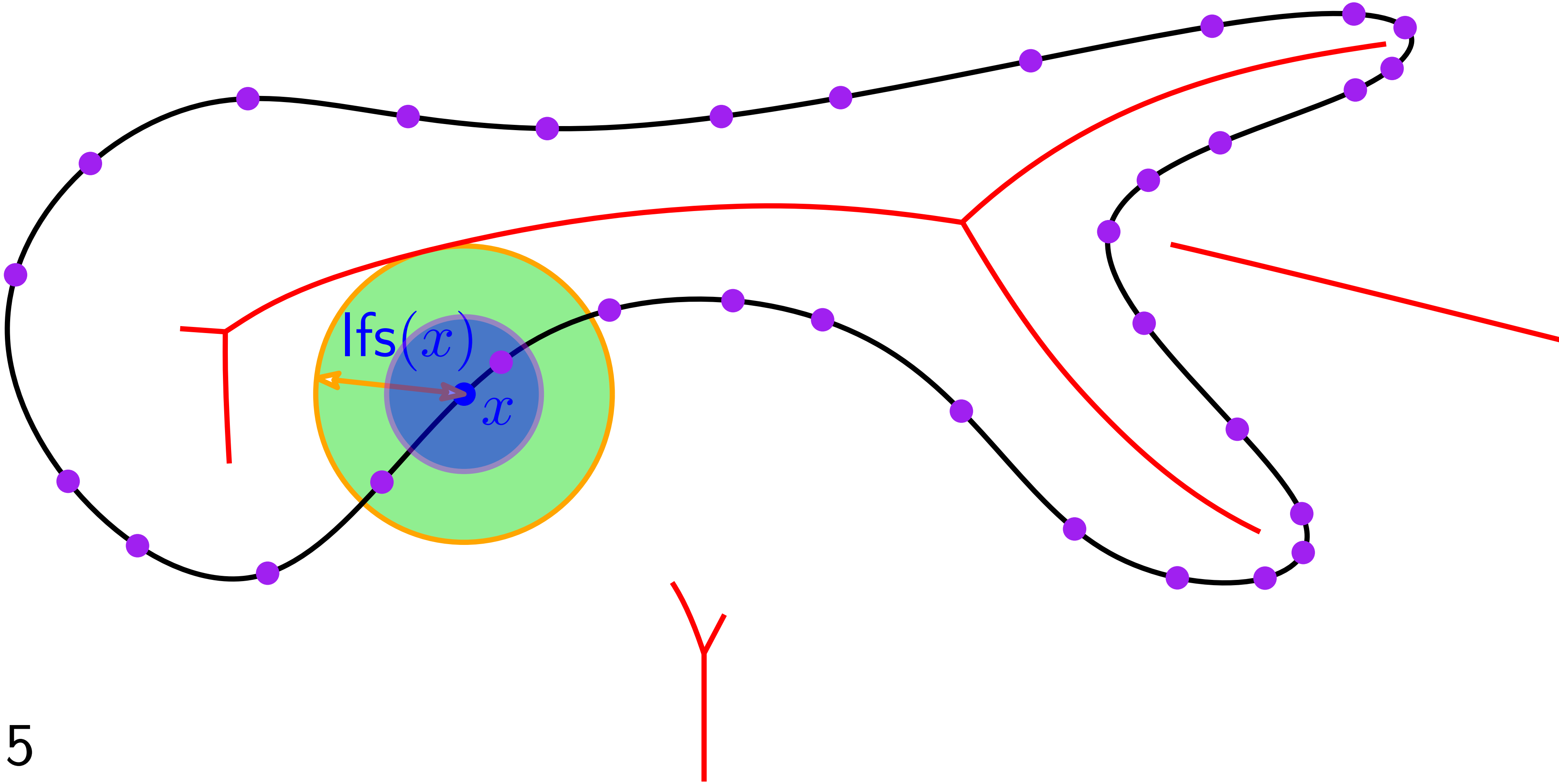
# Reconstruction

Delaunay is a good start

Sample is an

$\epsilon$-sample of a curve if $\forall x$, Disk$(x, \epsilon{\cdot}\mathrm{lfs}(x)) \cap$ Sample $\neq \emptyset$

Local feature size: $\mathrm{lfs}(x) =$ distance$(x$, medial axis$)$



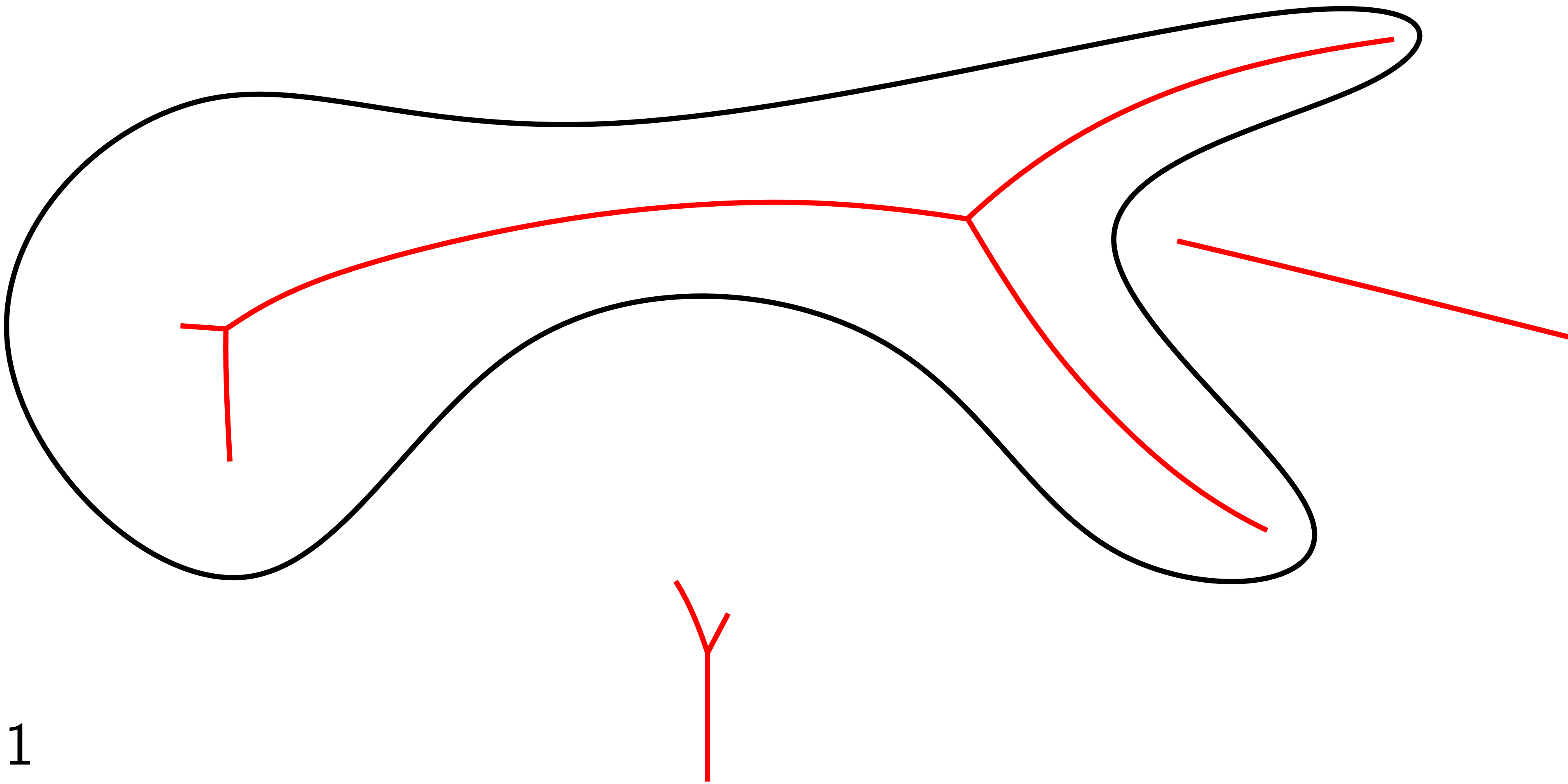
# Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$



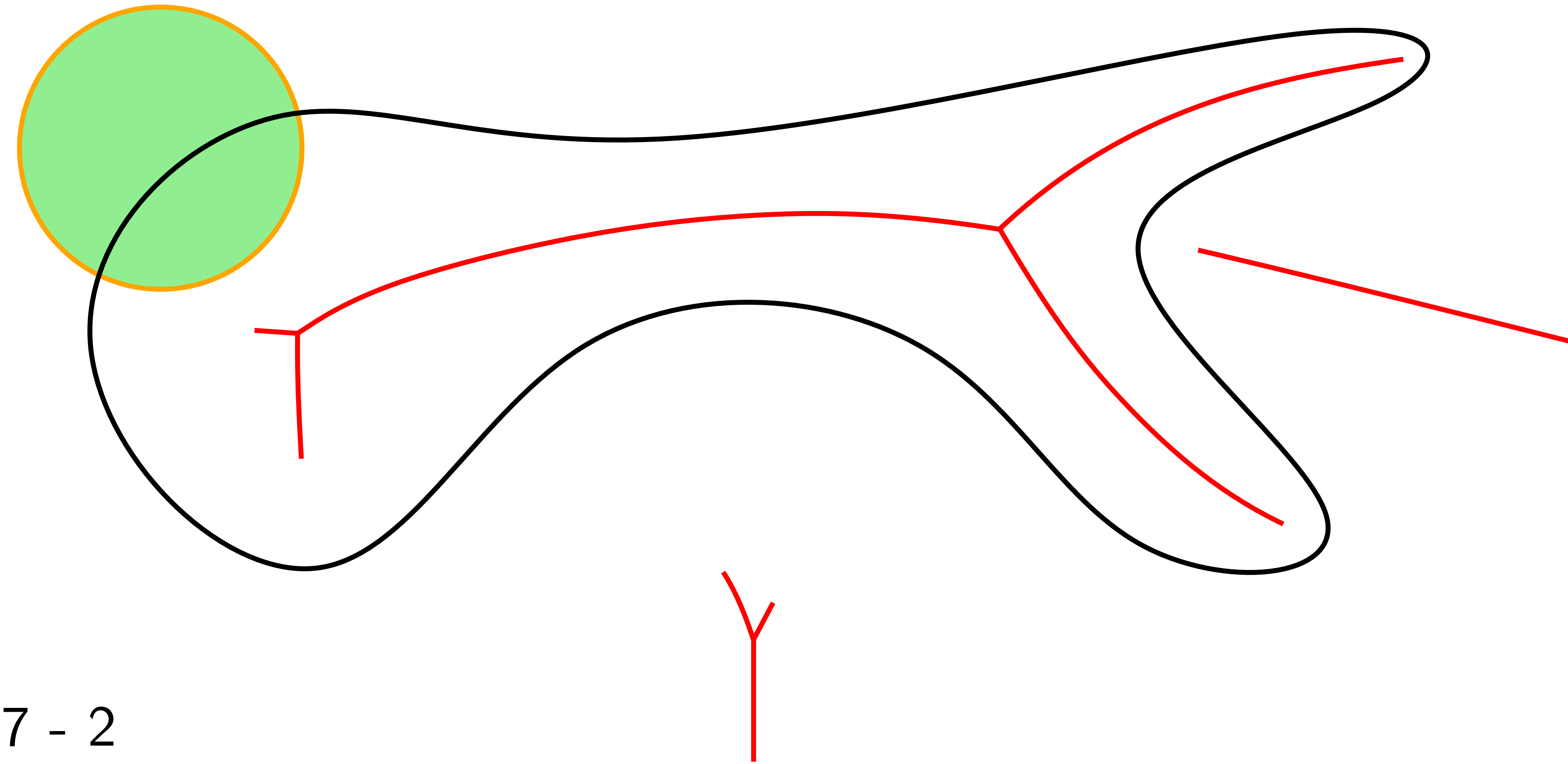
# Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$



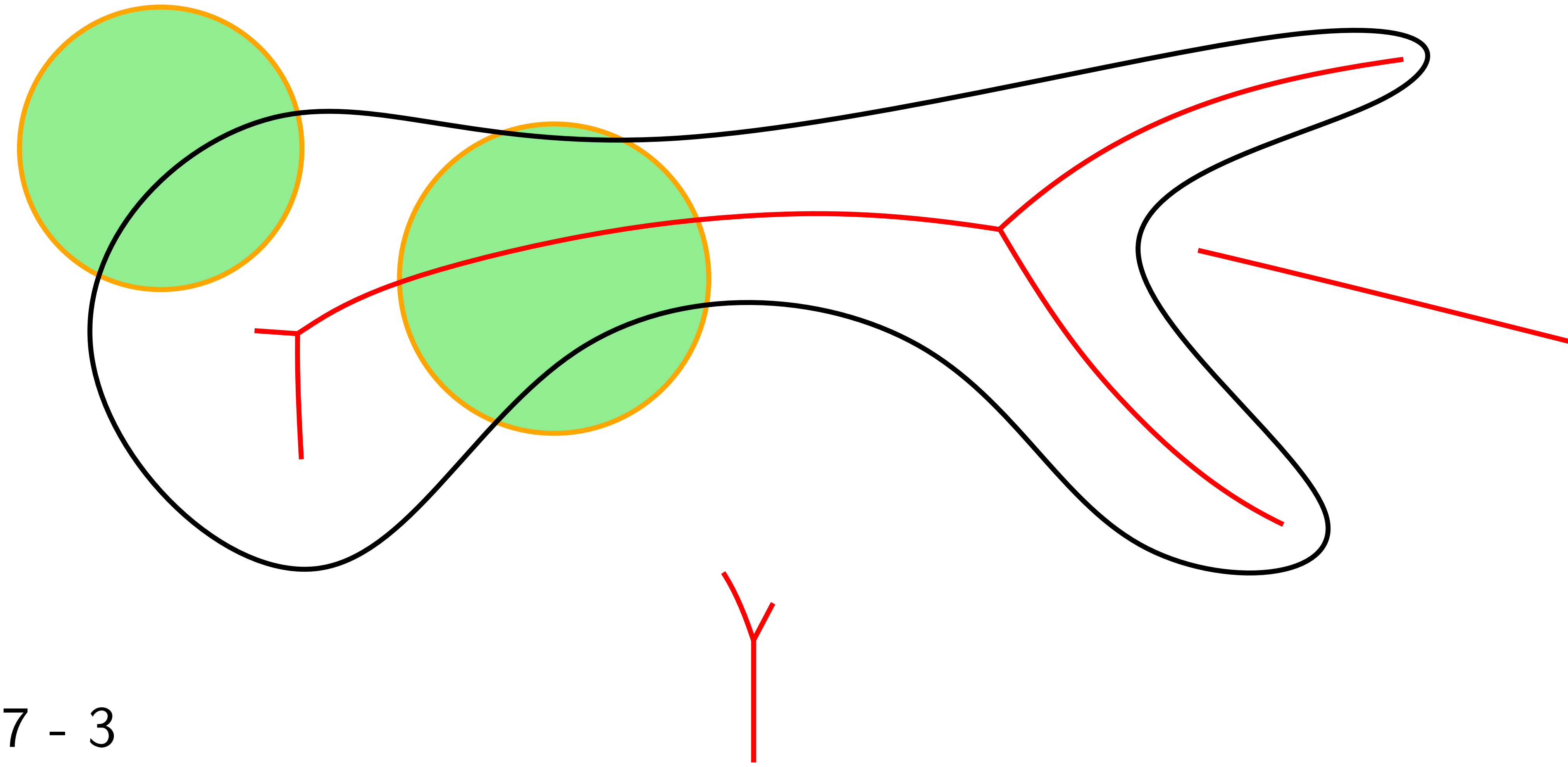
# Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$



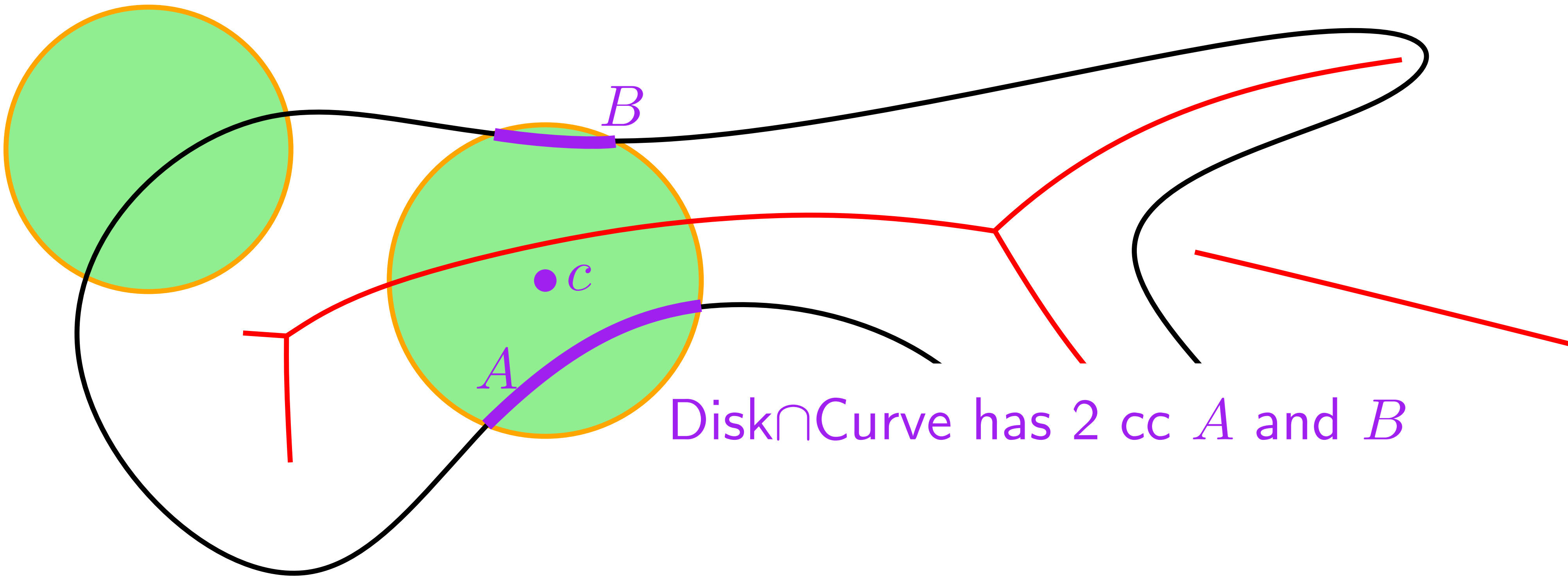
# Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$



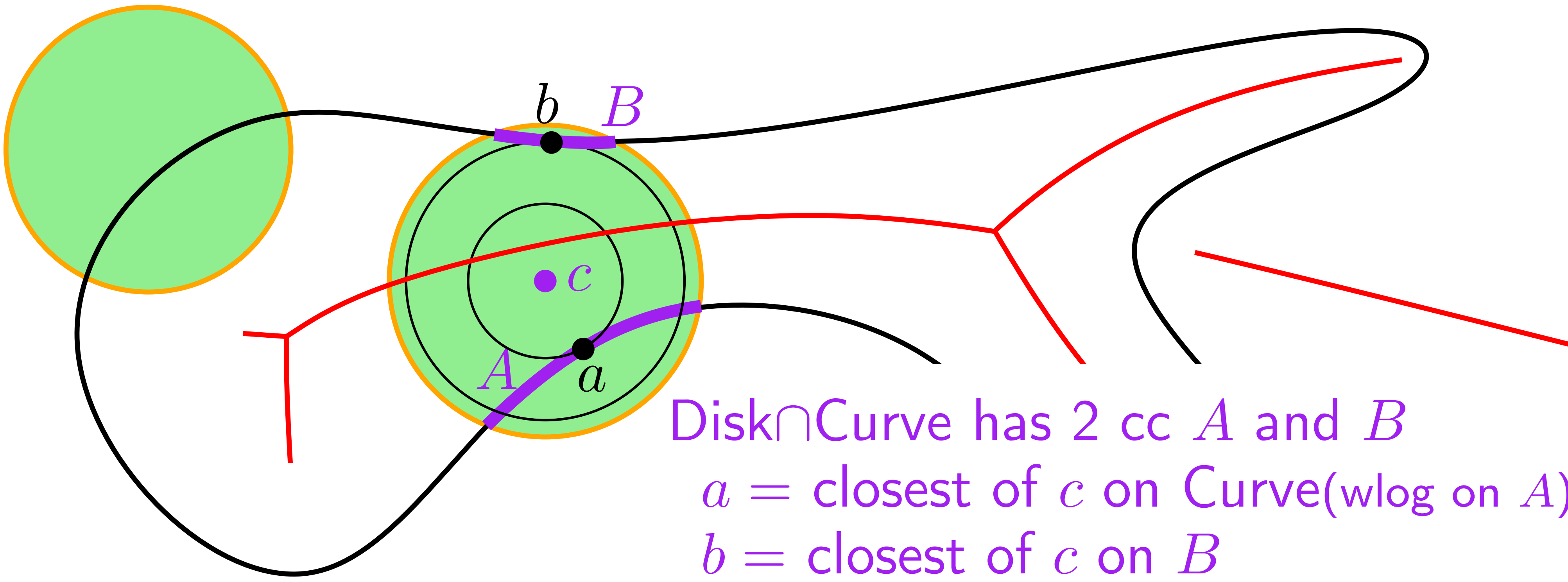
# Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$




Disk$\cap$Curve has 2 cc $A$ and $B$

$a =$ closest of $c$ on Curve(wlog on $A$)

$b =$ closest of $c$ on $B$

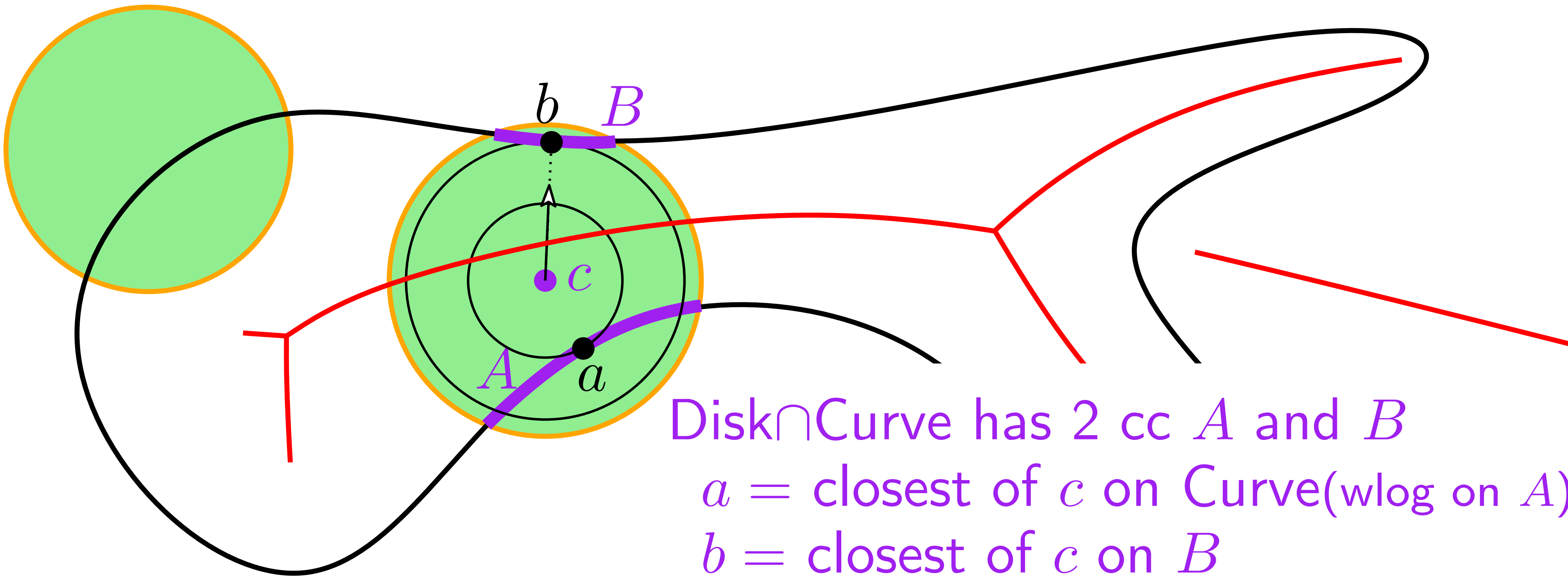
# Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$




Disk$\cap$Curve has 2 cc $A$ and $B$

$a =$ closest of $c$ on Curve(wlog on $A$)

$b =$ closest of $c$ on $B$

Moving from $c$ to $a$ dist to $B$ $\nearrow$

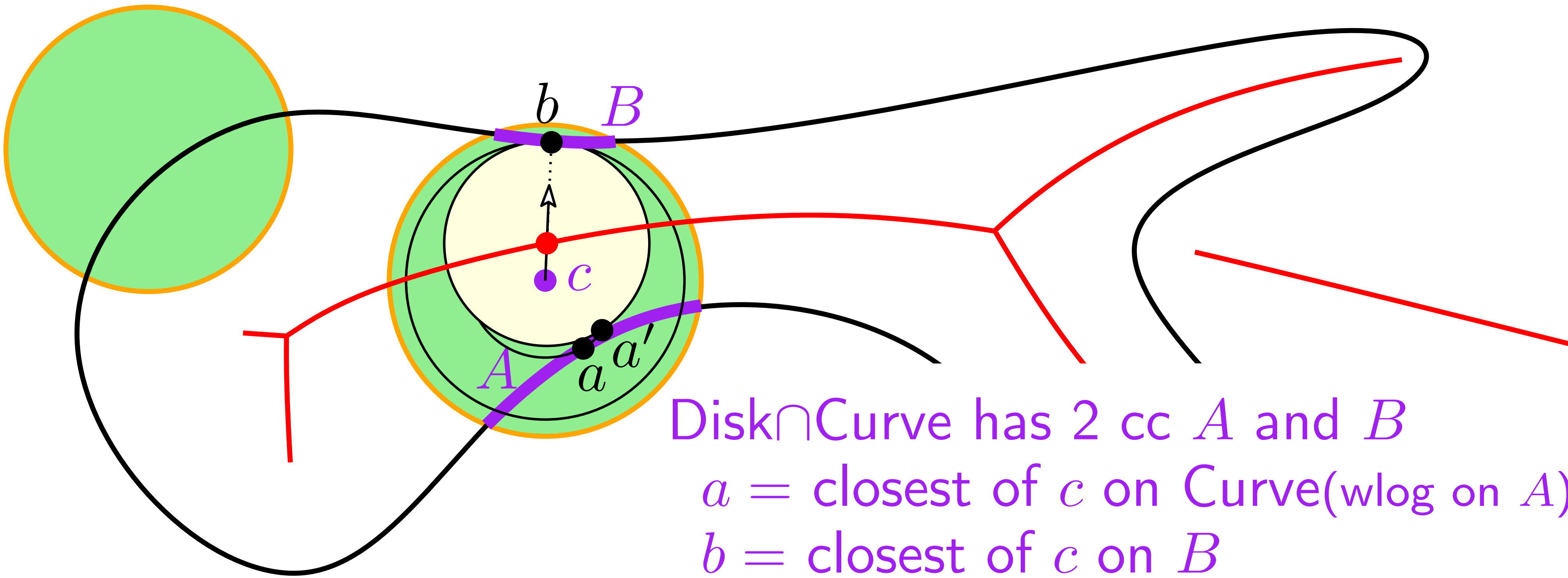
# Reconstruction

Delaunay is a good start

Lemma:

$\forall$ Disk, Disk$\cap$Curve has a single connected component

or Disk$\cap$Medial axis$\neq \emptyset$




Disk$\cap$Curve has 2 cc $A$ and $B$

$a =$ closest of $c$ on Curve(wlog on $A$)

$b =$ closest of $c$ on $B$

Moving from $c$ to $a$ dist to $B \nearrow$

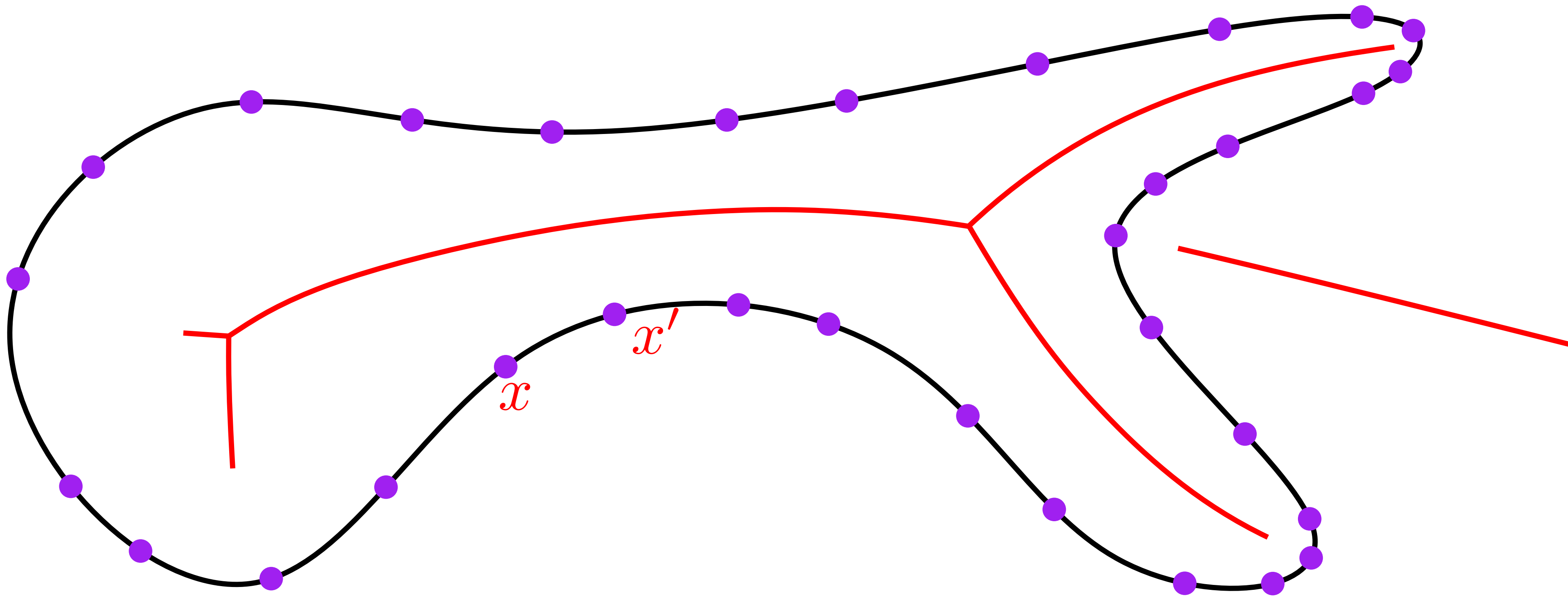
reach center of bitangent disk

# Reconstruction

## Delaunay is a good start

Theorem
If Sample is a $\epsilon$-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors

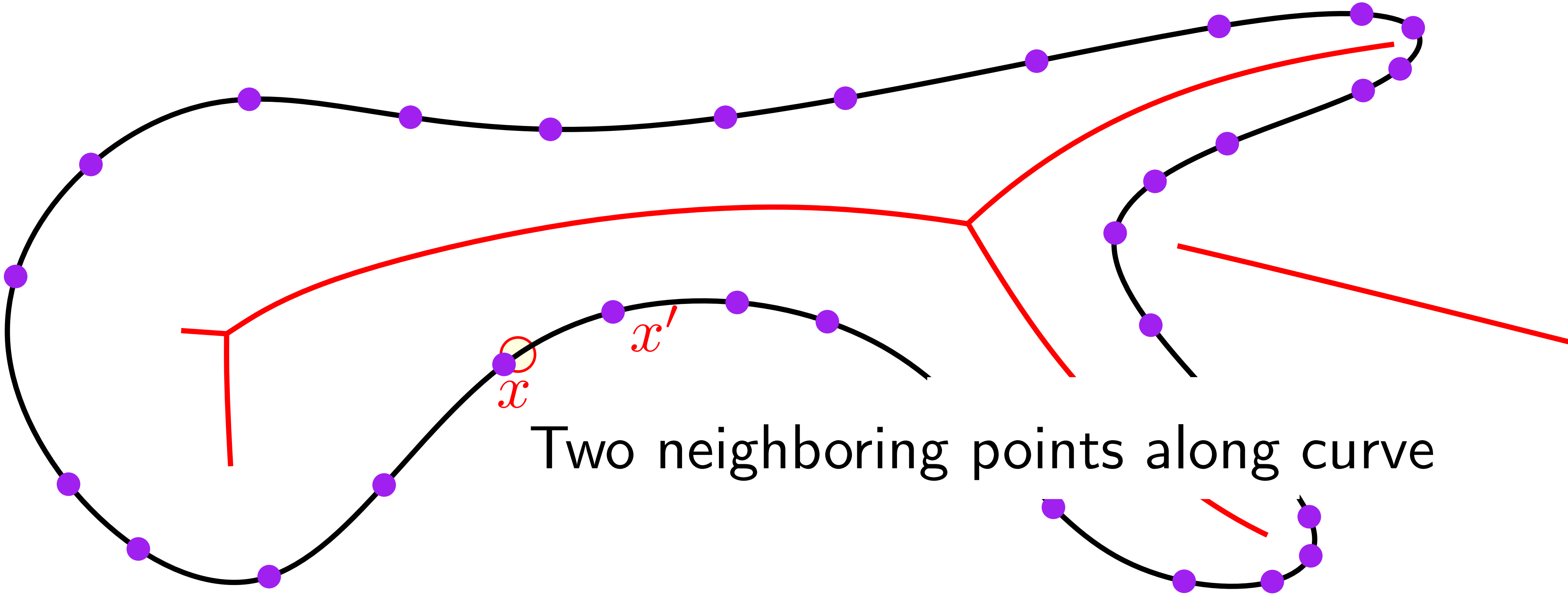


# Reconstruction

Delaunay is a good start

Theorem
If Sample is a $\epsilon$-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors



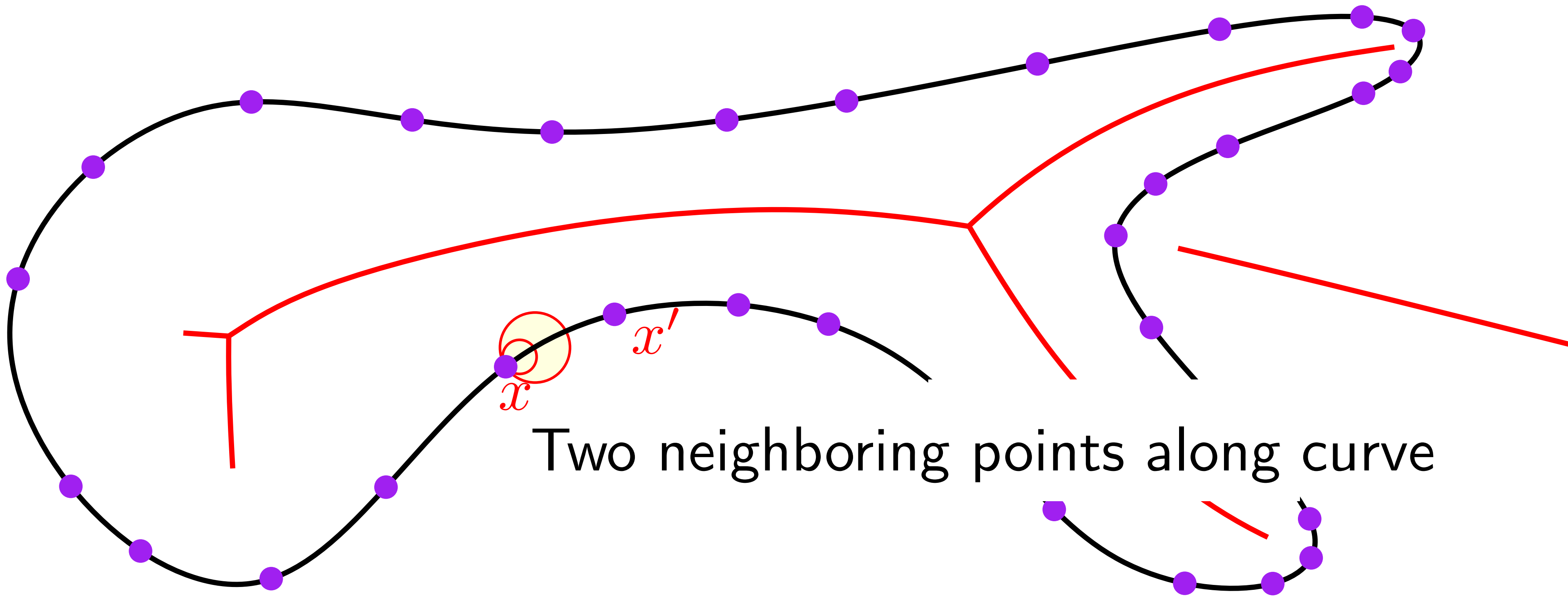

disks centered on Curve, through $x$

# Reconstruction

Delaunay is a good start

Theorem
If Sample is a $\epsilon$-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors



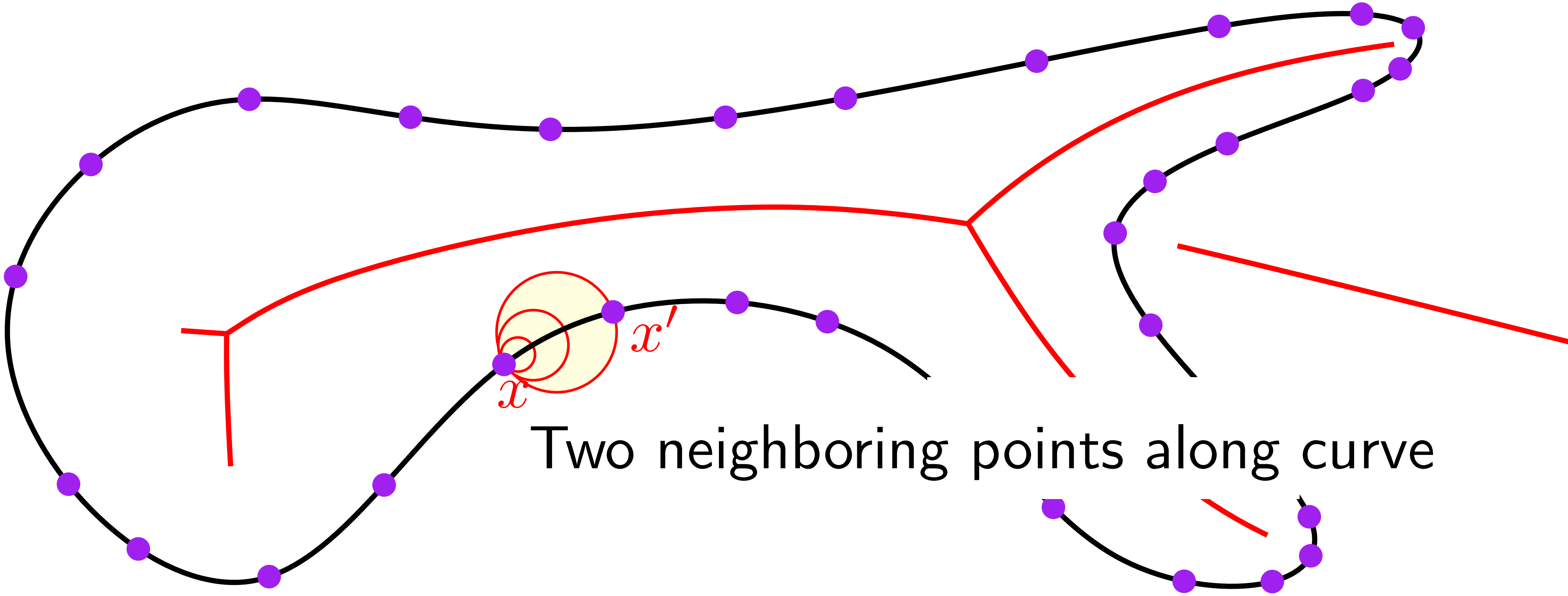

disks centered on Curve, through $x$

# Reconstruction

Delaunay is a good start

Theorem
If Sample is a $\epsilon$-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors




Two neighboring points along curve

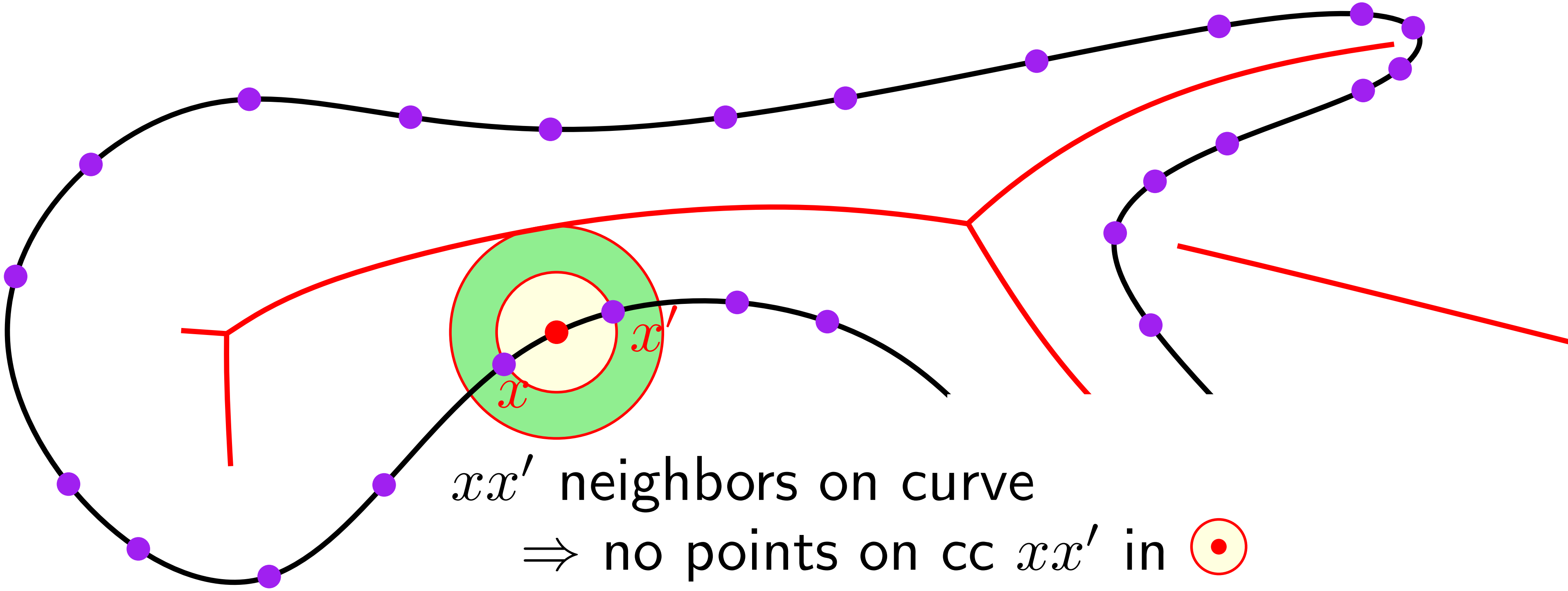
disks centered on Curve, through $x$

# Reconstruction

Delaunay is a good start

Theorem
If Sample is a $\epsilon$-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors




$xx'$ neighbors on curve

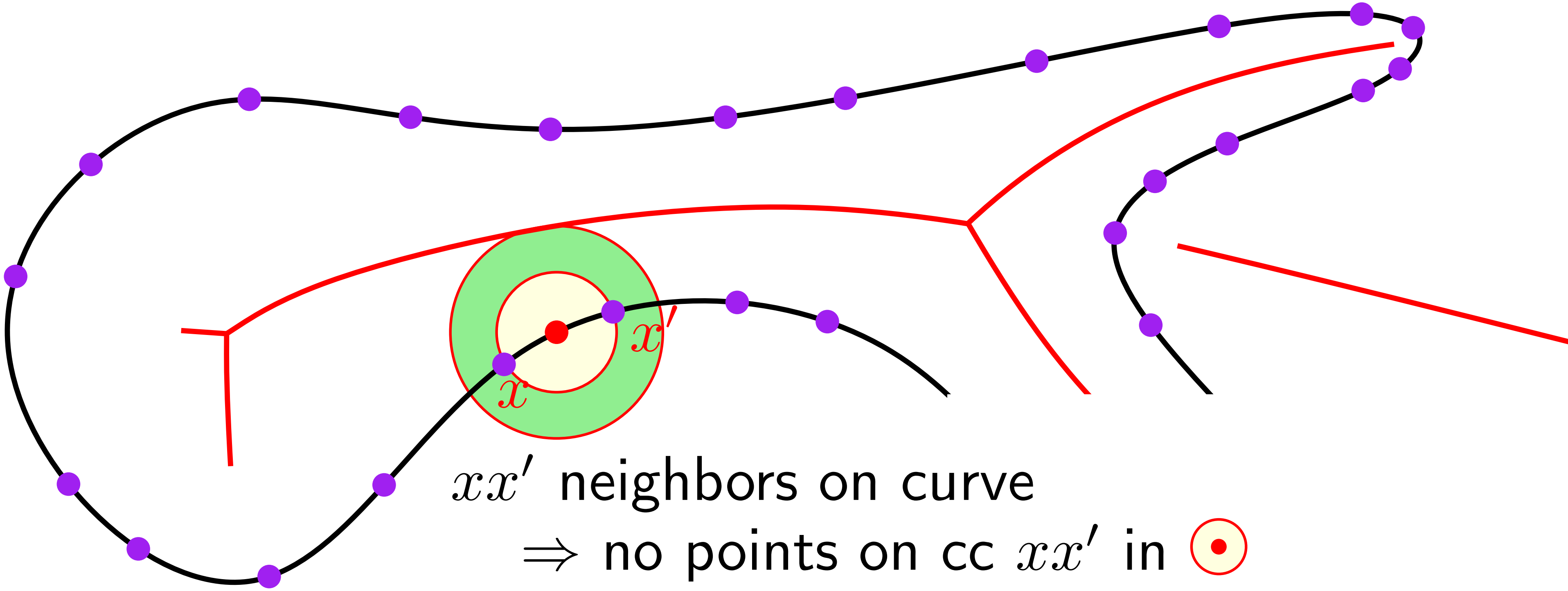
$\Rightarrow$ no points on cc $xx'$ in ⊙

# Reconstruction

## Delaunay is a good start

Theorem
If Sample is a $\epsilon$-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors




$xx'$ neighbors on curve
$\Rightarrow$ no points on cc $xx'$ in

1-sampling $\Rightarrow$ $\subset$

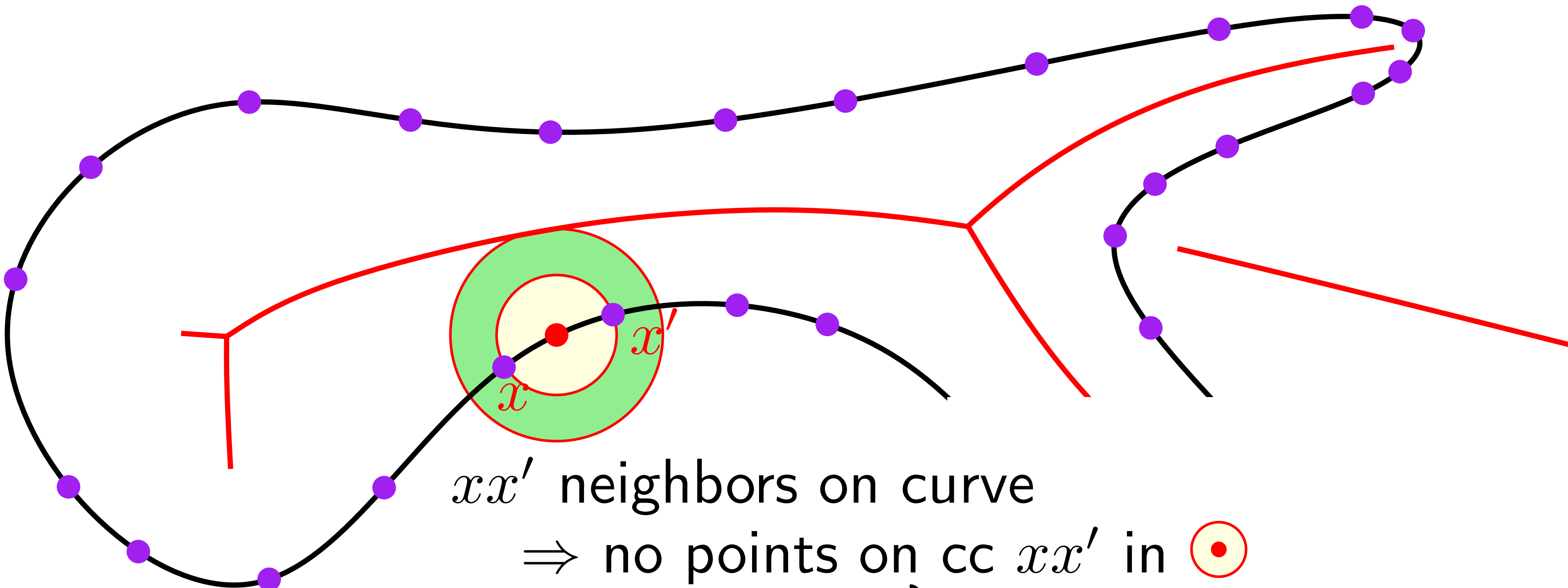
# Reconstruction

## Delaunay is a good start

Theorem

If Sample is a $\epsilon$-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors




