Parallel Meshing by Enumerating the Vertices of the Voronoi cells

Bruno Lévy
ALICE Géométrie & Lumière
CENTRE INRIA Nancy Grand-Est

OVERVIEW

Part 1. Introduction - Motivations

Part 2. Blowing Bubbles: CVT

Part 3. Anisotropy

Part 4. Journey in the 6th dimension

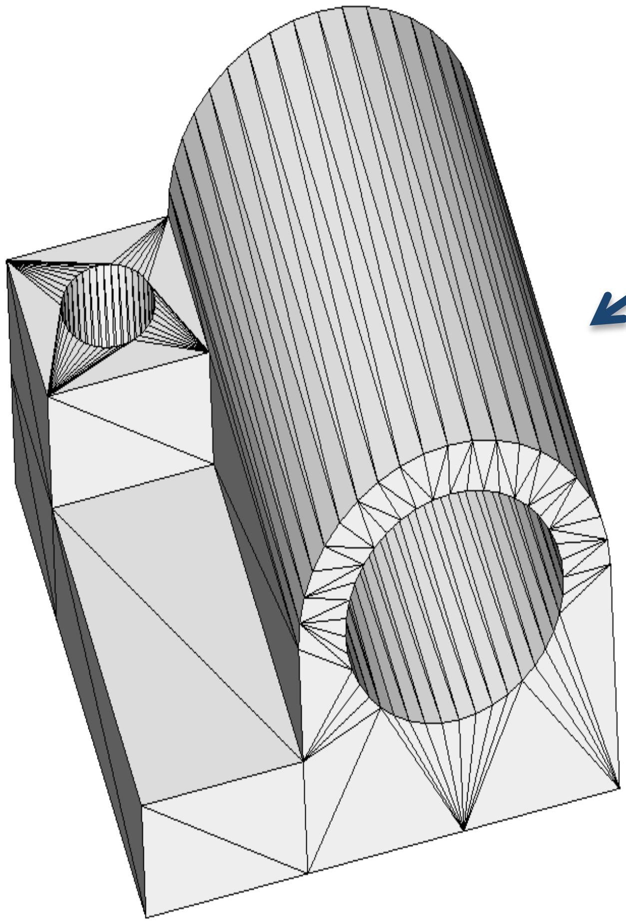
Part 5. The algorithm

Part 6. Results and conclusions

1

Introduction - Motivations

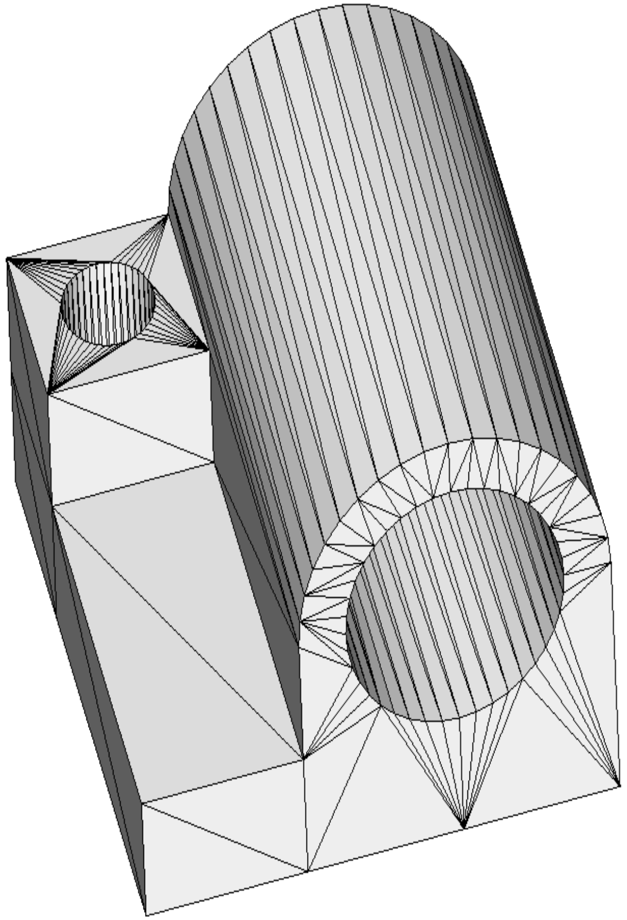
Part 1. Introduction: meshing (and re-meshing)



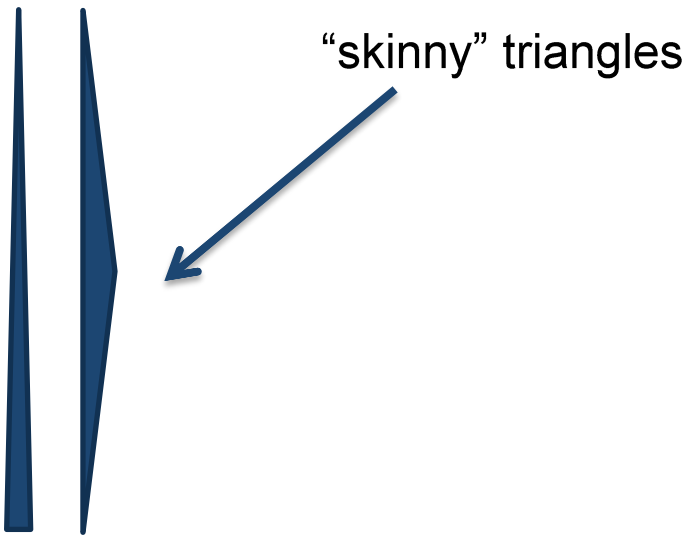
Input geometry, with “bad” triangles

(Re) meshing [Du et.al], [Alliez et.al], [Yan, **L** et.al]

Part 1. Introduction: meshing (and re-meshing)

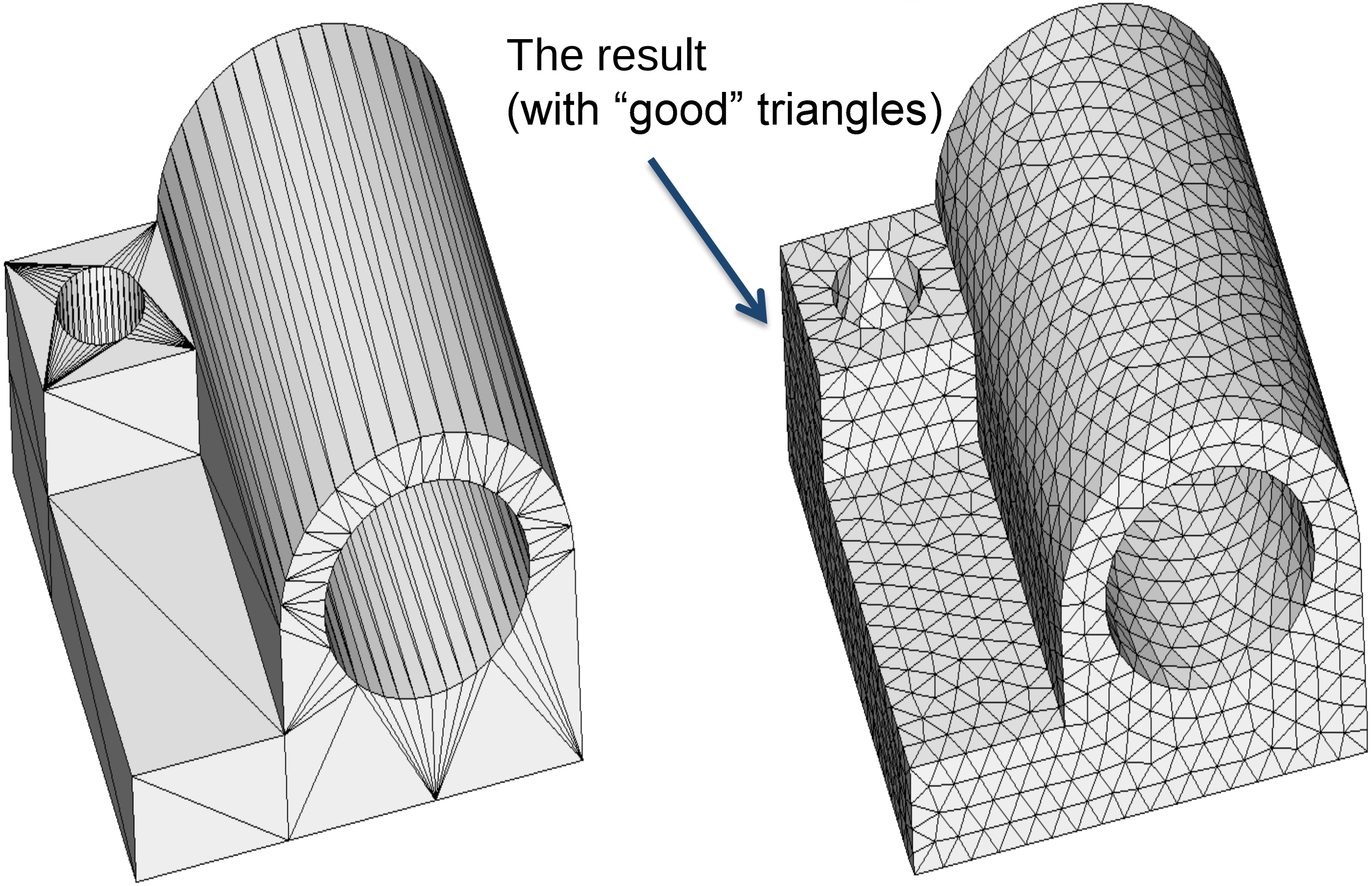


Why “bad” ?
Because extreme angles (near 0º or 180º) can cause numerical instabilities.



(Re)-meshing [Du et.al], [Alliez et.al], [Yan, **L** et.al]

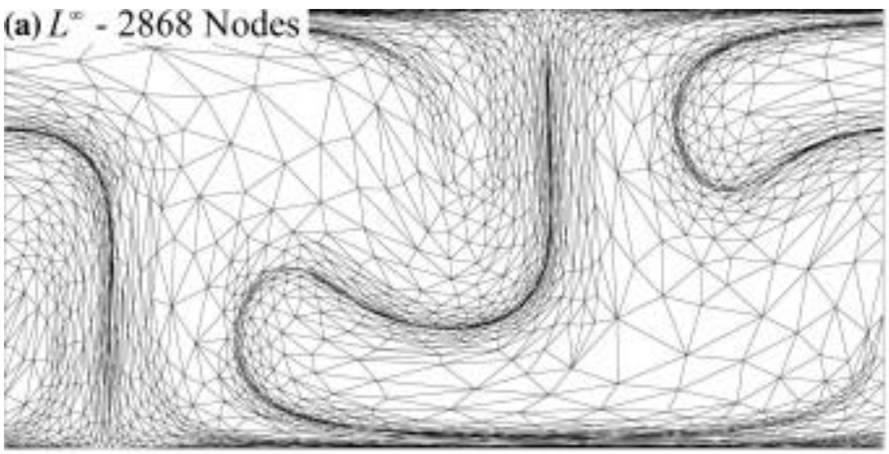
Part 1. Introduction: meshing (and re-meshing)



(Re)-meshing [Du et.al], [Alliez et.al], [Yan, **L** et.al]

Part 1. Introduction: meshing (and re-meshing)

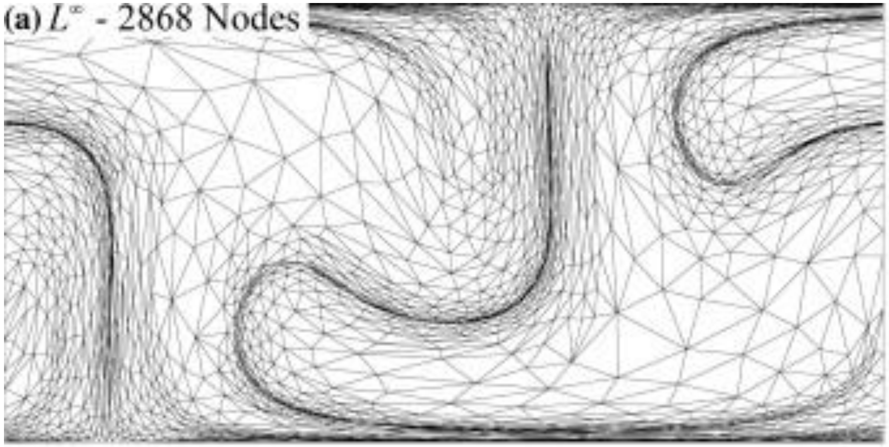
It has skinny triangles where we want them and oriented as we like (following the variation of the physics)



Solution-adapted mesh [Miron et.al, Journal of Computational Physics, 2010]

Part 1. Introduction: meshing (and re-meshing)

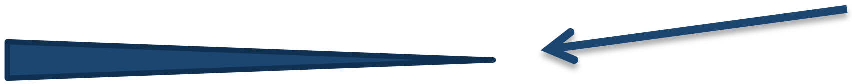
It has skinny triangles where we want them and oriented as we like (following the variation of the physics)



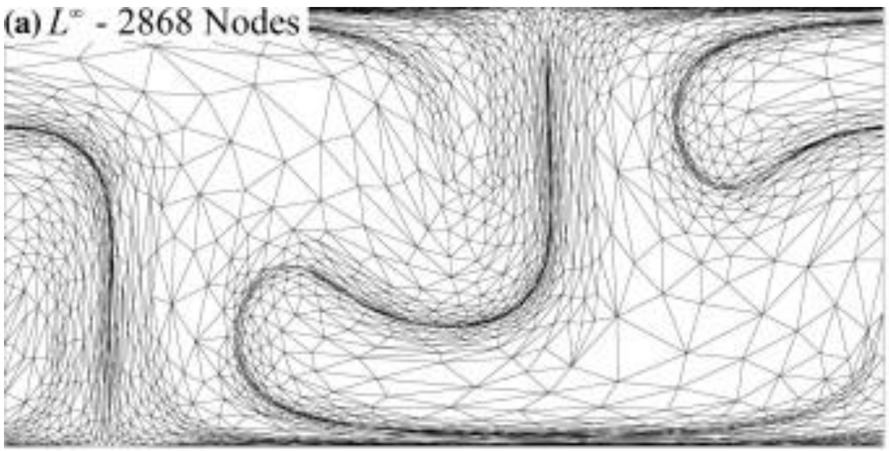
Benefit: **higher** accuracy with **smaller** number of elements.

Solution-adapted mesh [Miron et.al, Journal of Computational Physics, 2010]

Part 1. Introduction: meshing (and re-meshing)



It has skinny triangles where we want them and oriented as we like (following the variation of the physics)

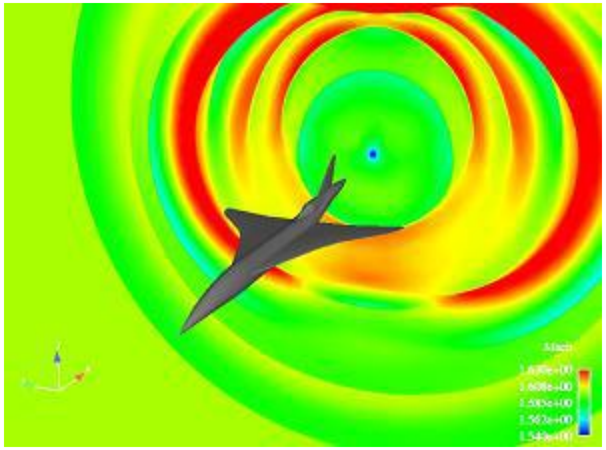


Benefit: **higher** accuracy with **smaller** number of elements.

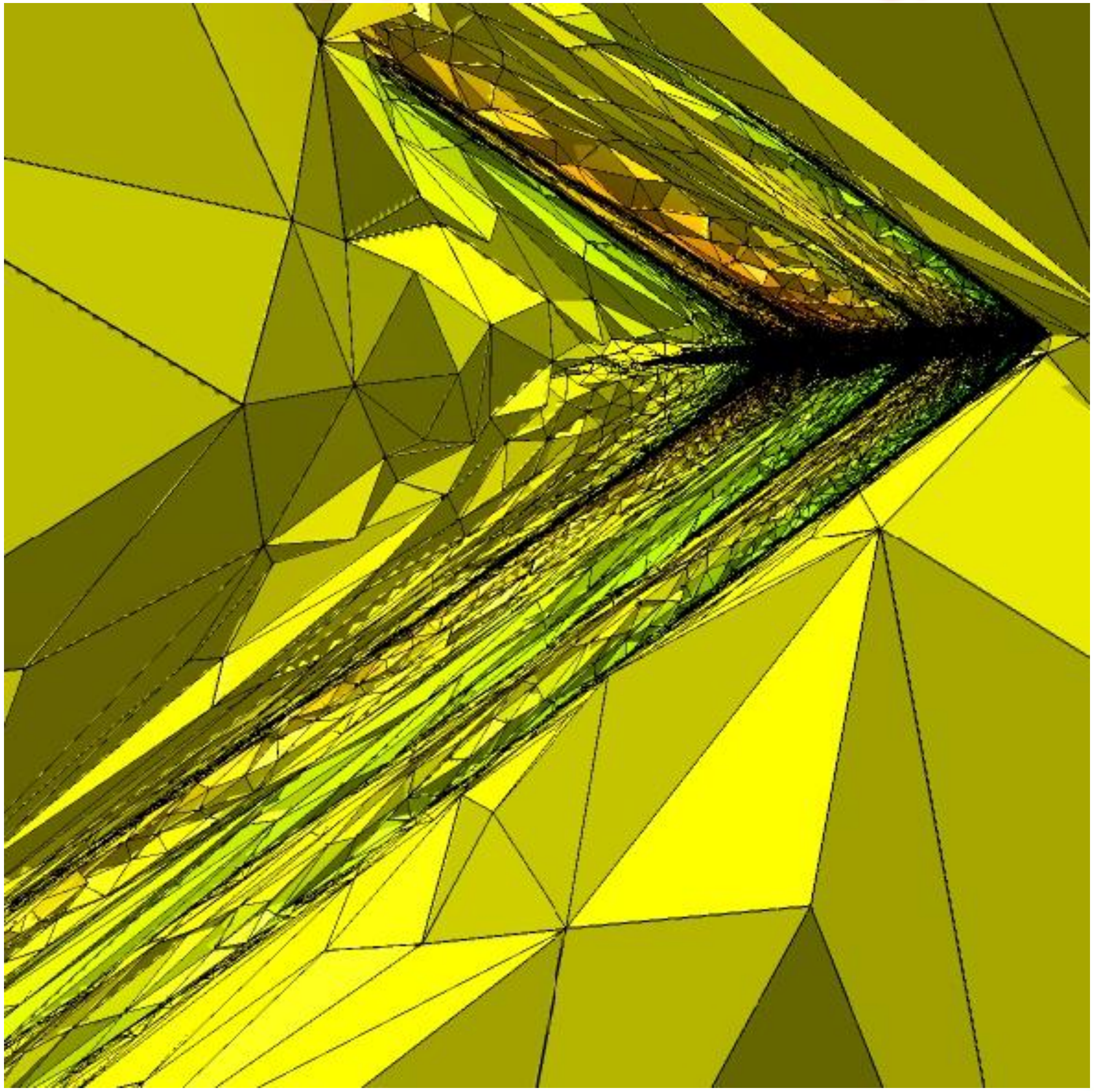
Skinny triangles: not always “bad”, even sometimes **desired** but with **controlled** shape, size and orientation.

Solution-adapted mesh [Miron et.al, Journal of Computational Physics, 2010]

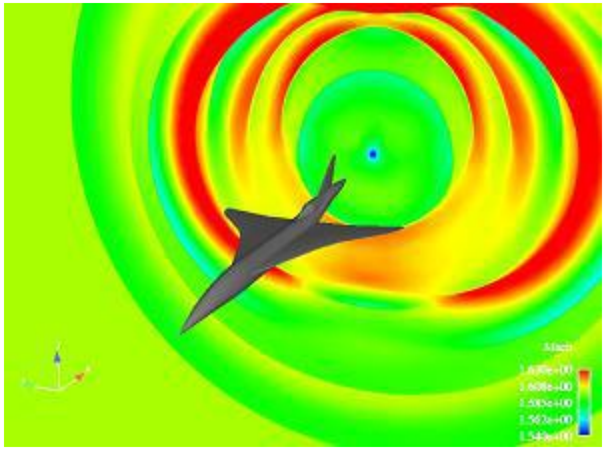
Part 1. Introduction: meshing (and re-meshing)



Supersonic flight
[Alauzet et.al]

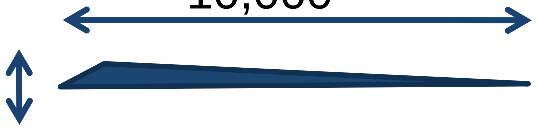


Part 1. Introduction: meshing (and re-meshing)



Supersonic flight
[Alauzet et.al]

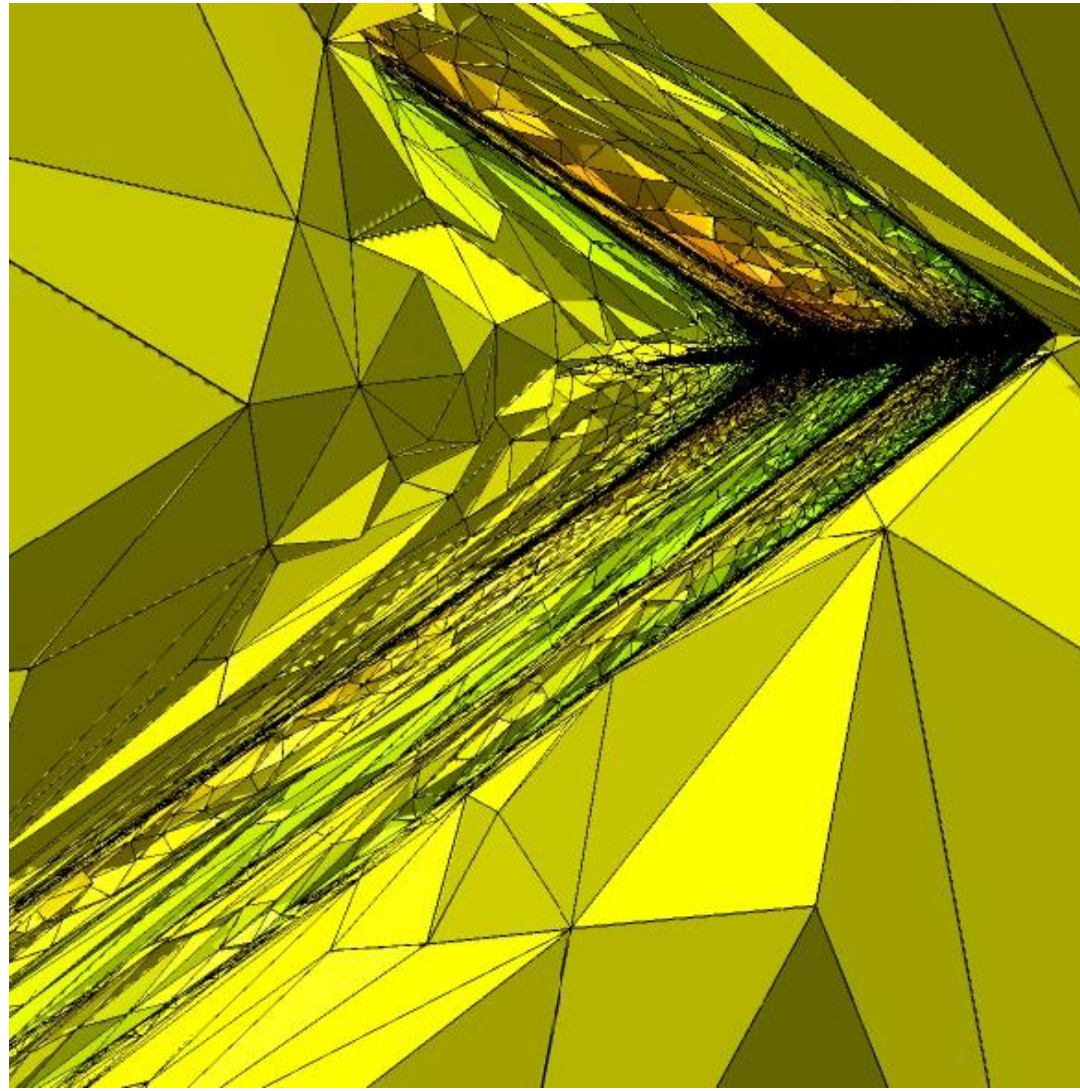
10,000



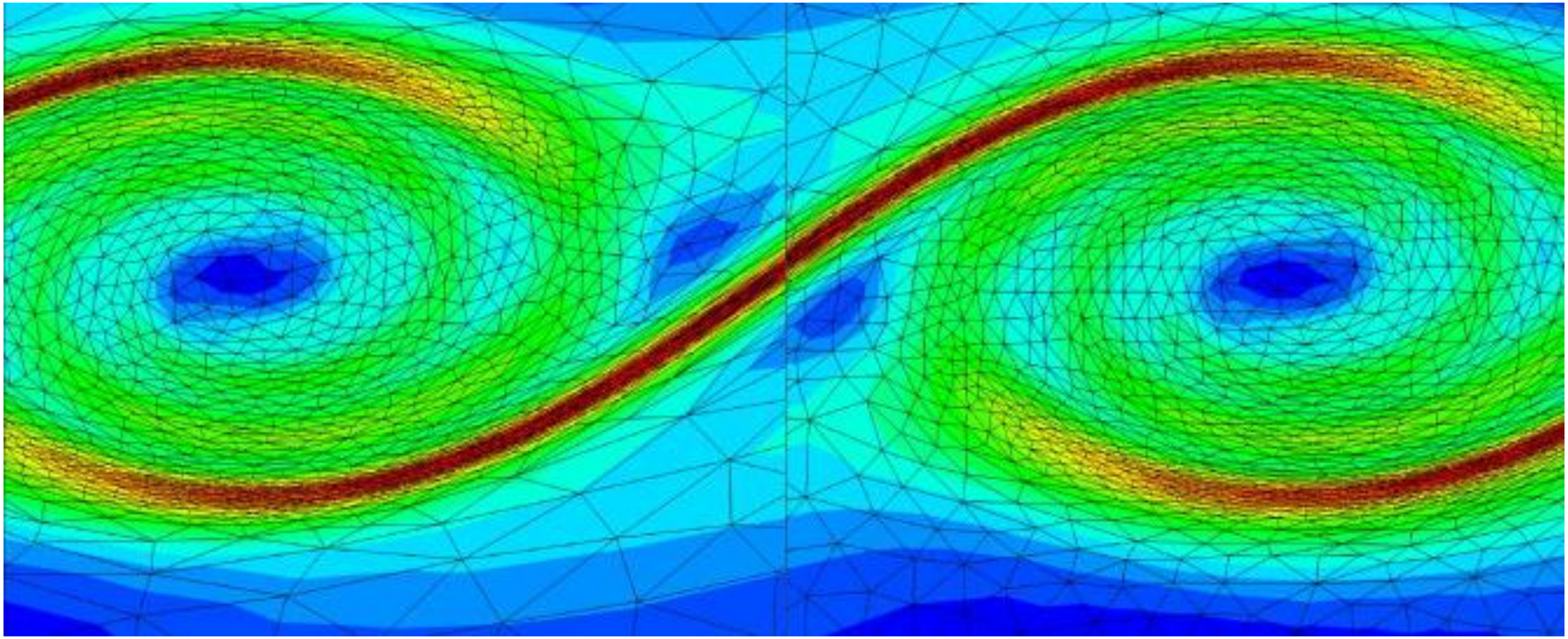
1

Aspect ratio:

1:10,000 (typically)

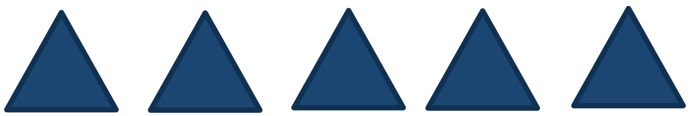


Part 1. Introduction: meshing (and re-meshing)

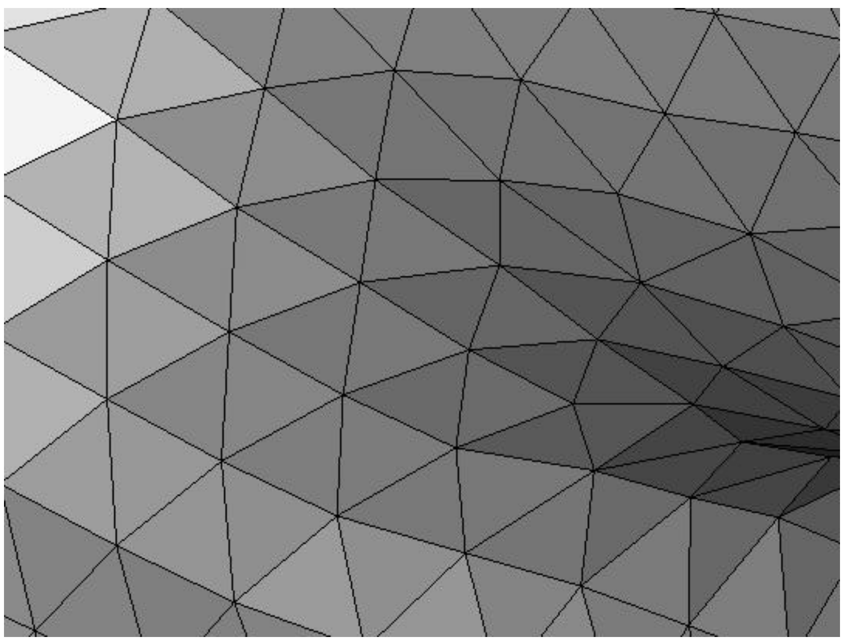


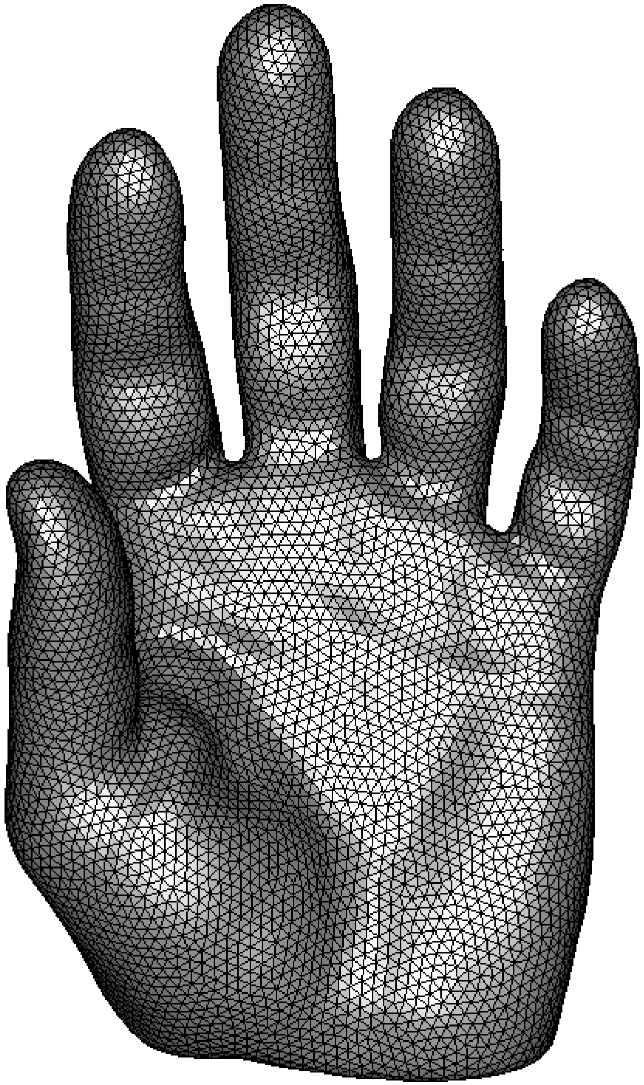
Solution-adapted **anisotropic** mesh
[Miron et.al, Journal of Computational Physics, 2010]

Part 1. Introduction : mesh classes

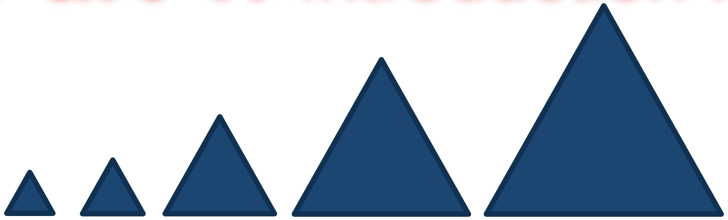


Isotropic mesh: All the triangles have
* the same shape (equilateral)
* the same size

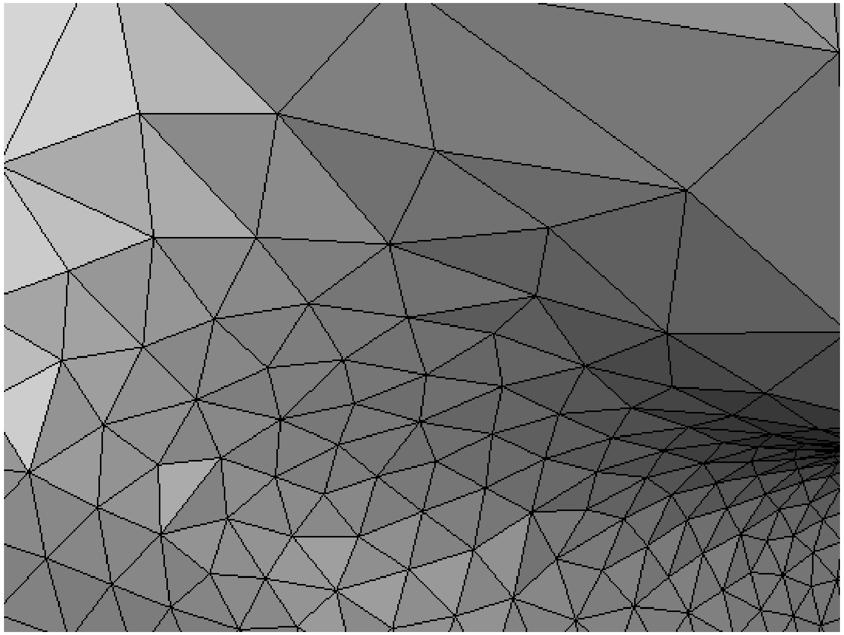


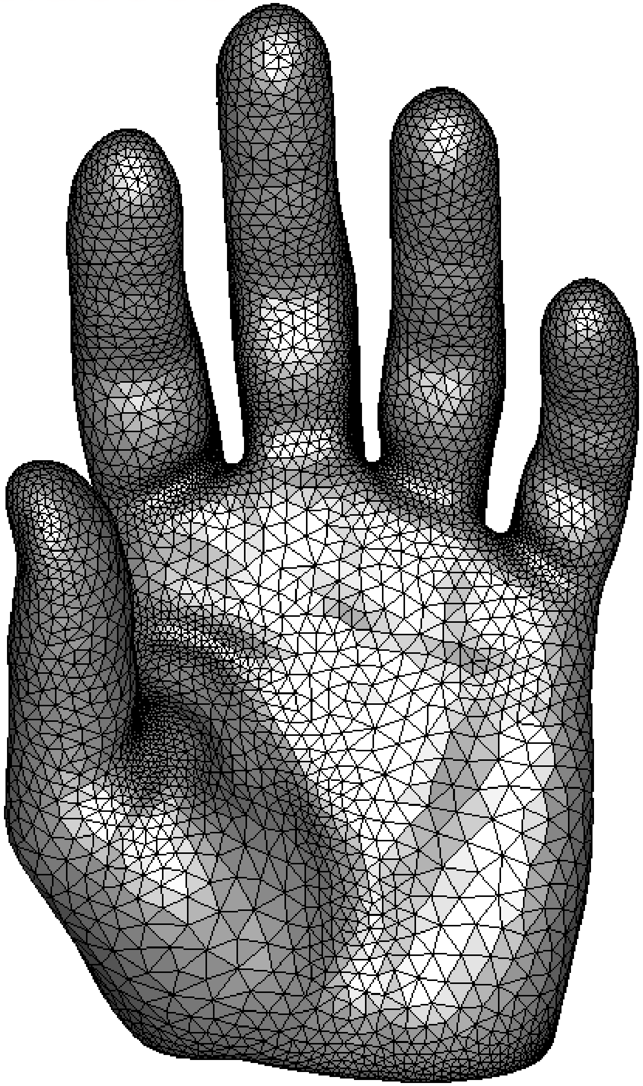


Part 1. Introduction : mesh classes

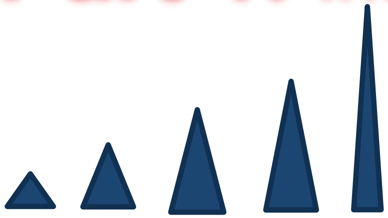


Isotropic graded mesh:
  * the same shape (equilateral)
  * size can vary

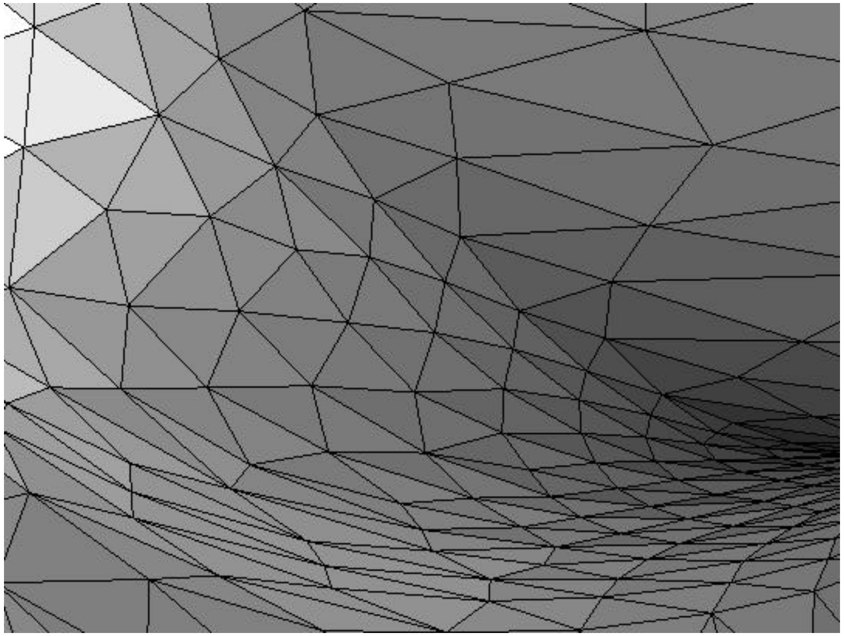


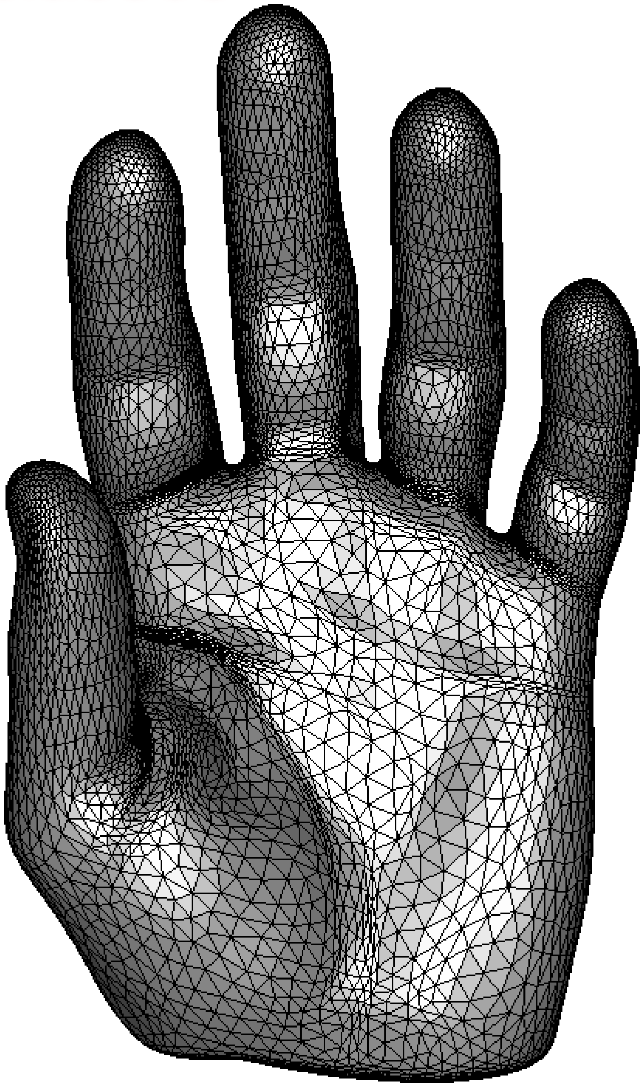


Part 1. Introduction : mesh classes

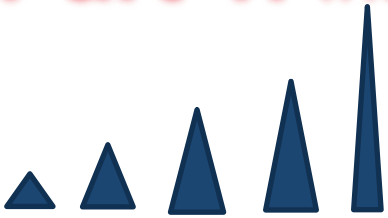


Anisotropic mesh:
* shape can vary
* size can vary



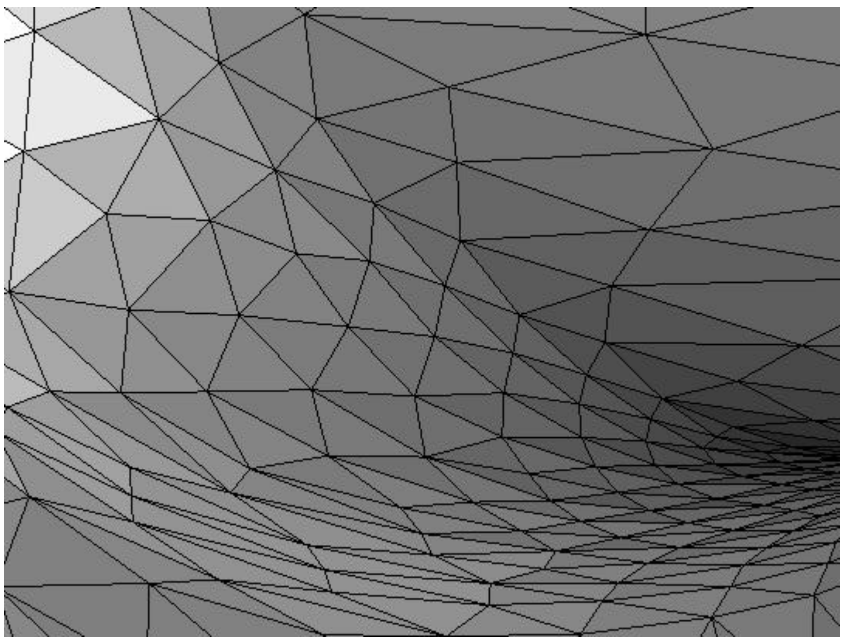


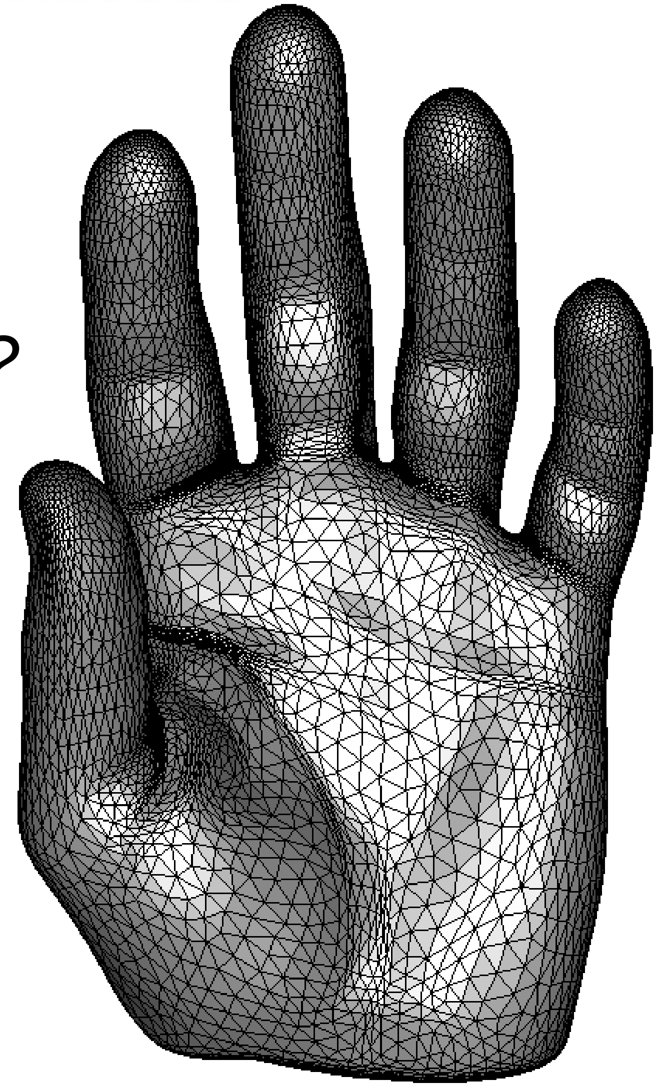
Part 1. Introduction : mesh classes



Anisotropic mesh:
* shape can vary
* size can vary

Q: How to generate an anisotropic surface mesh ?