$xx'$ neighbors on curve

$\Rightarrow$ no points on cc $xx'$ in

1-sampling $\Rightarrow$ $\subset$ $\Big\} \Rightarrow$ no other cc $\cap$

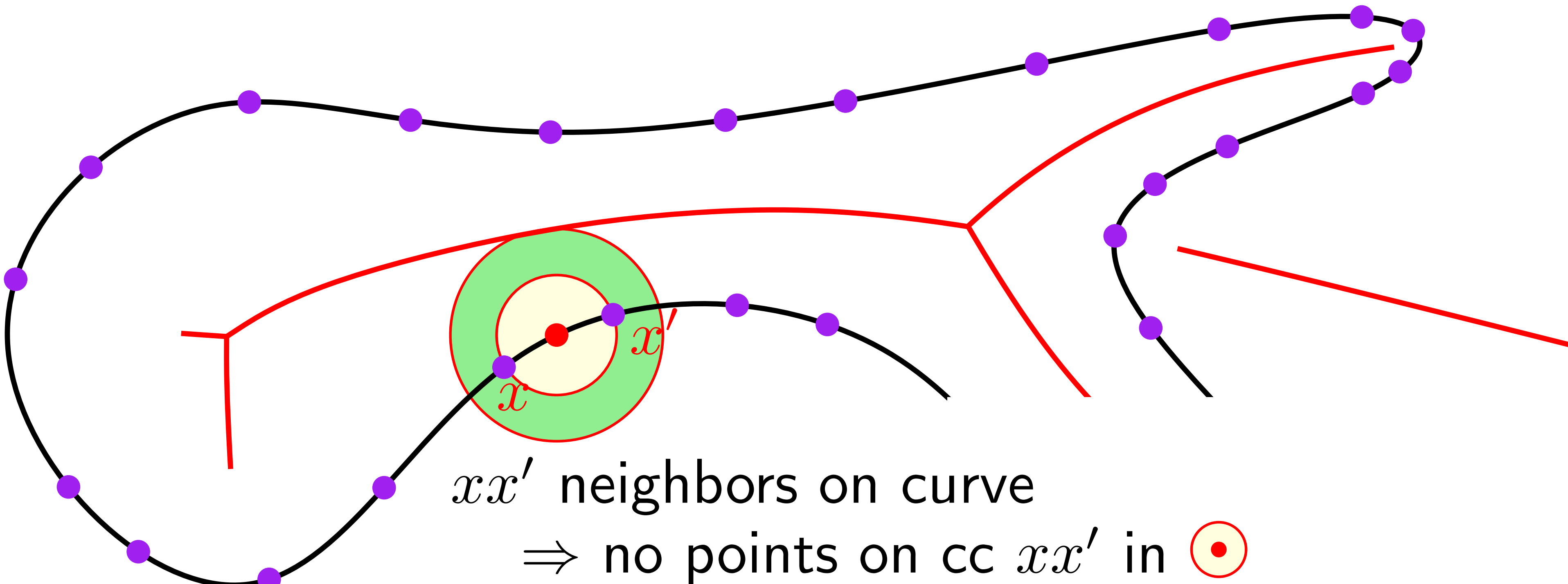
Lemma

# Reconstruction

## Delaunay is a good start

Theorem
If Sample is a $\epsilon$-sample, $\epsilon < 1$

neighboring points along Curve are Delaunay neighbors




$xx'$ neighbors on curve

$\Rightarrow$ no points on cc $xx'$ in

1-sampling $\Rightarrow$ $\subset$

Lemma

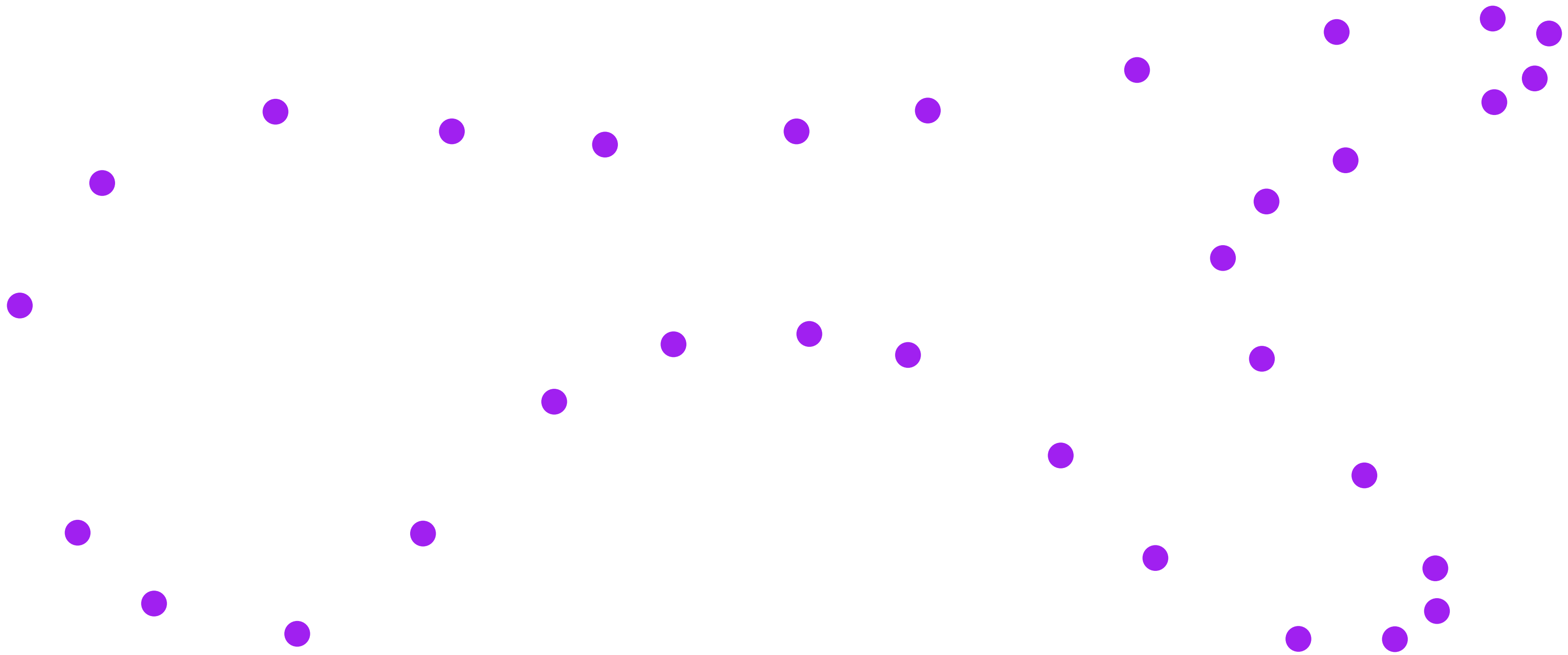
$\Rightarrow$ no other cc $\cap$

$\Rightarrow$ empty

# Reconstruction

Delaunay is a good start

Given a sampling

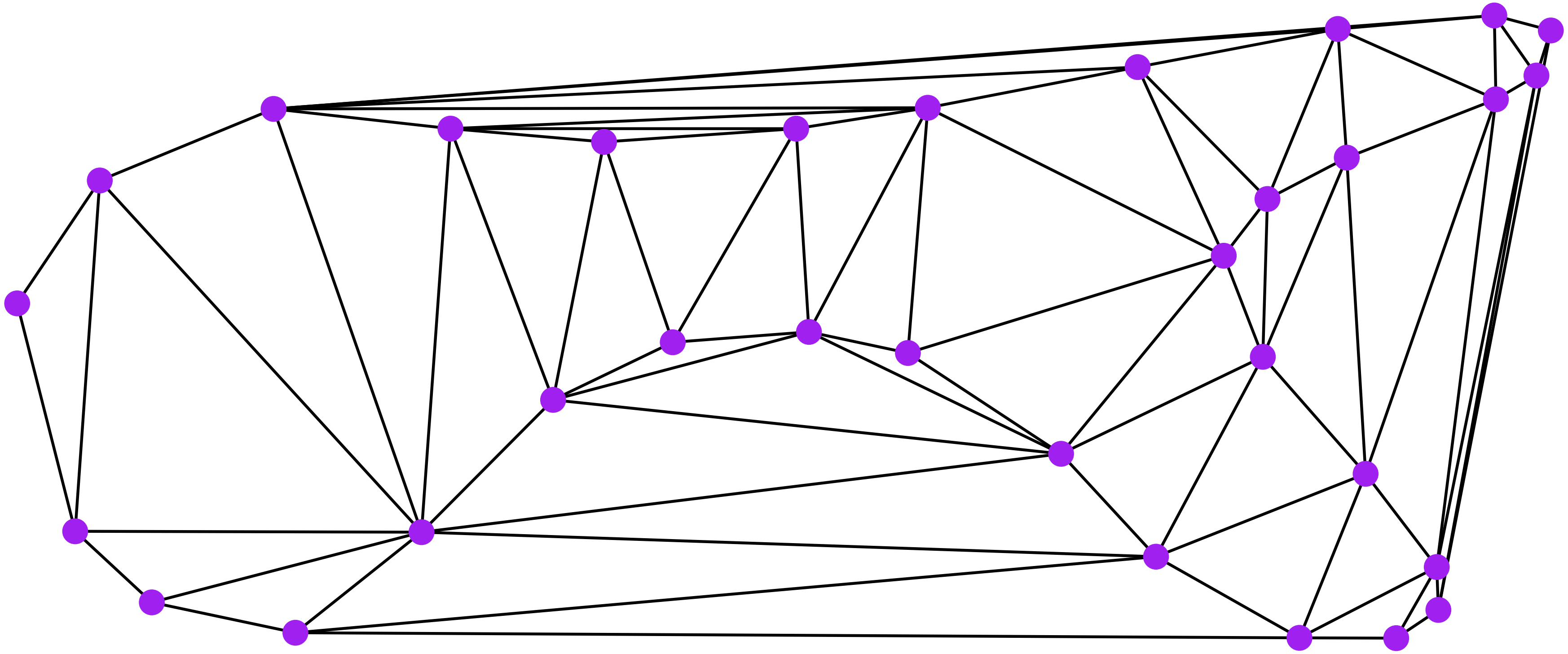


# Reconstruction

Delaunay is a good start

Given a sampling

Compute Delaunay



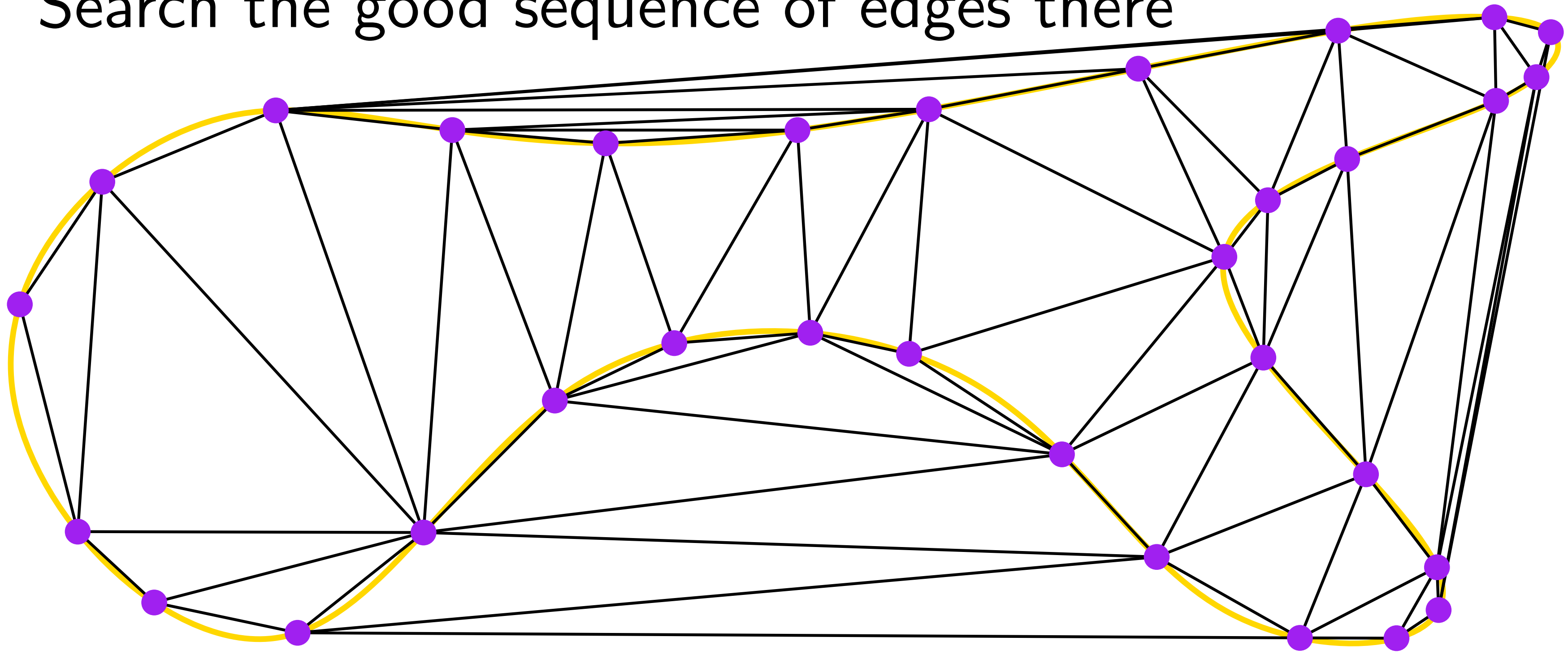
# Reconstruction

Delaunay is a good start

Given a sampling

Compute Delaunay

Search the good sequence of edges there



# Reconstruction

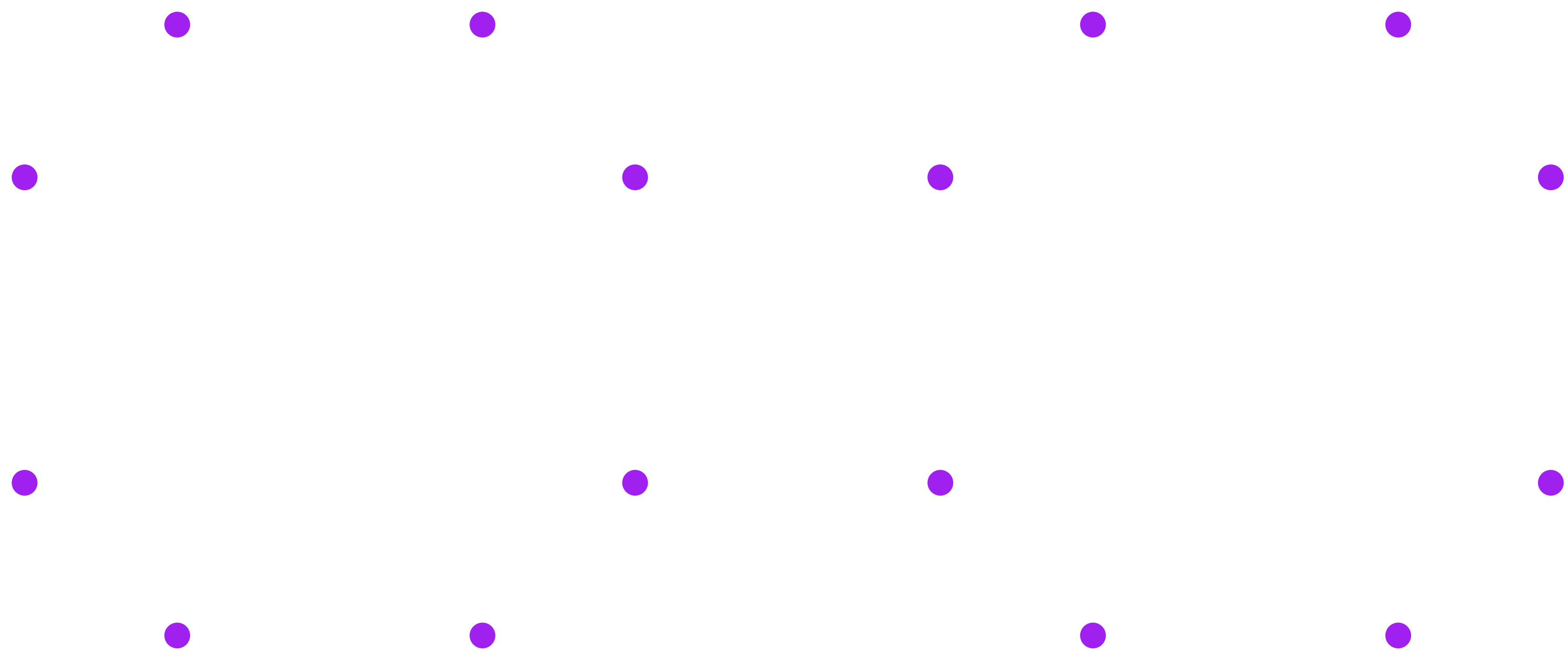
## Delaunay is a good start

1-sample is not enough

# Reconstruction

Delaunay is a good start

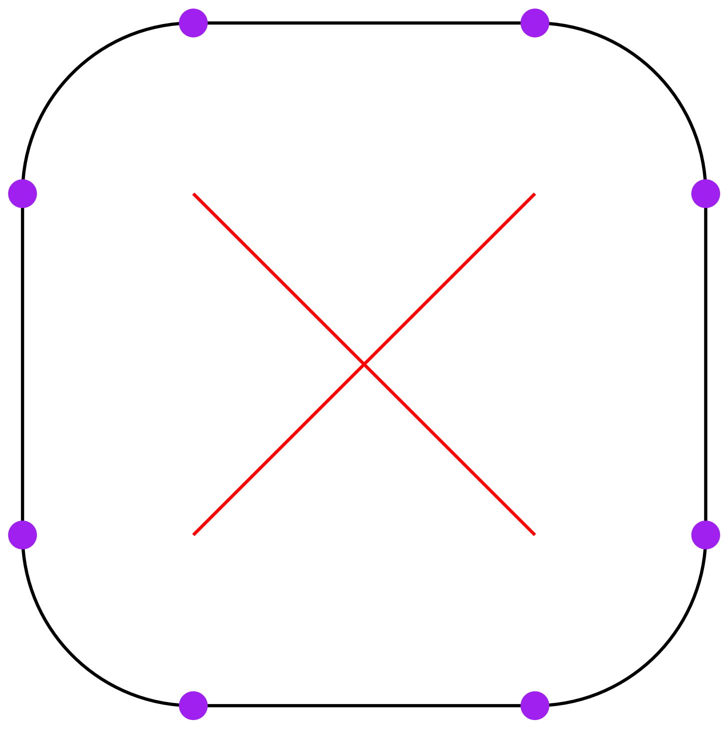
1-sample is not enough

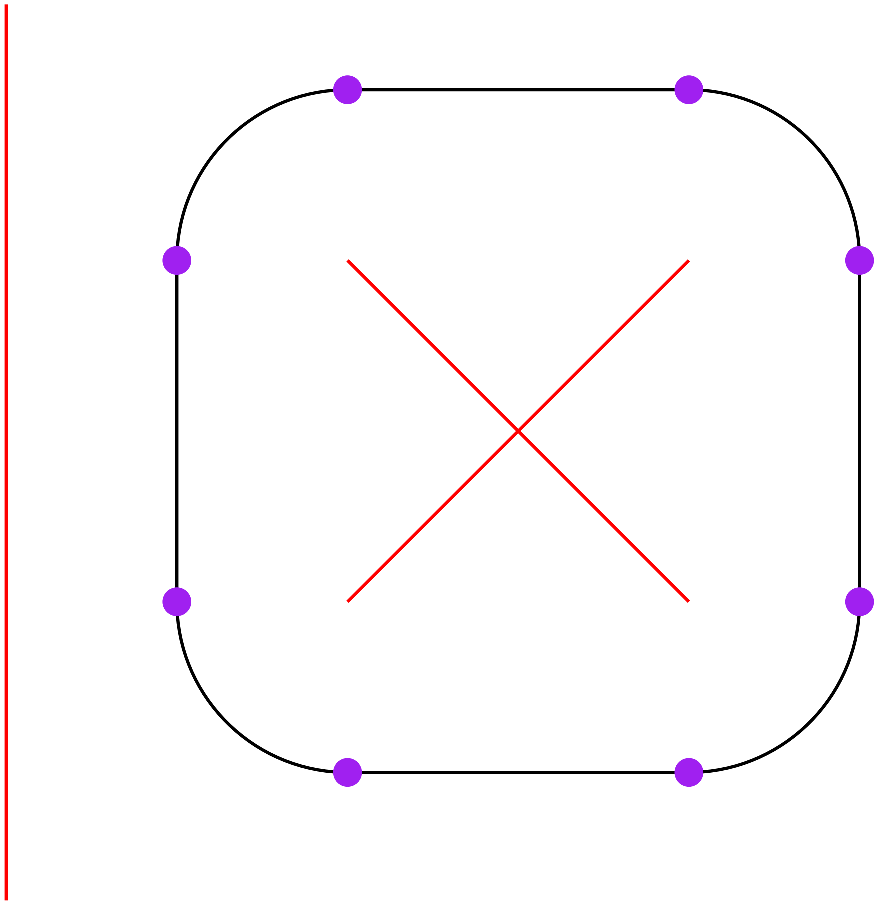


# Reconstruction

Delaunay is a good start

1-sample is not enough

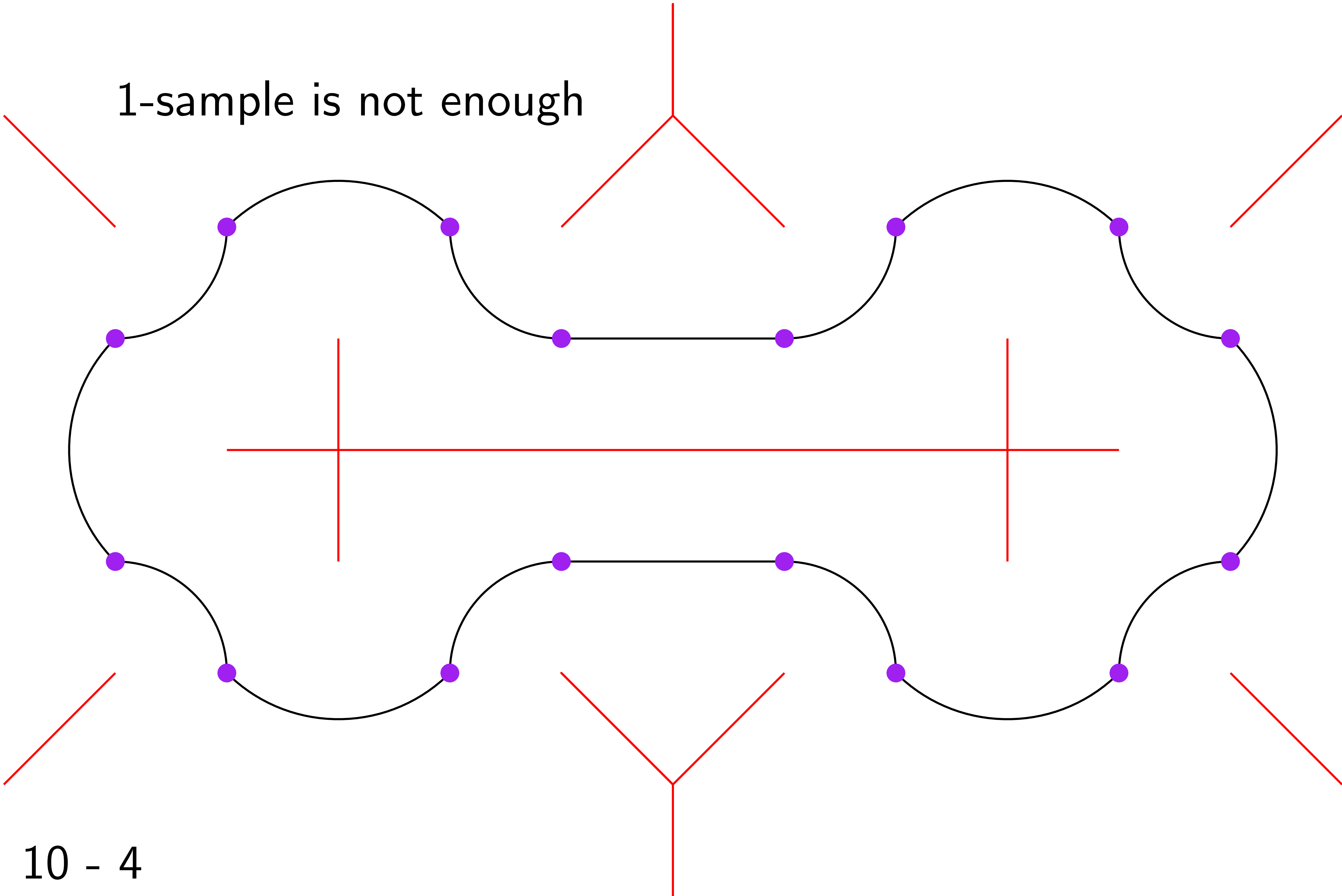




# Reconstruction

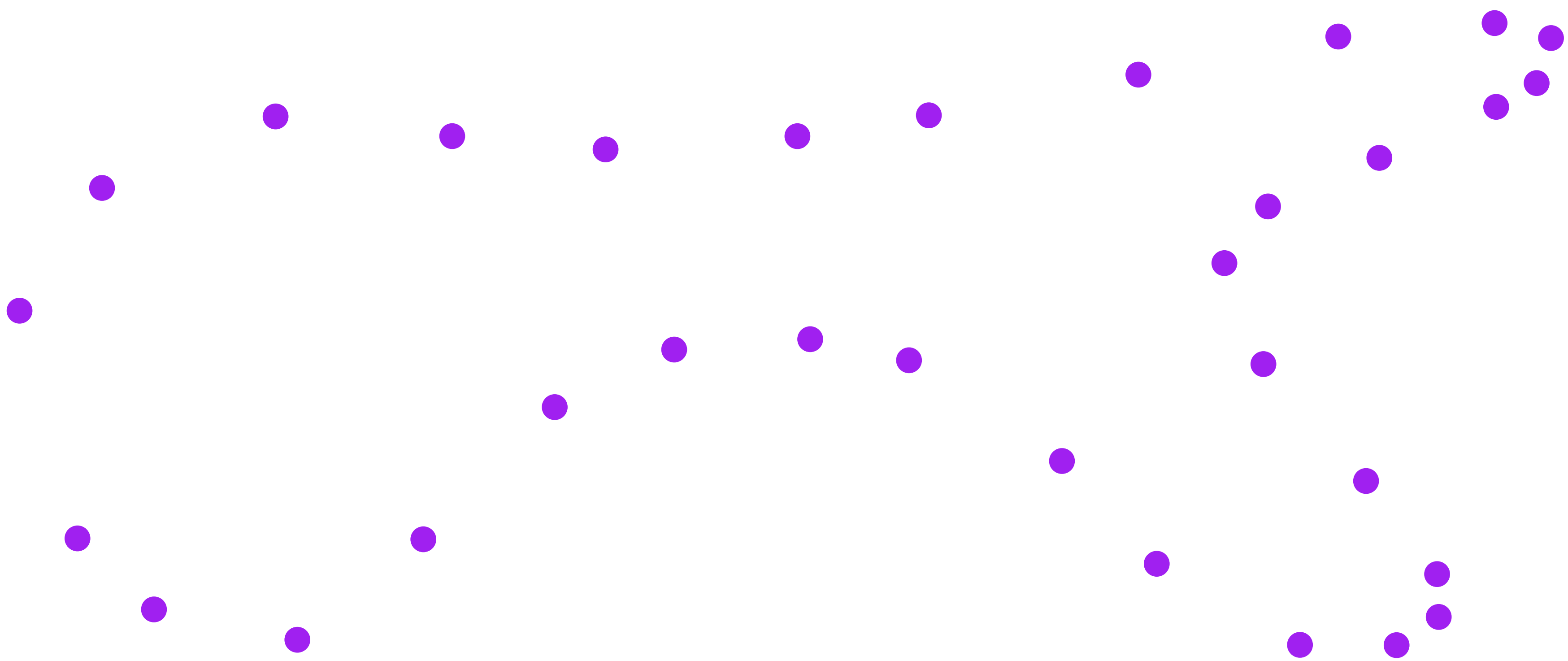
Delaunay is a good start

1-sample is not enough



# Reconstruction

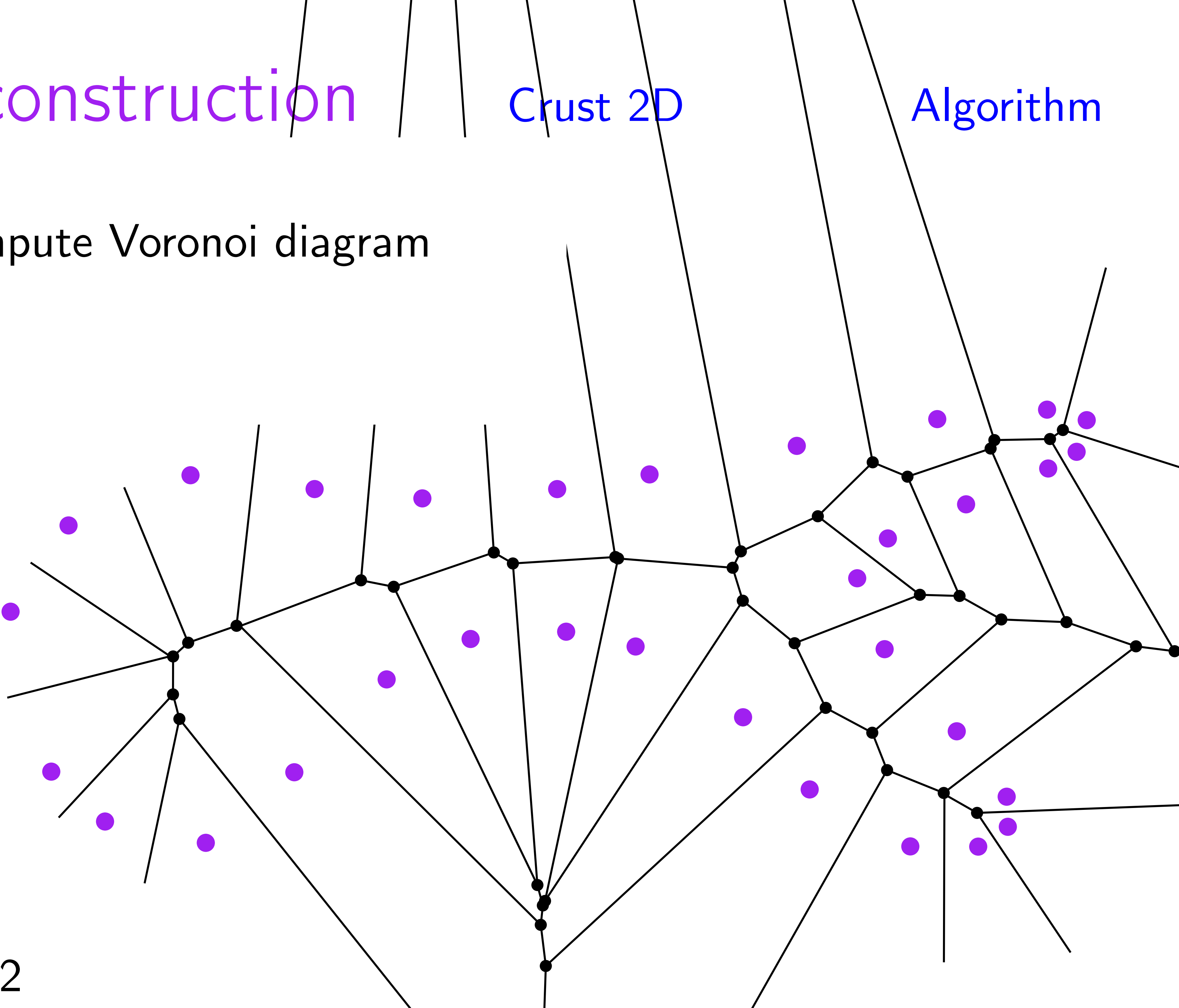




# Reconstruction

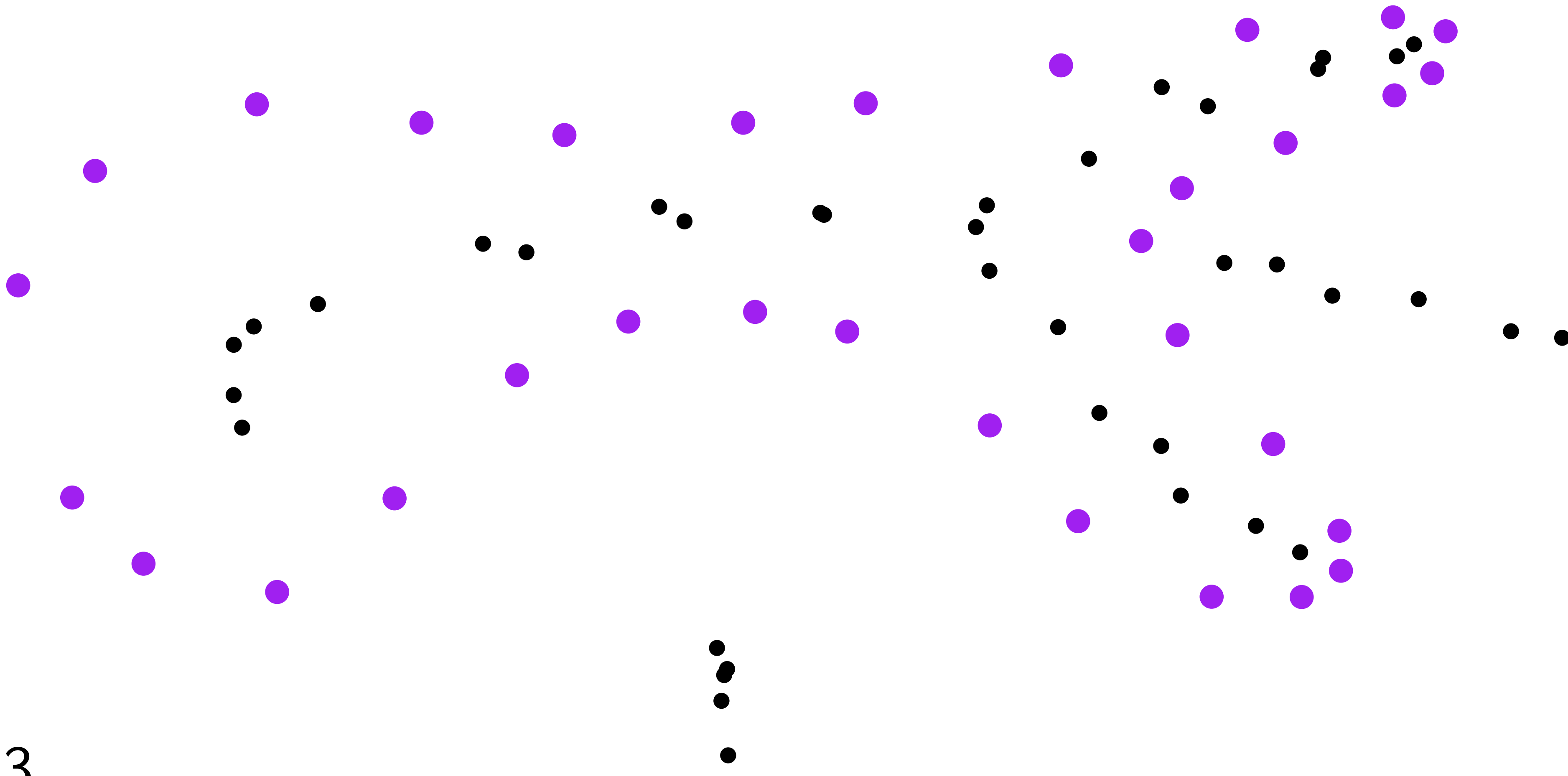


Compute Voronoi diagram



# Reconstruction Crust 2D Algorithm

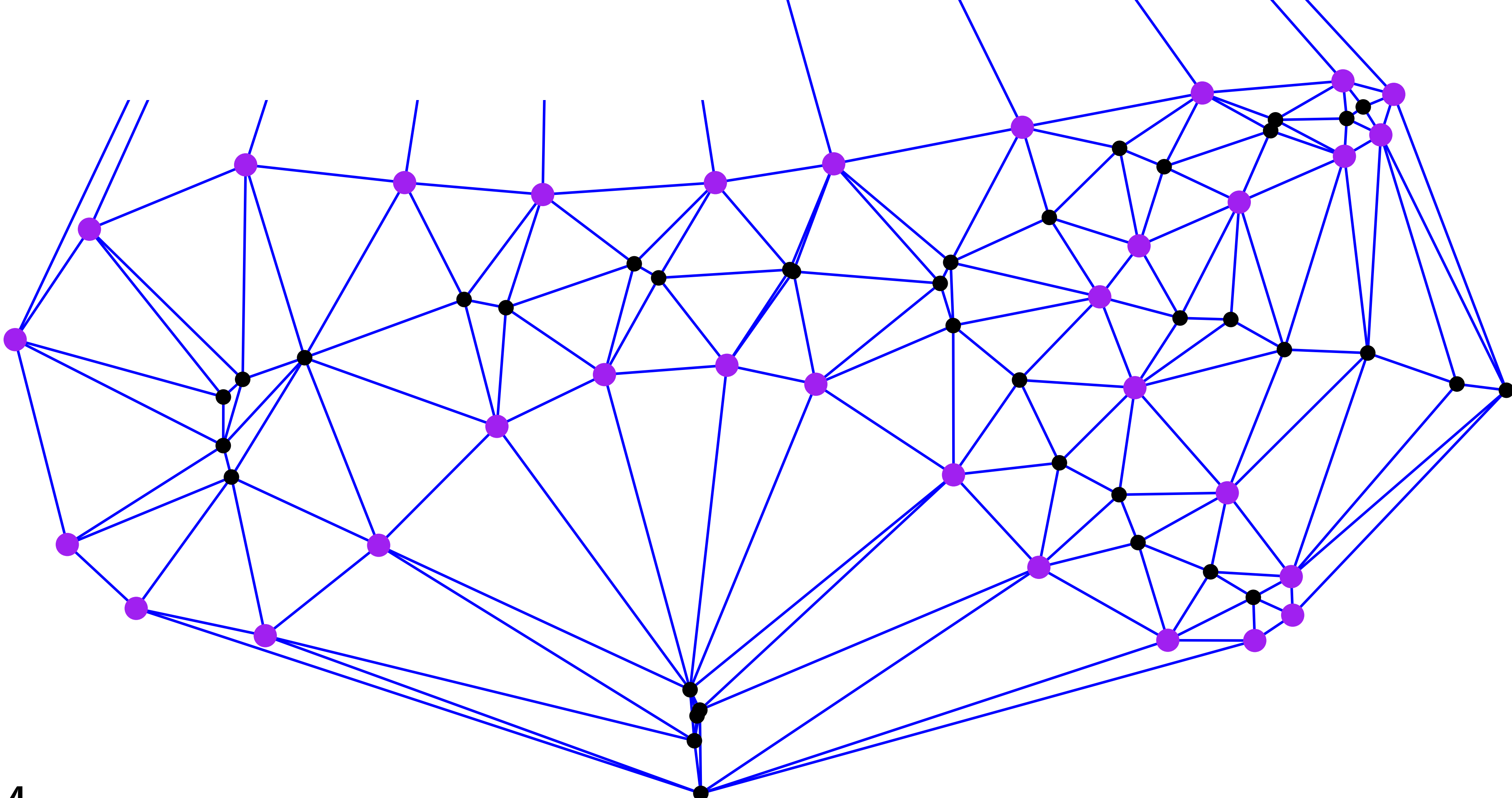
Keep Voronoi vertices



# Reconstruction Crust 2D Algorithm

Keep Voronoi vertices

Compute Delaunay triangulation

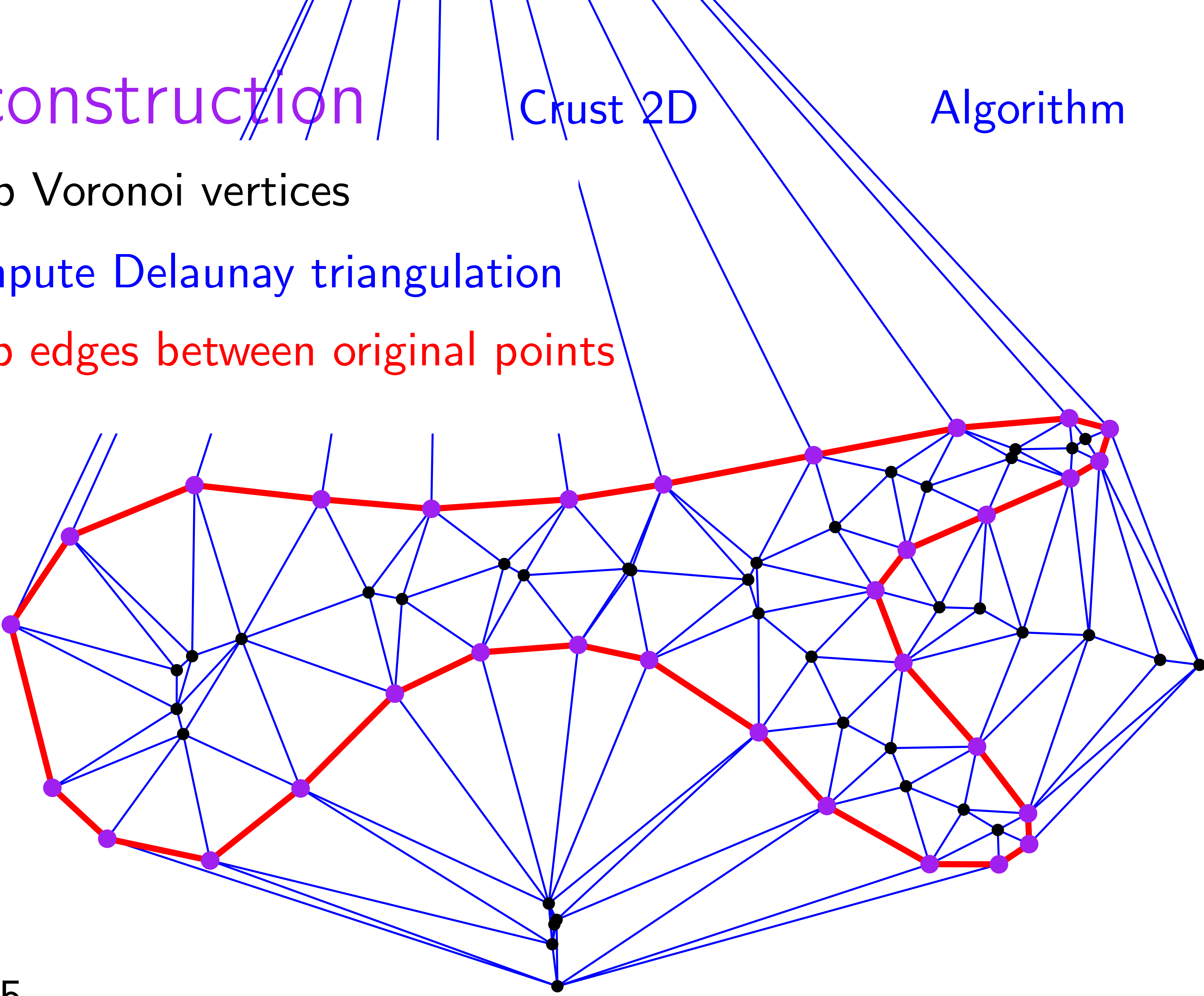


# Reconstruction    Crust 2D    Algorithm

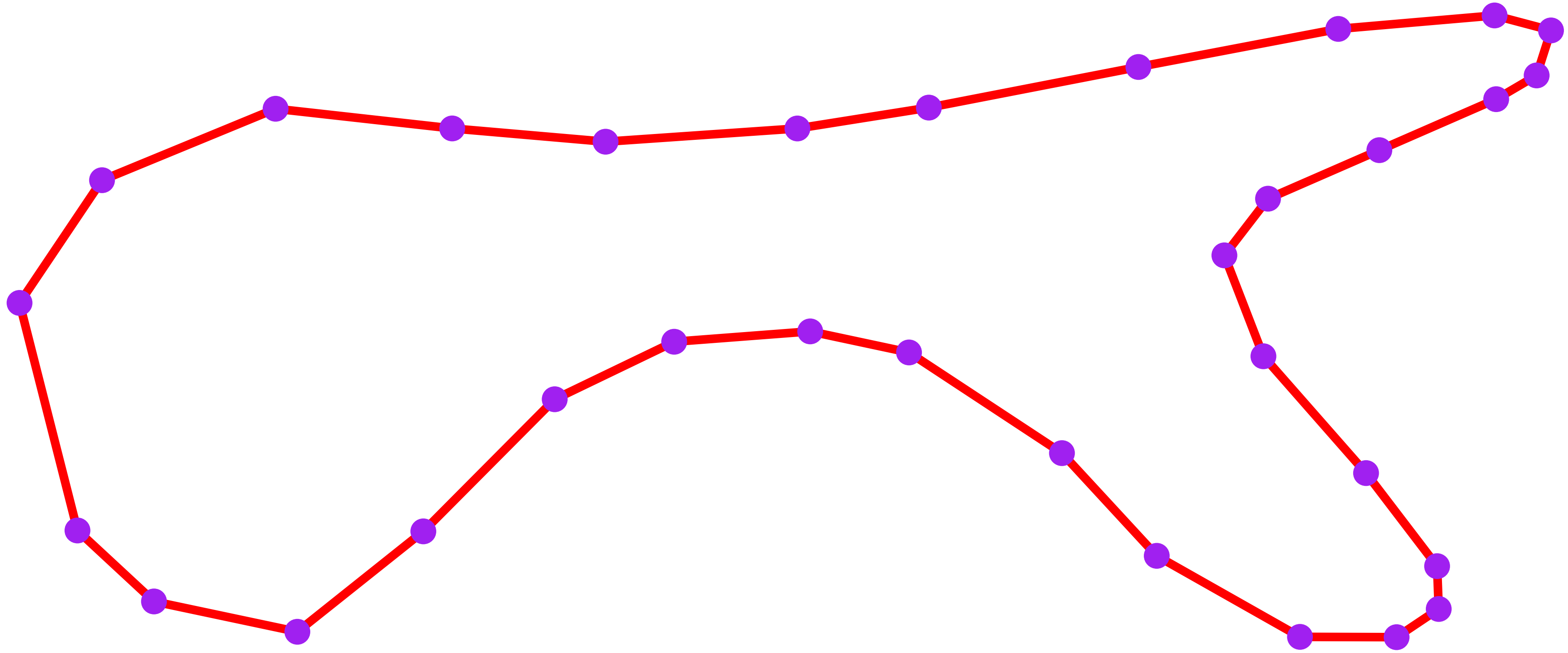
Keep Voronoi vertices

Compute Delaunay triangulation

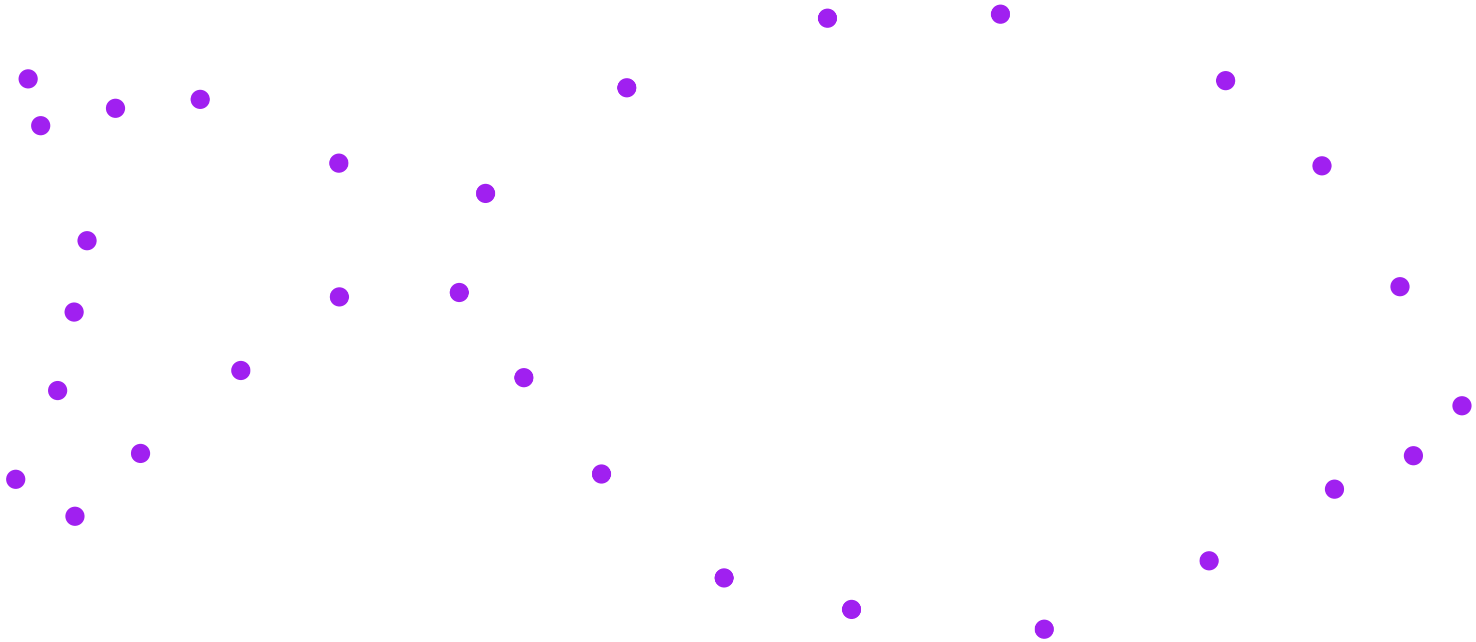
Keep edges between original points

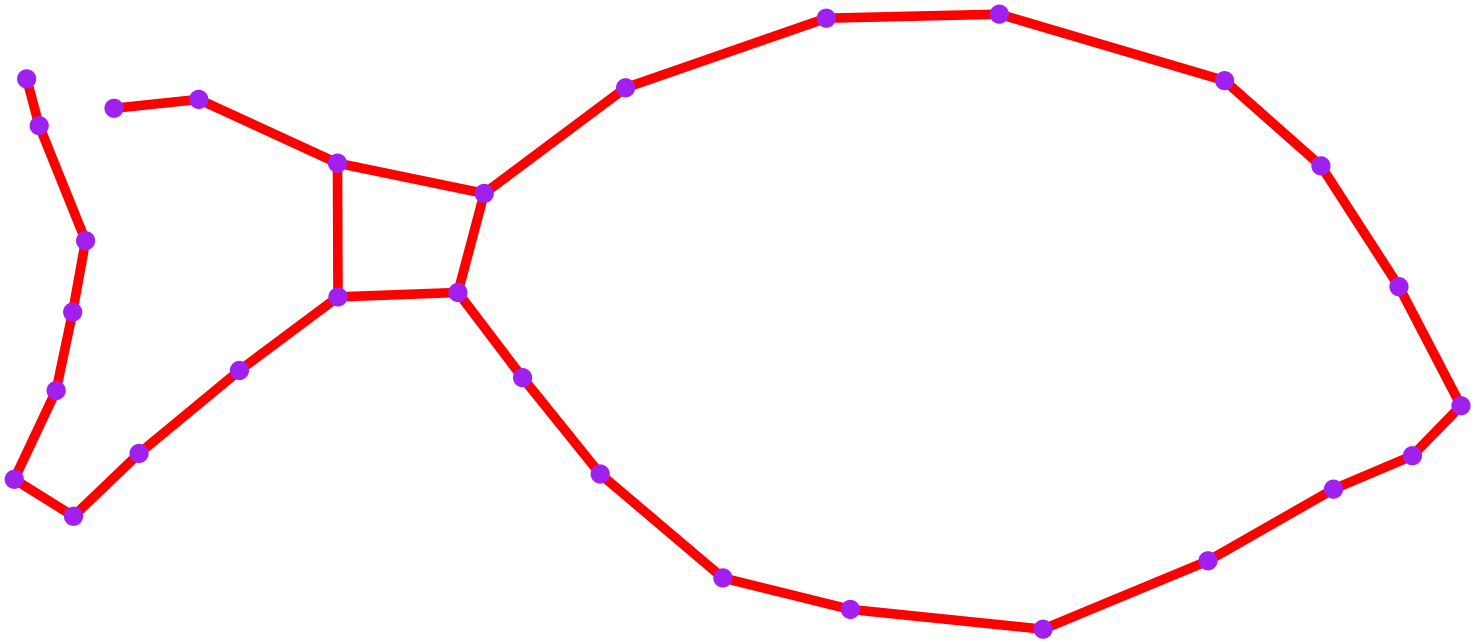


Keep edges between original points

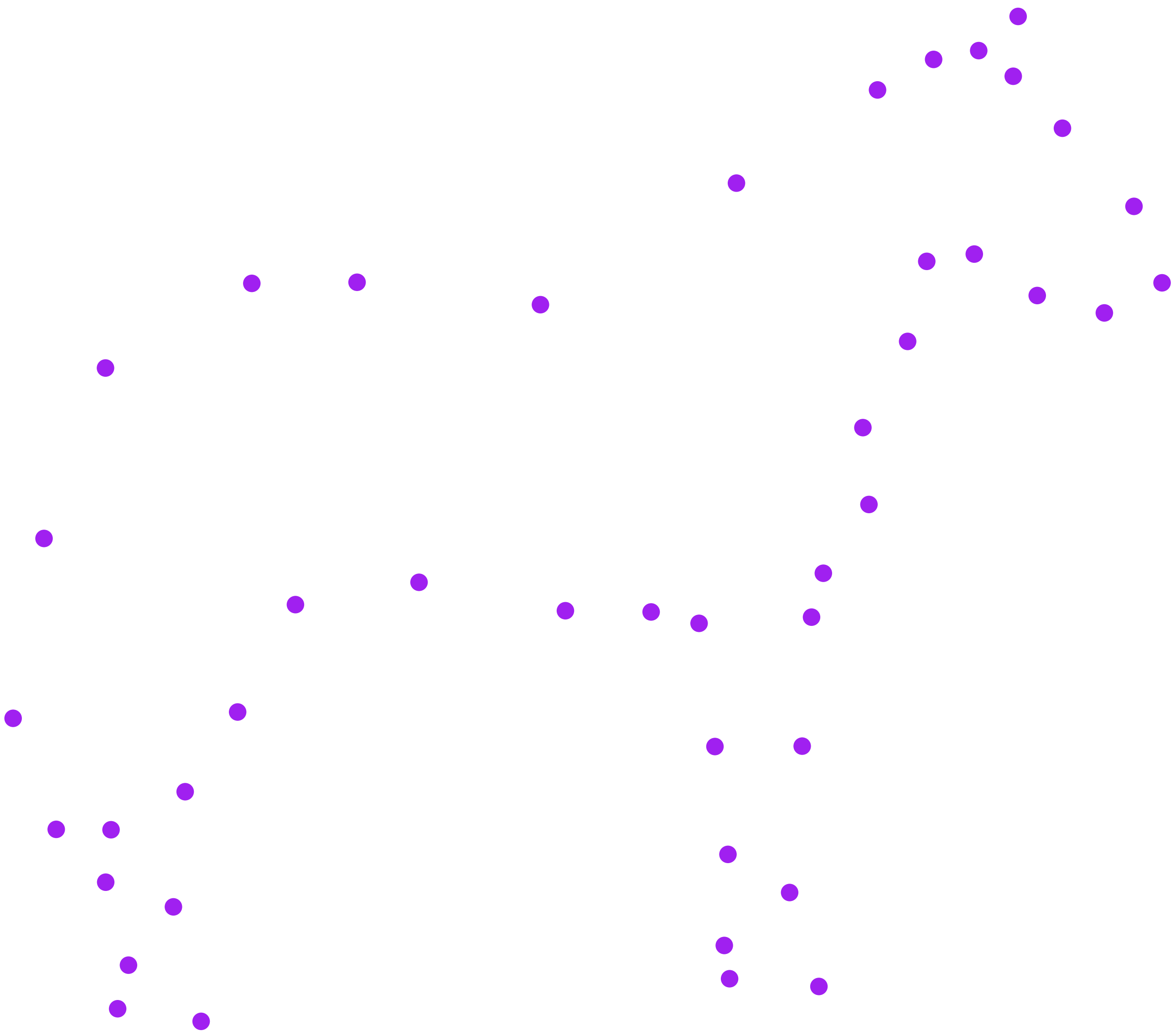


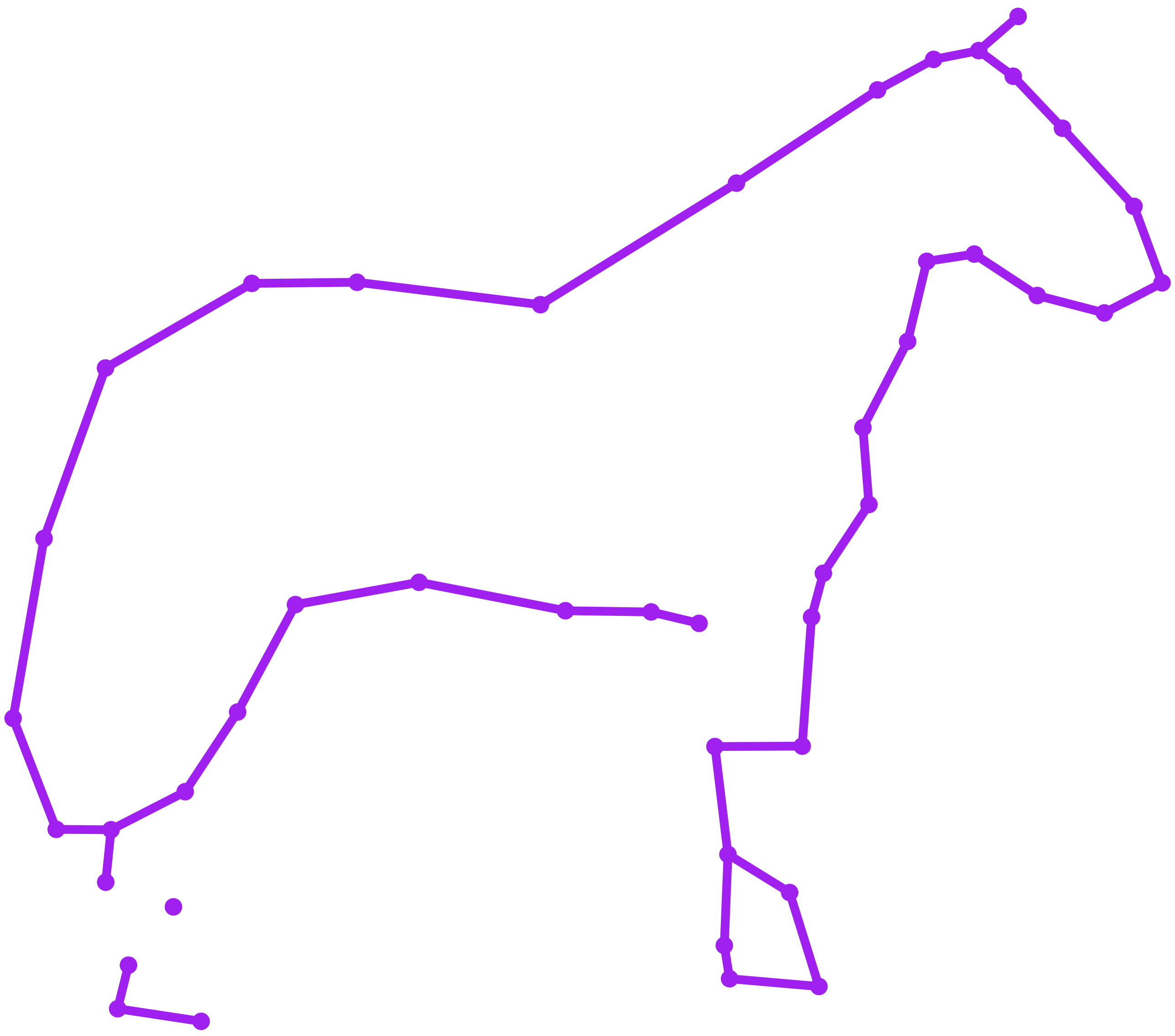
# Reconstruction Crust 2D Algorithm



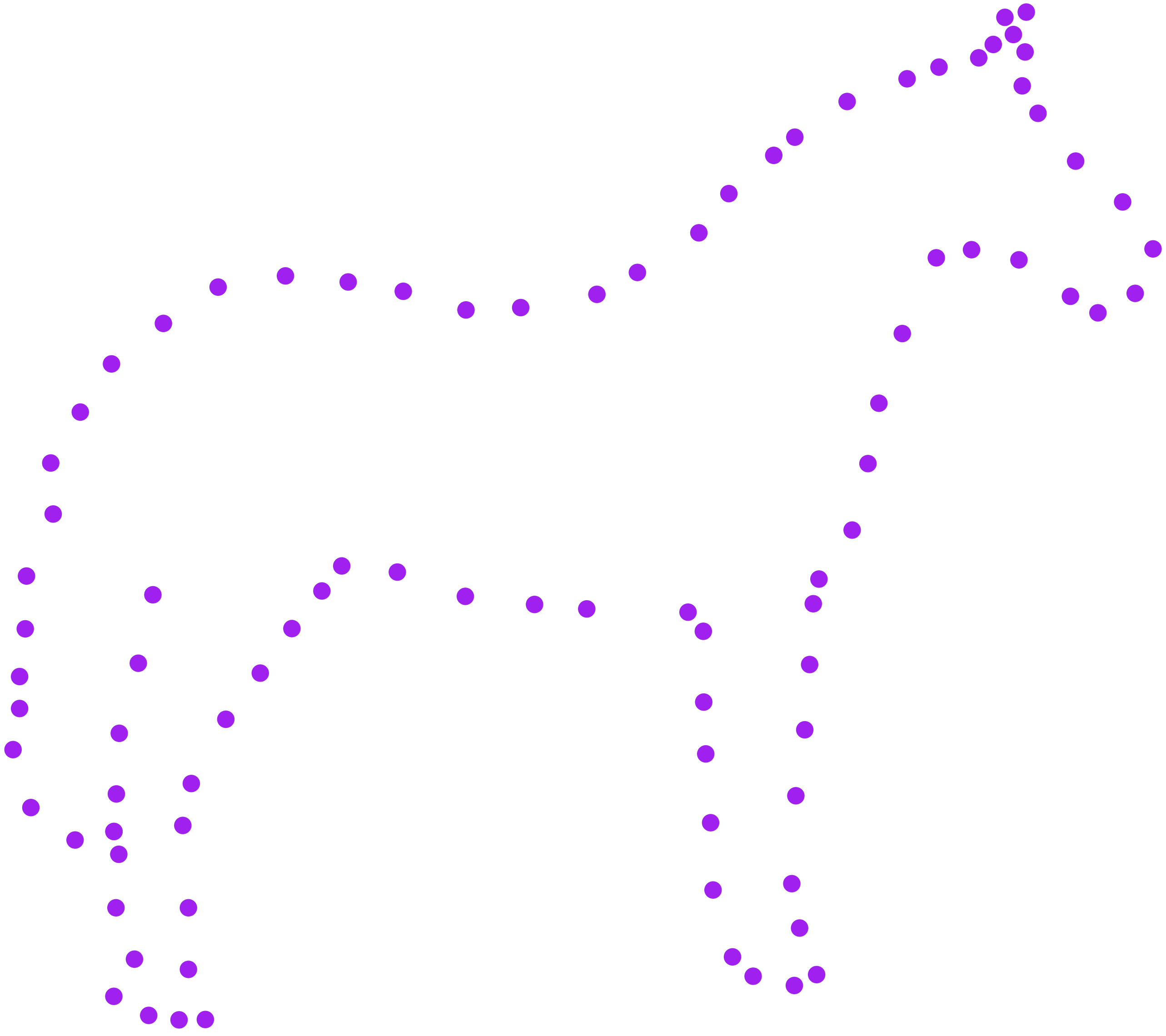


# Reconstruction Crust 2D Algorithm

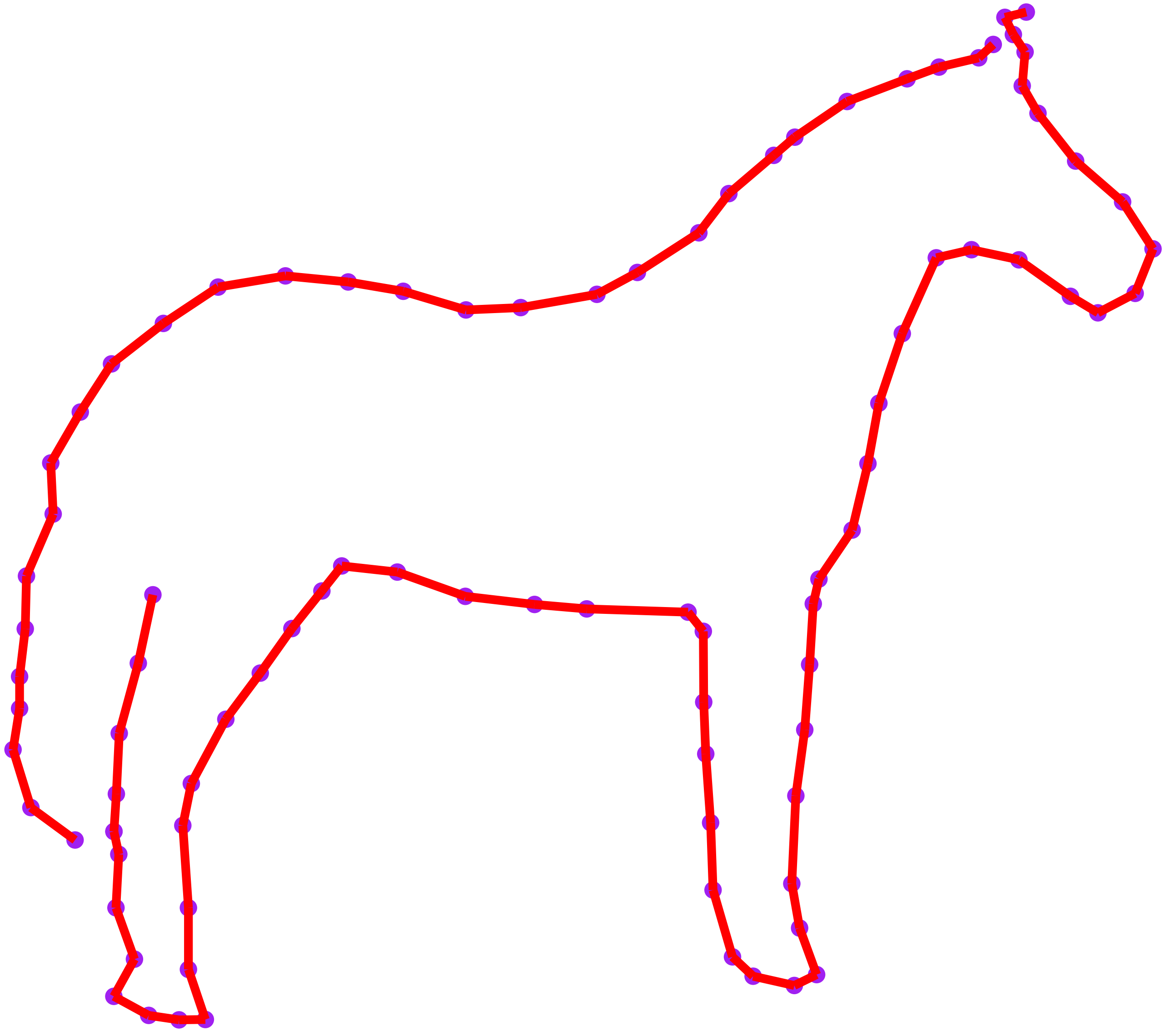




# Reconstruction Crust 2D Algorithm



# Reconstruction    Crust 2D    Algorithm



Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

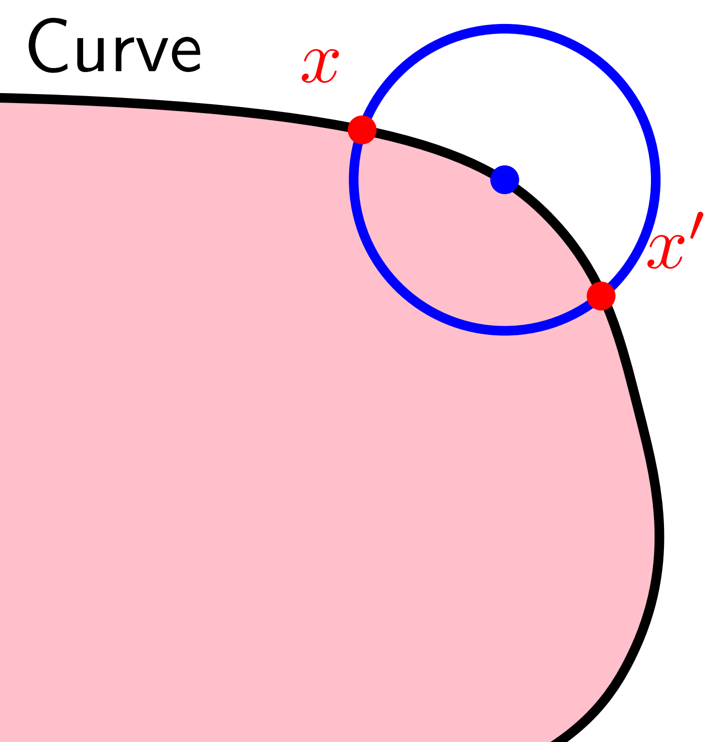
# Reconstruction



Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

$x$, $x'$ two neighboring points on Curve

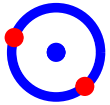
Circle thru $x$ and $x'$ centered on Curve

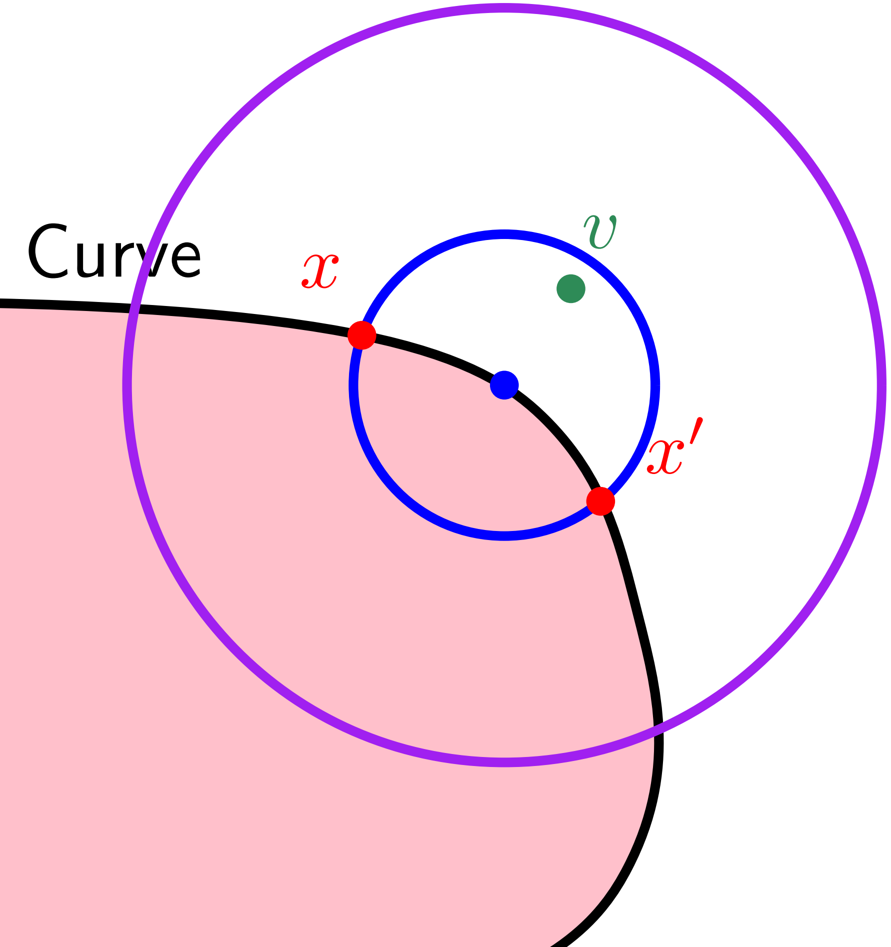


Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

$x$, $x'$ two neighboring points on Curve

Circle thru $x$ and $x'$ centered on Curve

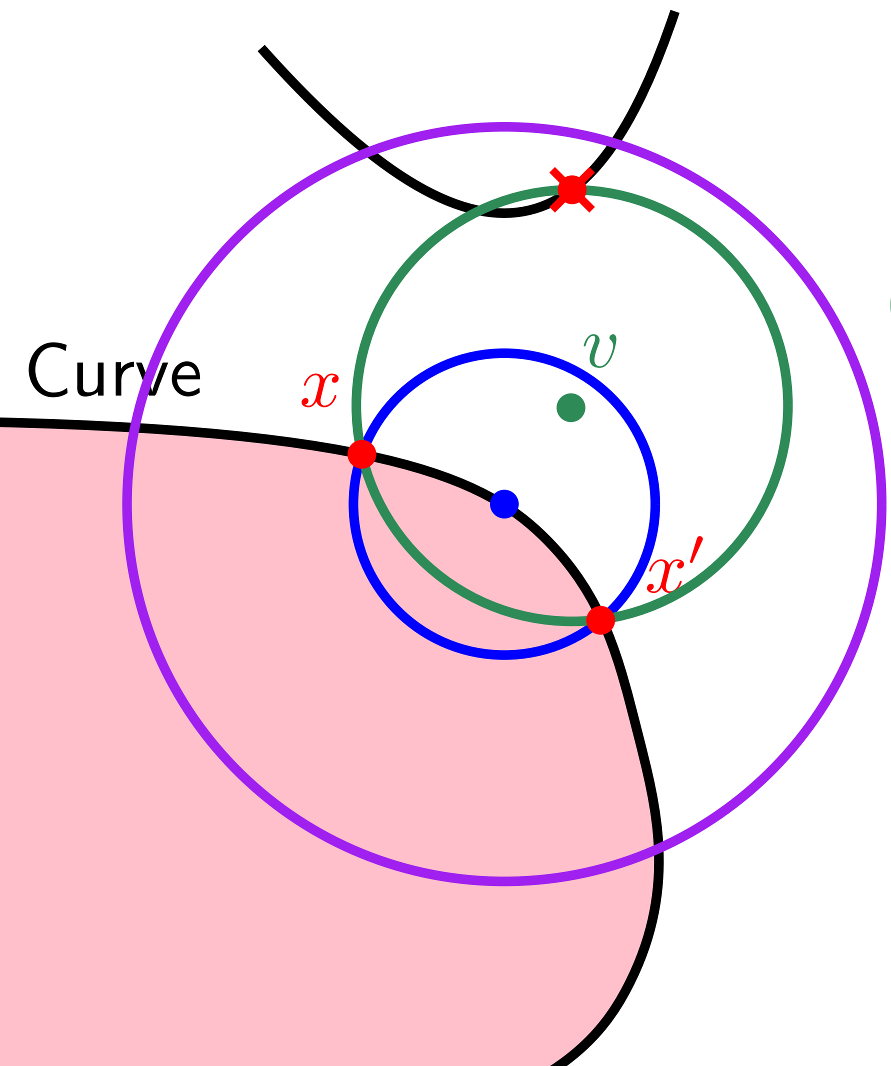
By contradiction assume $v \in$ 



# Reconstruction

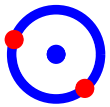


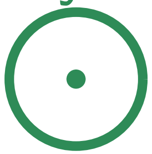
Theorem: $0.4$ sample $\Rightarrow$ wanted result $\subset$ crust




$x$, $x'$ two neighboring points on Curve

Circle thru $x$ and $x'$ centered on Curve

By contradiction assume $v \in$ 

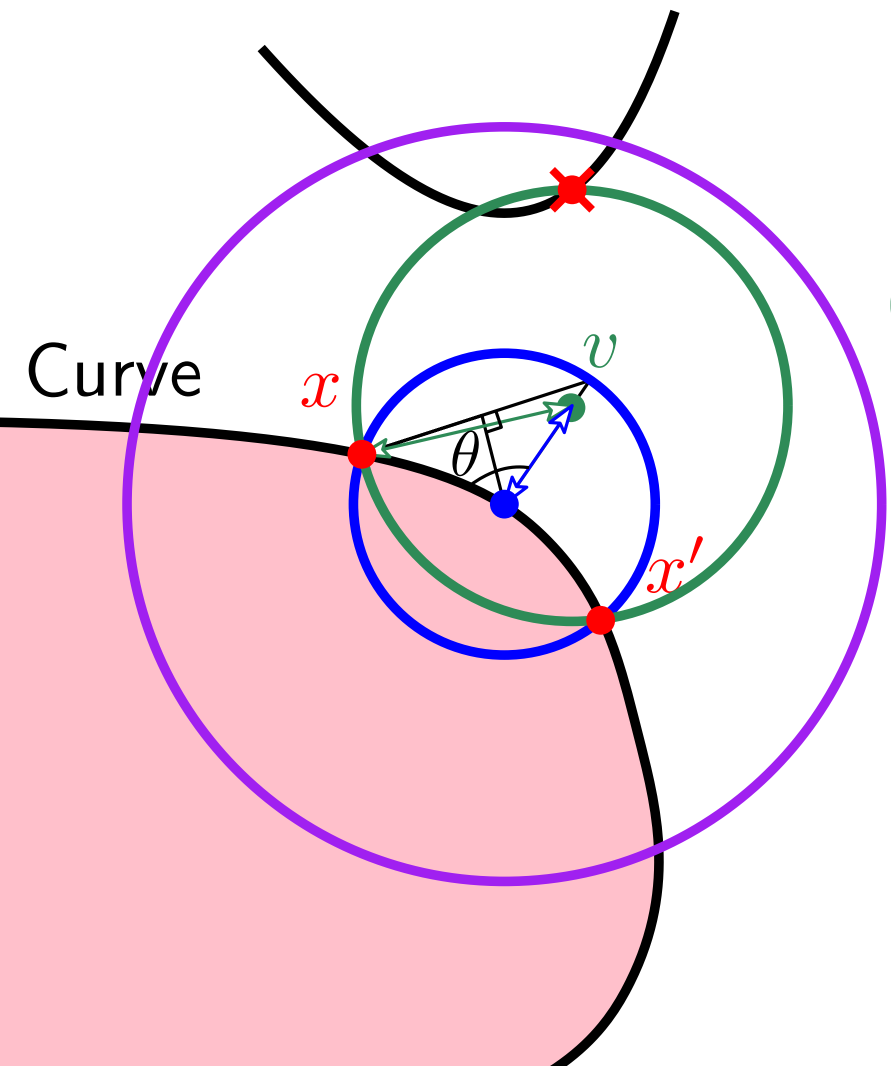
 intersects another cc of curve

(by Lemma)

# Reconstruction



Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust




$x$, $x'$ two neighboring points on Curve

Circle thru $x$ and $x'$ centered on Curve

By contradiction assume $v \in$

intersects another cc of curve

(by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

# Reconstruction



**Theorem:** $0.4$ sample $\Rightarrow$ wanted result $\subset$ crust

$x$, $x'$ two neighboring points on Curve

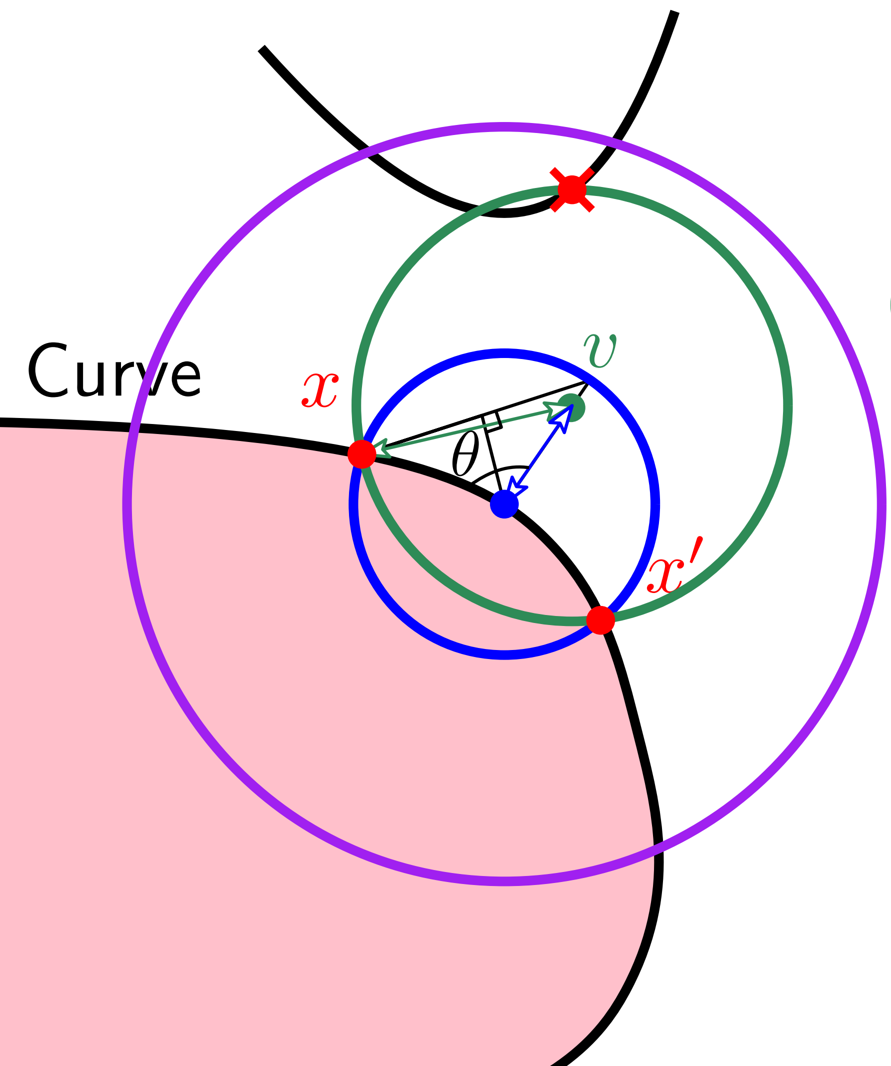
Circle thru $x$ and $x'$ centered on Curve

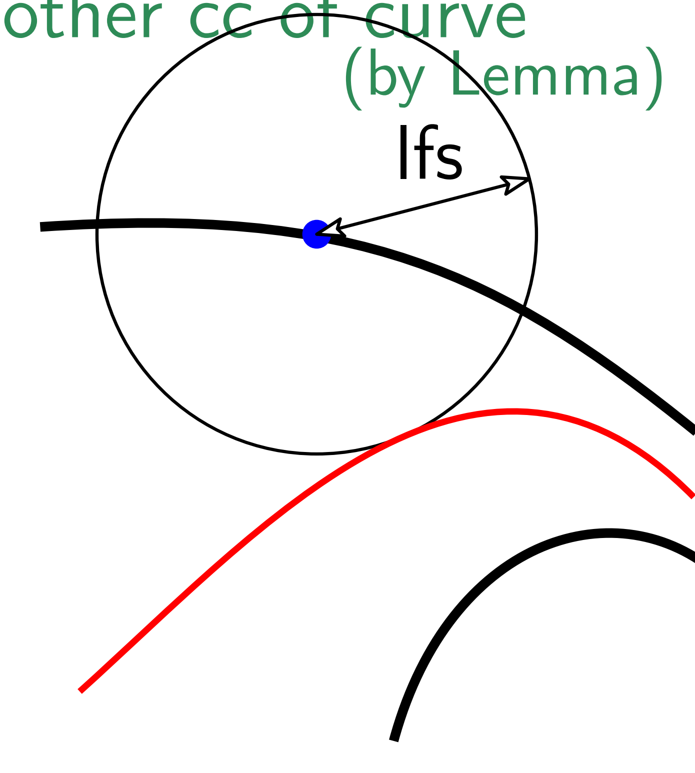
By contradiction assume $v \in$

intersects another cc of curve (by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq$



# Reconstruction



Theorem: 0.4 sample ⇒ wanted result ⊂ crust

$x$, $x'$ two neighboring points on Curve

Circle thru $x$ and $x'$ centered on Curve

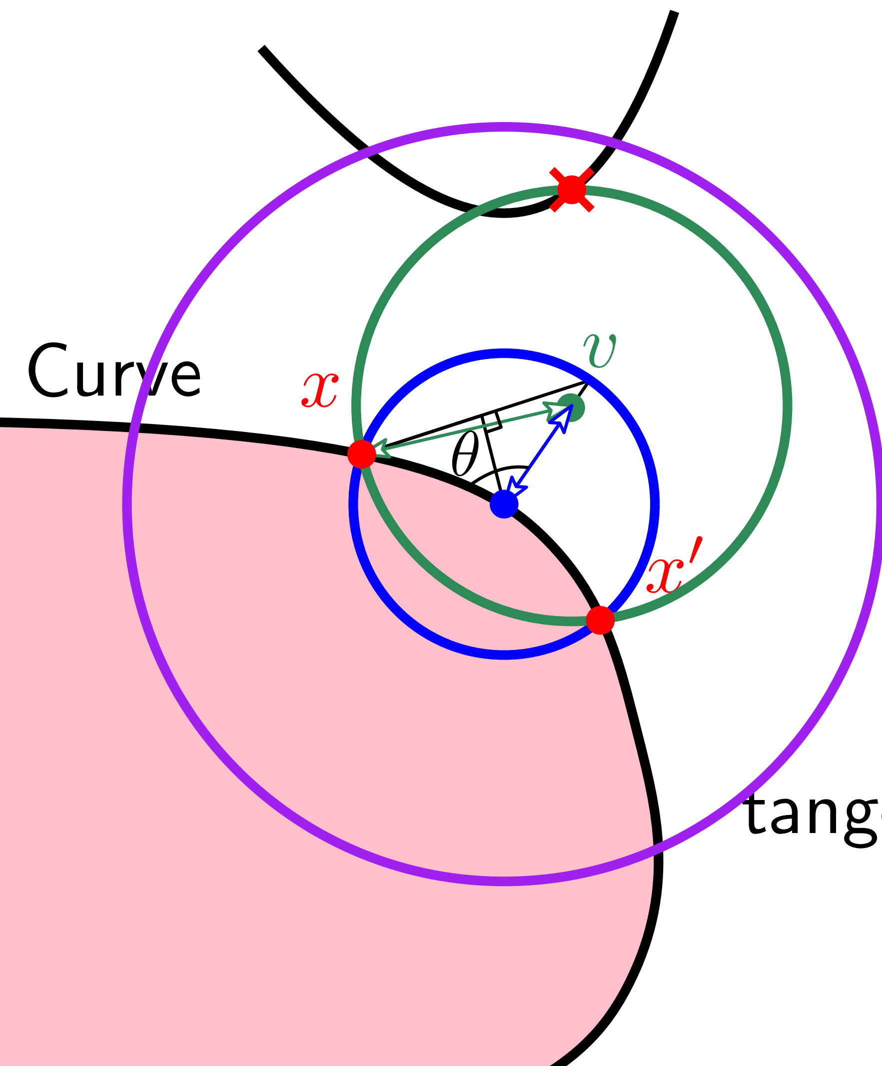
By contradiction assume $v \in$

intersects another cc of curve

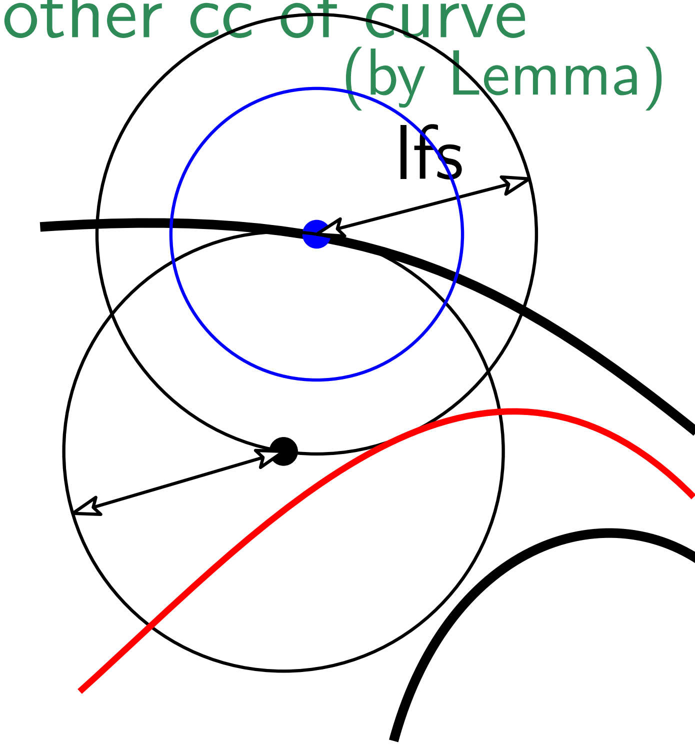
(by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq$




tangent disk is empty



# Reconstruction



Theorem: $0.4$ sample $\Rightarrow$ wanted result $\subset$ crust

$x$, $x'$ two neighboring points on Curve

Circle thru $x$ and $x'$ centered on Curve

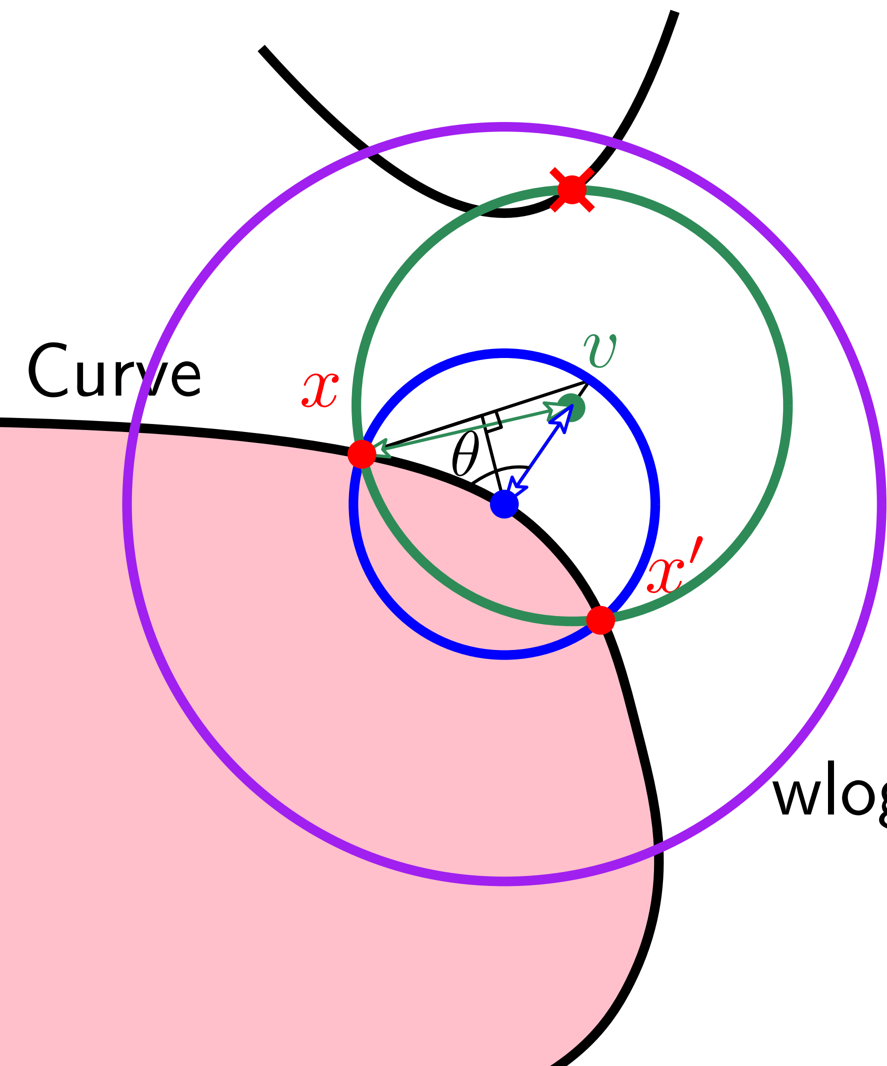
By contradiction assume $v \in$

intersects another cc of curve

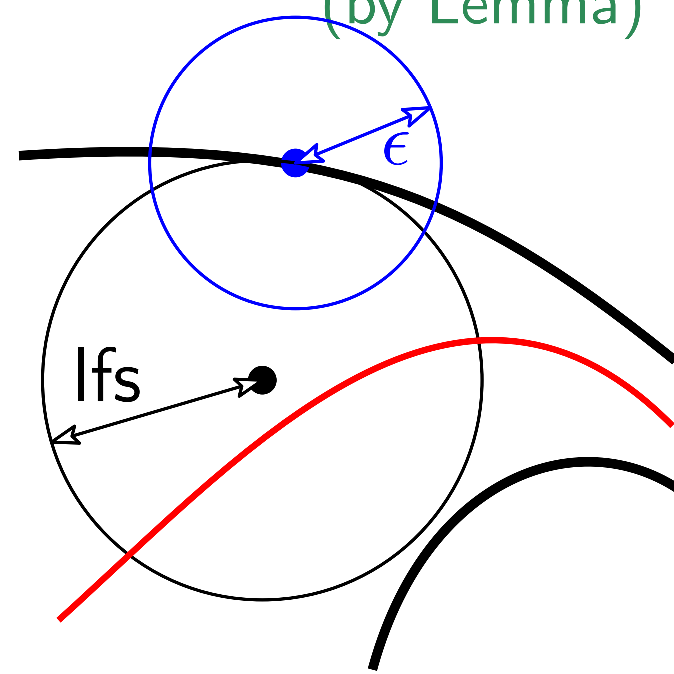
(by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq$



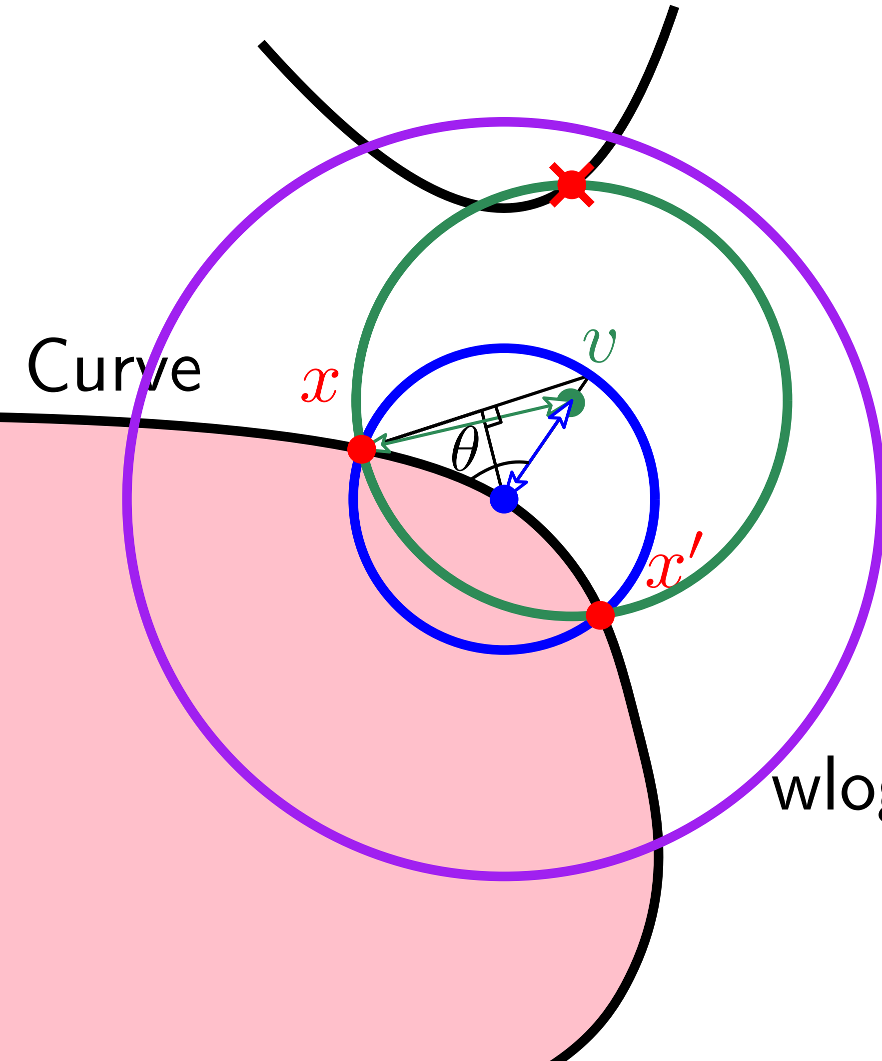

wlog lfs=1 and $r \leq \epsilon$



# Reconstruction



Theorem: $0.4$ sample $\Rightarrow$ wanted result $\subset$ crust




$x$, $x'$ two neighboring points on Curve

Circle thru $x$ and $x'$ centered on Curve

By contradiction assume $v \in$

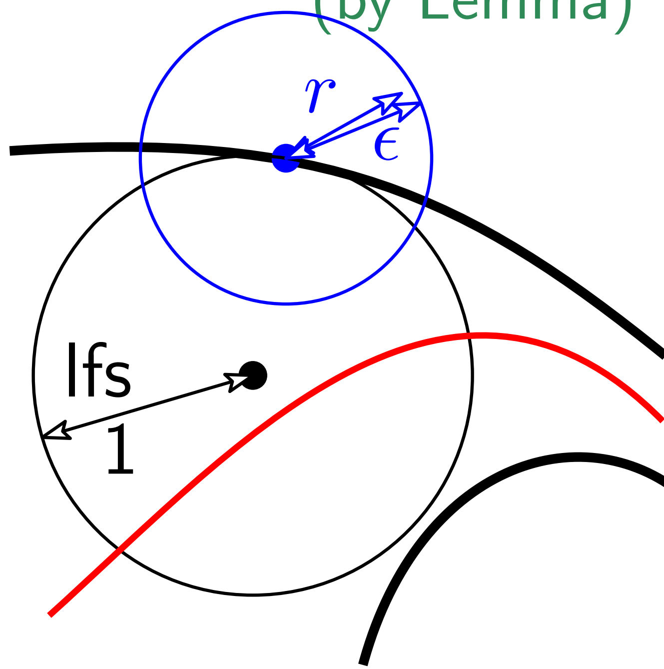
intersects another cc of curve

(by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq$

wlog lfs=1 and $r \leq \epsilon$



# Reconstruction

Crust 2D 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

$x$, $x'$ two neighboring points on Curve

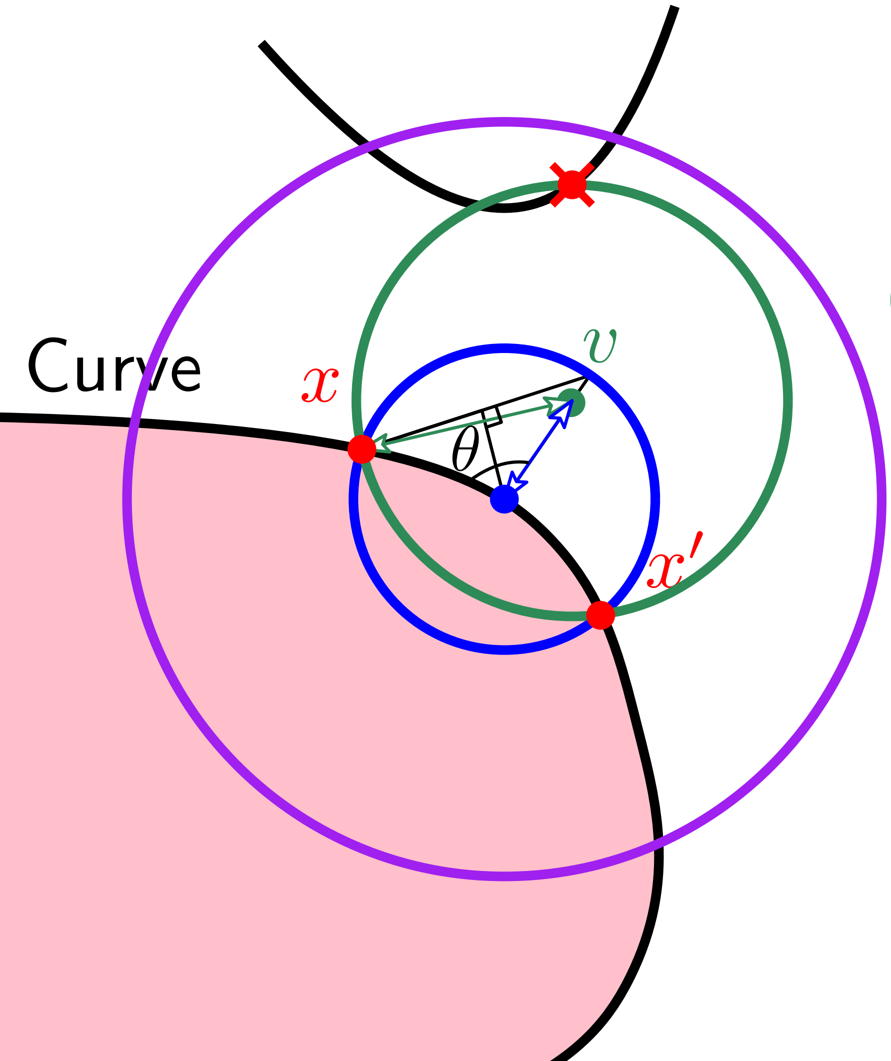
Circle thru $x$ and $x'$ centered on Curve

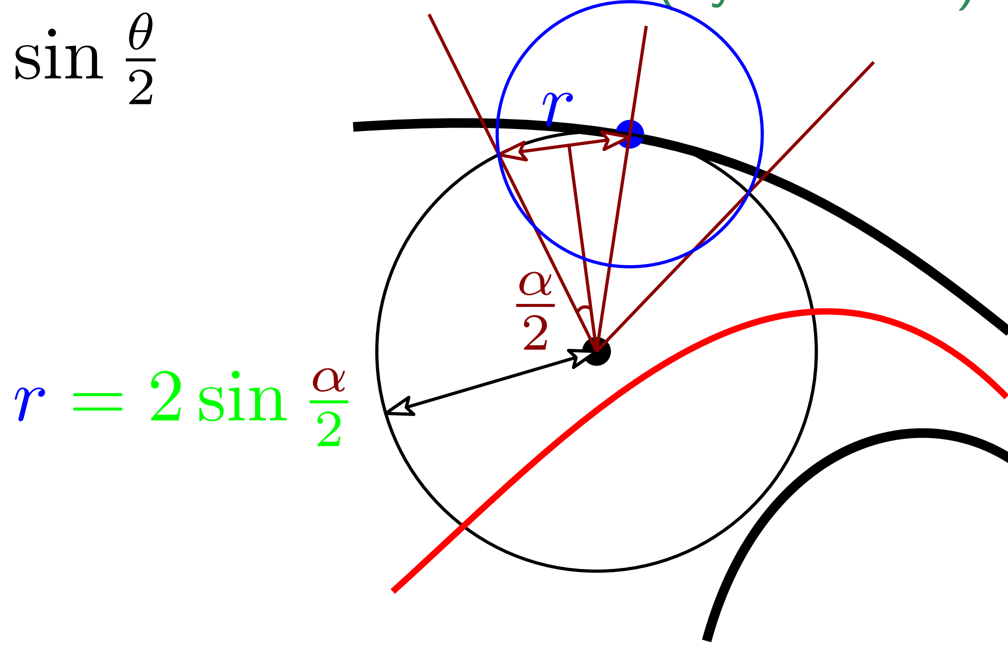
By contradiction assume $v \in$ ⊙

⊙ intersects another cc of curve (by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq$

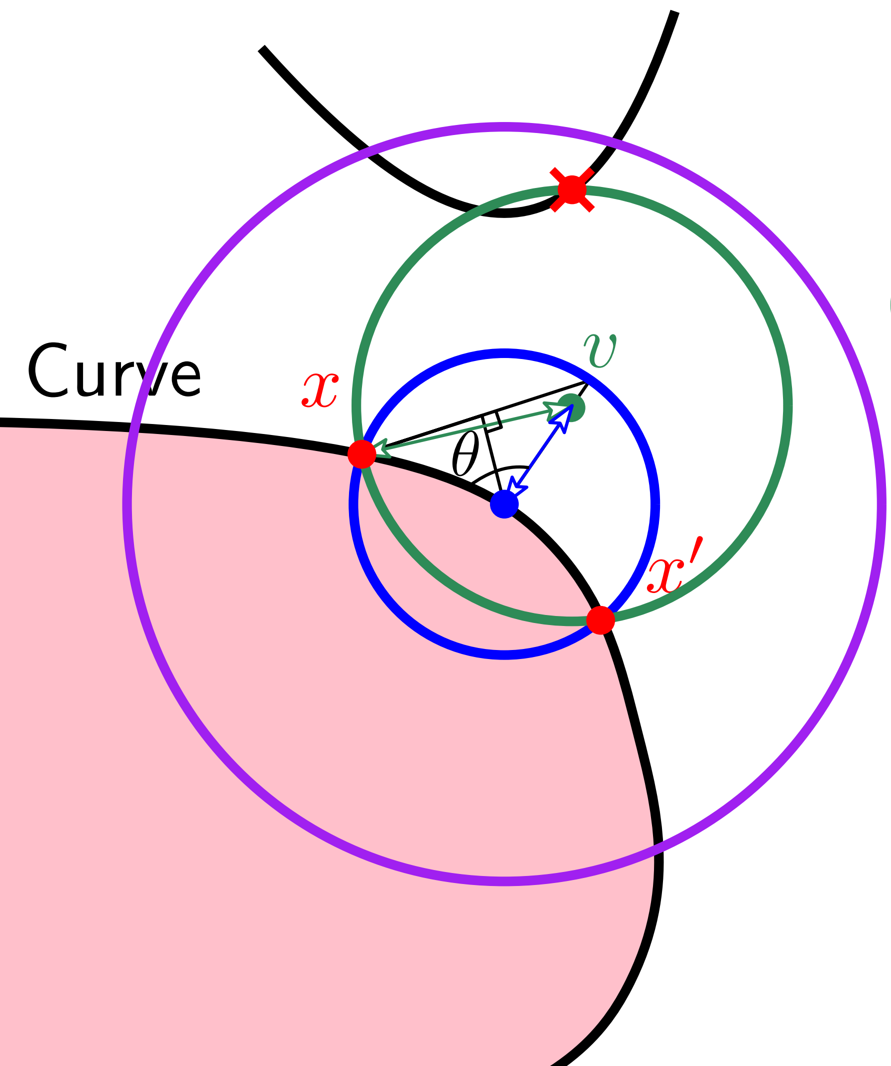


# Reconstruction



Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust




$x$, $x'$ two neighboring points on Curve

Circle thru $x$ and $x'$ centered on Curve

By contradiction assume $v \in$

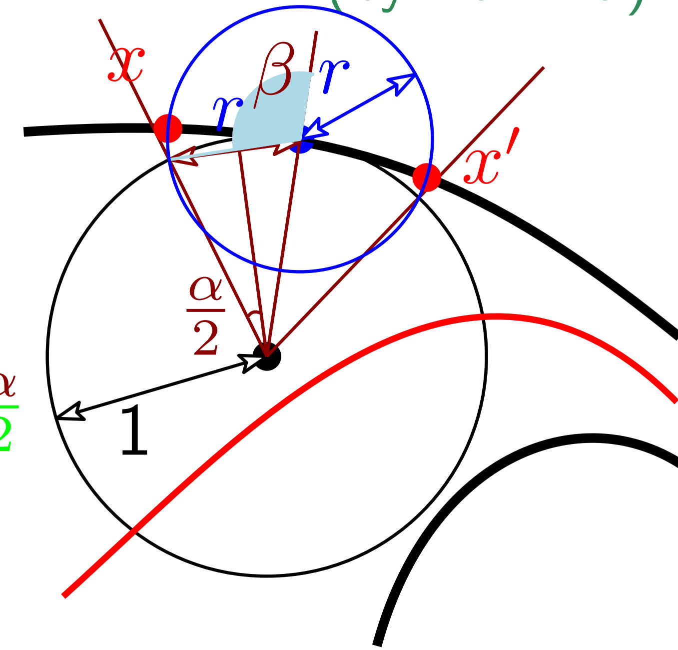
intersects another cc of curve

(by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq \beta = \pi - \frac{\pi - \alpha}{2}$

$r = 2 \sin \frac{\alpha}{2}$



# Reconstruction



Theorem: 0.4 sample ⇒ wanted result ⊂ crust

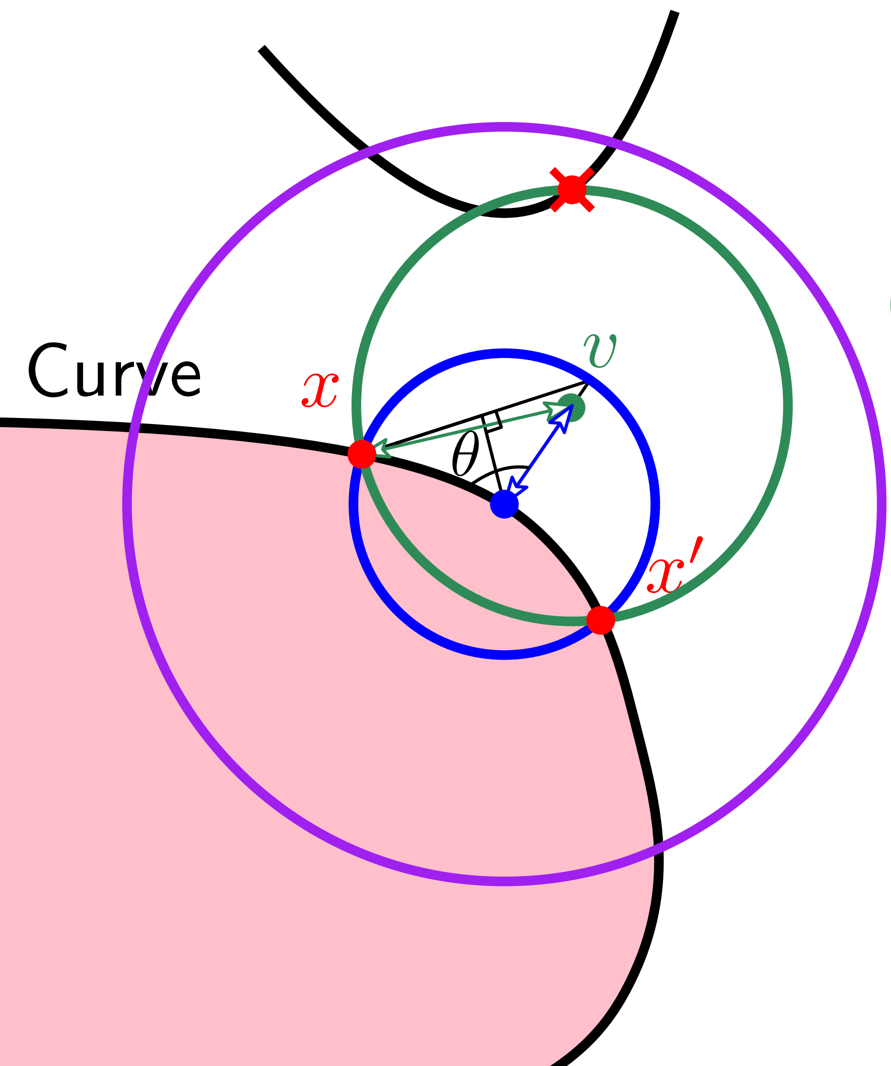
$x$, $x'$ two neighboring points on Curve

Circle thru $x$ and $x'$ centered on Curve

By contradiction assume $v \in$

intersects another cc of curve

(by Lemma)

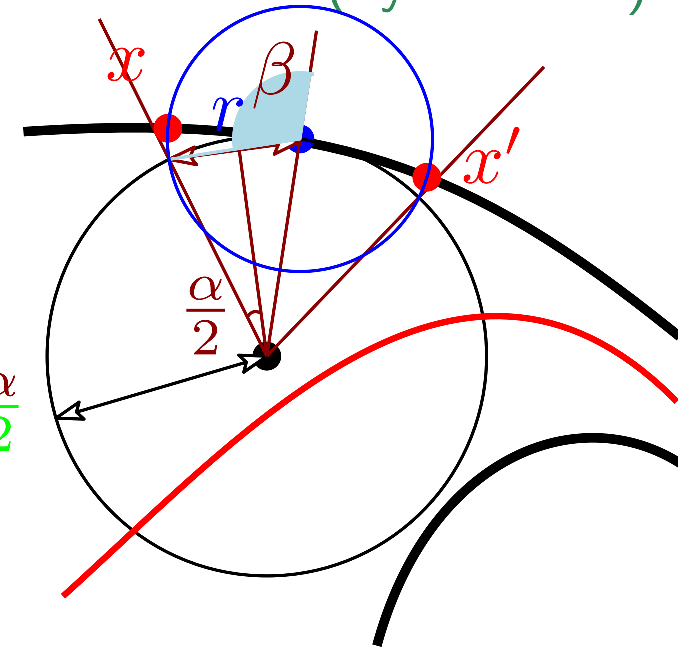



$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq \beta = \pi - \frac{\pi - \alpha}{2}$

$\leq \frac{\pi}{2} + \text{arcsin}\frac{r}{2}$

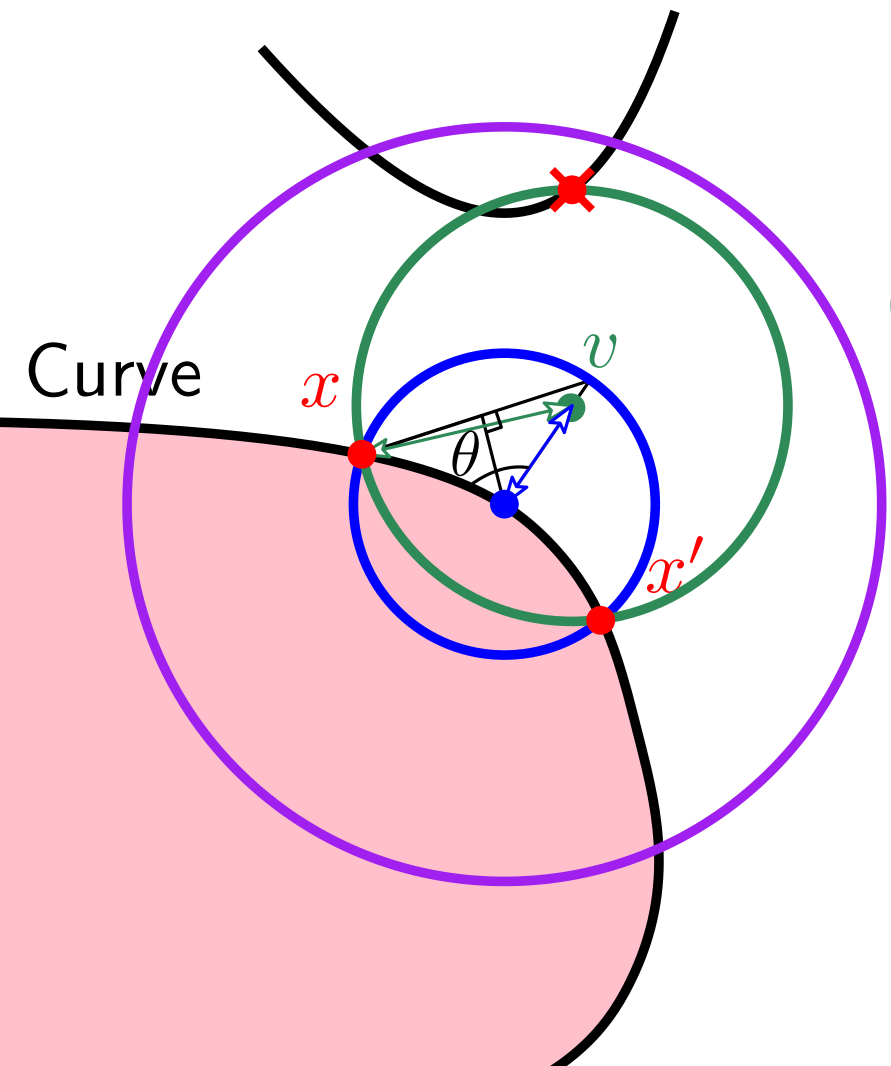
$r = 2 \sin \frac{\alpha}{2}$



# Reconstruction



Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust




$x$, $x'$ two neighboring points on Curve

Circle thru $x$ and $x'$ centered on Curve

By contradiction assume $v \in$

intersects another cc of curve

(by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

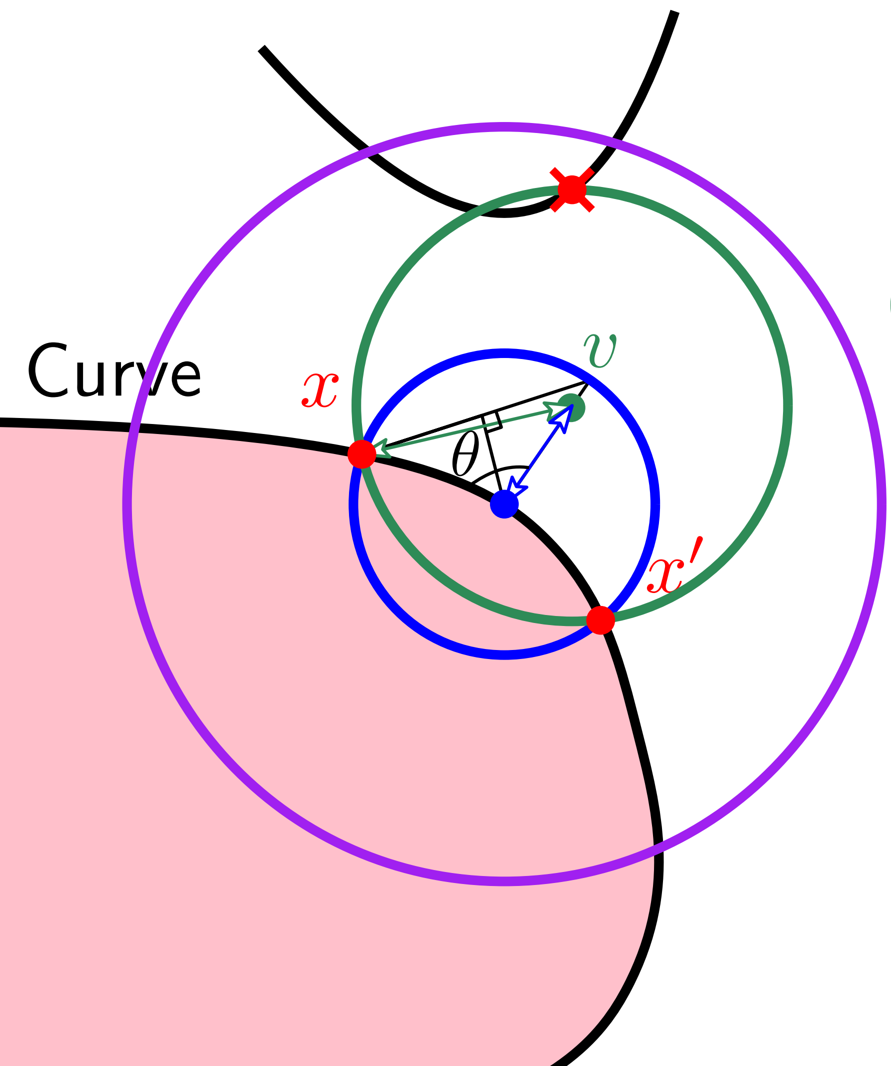
$\theta \leq \frac{\pi}{2} + \text{arcsin} \frac{r}{2}$

$\| \bullet \times \| \leq \| \bullet \ \bullet \| + \| \bullet \times \|$

# Reconstruction

## Crust 2D

Theorem: 0.4 sample ⇒ wanted result ⊂ crust




$x$, $x'$ two neighboring points on Curve

Circle thru $x$ and $x'$ centered on Curve

By contradiction assume $v \in$ $\odot$

$\odot$ intersects another cc of curve (by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

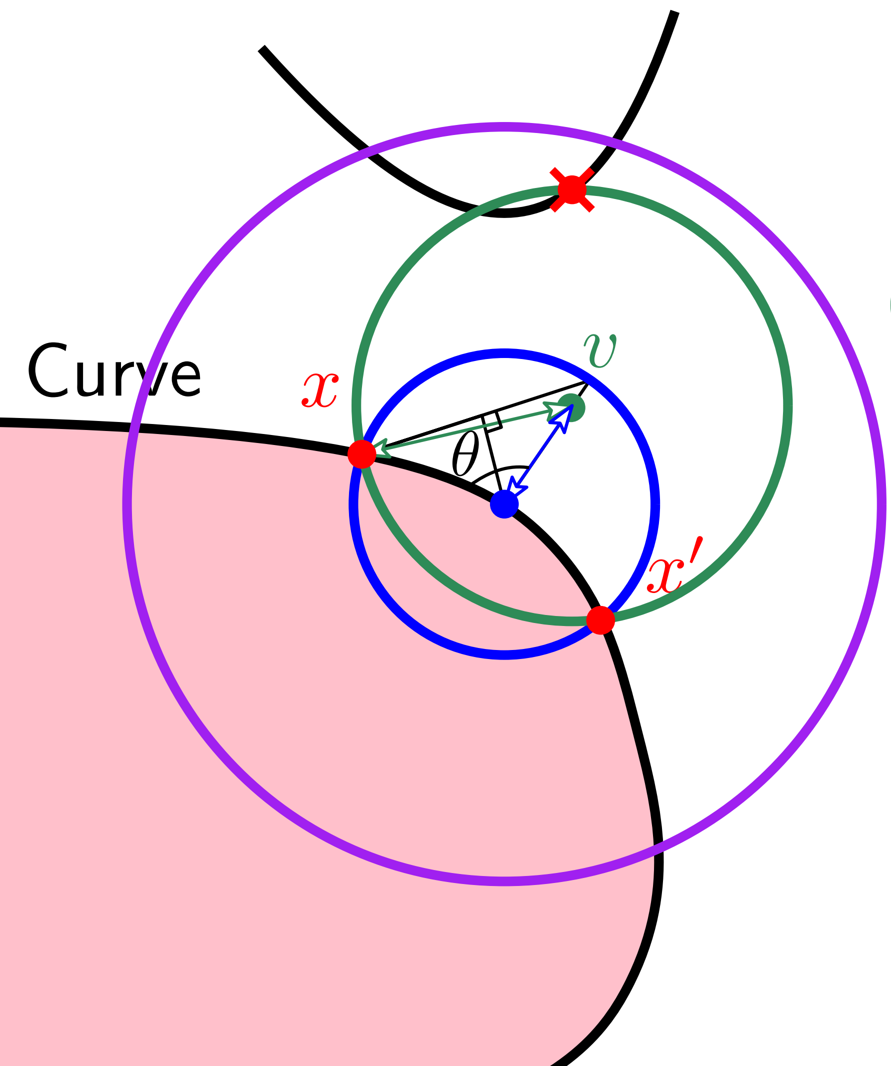
$\theta \leq \frac{\pi}{2} + \text{arcsin} \frac{r}{2}$

$$\| \bullet \times \| \leq \| \bullet \ \bullet \| + \| \bullet \times \|$$

$$\leq r + 2r \sin \left( \frac{\pi}{4} + \frac{1}{2} \arcsin \frac{r}{2} \right)$$

# Reconstruction

Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust




$x$, $x'$ two neighboring points on Curve

Circle thru $x$ and $x'$ centered on Curve

By contradiction assume $v \in$

intersects another cc of curve

(by Lemma)

$R \leq 2r \sin \frac{\theta}{2}$

$\theta \leq \frac{\pi}{2} + \arcsin \frac{r}{2}$

$\|\bullet \times\| \leq \|\bullet\ \bullet\| + \|\bullet \times\|$

$\leq r + 2r \sin\left(\frac{\pi}{4} + \frac{1}{2}\arcsin \frac{r}{2}\right)$

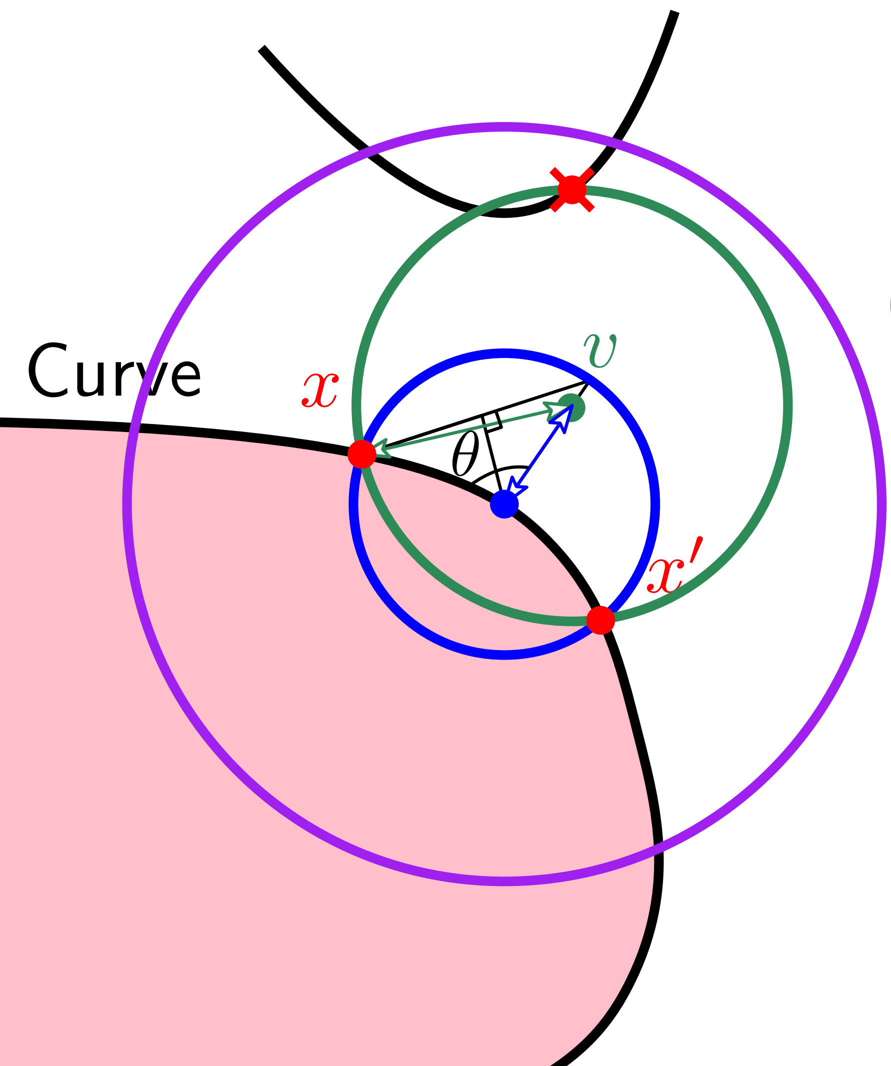
if $\|\bullet \times\| \leq$ lfs$= 1$ contradiction is reached

# Reconstruction

## Crust 2D

0.4 sample $\Rightarrow$ wanted result $\subset$ crust

Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust




$x$, $x'$ two nei

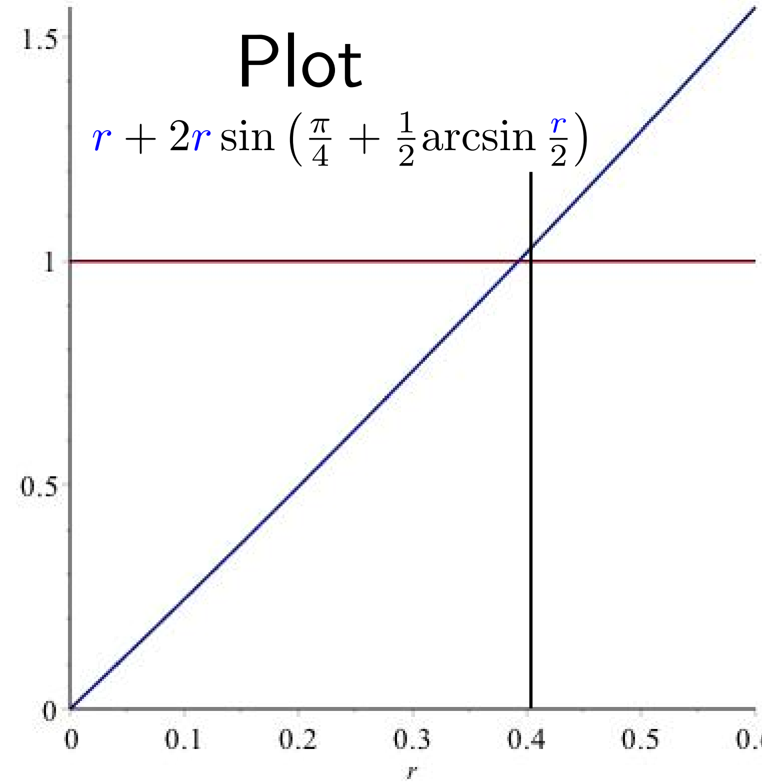
Circle thru $x$

By contradict

intersect

$R \leq 2r\sin\frac{\theta}{2}$

$\theta \leq \frac{\pi}{2} + \text{arcs}$




$\|\bullet\times\| \leq \|\bullet\ \bullet\| + \|\bullet\times\|$

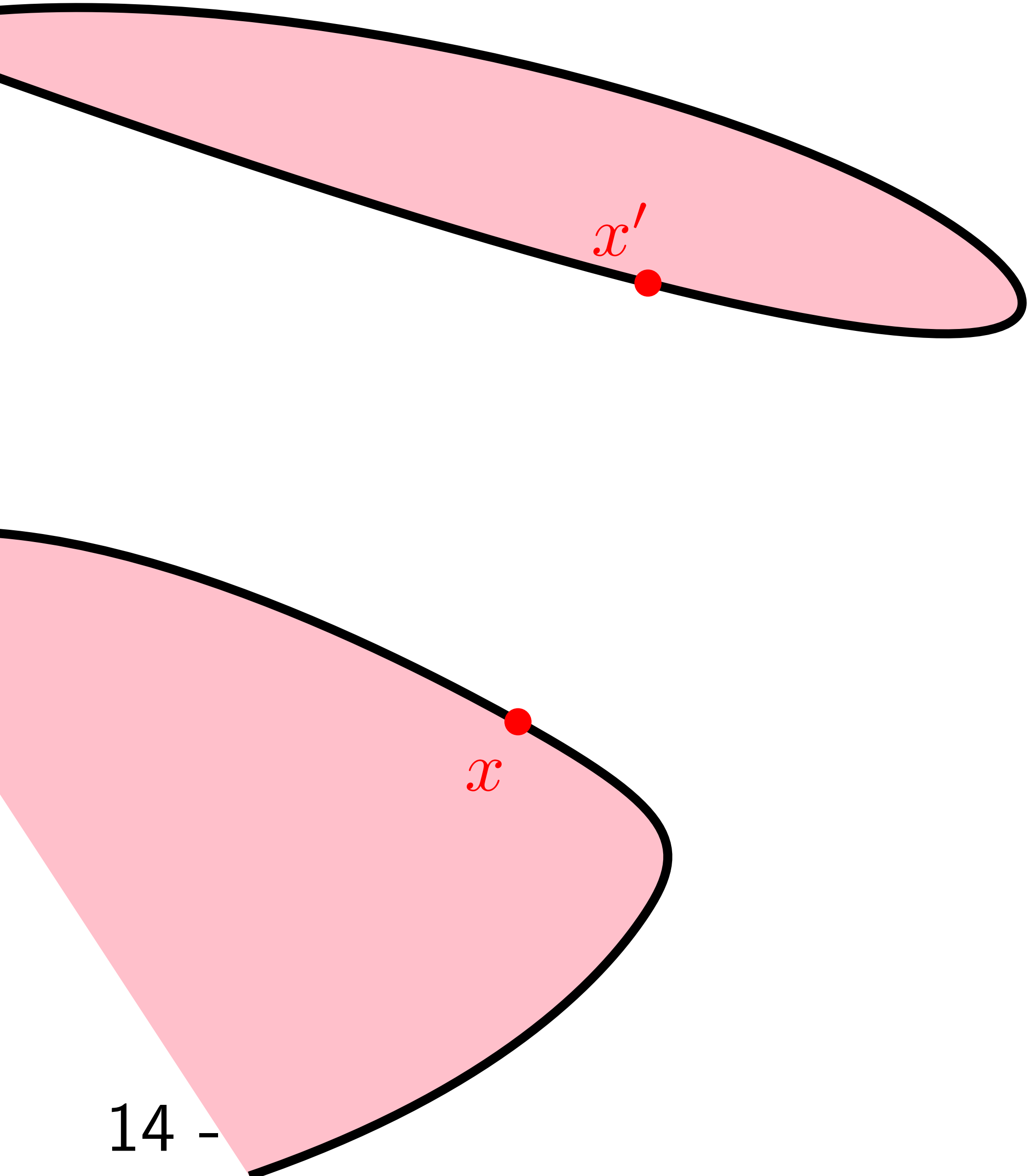
$\leq r + 2r\sin\left(\frac{\pi}{4} + \frac{1}{2}\arcsin\frac{r}{2}\right)$

if $\|\bullet\times\| \leq$lfs$=1$ contradiction is reached

Theorem: 0.4 sample $\Rightarrow$ wanted result $\subset$ crust

Theorem: 0.25 sample $\Rightarrow$ crust $\subset$ wanted result

Theorem: 0.25 sample ⇒ crust ⊂ wanted result

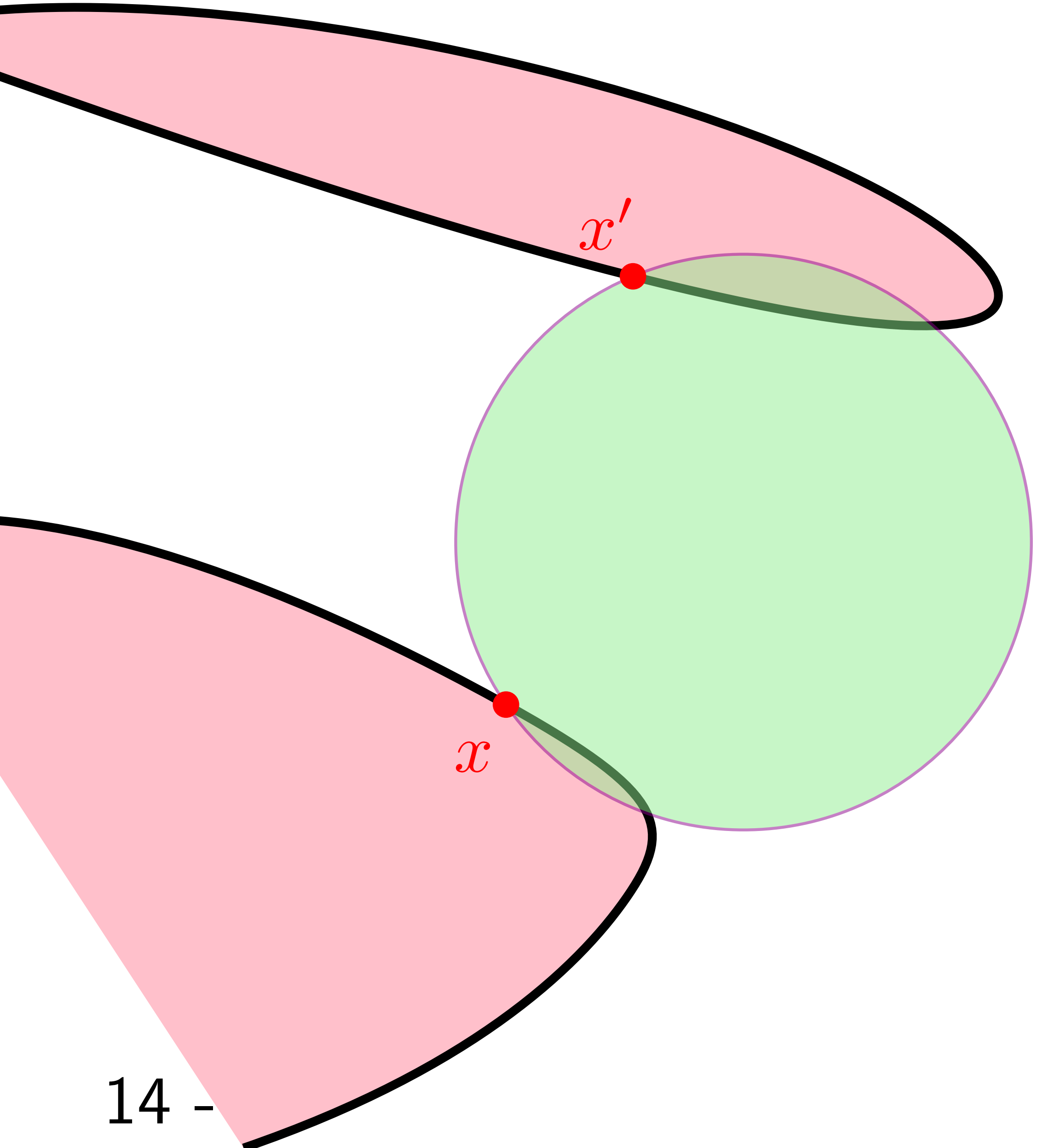


# Reconstruction



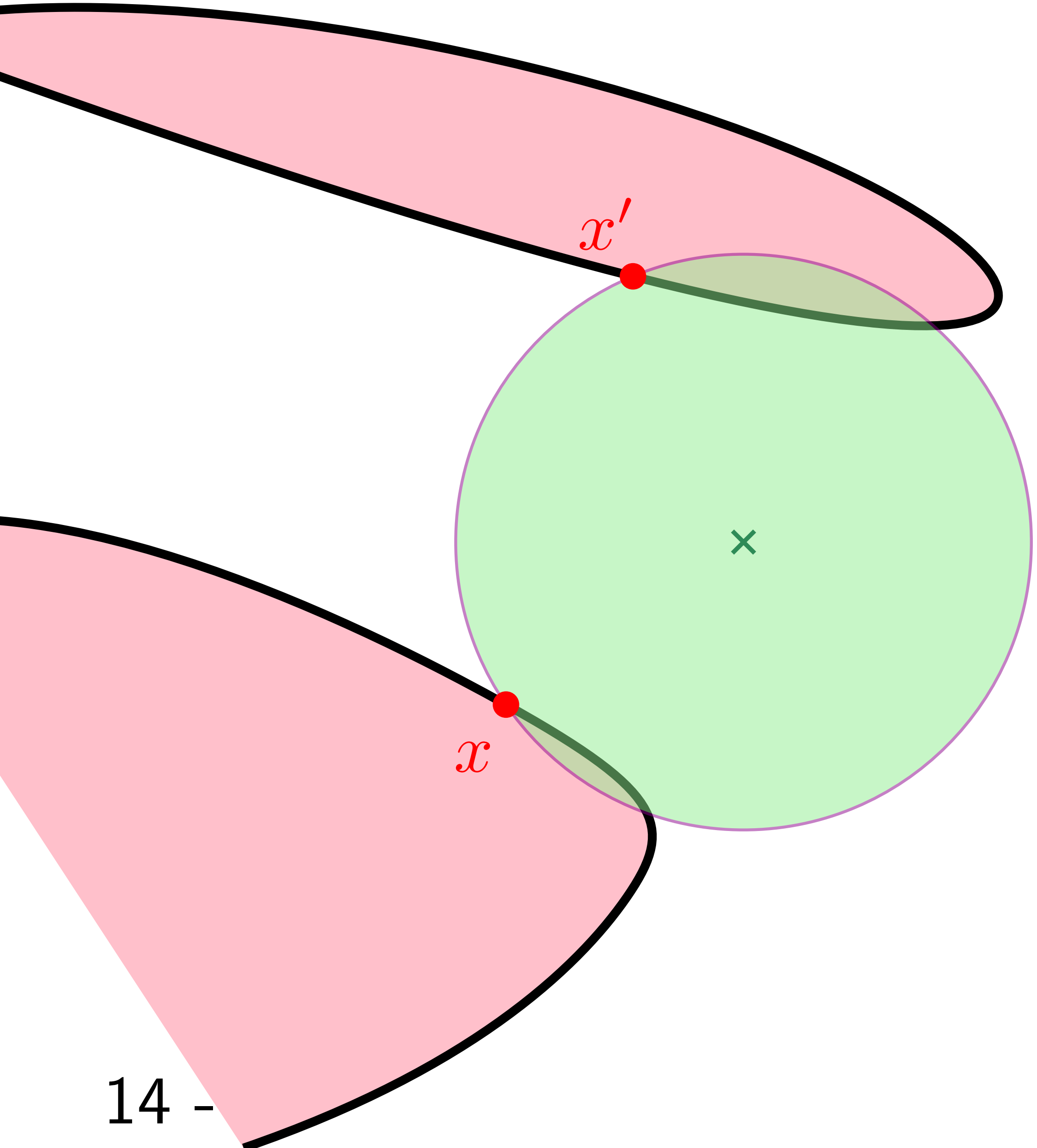
Theorem: 0.25 sample $\Rightarrow$ crust $\subset$ wanted result

Assume empty circle



# Reconstruction

Theorem: 0.25 sample ⇒ crust ⊂ wanted result



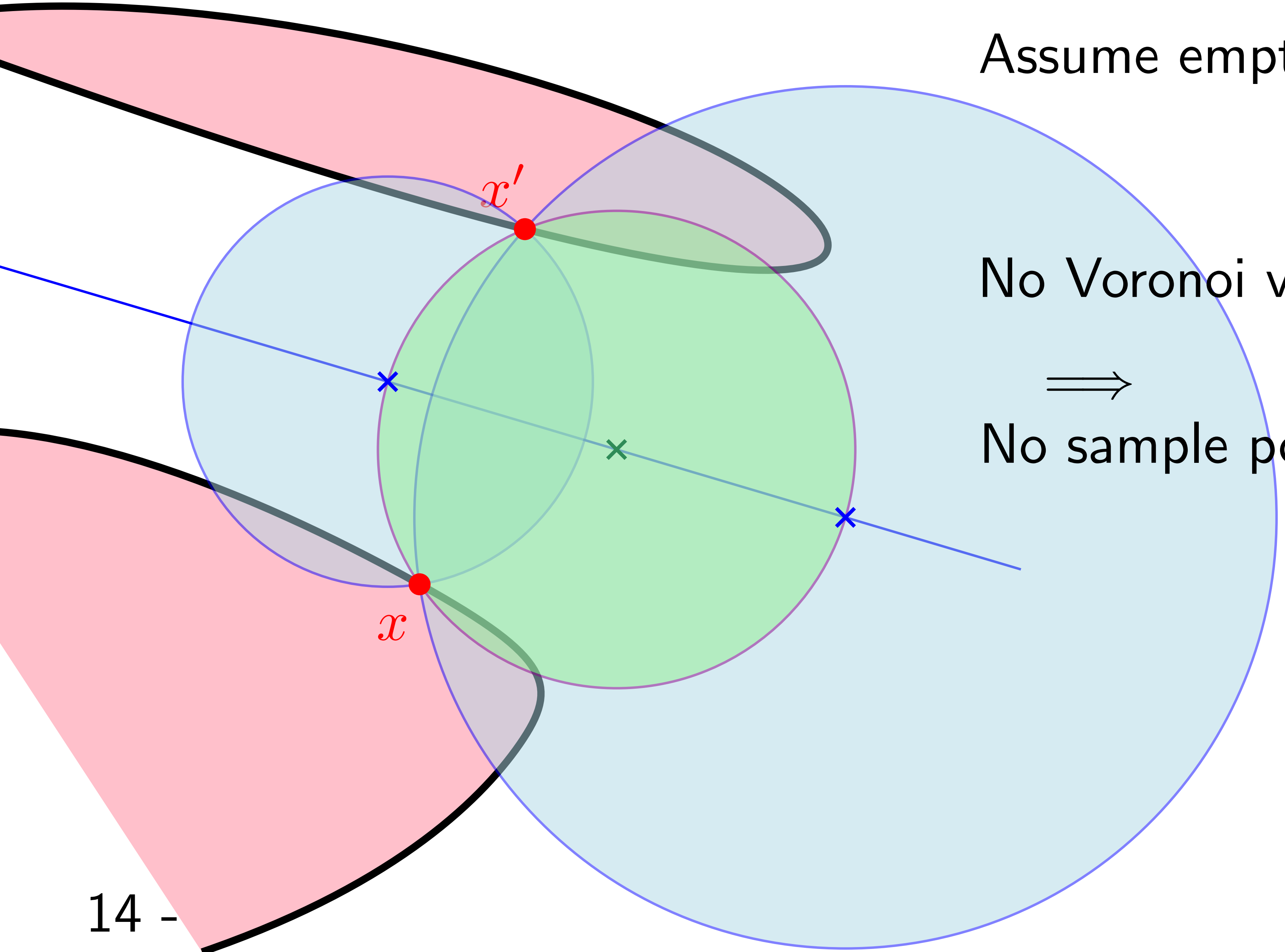

Assume empty circle

No Voronoi vertices there

$\Longrightarrow$

# Reconstruction

Theorem: $0.25$ sample $\Rightarrow$ crust $\subset$ wanted result




Assume empty circle

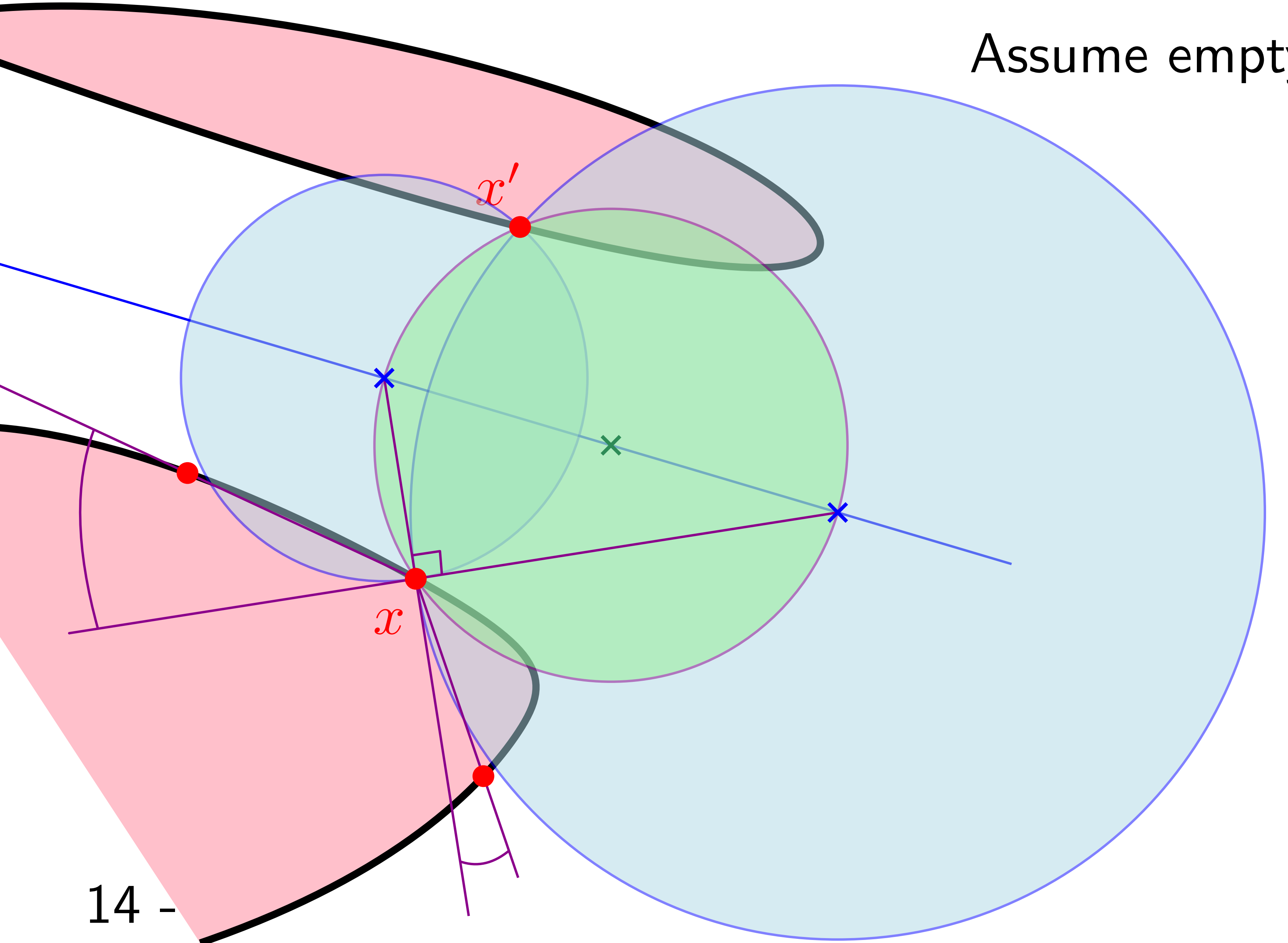
No Voronoi vertices there

$\Longrightarrow$

No sample points there

Theorem: 0.25 sample $\Rightarrow$ crust $\subset$ wanted result

Assume empty circle



# Reconstruction

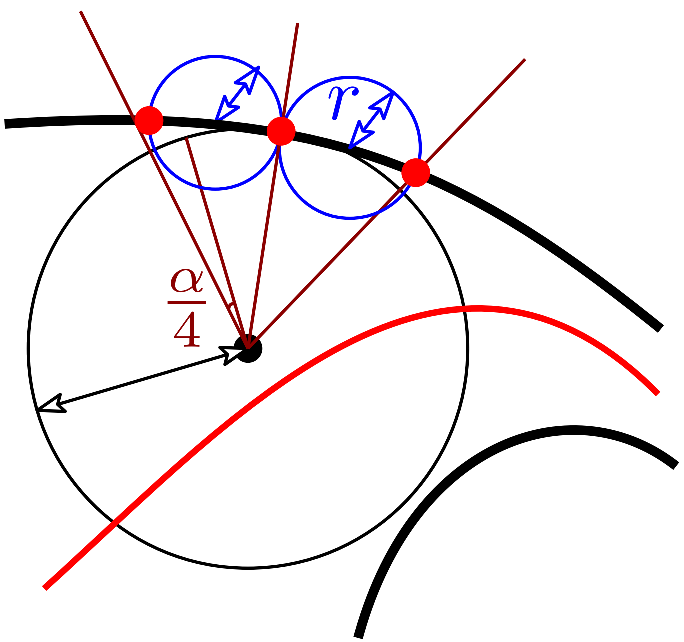
## Crust 2D

Theorem: 0.25 sample $\Rightarrow$ crust $\subset$ wanted result

$$\beta = \pi - \frac{\pi - \alpha}{2}$$

$$\leq \frac{\pi}{2} + 2\text{arcsin}\frac{r}{2}$$

$$r \simeq 2\sin\frac{\alpha}{4}$$



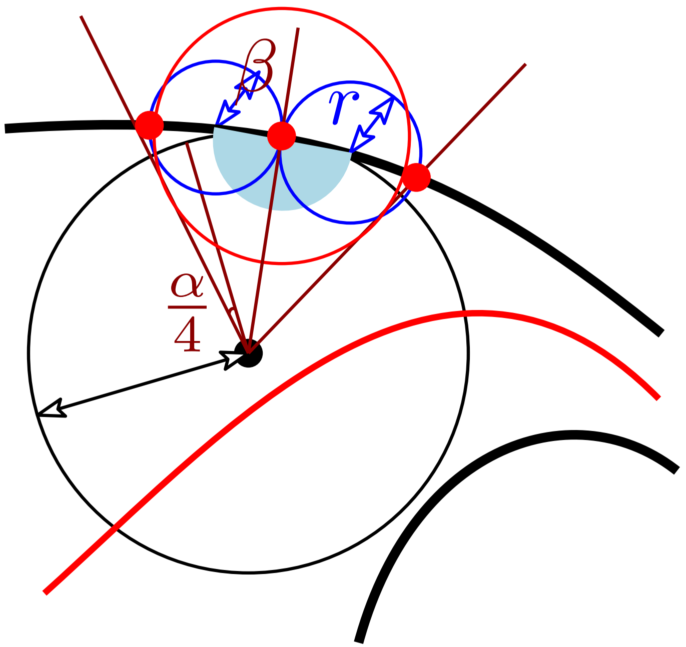
# Reconstruction

Theorem: 0.25 sample $\Rightarrow$ crust $\subset$ wanted result

$\beta = \pi - \frac{\pi - \alpha}{2}$

$\leq \frac{\pi}{2} + 2\text{arcsin}\frac{r}{2}$

$r \simeq 2\sin\frac{\alpha}{4}$



# Reconstruction

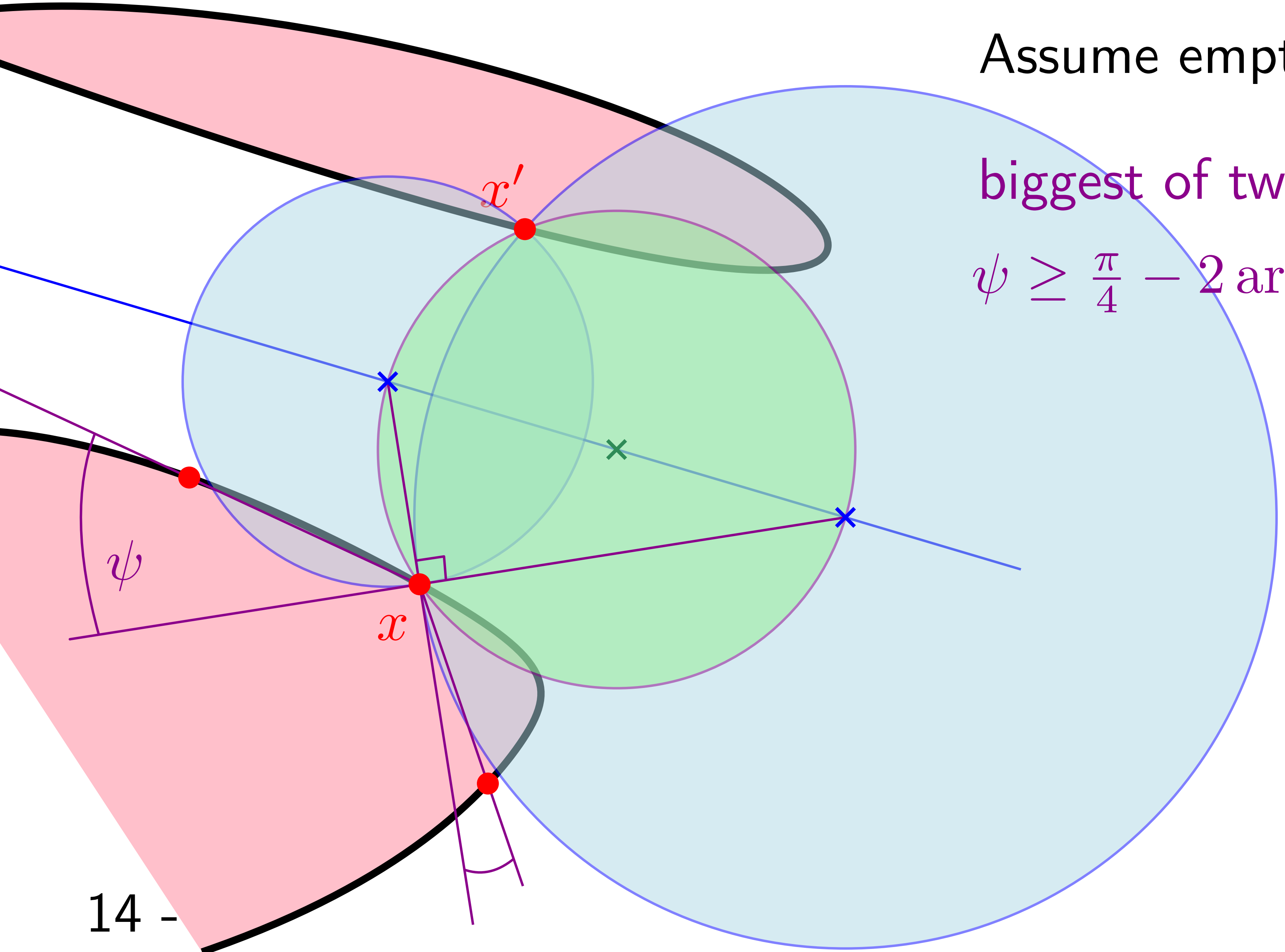


Theorem: 0.25 sample ⇒ crust ⊂ wanted result

Assume empty circle

biggest of two angles

$$\psi \geq \frac{\pi}{4} - 2\arcsin\frac{r}{2}$$



# Reconstruction



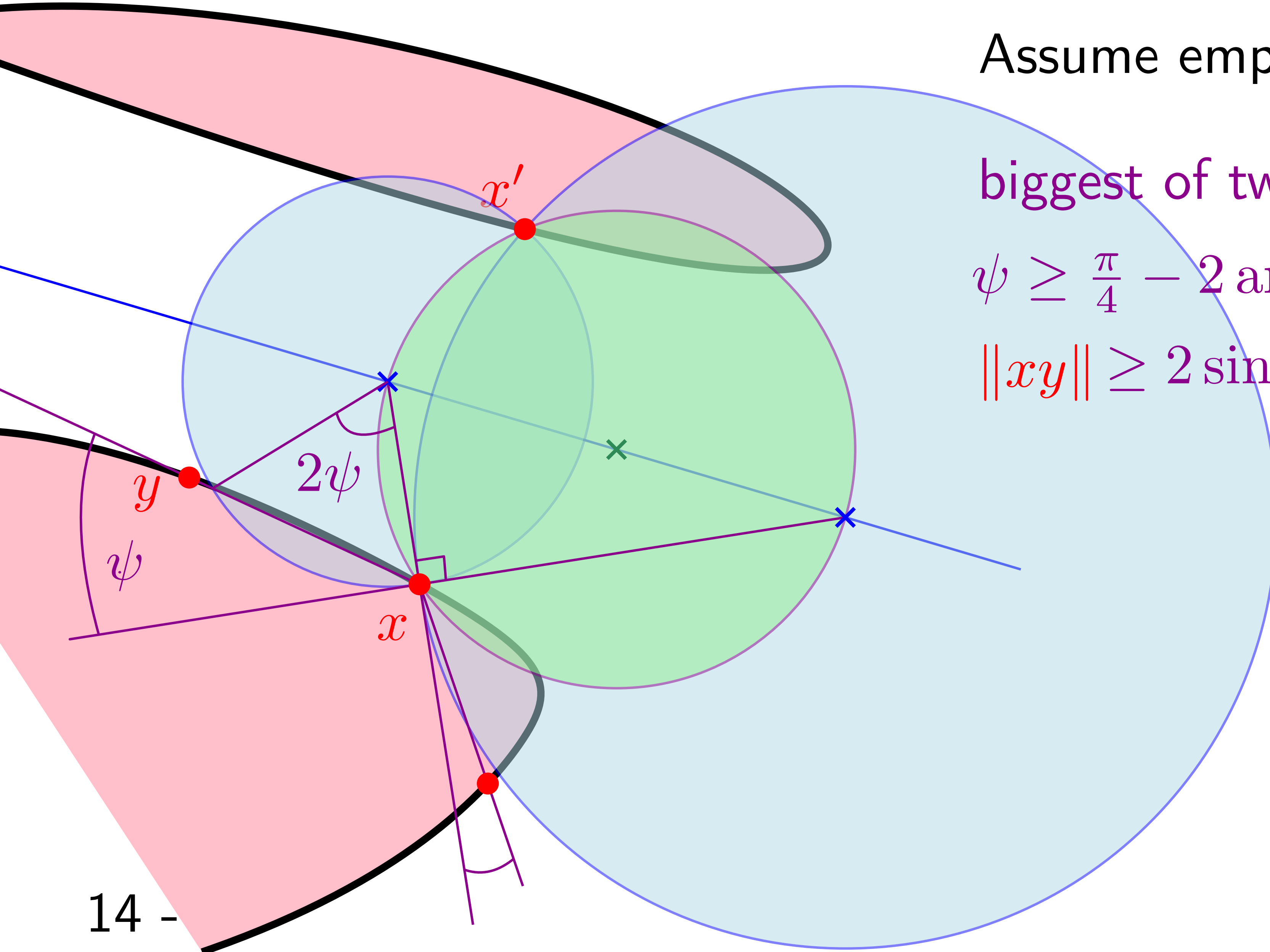
Theorem: 0.25 sample ⇒ crust ⊂ wanted result

Assume empty circle

biggest of two angles

$$\psi \geq \frac{\pi}{4} - 2\arcsin\frac{r}{2}$$

$$\|xy\| \geq 2\sin\psi$$



# Reconstruction

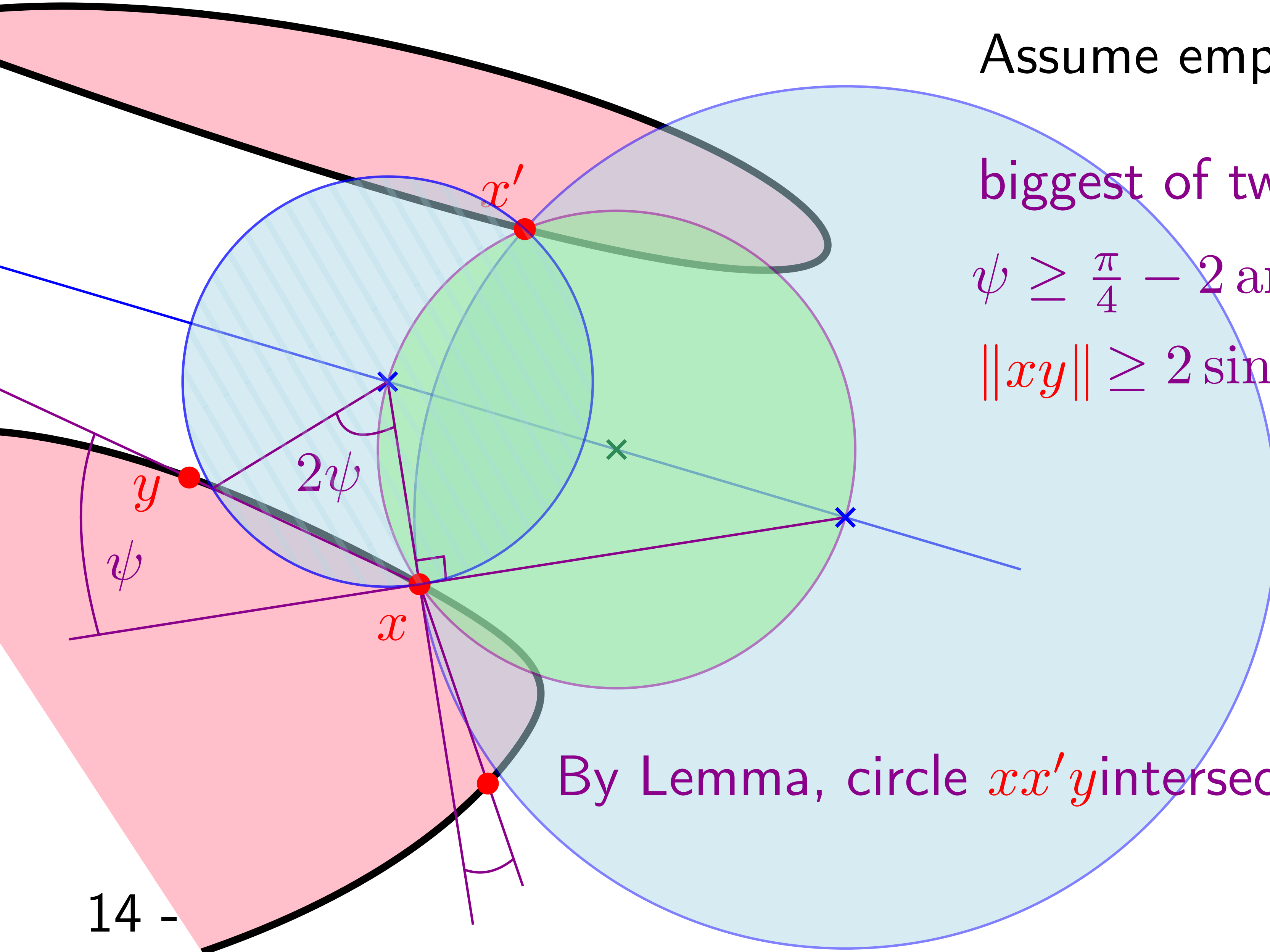
Theorem: 0.25 sample $\Rightarrow$ crust $\subset$ wanted result

Assume empty circle

biggest of two angles

$\psi \geq \frac{\pi}{4} - 2\arcsin\frac{r}{2}$

$\|xy\| \geq 2\sin\psi$

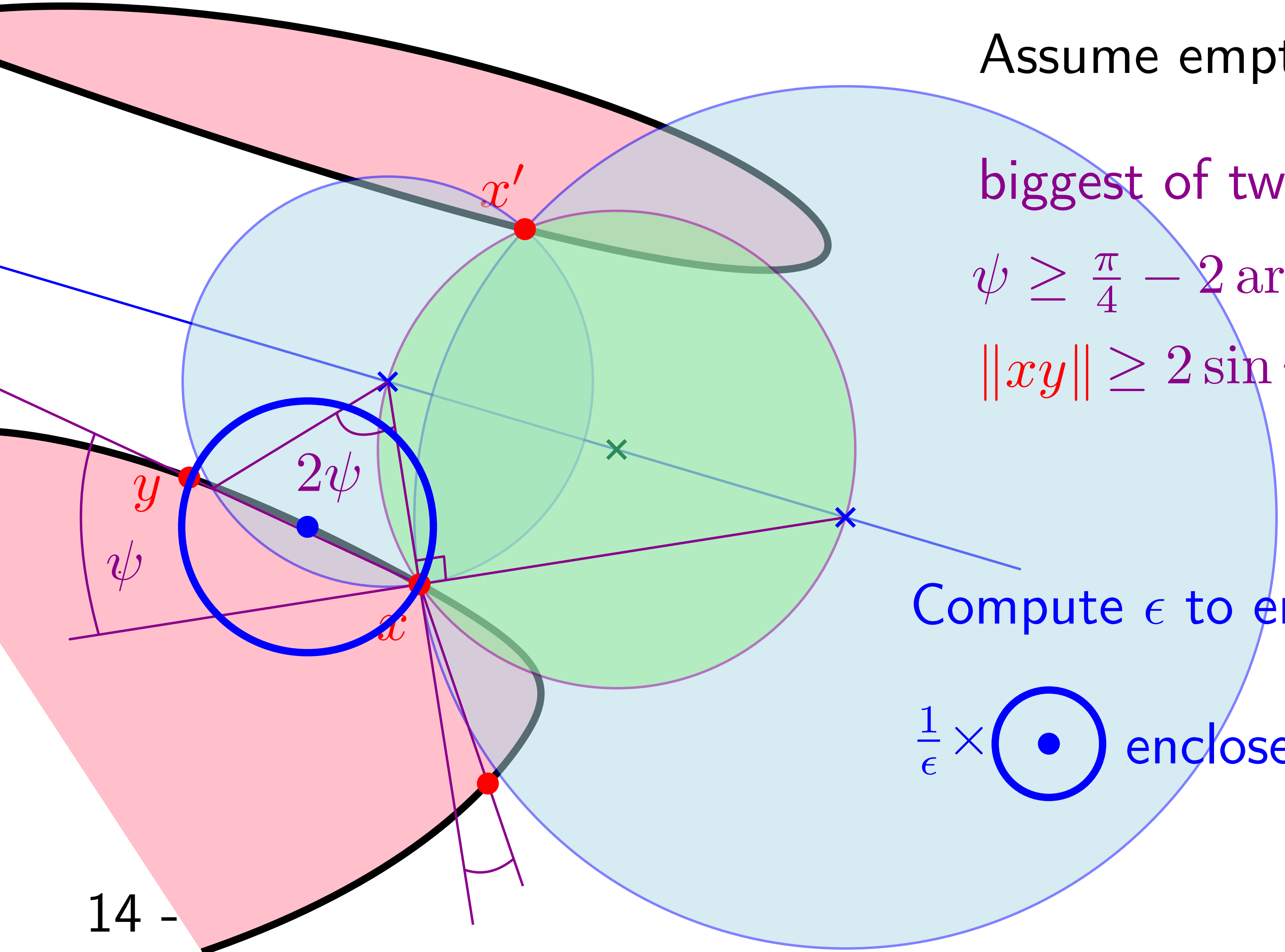



By Lemma, circle $xx'y$ intersects medial axis

# Reconstruction



Theorem: 0.25 sample ⇒ crust ⊂ wanted result




Assume empty circle

biggest of two angles

$\psi \geq \frac{\pi}{4} - 2\arcsin\frac{r}{2}$

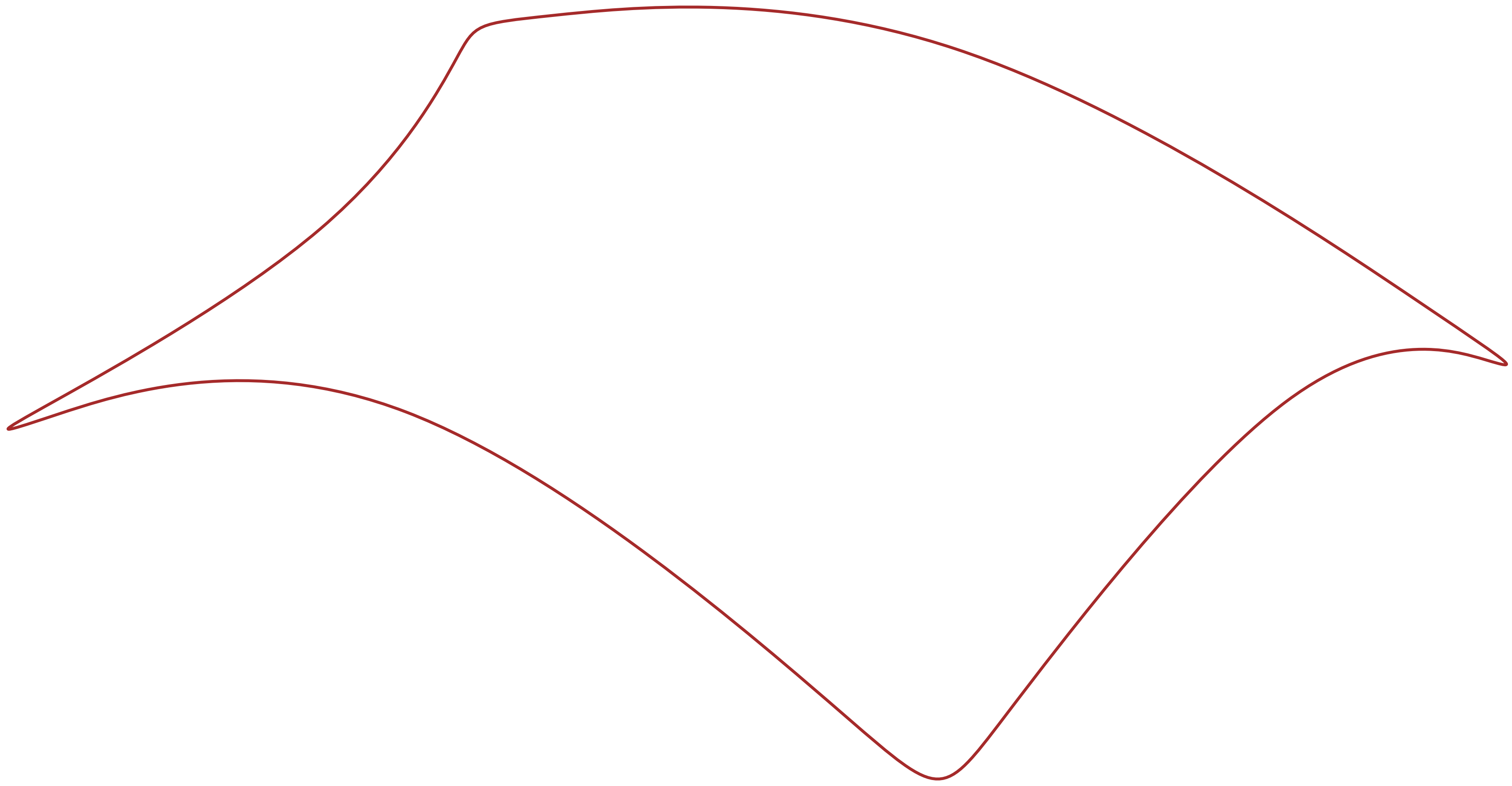
$\|xy\| \geq 2\sin\psi$

Compute $\epsilon$ to ensure that

$\frac{1}{\epsilon} \times$ ⊙ encloses

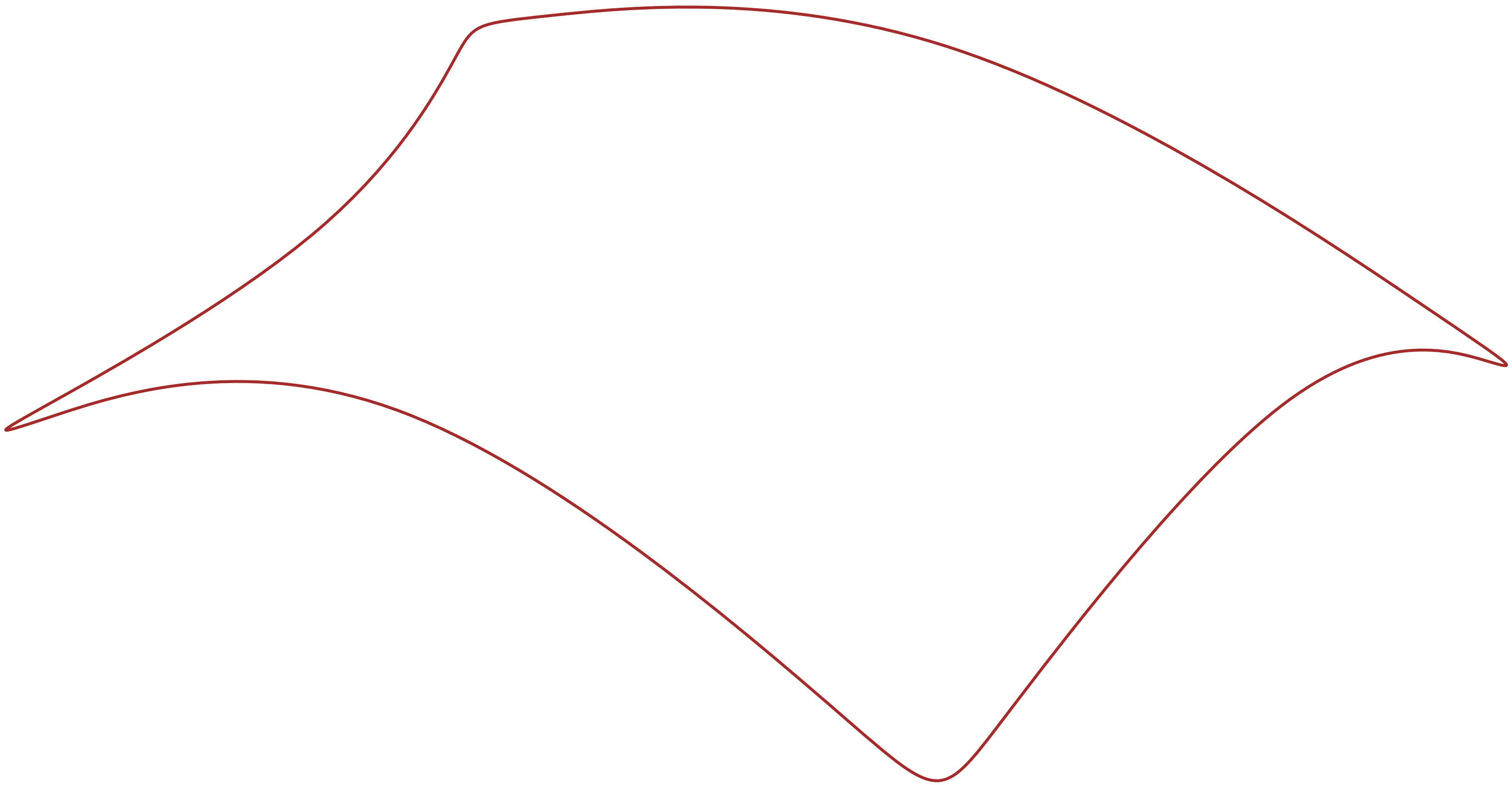
# Reconstruction

3D



# Reconstruction 3D

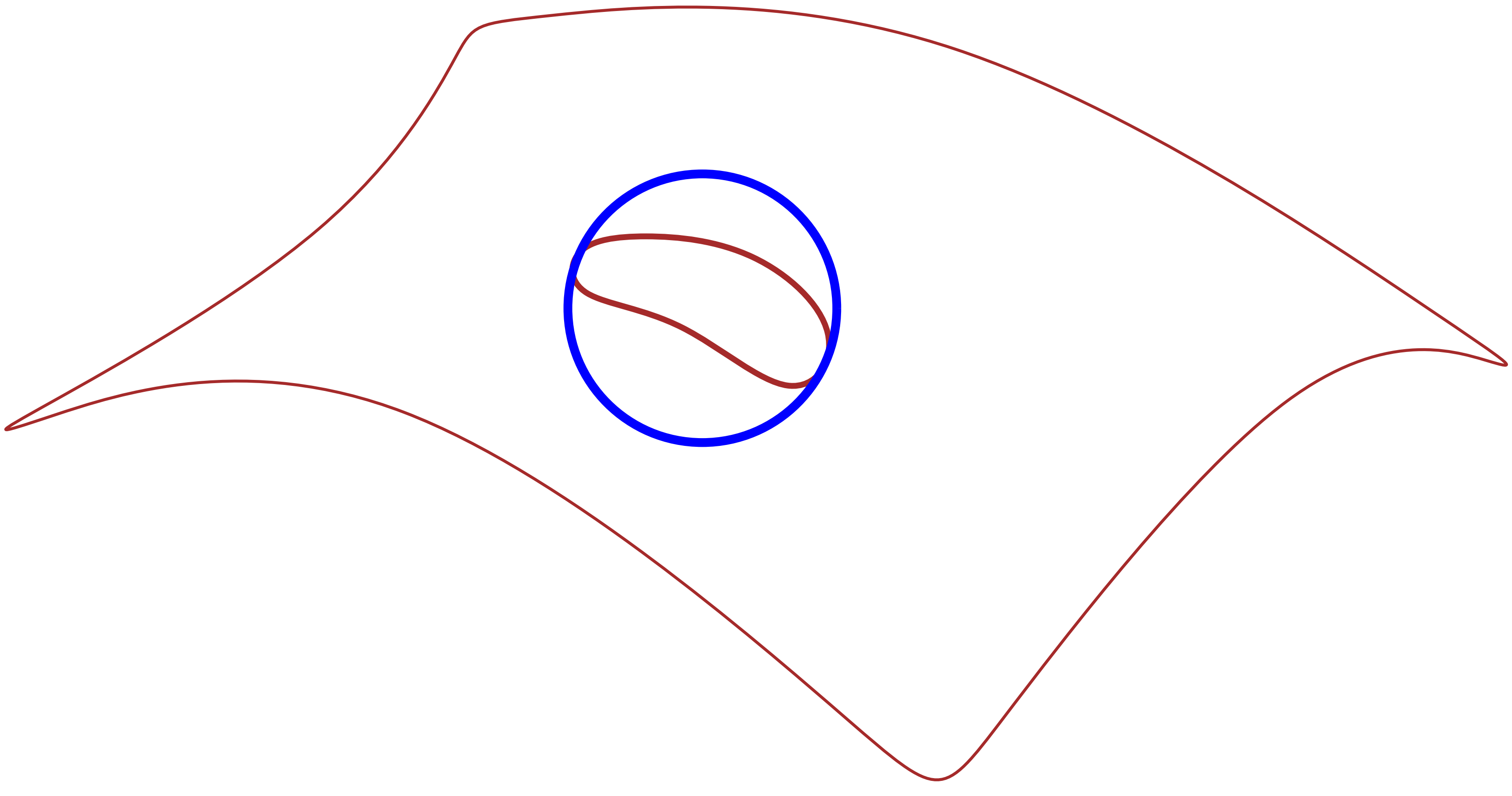
Difficulty: sliver



# Reconstruction 3D

Difficulty: sliver

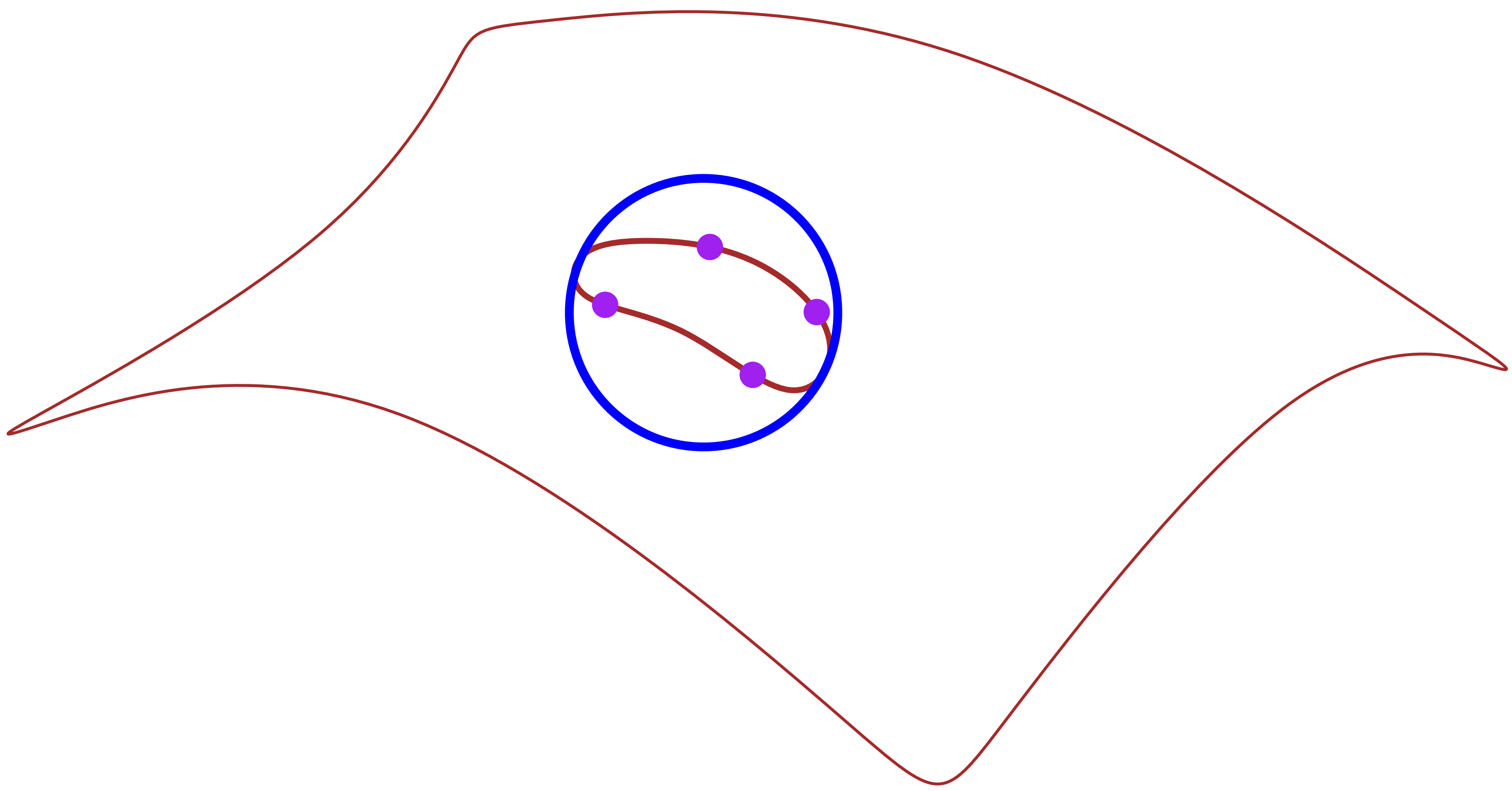
small sphere



# Reconstruction 3D

Difficulty: sliver

small sphere four sample points

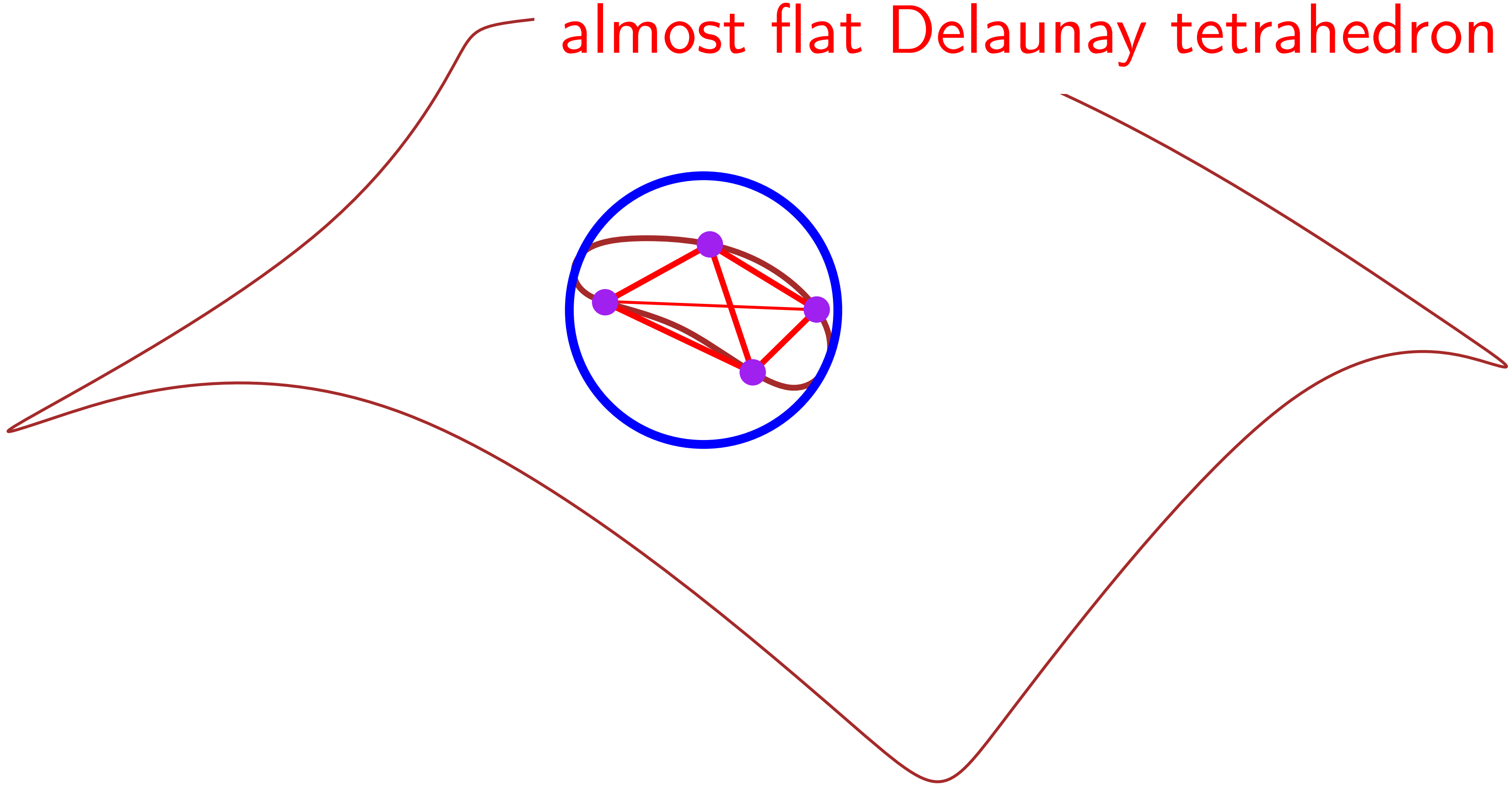


# Reconstruction 3D

Difficulty: sliver

small sphere four sample points

almost flat Delaunay tetrahedron

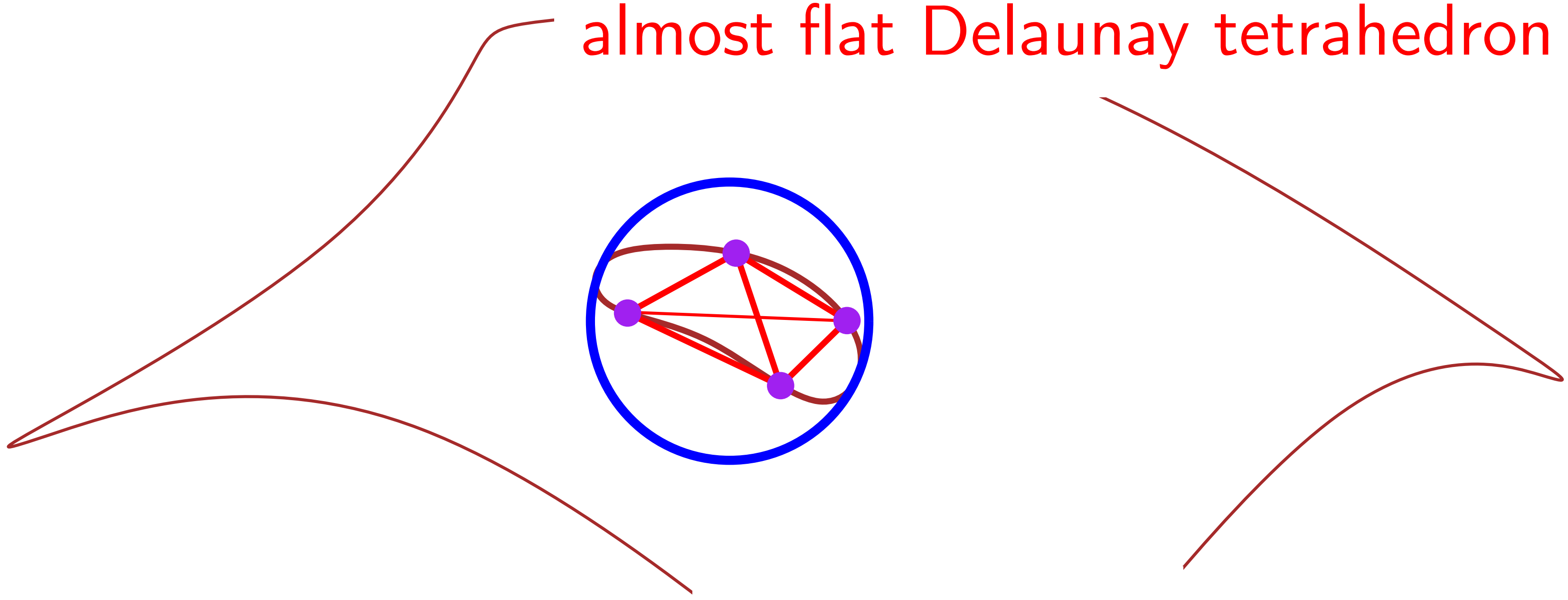


# Reconstruction 3D

Difficulty: sliver

small sphere four sample points

almost flat Delaunay tetrahedron



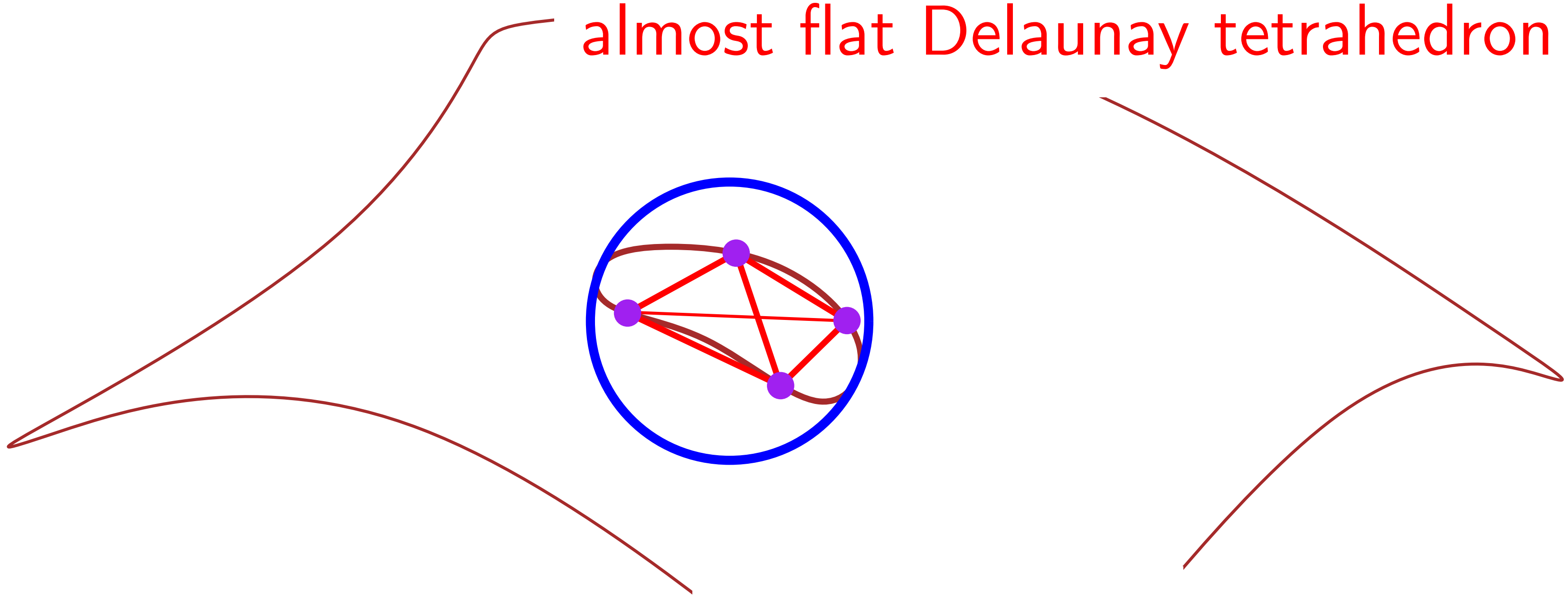
Which triangle belongs to reconstruction ?

# Reconstruction 3D

Difficulty: sliver

small sphere four sample points

almost flat Delaunay tetrahedron

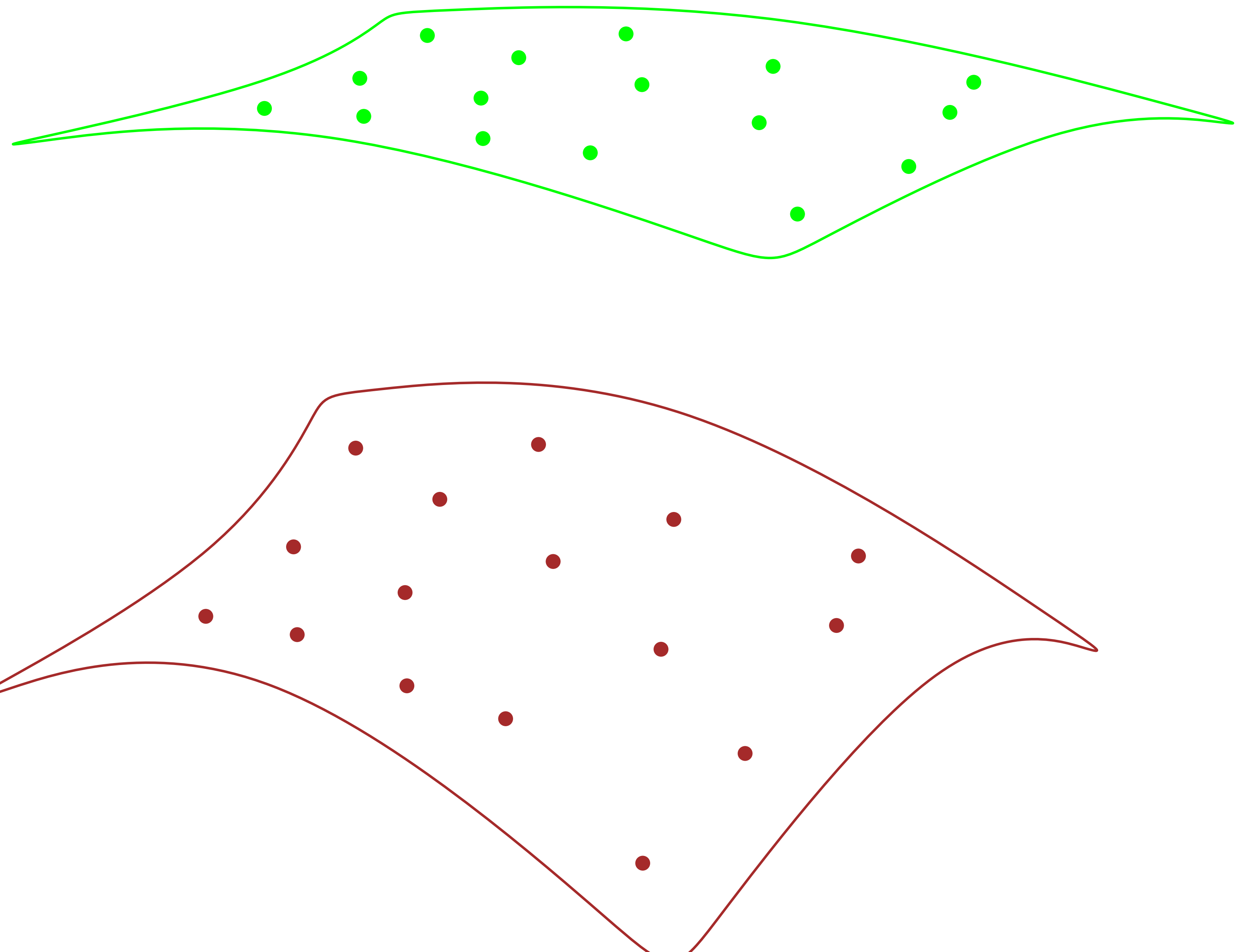


Which triangle belongs to reconstruction ?