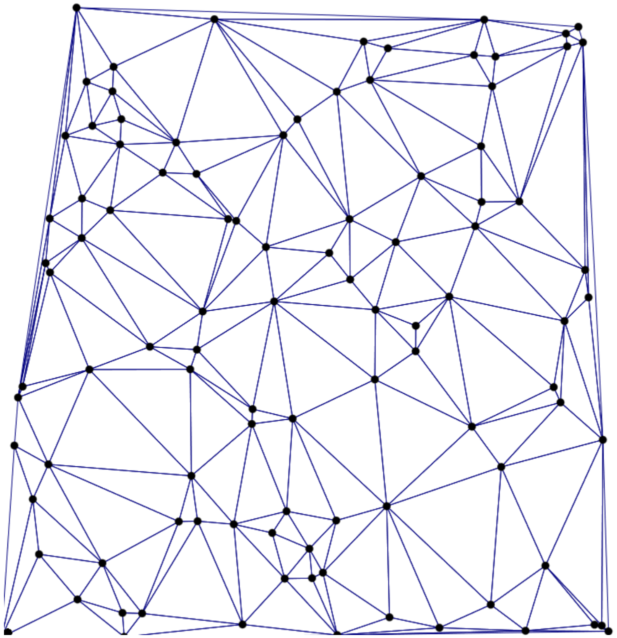
2

Blowing Bubbles

Centroidal Voronoi Tesselations

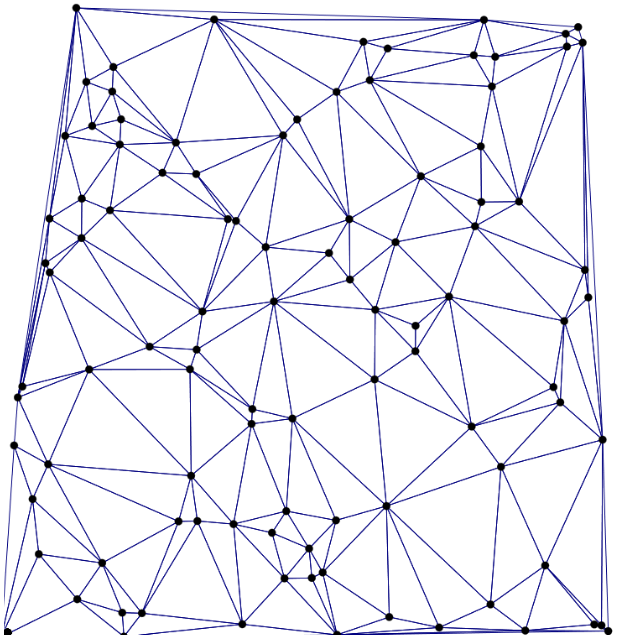
Part. 2. Centroidal Voronoi Tesselation

Optimize a Voronoi diagram from the point of view of sampling regularity (quantization noise power)



Part. 2. Centroidal Voronoi Tesselation

Optimize a Voronoi diagram from the point of view of sampling regularity
(quantization noise power)

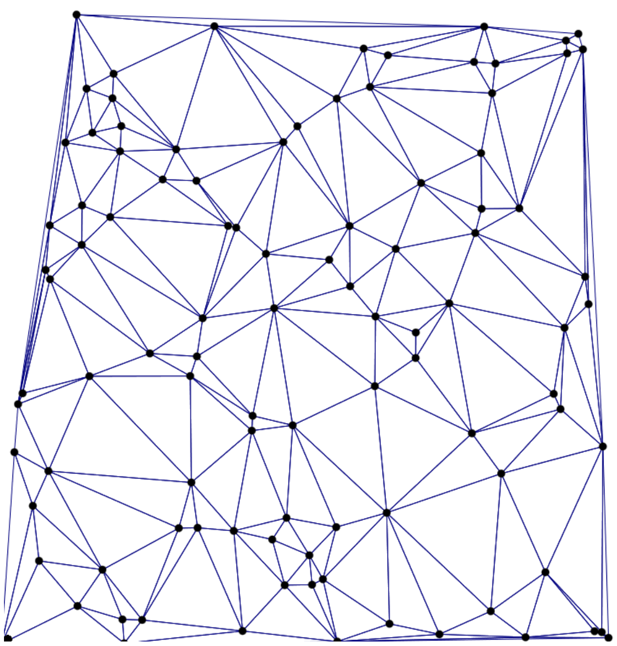


Minimize

$$F = \sum_i \int_{Vor(i)} \left\| \mathbf{x}_i - \mathbf{x} \right\|^2 d\mathbf{x}$$

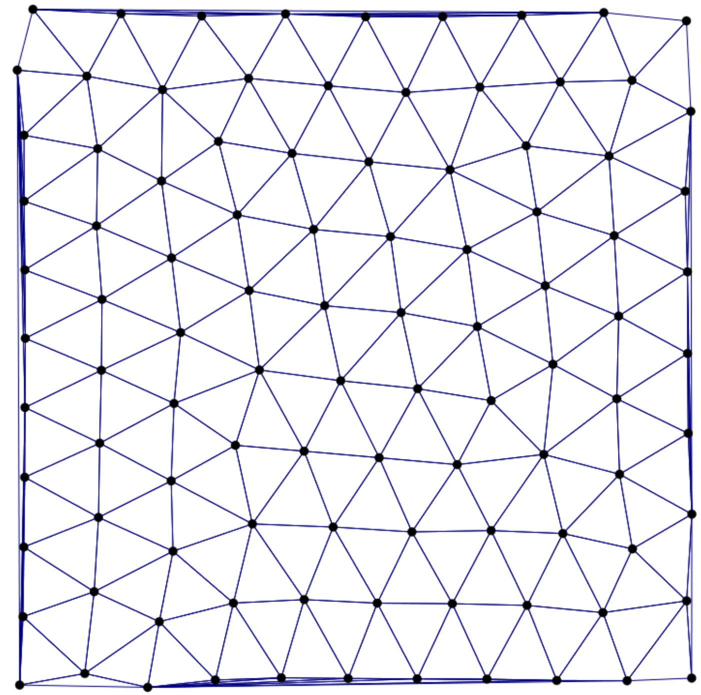
Part. 2. Centroidal Voronoi Tesselation

Optimize a Voronoi diagram from the point of view of sampling regularity (quantization noise power)



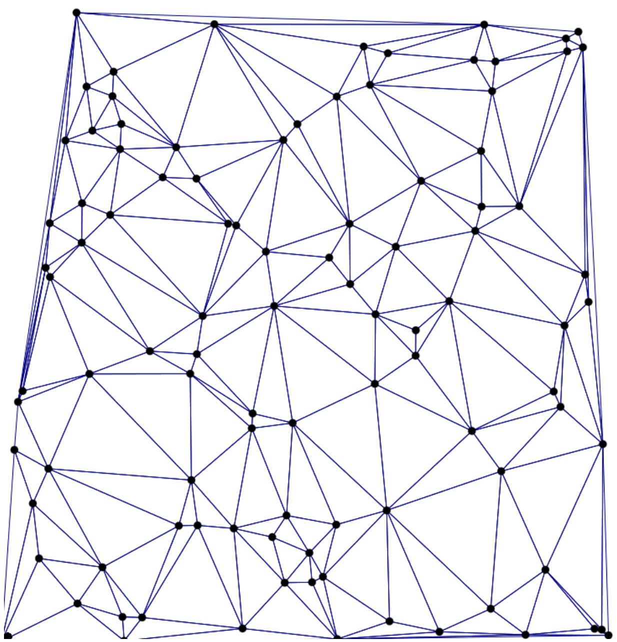
Minimize

$$F = \sum_i \int_{Vor(i)} \left\| \mathbf{x}_i - \mathbf{x} \right\|^2 d\mathbf{x}$$

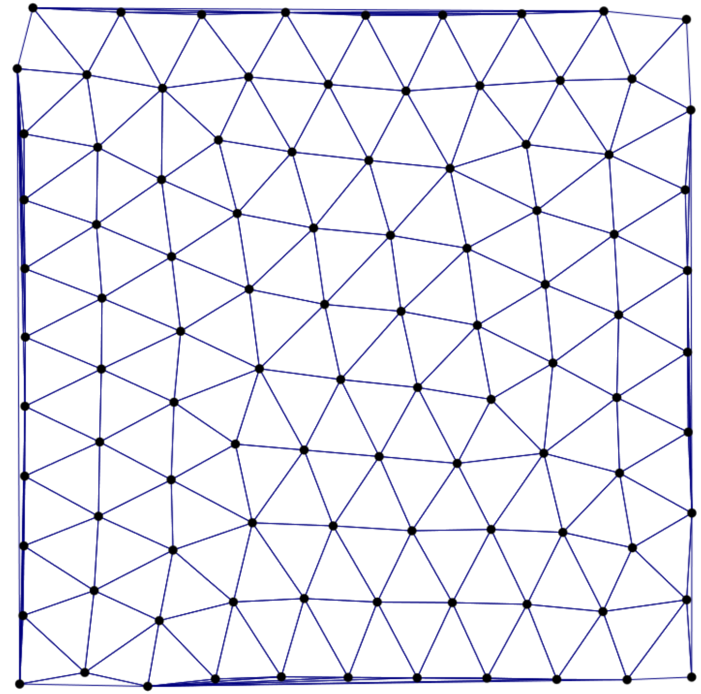


Part. 2. Centroidal Voronoi Tesselation

Optimize a Voronoi diagram from the point of view of sampling regularity (quantization noise power)



Minimize



$$F = \sum_i \int_{Vor(i)} \left\| \mathbf{x}_i - \mathbf{x} \right\|^2 d\mathbf{x}$$

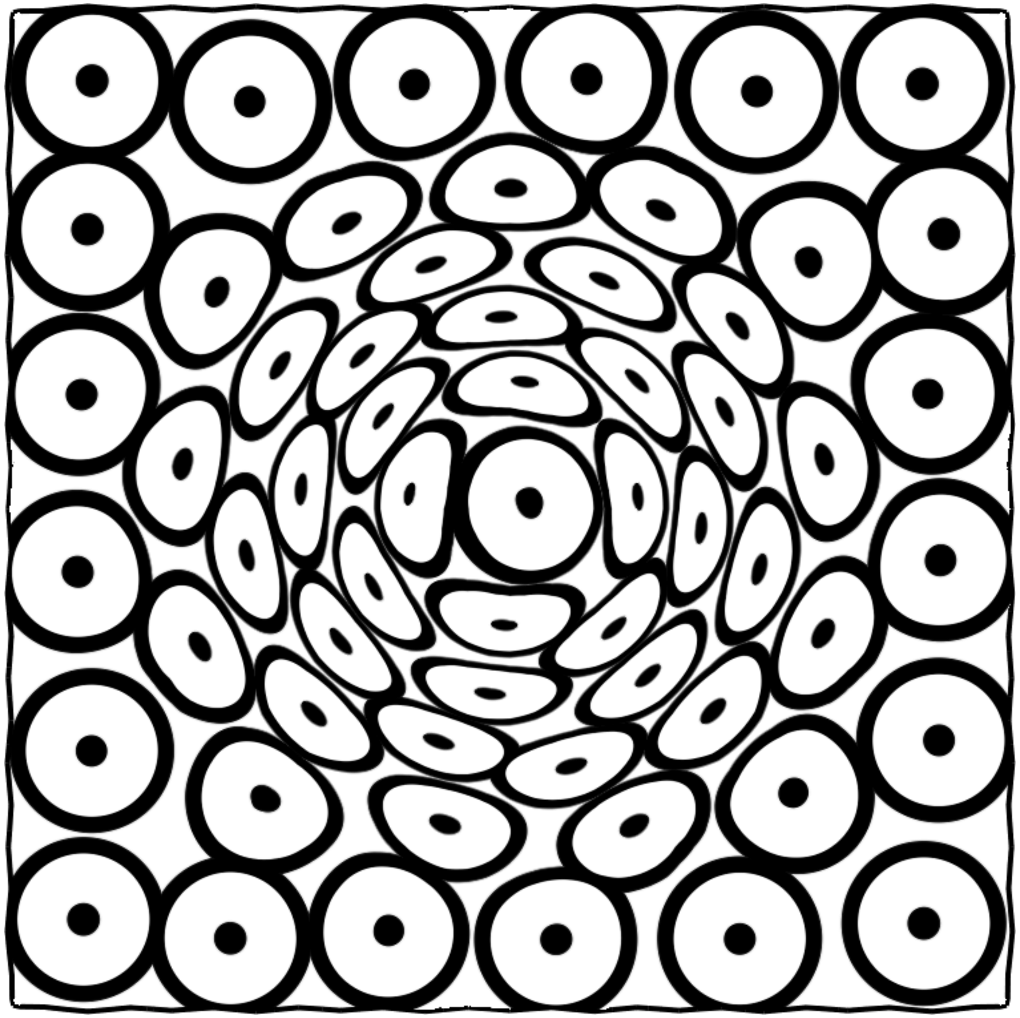
Theorem: F is of class C^2 [Liu, Wang, **L**, Yan, Lu, ACM TOG 2008]

Part. 2. Centroidal Voronoi Tesselation

3

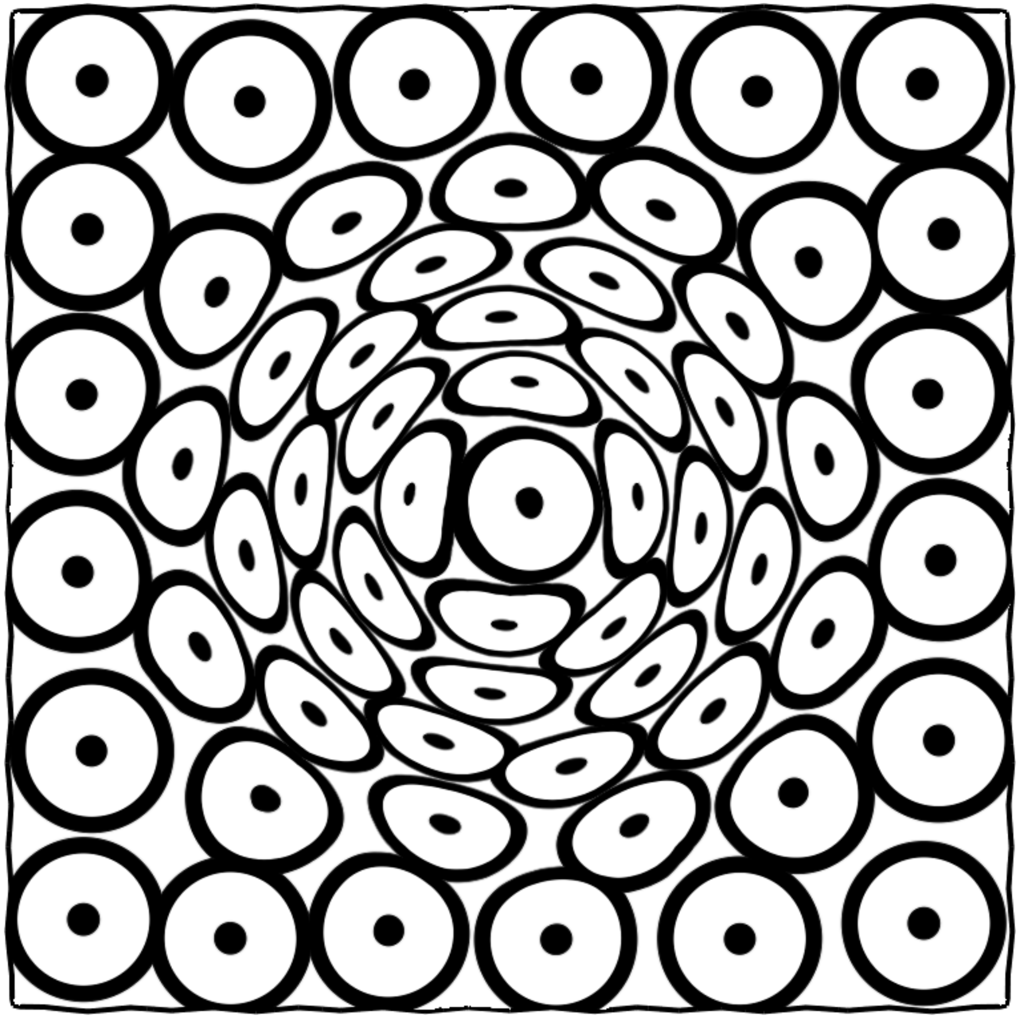
Anisotropy

Part. 3. Anisotropy



The input: anisotropy field
Specifies shape and orientation

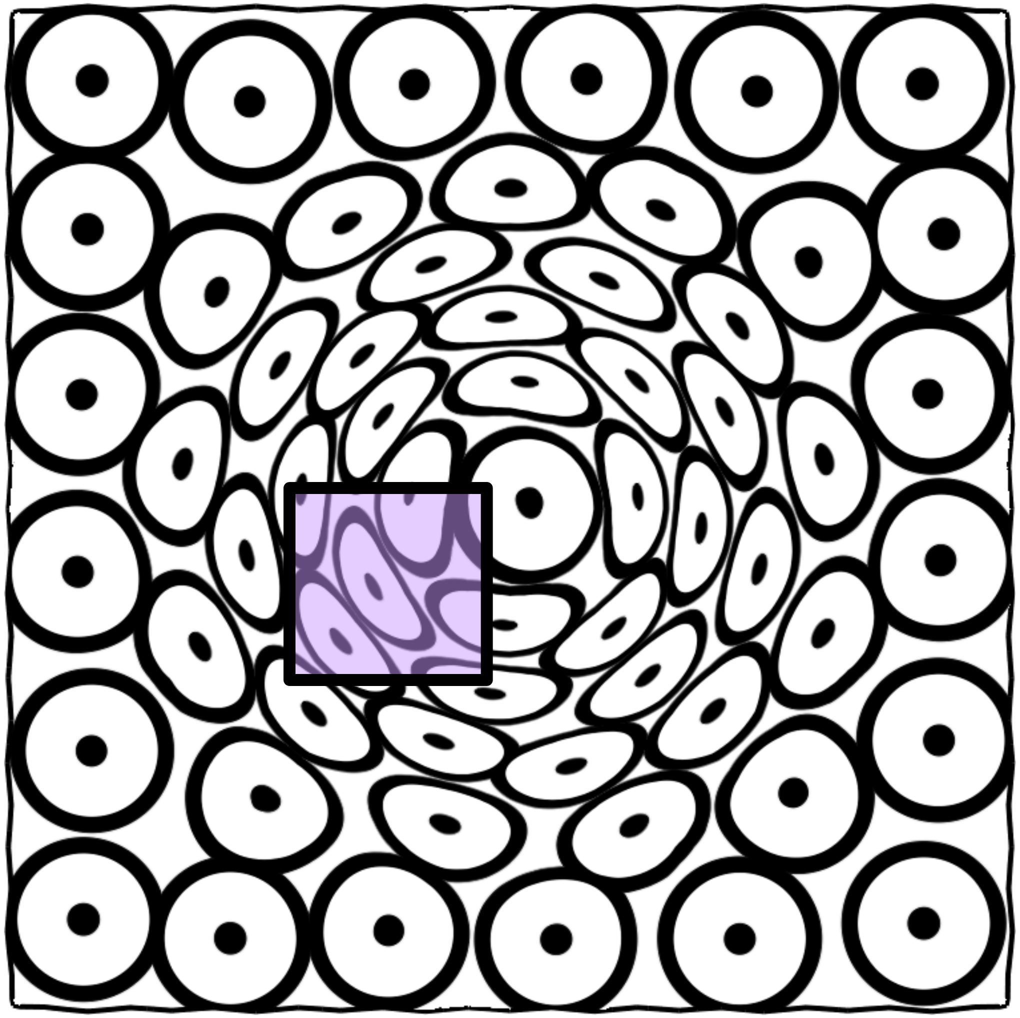
Part. 3. Anisotropy



The input: anisotropy field
Specifies shape and orientation

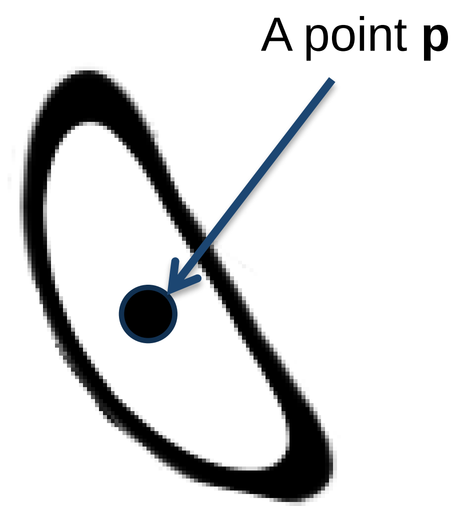
Anisotropy: An “alteration” of of distances and angles.

Part. 3. Anisotropy

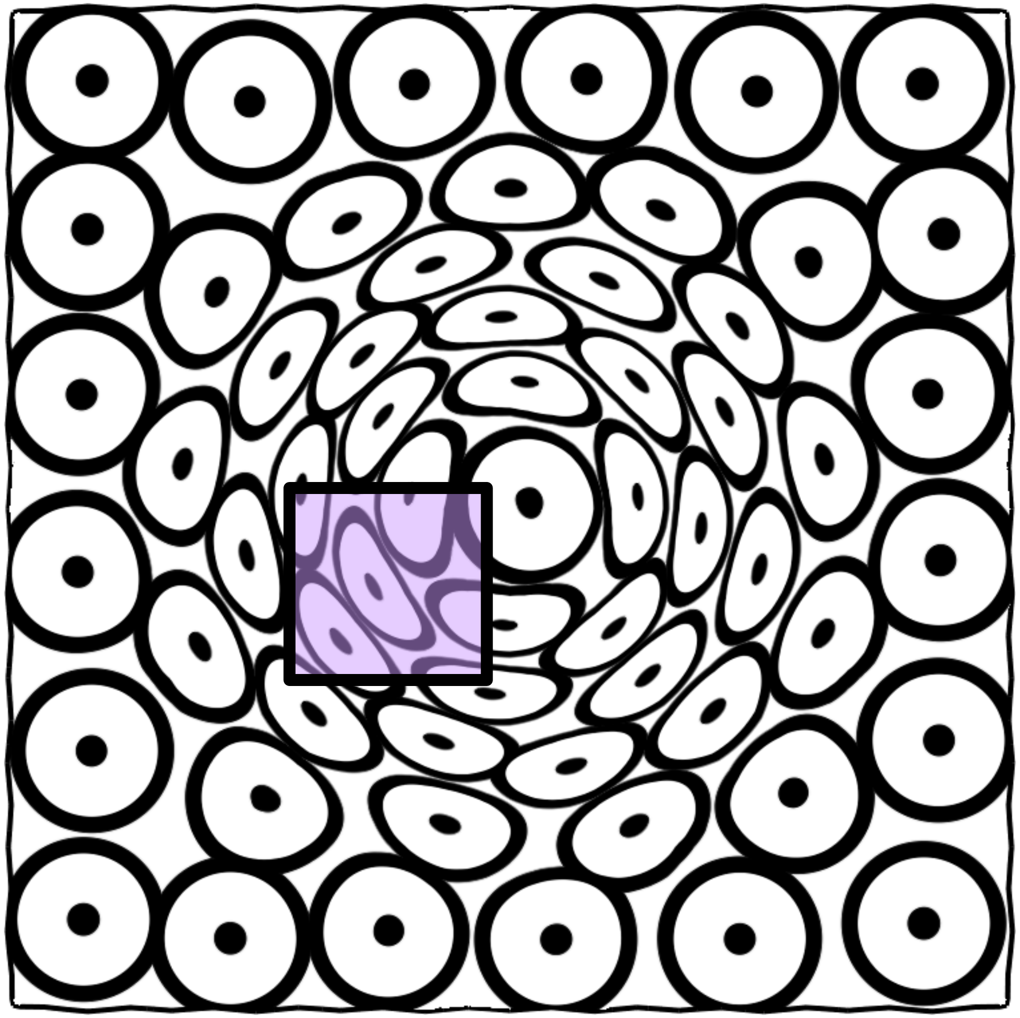


The input: anisotropy field
Specifies shape and orientation

Anisotropy: An "alteration" of of distances and angles.