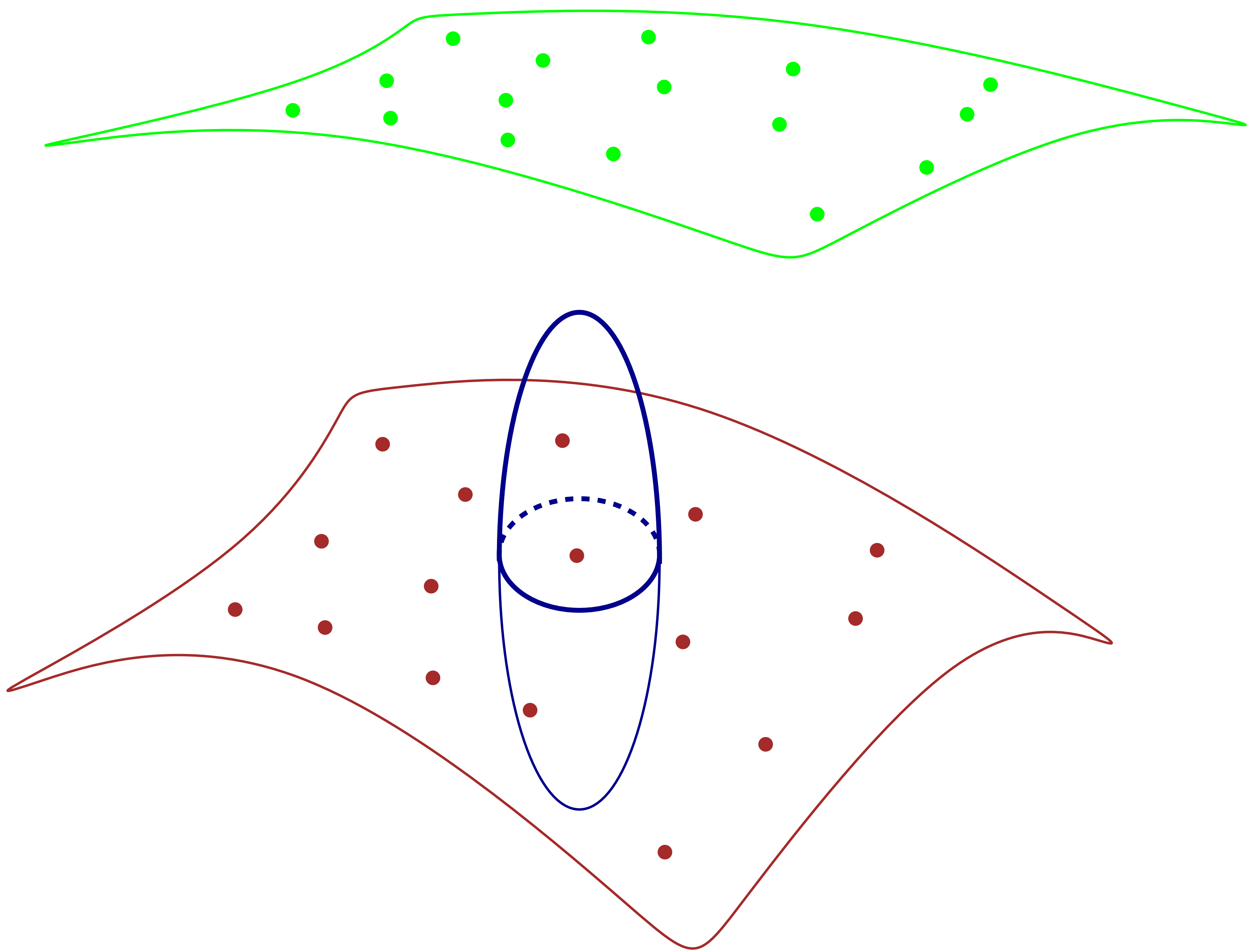
Crust: Voronoi vertices may kill useful triangles

# Reconstruction

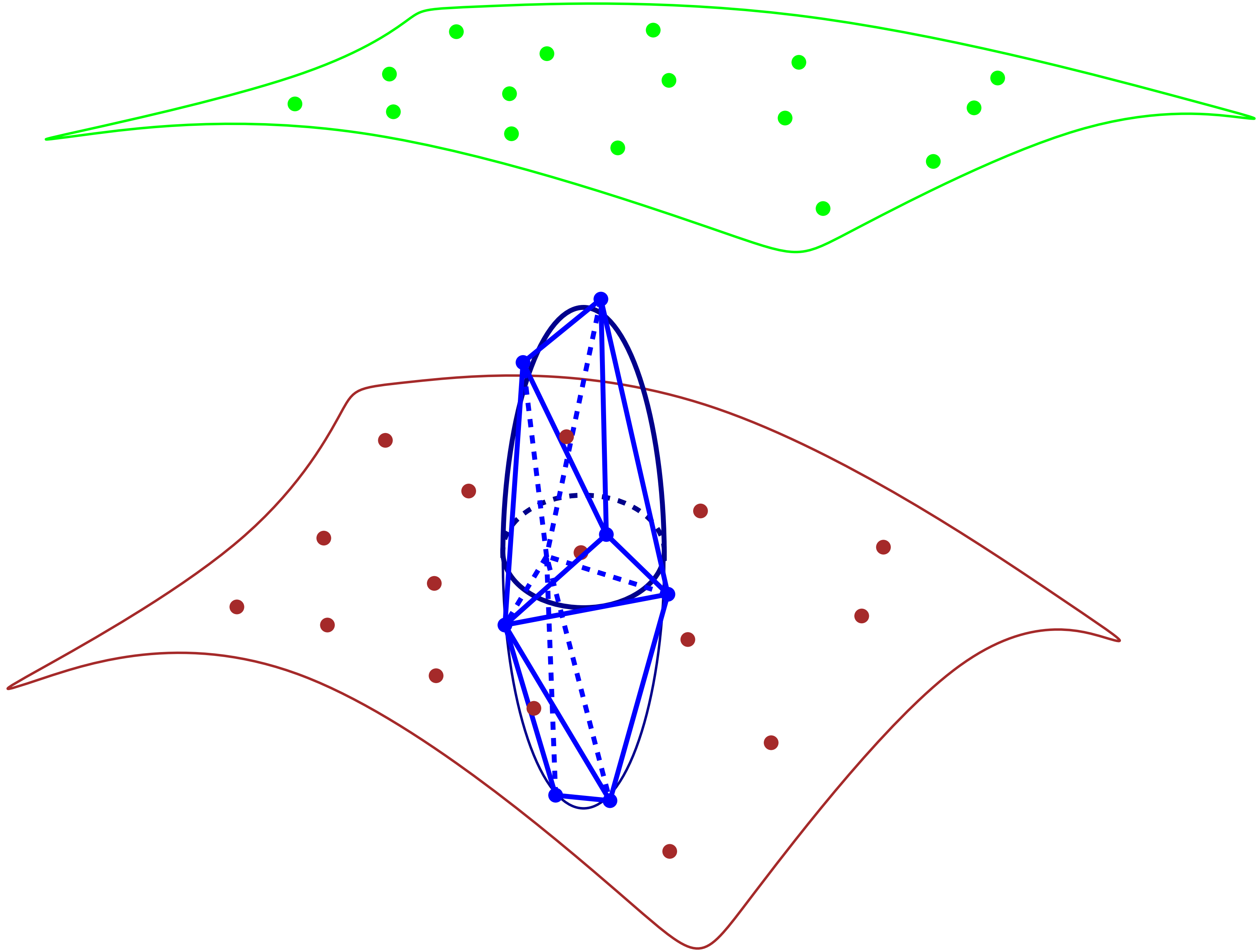
3D



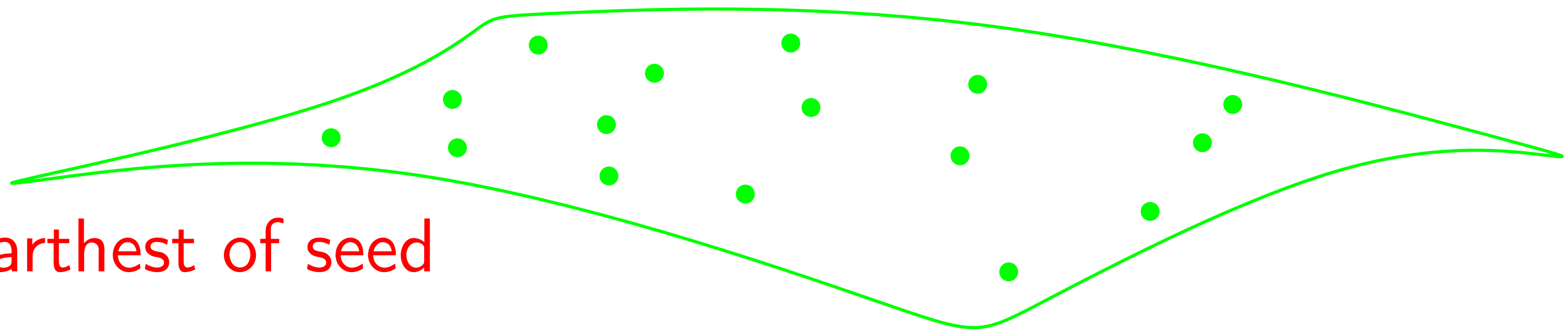
# Reconstruction 3D



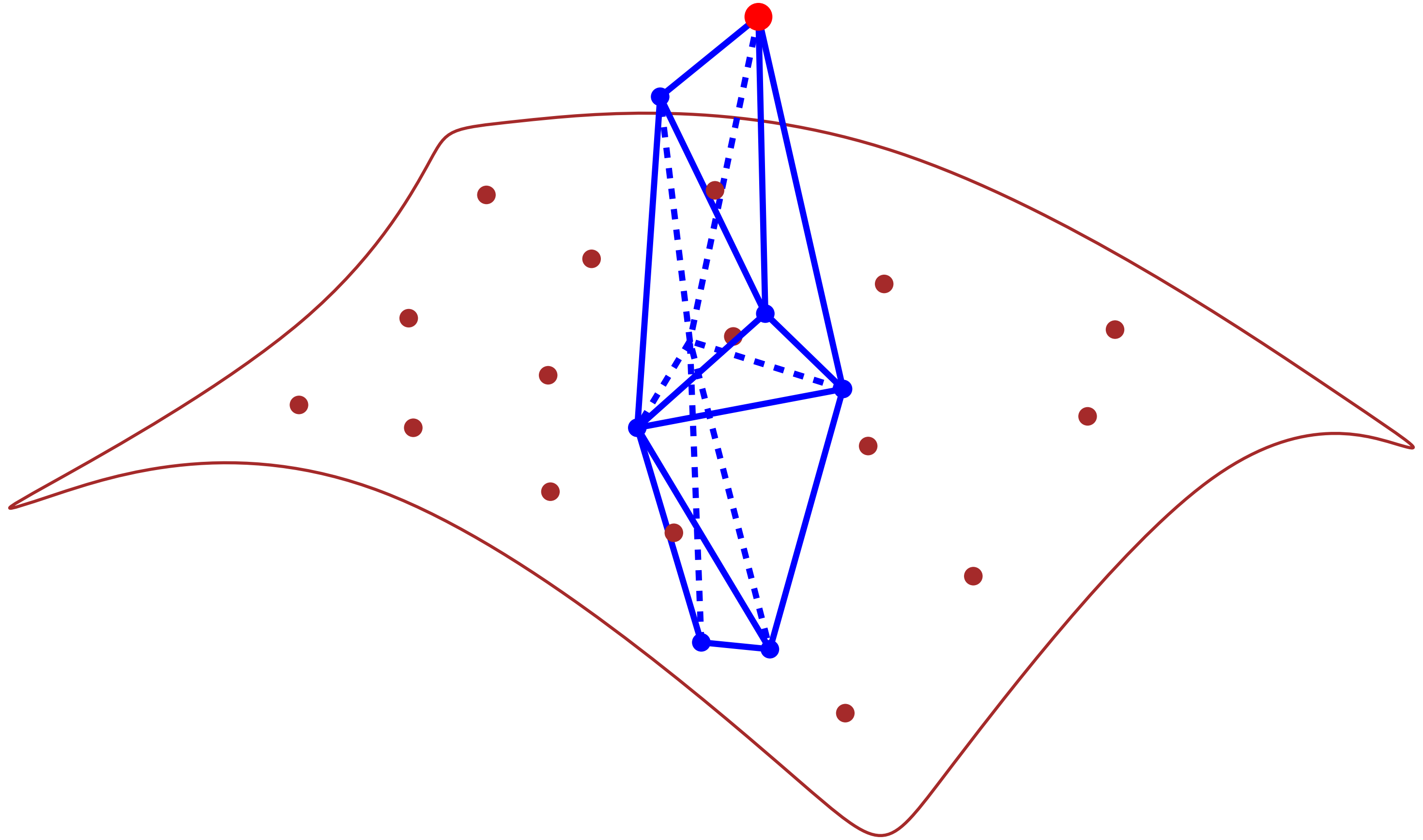
# Reconstruction 3D



# Reconstruction 3D



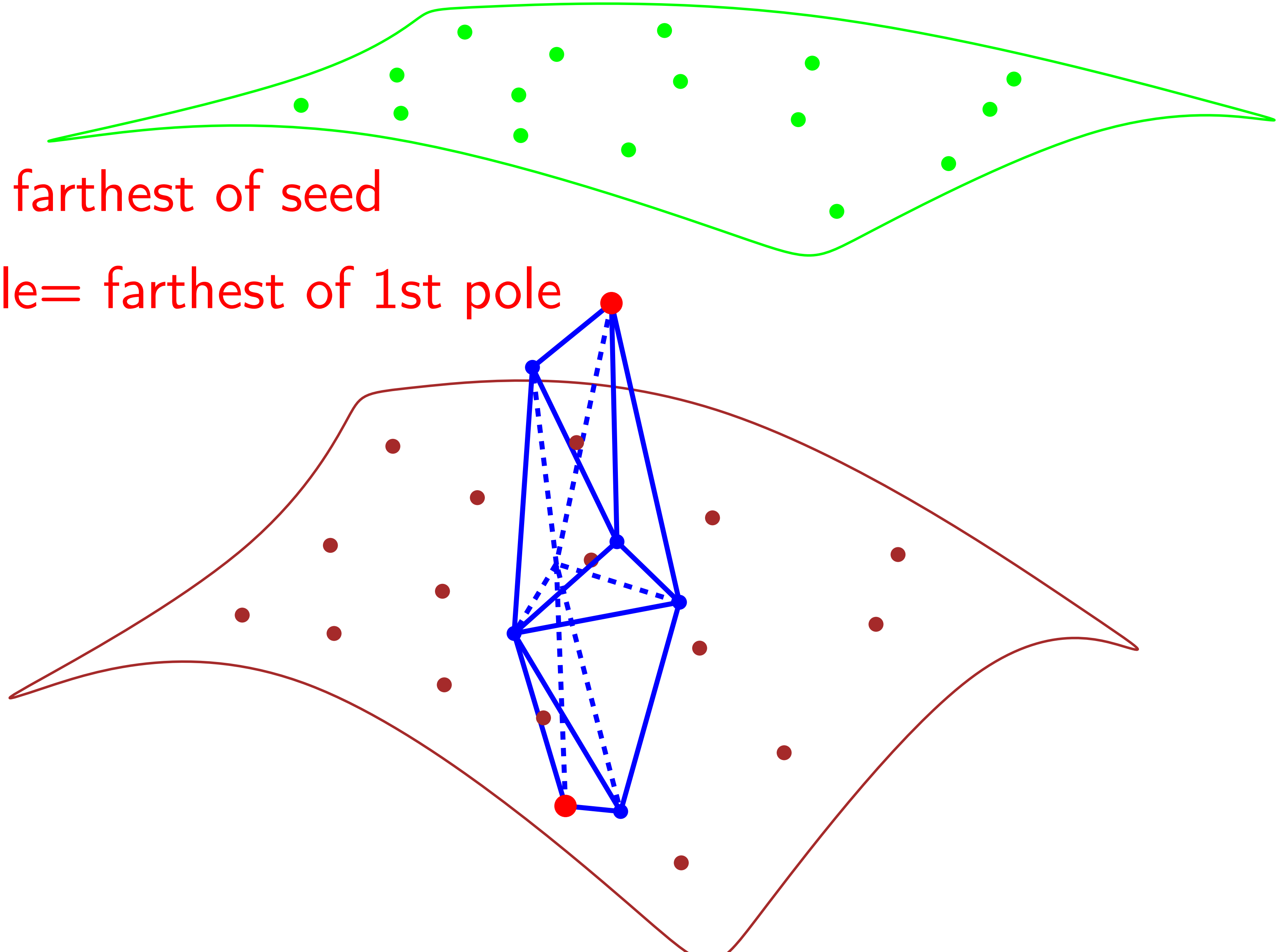
Pole = farthest of seed



# Reconstruction 3D

Pole = farthest of seed

2nd pole= farthest of 1st pole

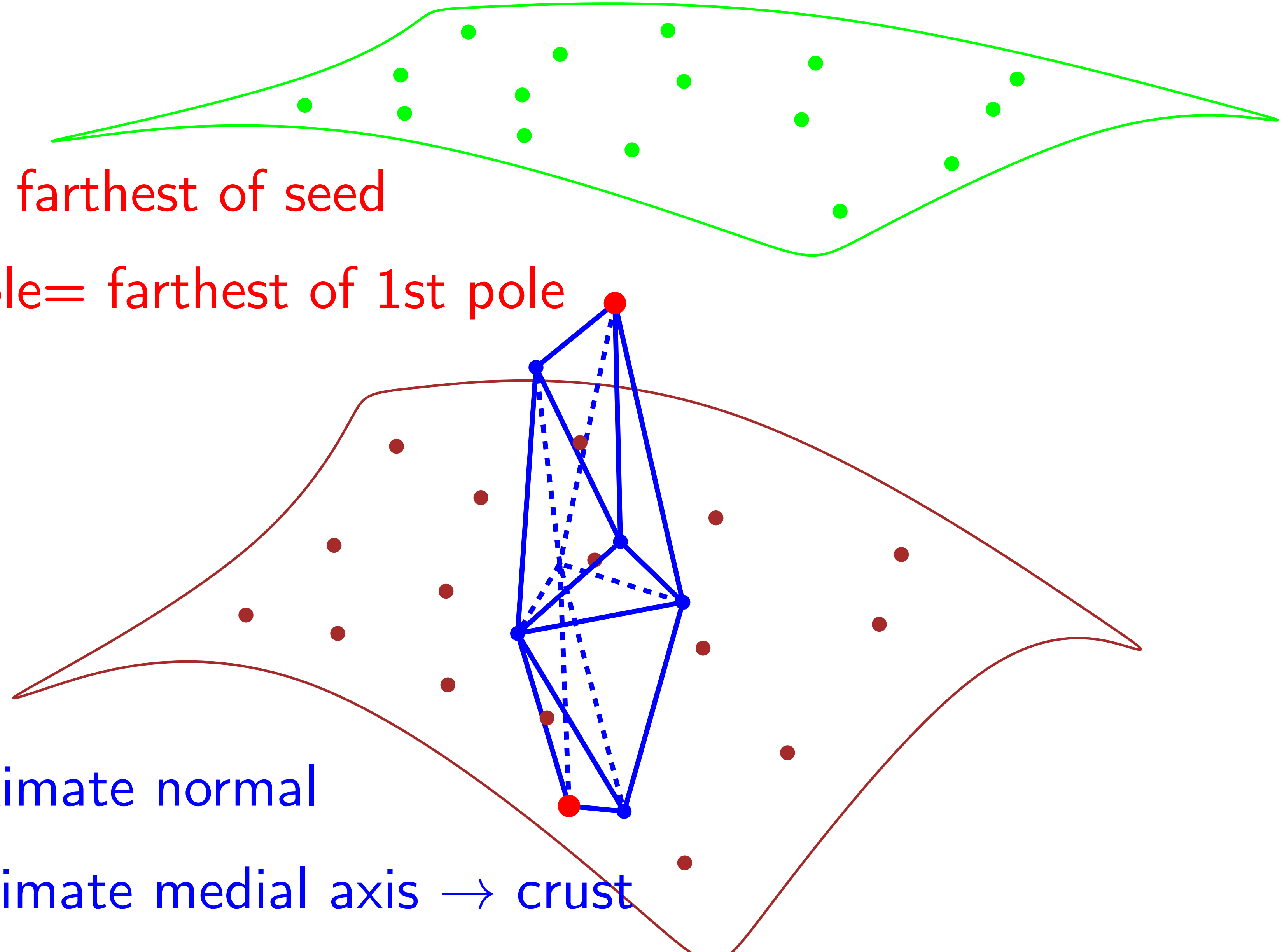


# Reconstruction

3D

Pole = farthest of seed

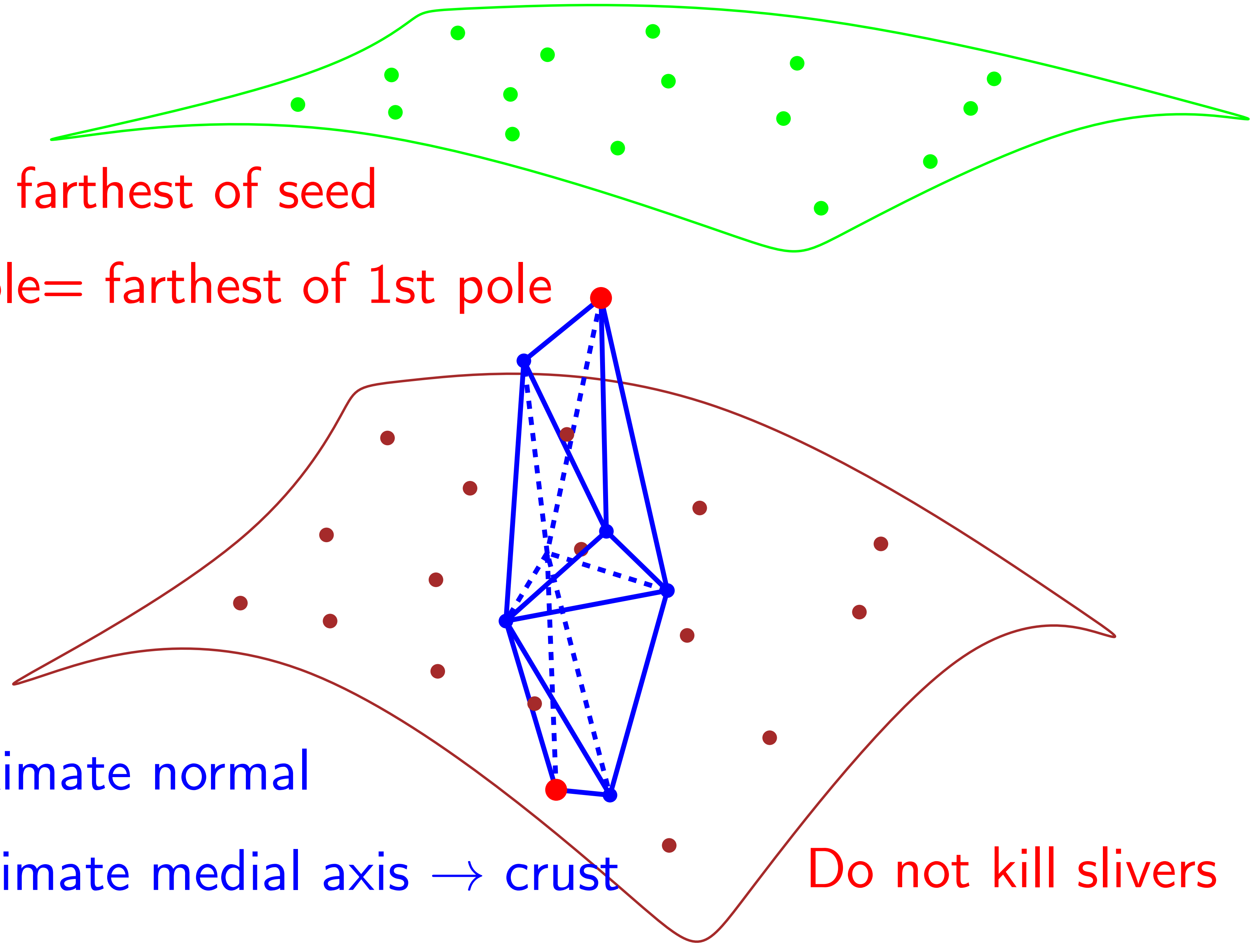
2nd pole= farthest of 1st pole



Approximate normal

Approximate medial axis $\rightarrow$ crust

# Reconstruction 3D



Pole = farthest of seed

2nd pole= farthest of 1st pole

Approximate normal

Approximate medial axis → crust

Do not kill slivers

# Meshing

# Meshing

Discretize space to solve (differential) equations

Finite elements

Finite differences

# Meshing

Discretize space to solve (differential) equations

- Finite elements
- Finite differences

Good mesh:

- Control shape of elements (no small angles)
- Control size of elements (adjust to function variability)
- Minimize number of elements

# Meshing

Gallery

Structured meshes (advancing front, deformation)

Delaunay mesh refinement

- [Ruppert]
- protecting small angles
- off-centers

Delaunay mesh optimization

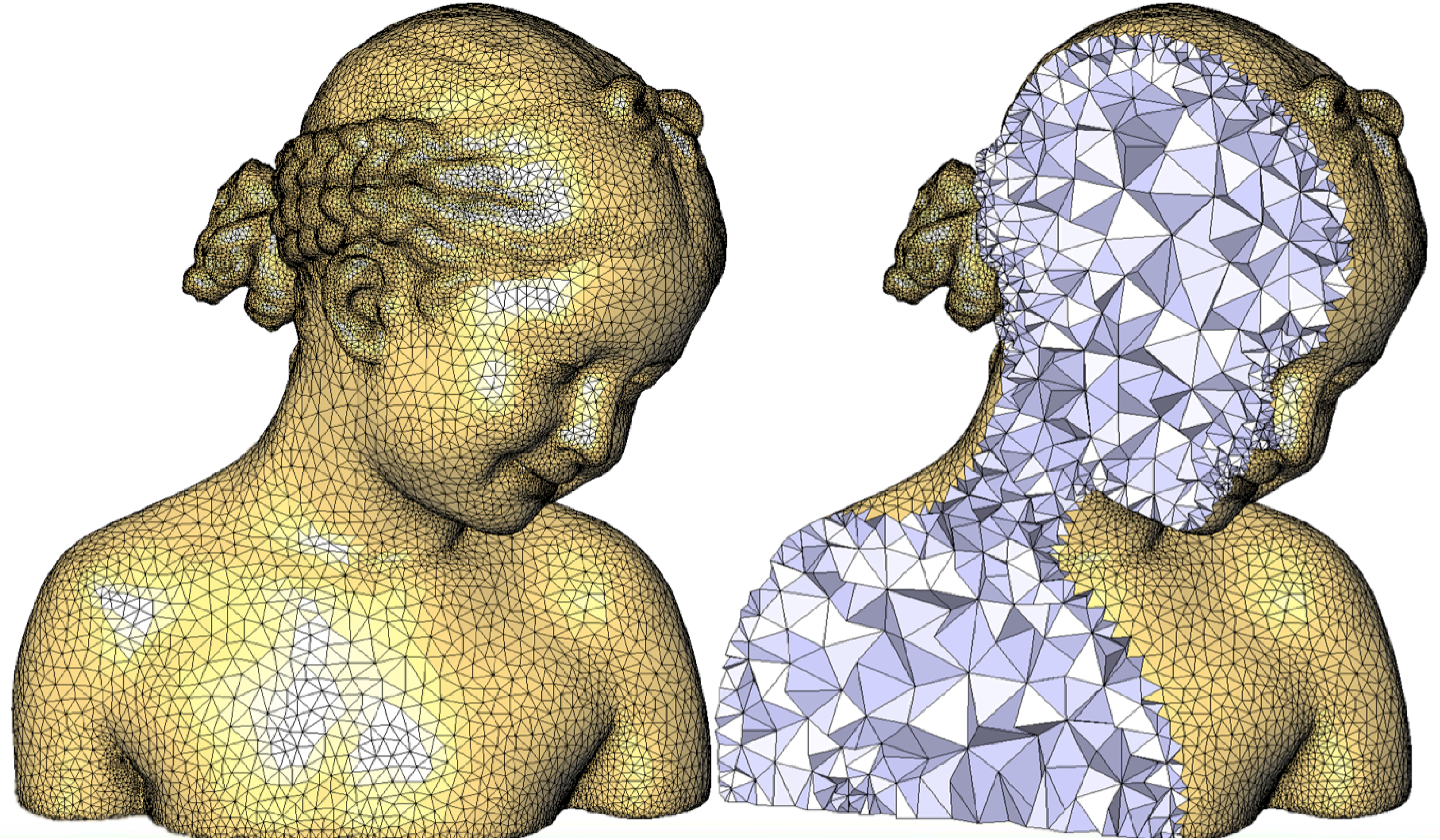
3D

# Meshing

## Gallery

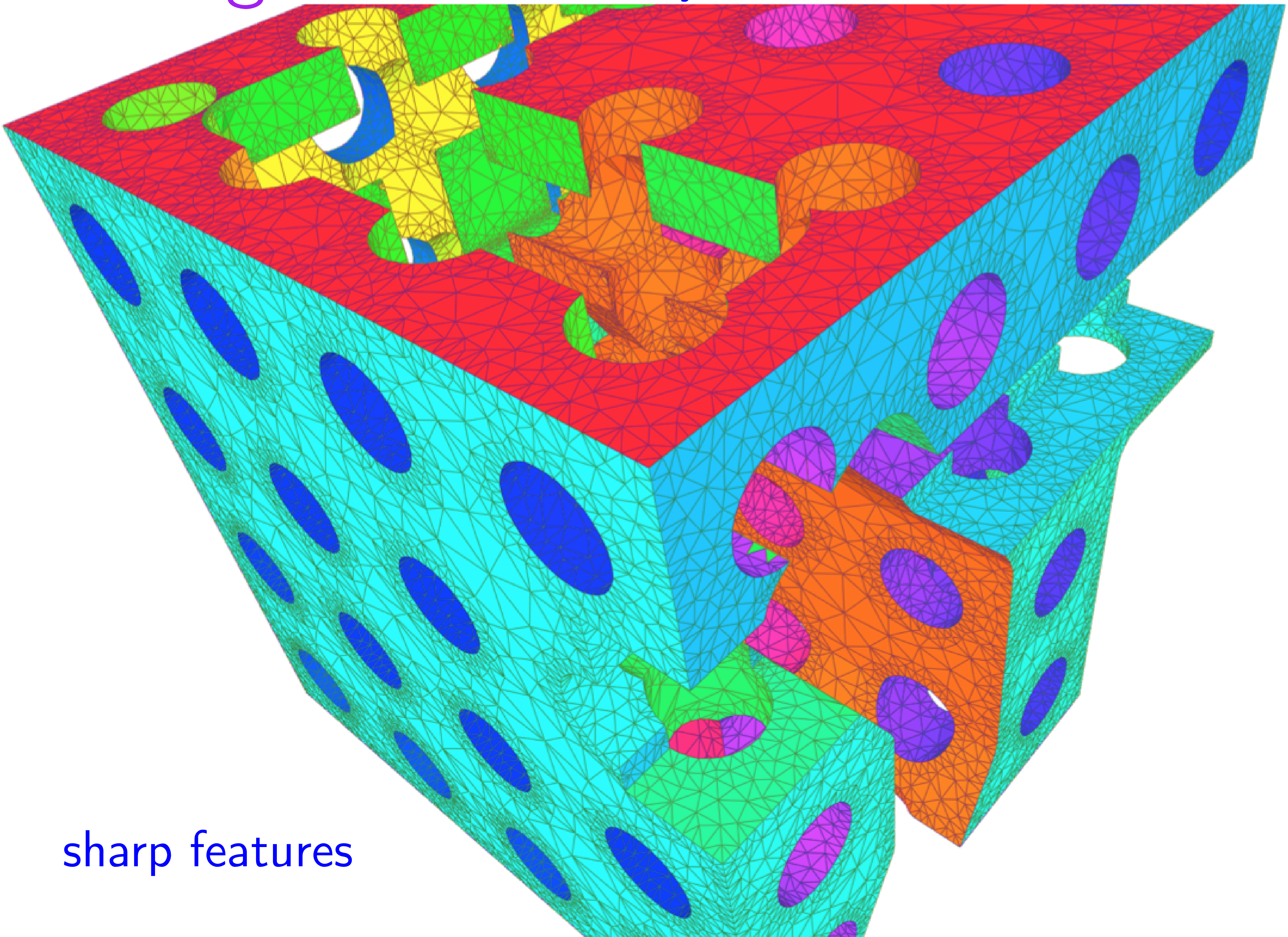
# Meshing

## Gallery



# Meshing

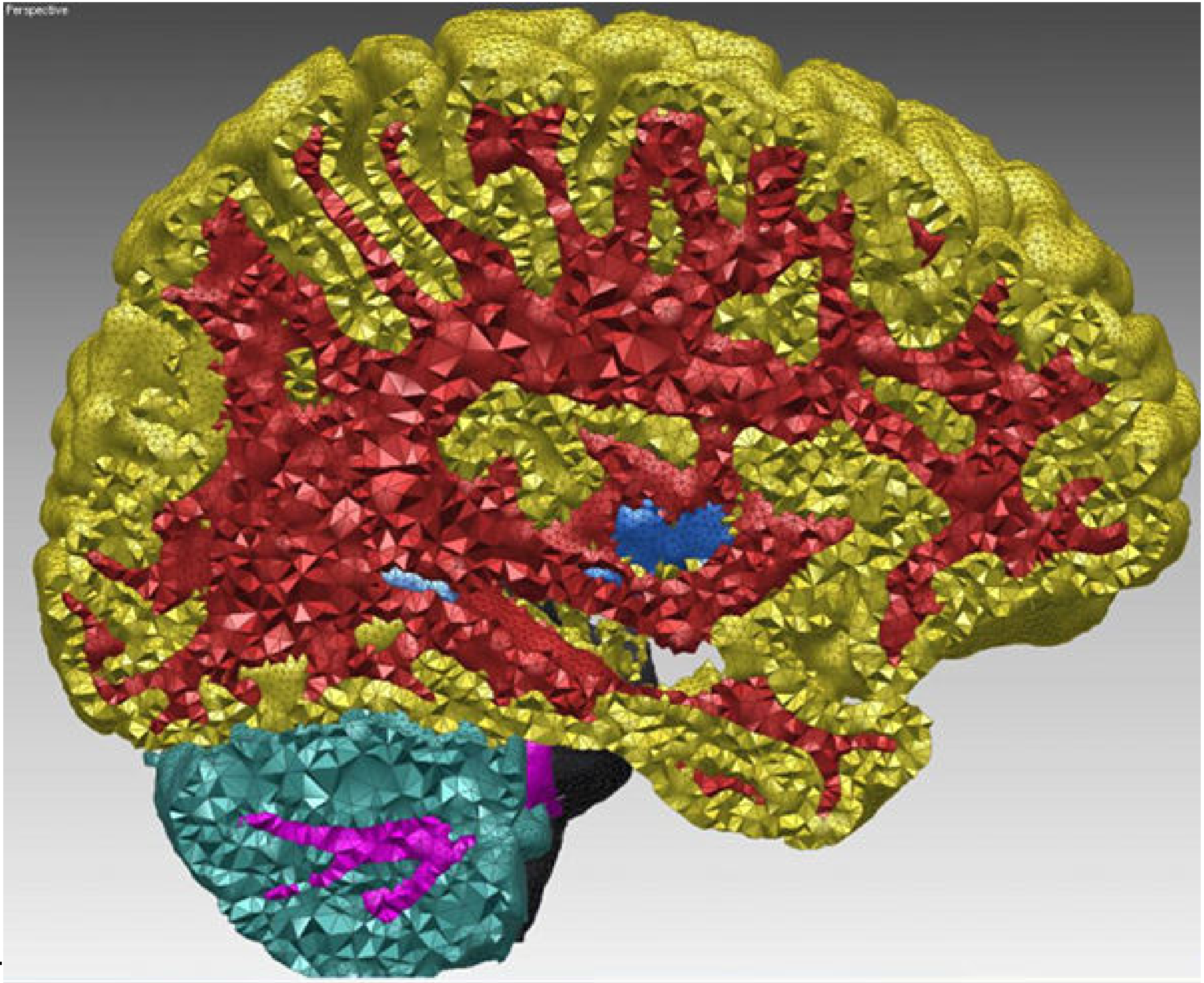
## Gallery



sharp features

# Meshing

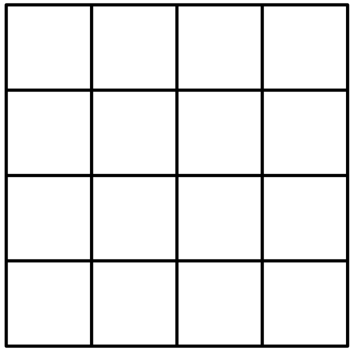
## Gallery



# Meshing

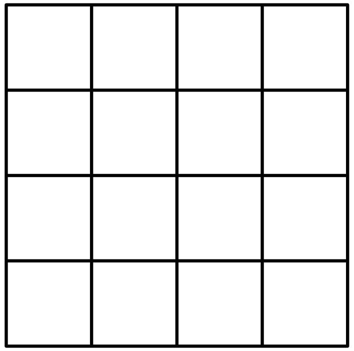


Regular grid

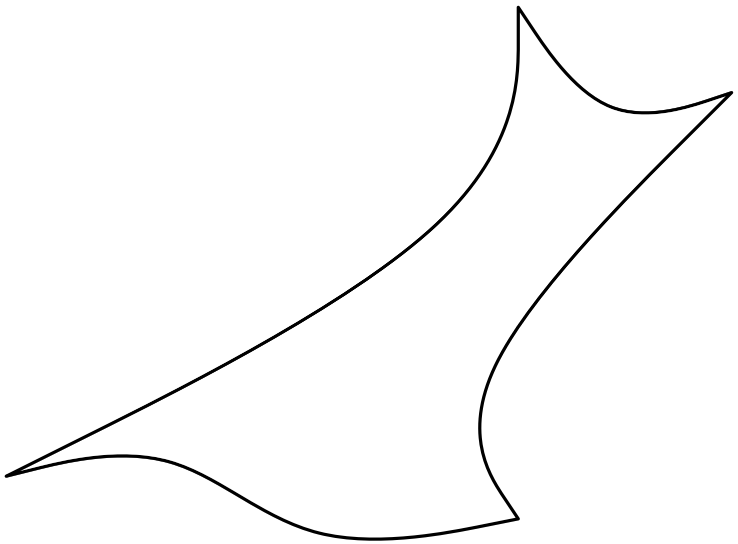


# Meshing

Regular grid



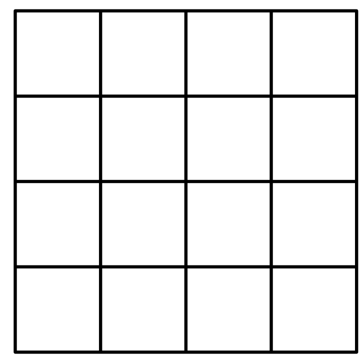
Shape



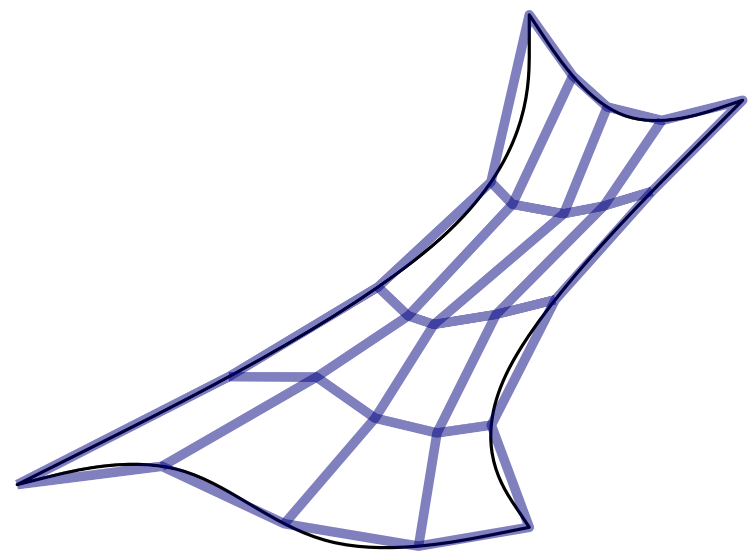
# Meshing

## Structured meshes

Regular grid



Shape



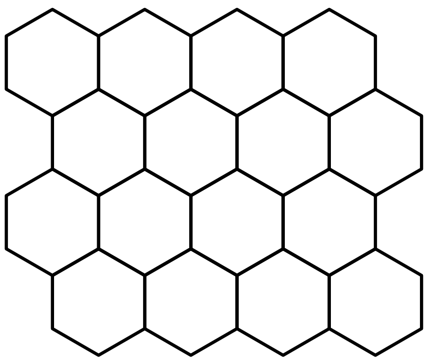
Deform

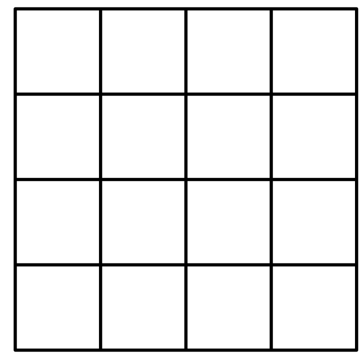
to fit the grid in the shape

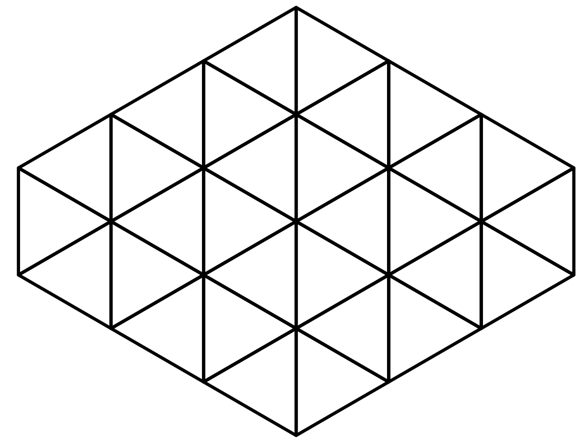
# Meshing

## Structured meshes

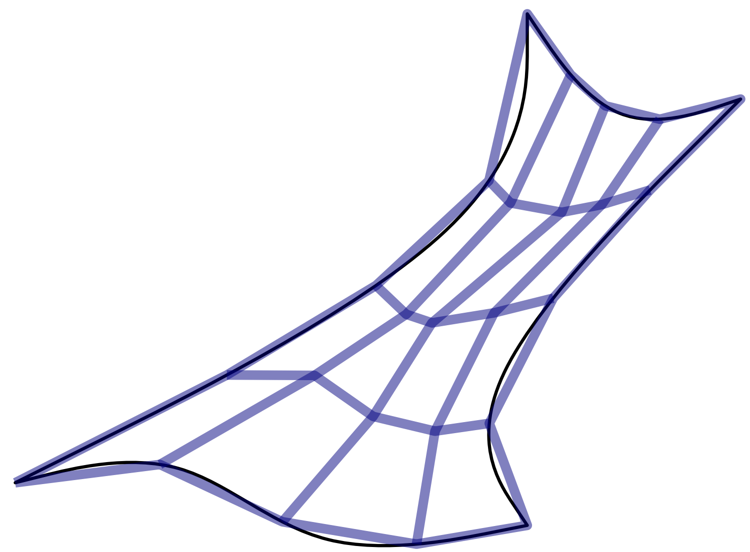
Regular grid







Shape



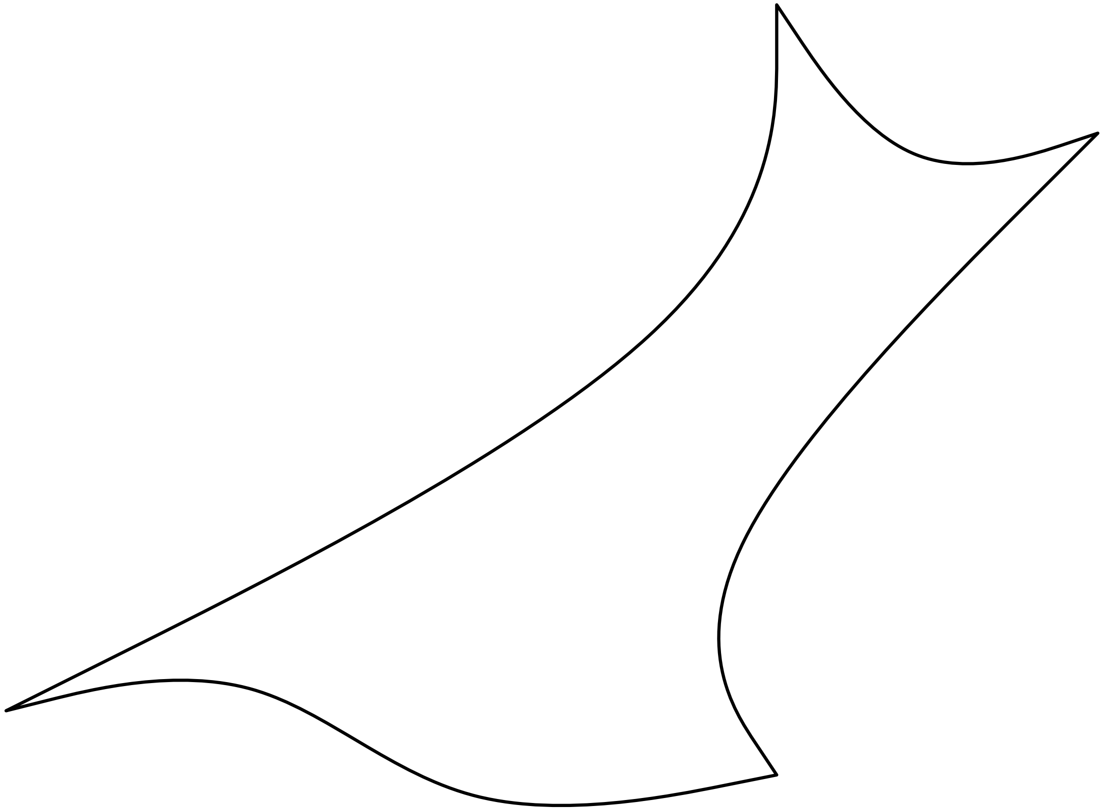
Deform

to fit the grid in the shape

# Meshing



Shape

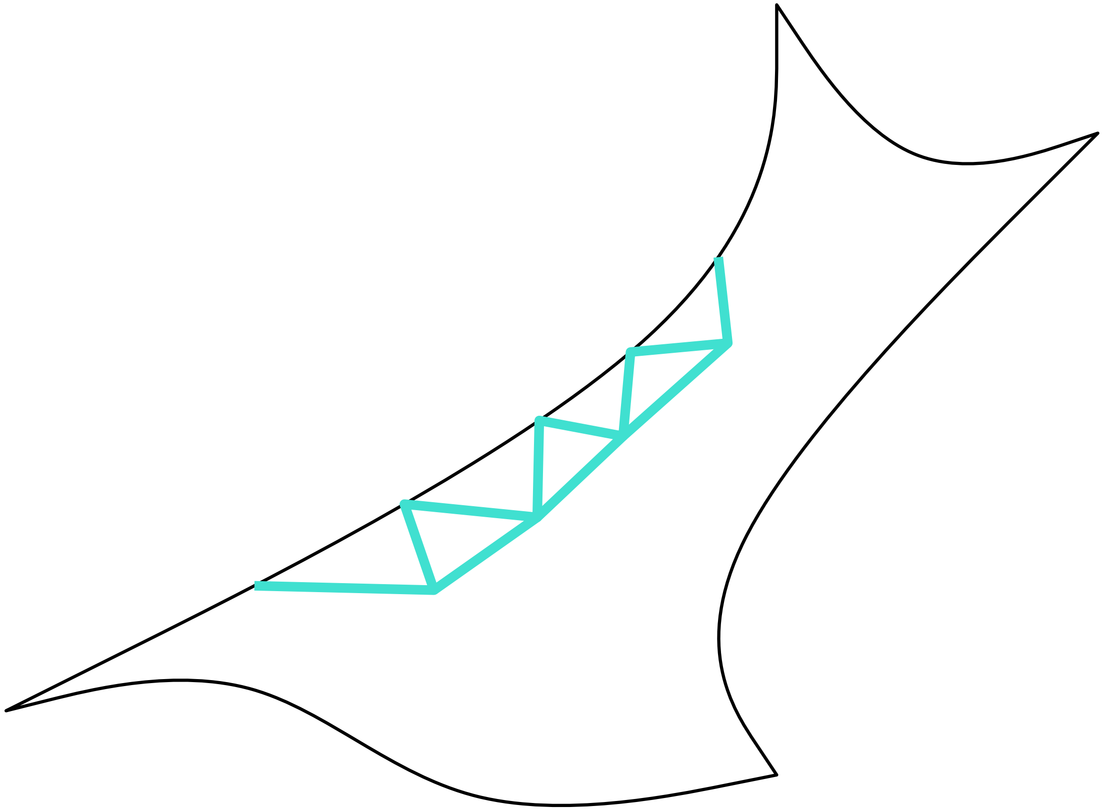


# Meshing



Shape

Advancing front

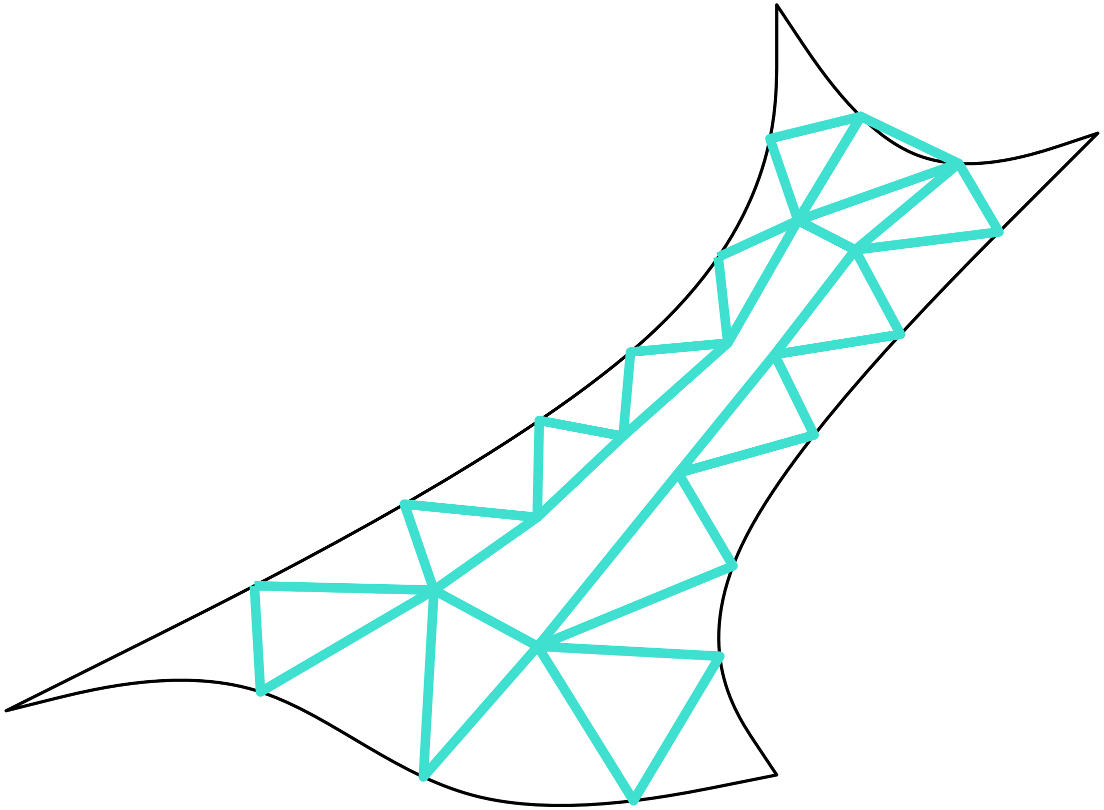


# Meshing



Shape

Advancing front

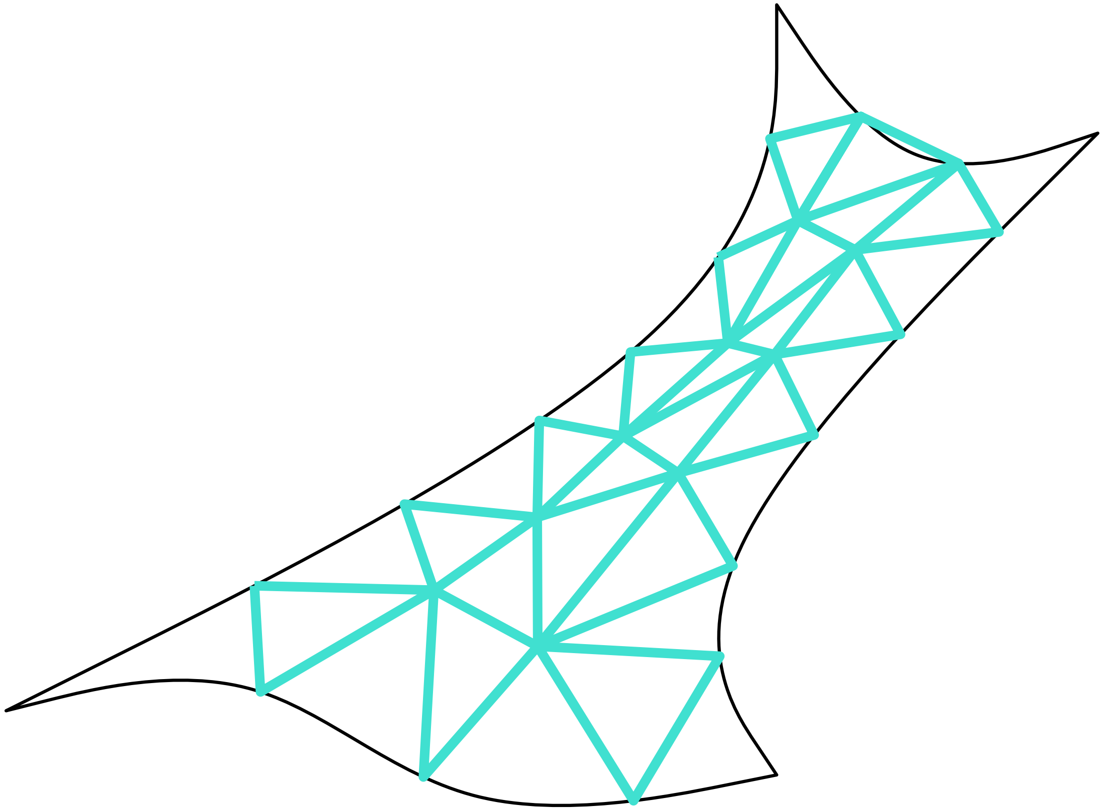


# Meshing



Shape

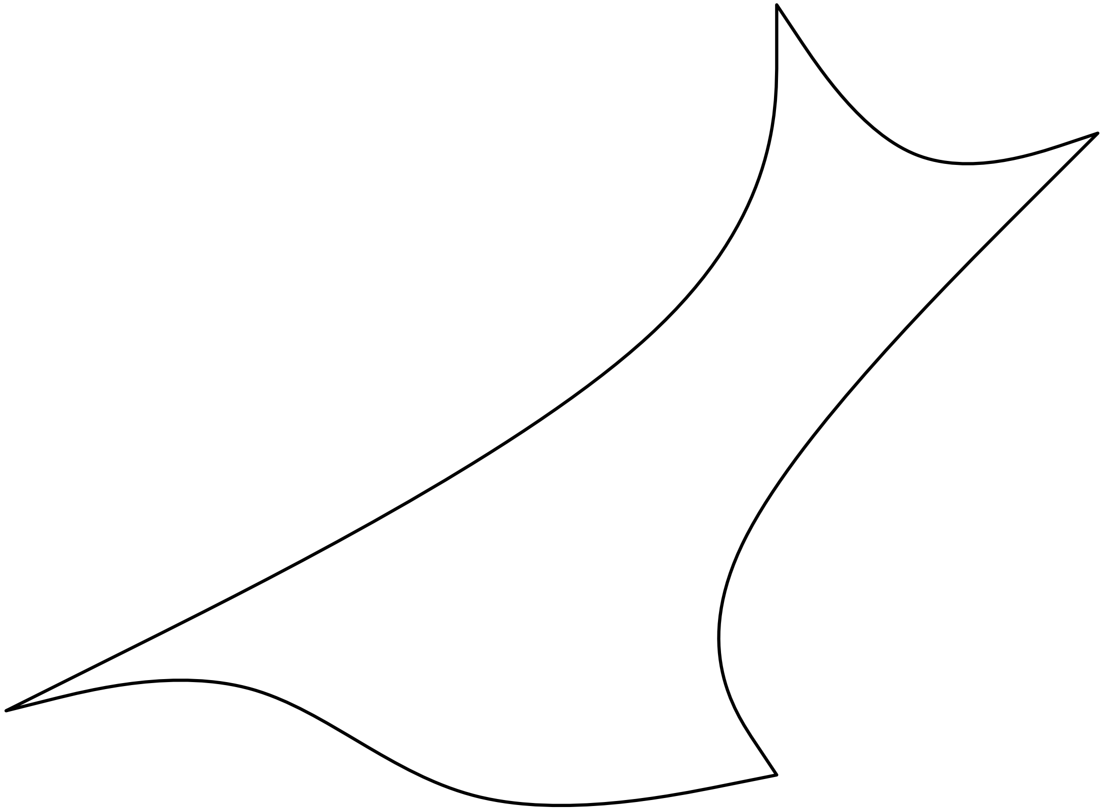
Advancing front



# Meshing



Shape

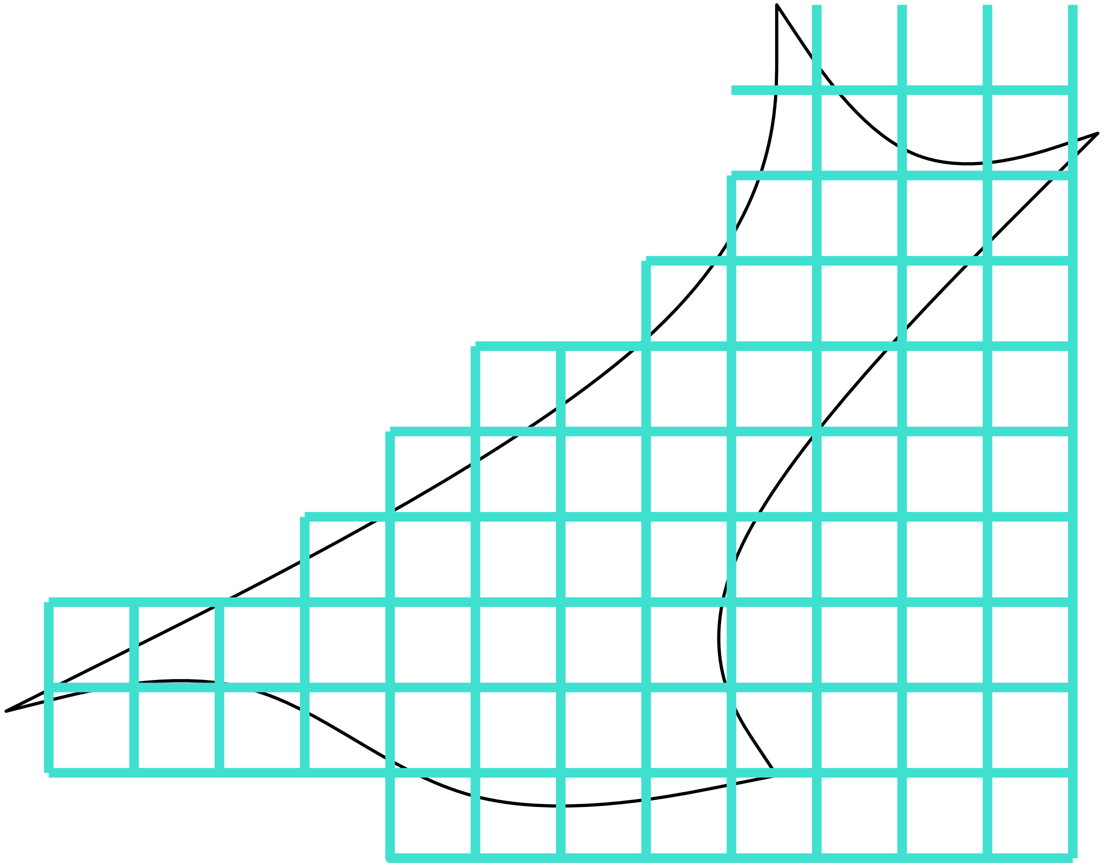


# Meshing



Shape

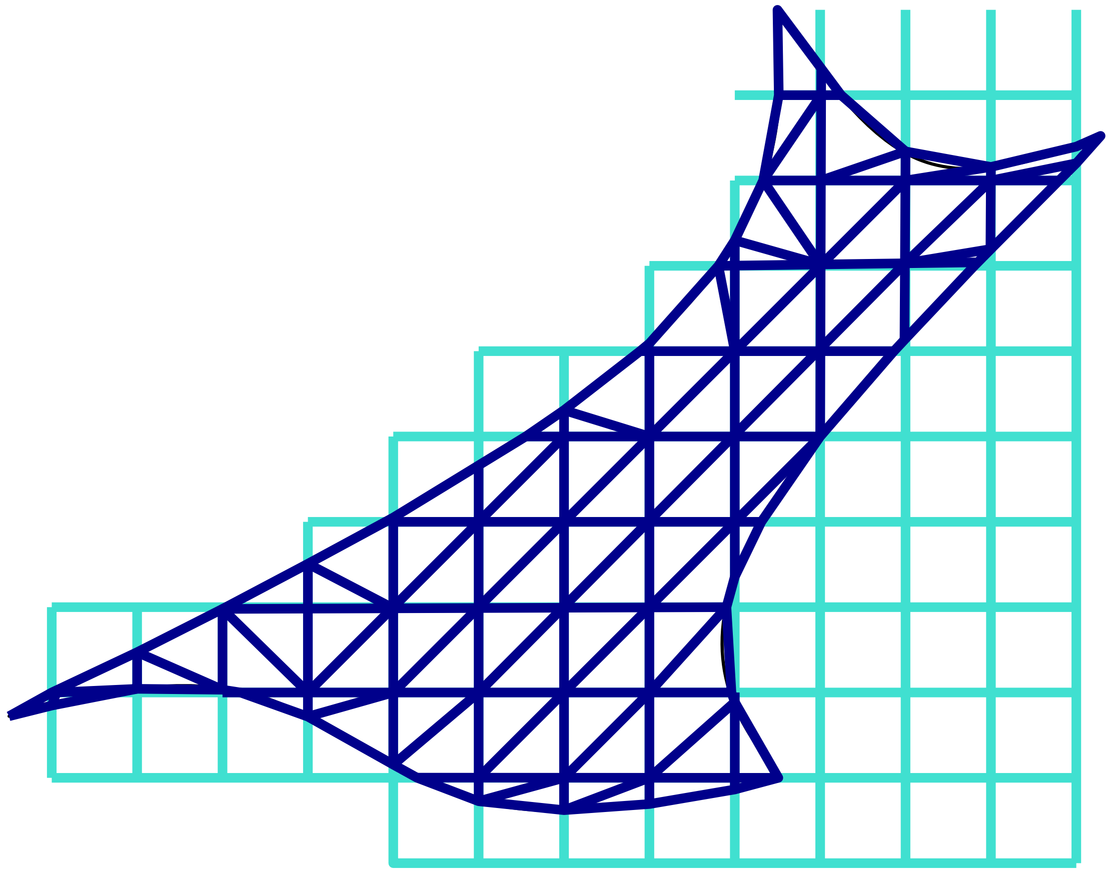
Add grid



# Meshing



Shape



Add grid

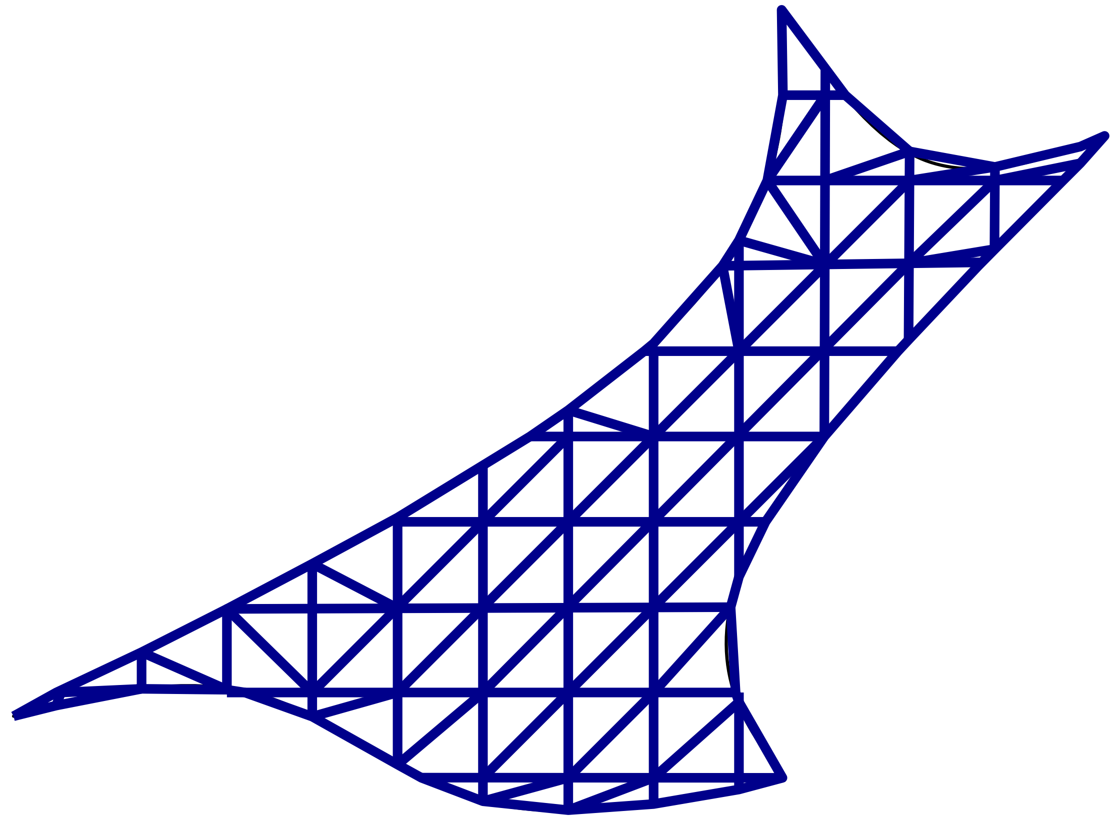
Triangulate

# Meshing

Structured meshes

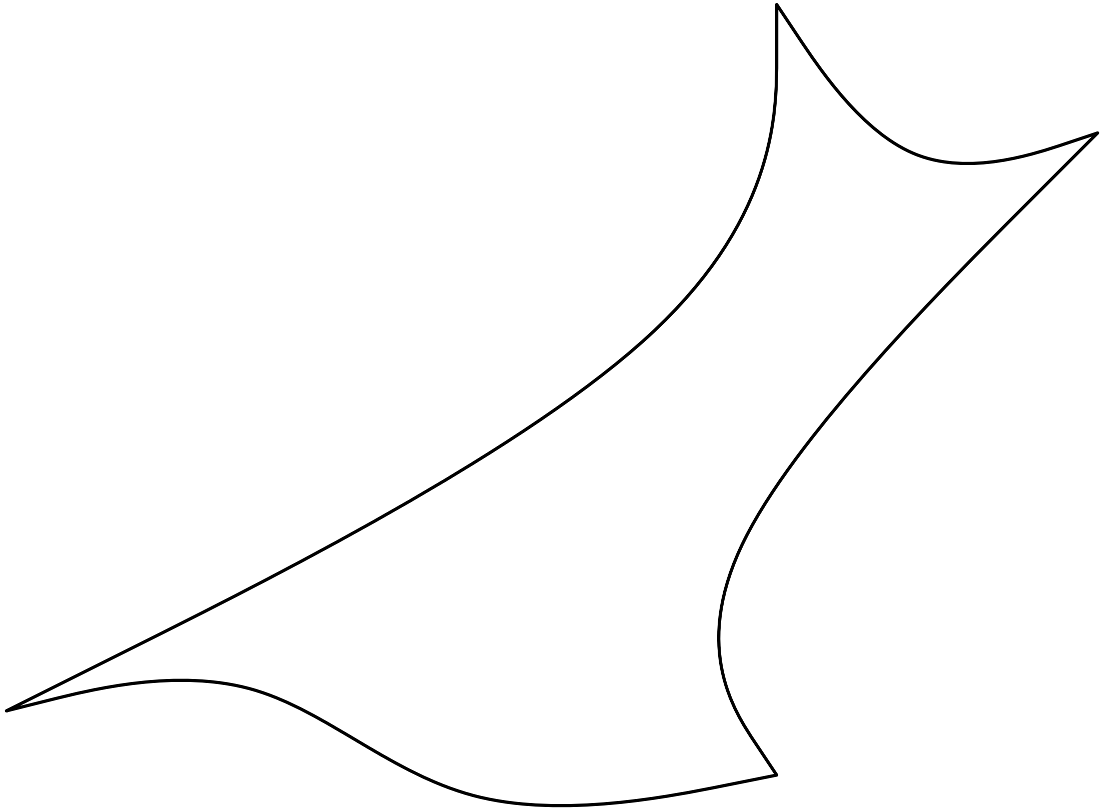
Shape

Triangulate



Uniform mesh

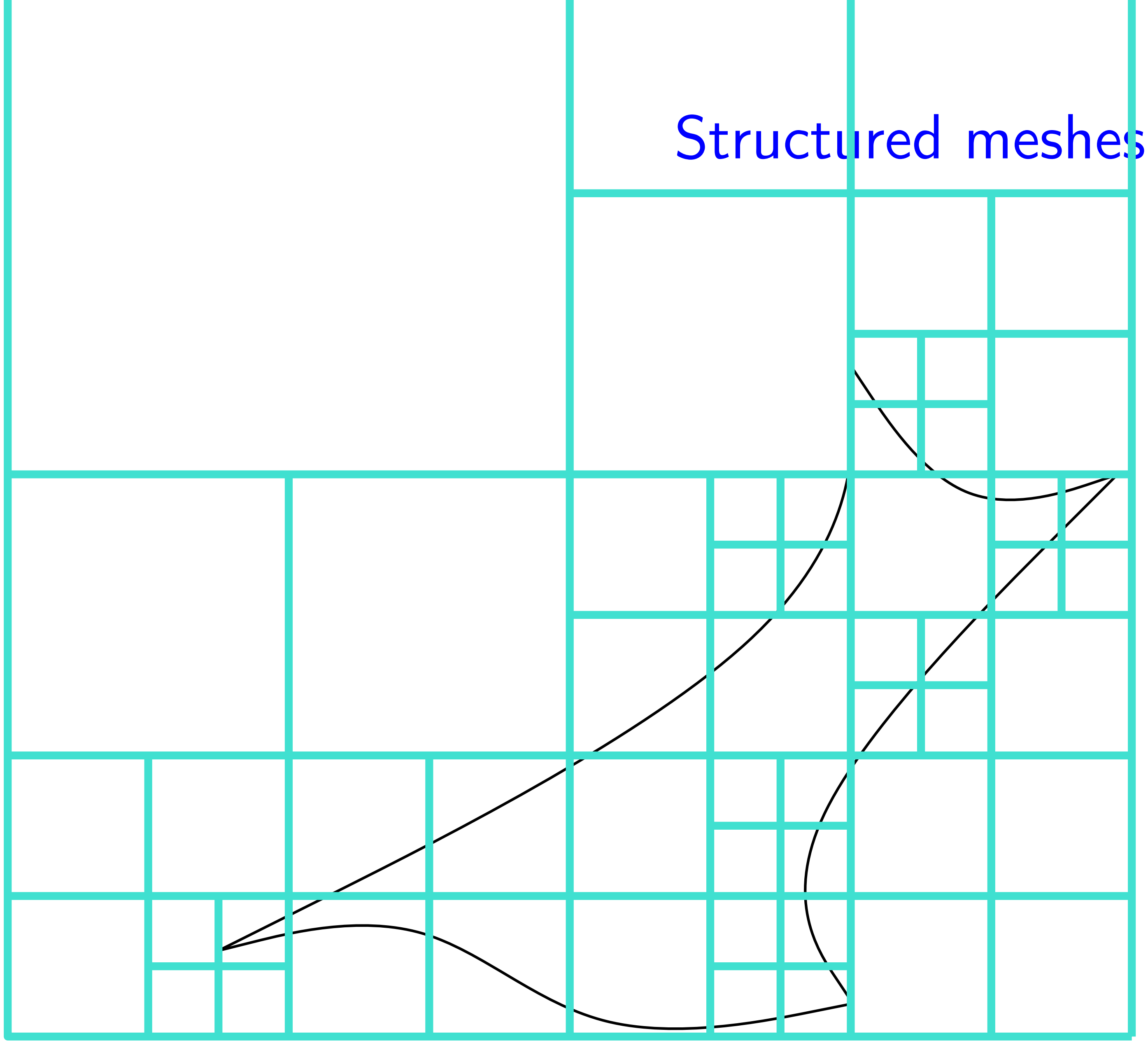
Shape



# Meshing

Structured meshes

Shape

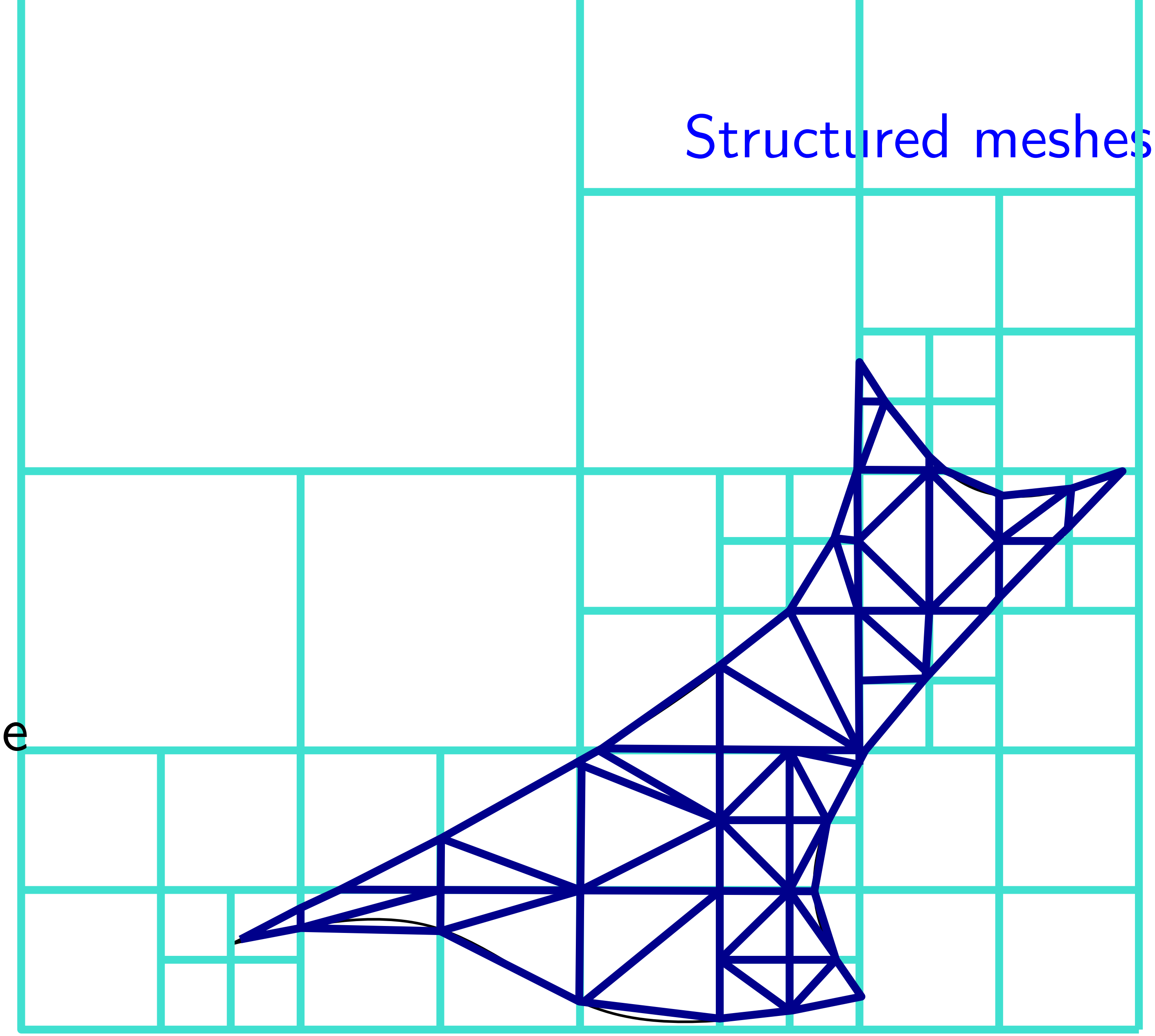


# Meshing

Structured meshes

Shape

Triangulate

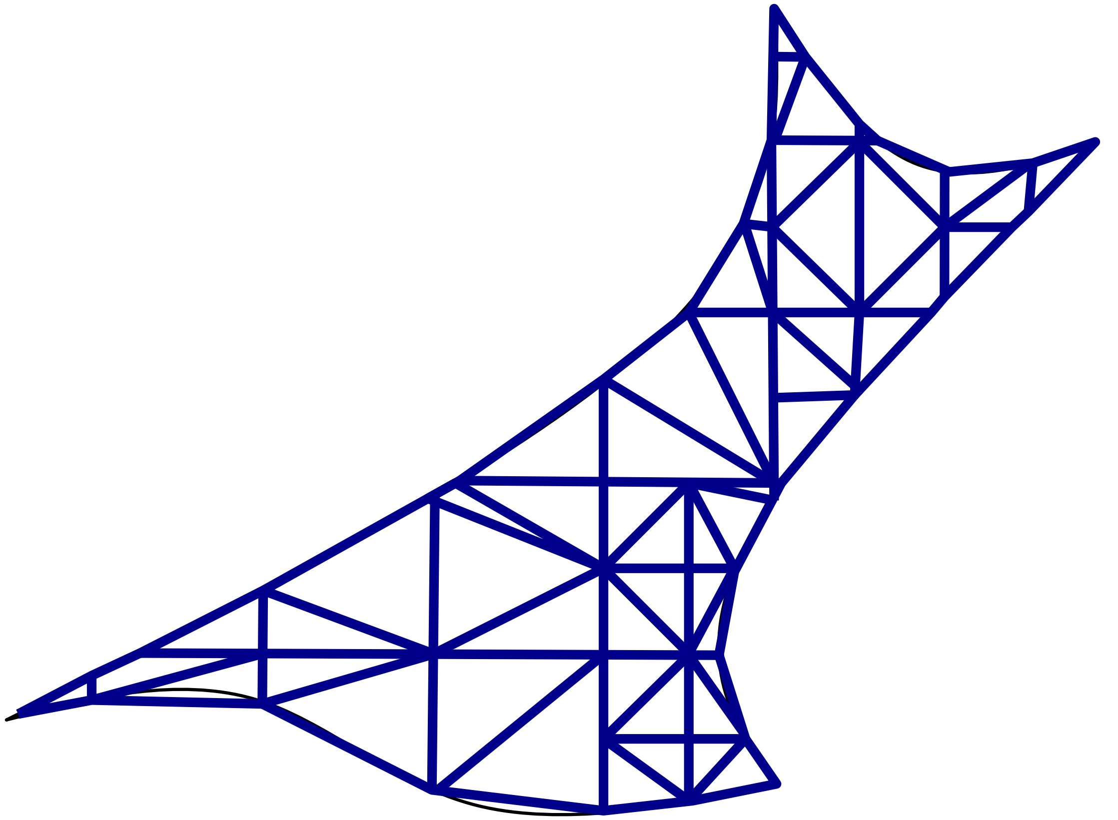


# Meshing

Structured meshes

Shape

Triangulate



Adaptive mesh

# Meshing

Delaunay mesh refinement [Ruppert]

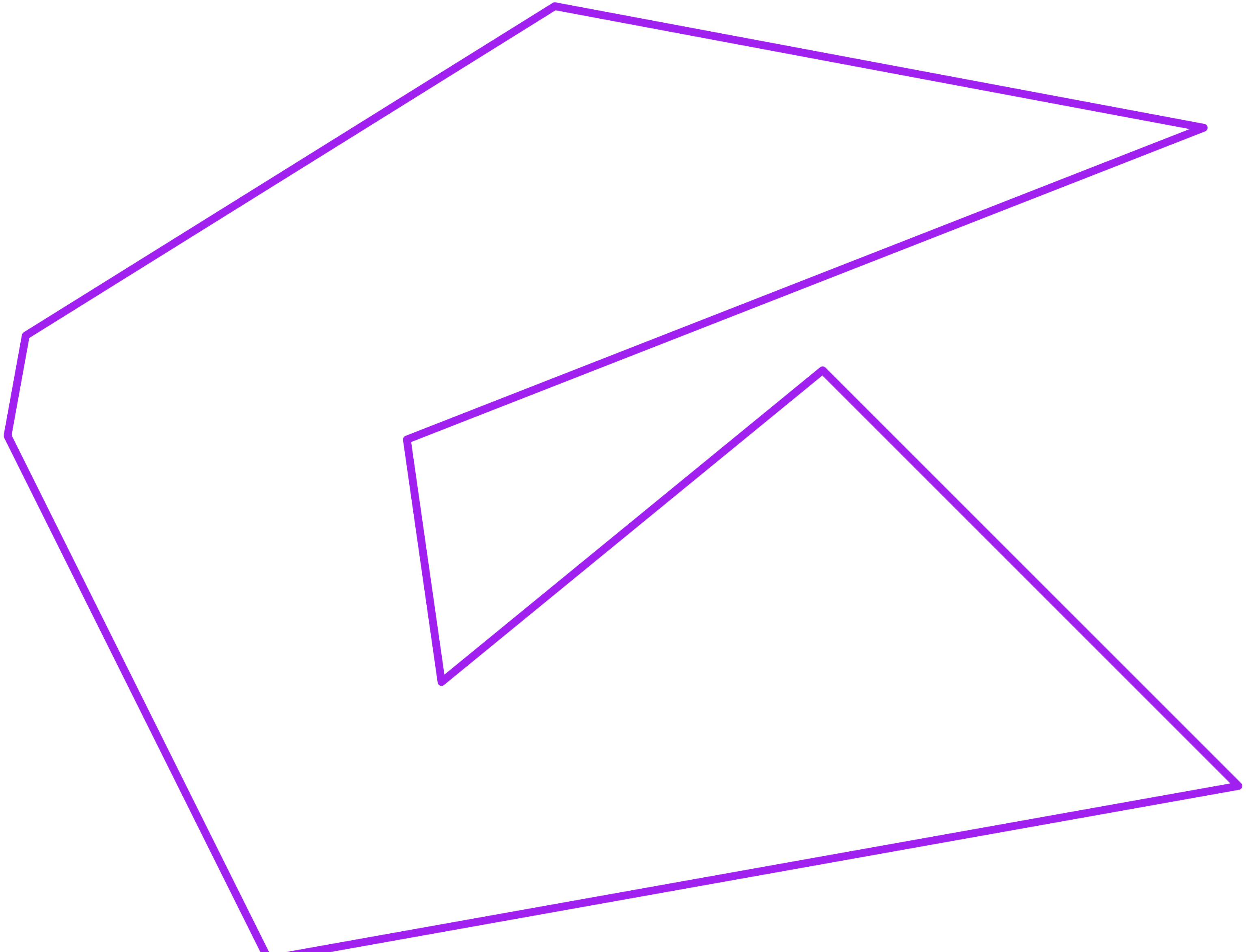
Unstructured mesh

- Use Delaunay (good angles property)
- Add vertices

# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

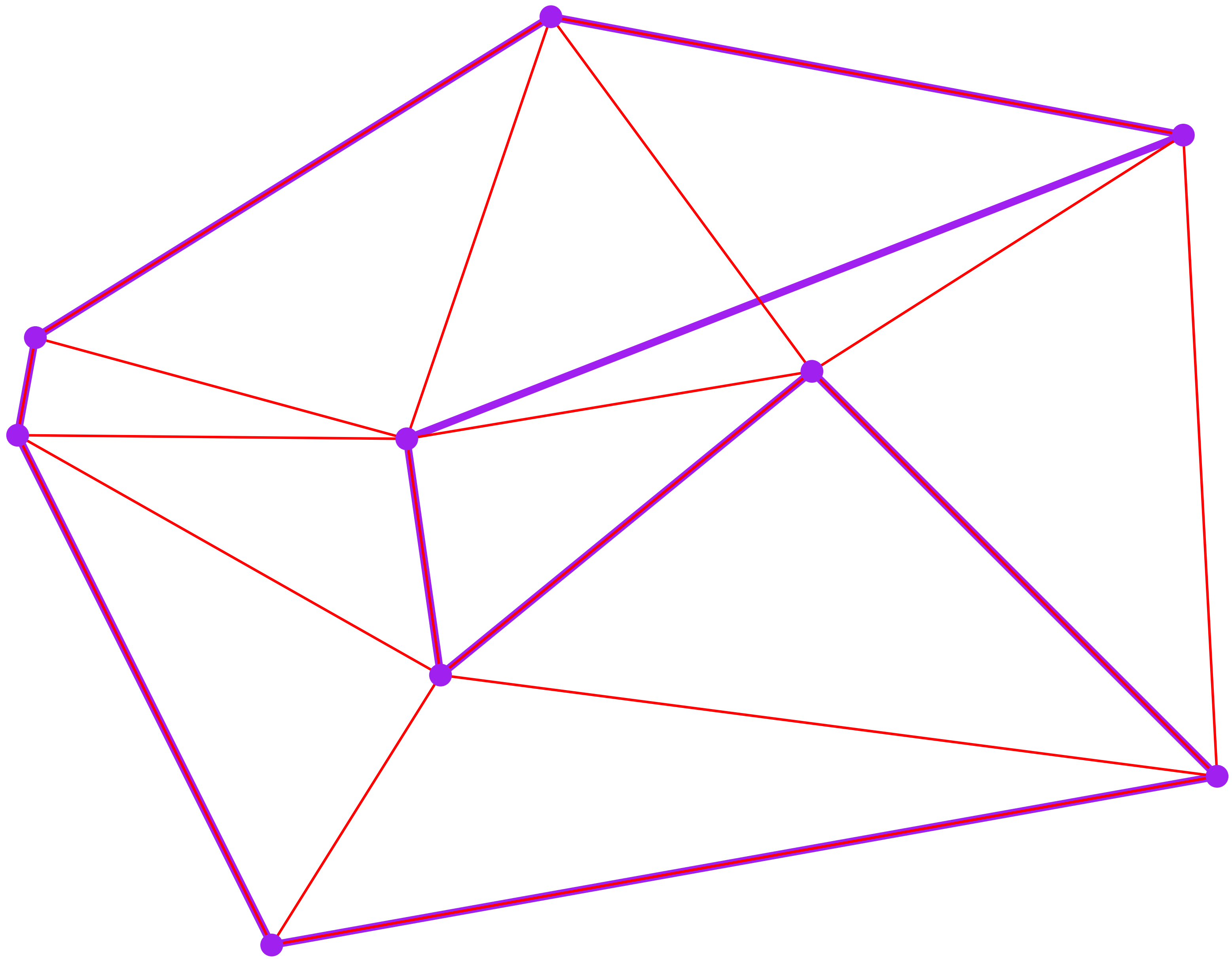


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay

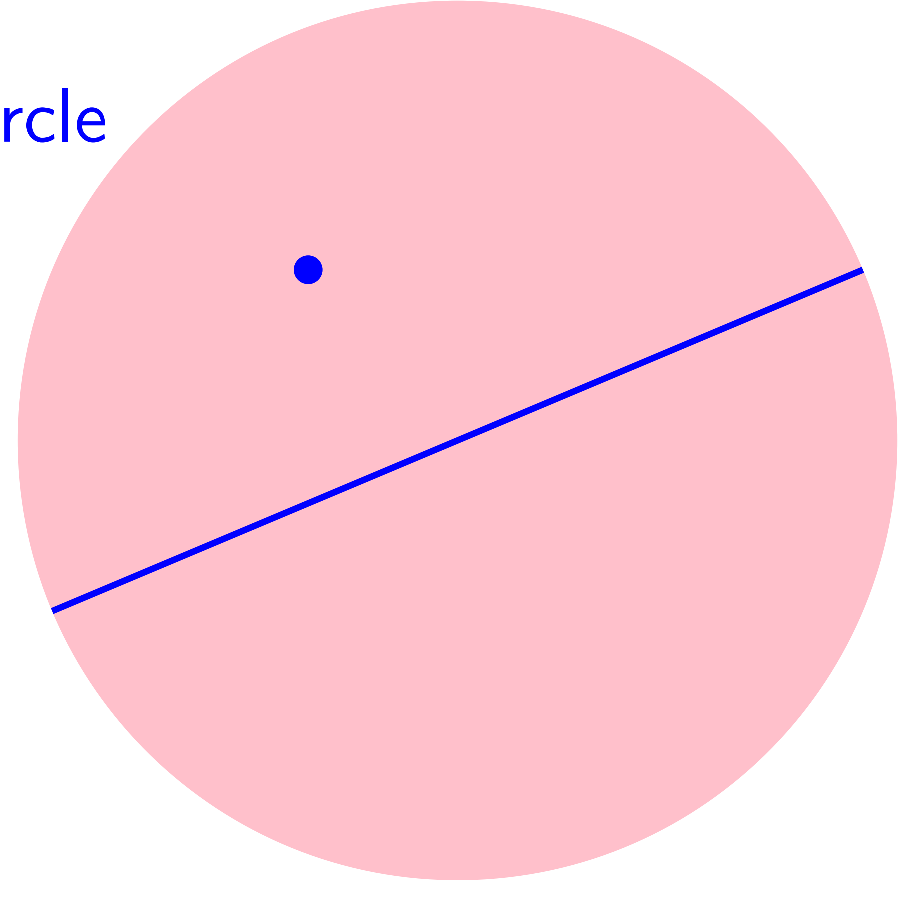


# Meshing

Delaunay mesh refinement [Ruppert]

Def: Edge encroached by vertex

if inside diametral circle



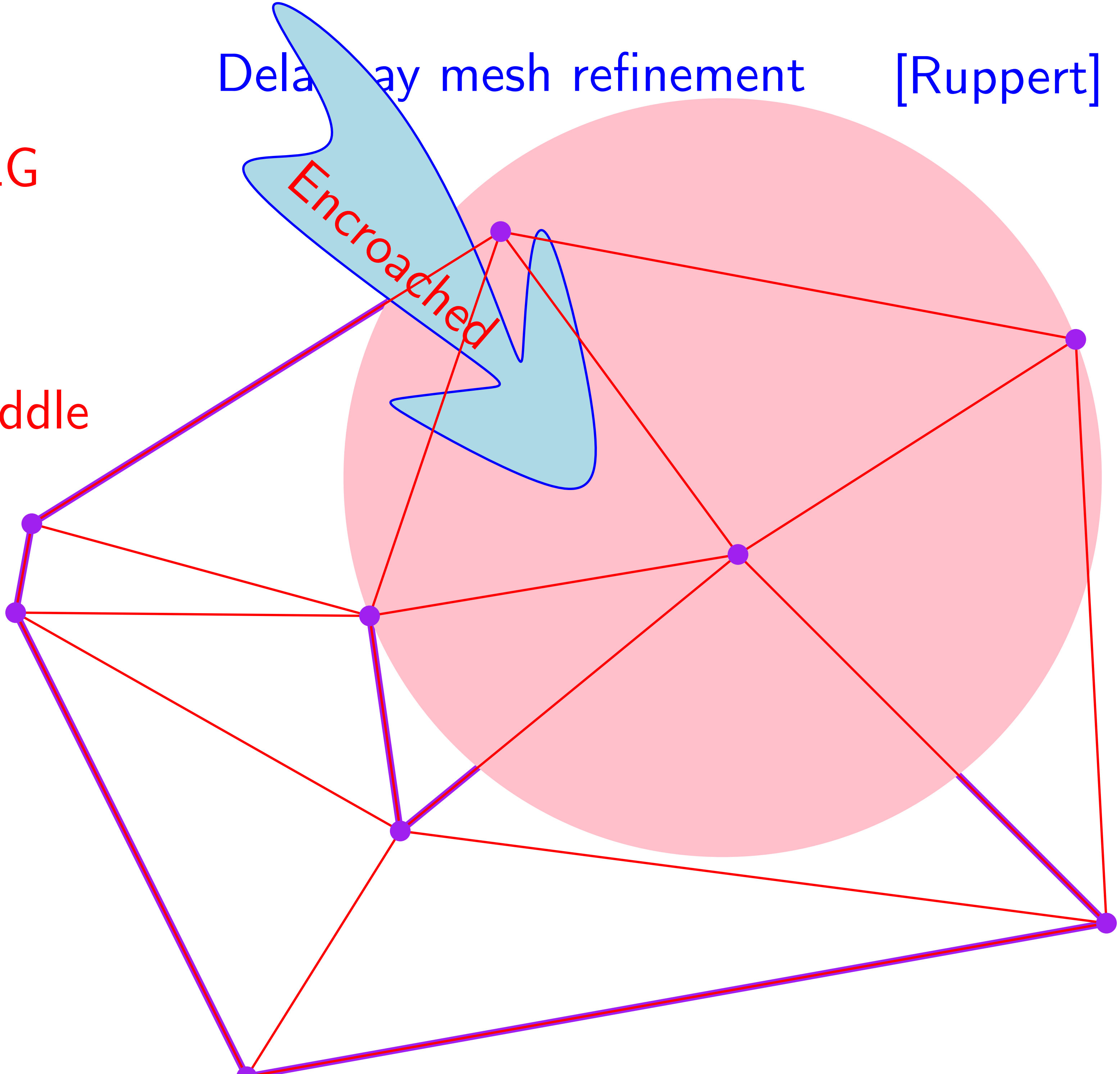
# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay