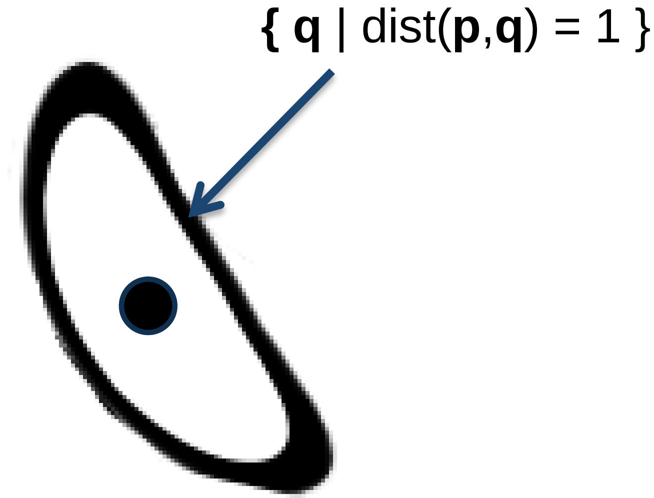


Part. 3. Anisotropy

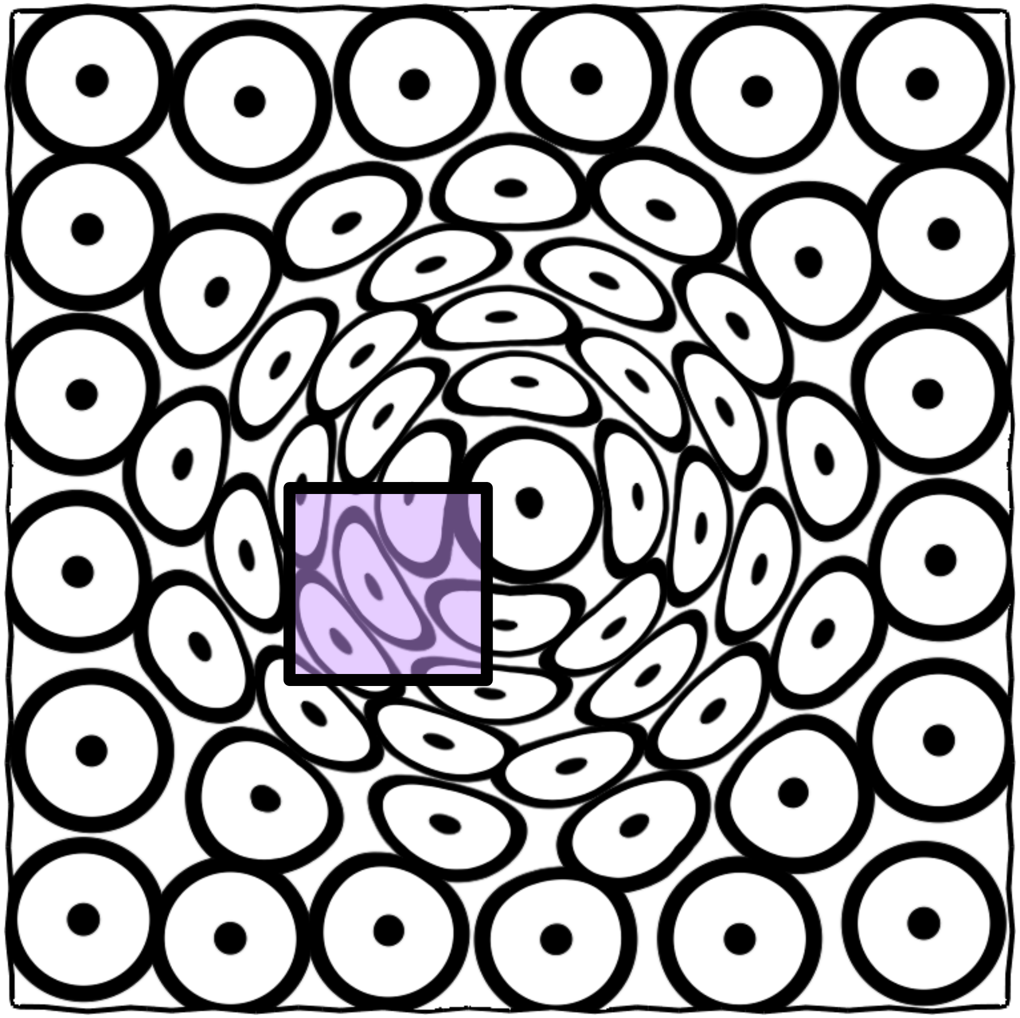


The input: anisotropy field
Specifies shape and orientation

Anisotropy: An "alteration" of
of distances and angles.



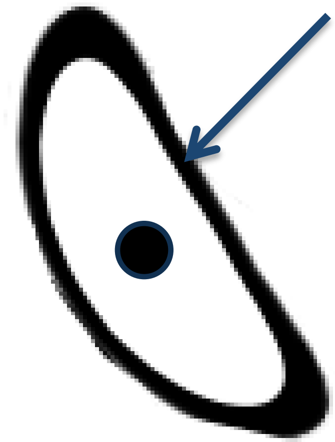
Part. 3. Anisotropy



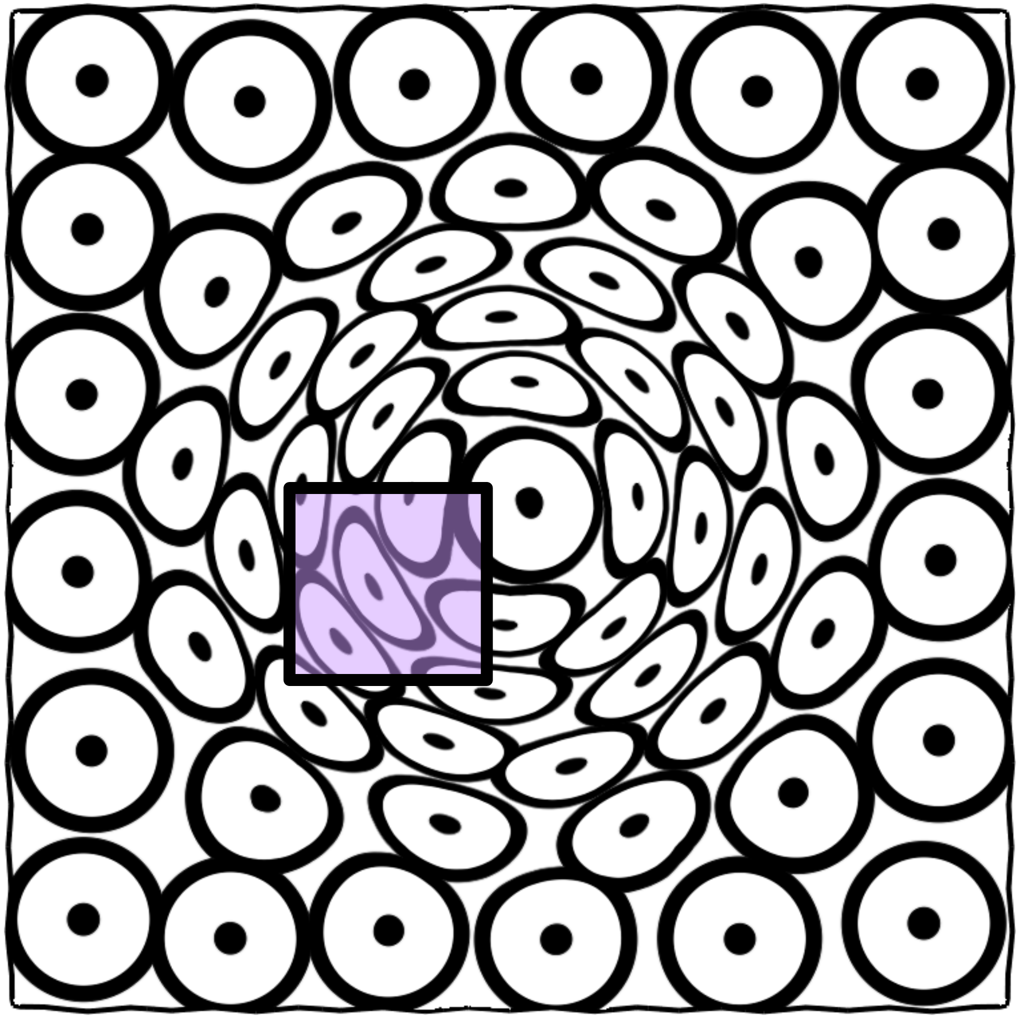
The input: anisotropy field
Specifies shape and orientation

Anisotropy: An “alteration” of of distances and angles.

$\{ \mathbf{q} \mid \underbrace{\mathrm{dist}(\mathbf{p},\mathbf{q})}_{\textit{anisotropic distance}} = 1 \}$



Part. 3. Anisotropy

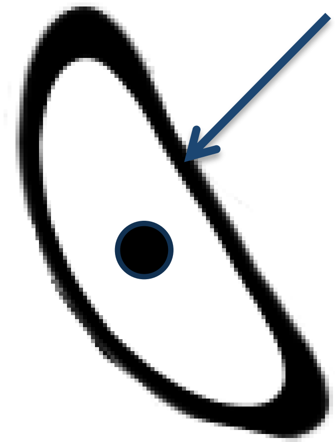


The input: anisotropy field
Specifies shape and orientation

Anisotropy: An “alteration” of of distances and angles.

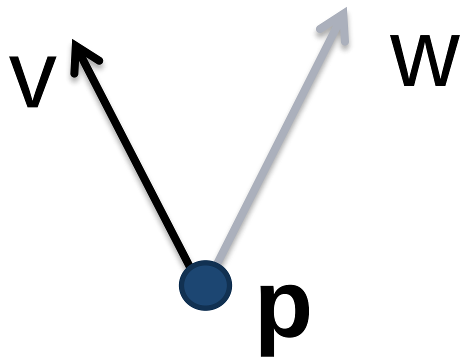
This is a circle !

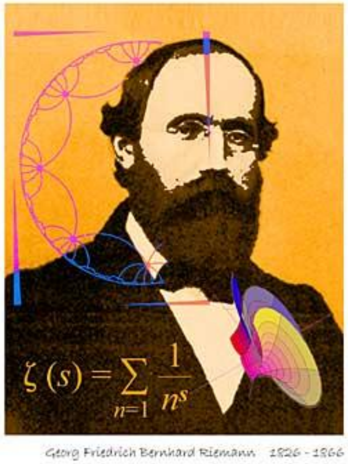
$\{ \mathbf{q} \mid \underbrace{\mathrm{dist}(\mathbf{p},\mathbf{q})}_{\textit{anisotropic distance}} = 1 \}$

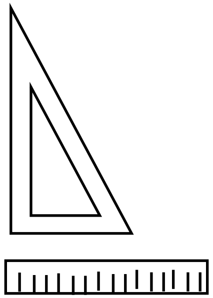


Part. 3. Anisotropy

The dot product: a geometric tool





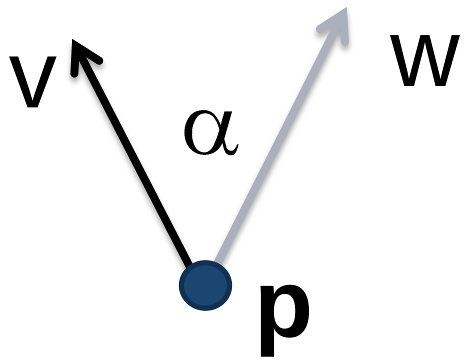


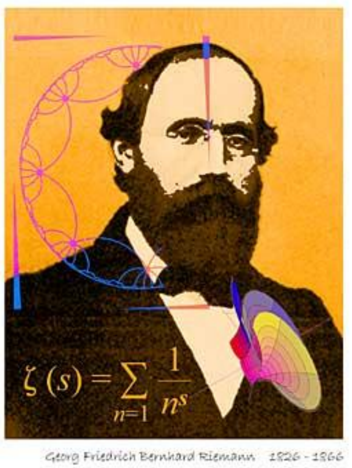
$$v \,.\, w = \langle v,w \rangle = v^t\, w$$

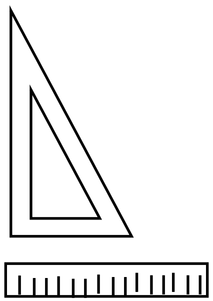
Part. 3. Anisotropy

The dot product: a geometric tool

Measuring angles





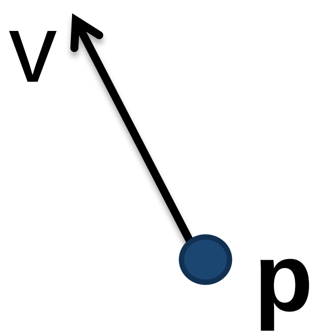


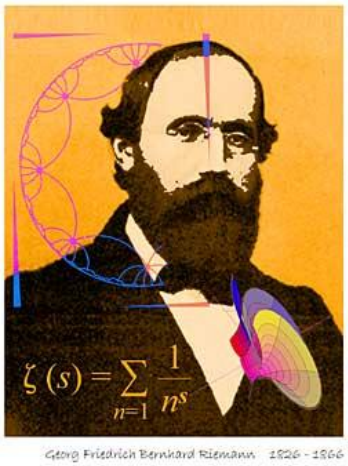
$$\cos(\alpha) = \langle v.w\rangle / \sqrt{\langle v.v\rangle \langle w.w\rangle}$$

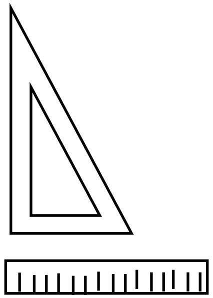
Part. 3. Anisotropy

The dot product: a geometric tool

Measuring length







$$||v|| = \sqrt{\langle v,v \rangle}$$

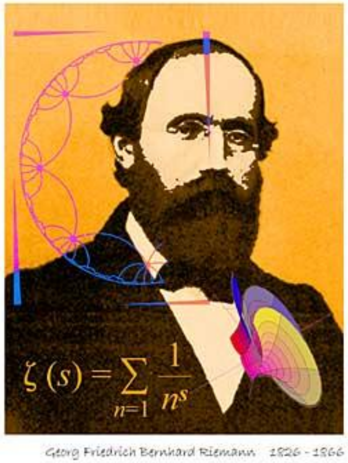
Part. 3. Anisotropy

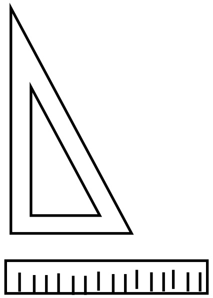
The dot product: a geometric tool

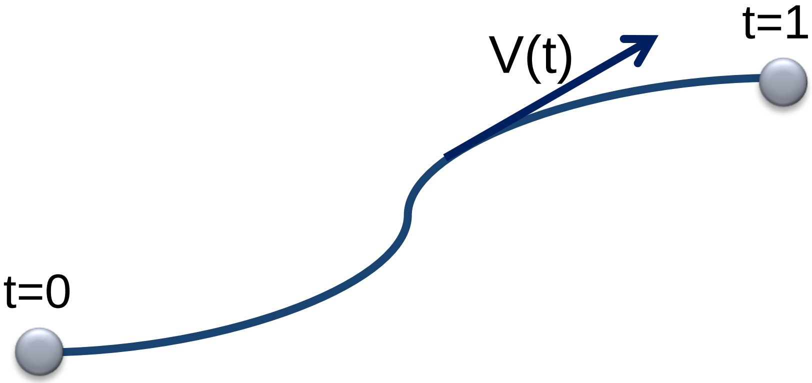
Measuring the length of a curve

$$l(C) = \int_{t=0}^{1} ||v(t)||\, dt$$

$$= \int_{t=0}^{1} \sqrt{\langle v(t), v(t) \rangle}\, dt$$



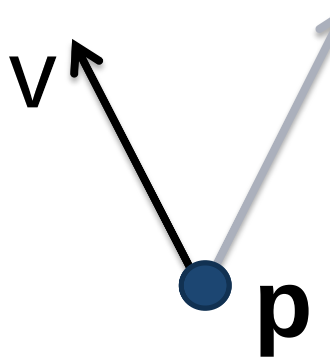


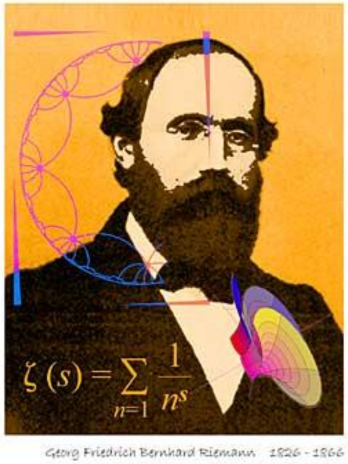


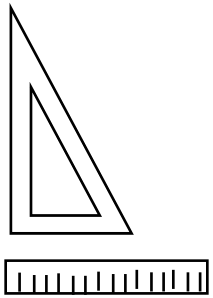
Part. 3. Anisotropy

The dot product: a geometric tool

Changing the dot product





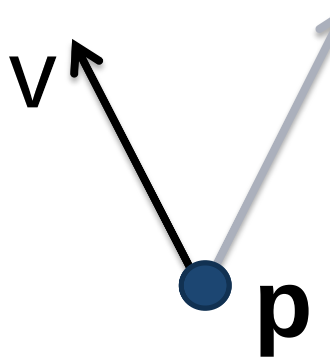


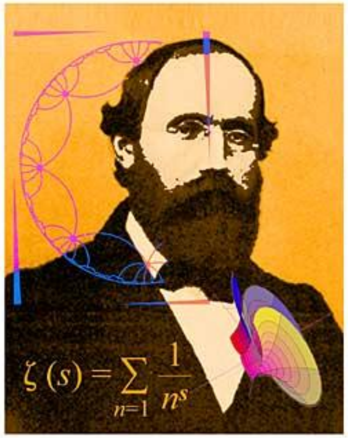
$$v \,.\, w = \langle v,w \rangle = v^t \, Id \, w$$

Part. 3. Anisotropy

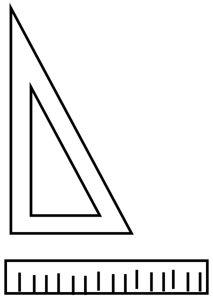
The dot product: a geometric tool

Changing the dot product





Georg Friedrich Bernhard Riemann 1826 - 1866



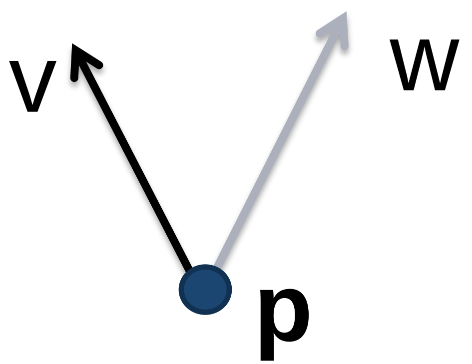
$$v \,.\, w = \langle v,w \rangle \quad = v^t \, \mathrm{Id} \, w$$

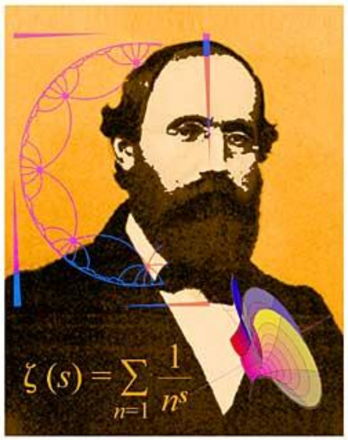
$$\langle v,w \rangle_G = v^t \, G(\mathbf{p}) \, w$$

Part. 3. Anisotropy

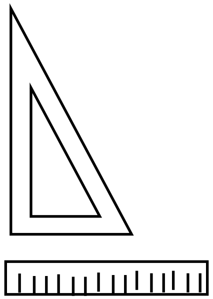
The dot product: a geometric tool

Changing the dot product





Georg Friedrich Bernhard Riemann 1826 - 1866



$$v \,.\, w = \langle v,w \rangle = v^t \, Id \, w$$

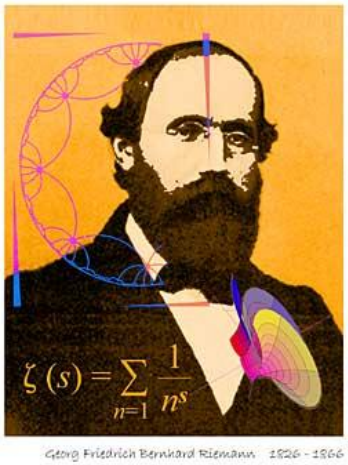
$$\langle v,w \rangle_G = v^t \, G(p) \, w$$

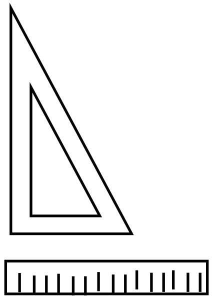
A 2x2 symmetric matrix that depends on p

Part. 3. Anisotropy

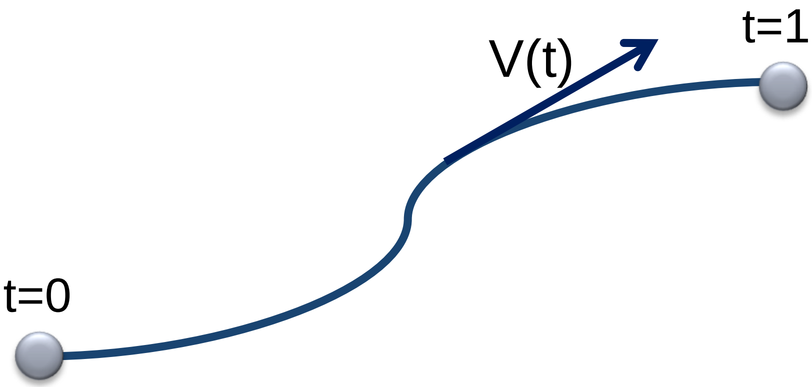
The dot product: a geometric tool

Measuring the anisotropic length of a curve





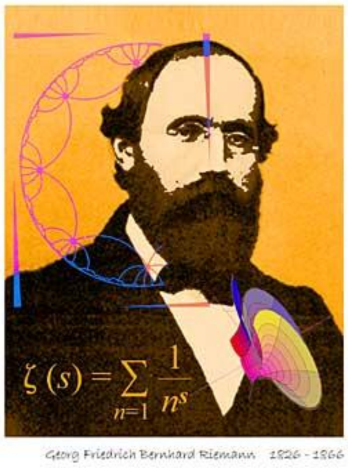
$$l_G(C) = \int_{t=0}^{1} \sqrt{v(t)^t\, G(t)\, v(t)}\, dt$$

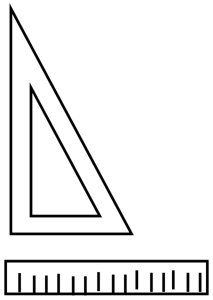


Part. 3. Anisotropy

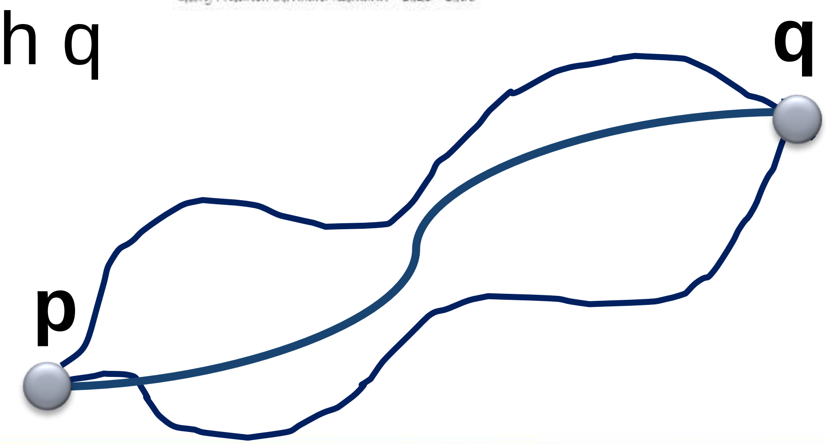
The dot product: a geometric tool

Anisotropic distance
between **p** *and* **q** *w.r.t.* G



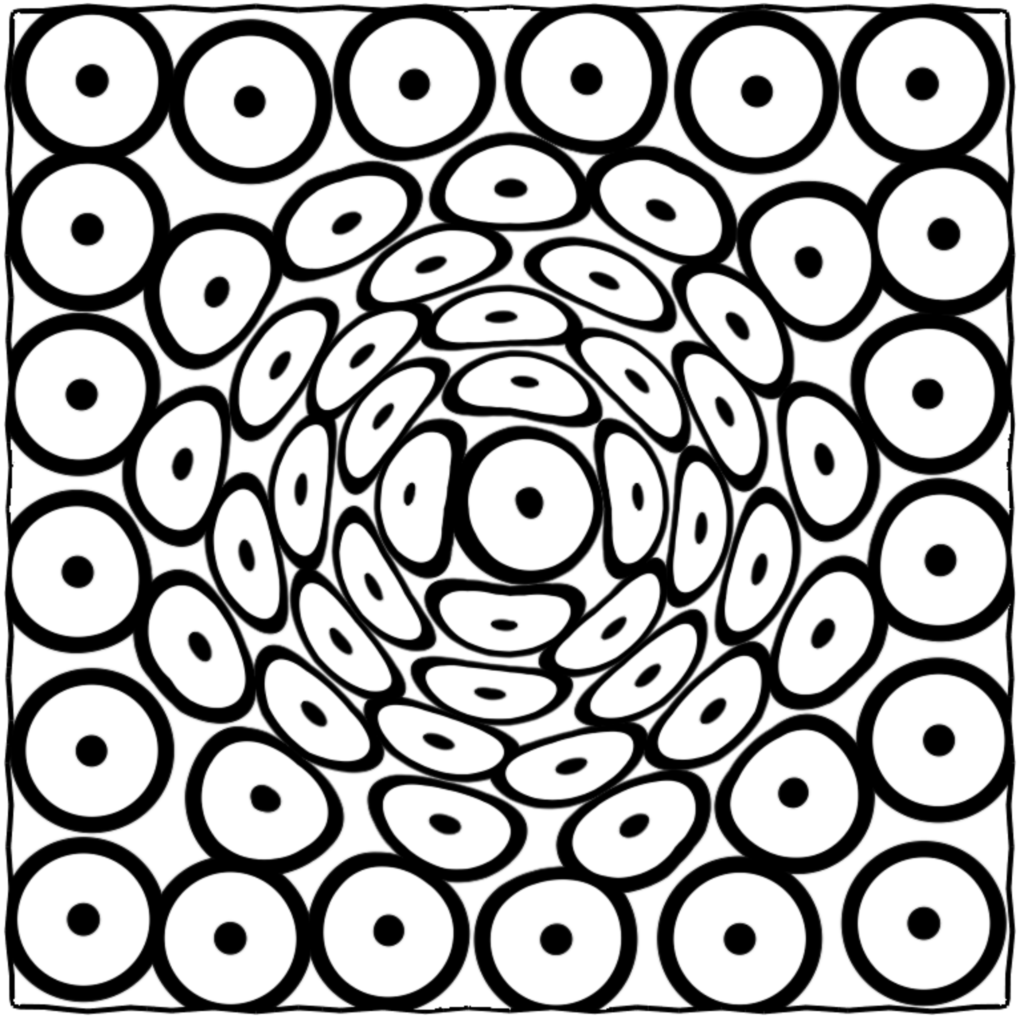


$d_G(\mathbf{p},\mathbf{q})$ = (anisotropic) length of
shortest curve
that connects p with q



$$l_G(C) = \int_{t=0}^{1} \sqrt{v(t)^t\, G(t)\, v(t)}\, dt$$

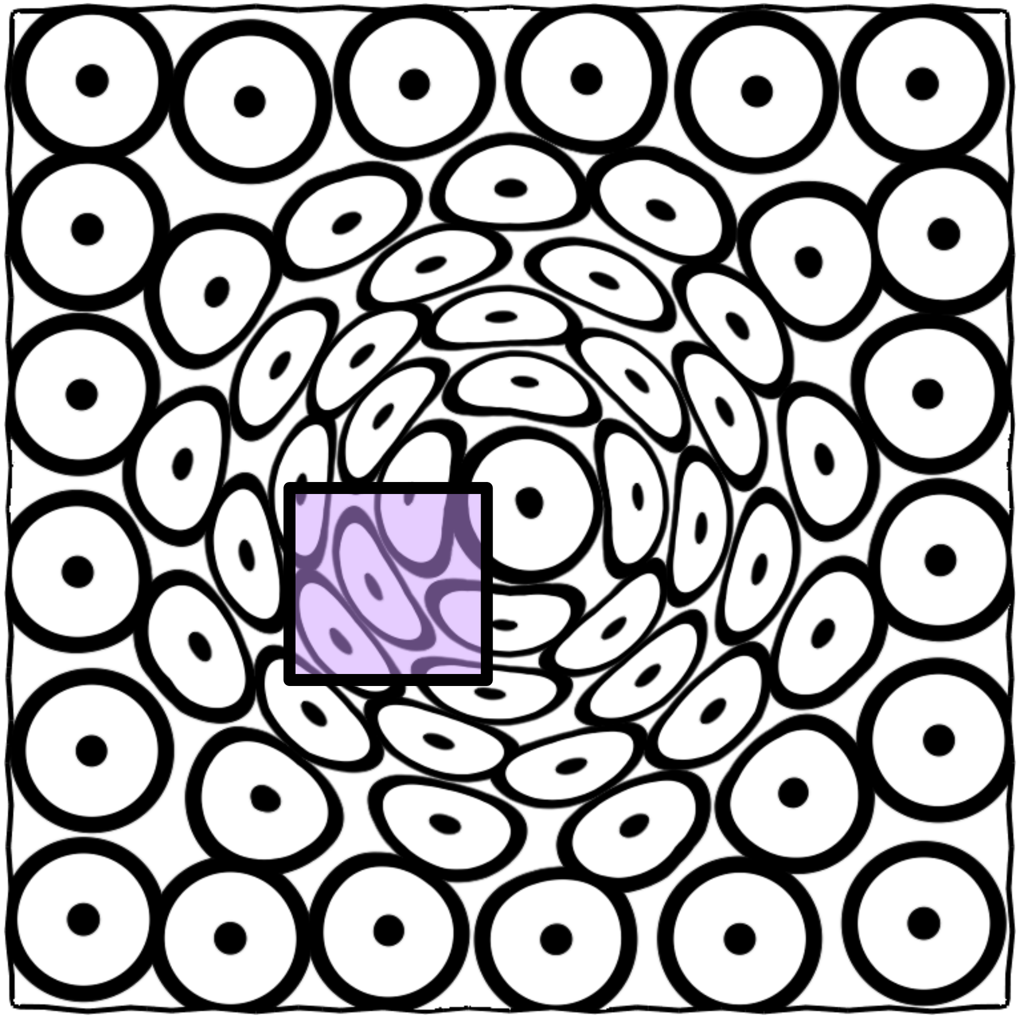
Part. 3. Anisotropy



The input: anisotropy field

$$G(x,y) = \begin{bmatrix} a(x,y) & b(x,y) \\ b(x,y) & c(x,y) \end{bmatrix}$$

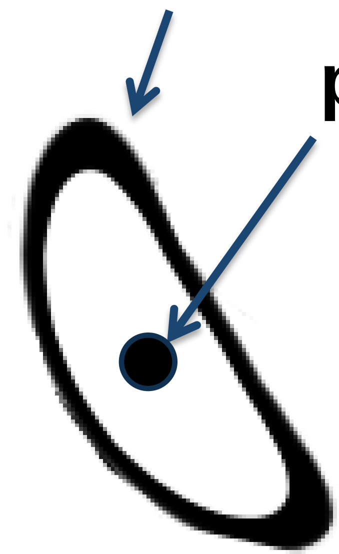
Part. 3. Anisotropy



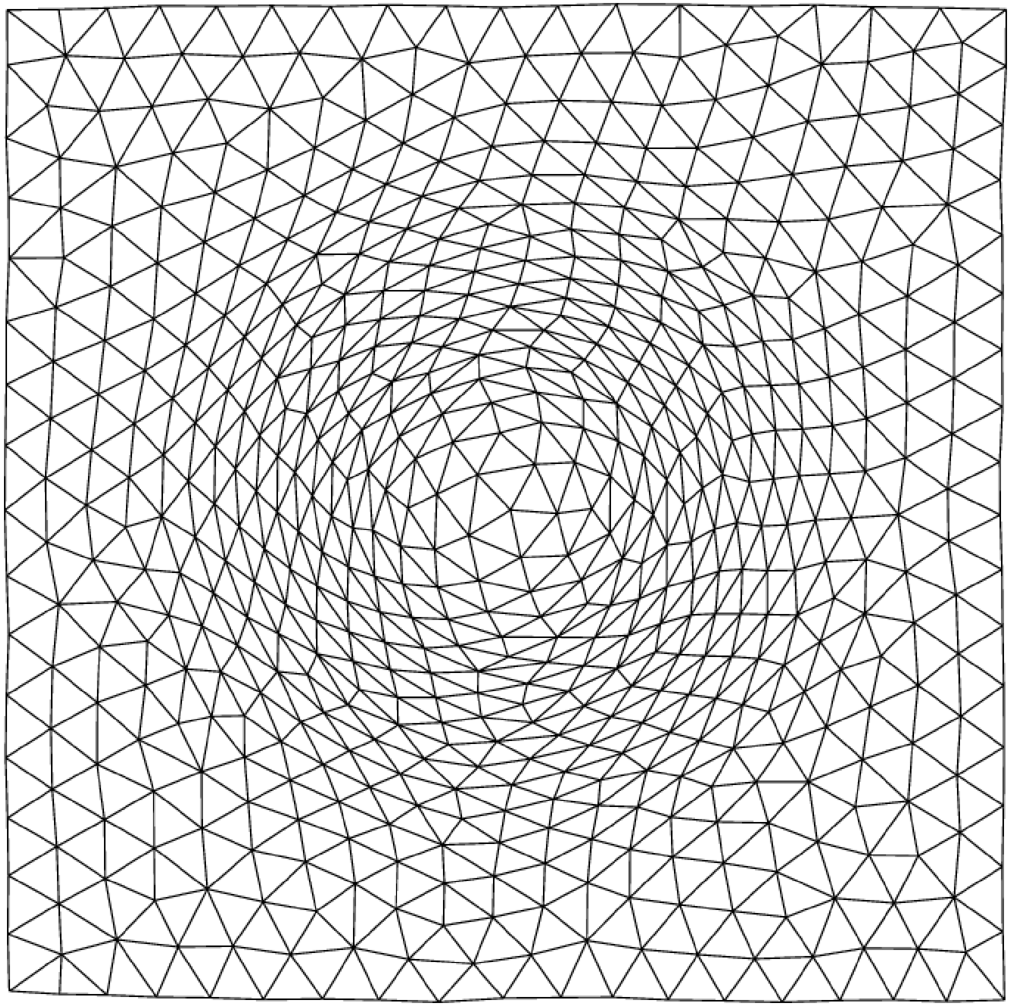
The input: anisotropy field

$$G(x,y) = \begin{bmatrix} a(x,y) & b(x,y) \\ b(x,y) & c(x,y) \end{bmatrix}$$

$$\{ \mathbf{q} \mid d_G(\mathbf{p},\mathbf{q}) = 1 \}$$

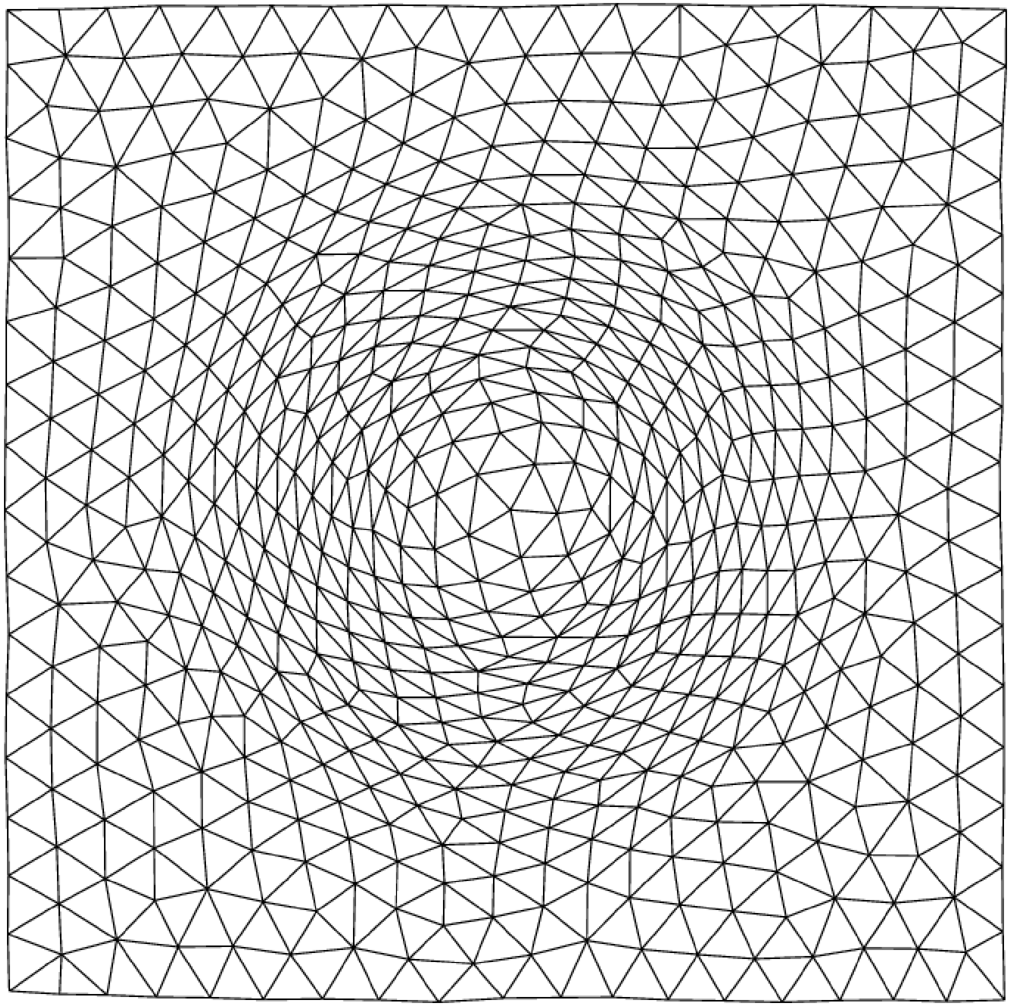


Part. 3. Anisotropy



<u>The result:</u> triangles are "deformed" by the anisotropy.

Part. 3. Anisotropy



The result: triangles are "deformed" by the anisotropy.

*Q: How to compute an **Anisotropic** Centroidal Voronoi Tessellation ?*

4

Journey in the 6th dimension

… and beyond

Part. 4 Journey in the 6th dimension

The key idea

This example:

Anisotropic mesh in 2d $\Longleftrightarrow$ Isotropic mesh in 3d

Part. 4 Journey in the 6^{th} dimension

The key idea

This example:

Anisotropic mesh in 2d ⟺ Isotropic mesh in 3d

Replace **anisotropy** with **additional dimensions**

Part. 4 Journey in the 6th dimension

The key idea

Replace **anisotropy** with **additional dimensions**

Note: more dimensions may be needed

Part. 4 Journey in the 6th dimension

The key idea

Replace **anisotropy** with **additional dimensions**

Note: more dimensions may be needed
How many ?

Part. 4 Journey in the 6th dimension

The key idea

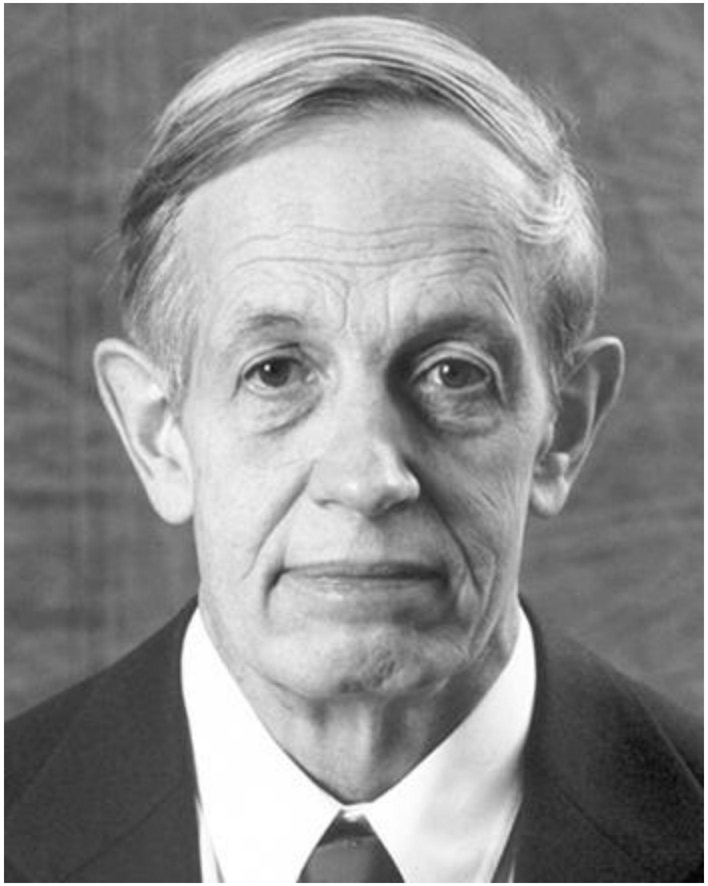
Replace **anisotropy** with **additional dimensions**

Note: more dimensions may be needed
How many ?
John Nash's isometric embedding theorem:

Maximum: depending on desired smoothness
C^1 : $2n$ [Nash-Kuiper]
C^k : bounded by $n(3n+11)/2$ [Nash, Nash-Moser]

Part. 4 Journey in the 6^{th} dimension

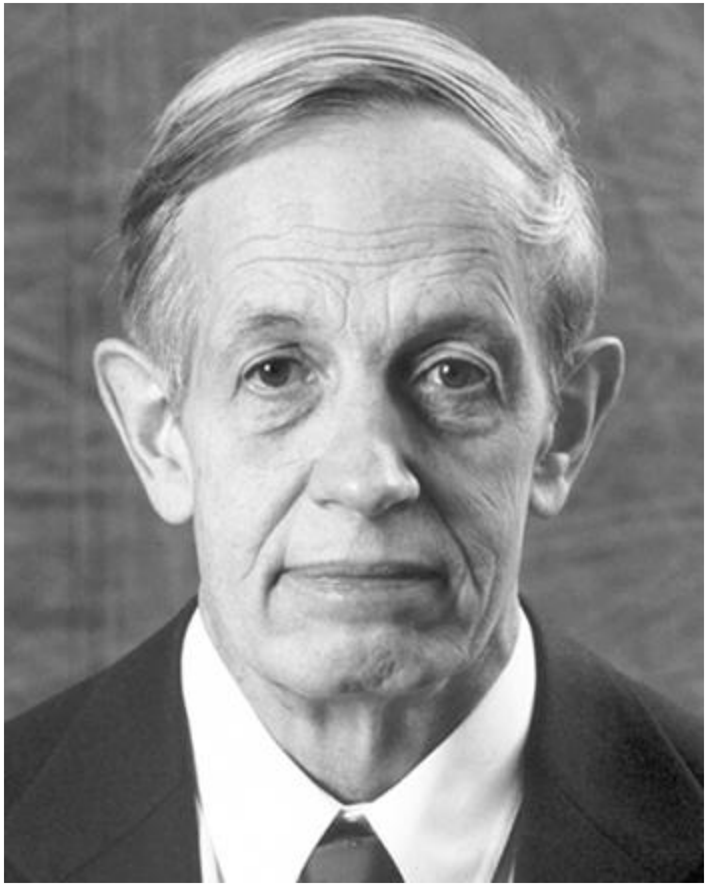
Two words about John Nash



- Isometric embedding theorem
- Nash Equilibrium → Nobel prize of economy

Part. 4 Journey in the 6th dimension

Two words about John Nash



- Isometric embedding theorem
- Nash Equilibrium → Nobel prize of economy

The **existence** is proved, but it does not tell me **how to compute** the embedding given a specified surface and anisotropy field.