Split at middle

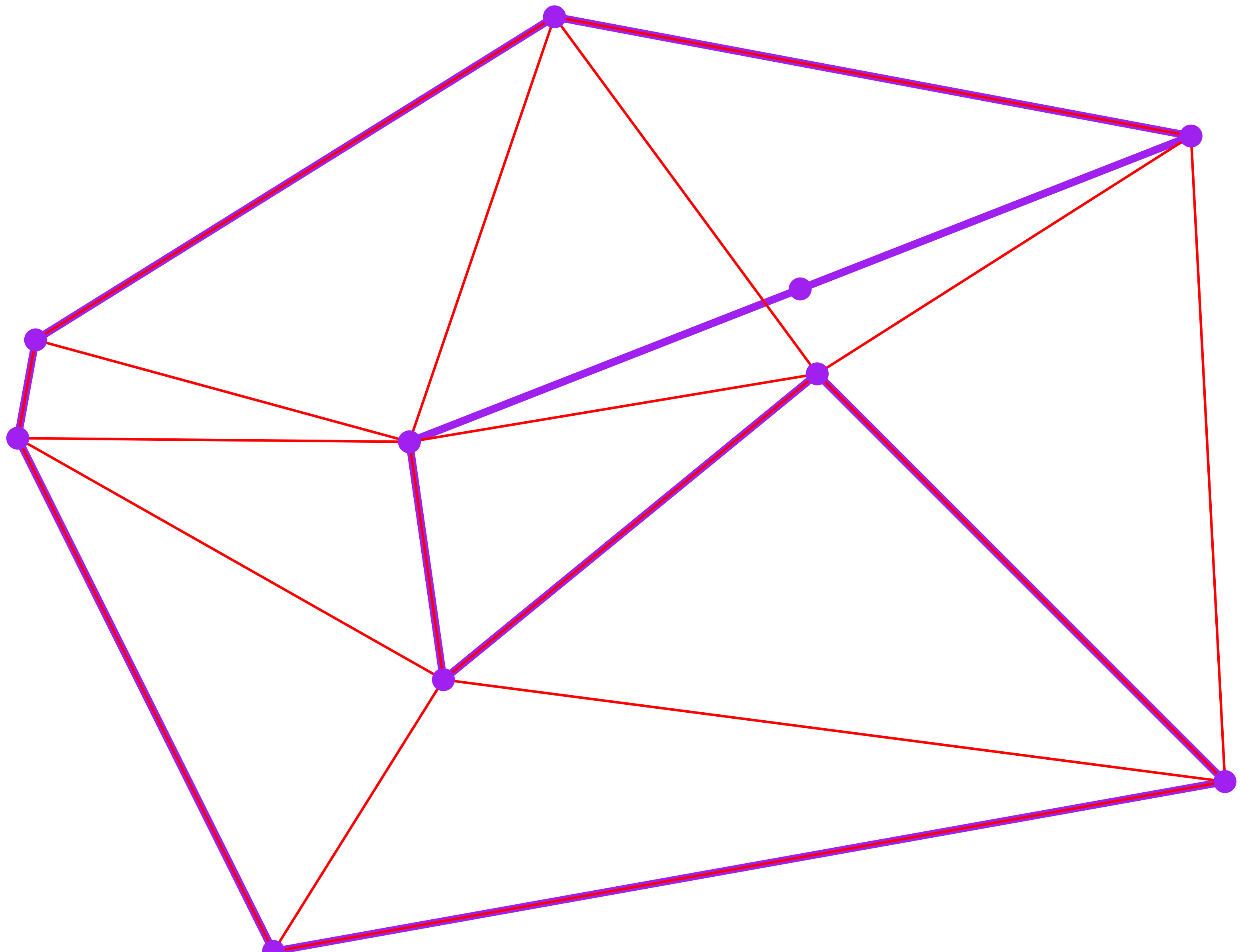


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay

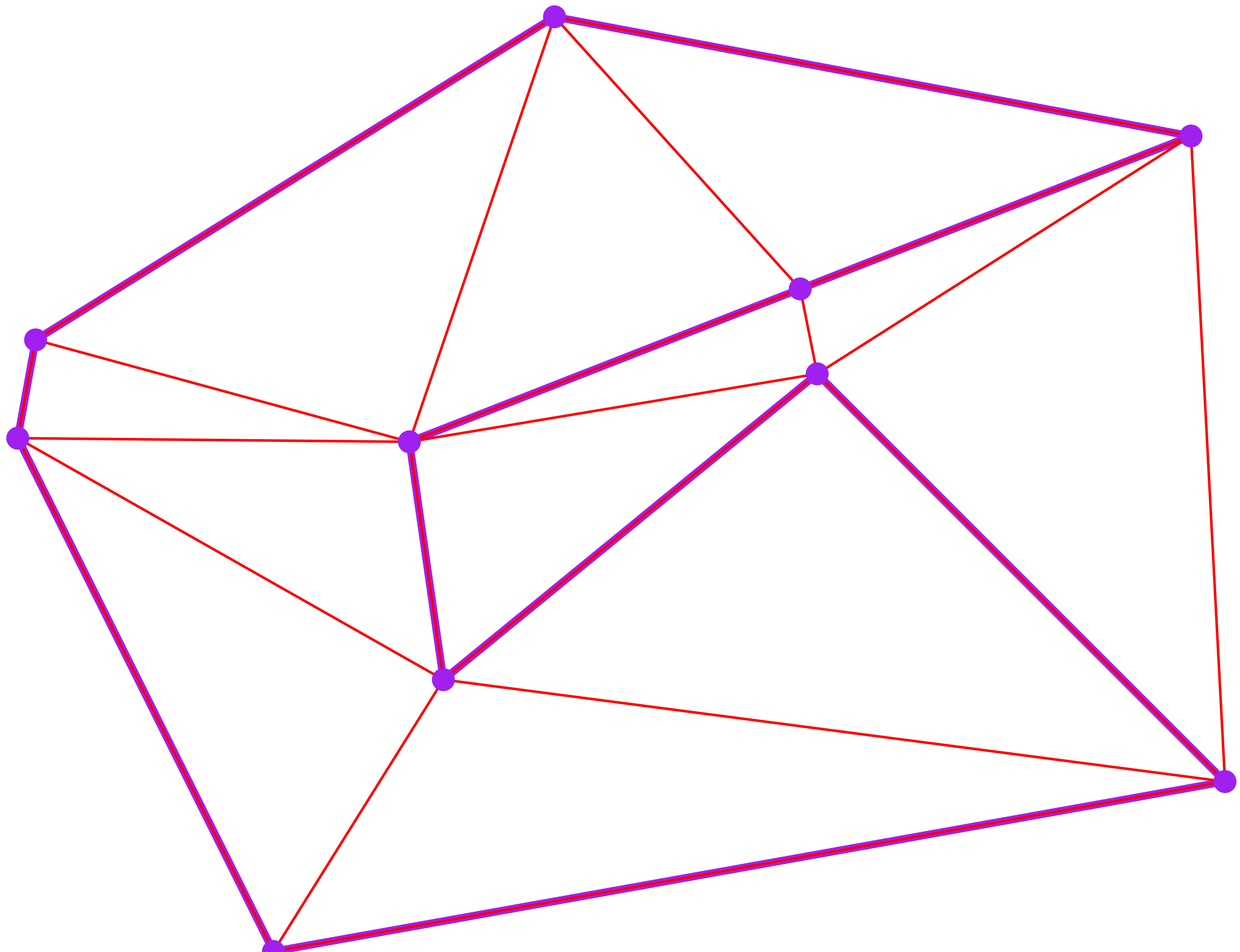


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay

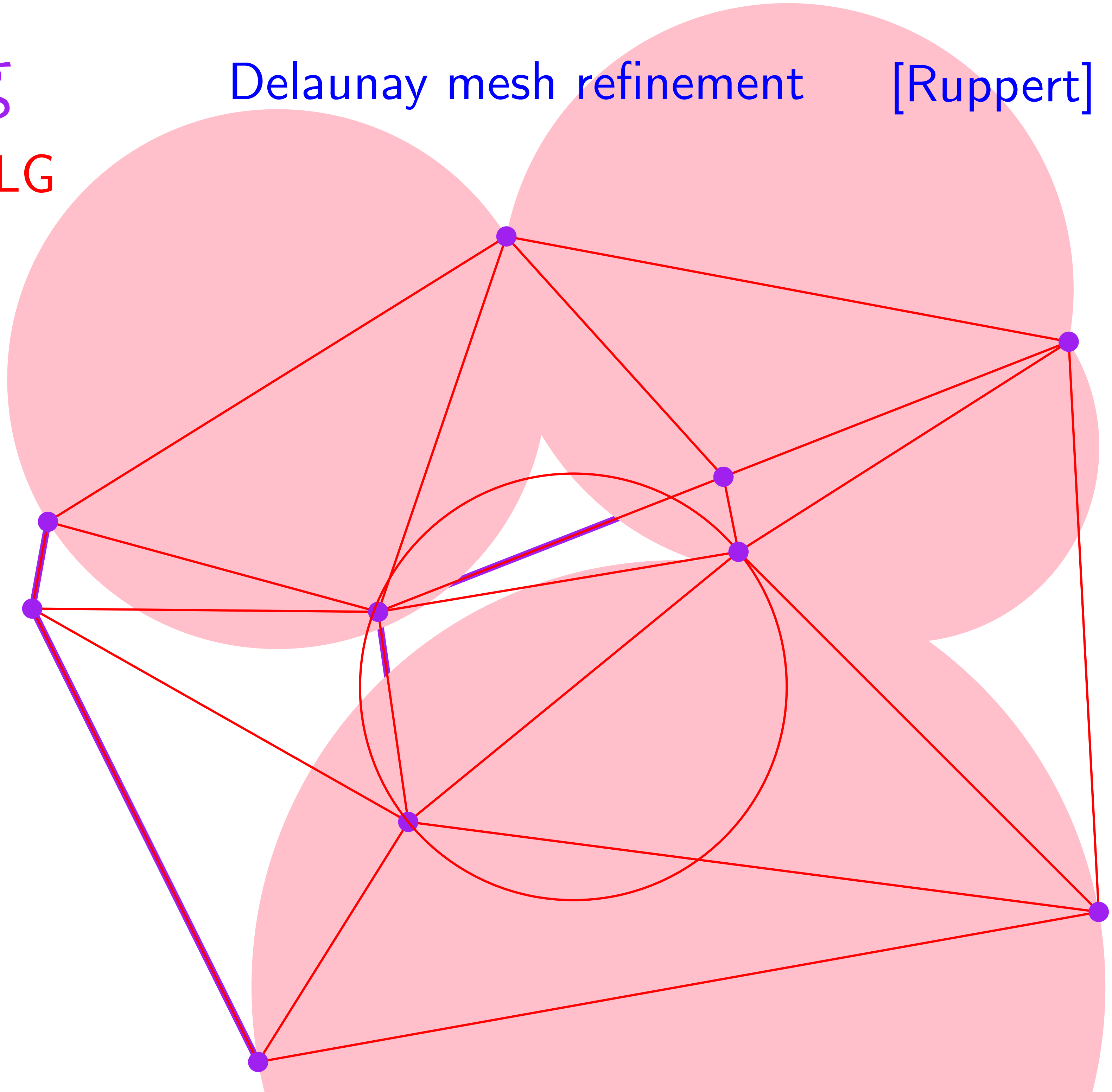


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay

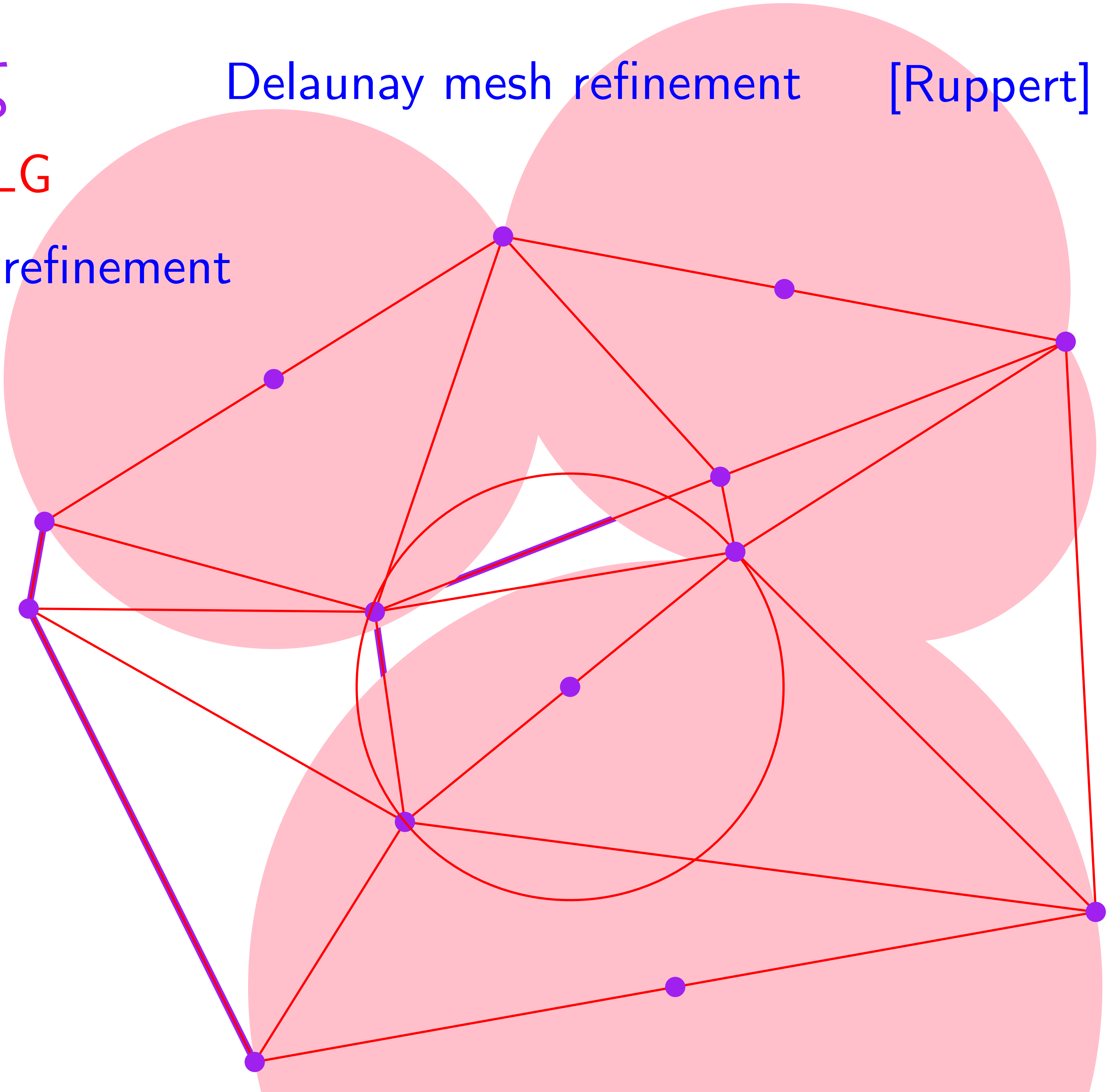


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

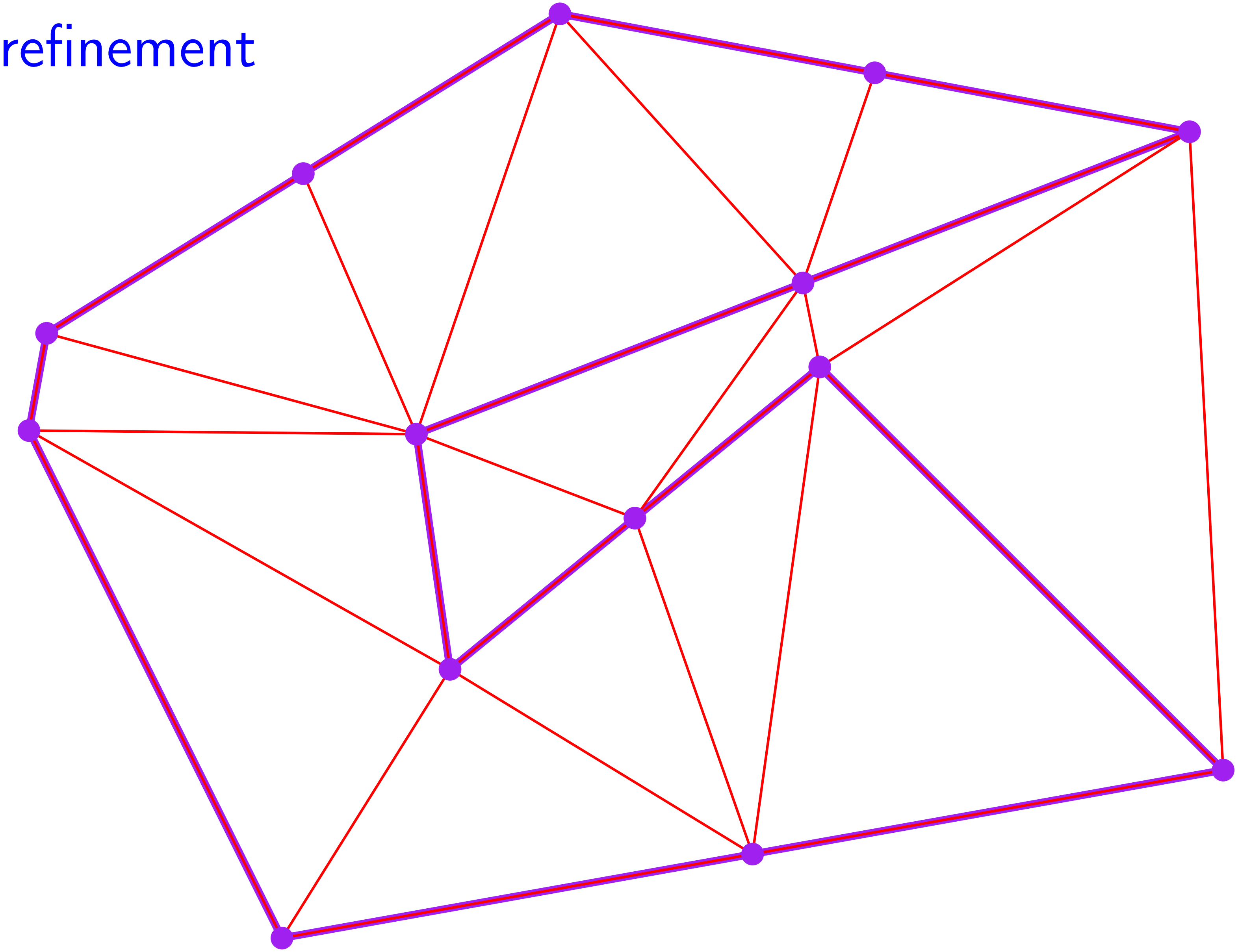


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

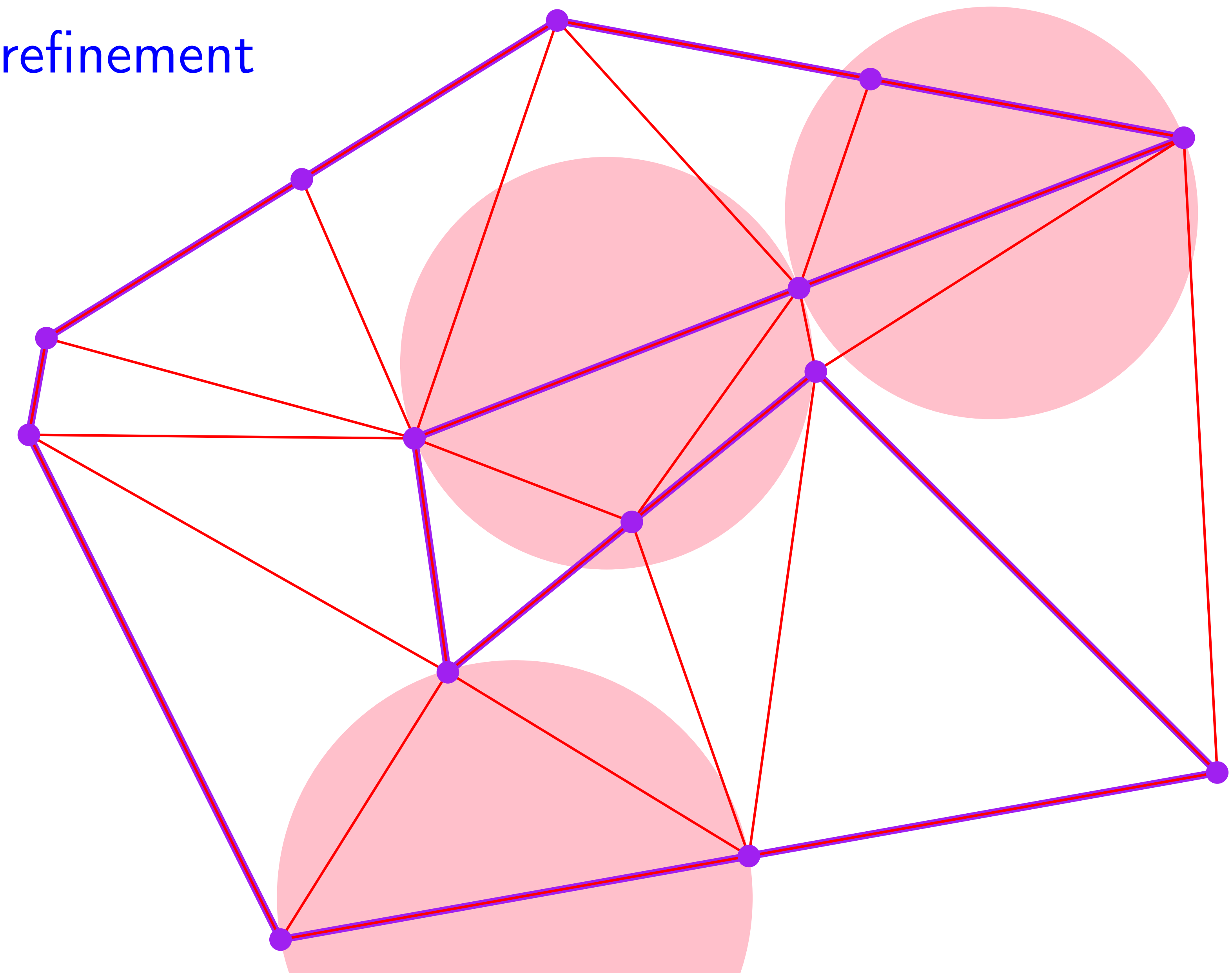


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

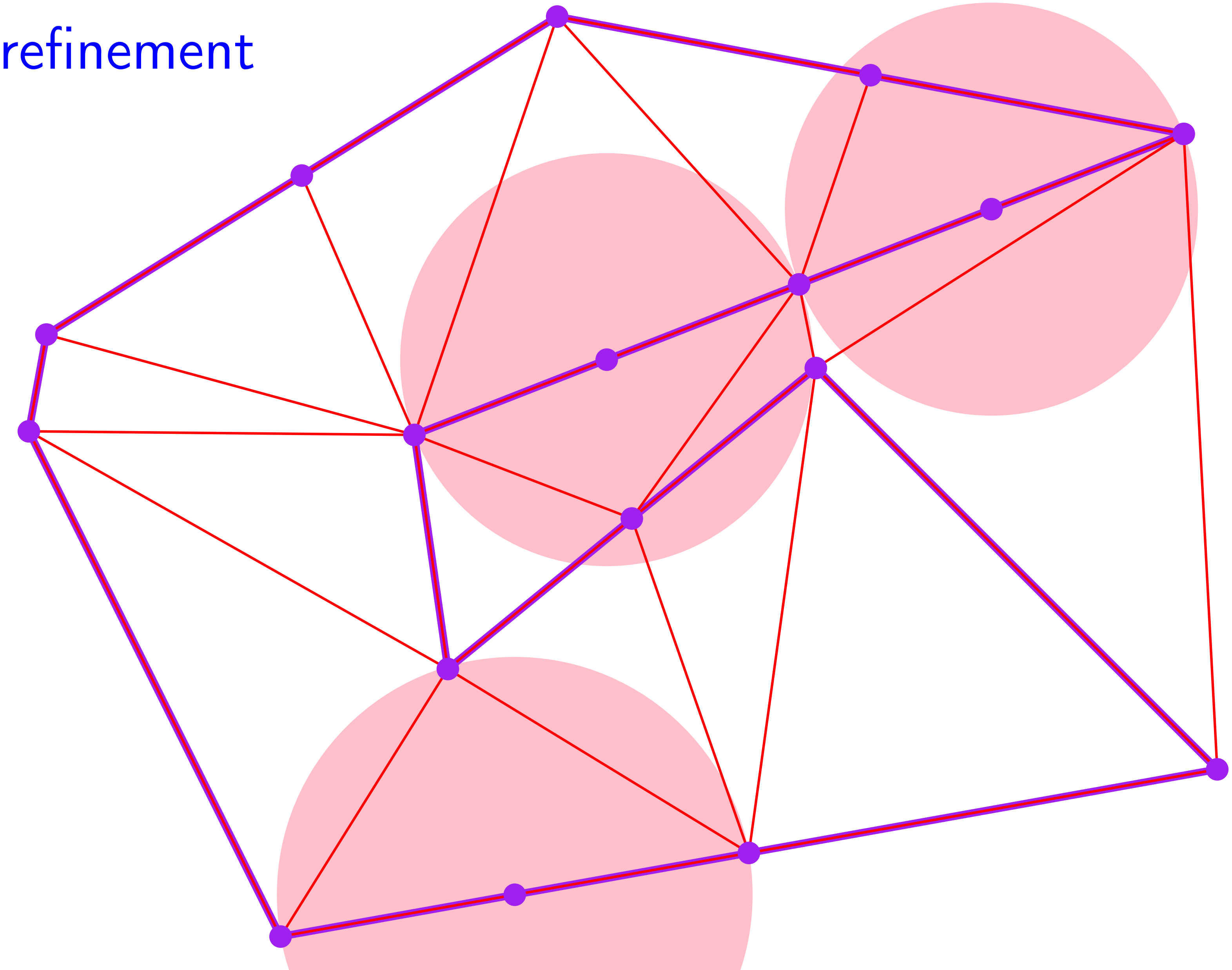


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

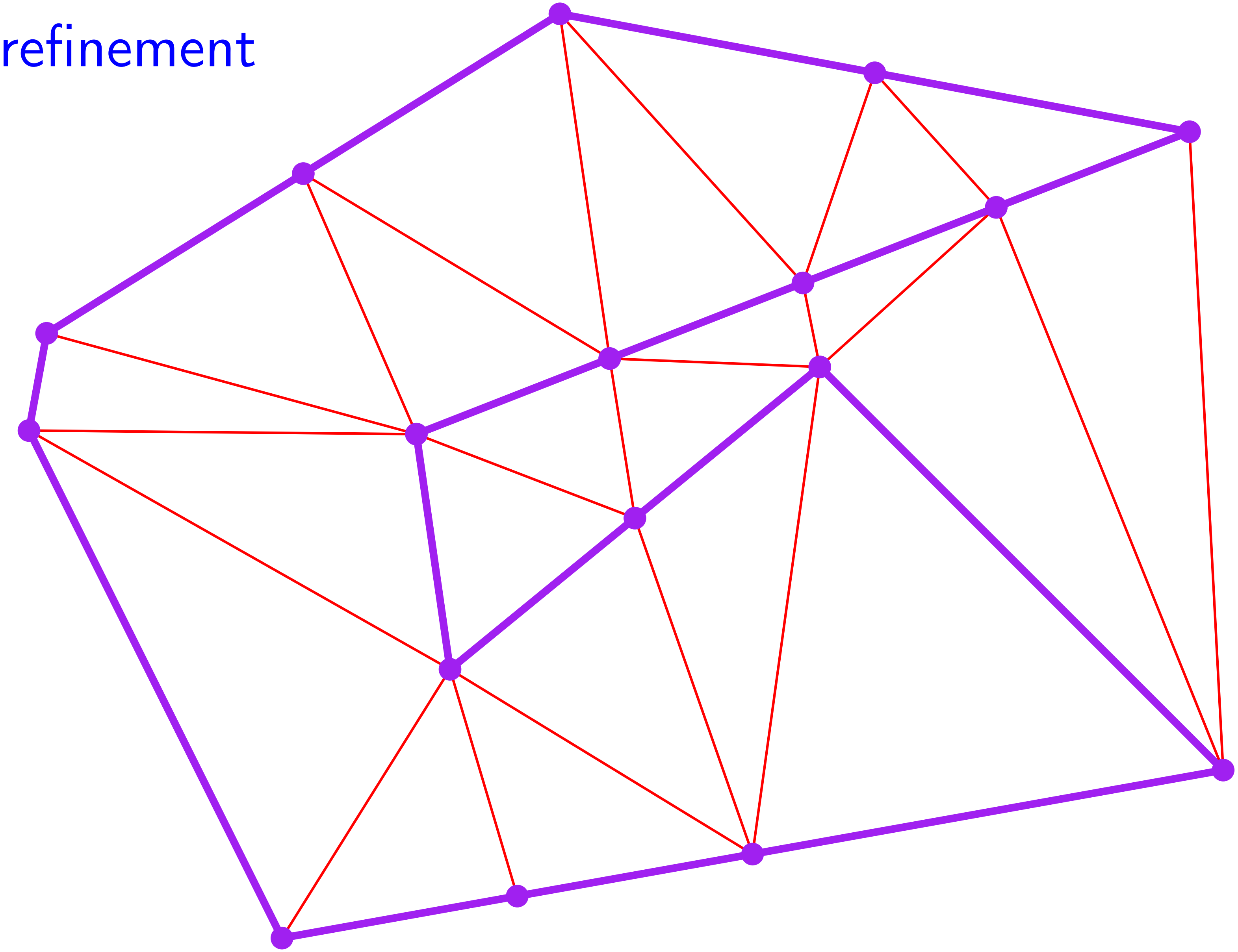


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement



# Meshing

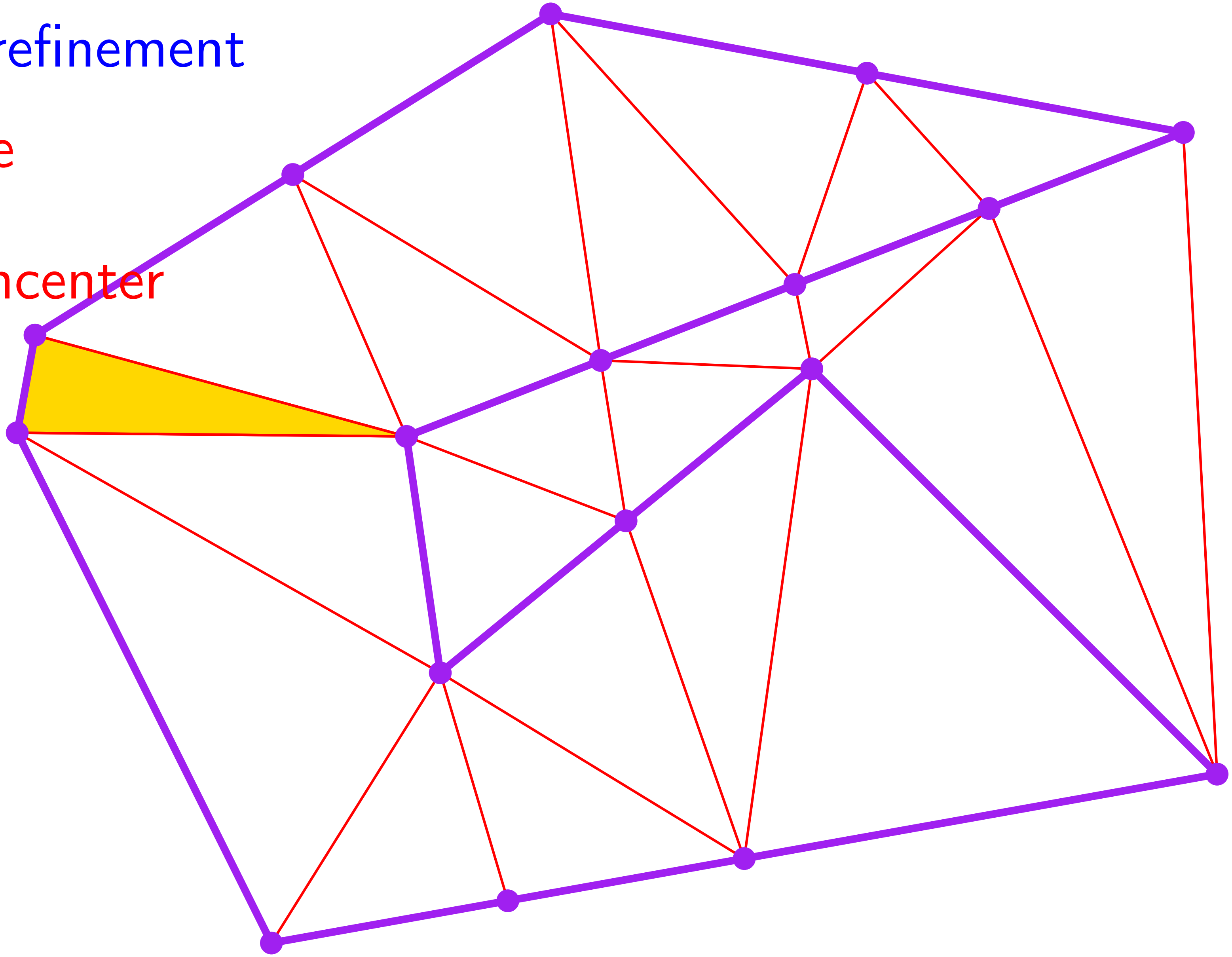
Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

Small angle

Add circumcenter



# Meshing

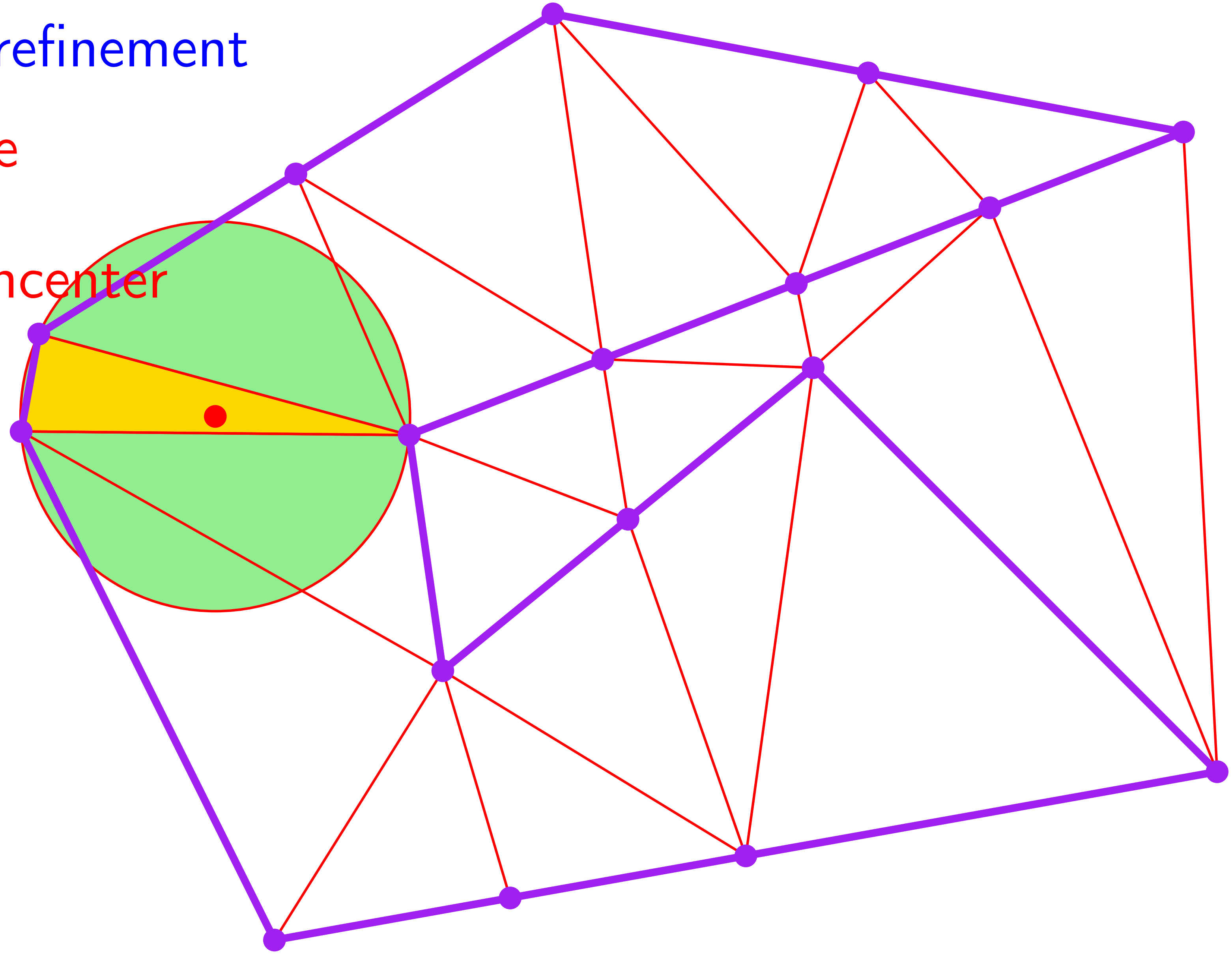
Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

Small angle

Add circumcenter

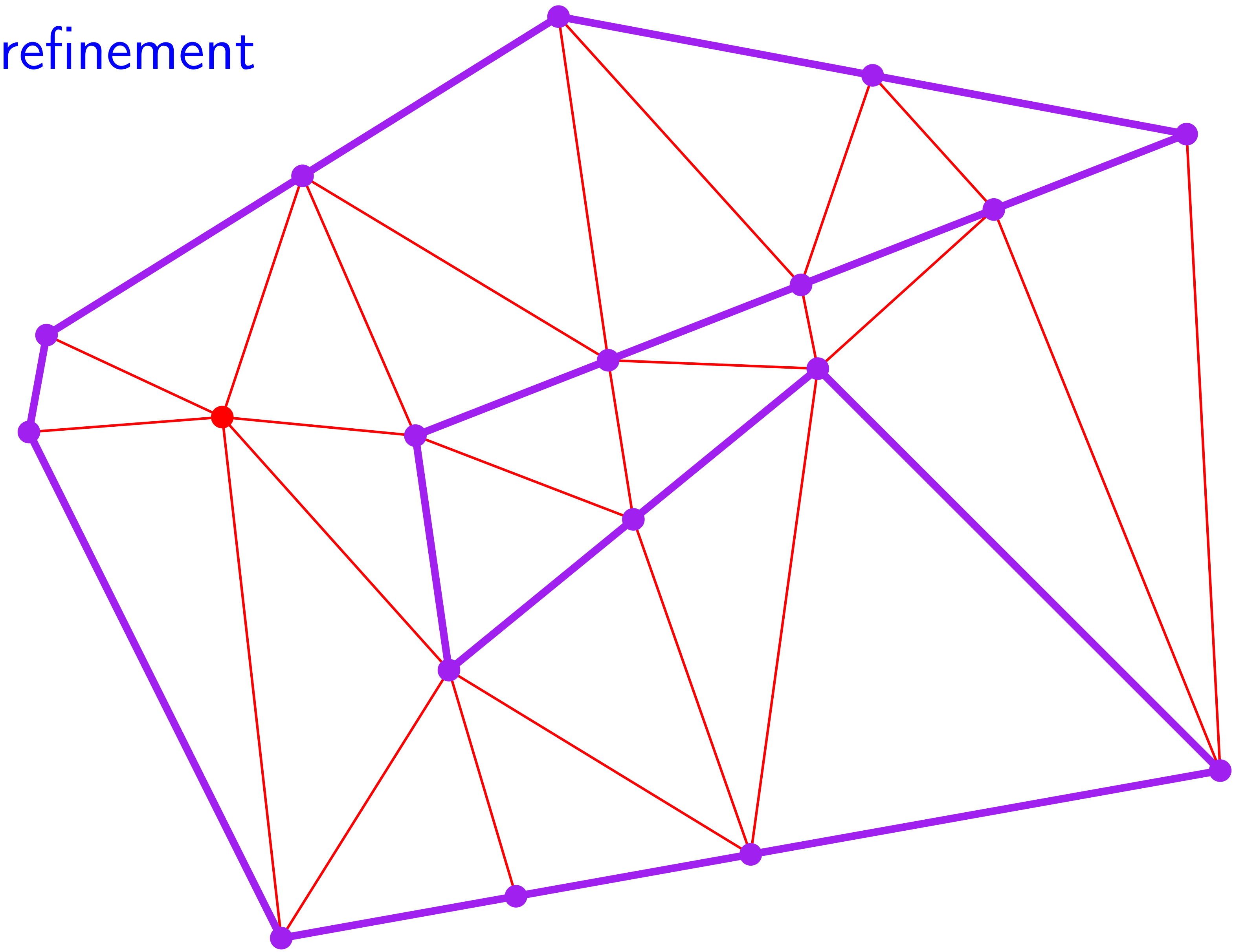


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement



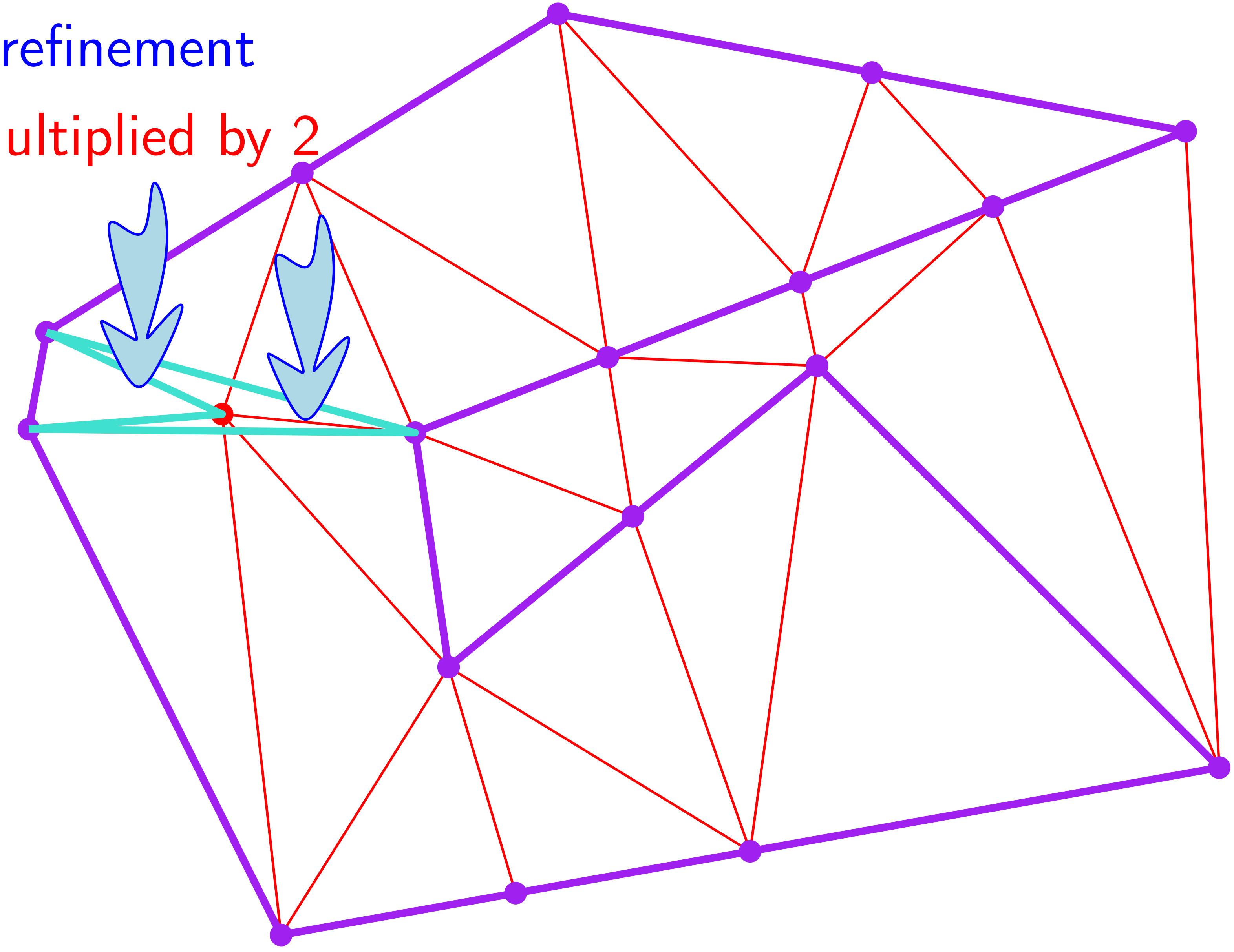
# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

Angle is multiplied by 2

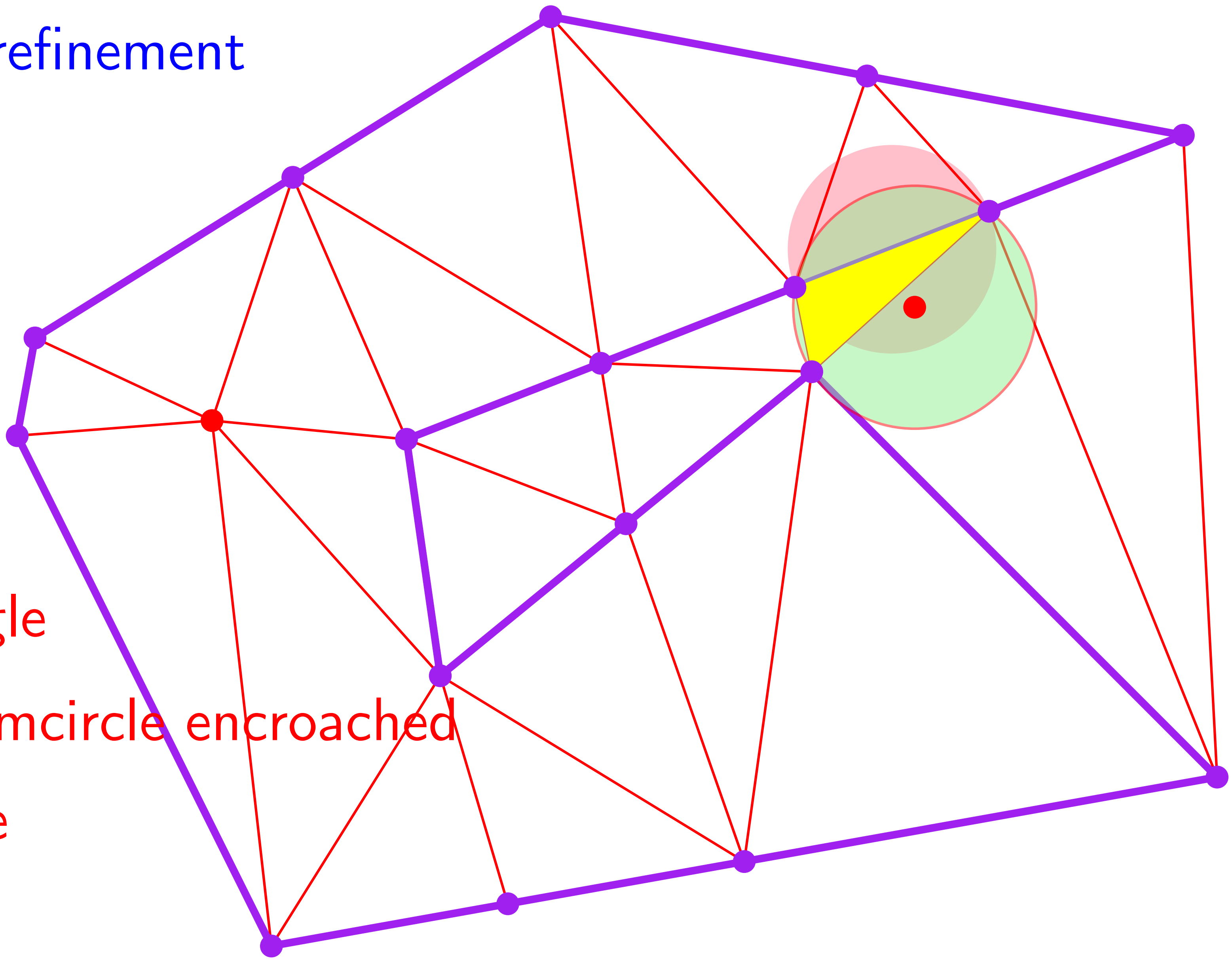


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement



Small angle

But circumcircle encroached

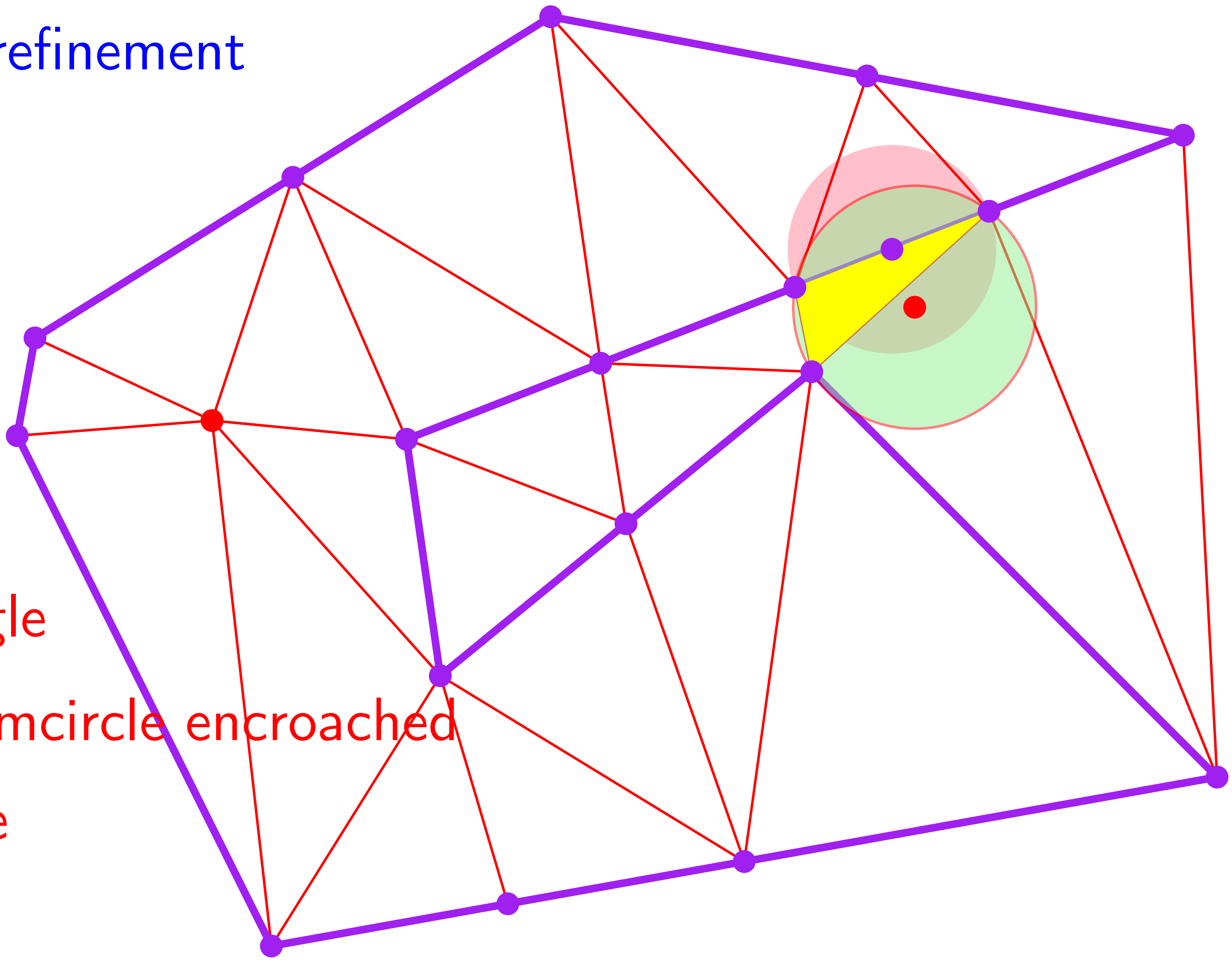
Split edge

# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement



Small angle

But circumcircle encroached

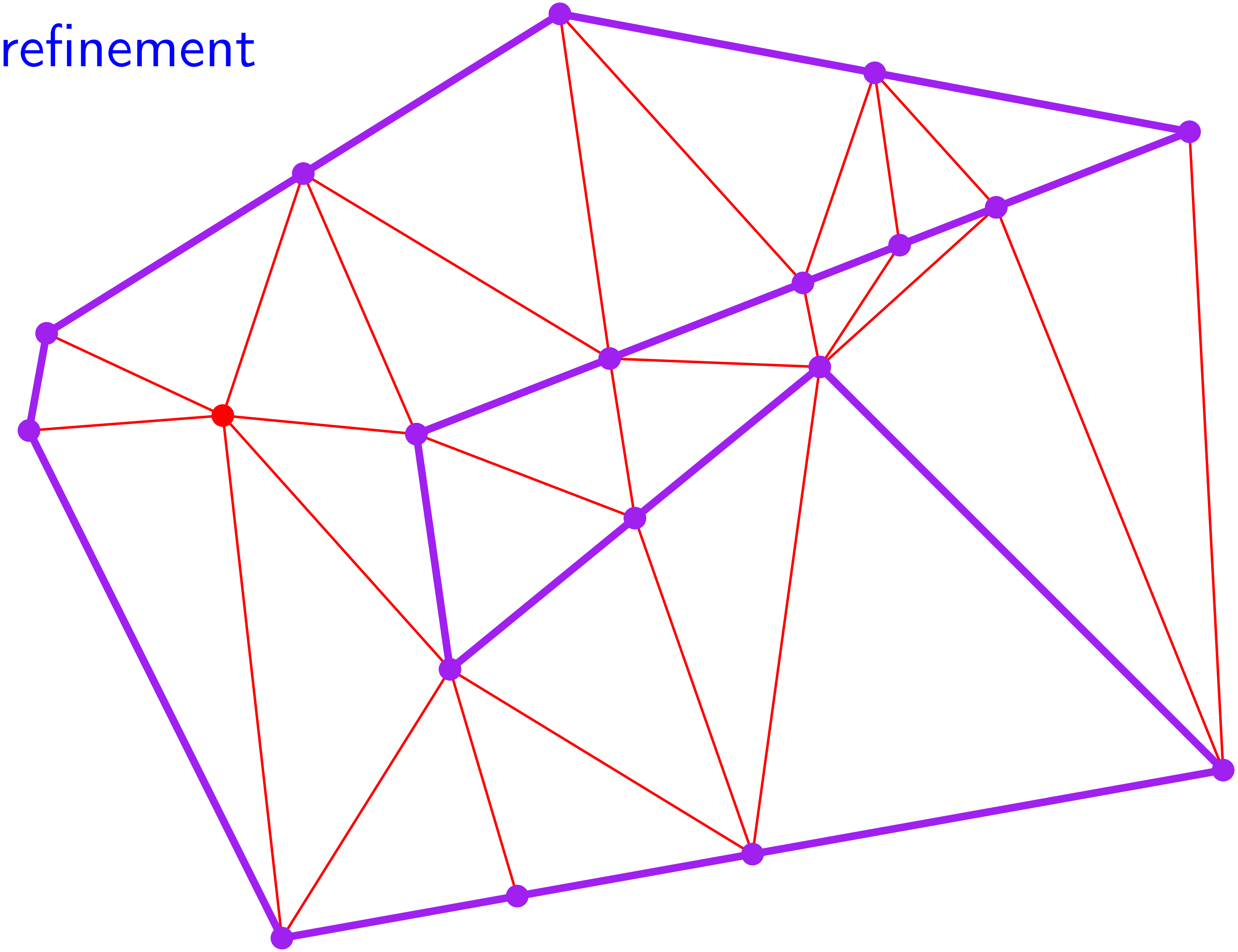
Split edge

# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

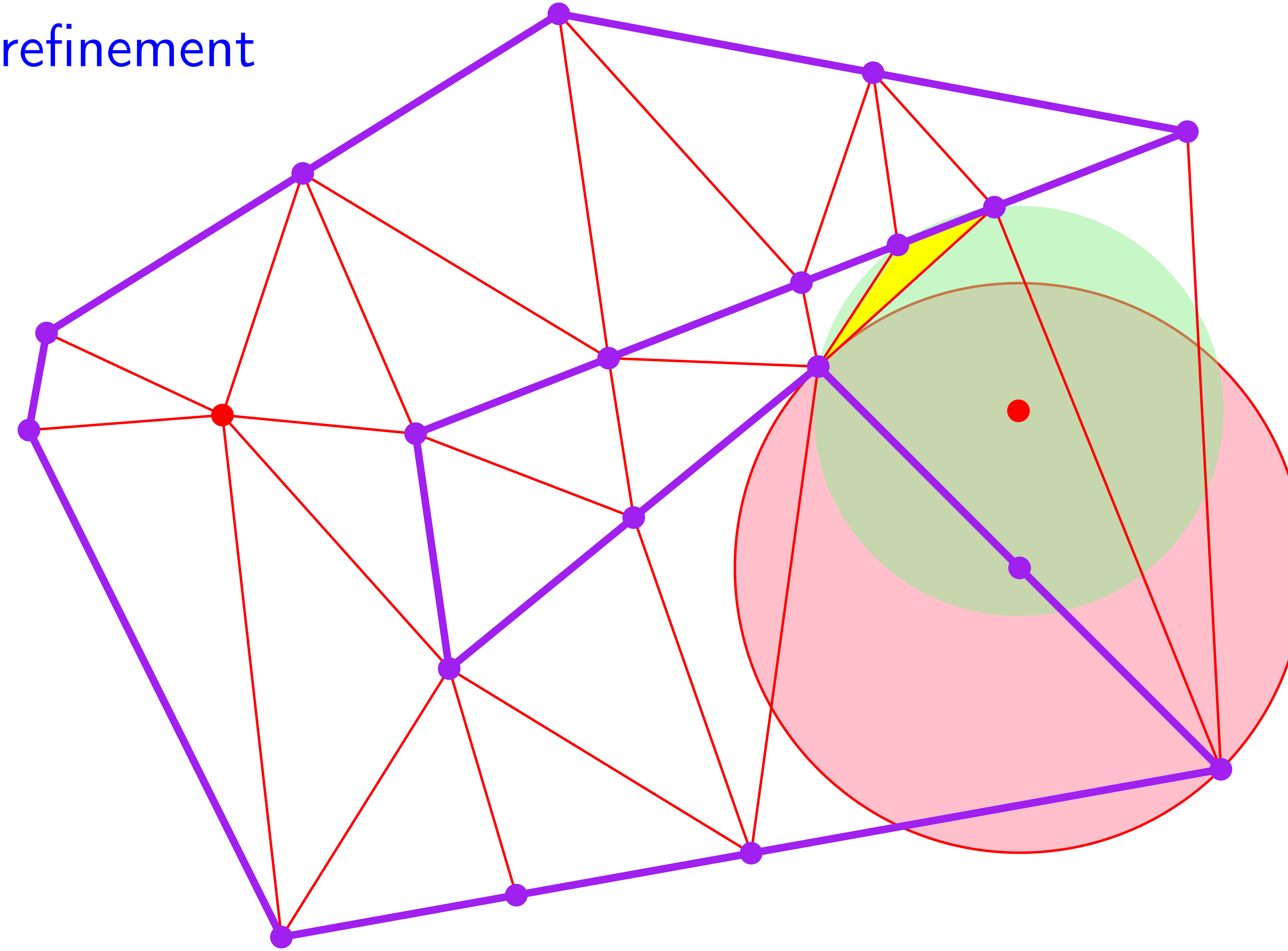


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

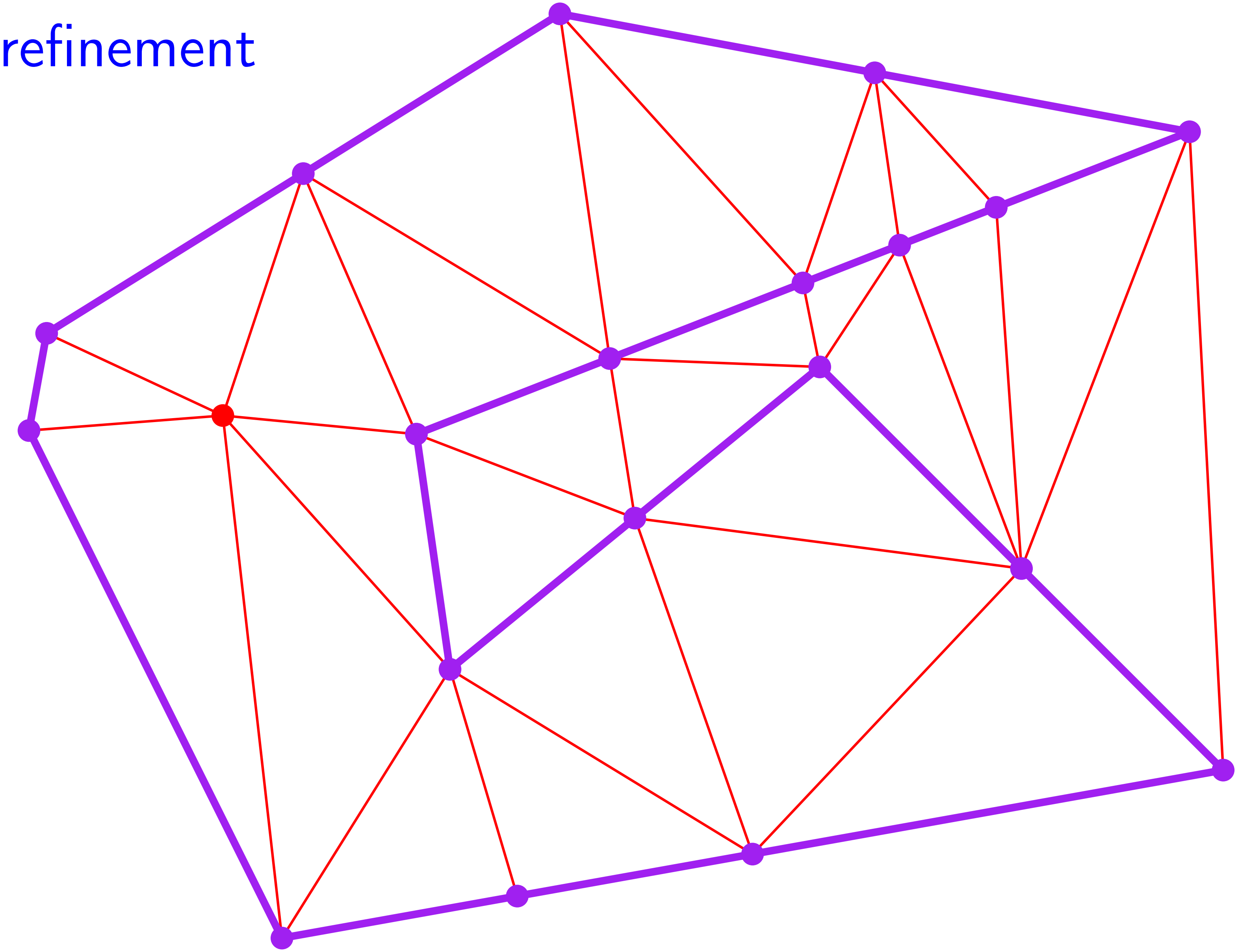


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

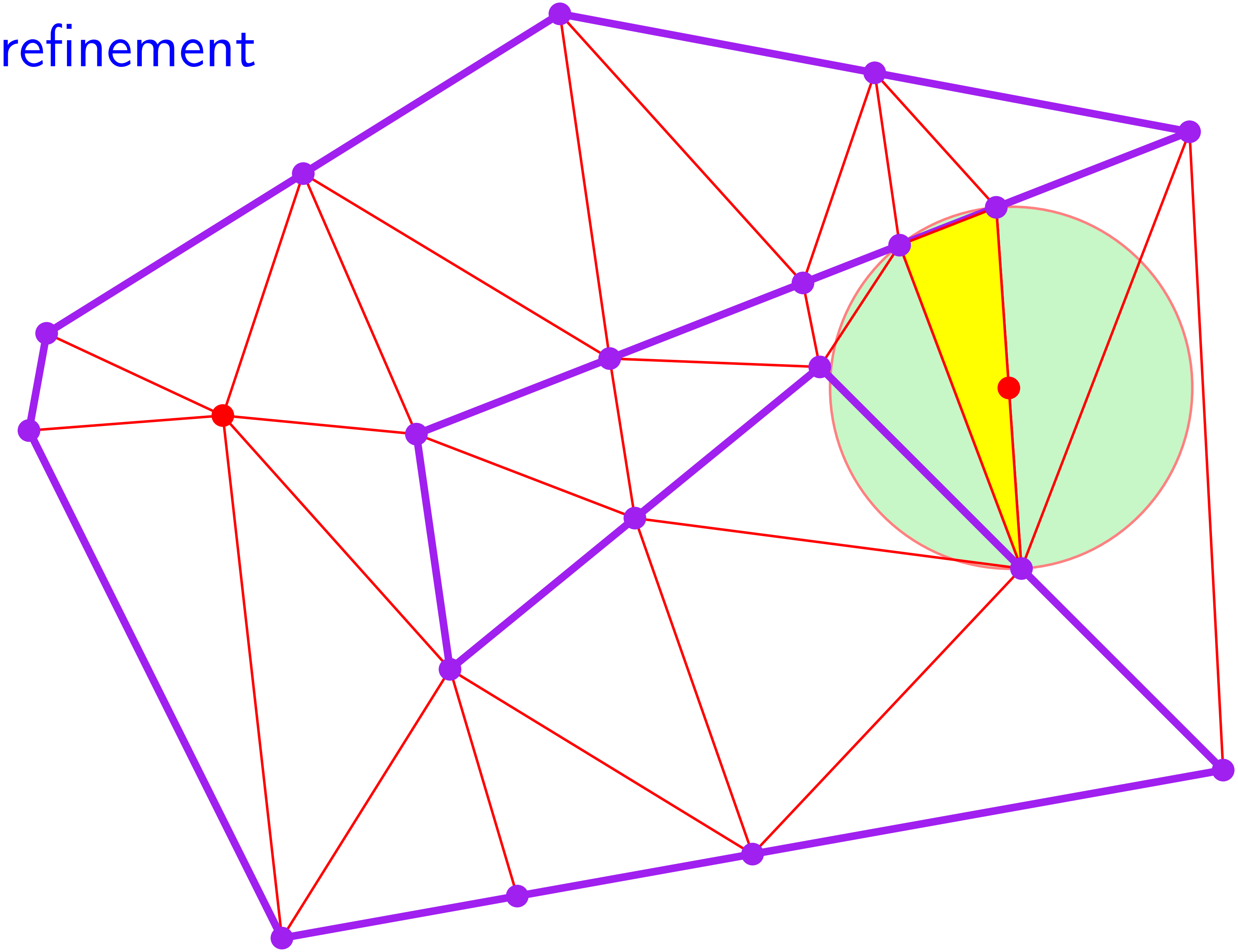


# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement



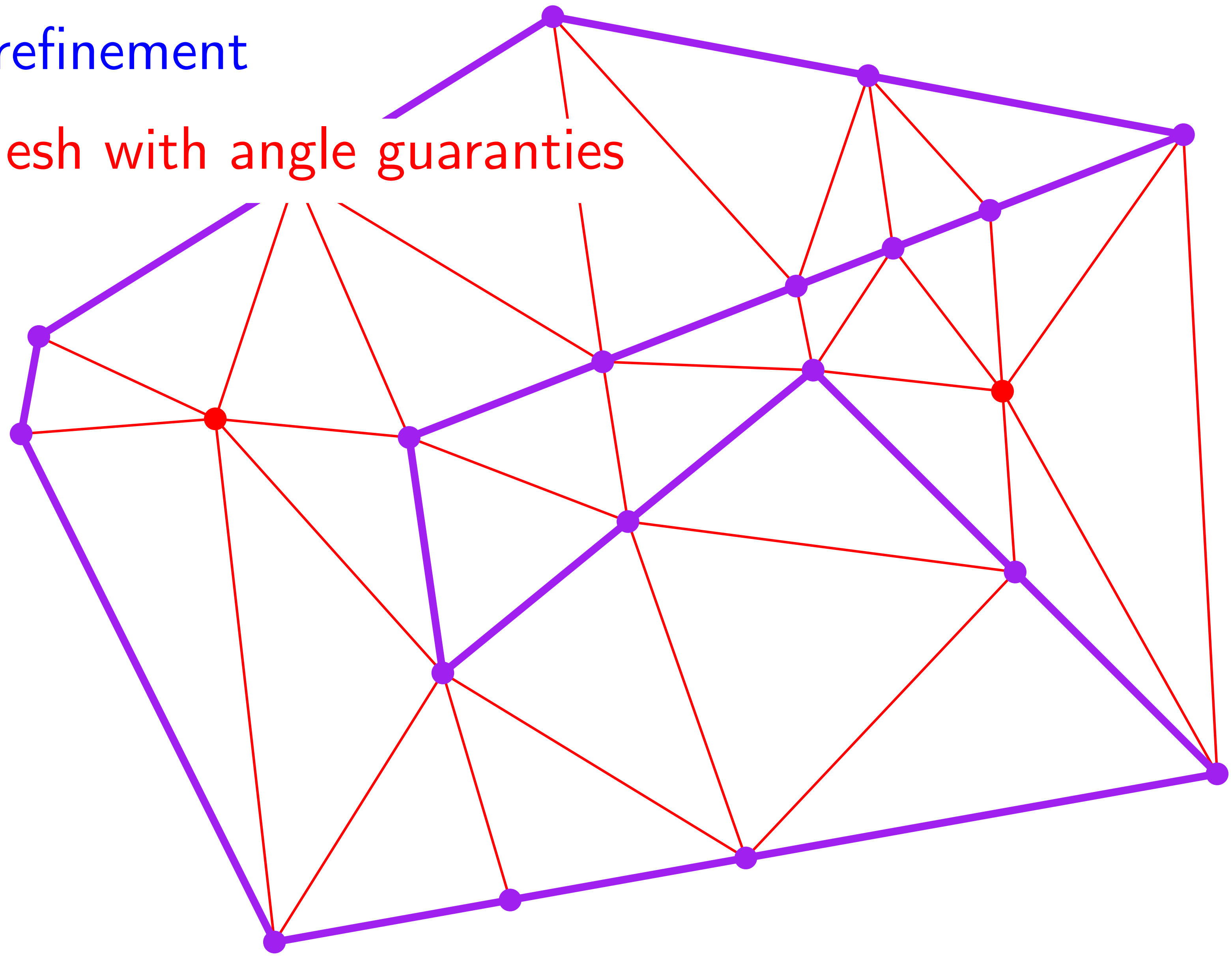
# Meshing

Delaunay mesh refinement [Ruppert]

Input: PSLG

Delaunay refinement

Output: Mesh with angle guaranties



# Meshing

## Delaunay mesh refinement [Ruppert]

Small angles means $< \alpha < 20^\circ$

Theorem: algorithm terminates with mesh of size $O$(optimal)

# Meshing

Delaunay mesh refinement [Ruppert]

lfs: $\mathbb{R}^2 \rightarrow \mathbb{R}$

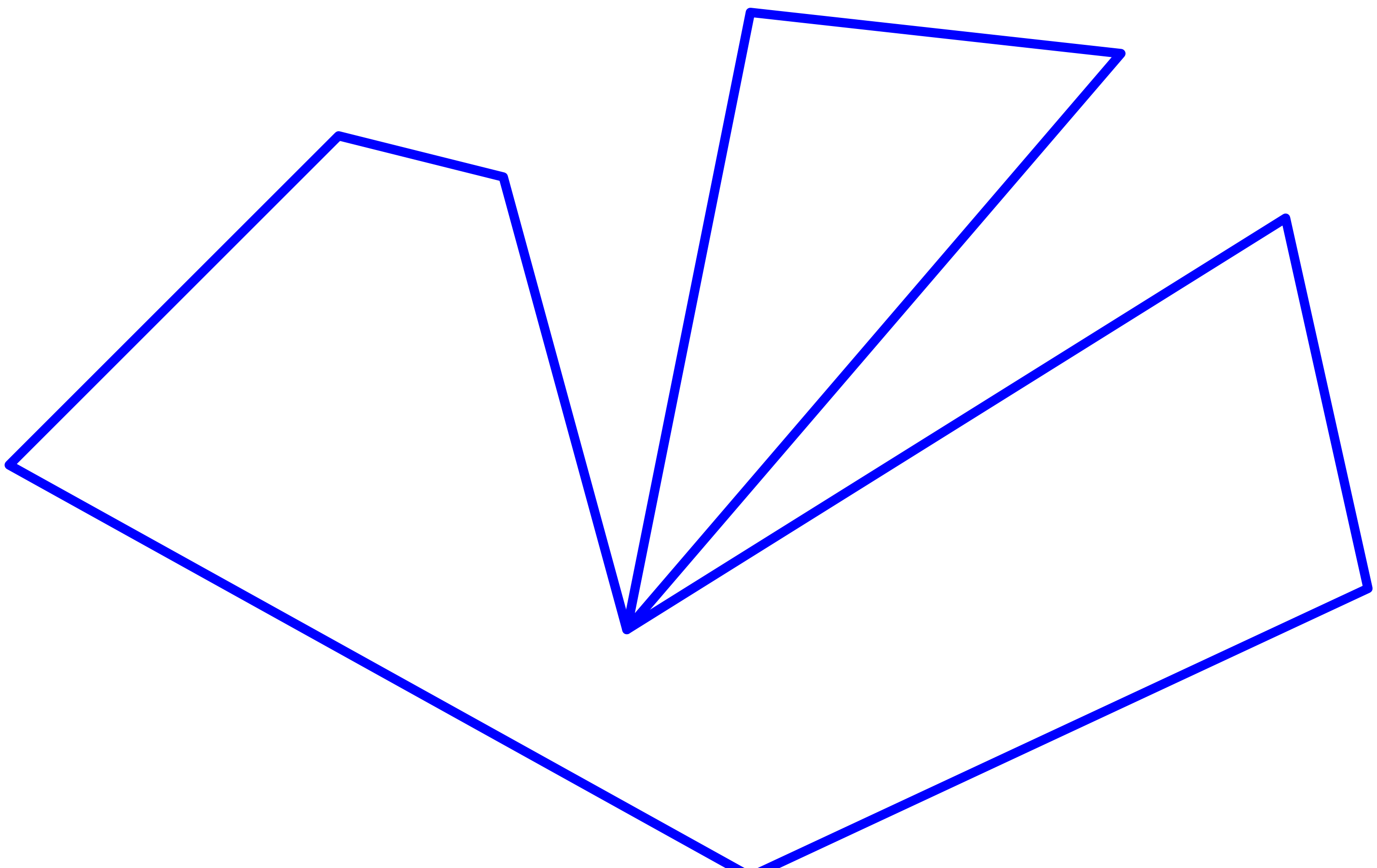
distance to second non incident segment

# Meshing

Delaunay mesh refinement [Ruppert]

lfs: $\mathbb{R}^2 \rightarrow \mathbb{R}$

distance to second non incident segment

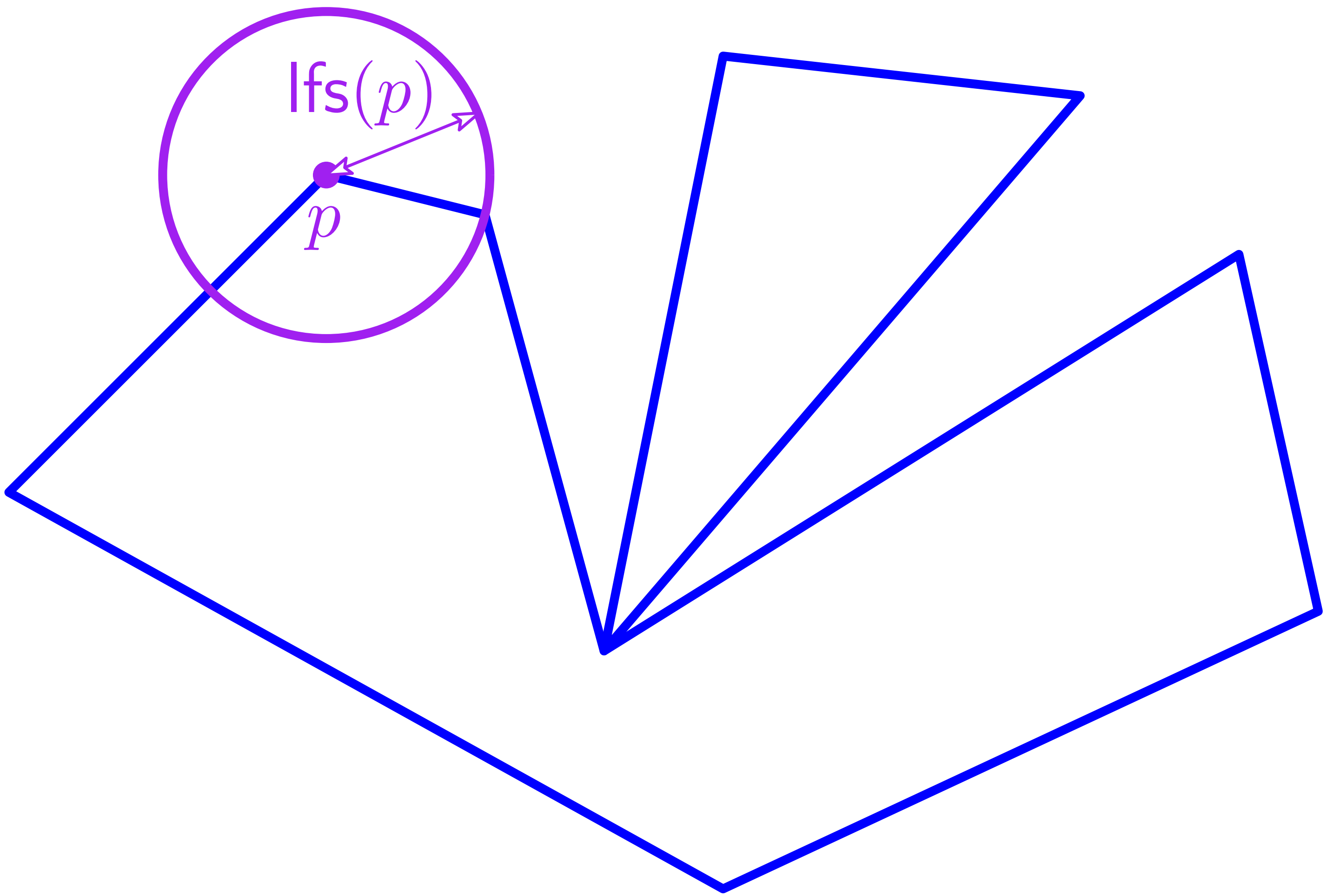


# Meshing

Delaunay mesh refinement [Ruppert]

lfs: $\mathbb{R}^2 \rightarrow \mathbb{R}$

distance to second non incident segment

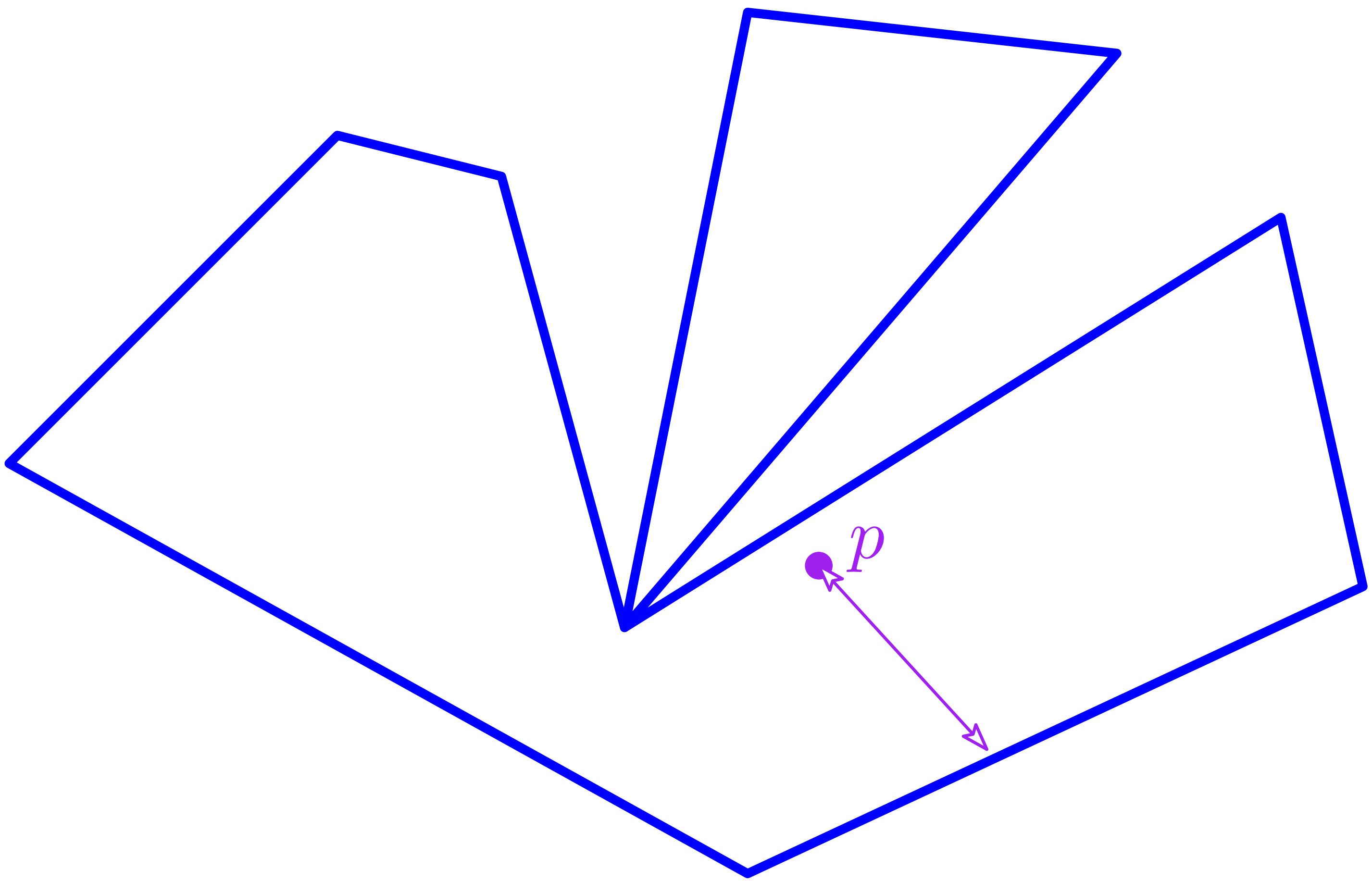


# Meshing

Delaunay mesh refinement [Ruppert]

lfs: $\mathbb{R}^2 \to \mathbb{R}$

distance to second non incident segment

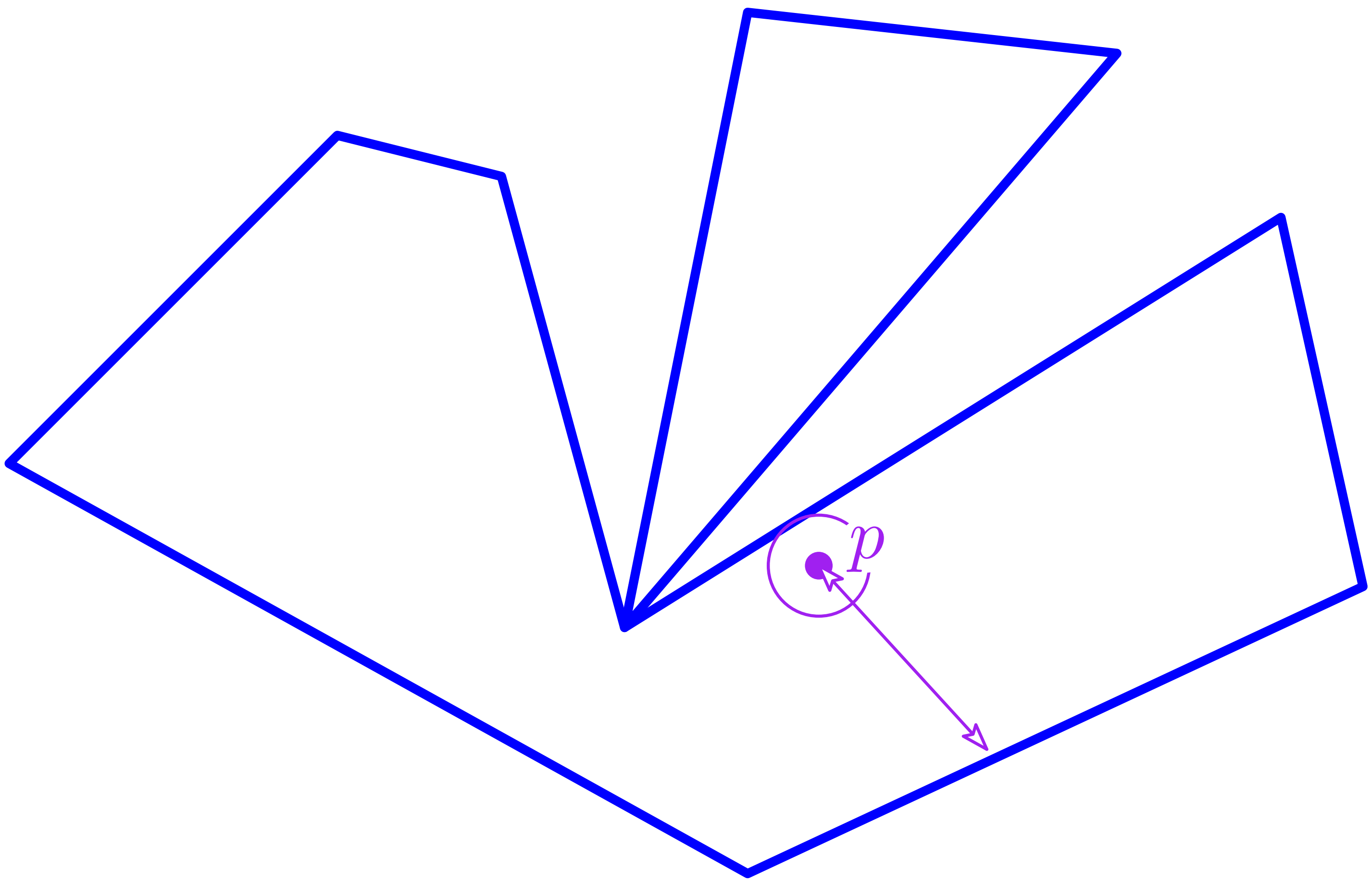


# Meshing

Delaunay mesh refinement [Ruppert]