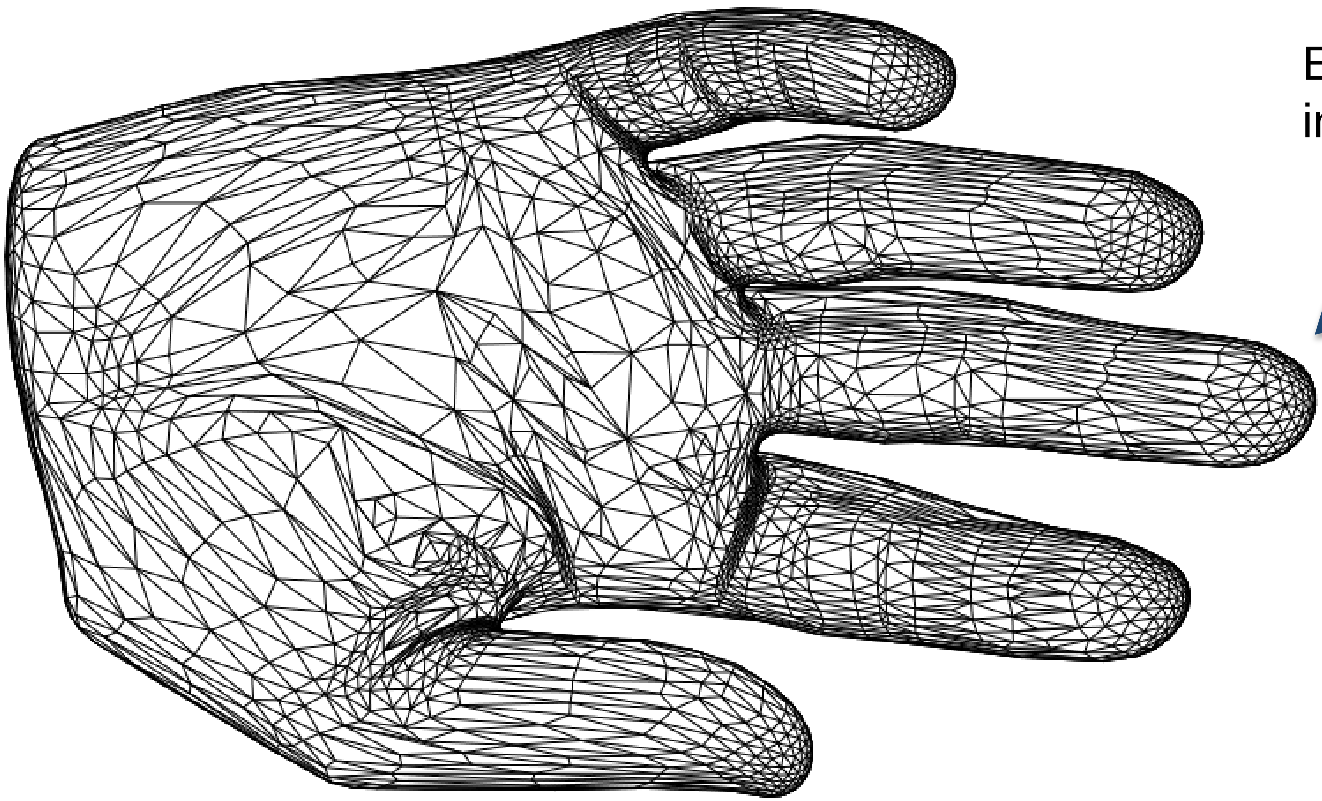
Part. 4 Journey in the 6th dimension

Convex integration – Flat Torus

[Borelli, Jabrane, Lazarus, Thibert]

Part. 4 Journey in the 6^{th} dimension

A 6d embedding for curvature-adapted meshing

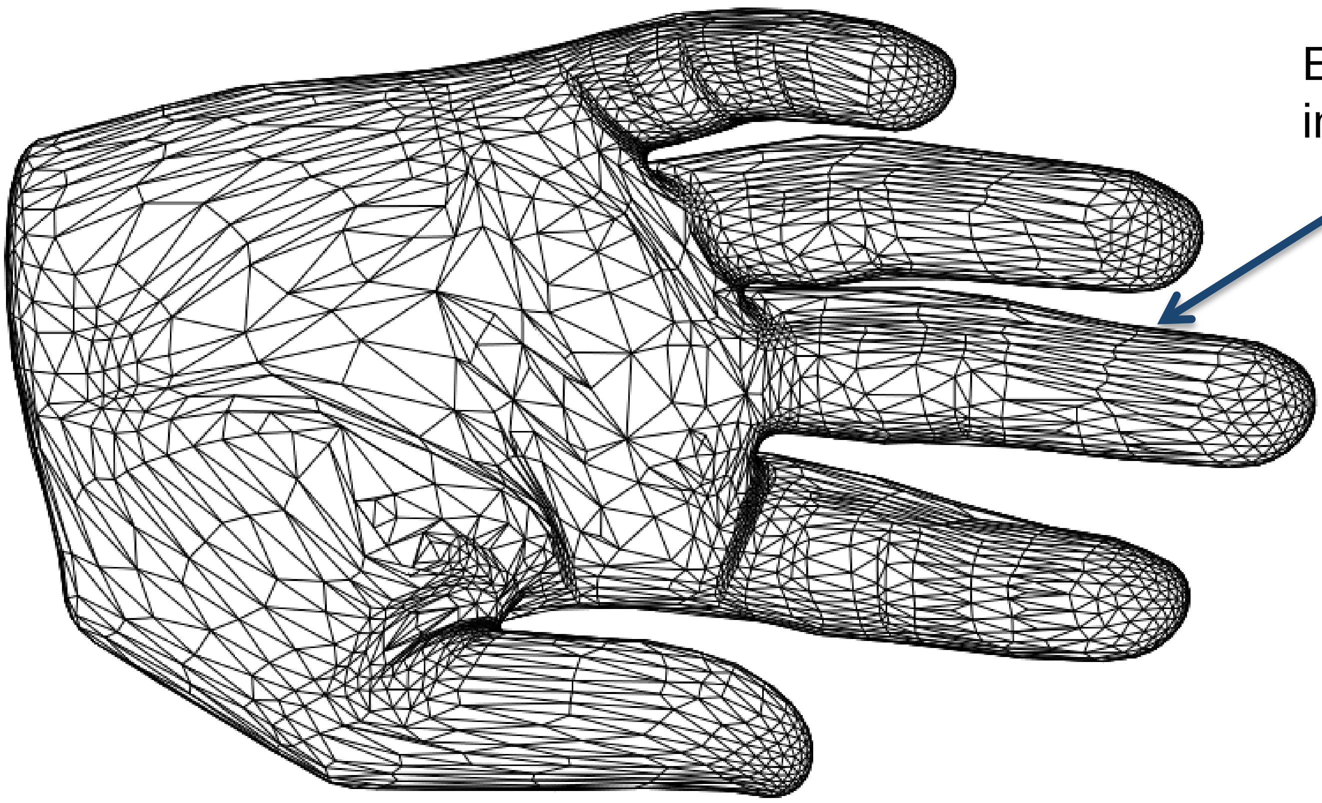


What we want:

Equilateral triangles
in spherical zones

Part. 4 Journey in the 6th dimension

A 6d embedding for curvature-adapted meshing



What we want:

Elongated triangles
in cylindrical zones

Part. 4 Journey in the 6th dimension

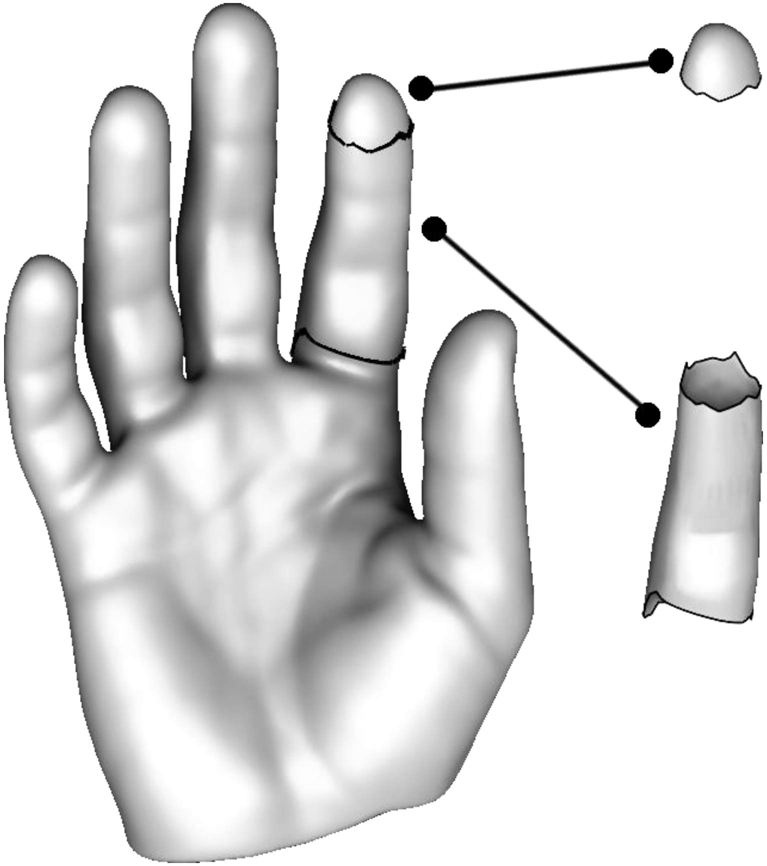
A 6d embedding for curvature-adapted meshing

The Gauss map: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix}$

Part. 4 Journey in the 6^{th} dimension

A 6d embedding for curvature-adapted meshing

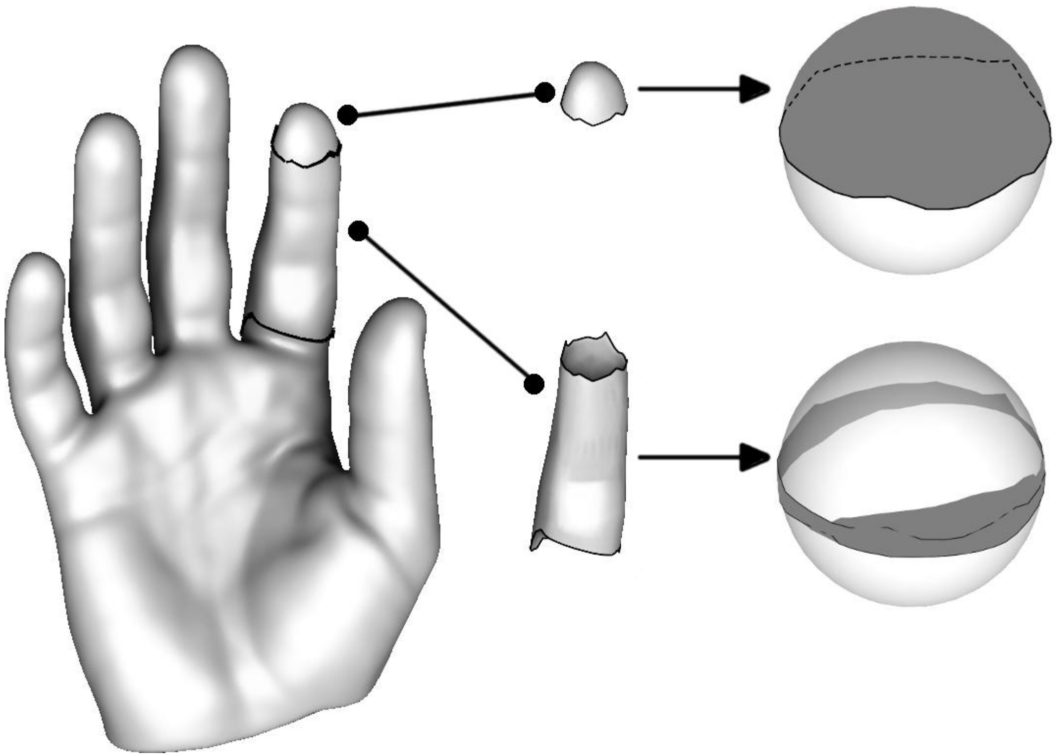
The Gauss map: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix}$



Part. 4 Journey in the 6th dimension

A 6d embedding for curvature-adapted meshing

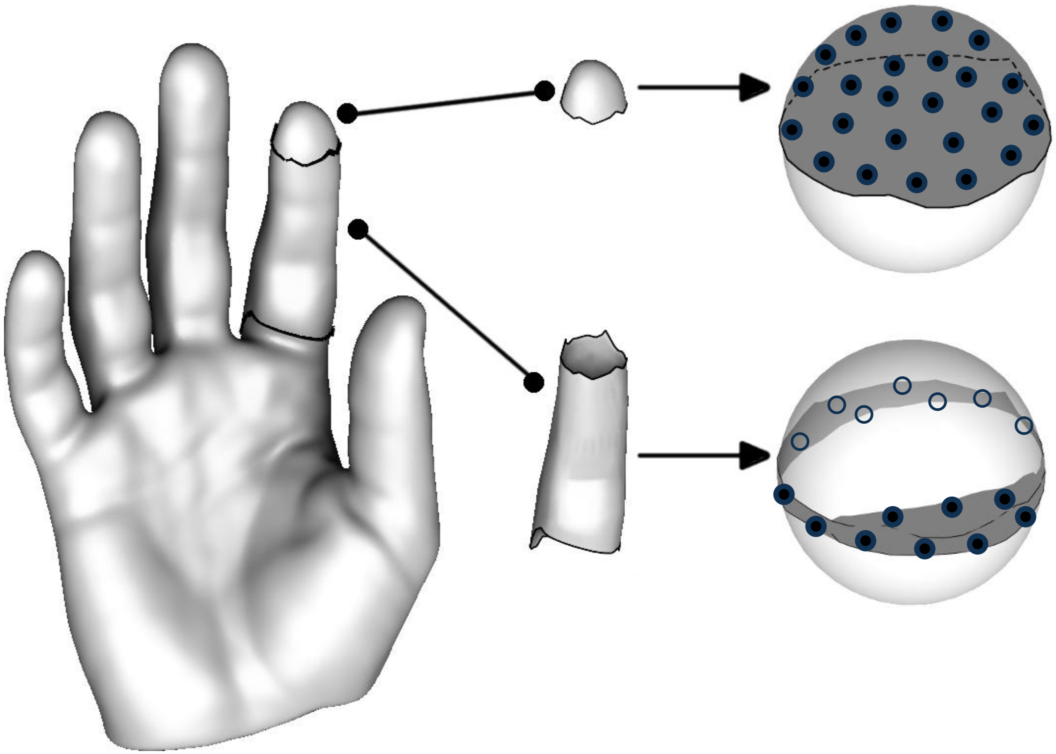
The Gauss map: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \rightarrow \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix}$



Part. 4 Journey in the 6^{th} dimension

A 6d embedding for curvature-adapted meshing

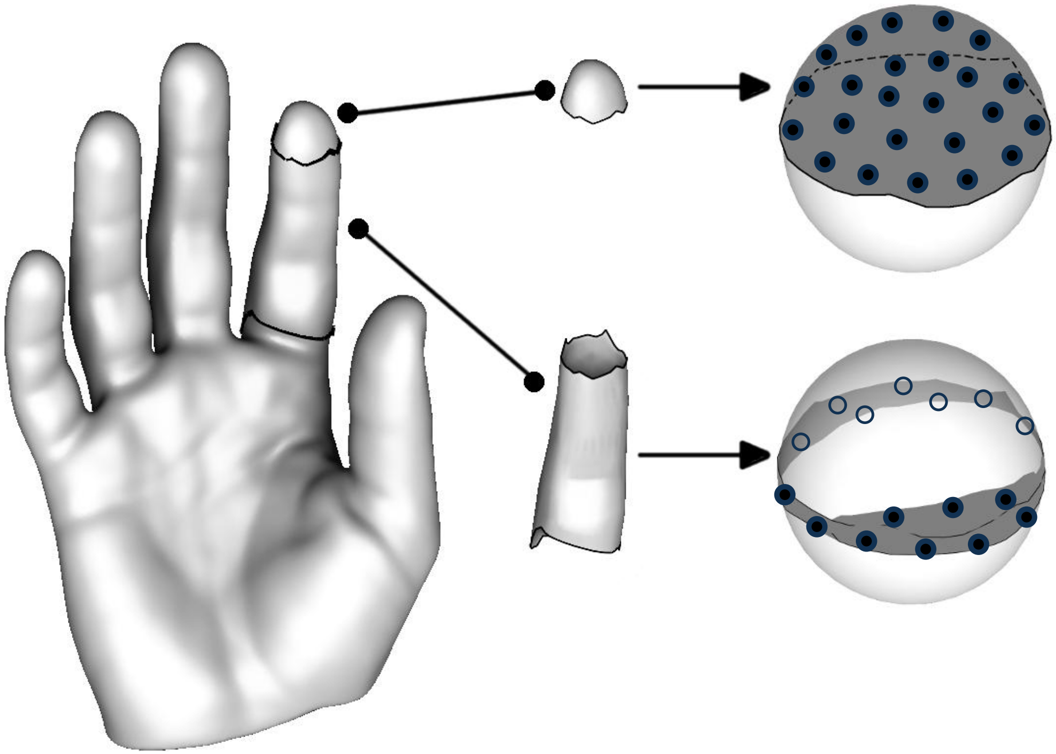
The Gauss map: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix}$

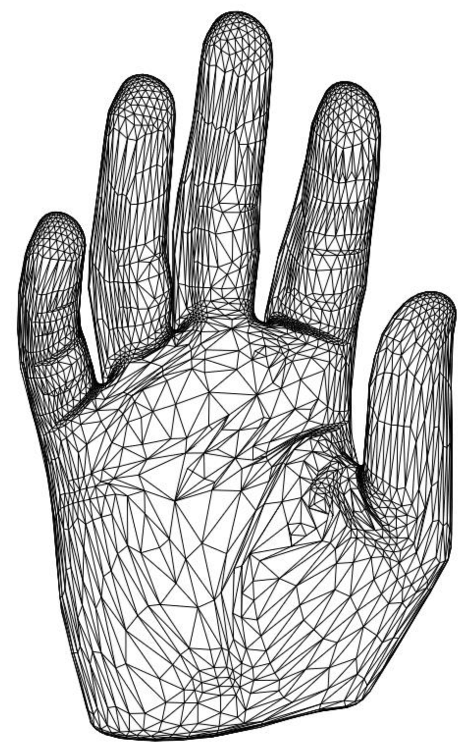


Part. 4 Journey in the 6^{th} dimension

A 6d embedding for curvature-adapted meshing

The Gauss map: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix}$





Part. 4 Journey in the 6th dimension

A 6d embedding for curvature-adapted meshing

The Gauss map: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix}$

The Gauss-map is **non-bijective** in general *(bijective only if convex object)*

Part. 4 Journey in the 6^{th} dimension

A 6d embedding for curvature-adapted meshing

The Gauss map: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix}$

The Gauss-map is **non-bijective** in general *(bijective only if convex object)*

Our embedding: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ z \\ sN_x \\ sN_y \\ sN_z \end{bmatrix}$

Part. 4 Journey in the 6th dimension

A 6d embedding for curvature-adapted meshing

Our embedding: $\begin{bmatrix} x \\ y \\ z \end{bmatrix} \Rightarrow \begin{bmatrix} x \\ y \\ z \\ sN_x \\ sN_y \\ sN_z \end{bmatrix}$

s: desired amount of anisotropy

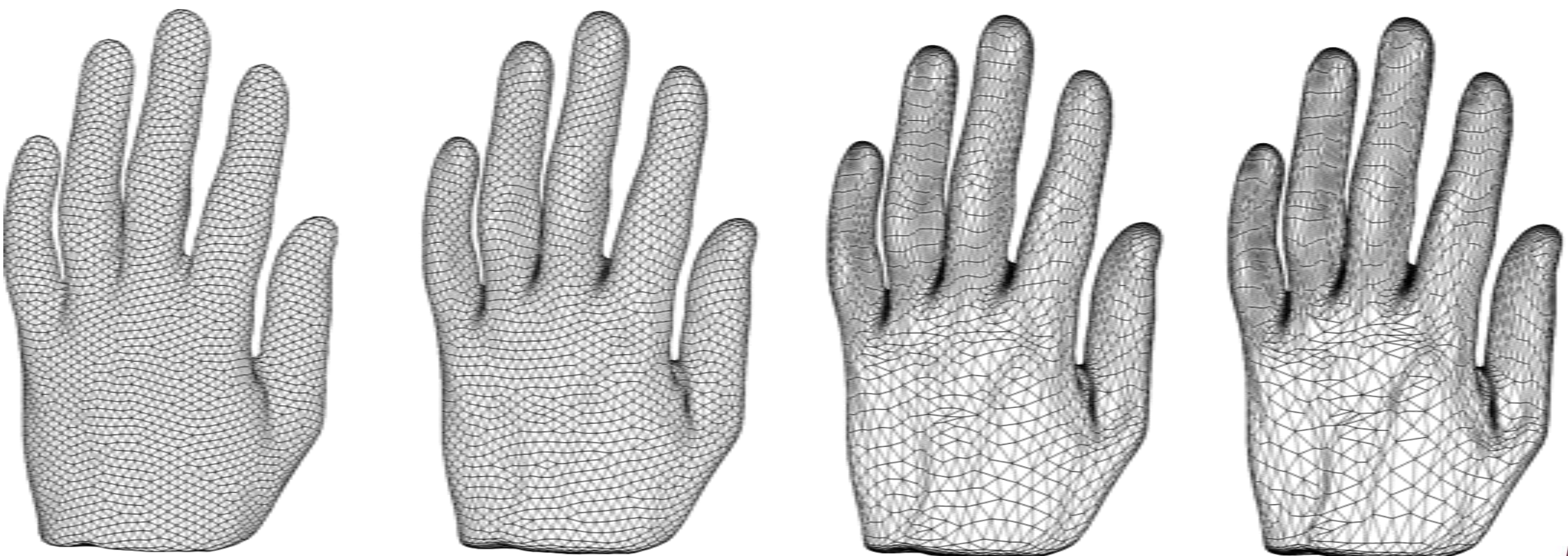
Part. 4 Journey in the 6^{th} dimension

A 6d embedding for curvature-adapted meshing

s: desired amount of anisotropy

Small s

Large s



5

The algorithm

Part. 5 The algorithm

Lloyd relaxation in $\mathbb{R}^6$ (Naïve version)

(1) Embed the surface **S** into $\mathbb{R}^6$

Part. 5 The algorithm

Lloyd relaxation in IR^6 (Naïve version)

(1) Embed the surface **S** into IR^6
(2) Compute initial point distrib. **X**

Part. 5 The algorithm

Lloyd relaxation in $\mathrm{I\!R}^6$ (Naïve version)

(1) Embed the surface **S** into $\mathrm{I\!R}^6$
(2) Compute initial point distrib. **X**
While convergence is not reached

Part. 5 The algorithm

Lloyd relaxation in $\mathbb{R}^6$ (Naïve version)

(1) Embed the surface **S** into $\mathbb{R}^6$
(2) Compute initial point distrib. **X**
While convergence is not reached
    (3) Compute Vor(**X**)

Part. 5 The algorithm

Lloyd relaxation in IR^6 (Naïve version)

(1) Embed the surface **S** into IR^6
(2) Compute initial point distrib. **X**
While convergence is not reached
    (3) Compute Vor(**X**)
    (4) Compute Vor(**X**) ∩ S

Part. 5 The algorithm

Lloyd relaxation in IR^6 (Naïve version)

(1) Embed the surface **S** into IR^6
(2) Compute initial point distrib. **X**
While convergence is not reached
  (3) Compute Vor(**X**)
  (4) Compute Vor(**X**) ∩ S
  (5) Move each $\mathbf{x}_i$ to the centroid of Vor($\mathbf{x}_i$) ∩ S

Part. 5 The algorithm

Lloyd relaxation in $\mathbb{R}^6$ (Naïve version)

(1) Embed the surface **S** into $\mathbb{R}^6$

(2) Compute initial point distrib. **X**

While convergence is not reached

(3) Compute Vor(**X**)

Costs d! for dimension d

(4) Compute Vor(**X**) ∩ S

(5) Move each $\mathbf{x}_i$ to the centroid of Vor($\mathbf{x}_i$) ∩ S

Part. 5 The algorithm

Lloyd relaxation in $\mathbb{R}^6$ (Naïve version)

(1) Embed the surface **S** into $\mathbb{R}^6$
(2) Compute initial point distrib. **X**
While convergence is not reached
    (3) Compute Vor(**X**)
    (4) Compute Vor(**X**) ∩ S
    (5) Move each $\mathbf{x}_i$ to the centroid of Vor($\mathbf{x}_i$) ∩ S

Costs d! for dimension d
d = 6 ; d! = 720

Part. 5 The algorithm

Lloyd relaxation in $\mathbb{R}^6$ (Naïve version)

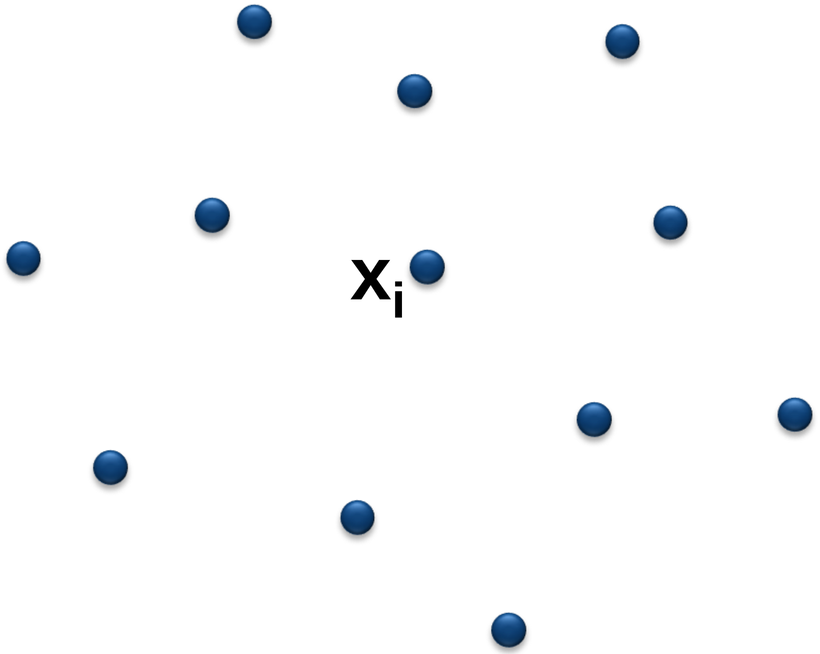
(1) Embed the surface **S** into $\mathbb{R}^6$
(2) Compute initial point distrib. **X**
While convergence is not reached
(3) Compute Vor(**X**)
(4) Compute Vor(**X**) ∩ S
(5) Move each $\mathbf{x}_i$ to the centroid of Vor($\mathbf{x}_i$) ∩ S

Costs d! for dimension d
d = 6 ; d! = 720

Curse of dimensionality
Some theoretical results
existence of bounds – Tangent Delaunay Complex [Boissonnat et.al.]