lfs: $\mathbb{R}^2 \to \mathbb{R}$

distance to second non incident segment

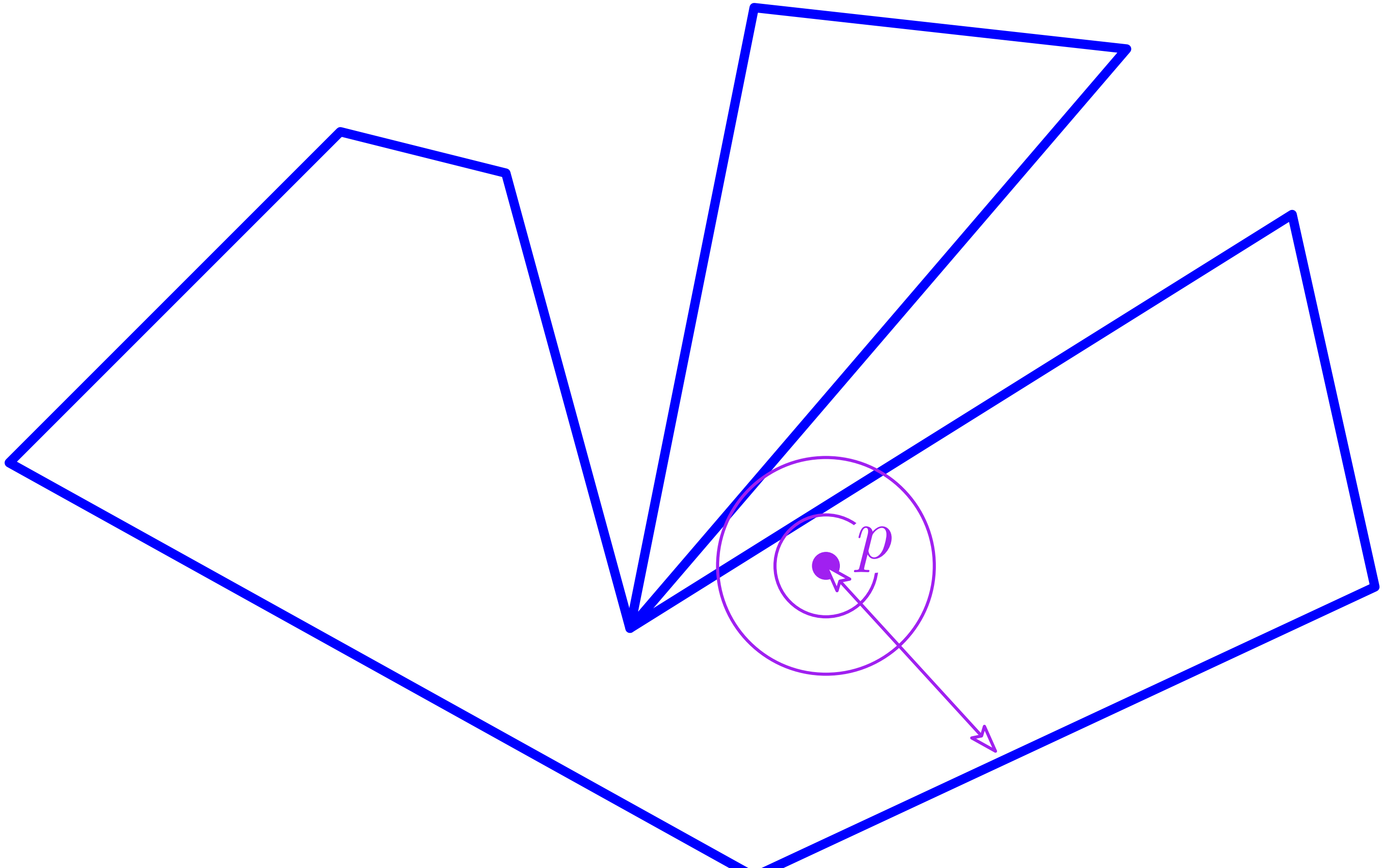


# Meshing

Delaunay mesh refinement [Ruppert]

lfs: $\mathbb{R}^2 \to \mathbb{R}$

distance to second non incident segment

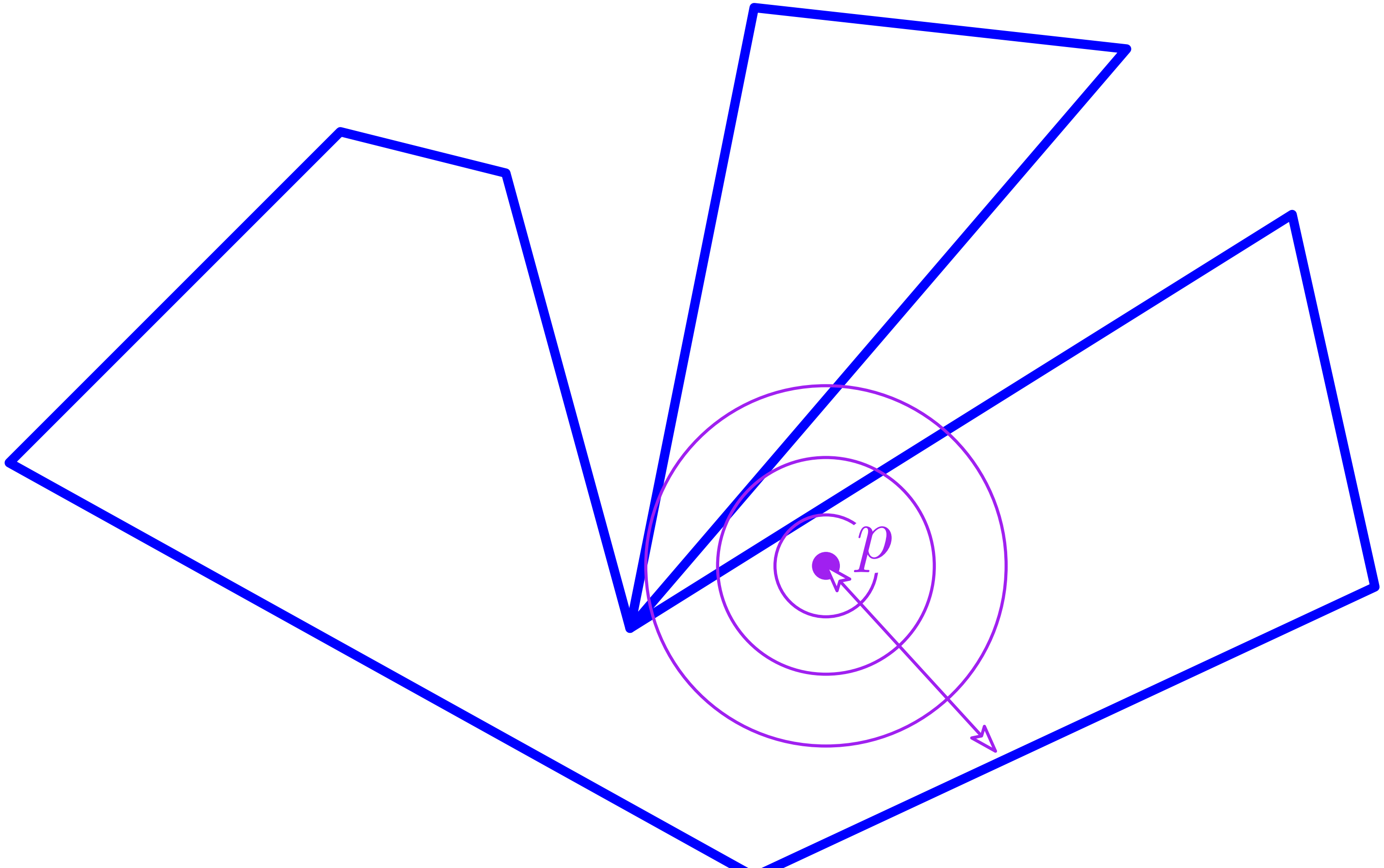


# Meshing

Delaunay mesh refinement [Ruppert]

lfs: $\mathbb{R}^2 \to \mathbb{R}$

distance to second non incident segment

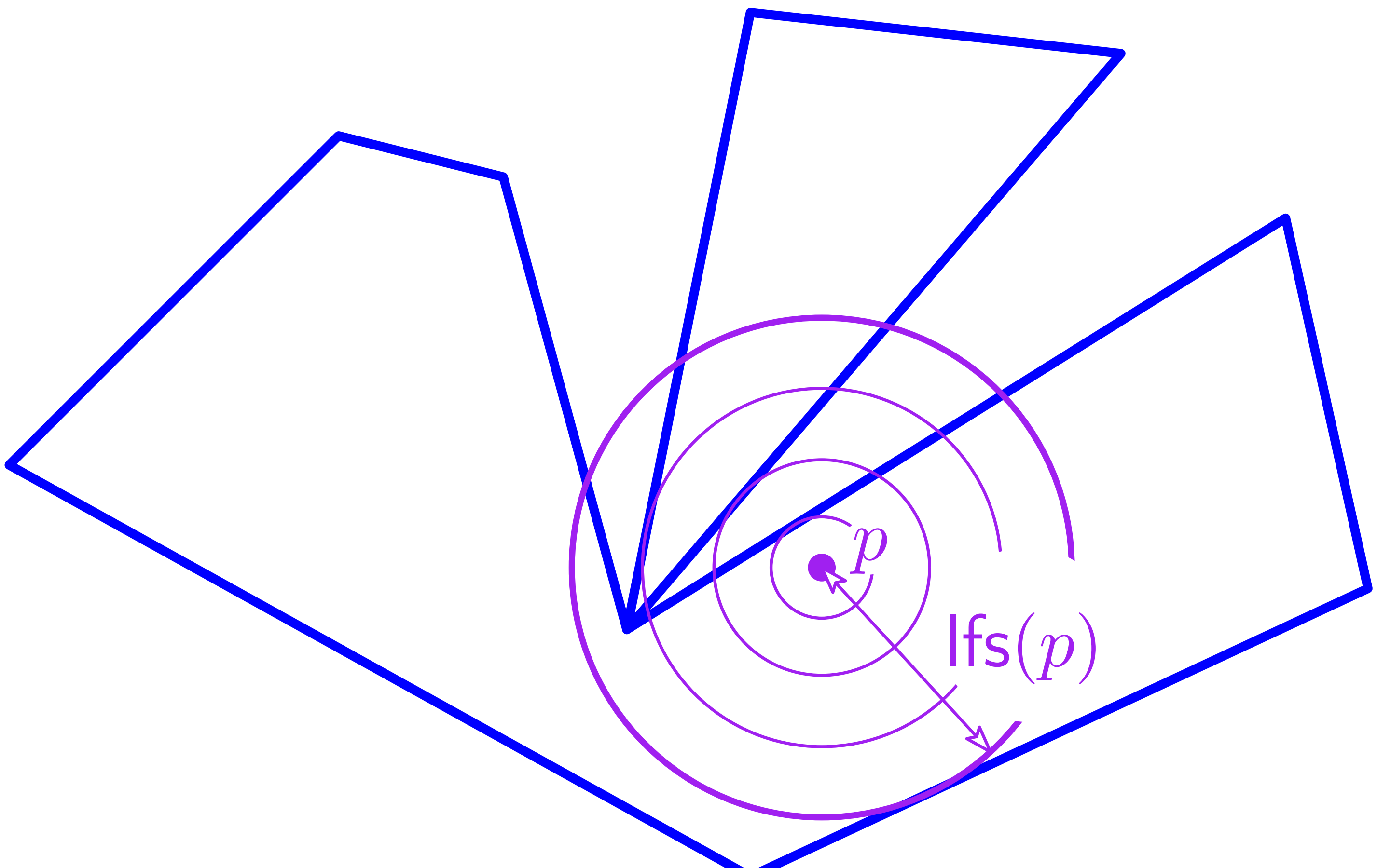


# Meshing

Delaunay mesh refinement [Ruppert]

lfs: $\mathbb{R}^2 \to \mathbb{R}$

distance to second non incident segment



# Meshing

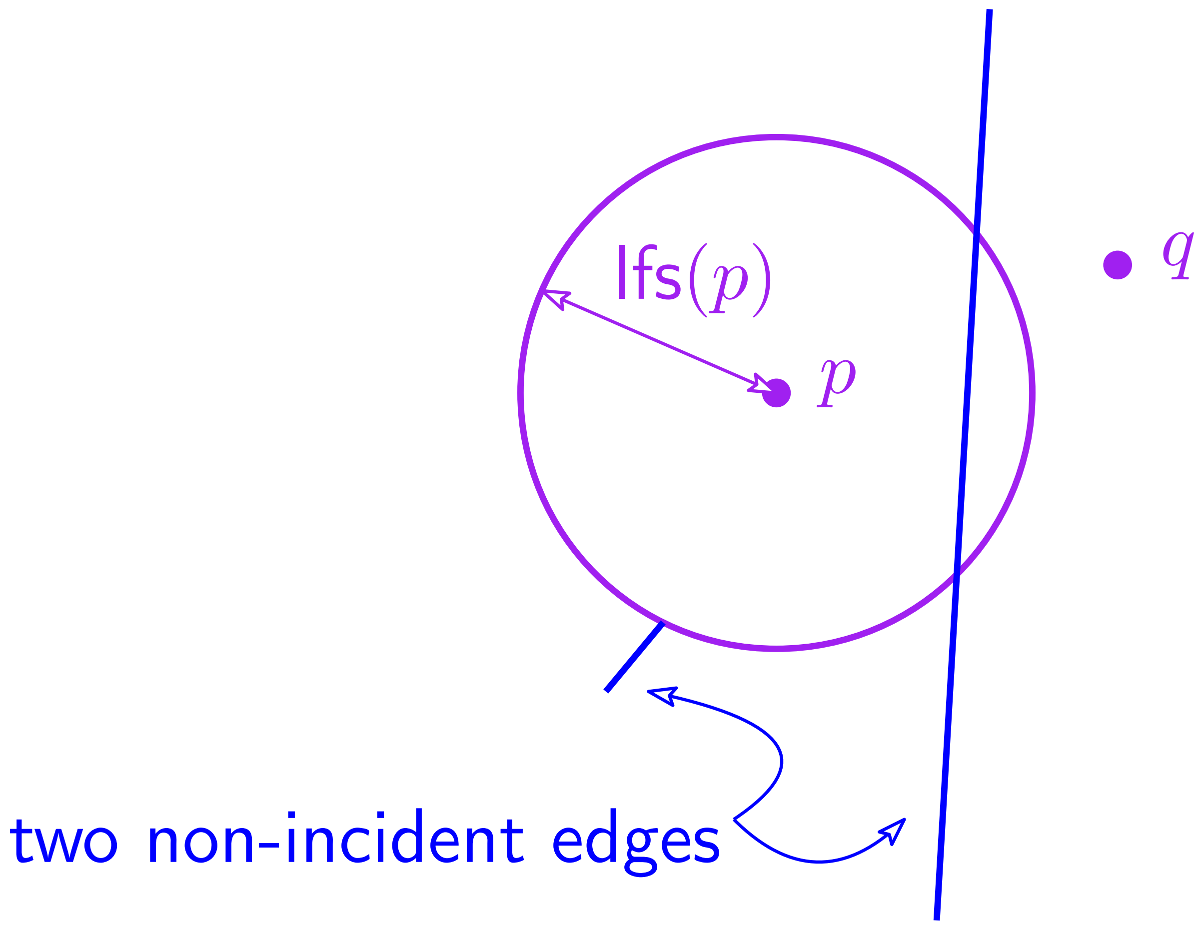
Delaunay mesh refinement [Ruppert]

Lemma: $\text{lfs}(q) \leq \text{lfs}(p) + \|pq\|$

# Meshing

Delaunay mesh refinement [Ruppert]

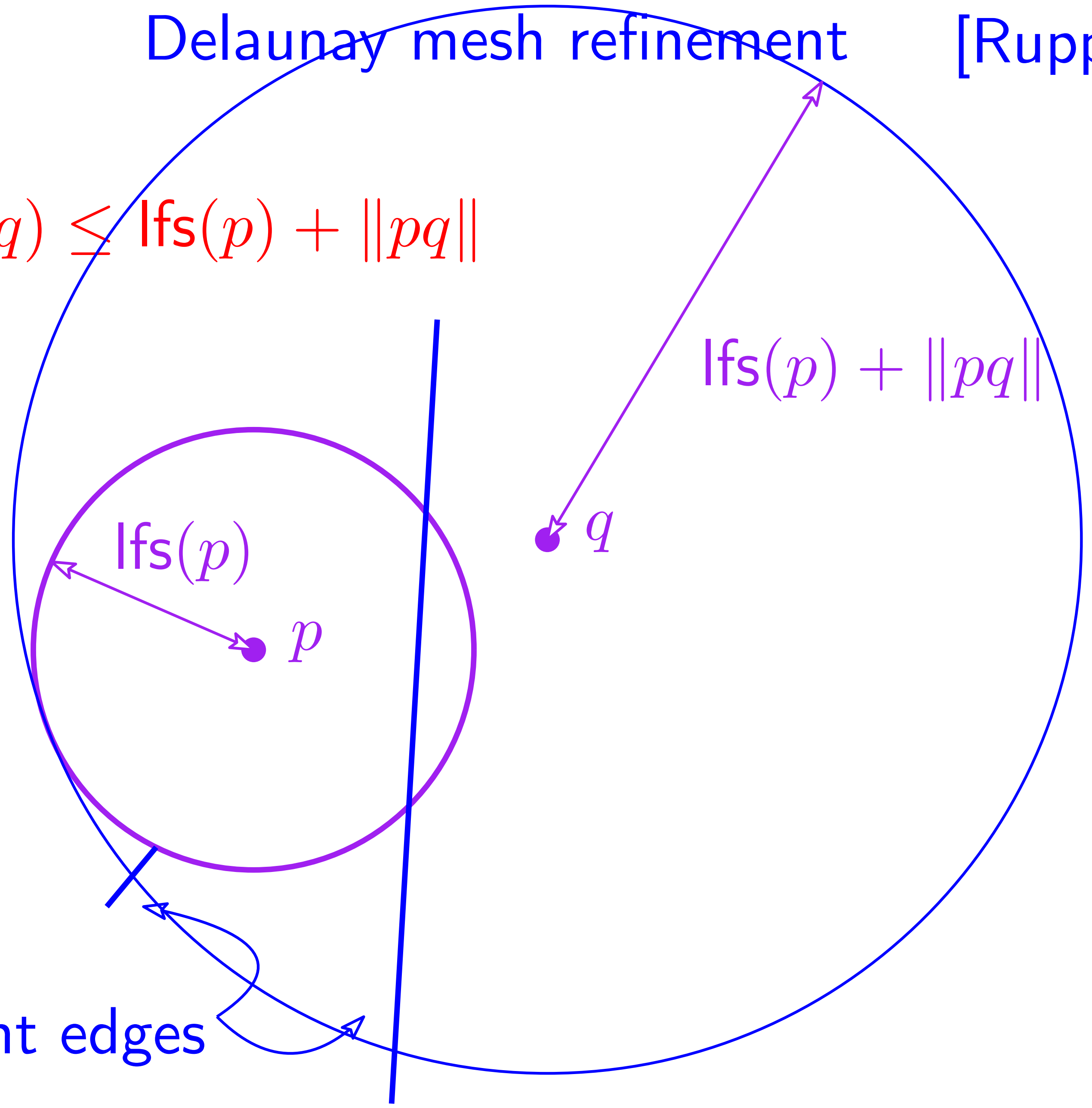
Lemma: $\mathsf{lfs}(q) \leq \mathsf{lfs}(p) + \|pq\|$



# Meshing

Delaunay mesh refinement [Ruppert]

Lemma: $\mathsf{lfs}(q) \leq \mathsf{lfs}(p) + \|pq\|$




two non-incident edges

# Meshing

Delaunay mesh refinement [Ruppert]

Assume no angles $\geq 90^\circ$ in input

Lemma:

There are constants $C_S \geq C_T \geq 1$ such that

At initialization, nearest vertex of vertex $p$
is at distance $\geq \mathrm{lfs}(p)$

Nearest vertex of circumcenter $p$ of skinny triangle
is at distance $\geq \frac{1}{C_T}\mathrm{lfs}(p)$

Nearest vertex of midpoint $p$ of split segment
is at distance $\geq \frac{1}{C_S}\mathrm{lfs}(p)$

# Meshing

Delaunay mesh refinement [Ruppert]

Lemma:

There are constants $C_S \geq C_T \geq 1$ such that

Easy

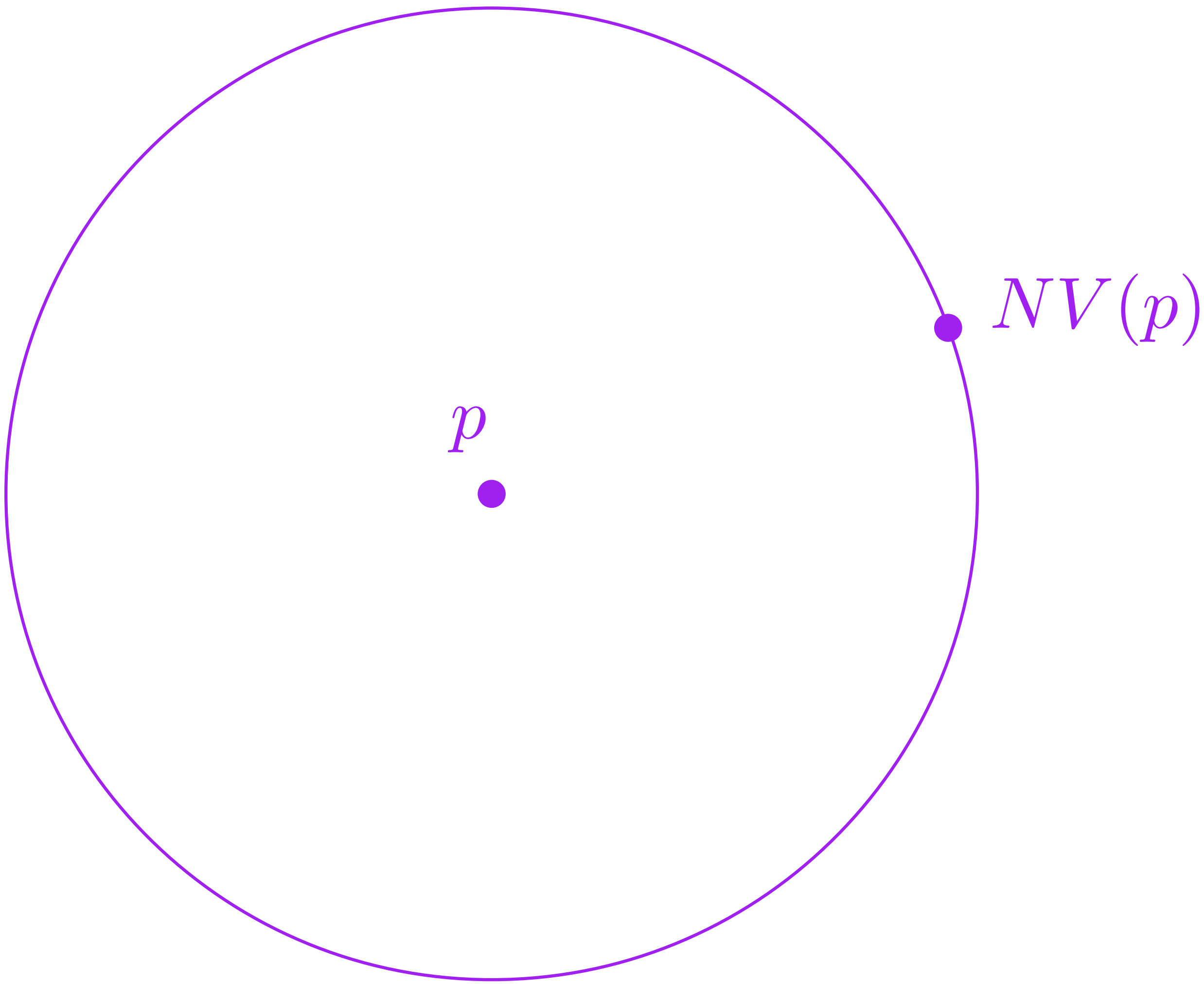
At initialization, nearest vertex of vertex $p$ is at distance $\geq \mathsf{lfs}(p)$

Nearest vertex of circumcenter $p$ of skinny triangle is at distance $\geq \frac{1}{C_T}\mathsf{lfs}(p)$

Nearest vertex of midpoint $p$ of split segment is at distance $\geq \frac{1}{C_S}\mathsf{lfs}(p)$

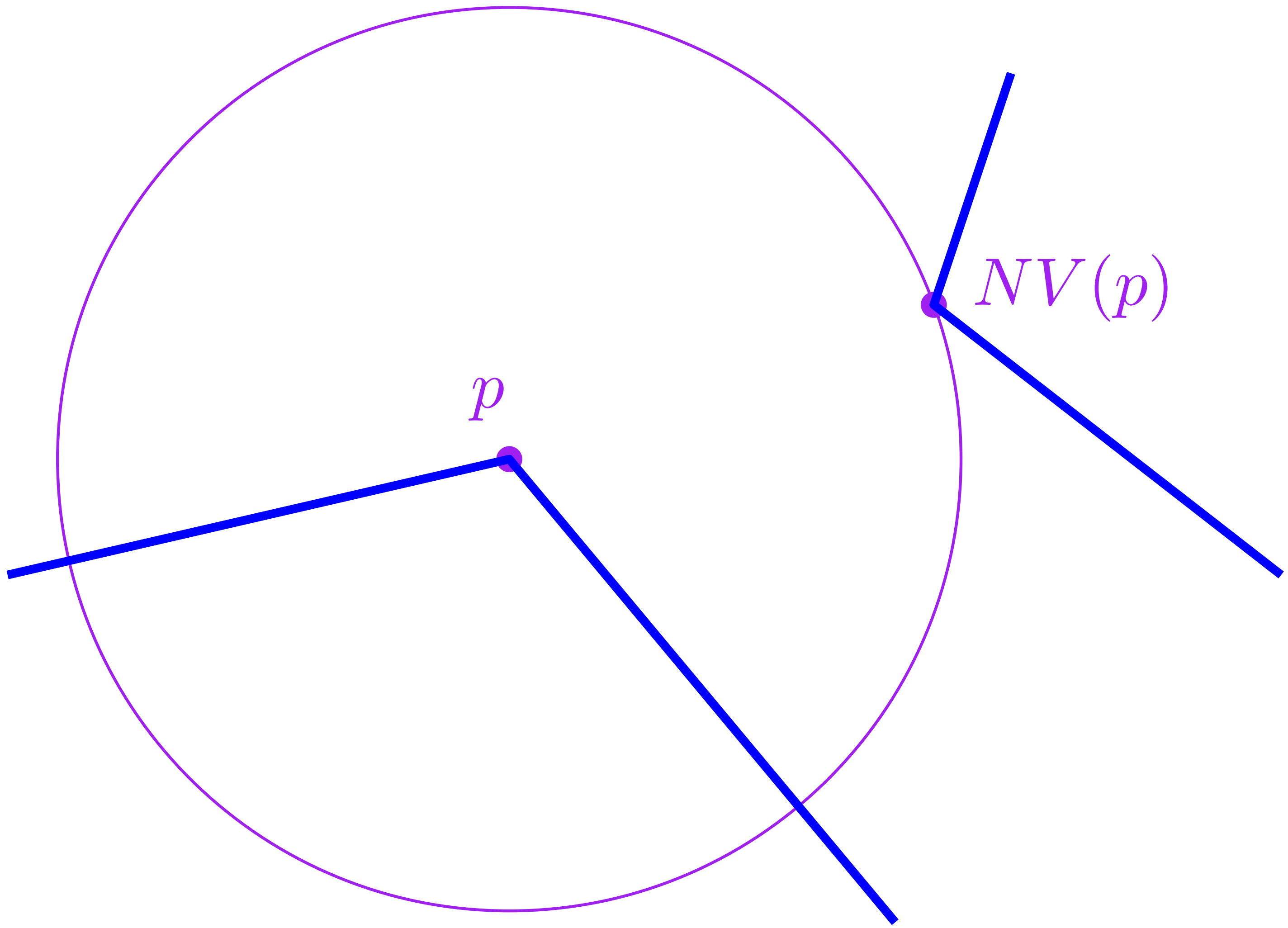
# Meshing

Delaunay mesh refinement [Ruppert]



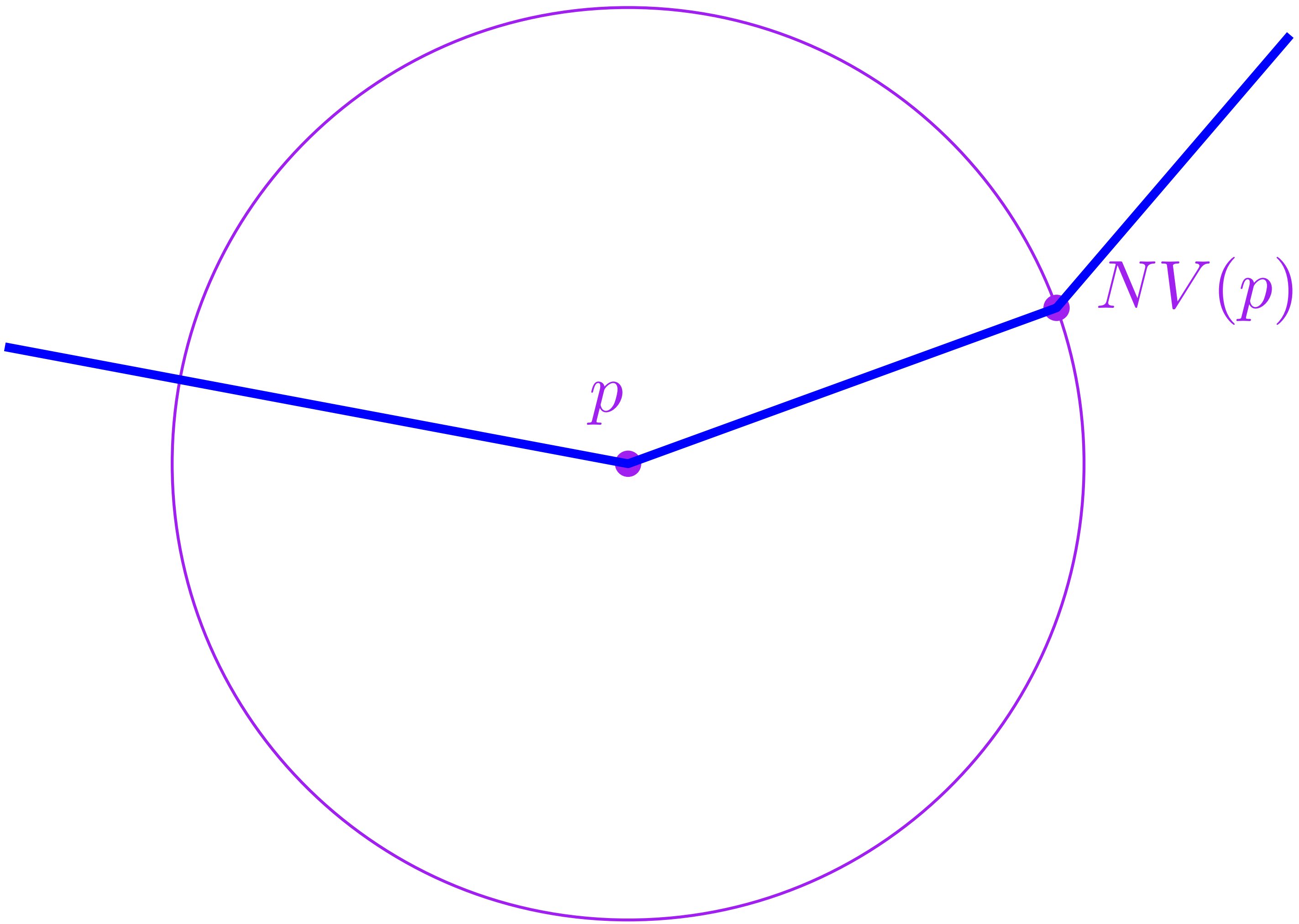
# Meshing

Delaunay mesh refinement [Ruppert]



# Meshing

## Delaunay mesh refinement [Ruppert]



Lemma:

There are constants $C_S \geq C_T \geq 1$ such that

At initialization, nearest vertex of vertex $p$
is at distance $\geq$ lfs$(p)$

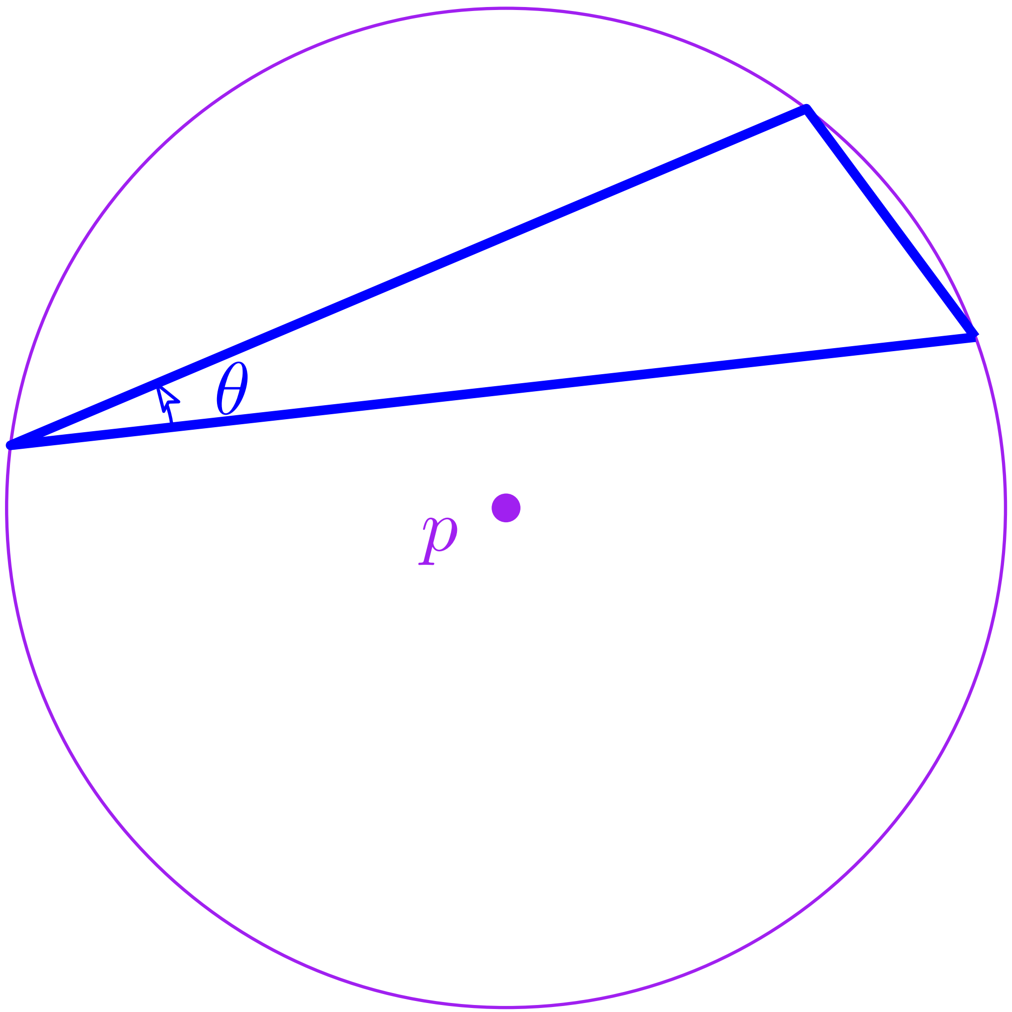
Nearest vertex of circumcenter $p$ of skinny triangle
is at distance $\geq \frac{1}{C_T}$lfs$(p)$

Nearest vertex of midpoint $p$ of split segment
is at distance $\geq \frac{1}{C_S}$lfs$(p)$

# Meshing

Delaunay mesh refinement [Ruppert]

skinny: $\theta < \alpha$

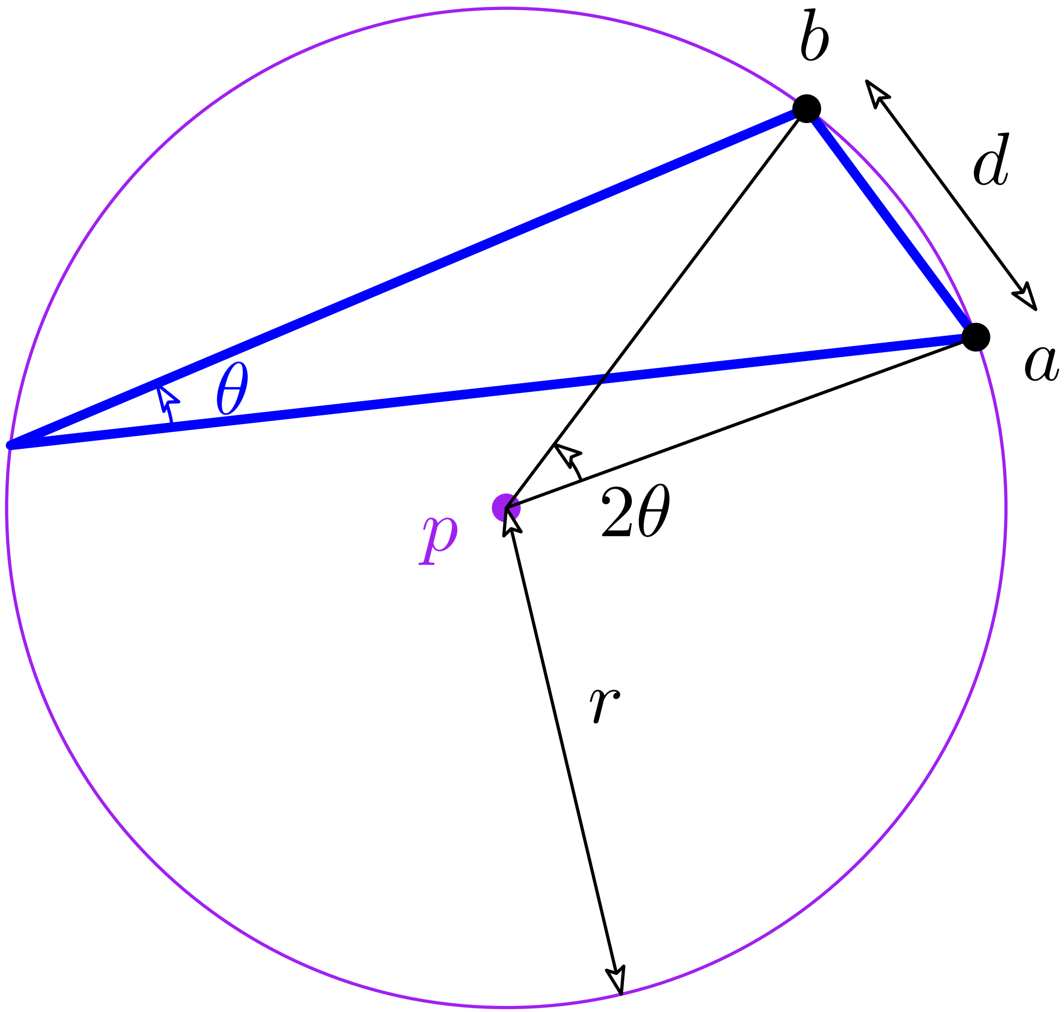


# Meshing

## Delaunay mesh refinement [Ruppert]

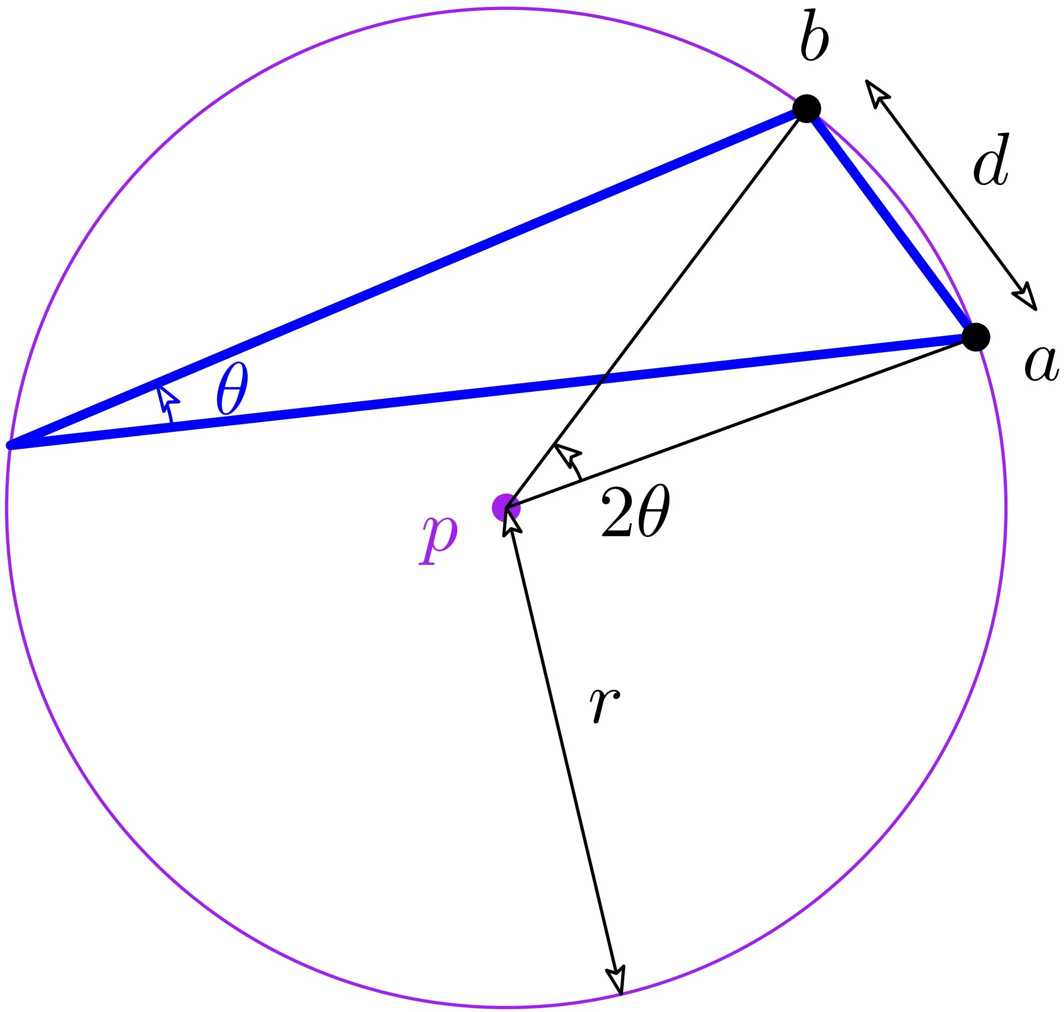
skinny: $\theta < \alpha$

wlog: $a$ added after $b$



# Meshing

Delaunay mesh refinement [Ruppert]



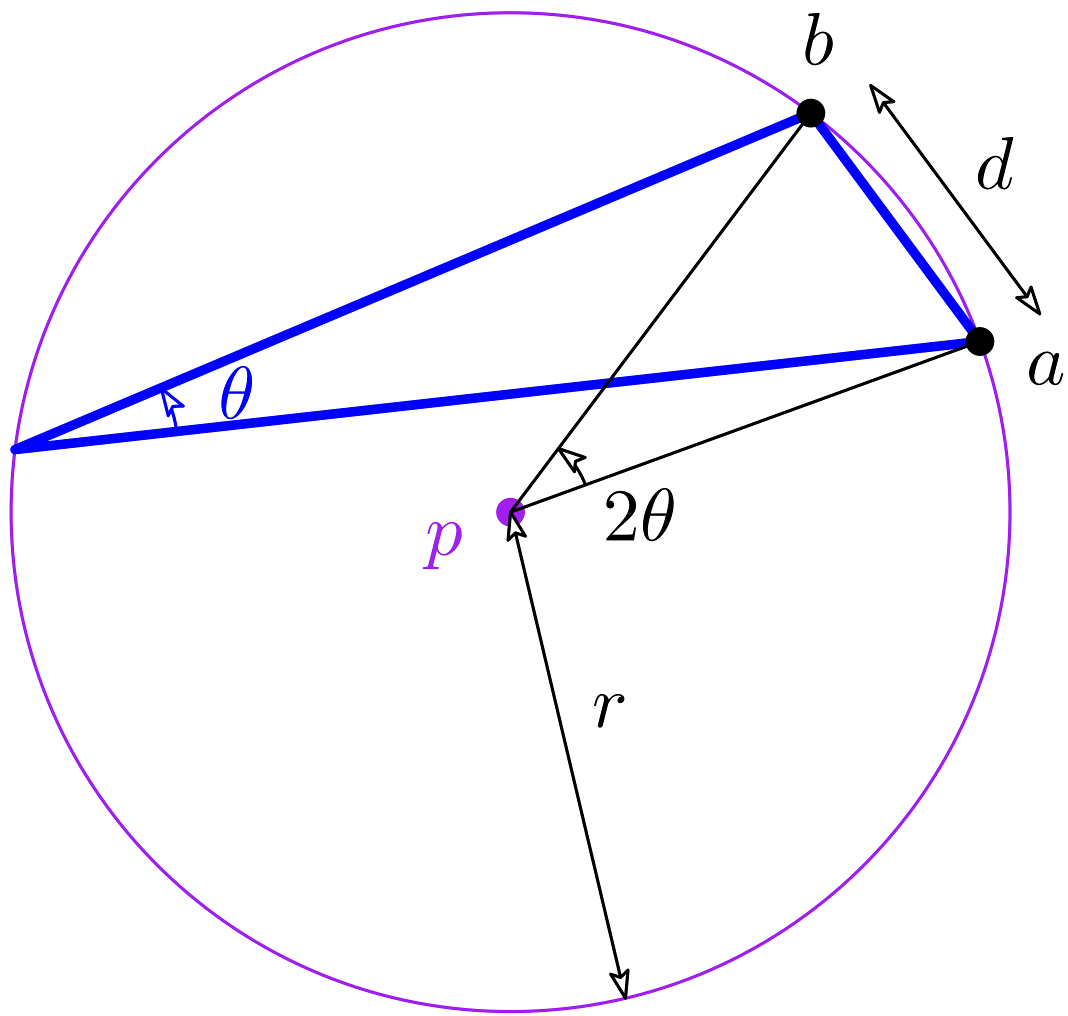

skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

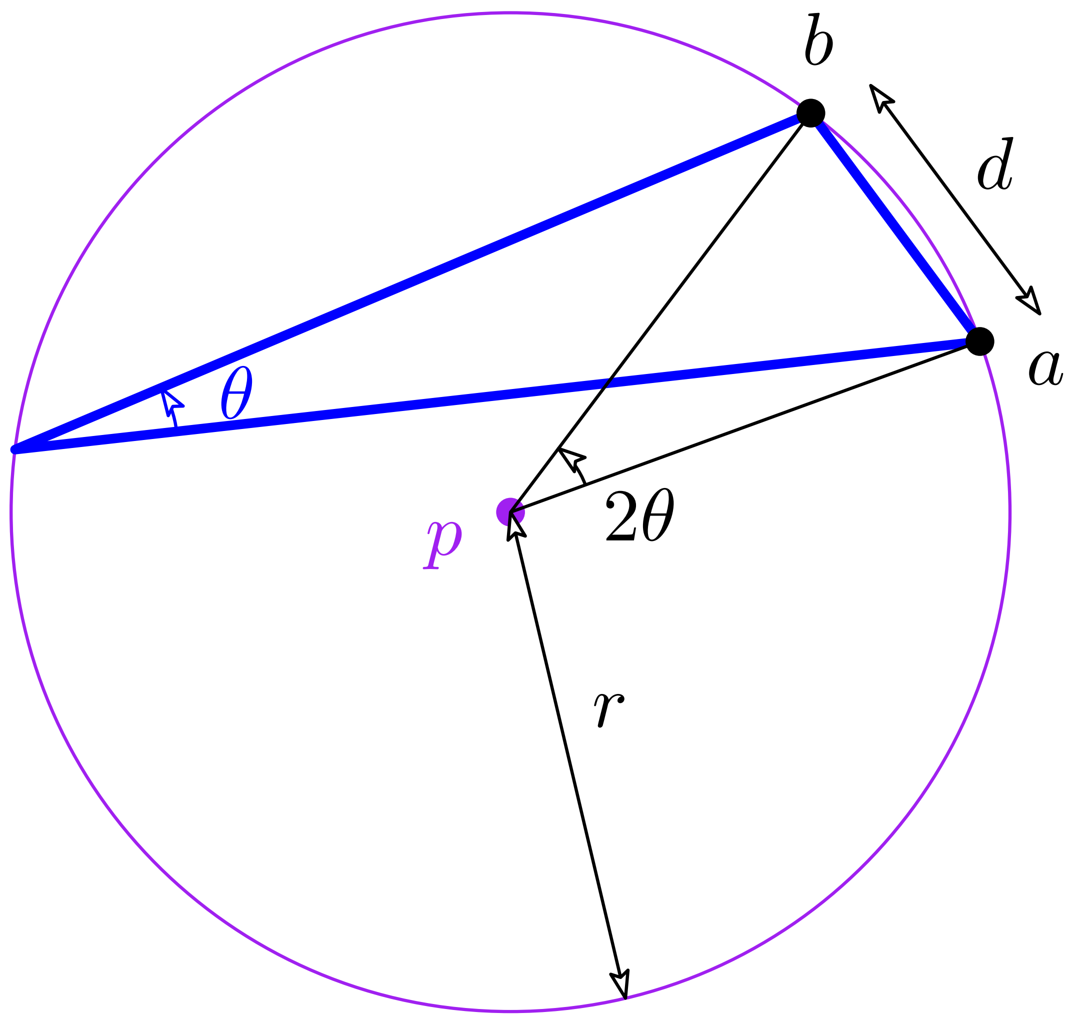
$b$ also input vertex

$\text{lfs}(a) \leq d$

by first statement

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

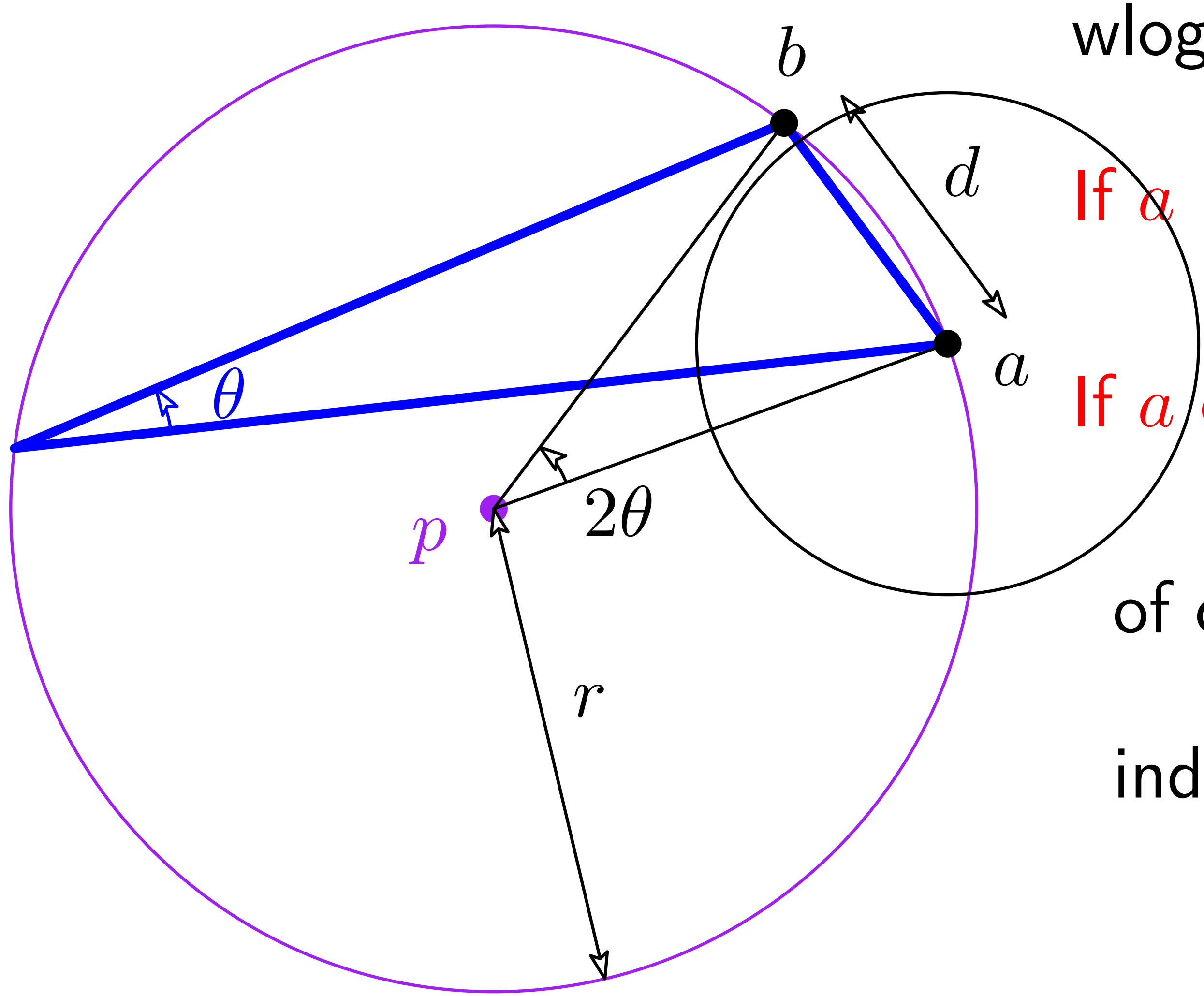
If $a$ input vertex

$\text{lfs}(a) \leq d$

If $a$ circumcenter

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

$\text{lfs}(a) \leq d$

If $a$ circumcenter

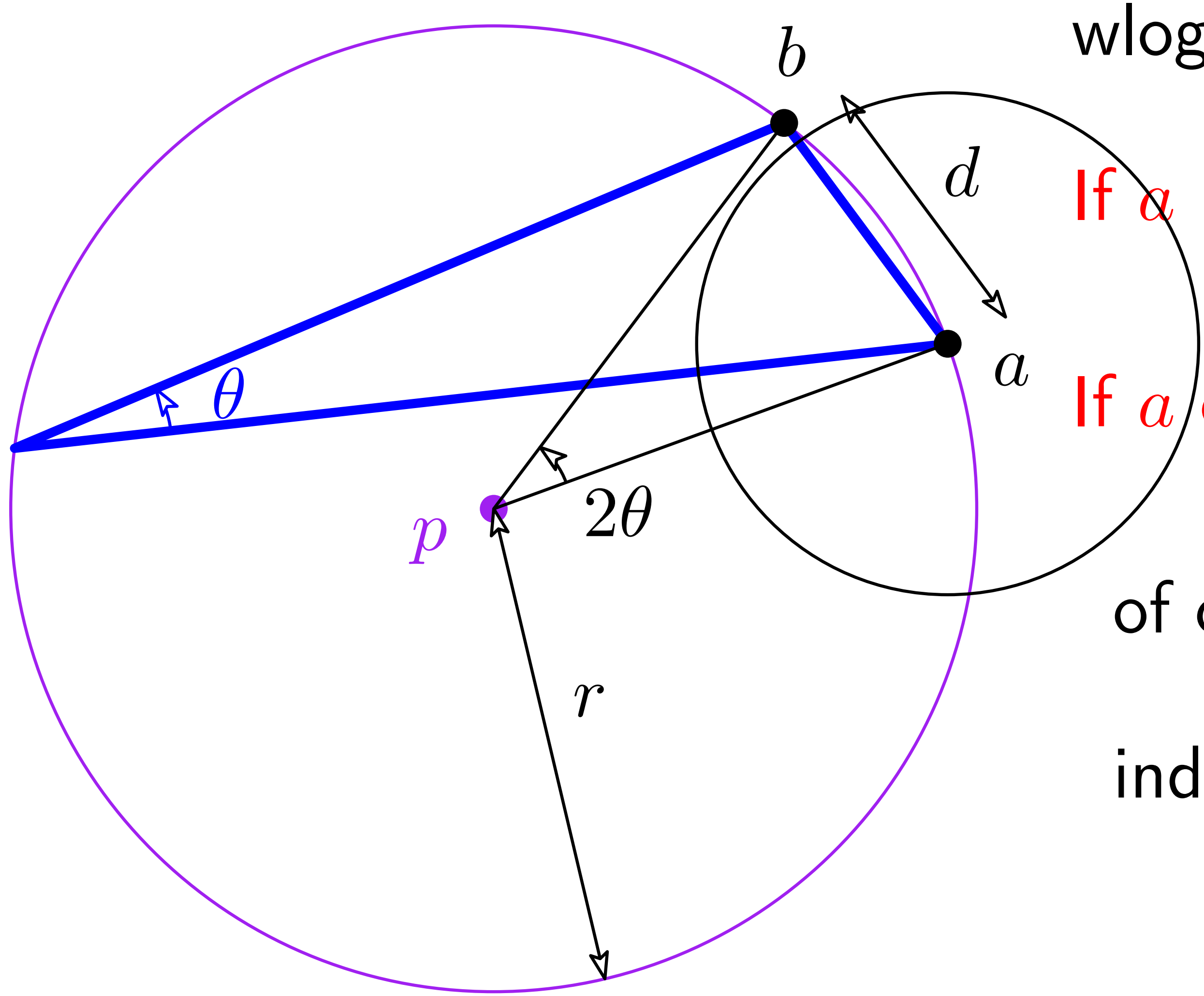
of circle of radius $\leq d$

induction:

$$d \geq \frac{1}{C_T}\ \text{lfs}(a)$$

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

$\text{lfs}(a) \leq d$

If $a$ circumcenter

$\text{lfs}(a) \leq C_T d$

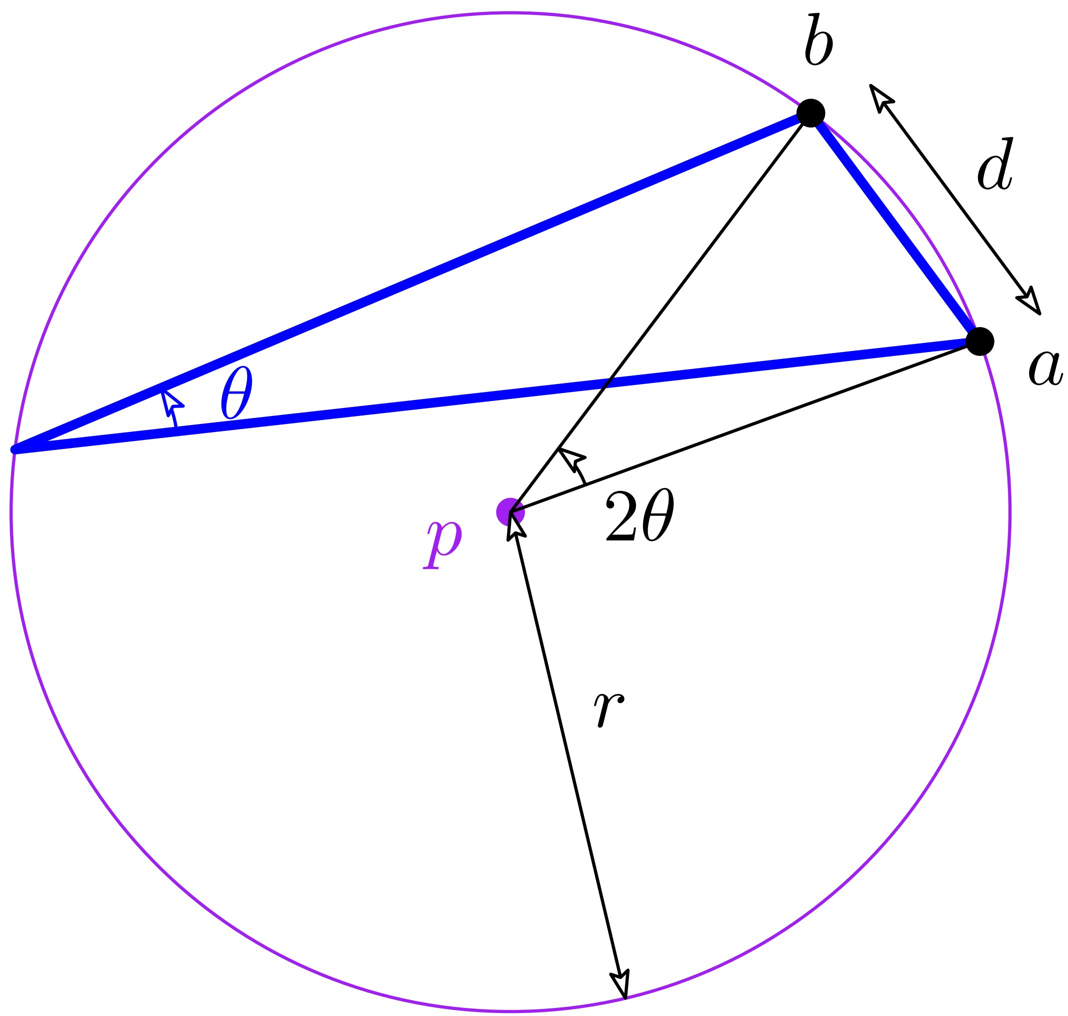
of circle of radius $\leq d$

induction:

$$d \geq \frac{1}{C_T}\ \text{lfs}(a)$$

# Meshing

## Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

$$\text{lfs}(a) \leq d$$

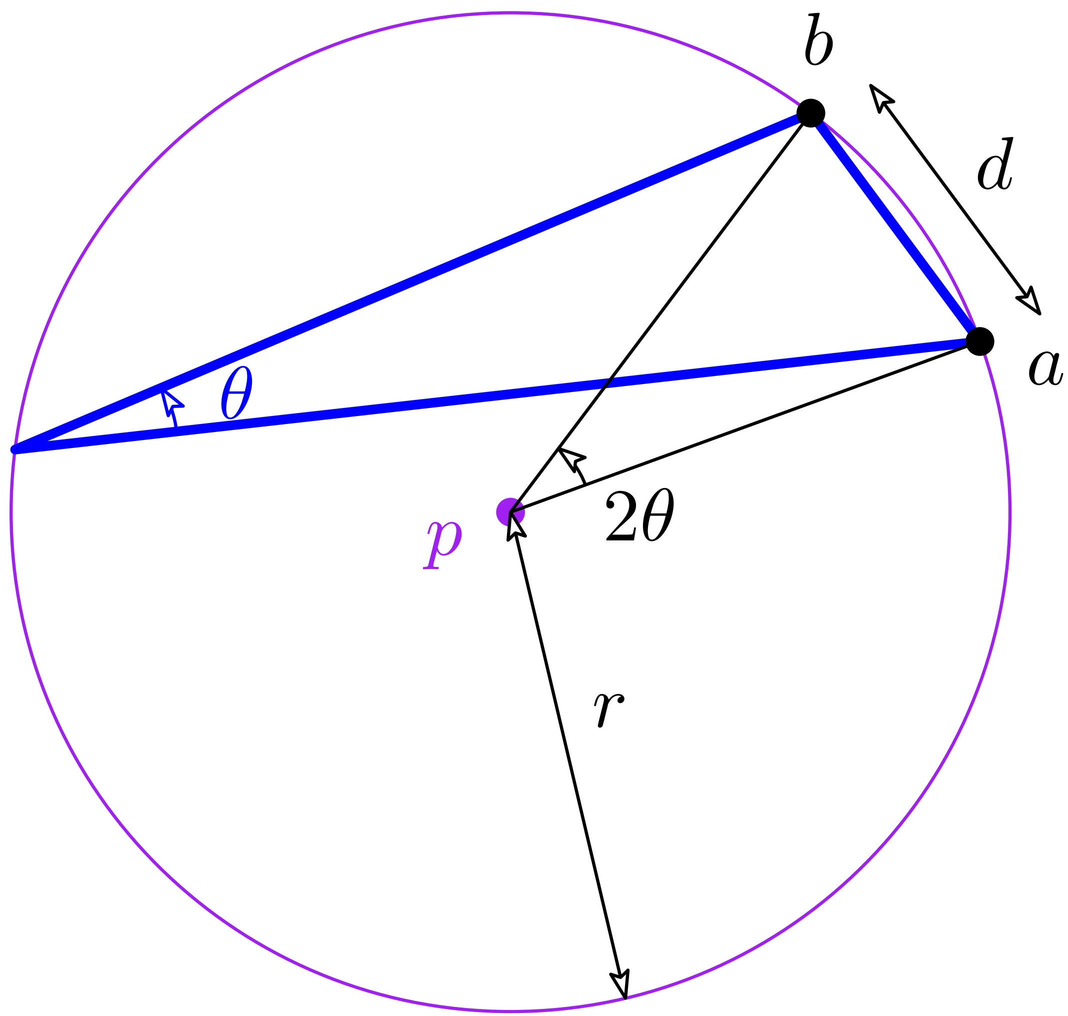
If $a$ circumcenter

$$\text{lfs}(a) \leq C_T d$$

If $a$ midpoint

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

$$\text{lfs}(a) \leq d$$

If $a$ circumcenter

$$\text{lfs}(a) \leq C_T d$$

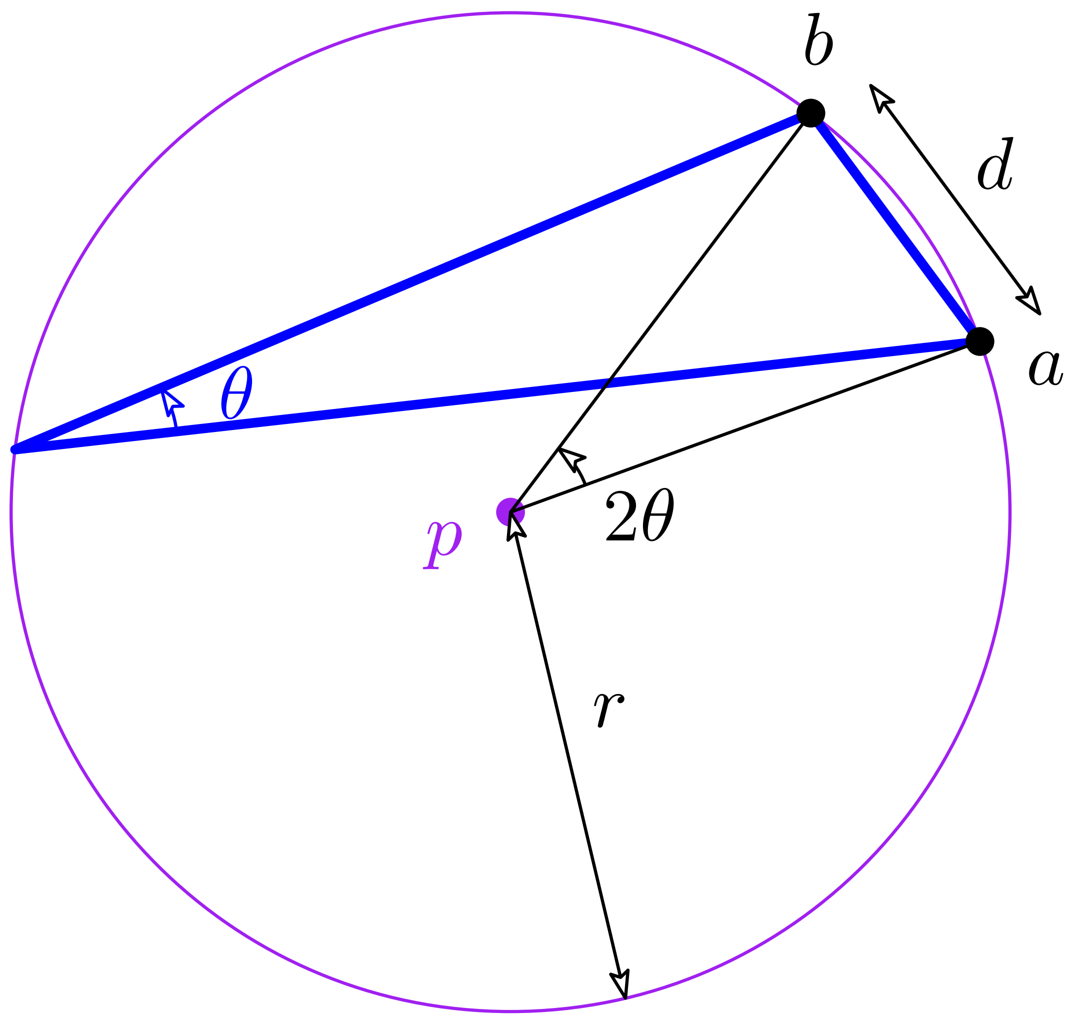
If $a$ midpoint

Induction:

$$d \geq \frac{1}{C_S}\ \text{lfs}(a)$$

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

$$\text{lfs}(a) \leq d$$

If $a$ circumcenter

$$\text{lfs}(a) \leq C_T d$$

If $a$ midpoint

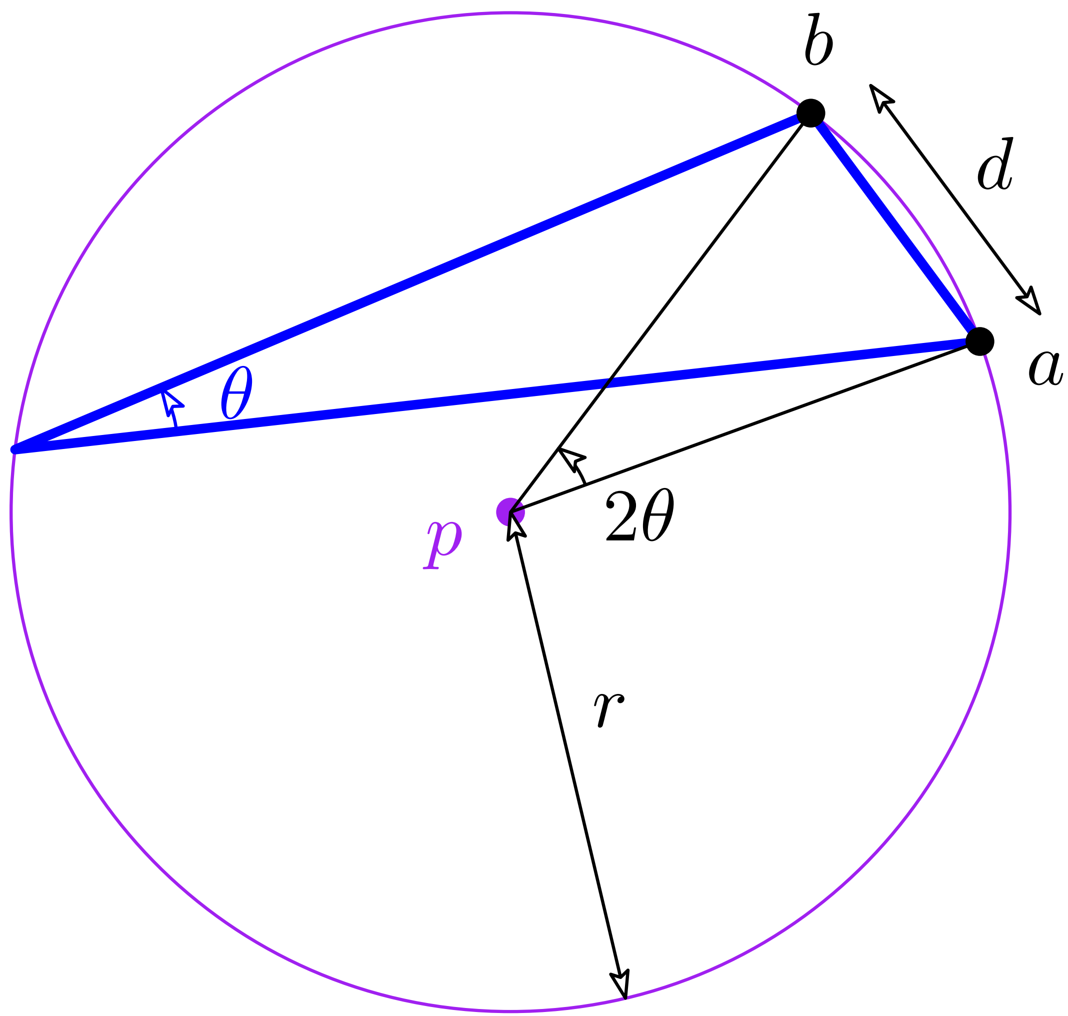
$$\text{lfs}(a) \leq C_S d$$

Induction:

$$d \geq \frac{1}{C_S}\ \text{lfs}(a)$$

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

$\mathsf{lfs}(a) \leq d$

If $a$ circumcenter

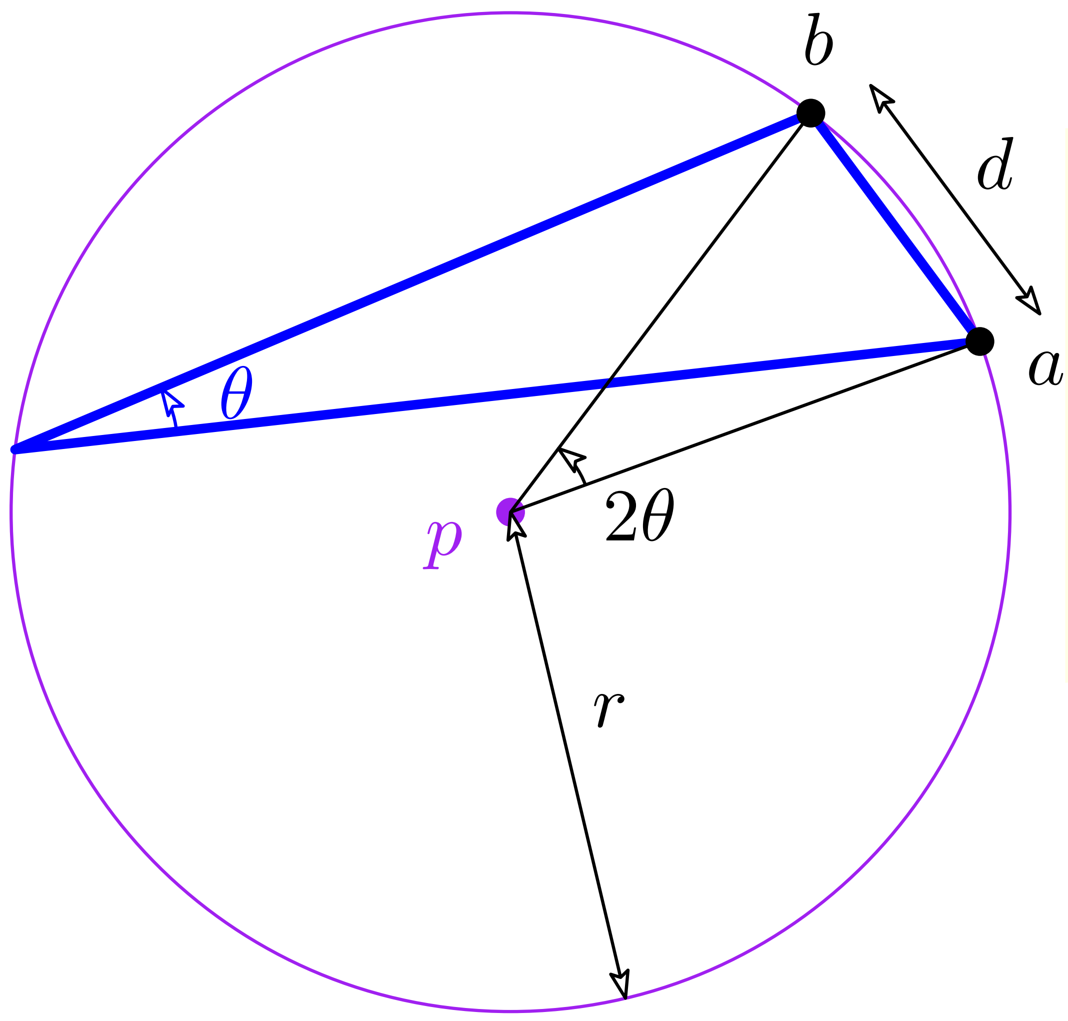
$\mathsf{lfs}(a) \leq C_T d$

If $a$ midpoint

$\mathsf{lfs}(a) \leq C_S d$

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

$\mathsf{lfs}(a) \leq d$

If $a$ circumcenter

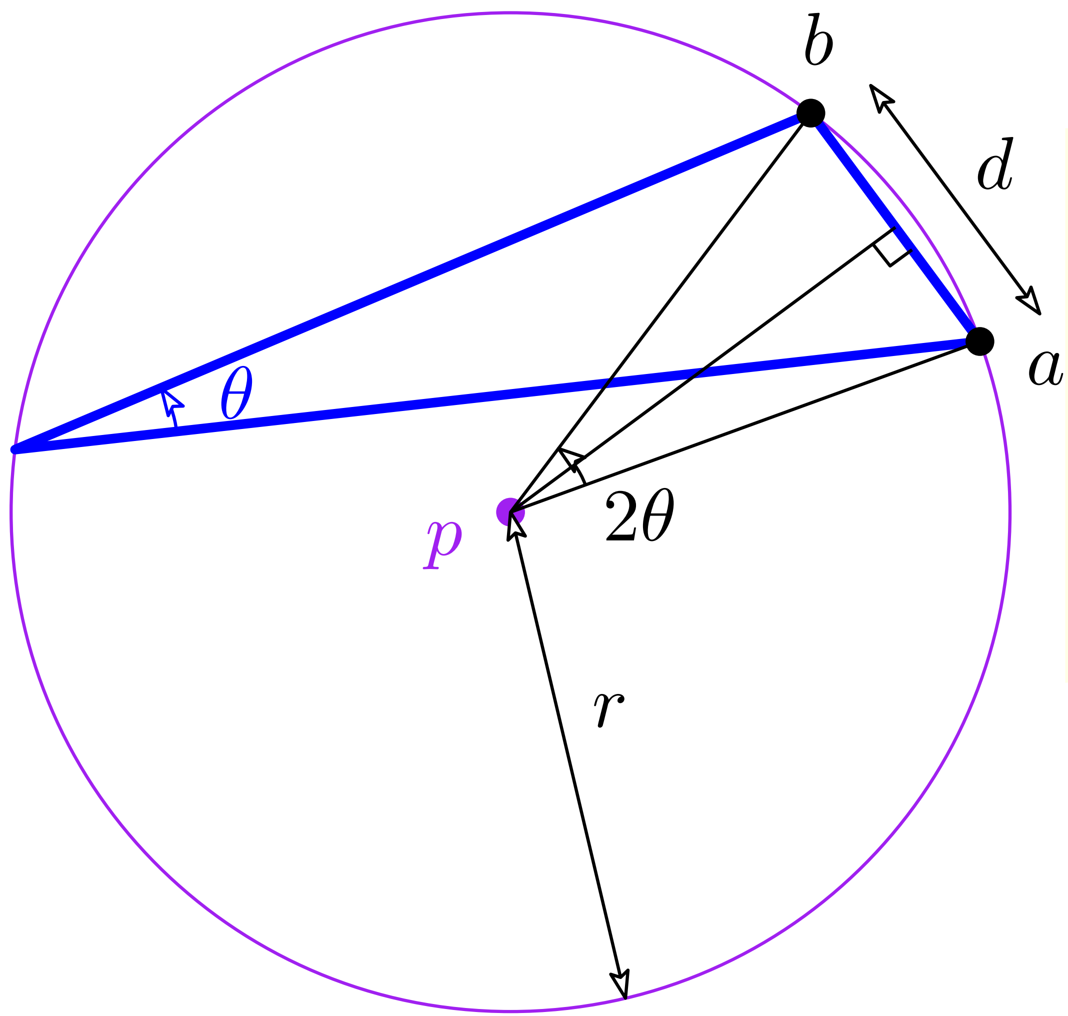
$\mathsf{lfs}(a) \leq C_T d$

If $a$ midpoint

$\mathsf{lfs}(a) \leq C_S d$

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

$\mathsf{lfs}(a) \le d$

If $a$ circumcenter

$\mathsf{lfs}(a) \le C_T d$

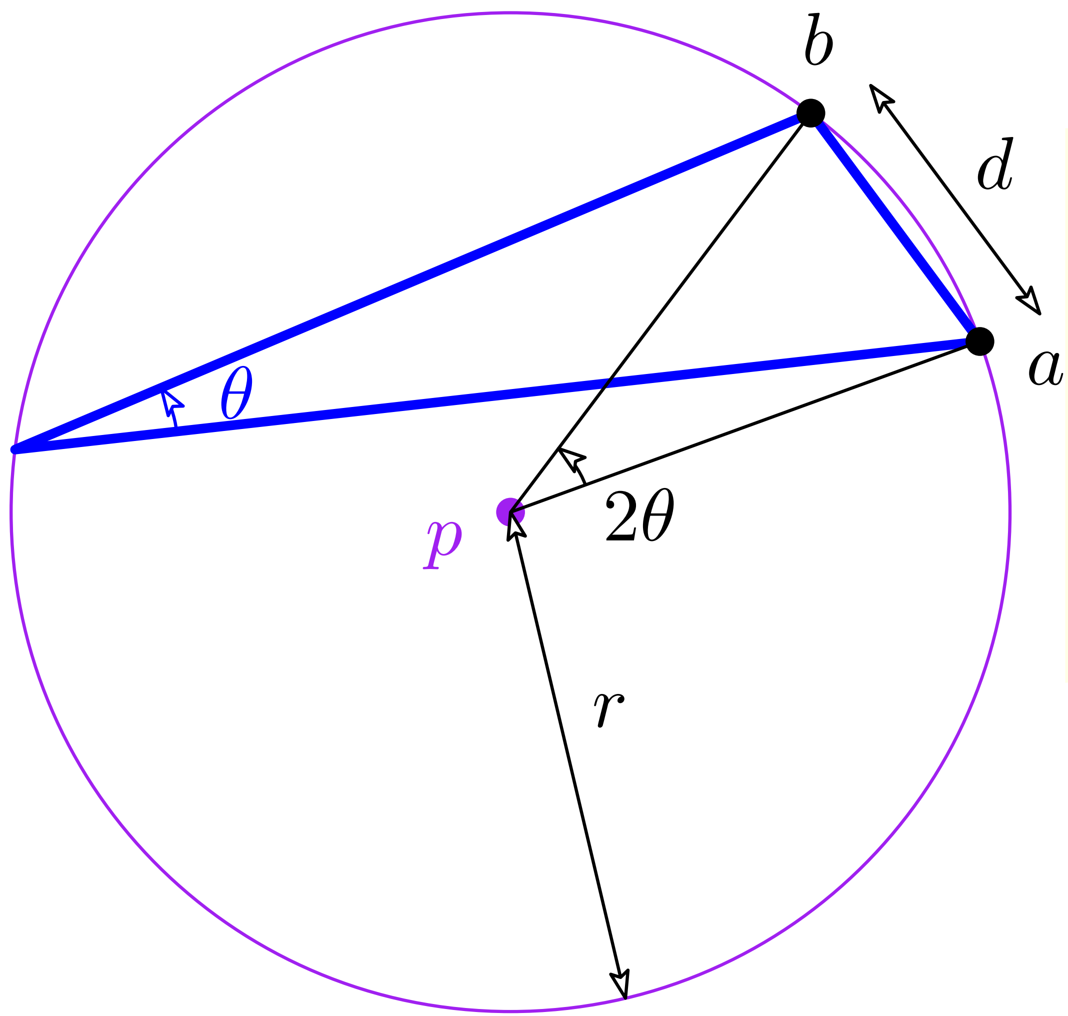
If $a$ midpoint

$\mathsf{lfs}(a) \le C_S d$

$d = 2r \sin\theta$

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

$\mathsf{lfs}(a) \leq d$

If $a$ circumcenter

$\mathsf{lfs}(a) \leq C_T d$

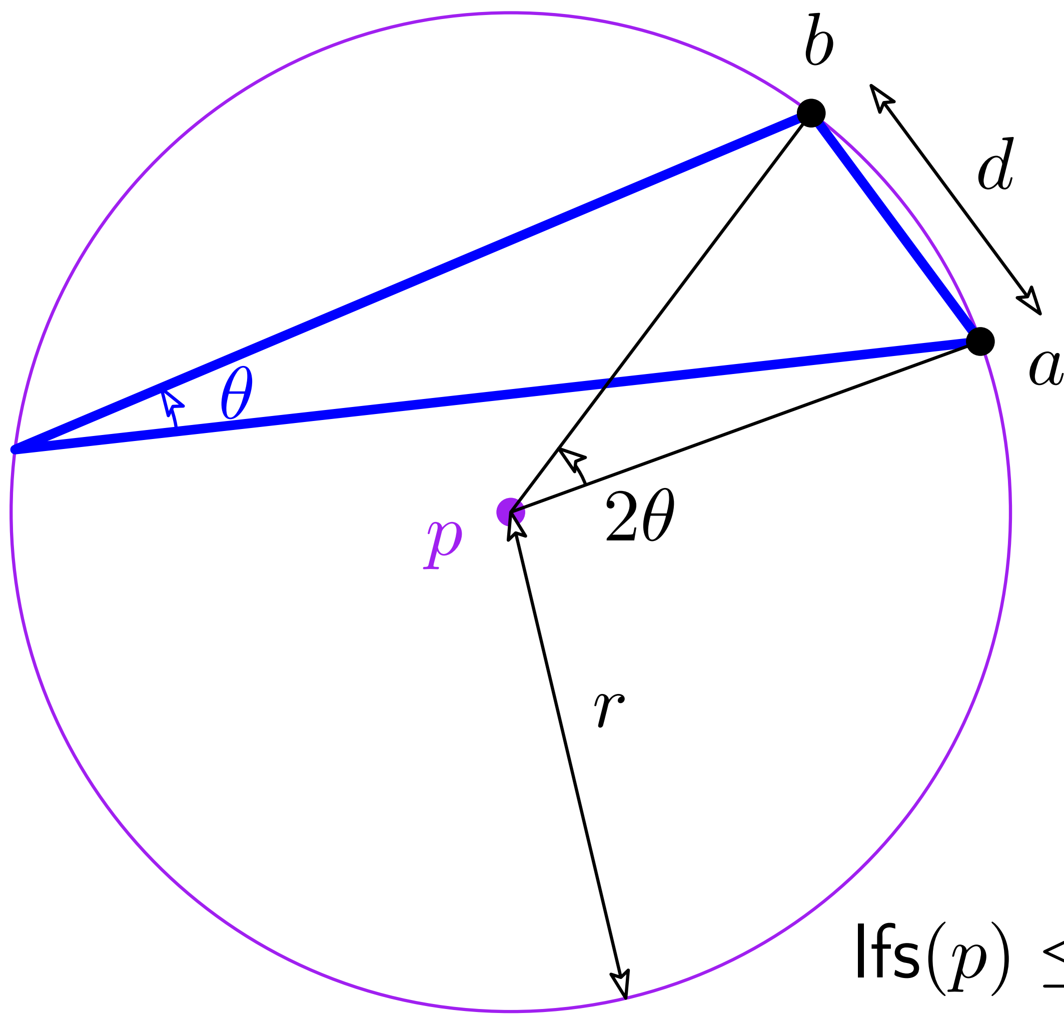
If $a$ midpoint

$\mathsf{lfs}(a) \leq C_S d$

$\mathsf{lfs}(a) \leq 2 C_S r \sin\theta$

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

$\mathsf{lfs}(a) \leq d$

If $a$ circumcenter

$\mathsf{lfs}(a) \leq C_T d$

If $a$ midpoint

$\mathsf{lfs}(a) \leq C_S d$

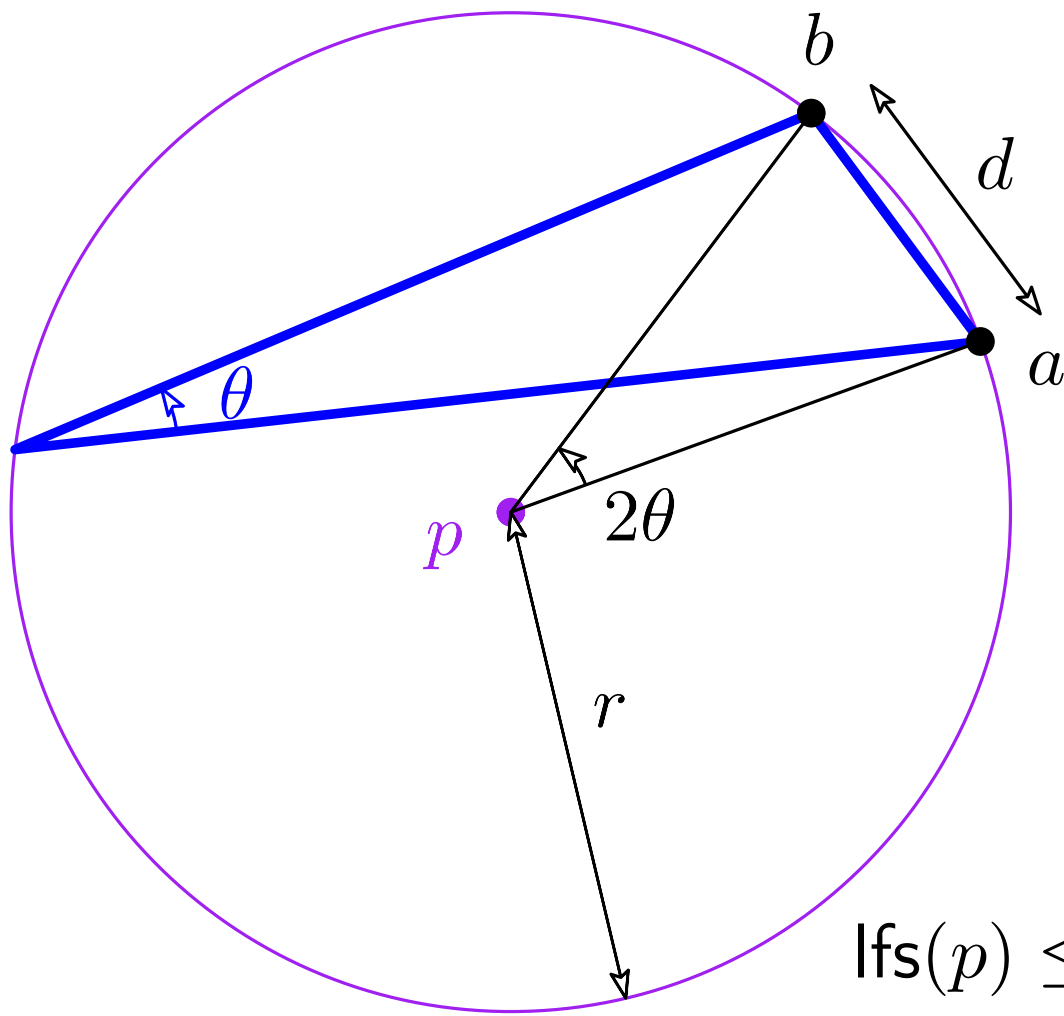
$\mathsf{lfs}(a) \leq 2 C_S r \sin\theta$

$\mathsf{lfs}(p) \leq \mathsf{lfs}(a) + r$

Lemma

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

$\mathsf{lfs}(a) \leq d$

If $a$ circumcenter

$\mathsf{lfs}(a) \leq C_T d$

If $a$ midpoint

$\mathsf{lfs}(a) \leq C_S d$

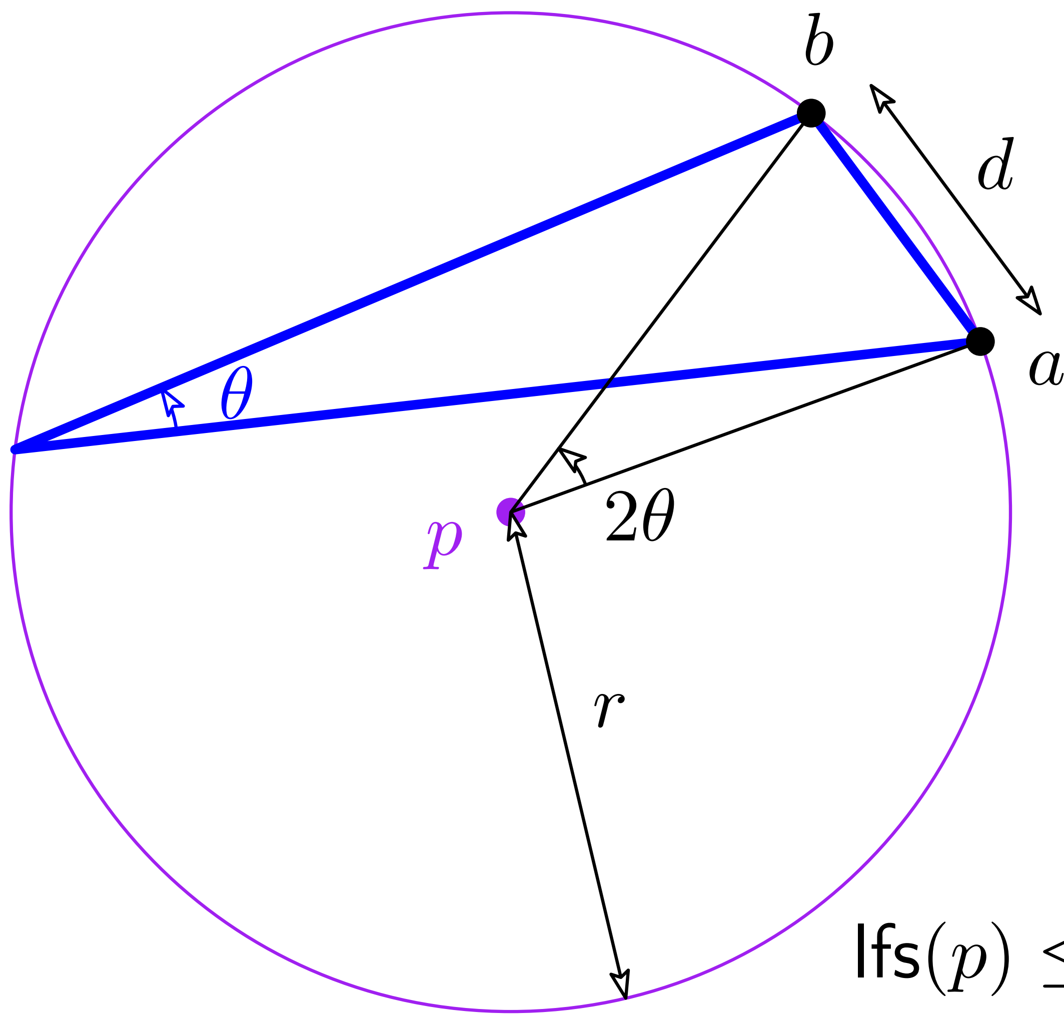
$\mathsf{lfs}(a) \leq 2C_S r \sin\theta$

$\mathsf{lfs}(p) \leq \mathsf{lfs}(a) + r$ Lemma

$\leq r(1 + 2C_S \sin\alpha)$

# Meshing

Delaunay mesh refinement [Ruppert]




skinny: $\theta < \alpha$

wlog: $a$ added after $b$

If $a$ input vertex

$\mathsf{lfs}(a) \leq d$

If $a$ circumcenter

$\mathsf{lfs}(a) \leq C_T d$

If $a$ midpoint

$\mathsf{lfs}(a) \leq C_S d$

$\mathsf{lfs}(a) \leq 2C_S r \sin\theta$

Lemma

$\mathsf{lfs}(p) \leq \mathsf{lfs}(a) + r$

$\leq r(1 + 2C_S \sin\alpha)$

$\boxed{\text{OK if } C_T \geq 1 + 2C_S \sin\alpha}$

Lemma:

There are constants $C_S \geq C_T \geq 1$ such that

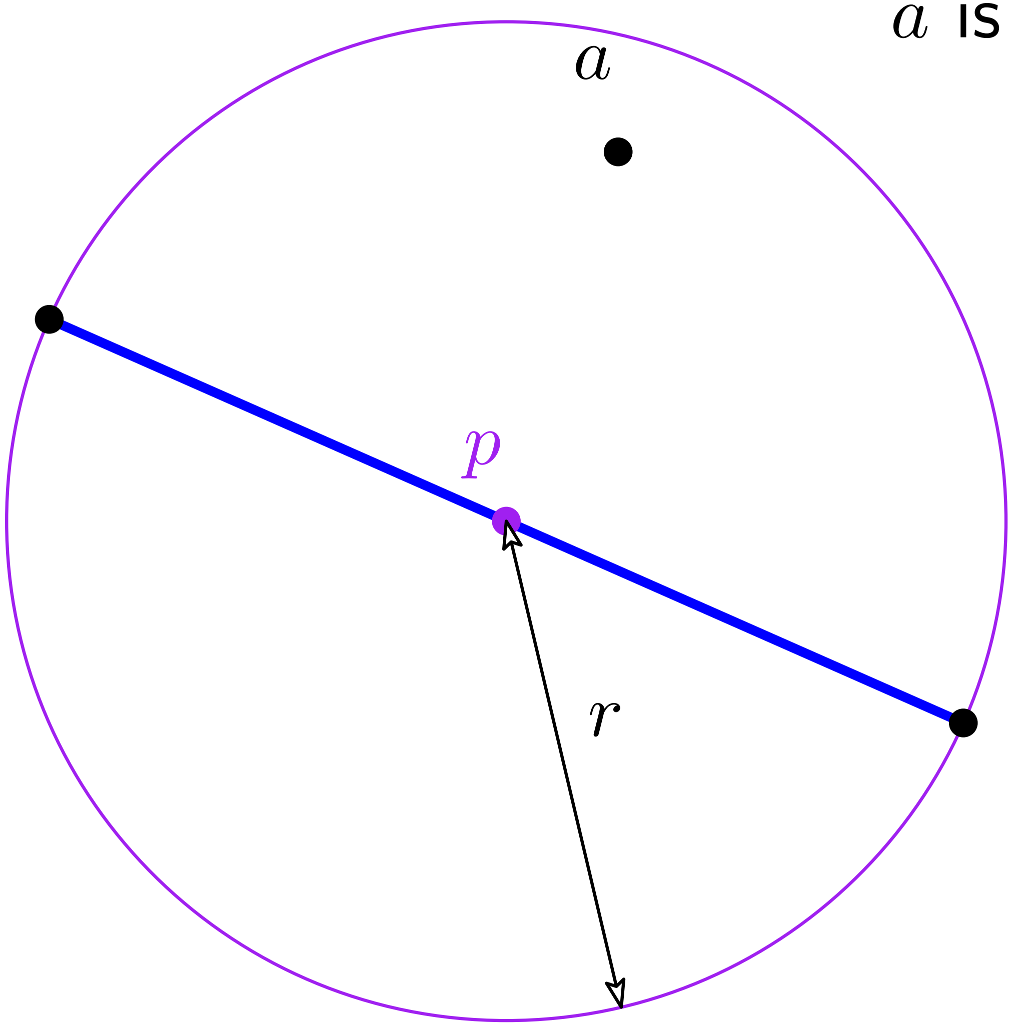
At initialization, nearest vertex of vertex $p$
is at distance $\geq$ lfs$(p)$

Nearest vertex of circumcenter $p$ of skinny triangle
is at distance $\geq \frac{1}{C_T}$lfs$(p)$

Nearest vertex of midpoint $p$ of split segment
is at distance $\geq \frac{1}{C_S}$lfs$(p)$

# Meshing

## Delaunay mesh refinement [Ruppert]

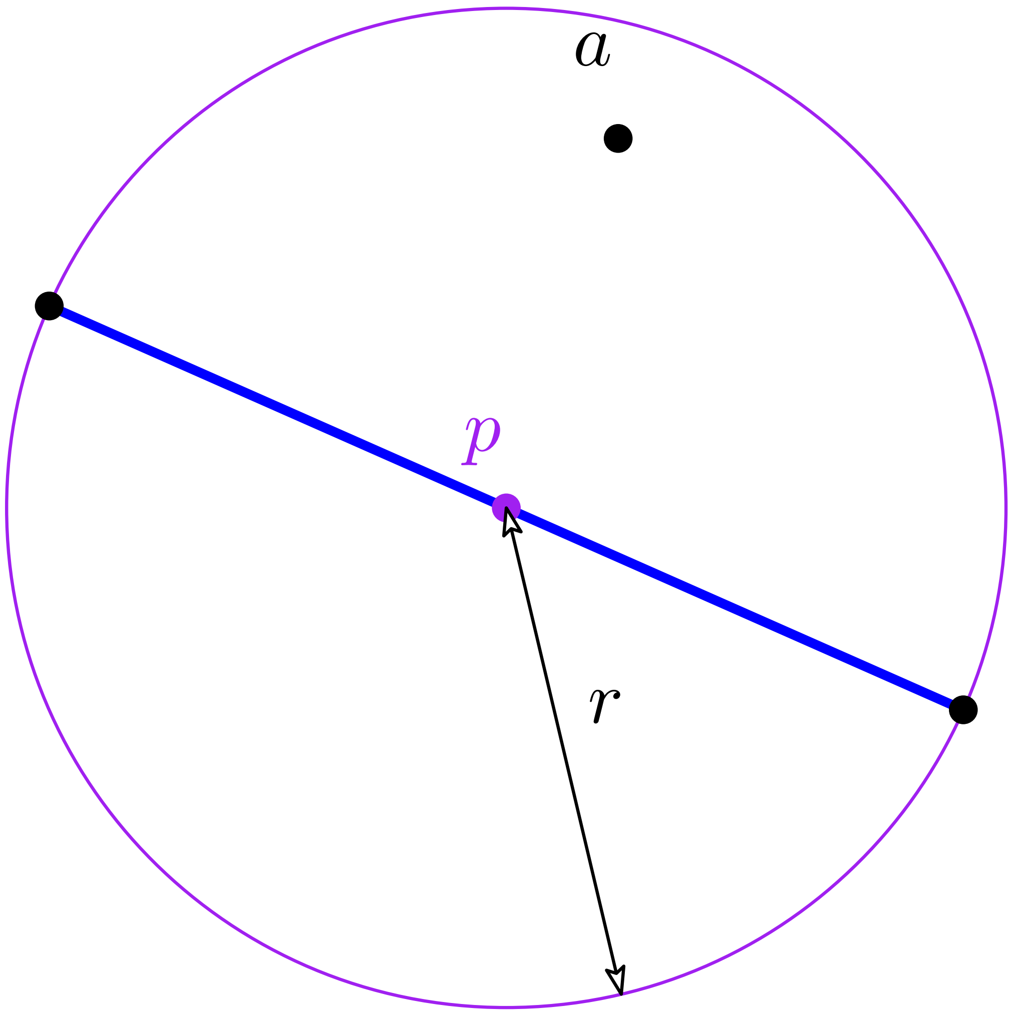



$a$ is creating the edge split

# Meshing

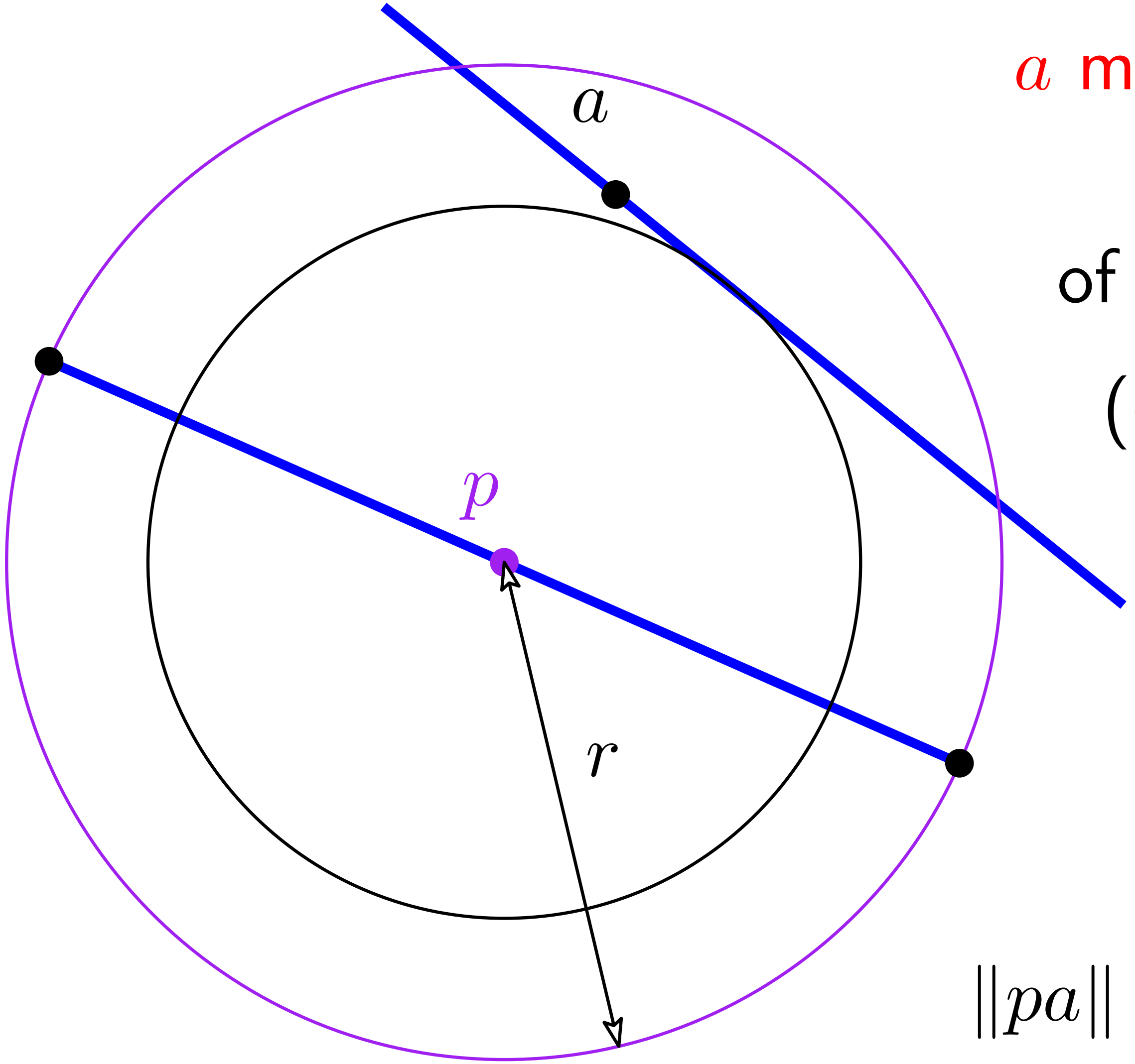
Delaunay mesh refinement [Ruppert]

$a$ midpoint



# Meshing

Delaunay mesh refinement [Ruppert]




$a$ midpoint

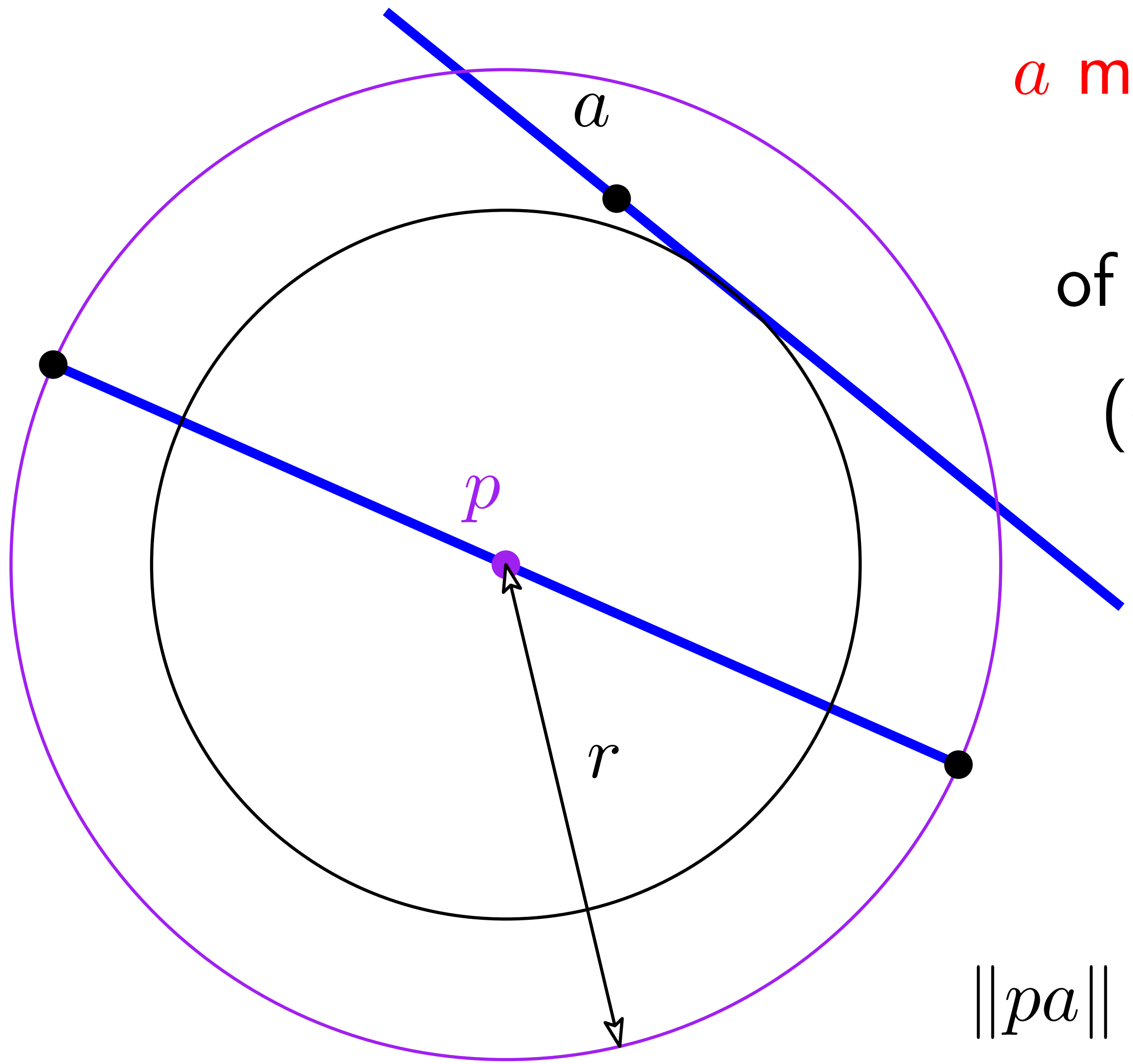
of non incident segment

($\Leftarrow$angle condition)

$$\|pa\| \geq \mathsf{lfs}(p) \geq \frac{1}{C_S}\mathsf{lfs}(p)$$

# Meshing

Delaunay mesh refinement [Ruppert]




$a$ midpoint

$\|pa\| \geq \frac{1}{C_S}\mathsf{lfs}(p)$

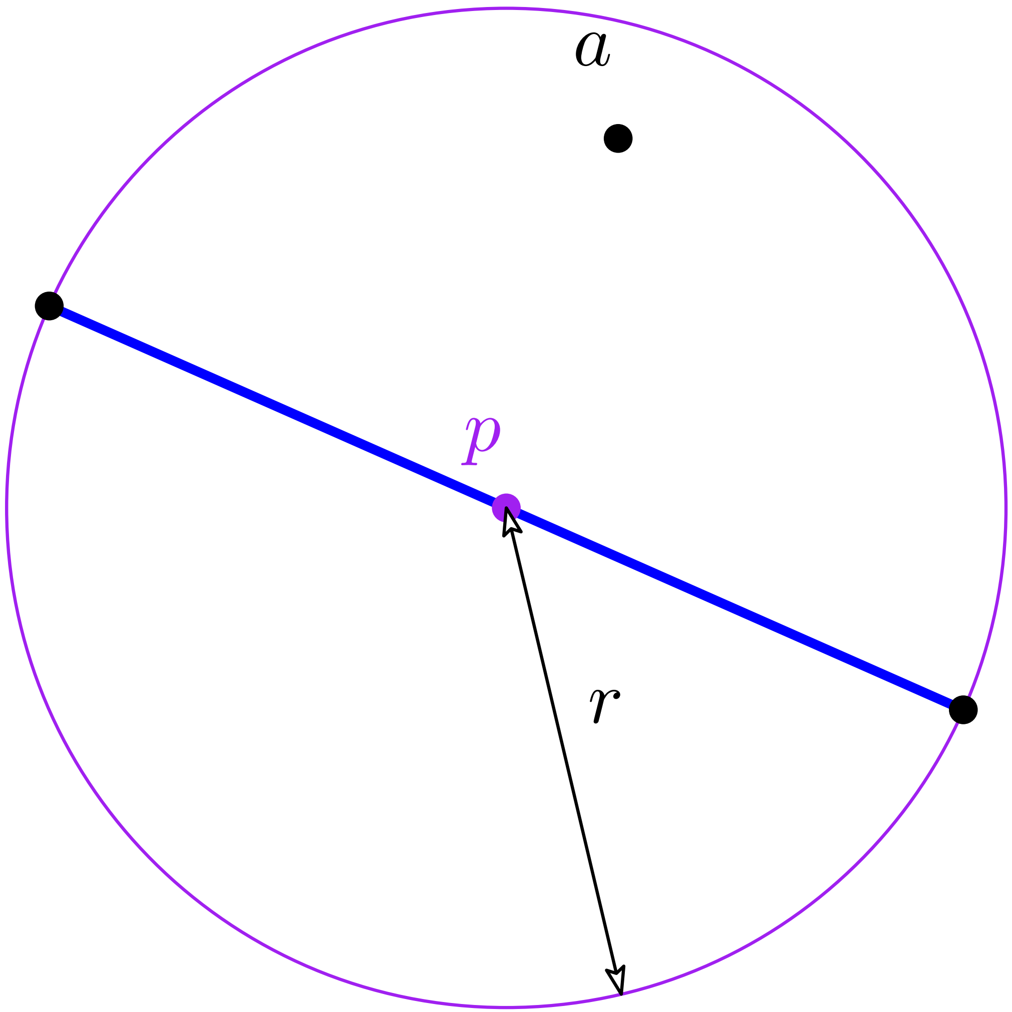
of non incident segment

($\Leftarrow$angle condition)

$\|pa\| \geq \mathsf{lfs}(p) \geq \frac{1}{C_S}\mathsf{lfs}(p)$

# Meshing

Delaunay mesh refinement [Ruppert]



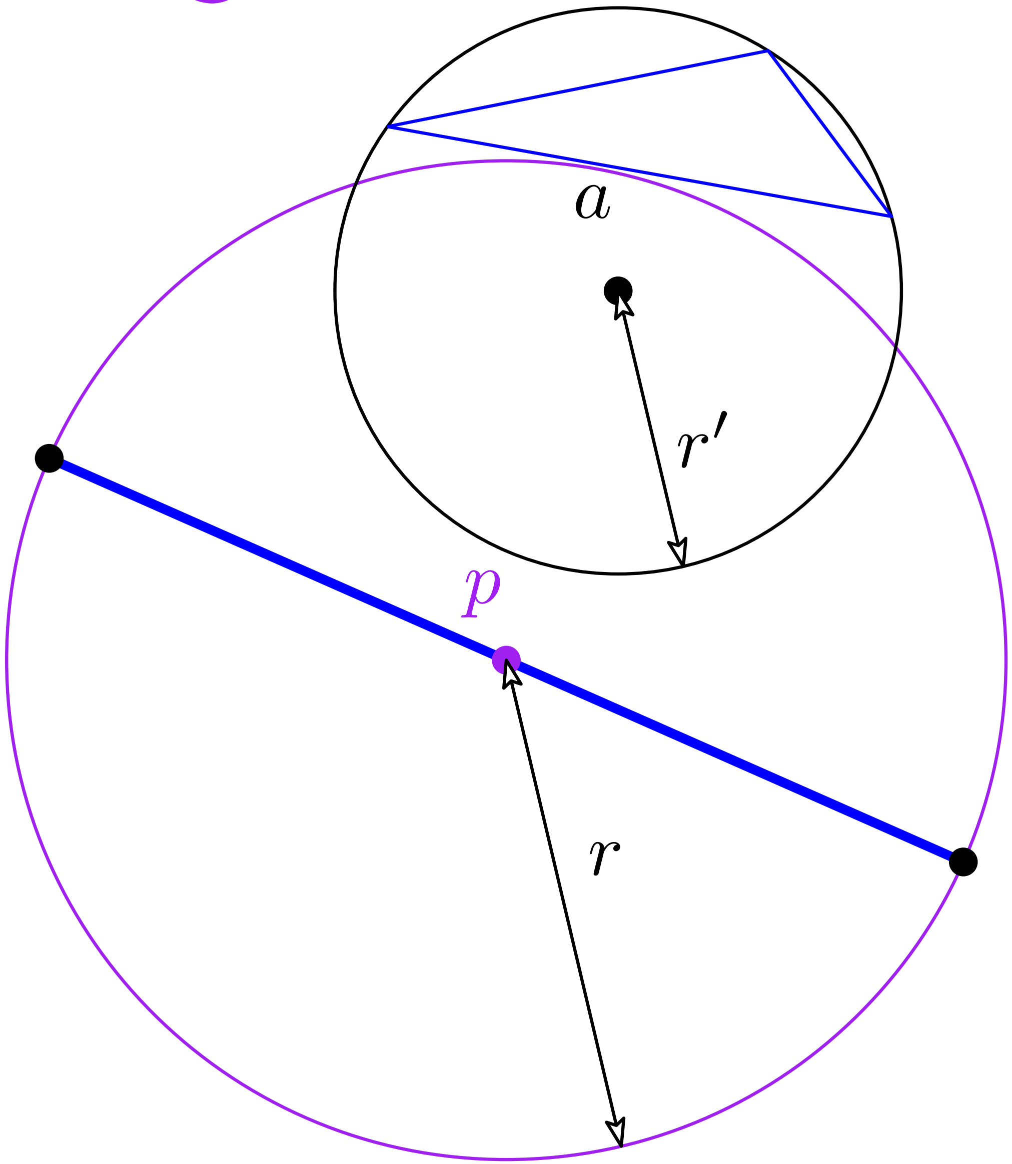

$a$ midpoint

$$\|pa\| \geq \frac{1}{C_S}\mathsf{lfs}(p)$$

$a$ circumcenter(not inserted)

# Meshing

Delaunay mesh refinement [Ruppert]




$a$ midpoint

$$\|pa\| \geq \frac{1}{C_S}\mathsf{lfs}(p)$$

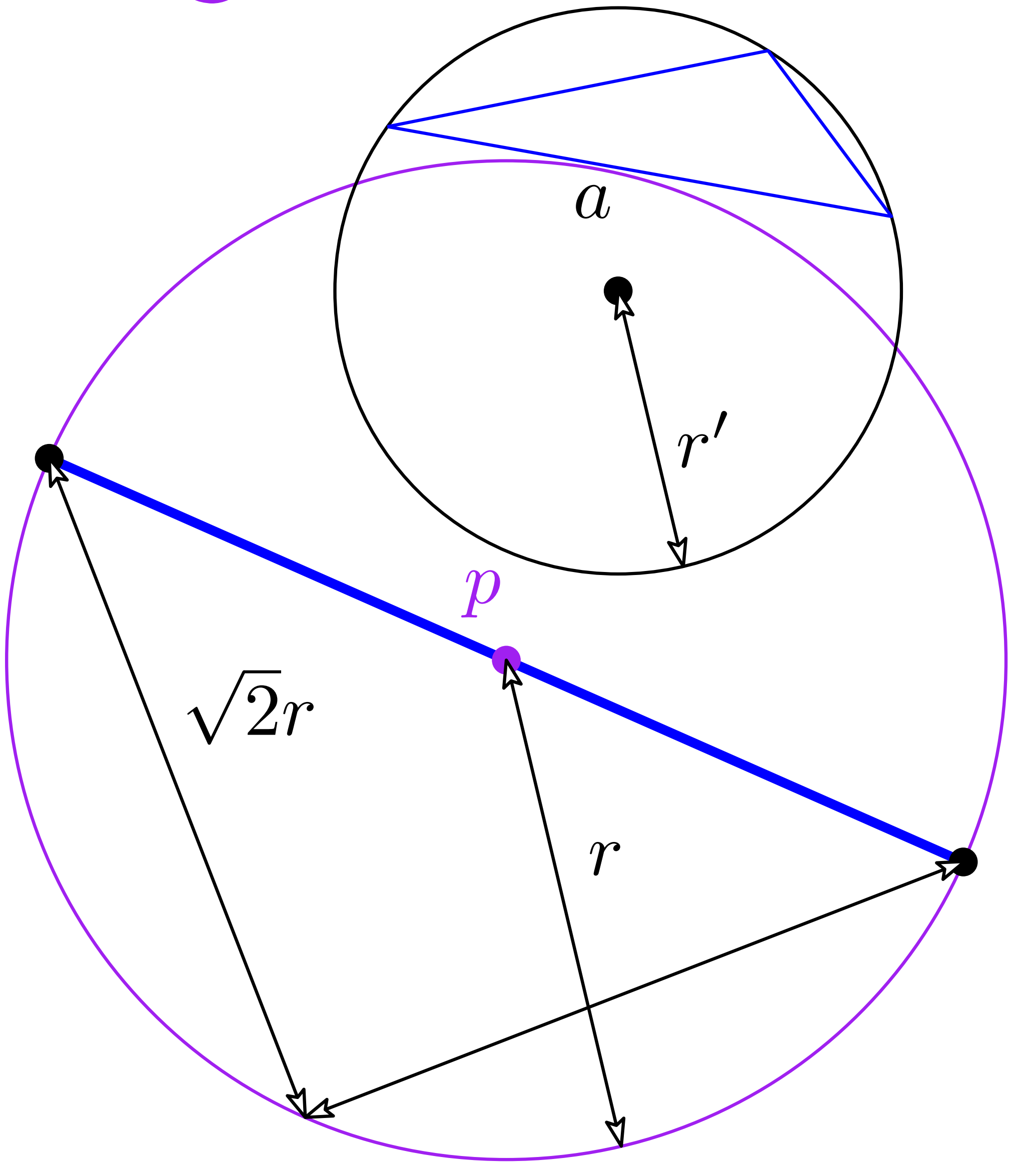
$a$ circumcenter (not inserted)

Induction:

$$r' \geq \frac{1}{C_T}\mathsf{lfs}(a)$$

# Meshing

Delaunay mesh refinement [Ruppert]




$a$ midpoint

$$\|pa\| \geq \frac{1}{C_S}\mathsf{lfs}(p)$$

$a$ circumcenter(not inserted)

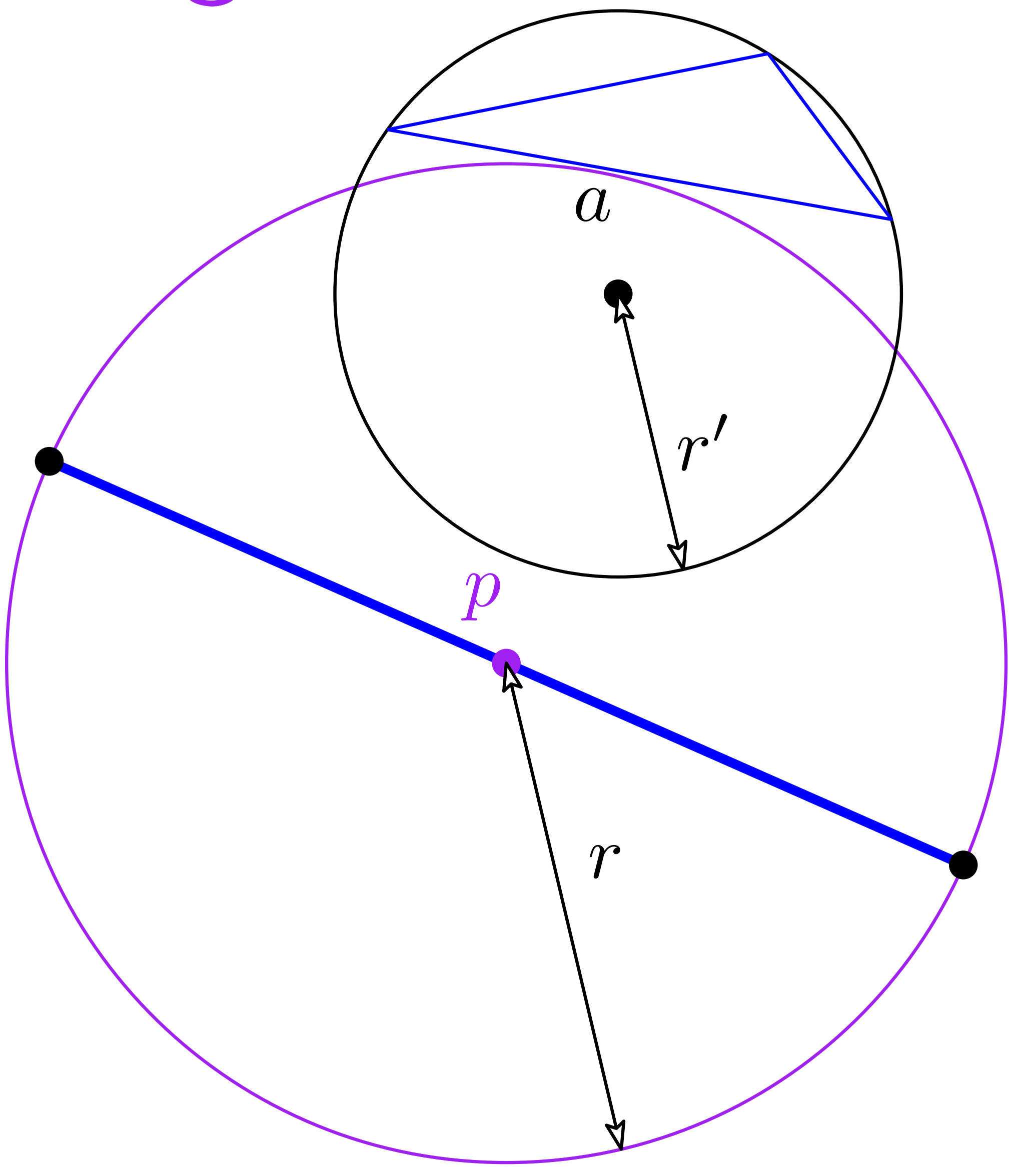
Induction:

$$r' \geq \frac{1}{C_T}\mathsf{lfs}(a)$$

$$r' \leq \sqrt{2}r$$

# Meshing

Delaunay mesh refinement [Ruppert]




$a$ midpoint

$$\|pa\| \geq \frac{1}{C_S}\mathsf{lfs}(p)$$

$a$ circumcenter(not inserted)

Induction:

$$r' \geq \frac{1}{C_T}\mathsf{lfs}(a)$$

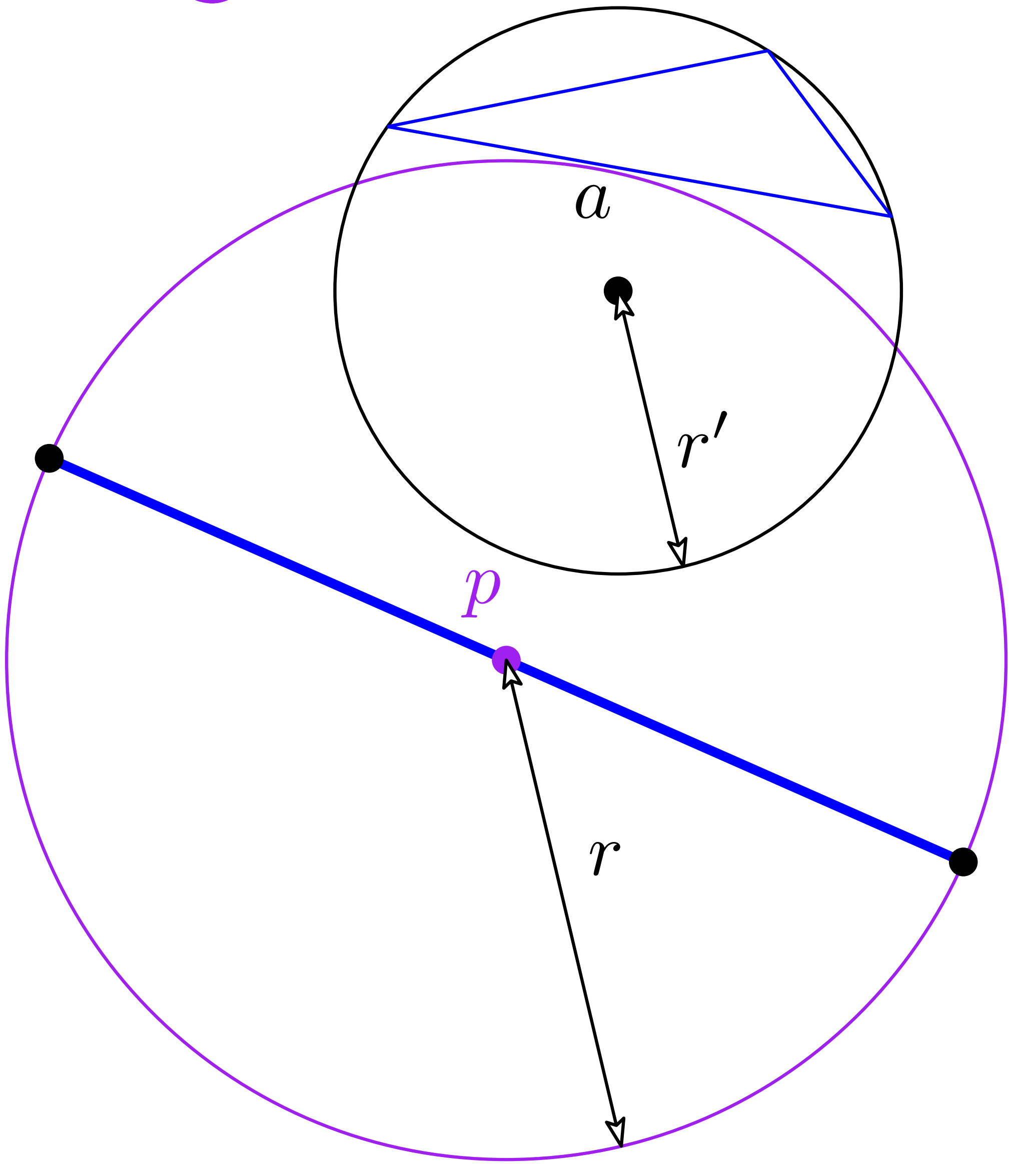
$$r' \leq \sqrt{2}r$$

$$\mathsf{lfs}(p \leq \mathsf{lfs}(a) + r$$

Lemma

# Meshing

Delaunay mesh refinement [Ruppert]




$a$ midpoint

$$\|pa\| \geq \frac{1}{C_S}\mathsf{lfs}(p)$$

$a$ circumcenter(not inserted)

Induction:

$$r' \geq \frac{1}{C_T}\mathsf{lfs}(a)$$

$$r' \leq \sqrt{2}r$$

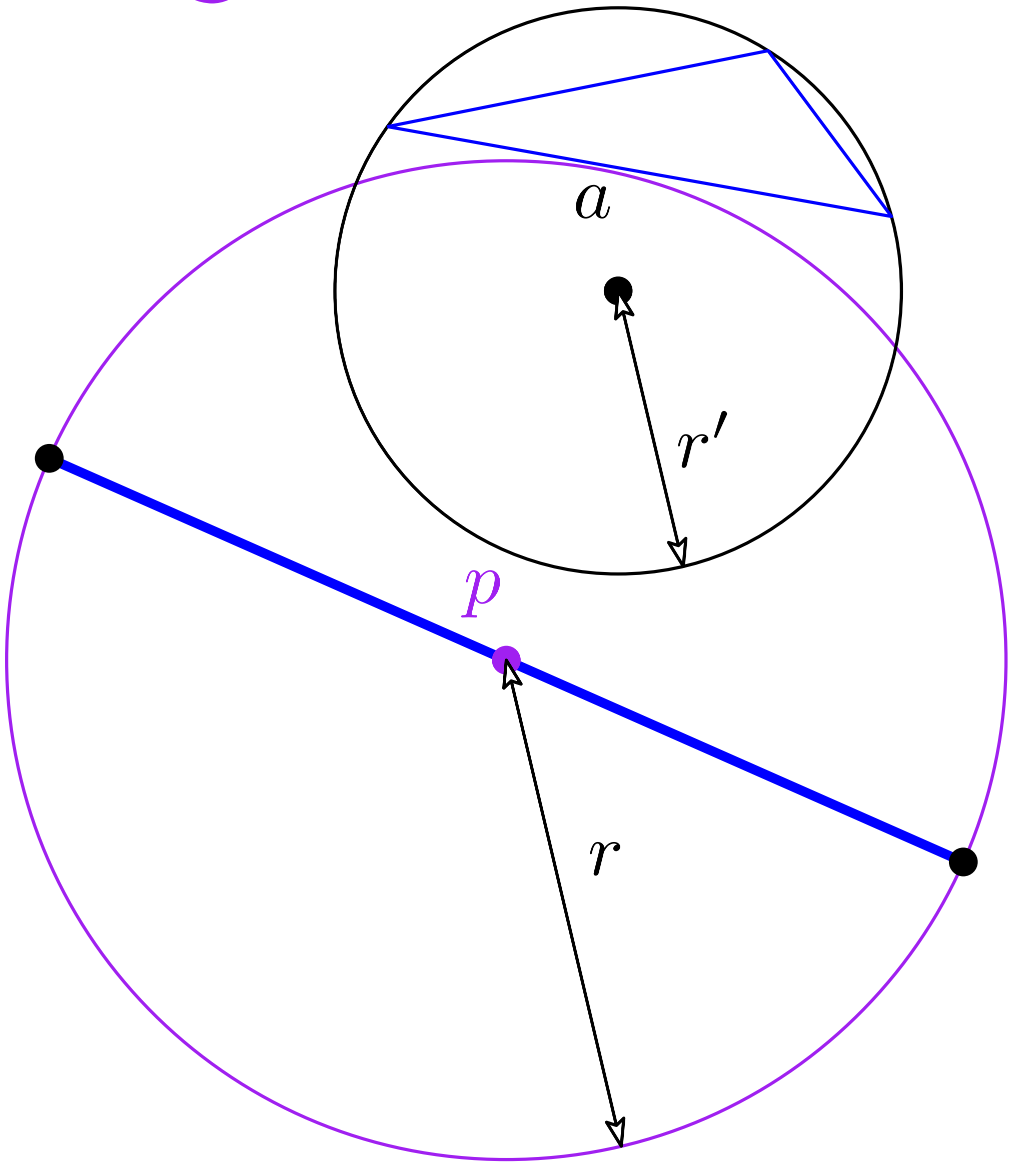
Lemma

$$\mathsf{lfs}(p \leq \mathsf{lfs}(a) + r$$

$$\leq r' C_T + r$$

# Meshing

Delaunay mesh refinement [Ruppert]




$a$ midpoint

$$\|pa\| \geq \frac{1}{C_S}\mathsf{lfs}(p)$$

$a$ circumcenter(not inserted)

Induction:

$$r' \geq \frac{1}{C_T}\mathsf{lfs}(a)$$

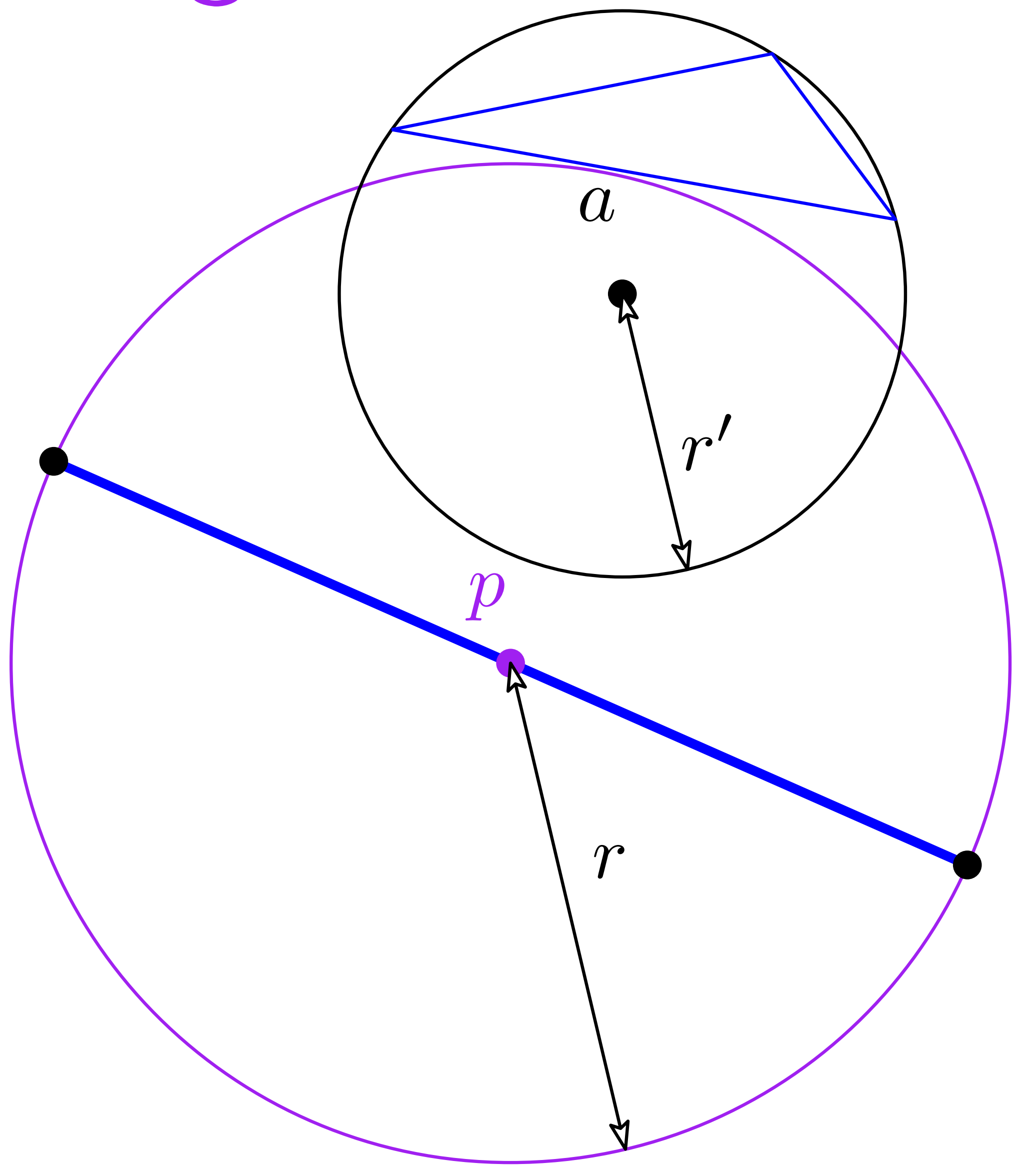
$$r' \leq \sqrt{2}r$$

Lemma

$$\begin{aligned}\mathsf{lfs}(p &\leq \mathsf{lfs}(a) + r \\ &\leq r'C_T + r \\ &\leq r(\sqrt{2}C_T + 1)\end{aligned}$$

# Meshing

## Delaunay mesh refinement [Ruppert]




$a$ midpoint

$$\|pa\| \geq \frac{1}{C_S}\mathsf{lfs}(p)$$

$a$ circumcenter(not inserted)

Induction:

$$r' \geq \frac{1}{C_T}\mathsf{lfs}(a)$$

$$r' \leq \sqrt{2}r$$

Lemma

$$\mathsf{lfs}(p \leq \mathsf{lfs}(a) + r$$
$$\leq r'C_T + r$$
$$\leq r(\sqrt{2}C_T + 1)$$

OK if $C_S \geq 1 + \sqrt{2}C_T$

| |
|---|
| OK if $C_T \geq 1 + 2C_S \sin\alpha$ |
| OK if $C_S \geq 1 + \sqrt{2}C_T$ |

# Meshing

Delaunay mesh refinement [Ruppert]

| |
|---|
| OK if $C_T \geq 1 + 2C_S \sin\alpha$ |
| OK if $C_S \geq 1 + \sqrt{2}C_T$ |

$$C_T \geq 1 + 2(1 + \sqrt{2}C_T)\sin\alpha$$

$$\frac{C_T - 1}{2(1+\sqrt{2}C_T)} \geq \sin\alpha$$

# Meshing

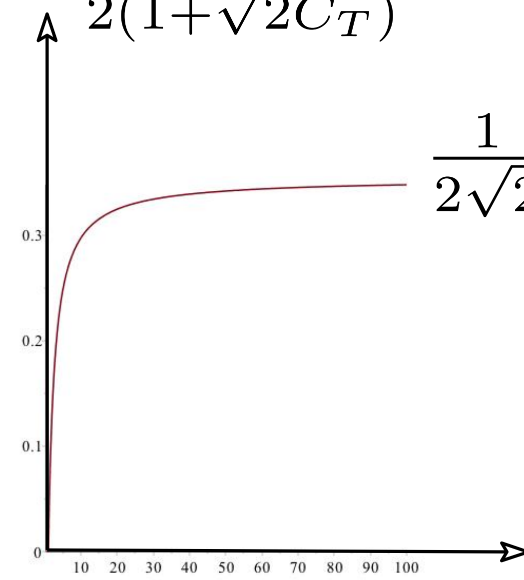
Delaunay mesh refinement [Ruppert]

| |
|---|
| OK if $C_T \geq 1 + 2C_S \sin\alpha$ |
| OK if $C_S \geq 1 + \sqrt{2}C_T$ |

$$C_T \geq 1 + 2(1+\sqrt{2}C_T)\sin\alpha$$

$$\frac{C_T - 1}{2(1+\sqrt{2}C_T)} \geq \sin\alpha$$

$$\frac{C_T - 1}{2(1+\sqrt{2}C_T)}$$




$$\frac{1}{2\sqrt{2}} \simeq \sin 20.7°$$

$C_T$

| |
|---|
| OK if $C_T \geq 1 + 2C_S \sin\alpha$ |
| OK if $C_S \geq 1 + \sqrt{2}C_T$ |

$$C_T \geq 1 + 2(1 + \sqrt{2}C_T)\sin\alpha$$

$$\frac{C_T - 1}{2(1+\sqrt{2}C_T)} \geq \sin\alpha$$

Choose $\alpha \leq 20°$

$$C_T := \frac{1+2\sin\alpha}{1-2\sqrt{2}\sin\alpha}$$

$$C_S := \frac{1+\sqrt{2}}{1-2\sqrt{2}\sin\alpha}$$

| |
|---|
| OK if $C_T \geq 1 + 2C_S \sin\alpha$ |
| OK if $C_S \geq 1 + \sqrt{2}C_T$ |

$$C_T \geq 1 + 2(1 + \sqrt{2}C_T)\sin\alpha$$

$$\frac{C_T - 1}{2(1+\sqrt{2}C_T)} \geq \sin\alpha$$

Choose $\alpha \leq 20^\circ$ $20^\circ$

$C_T := \frac{1+2\sin\alpha}{1-2\sqrt{2}\sin\alpha}$ 51

$C_S := \frac{1+\sqrt{2}}{1-2\sqrt{2}\sin\alpha}$ 74

# Meshing

Delaunay mesh refinement [Ruppert]

| |
|---|
| OK if $C_T \geq 1 + 2C_S \sin\alpha$ |
| OK if $C_S \geq 1 + \sqrt{2}C_T$ |

$$C_T \geq 1 + 2(1 + \sqrt{2}C_T)\sin\alpha$$

$$\frac{C_T - 1}{2(1+\sqrt{2}C_T)} \geq \sin\alpha$$

| Choose $\alpha \leq 20°$ | $20°$ | $10°$ |
|---|---|---|
| $C_T := \frac{1+2\sin\alpha}{1-2\sqrt{2}\sin\alpha}$ | 51 | 2.7 |
| $C_S := \frac{1+\sqrt{2}}{1-2\sqrt{2}\sin\alpha}$ | 74 | 4.8 |

Lemma: no vertex close to the last inserted vertex

# Meshing

Delaunay mesh refinement [Ruppert]

Lemma: no vertex close to the last inserted vertex

Theorem: no vertex close to another vertex

# Meshing

Delaunay mesh refinement [Ruppert]

Lemma: no vertex close to the last inserted vertex

Theorem: no vertex close to another vertex

$$\forall p, q \in \text{Output}; \; \|pq\| \geq \frac{1}{C_S+1}\mathsf{lfs}(p)$$

# Meshing

Delaunay mesh refinement [Ruppert]

Lemma: no vertex close to the last inserted vertex

Theorem: no vertex close to another vertex

$$\forall p, q \in \text{Output};\ \|pq\| \geq \frac{1}{C_S+1}\mathsf{lfs}(p)$$

$p$ after $q$

$$\|pq\| \geq \frac{1}{C_S}\mathsf{lfs}(p) \geq \frac{1}{C_S+1}\mathsf{lfs}(p)$$

# Meshing

Delaunay mesh refinement [Ruppert]

Lemma: no vertex close to the last inserted vertex

Theorem: no vertex close to another vertex

$$\forall p, q \in \text{Output};\ \|pq\| \geq \frac{1}{C_S+1}\mathsf{lfs}(p)$$

$p$ after $q$

$$\|pq\| \geq \frac{1}{C_S}\mathsf{lfs}(p) \geq \frac{1}{C_S+1}\mathsf{lfs}(p)$$

$q$ after $p$

$$\|pq\| \geq \frac{1}{C_S}\mathsf{lfs}(q) \geq \frac{\mathsf{lfs}(p)-\|pq\|}{C_S}$$

$$\|pq\| \geq \frac{1}{C_S+1}\mathsf{lfs}(p)$$

# Meshing

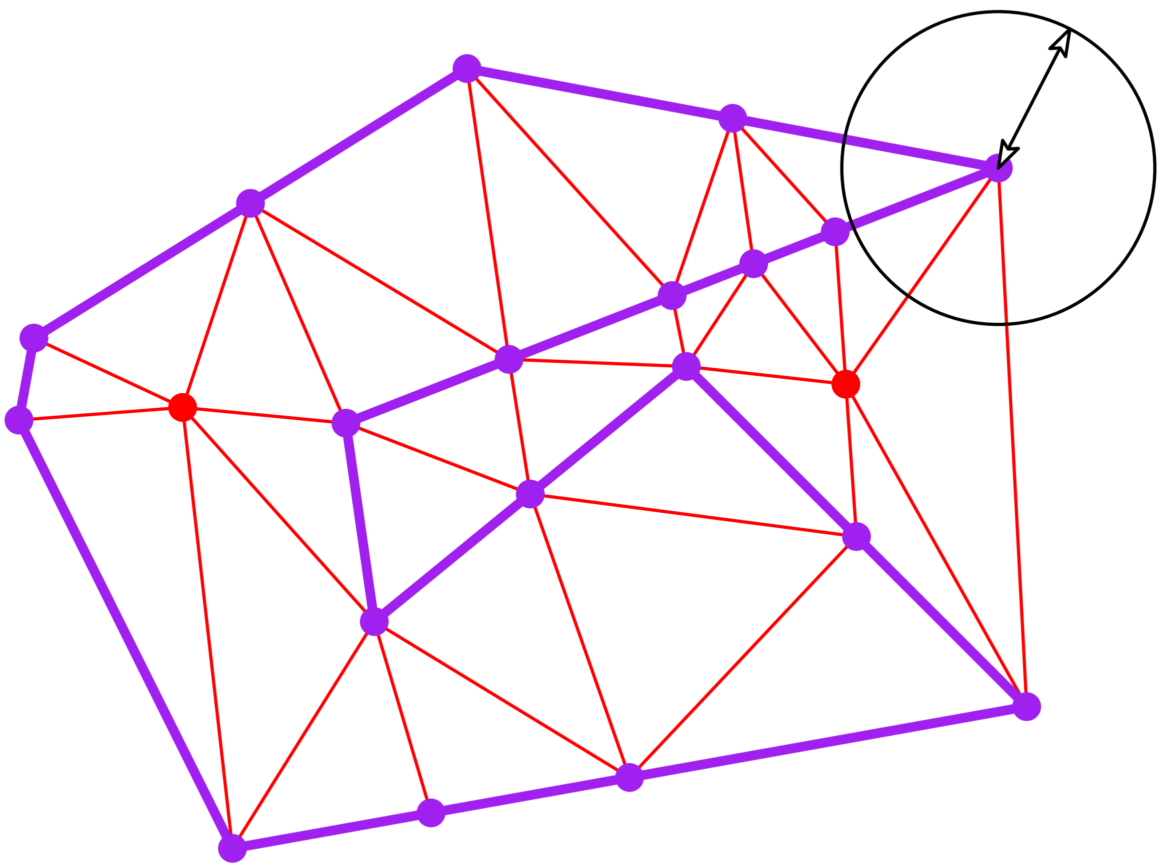
Delaunay mesh refinement [Ruppert]

Theorem: number of vertices in ouput is $O\left(\int \frac{1}{\mathrm{lfs}^2(x)} dx\right)$

# Meshing

Delaunay mesh refinement [Ruppert]

Theorem: number of vertices in ouput is $O\left(\int \frac{1}{\mathrm{lfs}^2(x)} dx\right)$



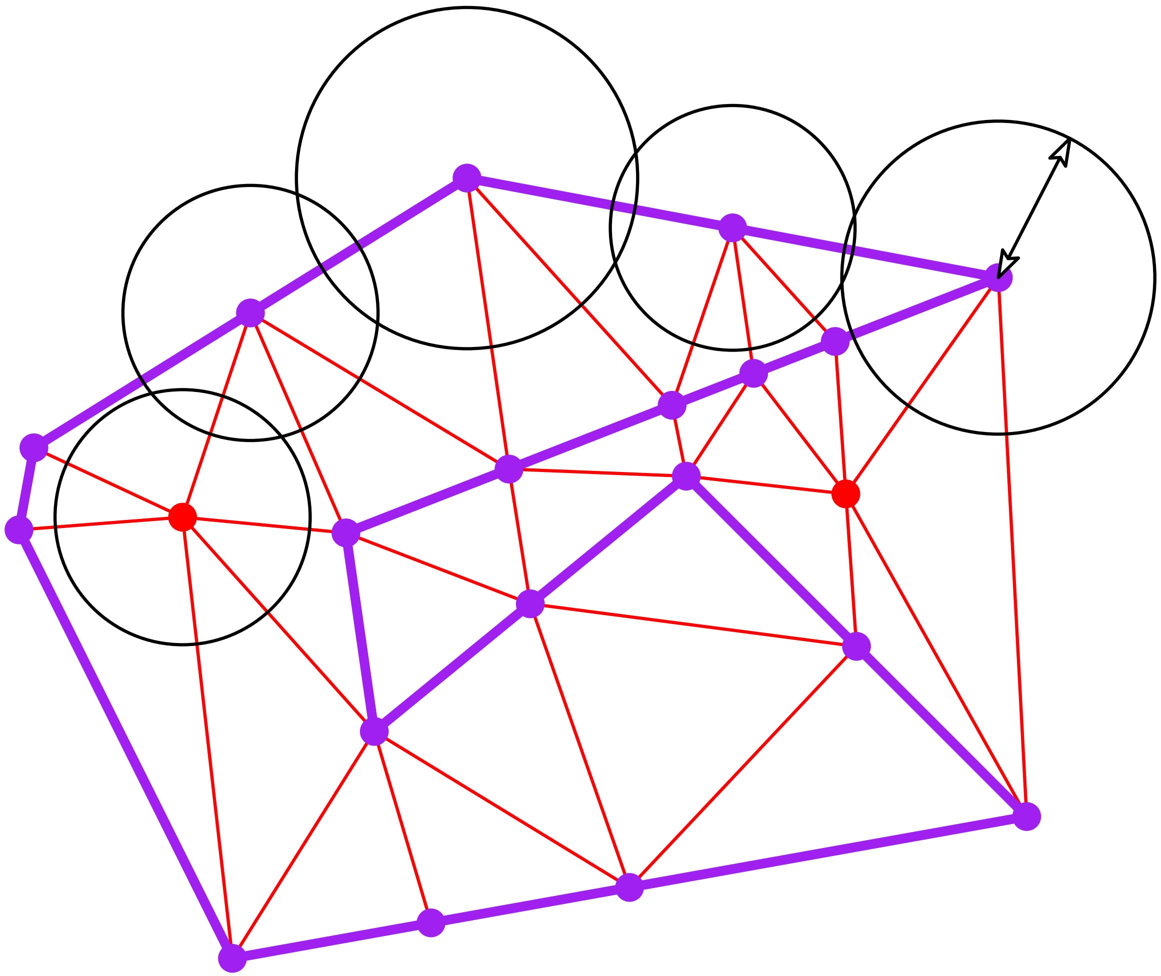
$\frac{1}{C_S+1}\mathsf{lfs}(p)$

is empty

# Meshing

Delaunay mesh refinement [Ruppert]

Theorem: number of vertices in ouput is $O\left(\int \frac{1}{\mathrm{lfs}^2(x)} dx\right)$



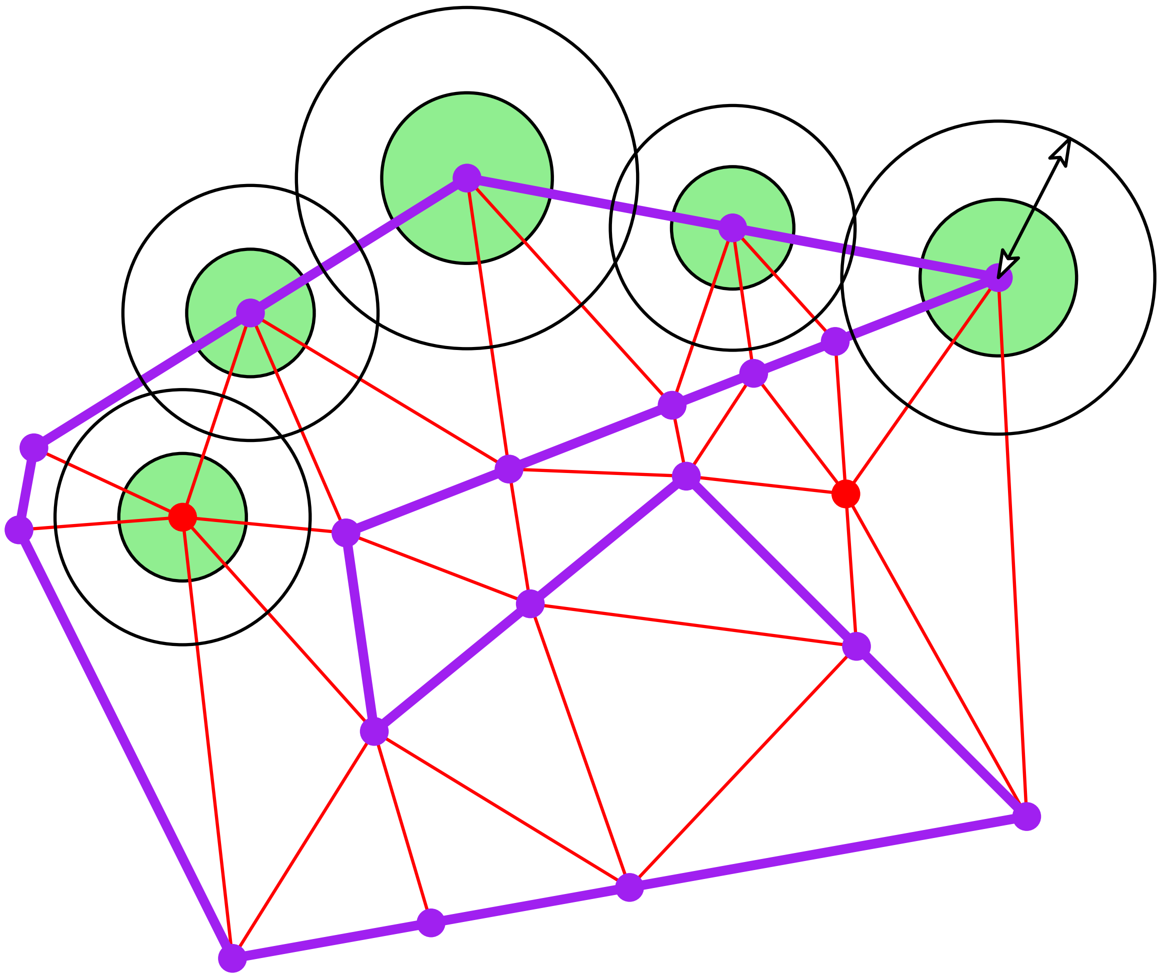
$\frac{1}{C_S+1}\mathsf{lfs}(p)$

is empty

# Meshing

Delaunay mesh refinement [Ruppert]

Theorem: number of vertices in ouput is $O\left(\int \frac{1}{\mathrm{lfs}^2(x)} dx\right)$



$\frac{1}{C_S+1}\mathsf{lfs}(p)$

is empty

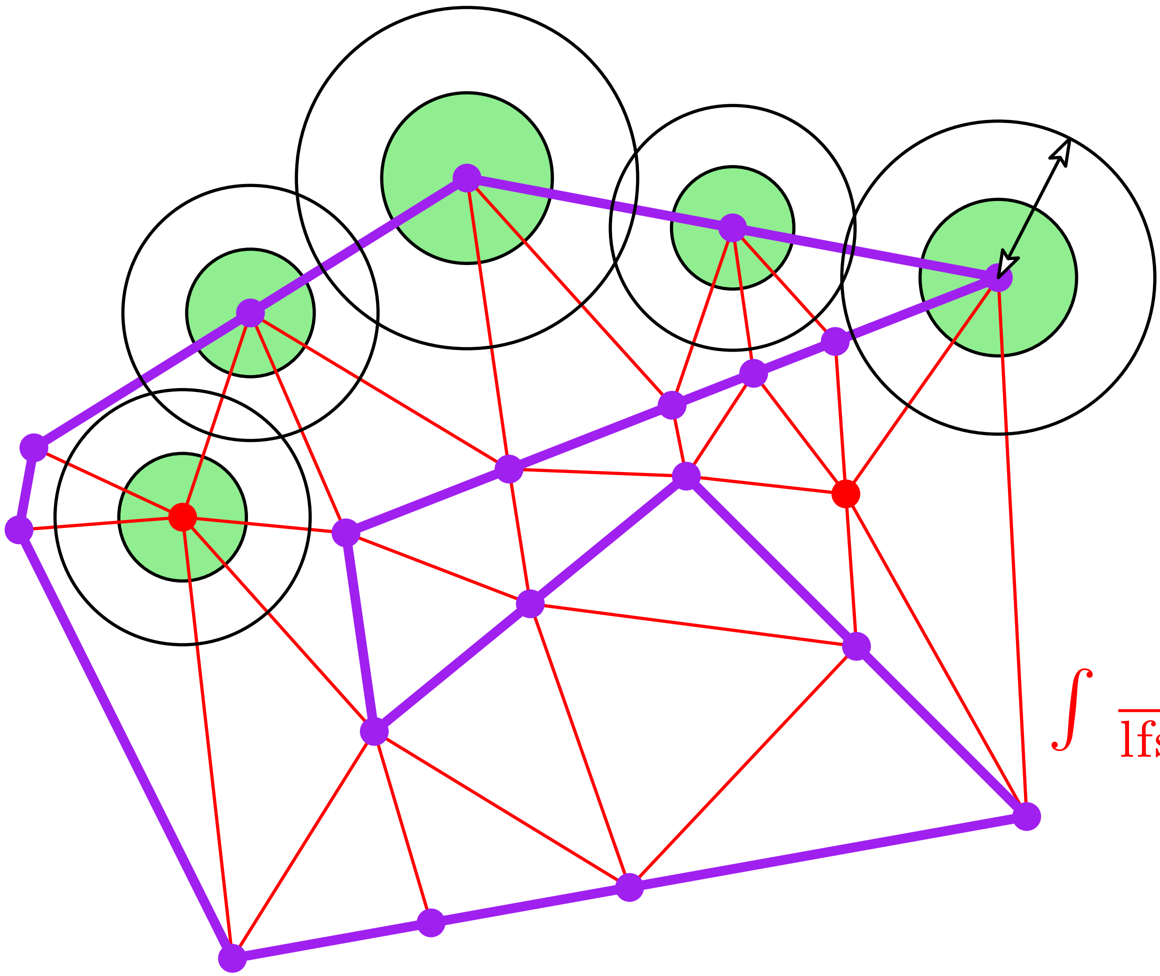
$\frac{1}{2} \cdot \frac{1}{C_S+1}\mathsf{lfs}(p)$

are disjoint

# Meshing

Delaunay mesh refinement [Ruppert]

Theorem: number of vertices in ouput is $O\left(\int \frac{1}{\mathrm{lfs}^2(x)} dx\right)$



$\frac{1}{C_S+1}\mathsf{lfs}(p)$

is empty

$\frac{1}{2} \cdot \frac{1}{C_S+1}\mathsf{lfs}(p)$

are disjoint

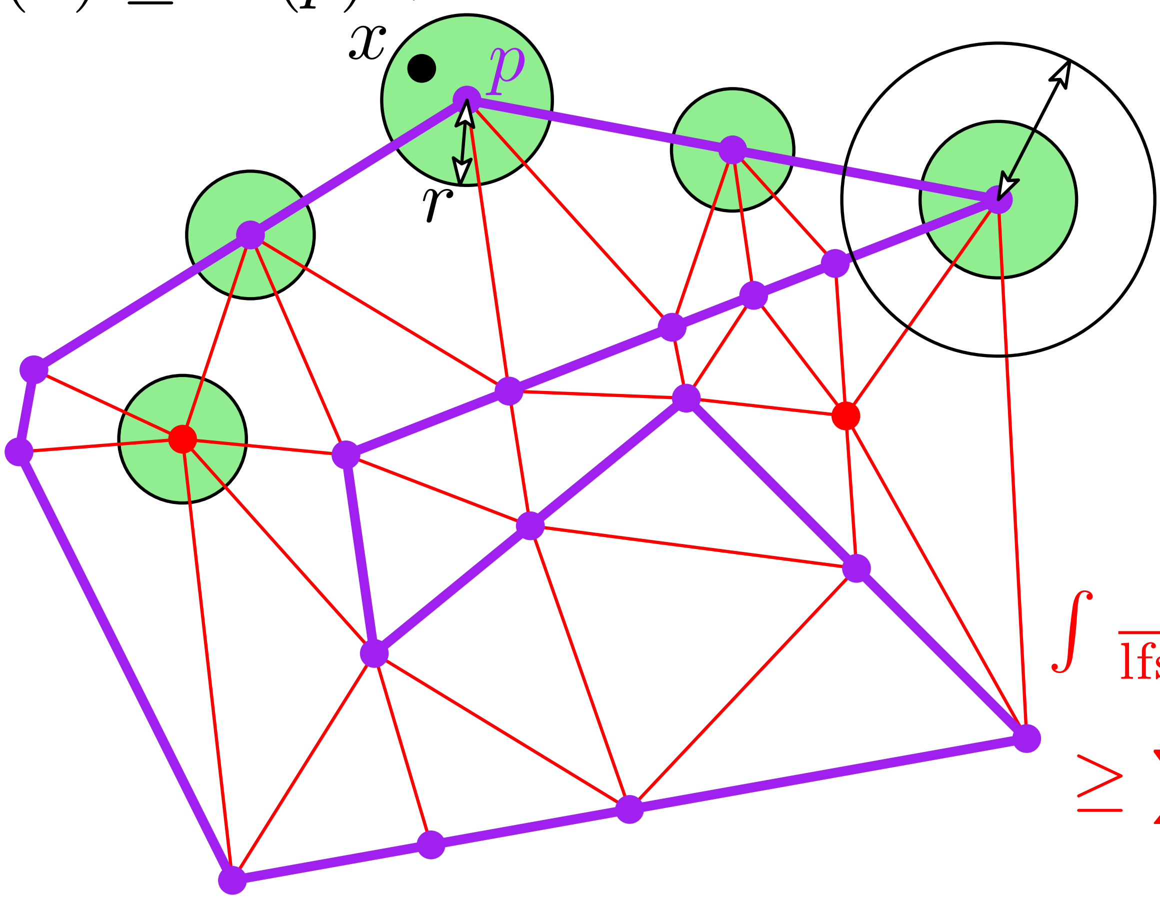
$$\int \frac{1}{\mathrm{lfs}^2(x)} dx \geq \int_{\mathrm{Disks}} \frac{1}{\mathrm{lfs}^2(x)} dx$$

# Meshing

Delaunay mesh refinement [Ruppert]

Theorem: number of vertices in ouput is $O\left(\int \frac{1}{\mathrm{lfs}^2(x)}dx\right)$

$\mathsf{lfs}(x) \leq \mathsf{lfs}(p) + r$




$\frac{1}{C_S+1}\mathsf{lfs}(p)$

is empty

$\frac{1}{2} \cdot \frac{1}{C_S+1}\mathsf{lfs}(p)$

are disjoint

$\int \frac{1}{\mathrm{lfs}^2(x)}dx \geq \int_{\mathrm{Disks}} \frac{1}{\mathrm{lfs}^2(x)}dx$

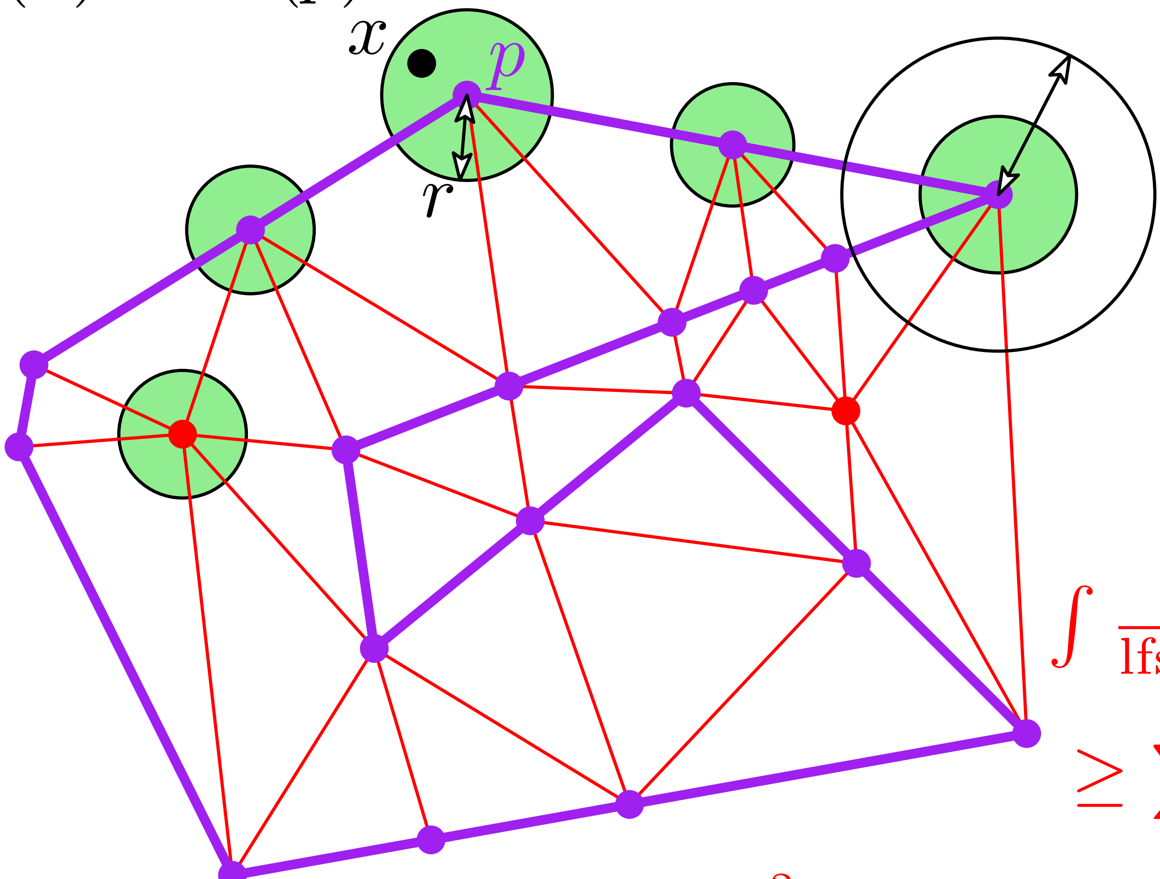
$\geq \sum_{\mathrm{Disks}} \int_{\mathrm{Disk}} \frac{1}{(\mathrm{lfs}(p)+r)^2}dx$

# Meshing

Delaunay mesh refinement [Ruppert]

Theorem: number of vertices in ouput is $O\left(\int \frac{1}{\mathrm{lfs}^2(x)}dx\right)$

$\mathsf{lfs}(x) \leq \mathsf{lfs}(p) + r$




$\frac{1}{C_S+1}\mathsf{lfs}(p)$ is empty

$\frac{1}{2}\cdot\frac{1}{C_S+1}\mathsf{lfs}(p)$ are disjoint

$\int \frac{1}{\mathrm{lfs}^2(x)}dx \geq \int_{\mathrm{Disks}} \frac{1}{\mathrm{lfs}^2(x)}dx$

$\geq \sum_{\mathrm{Disks}} \int_{\mathrm{Disk}} \frac{1}{(\mathrm{lfs}(p)+r)^2}dx$

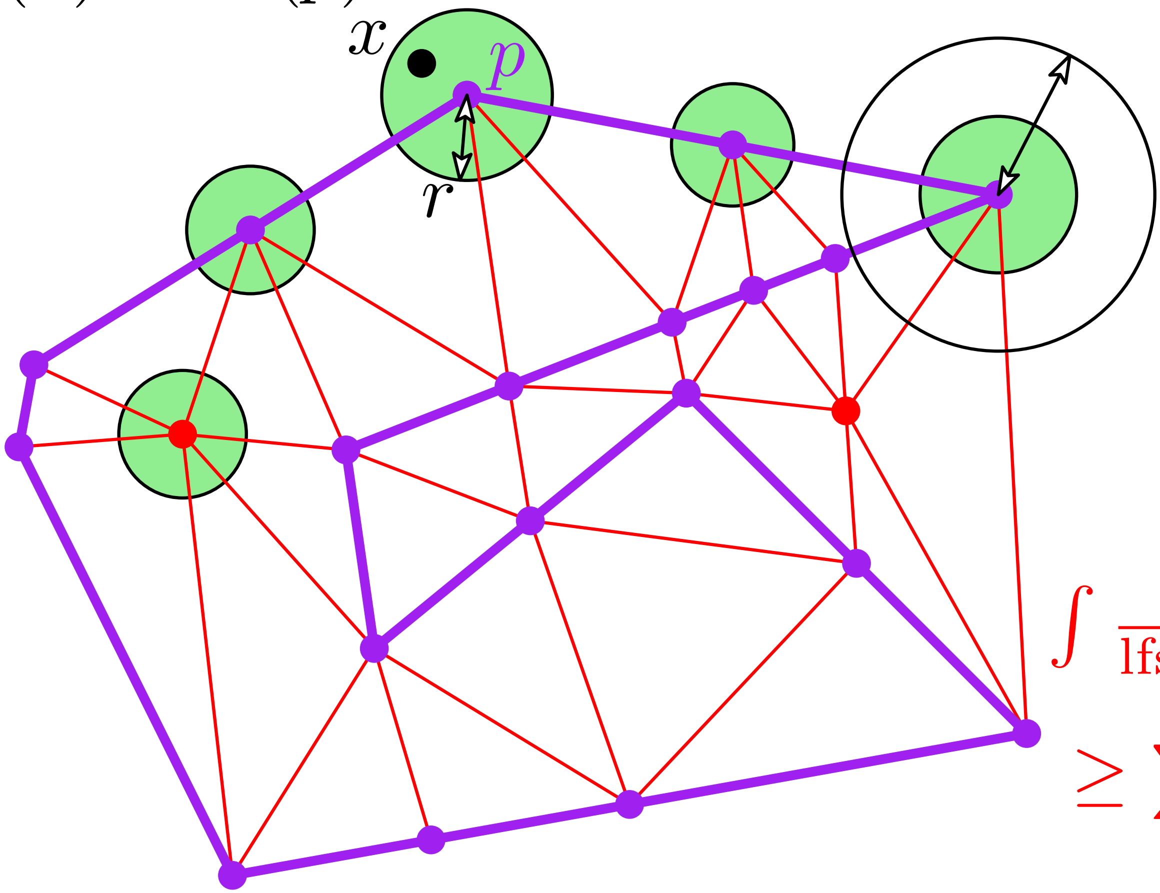
$\int_{\mathrm{Disk}} \frac{1}{(\mathrm{lfs}(p)+r)^2}dx = \frac{\pi r^2}{(\mathrm{lfs}(p)+r)^2} = \frac{\pi r^2}{(2(C_S+1)r+r)^2} = \frac{\pi}{2C_S+3}$

# Meshing

Delaunay mesh refinement [Ruppert]

Theorem: number of vertices in ouput is $O\left(\int \frac{1}{\text{lfs}^2(x)} dx\right)$

$\text{lfs}(x) \leq \text{lfs}(p) + r$




$\frac{1}{C_S+1}\text{lfs}(p)$

is empty

$\frac{1}{2} \cdot \frac{1}{C_S+1}\text{lfs}(p)$

are disjoint

$$\int \frac{1}{\text{lfs}^2(x)} dx \geq \int_{\text{Disks}} \frac{1}{\text{lfs}^2(x)} dx$$

$$\geq \sum_{\text{Disks}} \int_{\text{Disk}} \frac{1}{(\text{lfs}(p)+r)^2} dx$$

$$\geq \sharp\text{vertices} \frac{\pi}{2C_S+3}$$

# Meshing

Delaunay mesh refinement [Ruppert]

Optimality (up to a constant)

# Meshing

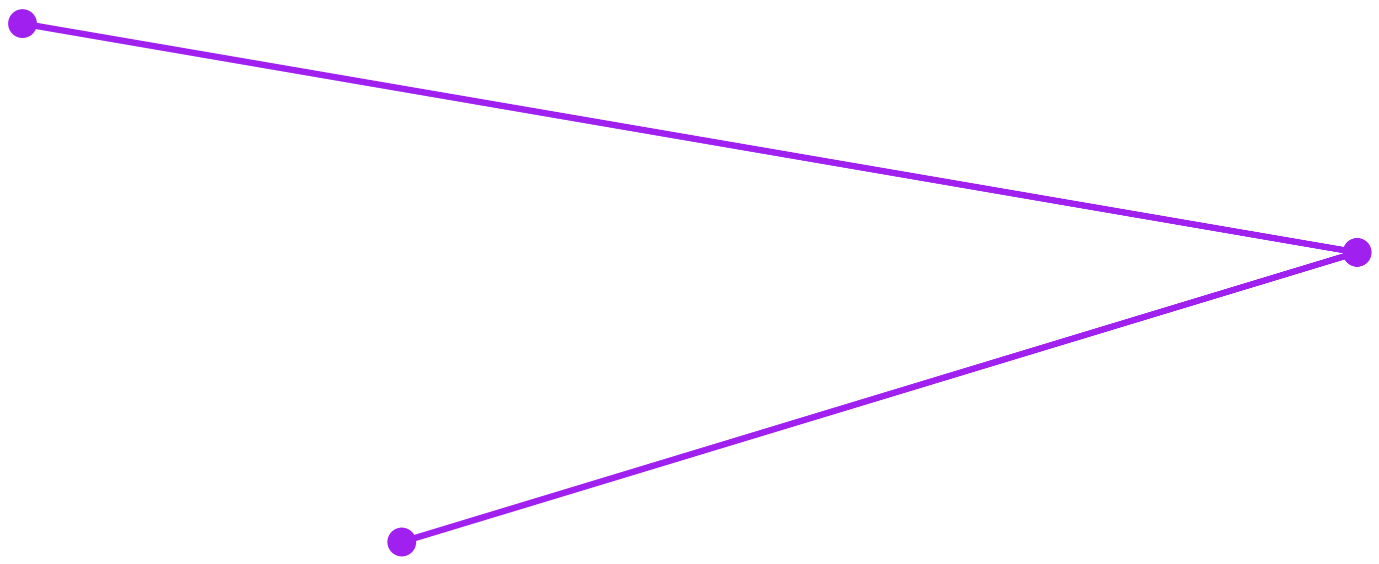


Assume no angles $\geq 90^\circ$ in input

# Meshing

Delaunay mesh refinement small angles

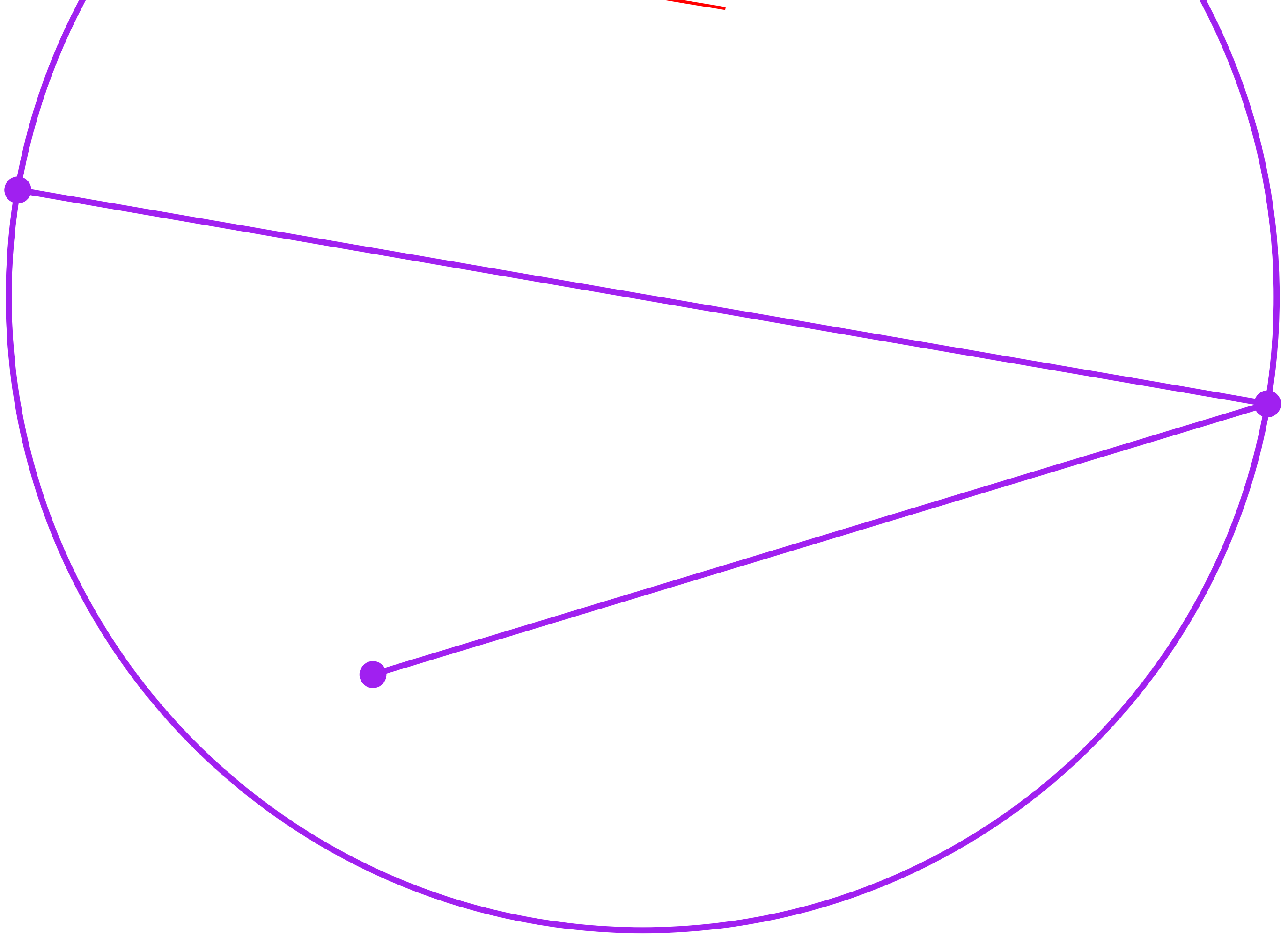
~~Assume no angles $\geq 90^\circ$ in input~~



# Meshing

Delaunay mesh refinement small angles

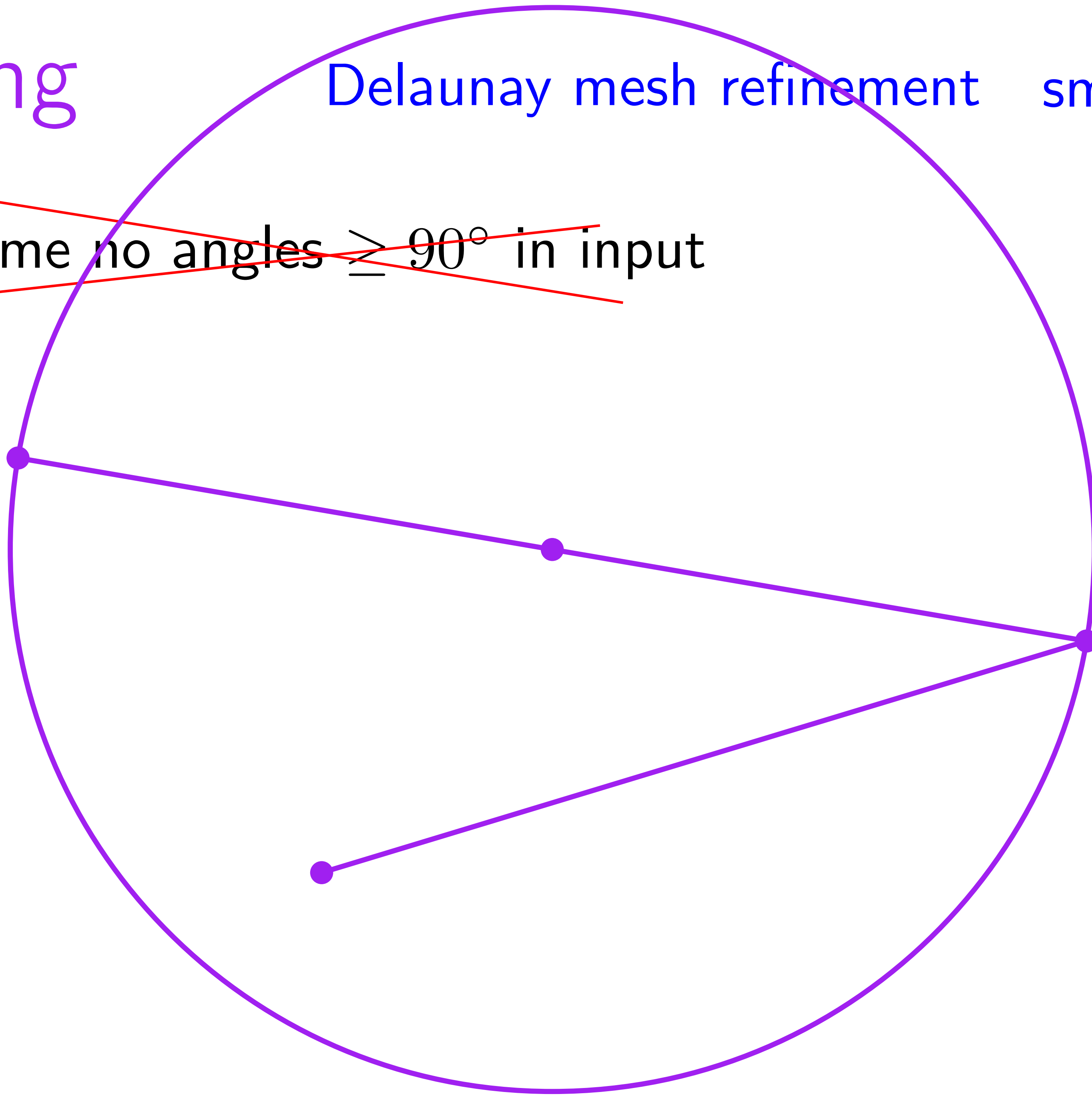
~~Assume no angles $\geq 90^\circ$ in input~~



# Meshing

Delaunay mesh refinement  small angles

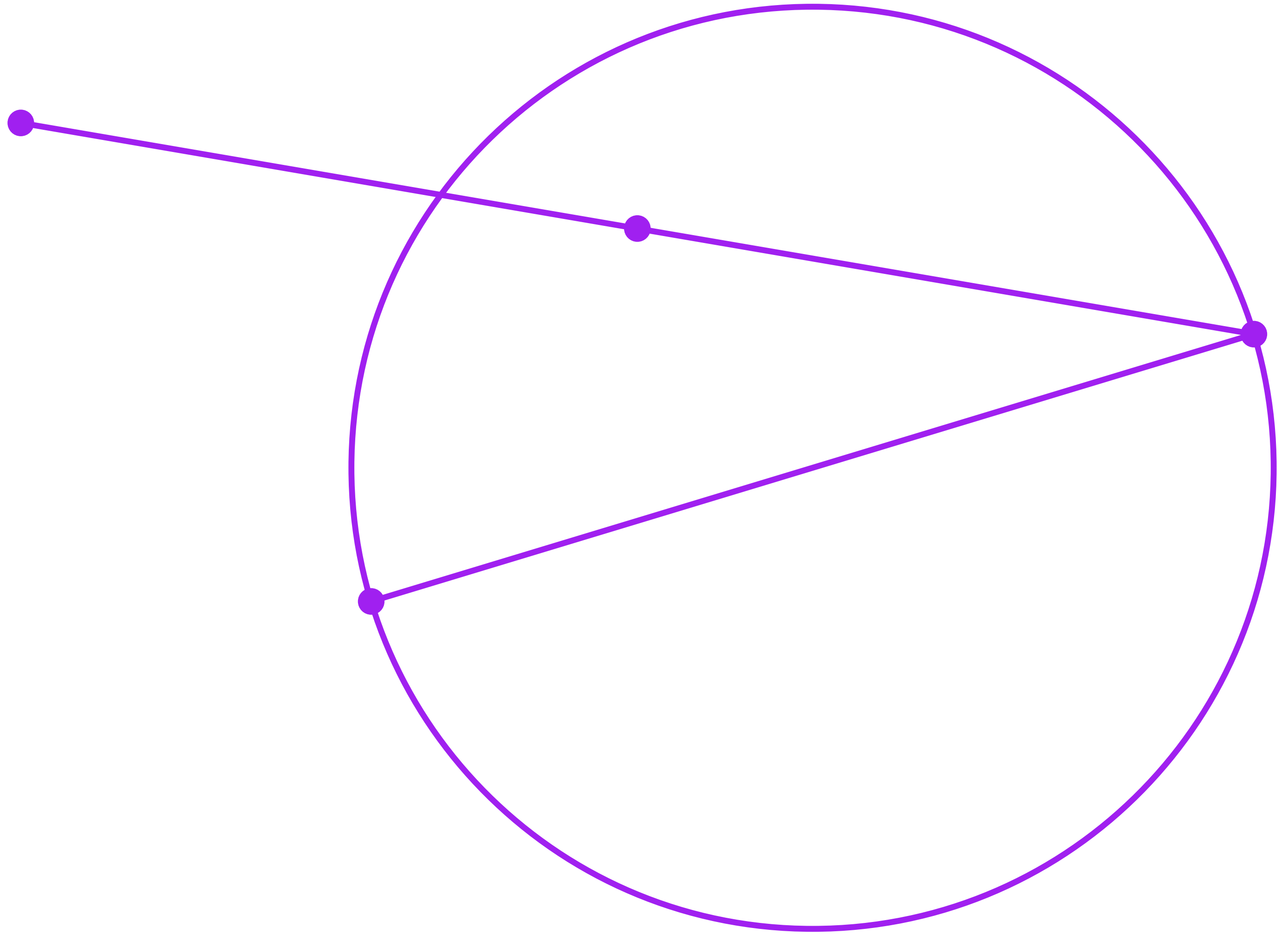
~~Assume no angles $\geq 90^\circ$ in input~~



# Meshing

Delaunay mesh refinement small angles

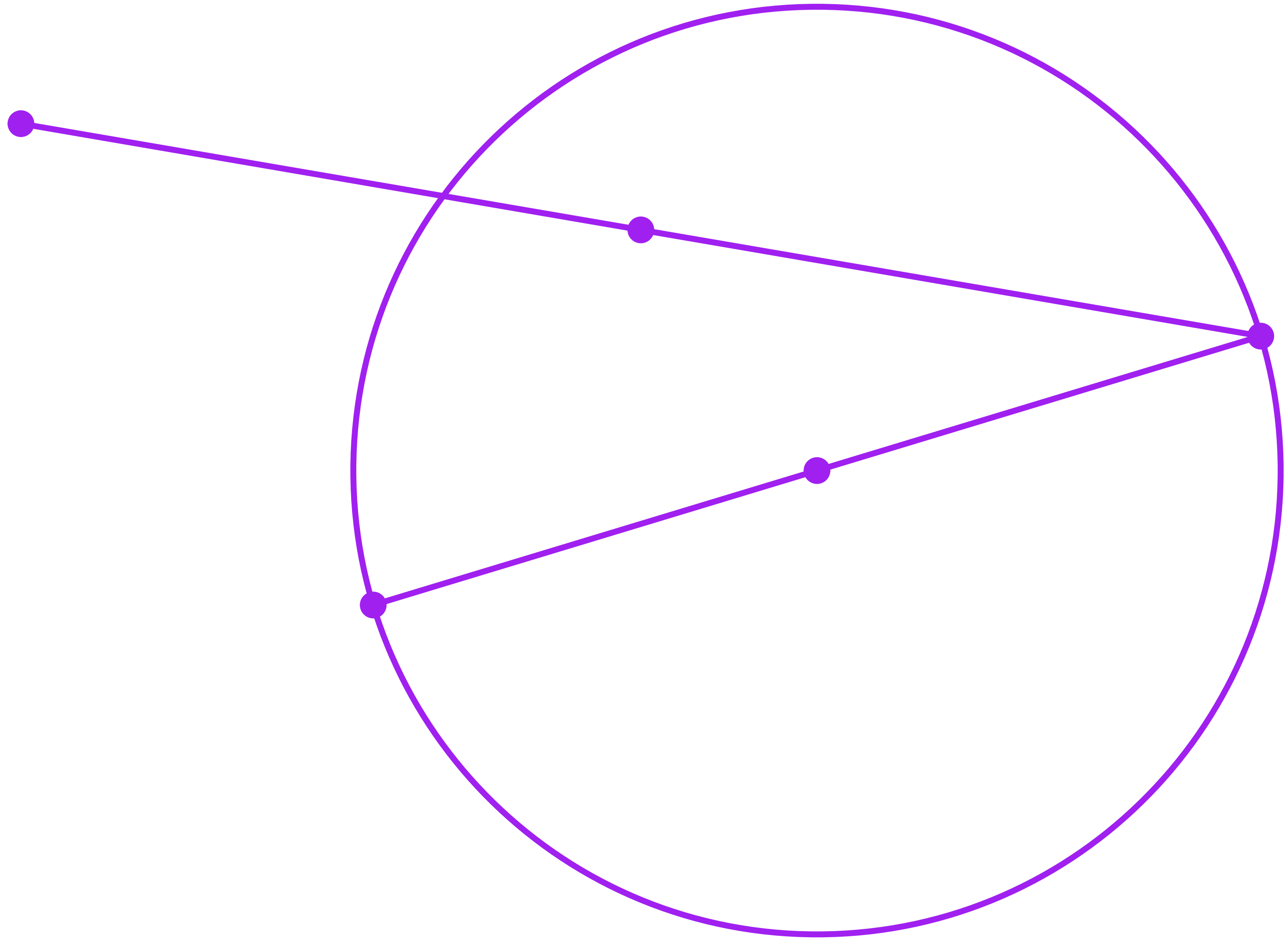
~~Assume no angles $\geq 90^\circ$ in input~~



# Meshing

Delaunay mesh refinement small angles

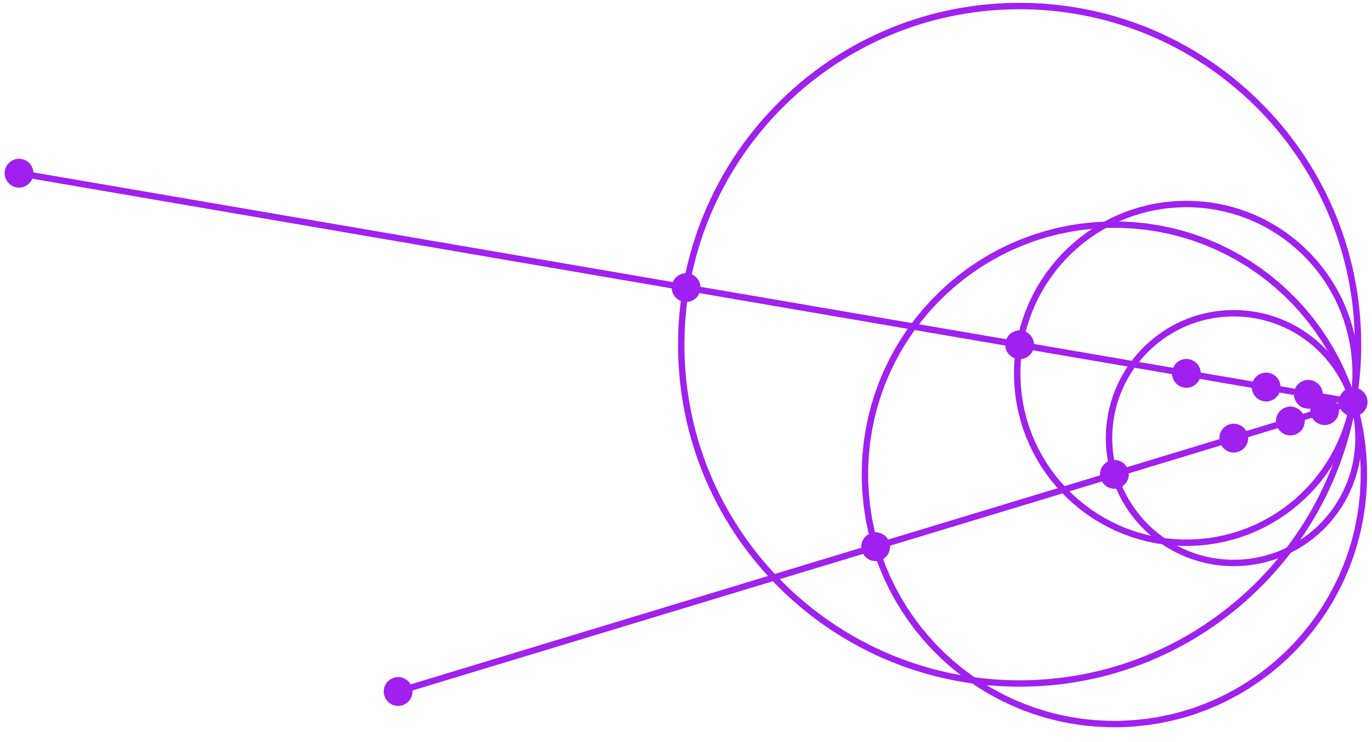
~~Assume no angles $\geq 90^\circ$ in input~~



# Meshing

Delaunay mesh refinement small angles

~~Assume no angles $\geq 90^\circ$ in input~~

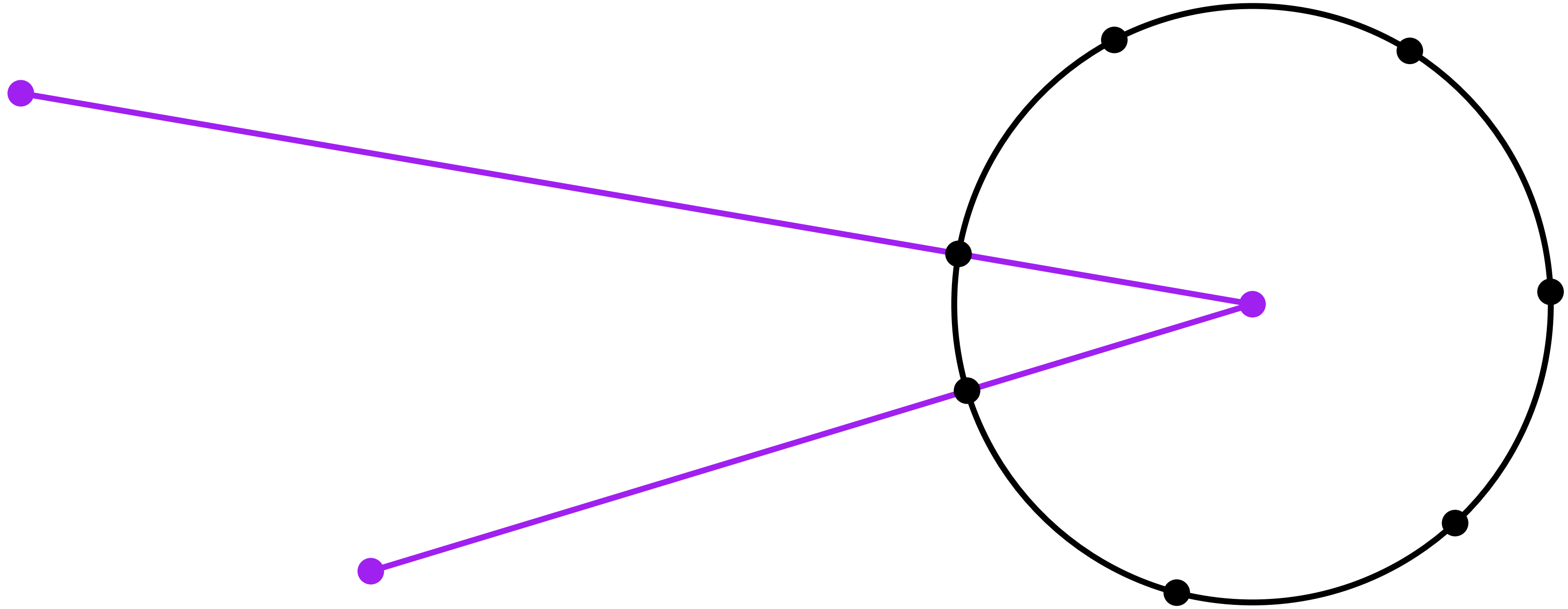


# Meshing



~~Assume no angles $\geq 90^\circ$ in input~~

Protect vertices

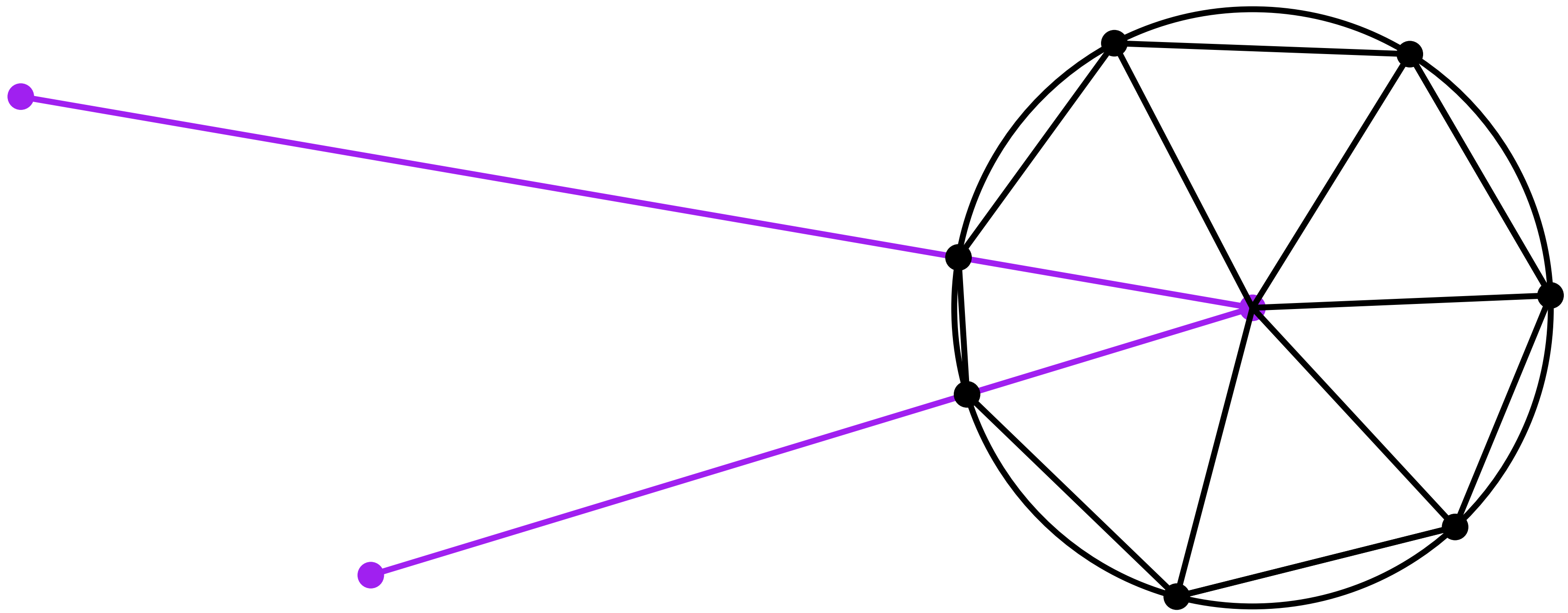


# Meshing

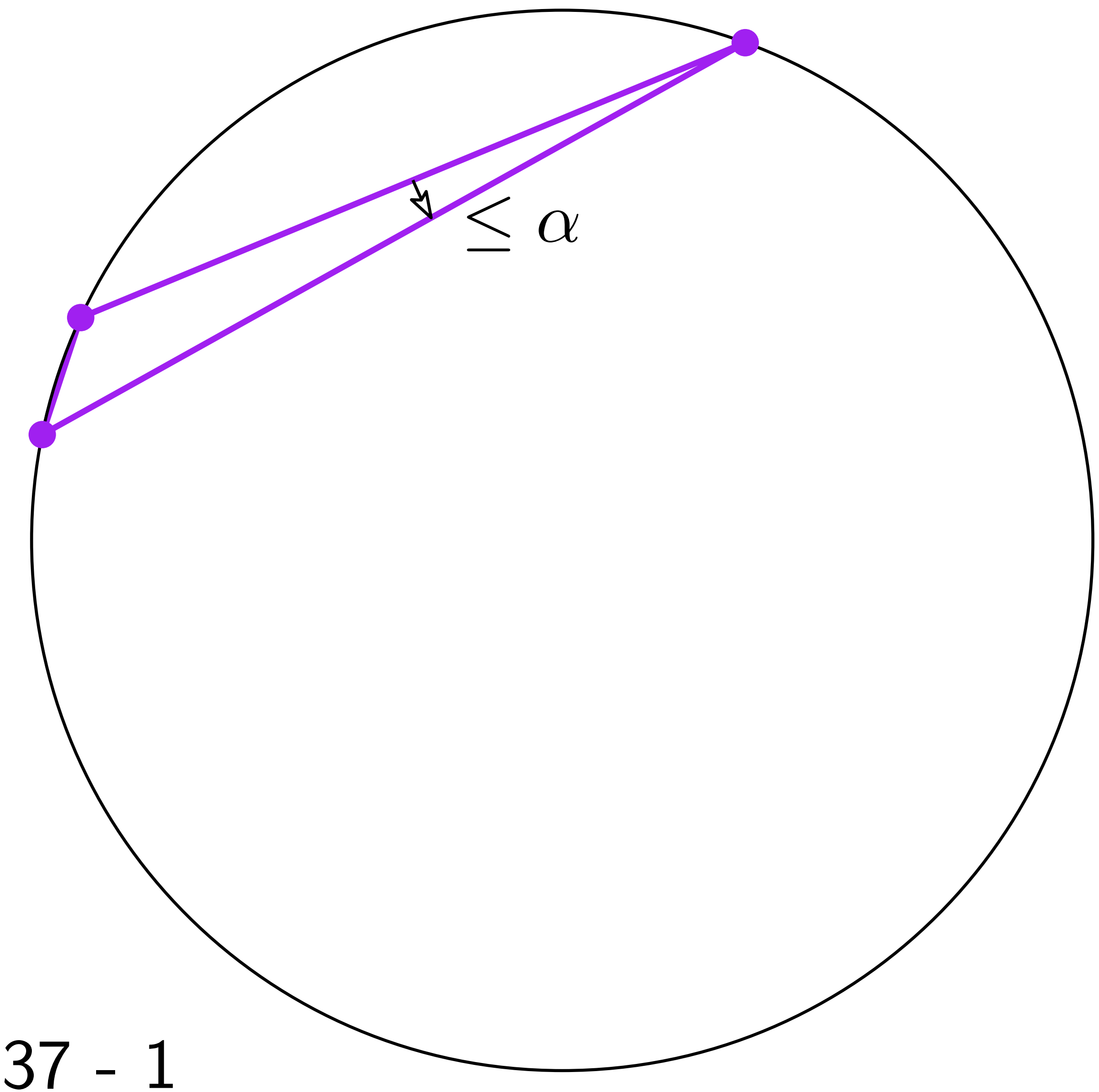
Delaunay mesh refinement small angles

~~Assume no angles $\geq 90^\circ$ in input~~

Protect vertices

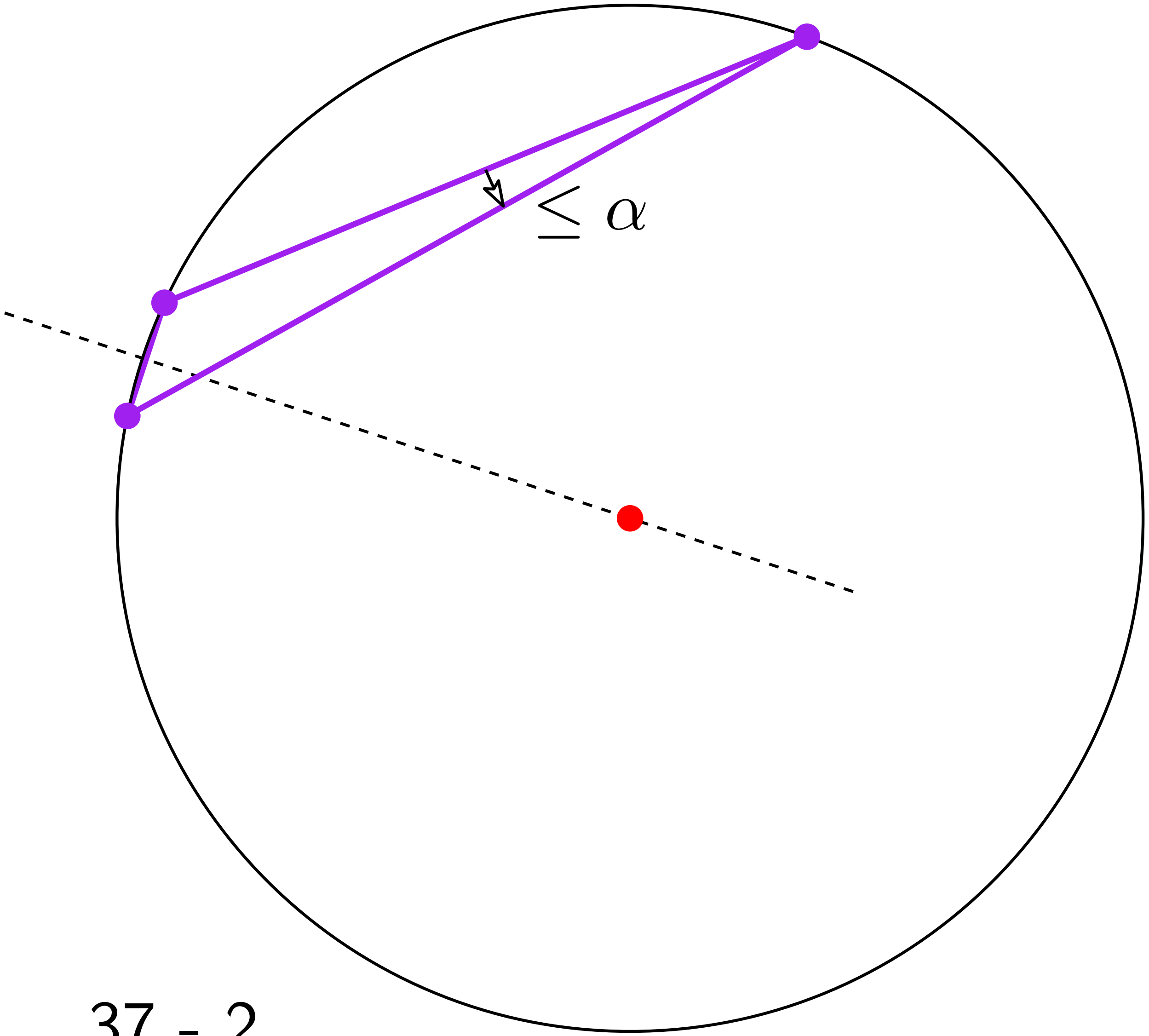


Very skinny triangle



# Meshing

Delaunay mesh refinement off-centers

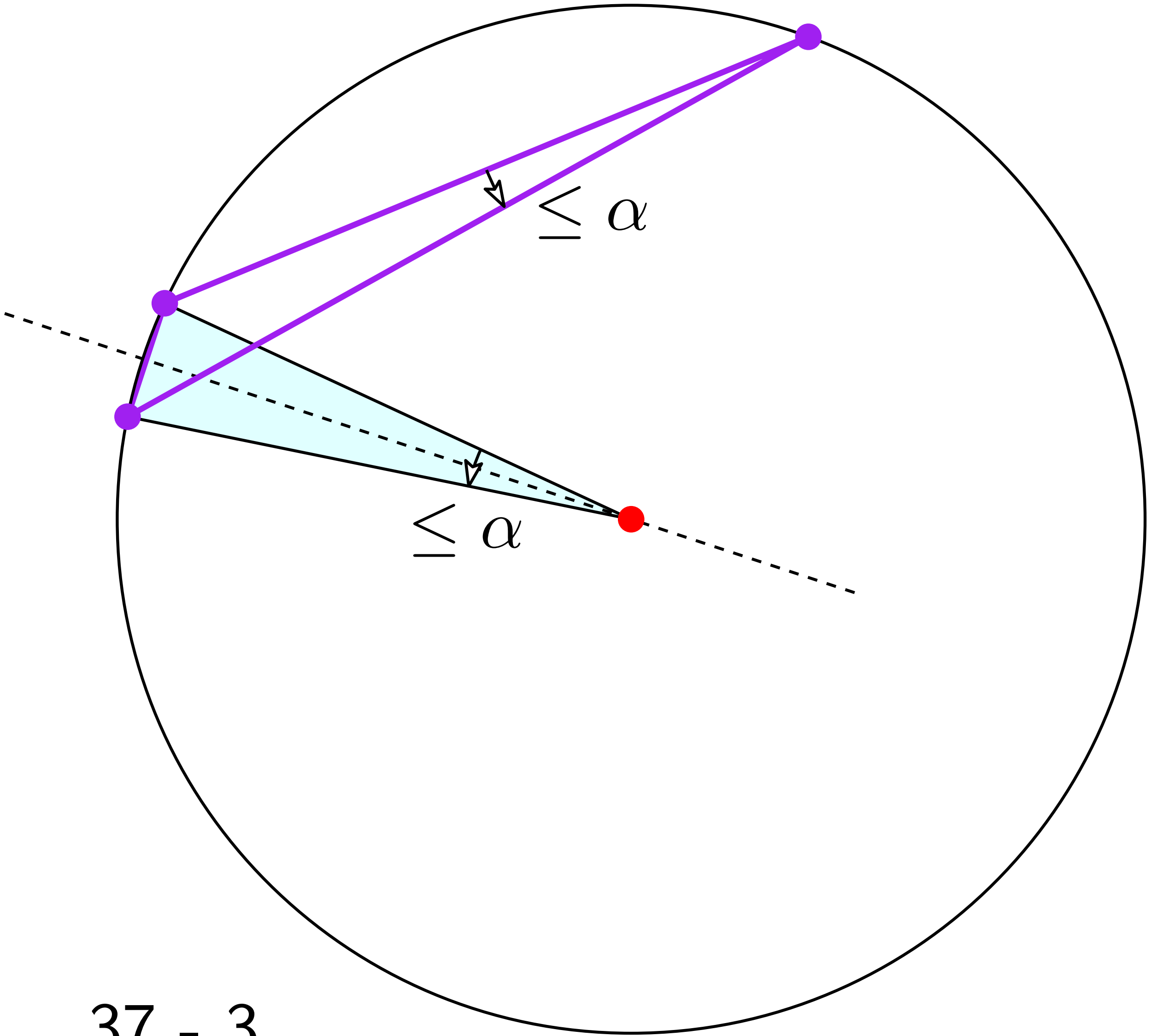



Very skinny triangle

Insert circumcenter

# Meshing

Delaunay mesh refinement off-centers



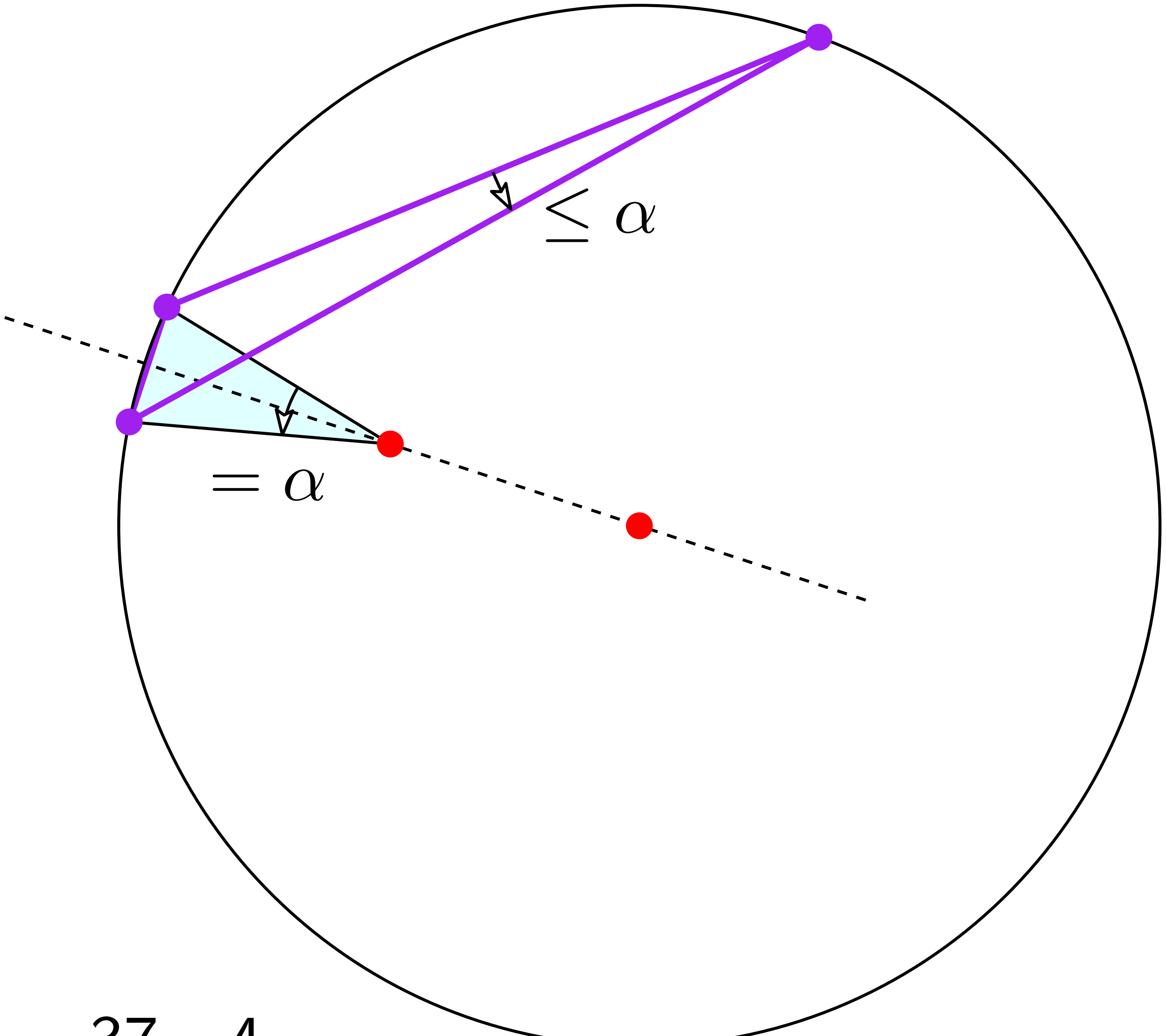

Very skinny triangle

Insert circumcenter

Still skinny triangle

# Meshing

Delaunay mesh refinement off-centers




Very skinny triangle

Insert circumcenter

Still skinny triangle

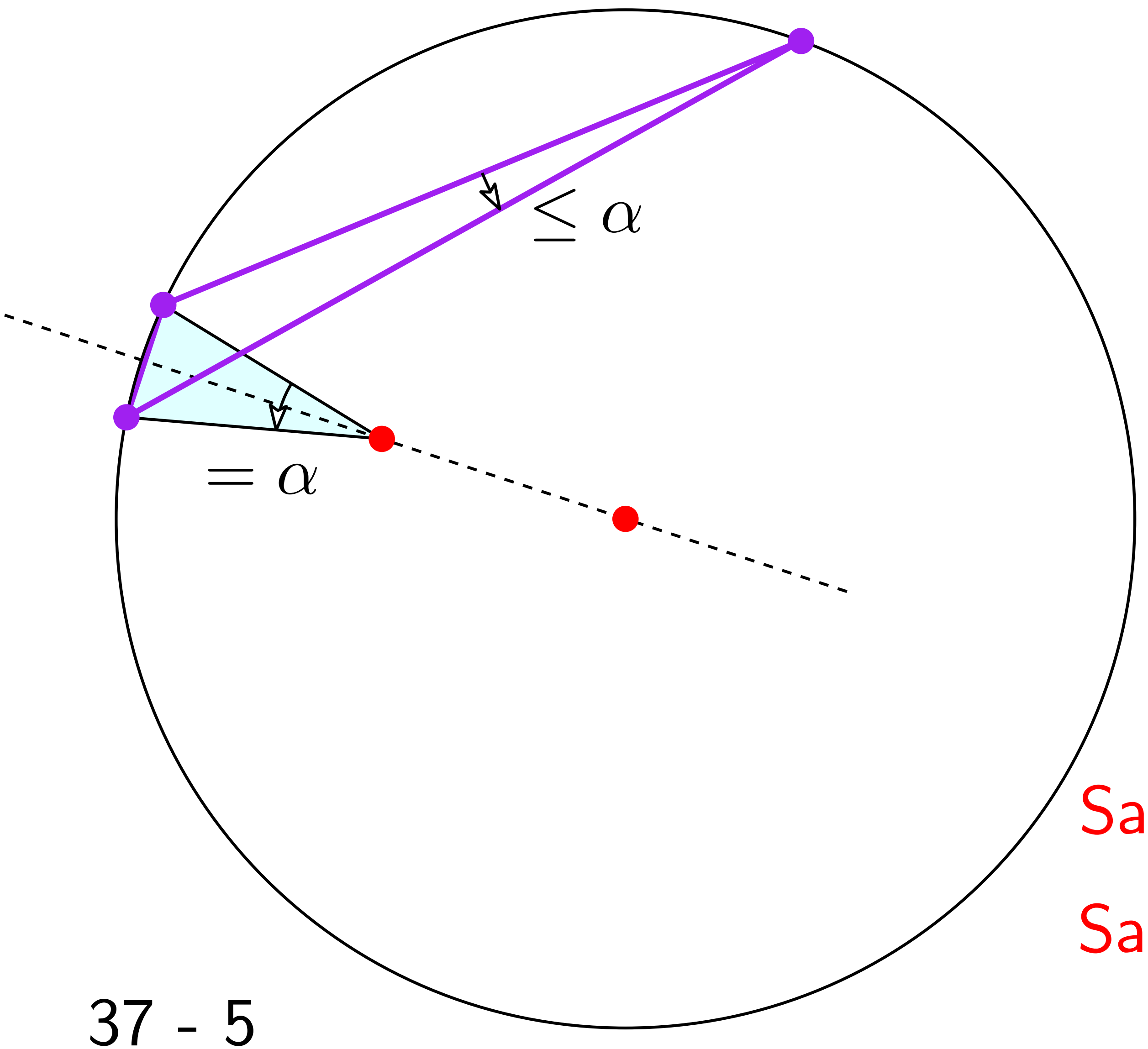
Off center is point

that creates

a non skinny triangle

# Meshing

Delaunay mesh refinement off-centers




Very skinny triangle

Insert circumcenter

Still skinny triangle

Off center is point

that creates

a non skinny triangle

Same theoretical guarantees

Save 30% in practice

# Meshing



LLoyd iteration

# Meshing



LLoyd iteration

# Meshing



## LLoyd iteration

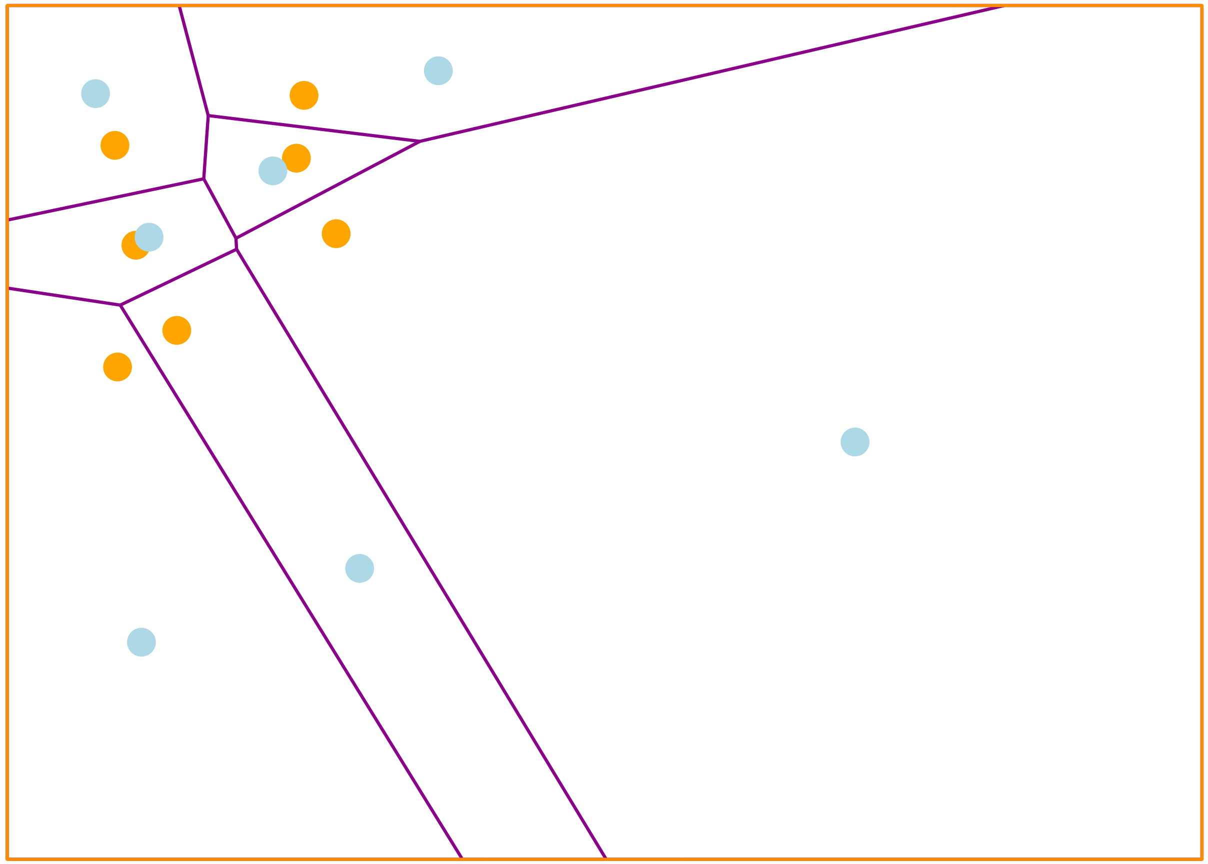
# Meshing

Delaunay mesh optimization

LLoyd iteration

Move to barycenter

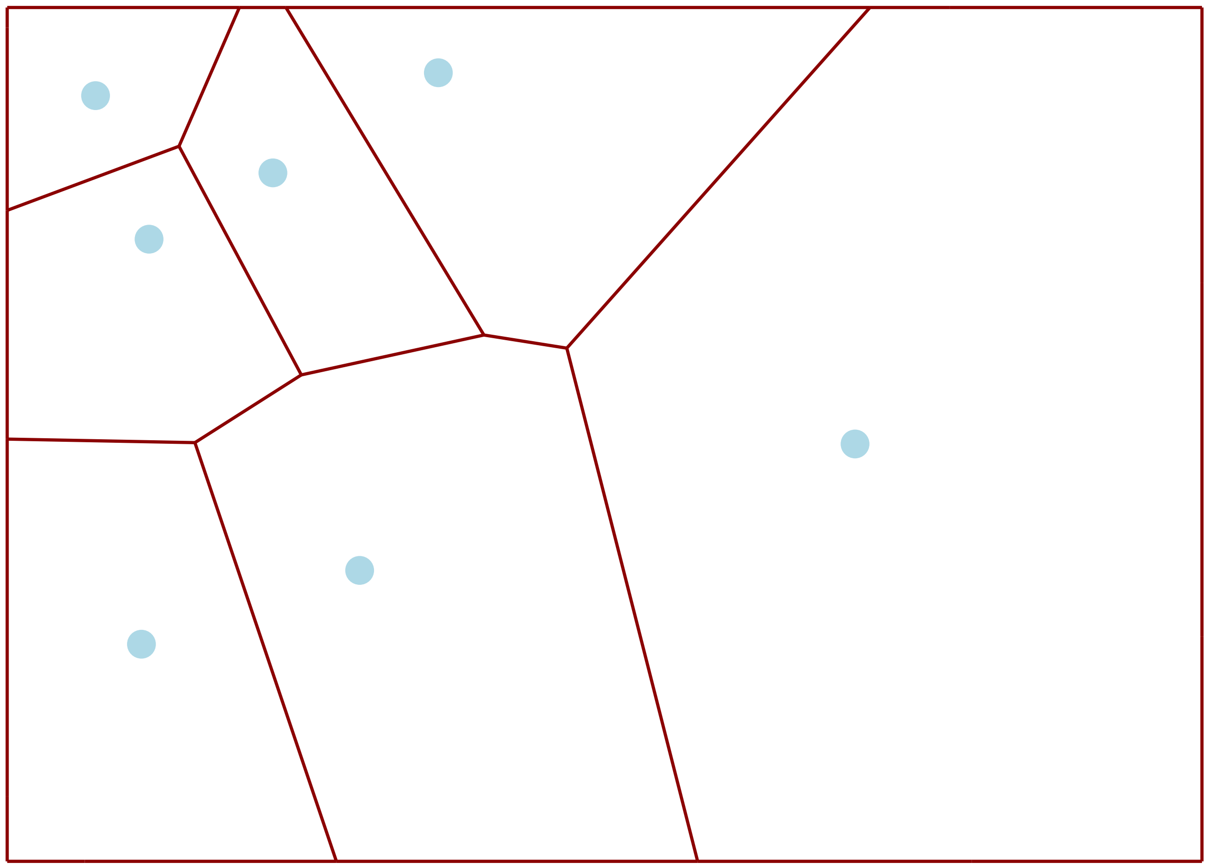
Clip by some boundary



# Meshing



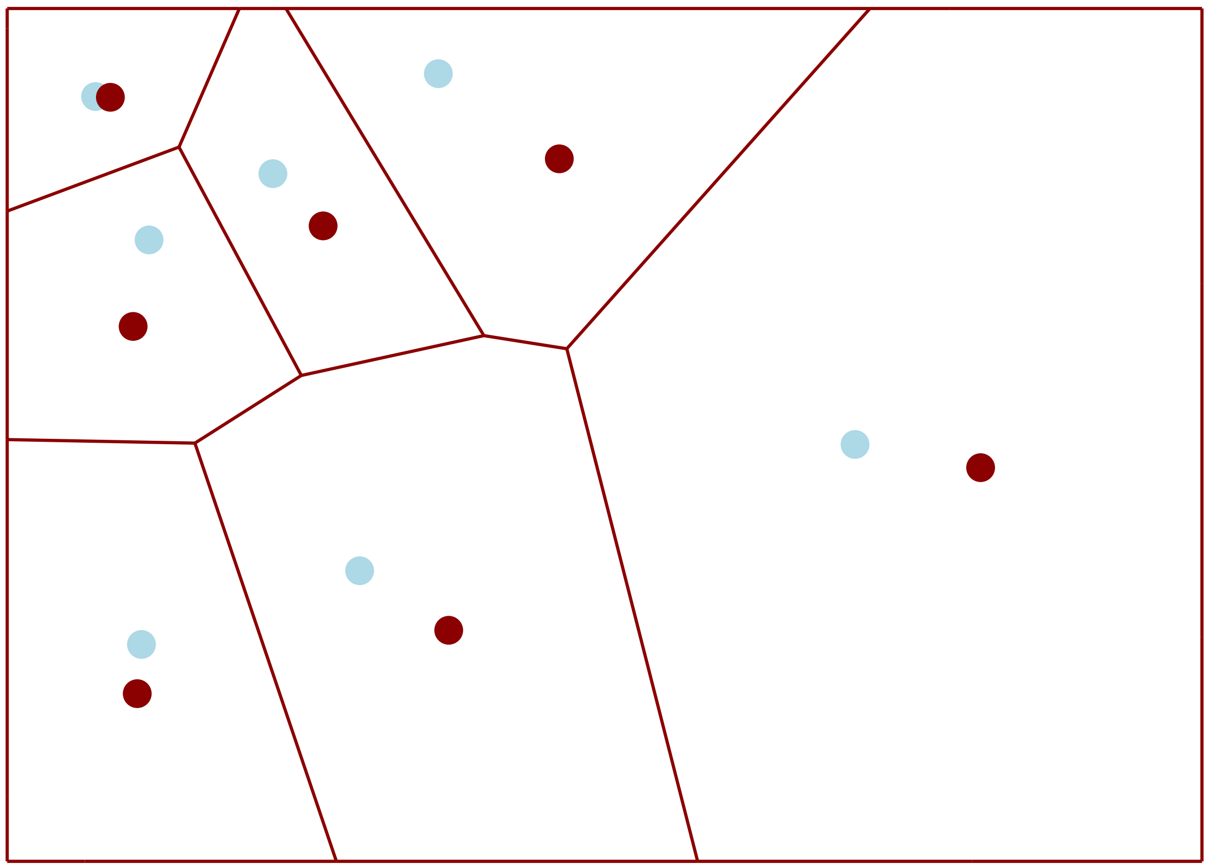
## LLoyd iteration



# Meshing



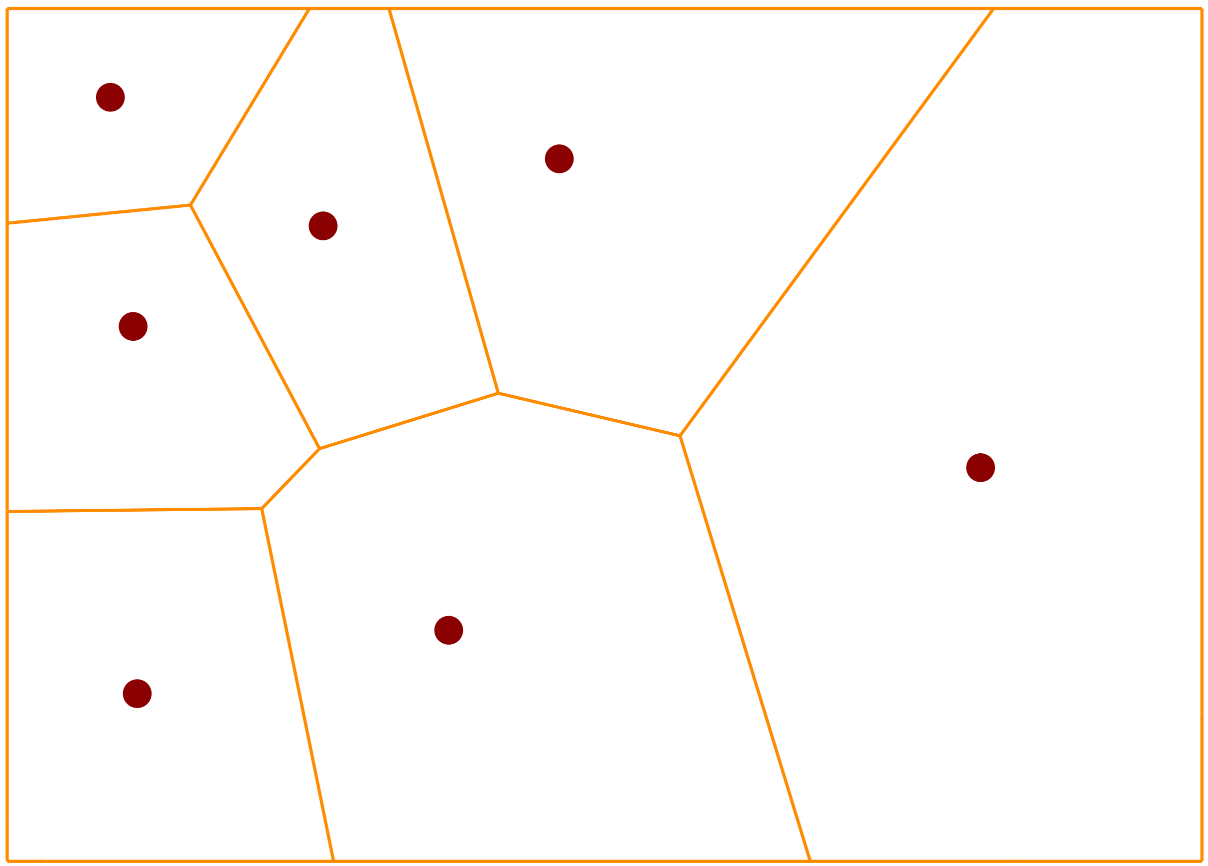
## LLoyd iteration



# Meshing



## LLoyd iteration

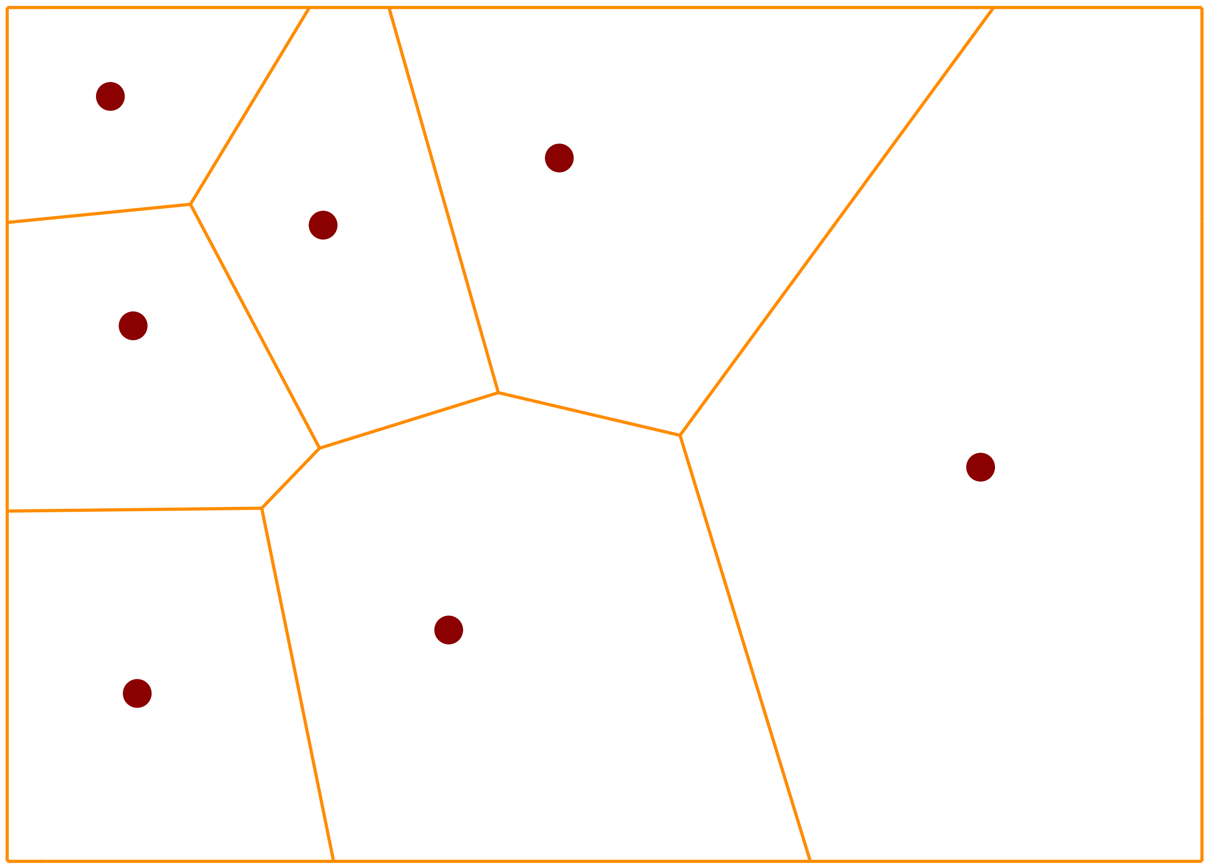


# Meshing

Delaunay mesh optimization

LLoyd iteration

Reach a nice point distribution



# Meshing

Delaunay mesh optimization

Alternate

Delaunay mesh refinement

Lloyd smooting    or different kind of smoothing

# Meshing

## Delaunay mesh optimization

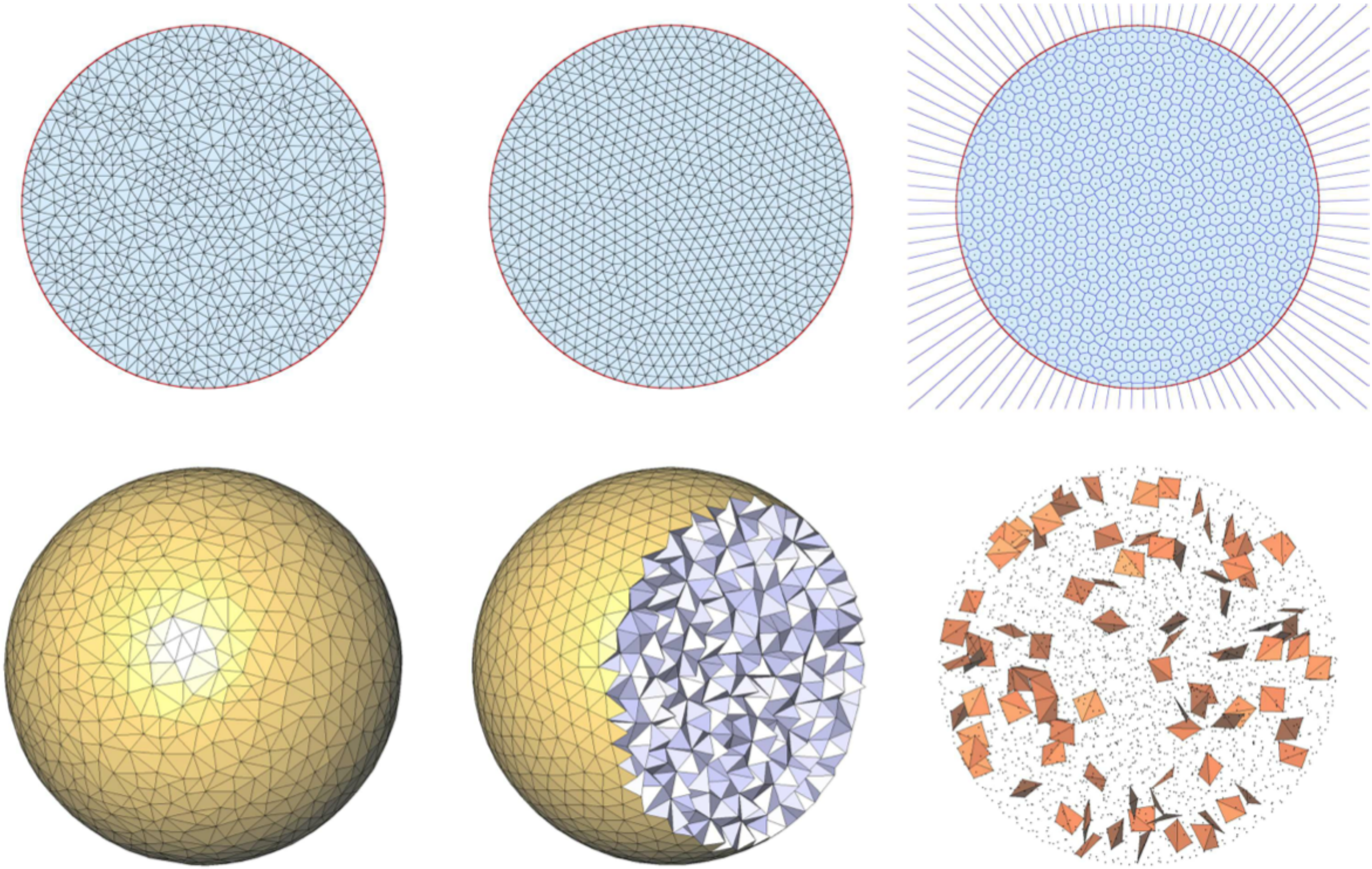
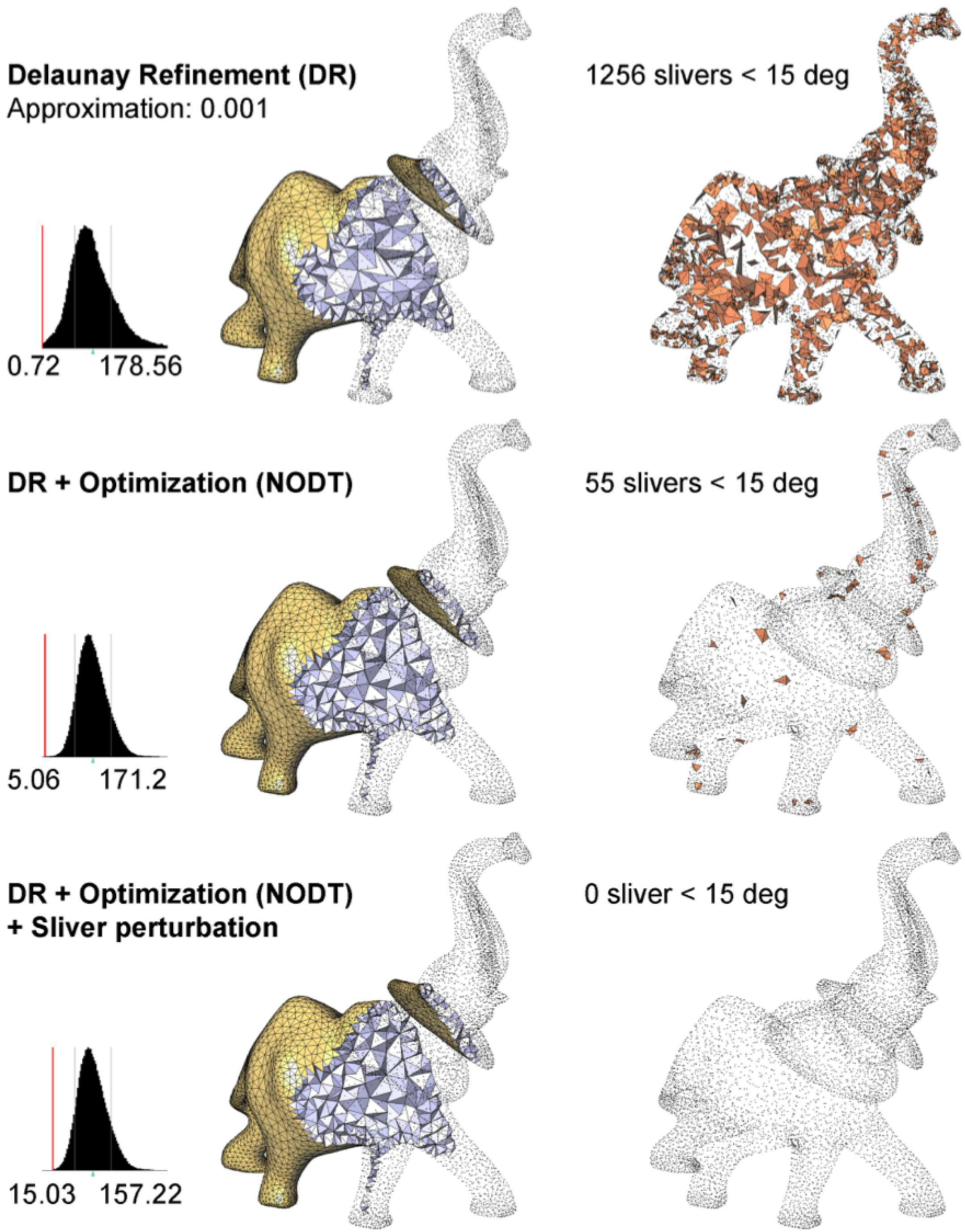


Figure 1.6: CVT mesh optimization. In 2D (top), (left) a 2D Delaunay mesh $M_2$ generated by Delaunay refinement, (center) $M_2$ optimized with CVT, and (right) $M_2$'s Voronoi diagram. In 3D (bottom), (left) a 3D Delaunay mesh $M_3$ generated by Delaunay refinement, (center) $M_3$ optimized with CVT, and (right) $M_3$'s slivers (tetrahedra with dihedral angles smaller than 5°).

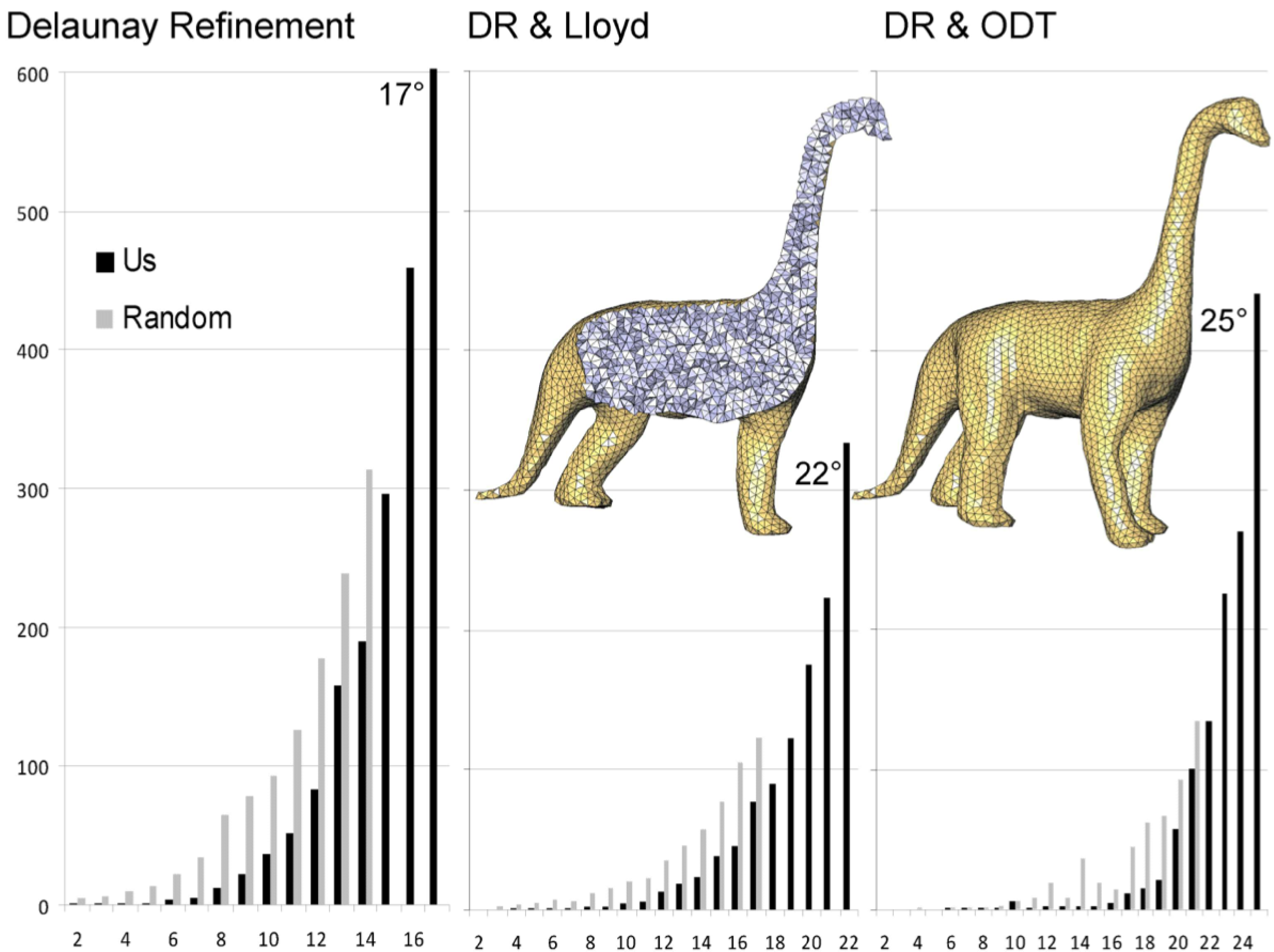
# Meshing

## Delaunay mesh optimization



# Meshing

## Delaunay mesh optimization

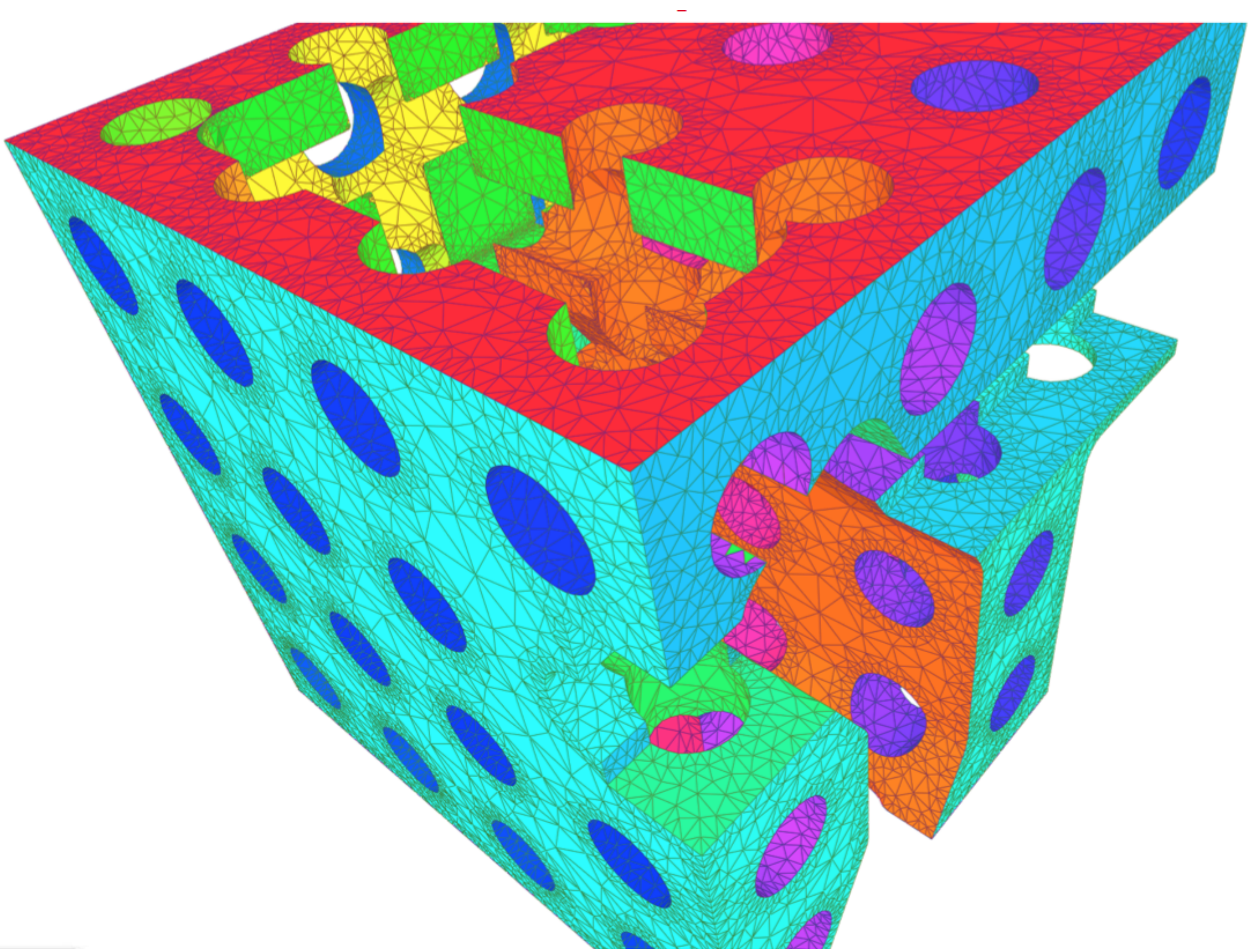


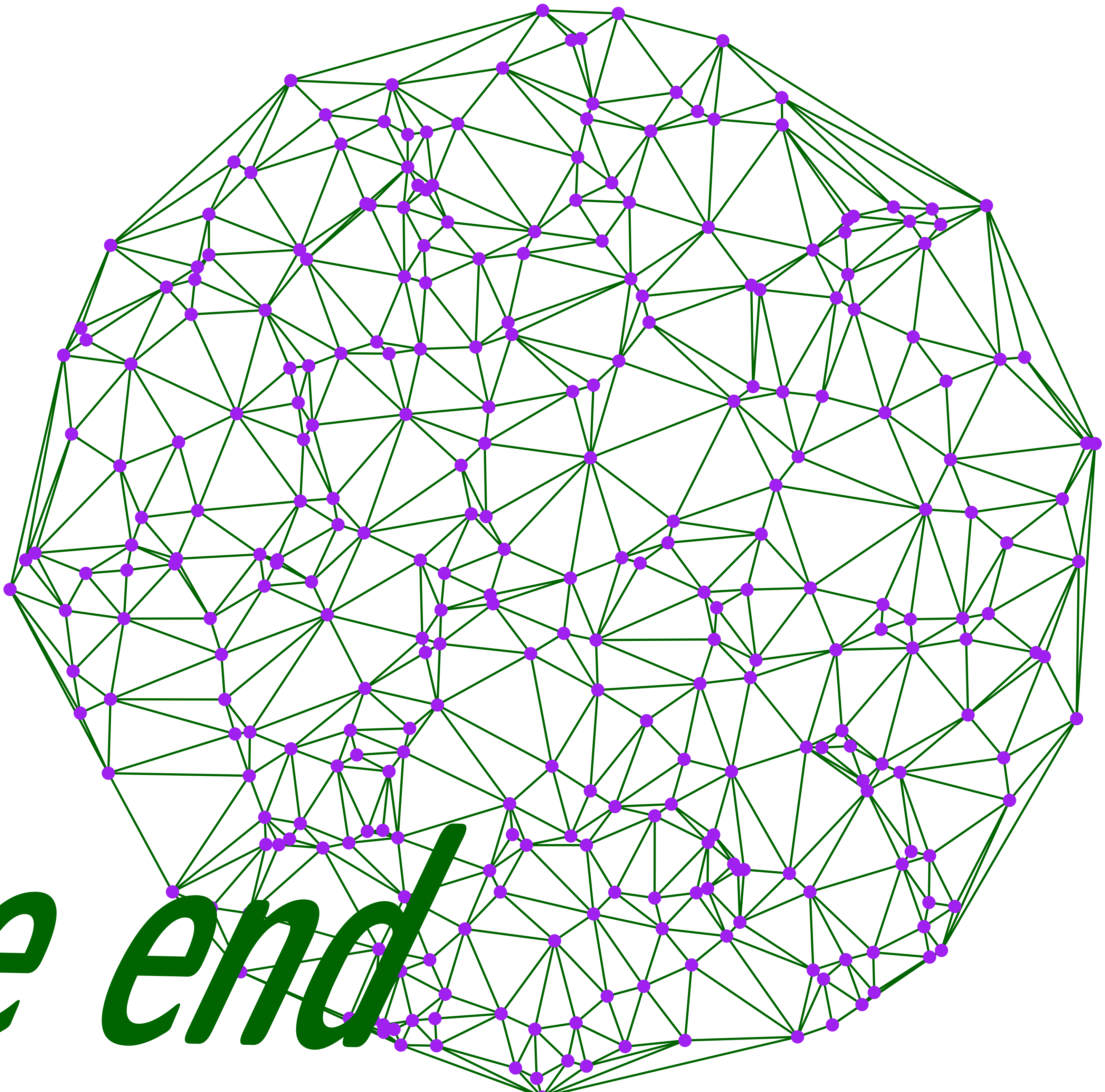
# Meshing

3D

Constraints: edges and faces

Point to insert may be encroached by edges or faces





# *The end*