Part. 5 The algorithm

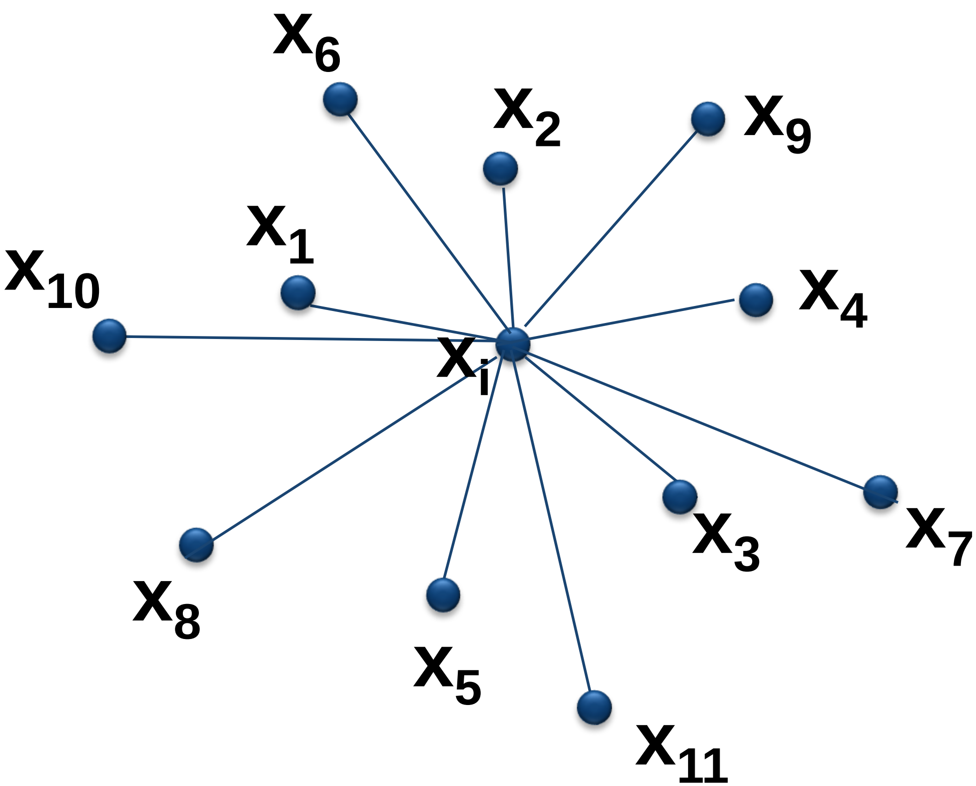
Voronoi cells as iterative convex clipping



Part. 5 The algorithm

Voronoi cells as iterative convex clipping

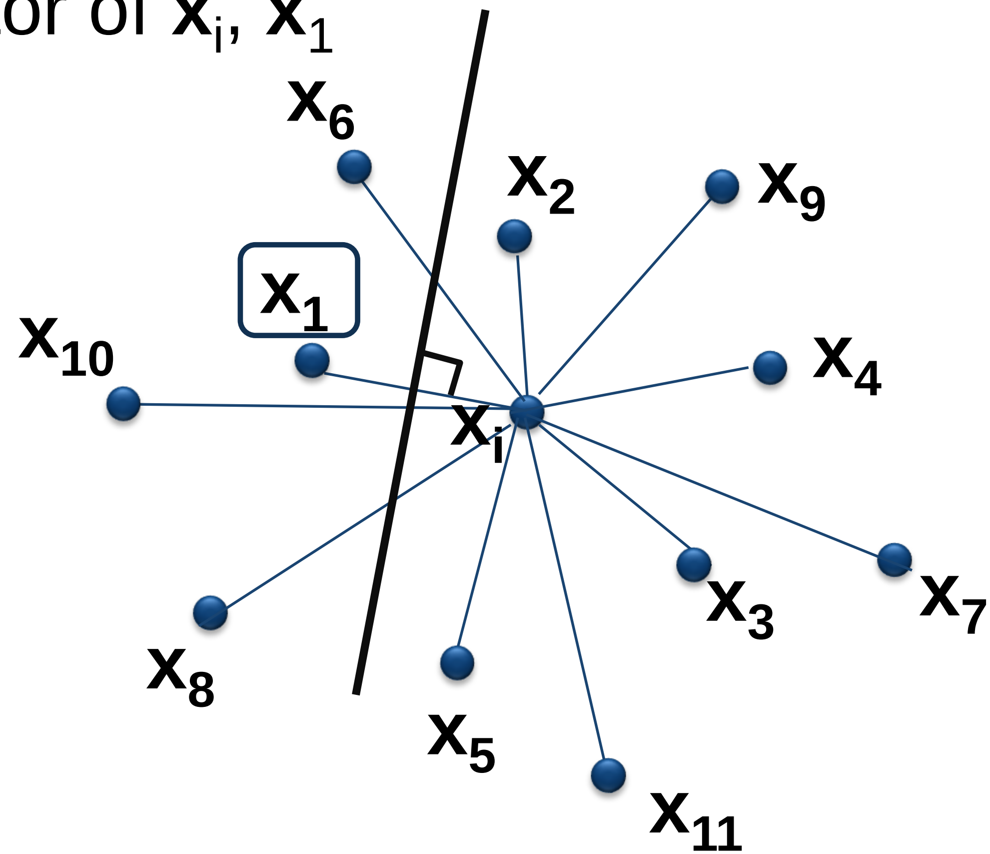
Neighbors in increasing (6d) distance from $\mathbf{x}_i$



Part. 5 The algorithm

Voronoi cells as iterative convex clipping

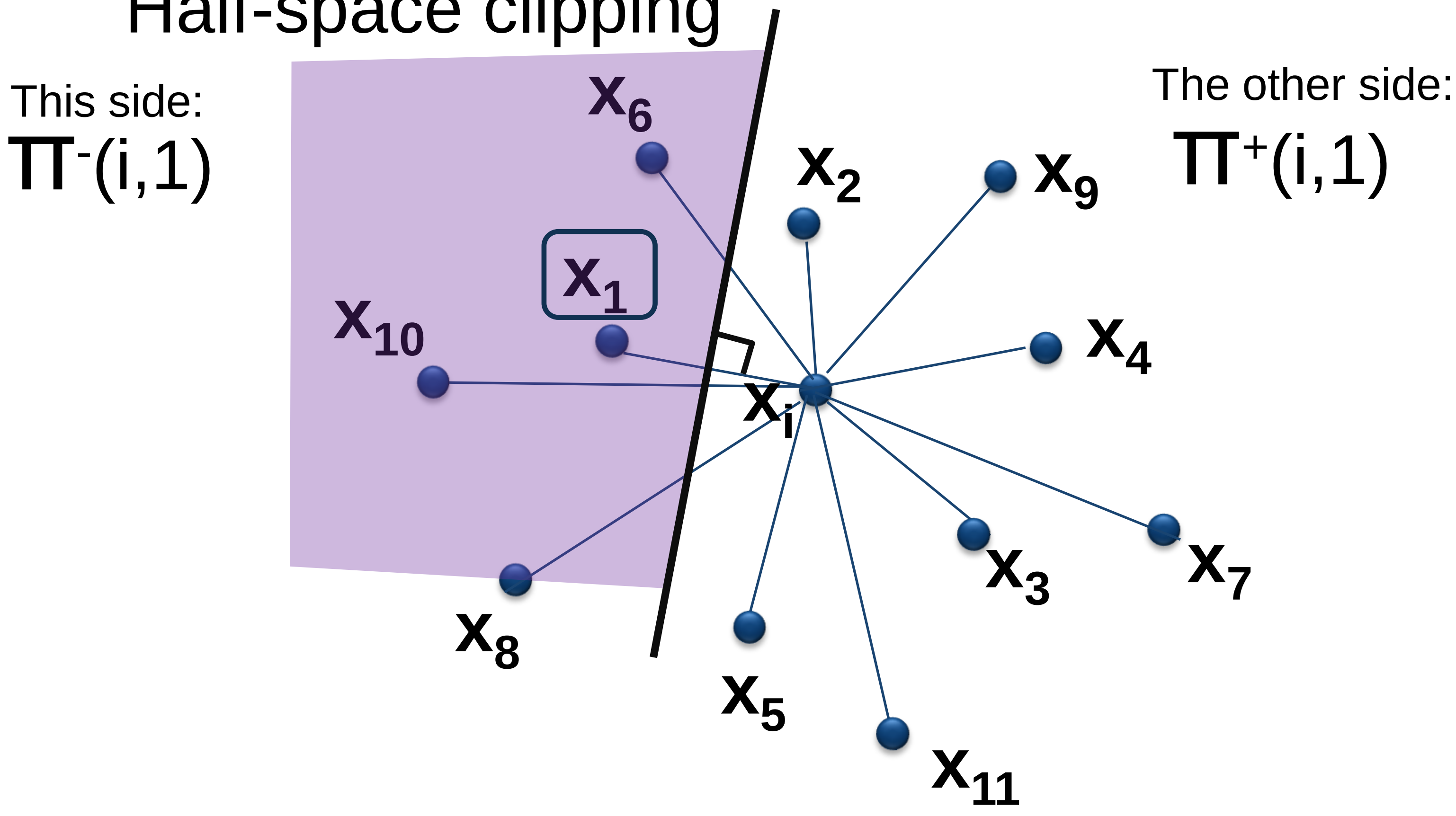
Bisector of $\mathbf{x}_i$, $\mathbf{x}_1$



Part. 5 The algorithm

Voronoi cells as iterative convex clipping

Half-space clipping



Part. 5 The algorithm

Voronoi cells as iterative convex clipping

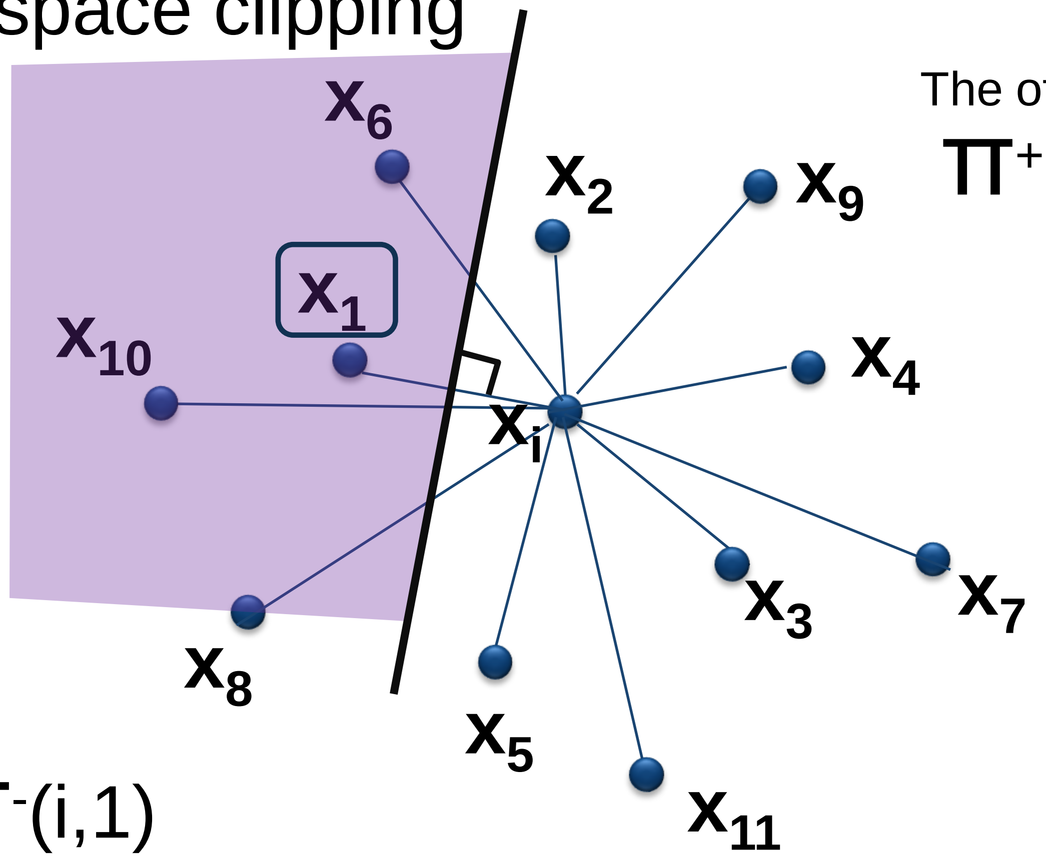
Half-space clipping

This side:

$\Pi^{-}(i,1)$

The other side:

$\Pi^{+}(i,1)$

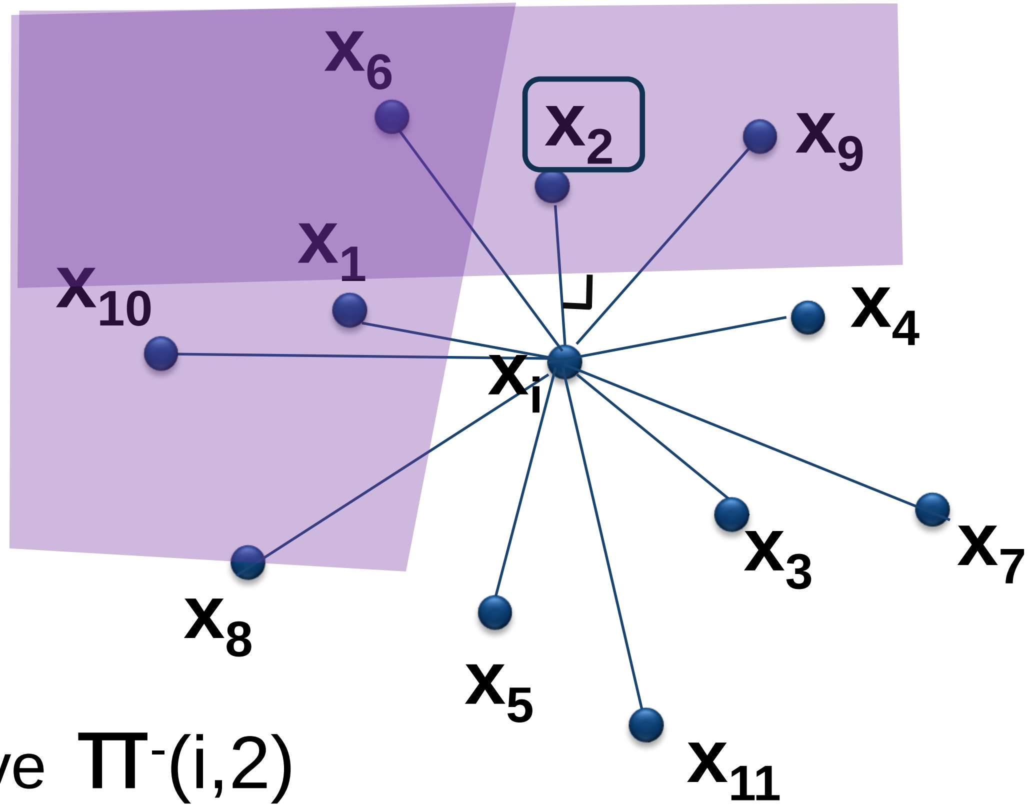


Remove $\Pi^{-}(i,1)$

Part. 5 The algorithm

Voronoi cells as iterative convex clipping

Half-space clipping

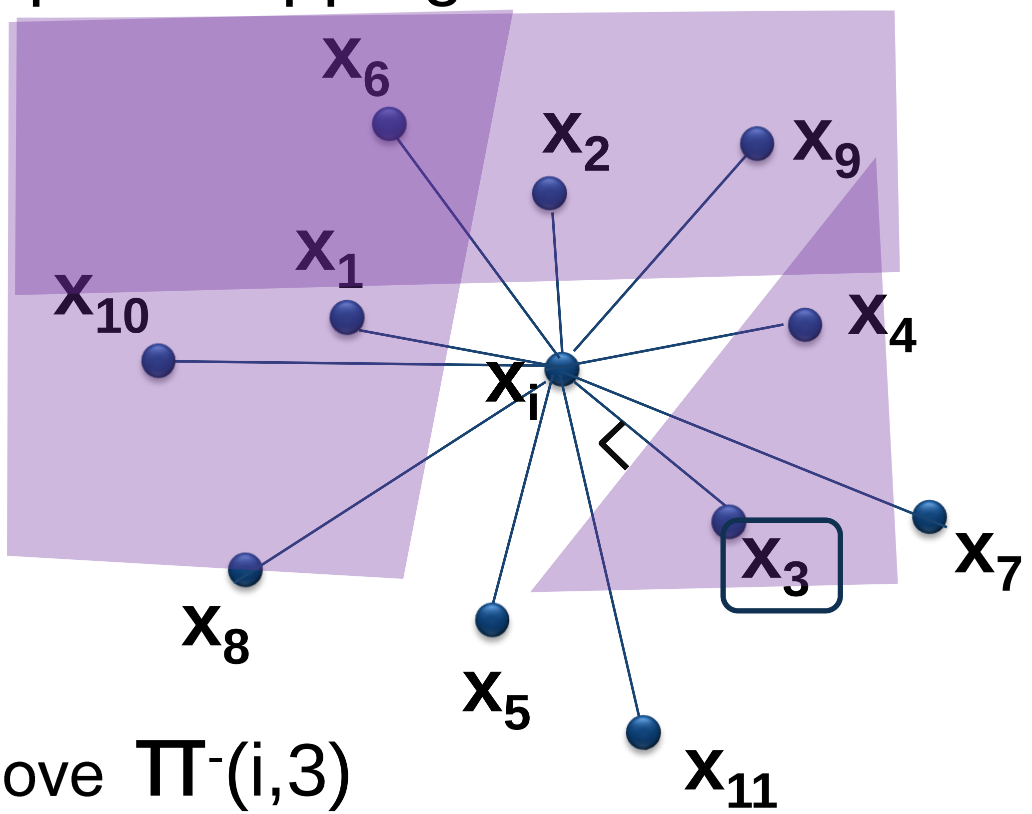


Then remove $\Pi^-(i,2)$

Part. 5 The algorithm

Voronoi cells as iterative convex clipping

Half-space clipping

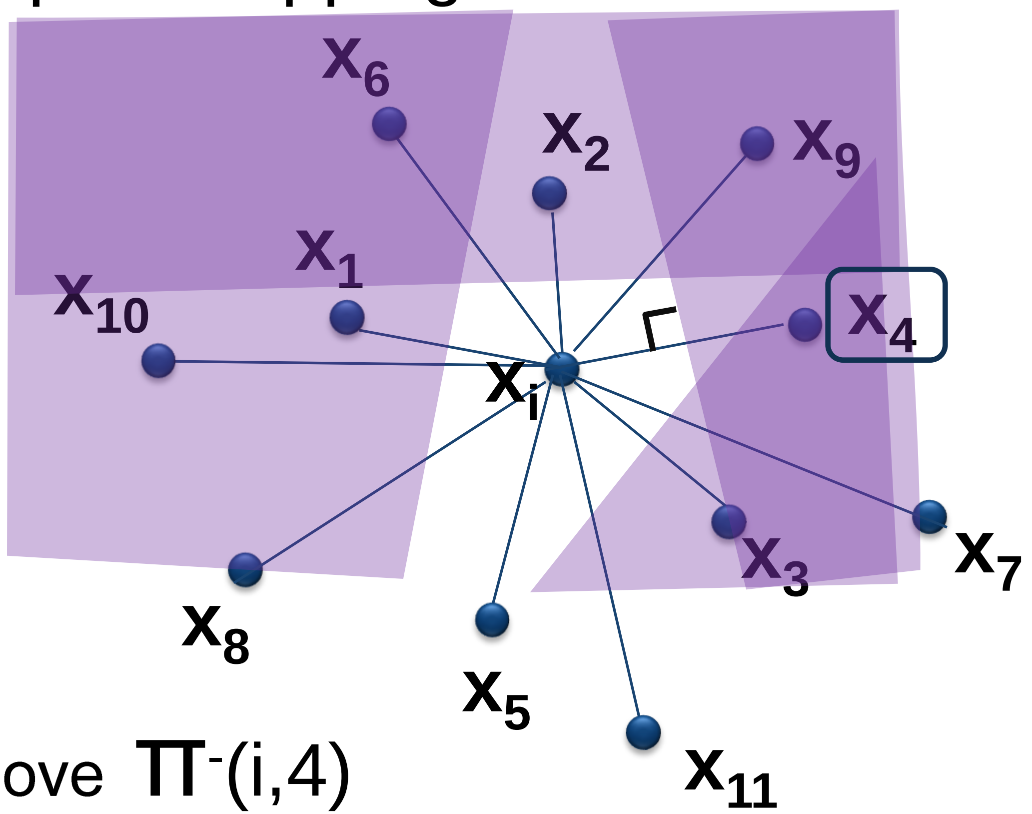


… then remove $\Pi^-(i,3)$

Part. 5   The algorithm

Voronoi cells as iterative convex clipping

Half-space clipping

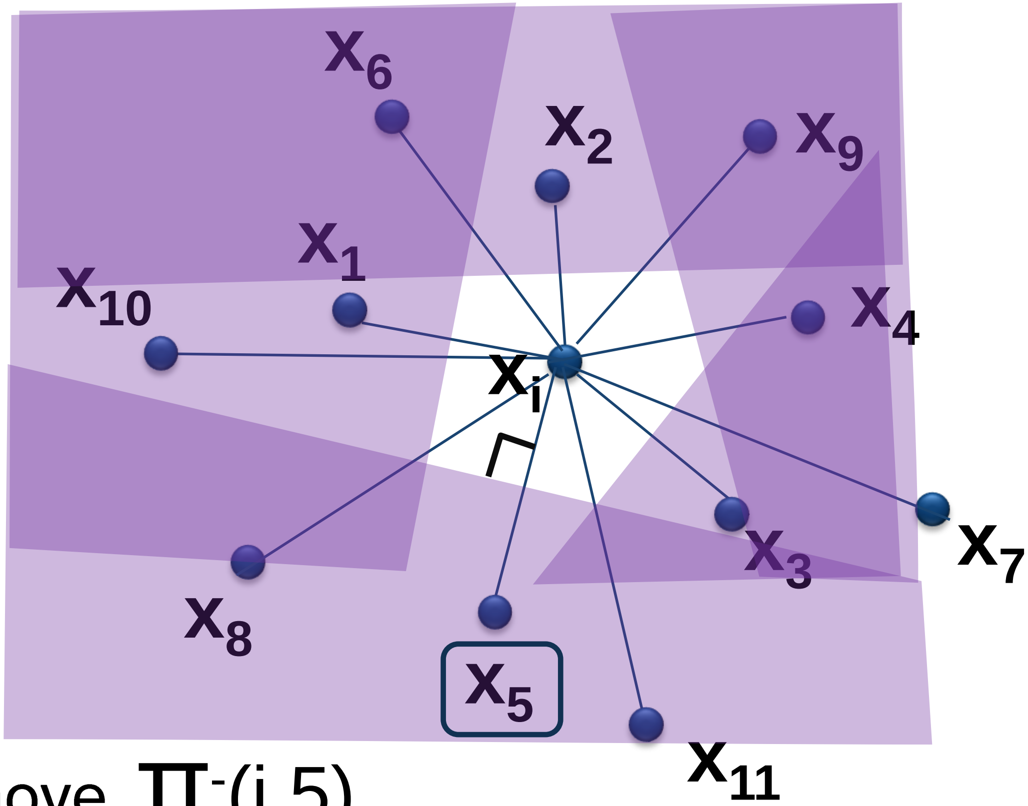


… then remove $\Pi^-(i,4)$

Part. 5   The algorithm

Voronoi cells as iterative convex clipping

Half-space clipping

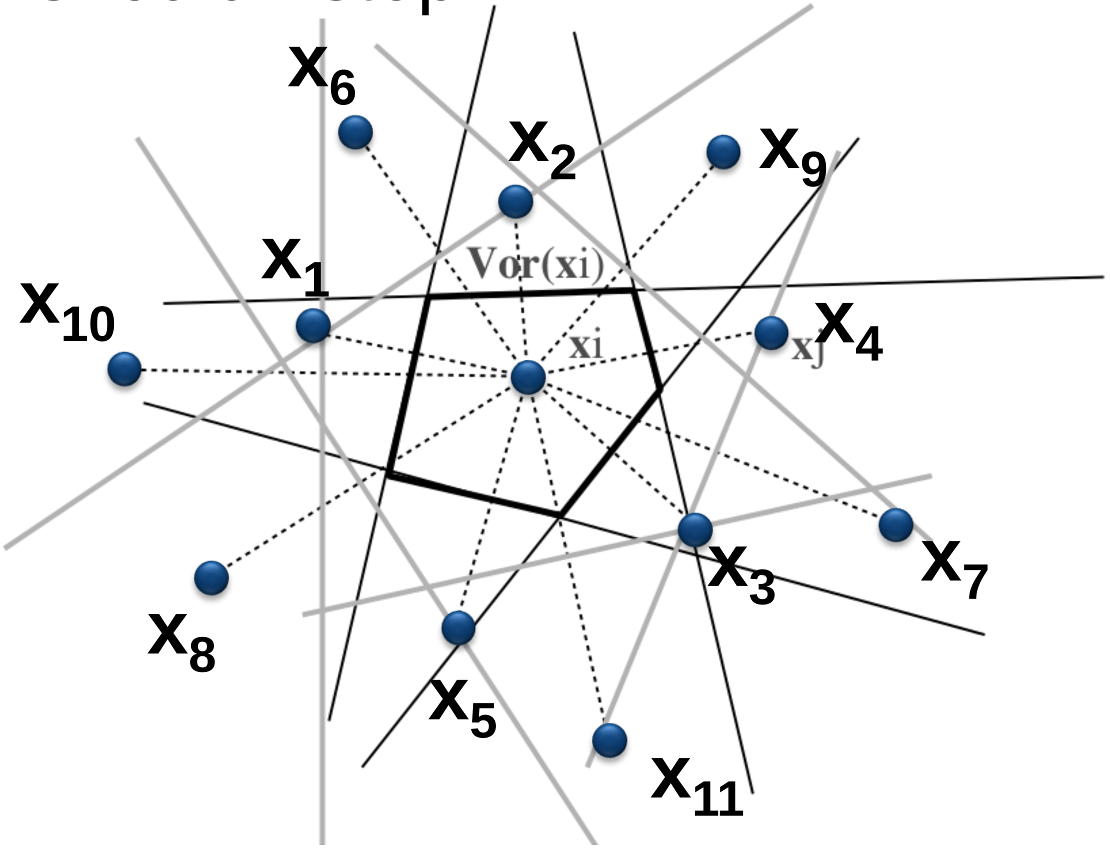


… then remove $\Pi^-(i,5)$

Part. 5 The algorithm

Voronoi cells as iterative convex clipping

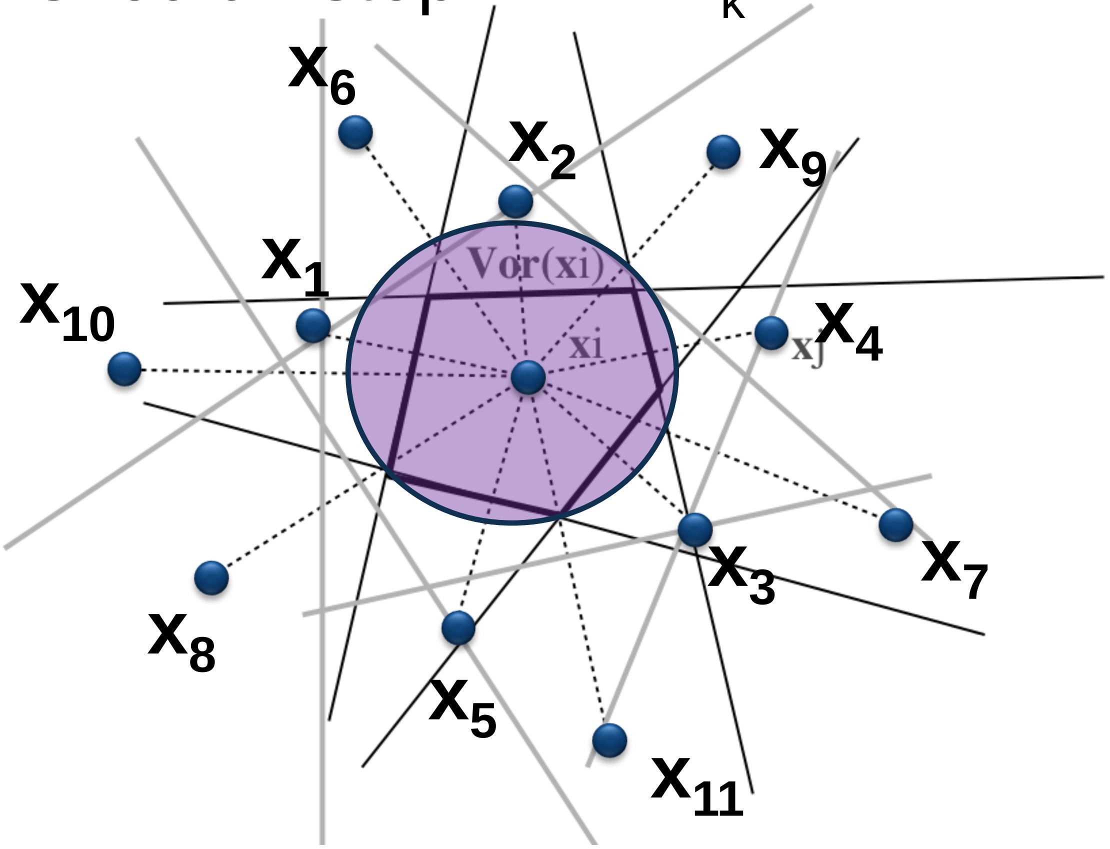
When should I stop ?



Part. 5 The algorithm

Voronoi cells as iterative convex clipping

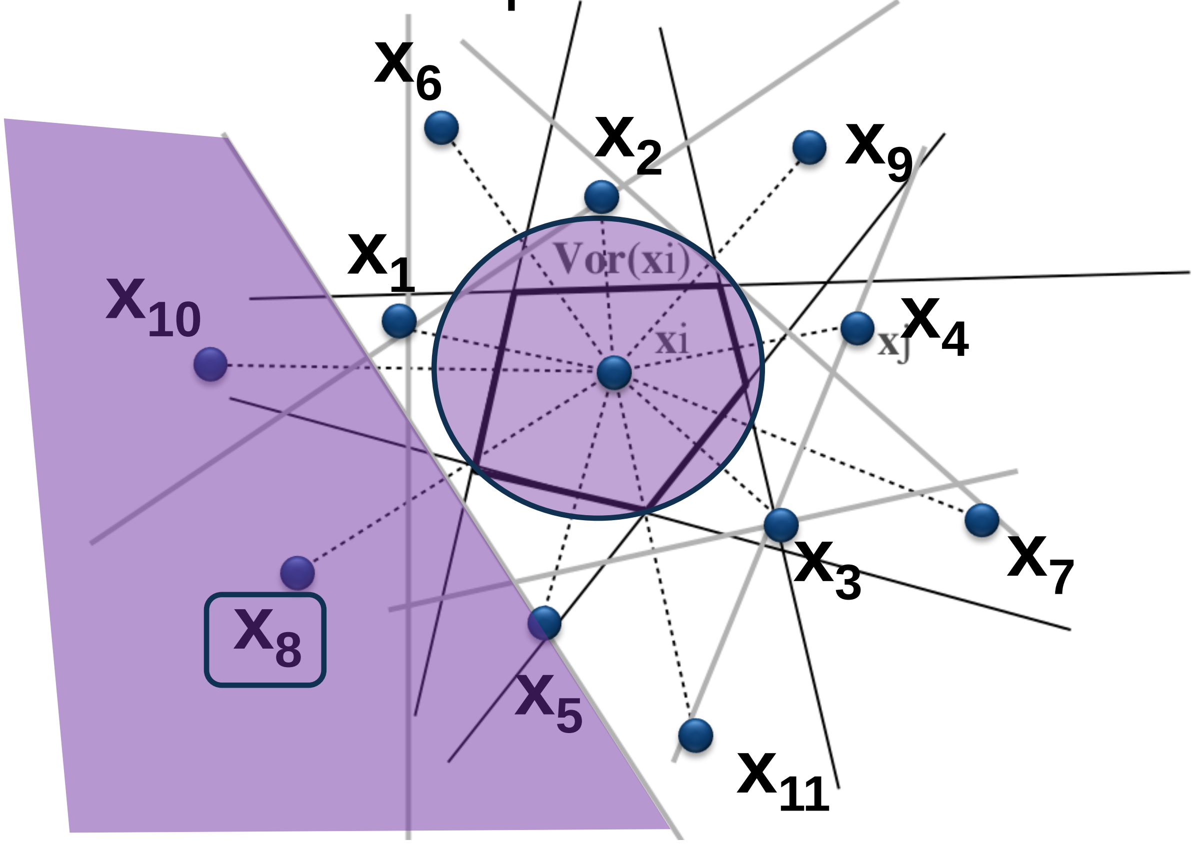
When should I stop ? R_k



Part. 5 The algorithm

Voronoi cells as iterative convex clipping

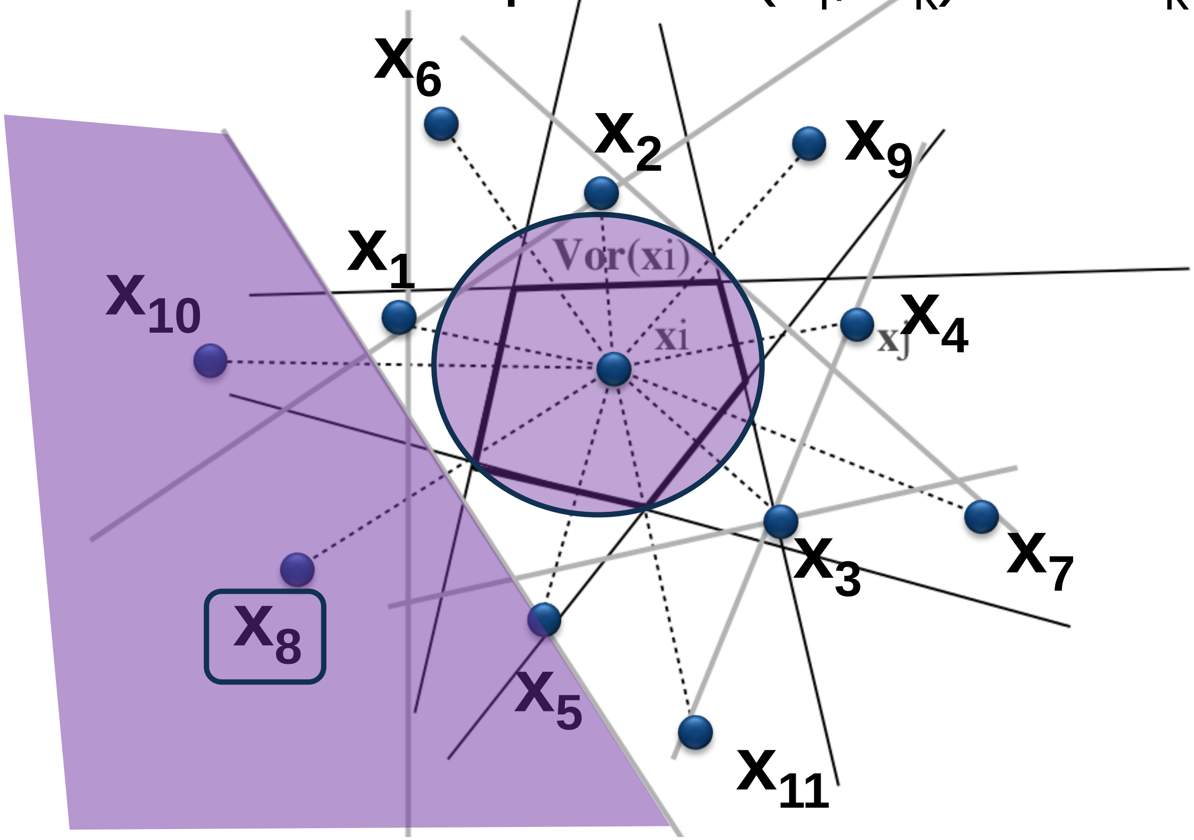
When should I stop ?



Part. 5 The algorithm

Voronoi cells as iterative convex clipping

When should I stop ? $d(\mathbf{x}_i, \mathbf{x}_k) > 2\ R_k$



Part. 5 The algorithm

Voronoi cells as iterative convex clipping

Theorem: $d(\mathbf{x}_i, \mathbf{x}_{k+1}) > 2R_k \rightarrow \bigcap \Pi^+(i,k) = \mathrm{Vor}(\mathbf{x}_i)$

[**L** and Bonneel, IMR 2012]

Part. 5 The algorithm

Voronoi cells as iterative convex clipping

When should I stop ? $d(\mathbf{x}_i, \mathbf{x}_k) > 2\,Rk$

"Radius of security" is reached

Part. 5 The algorithm

Voronoi cells as iterative convex clipping

When should I stop ? $d(\mathbf{x}_i, \mathbf{x}_k) > 2$ Rk

"Radius of security" is reached

Note: R_k decreases and $d(\mathbf{x}_i, \mathbf{x}_k)$ increases

Part. 5 The algorithm

Voronoi cells as iterative convex clipping

When should I stop ? $d(\mathbf{x}_i, \mathbf{x}_k) > 2$ Rk

"Radius of security" is reached

Note: R_k decreases and $d(\mathbf{x}_i, \mathbf{x}_k)$ increases

Advantages:

Part. 5 The algorithm

Voronoi cells as iterative convex clipping

When should I stop ? $d(\mathbf{x}_i, \mathbf{x}_k) > 2$ Rk

"Radius of security" is reached

Note: R_k decreases and $d(\mathbf{x}_i, \mathbf{x}_k)$ increases

Advantages:

(1) Compute Vor($\mathbf{X}$) ∩ S directly (start with f and clip)

Part. 5 The algorithm

Voronoi cells as iterative convex clipping

When should I stop ? $d(\mathbf{x}_i, \mathbf{x}_k) > 2\ Rk$

"Radius of security" is reached

Note: R_k decreases and $d(\mathbf{x}_i, \mathbf{x}_k)$ increases

Advantages:

(1) Compute $\mathrm{Vor}(\mathbf{X}) \cap S$ directly (start with f and clip)

(2) Replace Delaunay with ANN ! (no d! factor)

Part. 5 The algorithm

Voronoi cells as iterative convex clipping

When should I stop ? $d(\mathbf{x}_i, \mathbf{x}_k) > 2$ Rk

"Radius of security" is reached

Note: R_k decreases and $d(\mathbf{x}_i, \mathbf{x}_k)$ increases

Advantages:

(1) Compute Vor($\mathbf{X}$) ∩ S directly (start with f and clip)

(2) Replace Delaunay with ANN ! (no d! factor)

(3) Parallelization is trivial (partition S and // in parts)

6

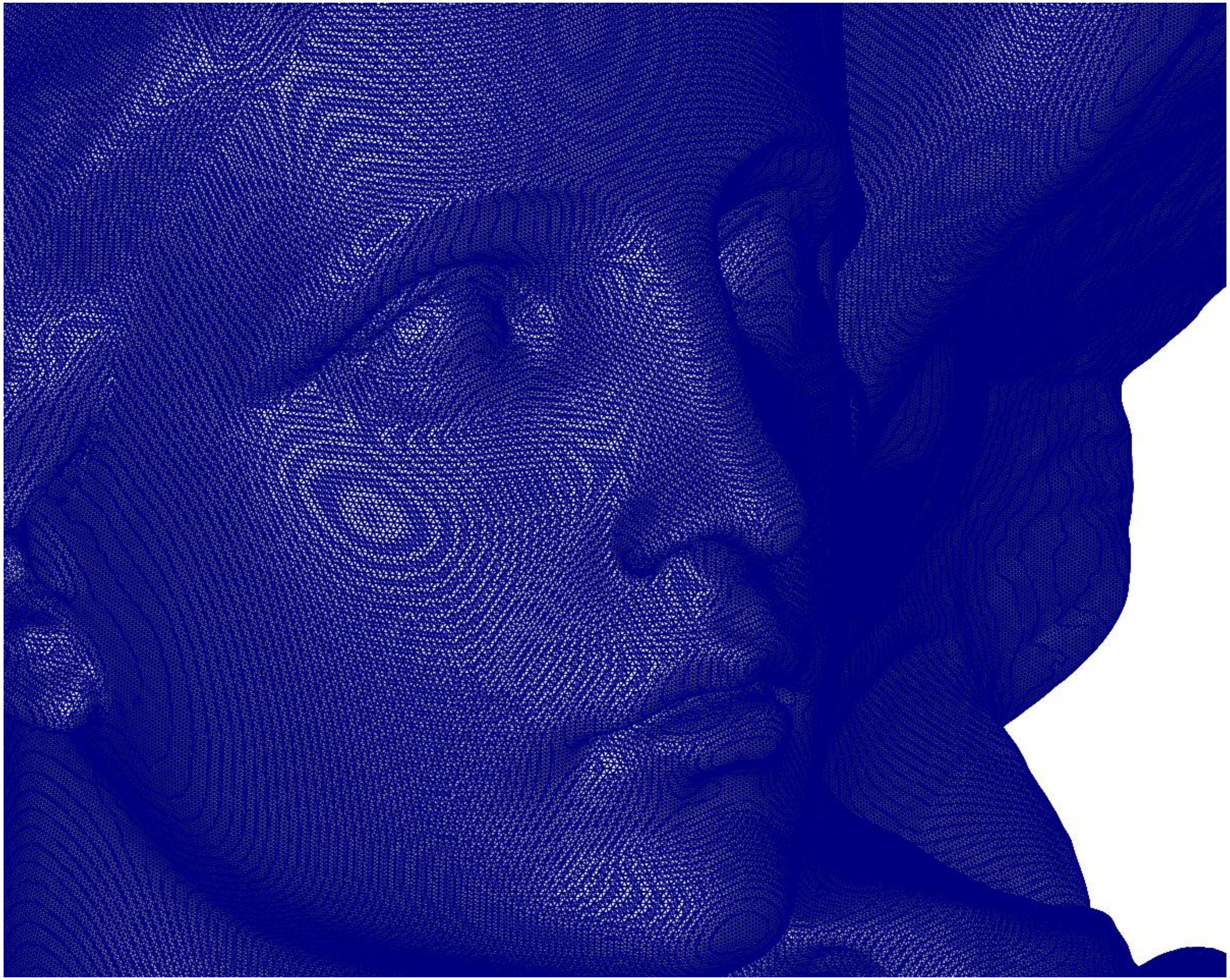
Some results

Part. 6 Some results – Filigree (Aim@Shape)

Part. 6 Some results – AntEater (Konstanz U.)

Part. 6 Some results

Lucy (Stanford): 28 million triangles

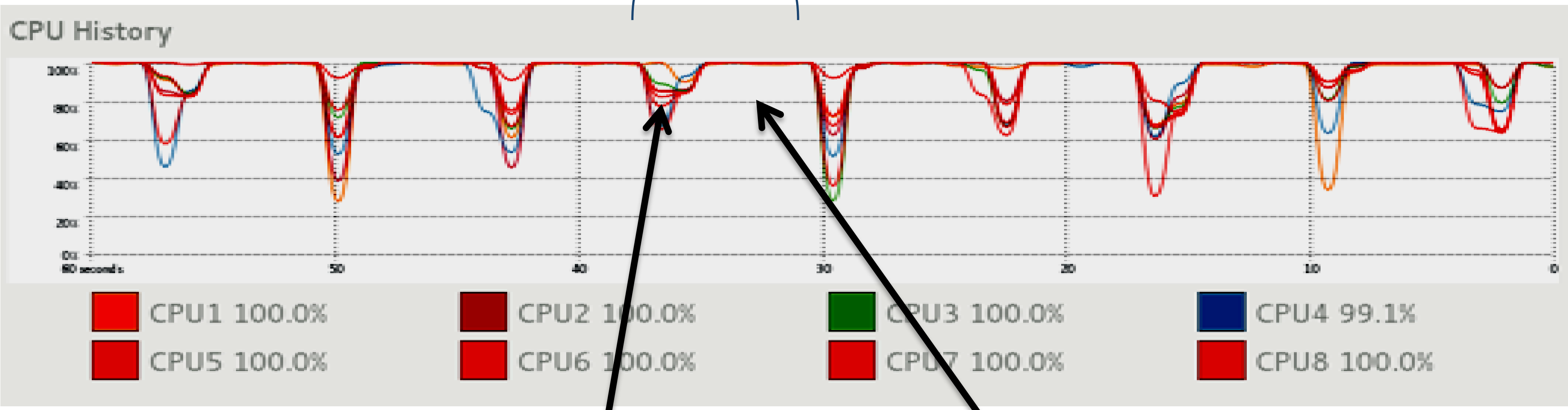


Parallel implementation, 8 threads, core i7 : 5min

Part. 6 Some results

Lucy (Stanford): 28 million triangles

One iteration



ANN construction (sequential)

Voronoi Parallel Linear Enumeration

Parallel implementation, 8 threads, core i7 : 5min

Part. 6 Some results

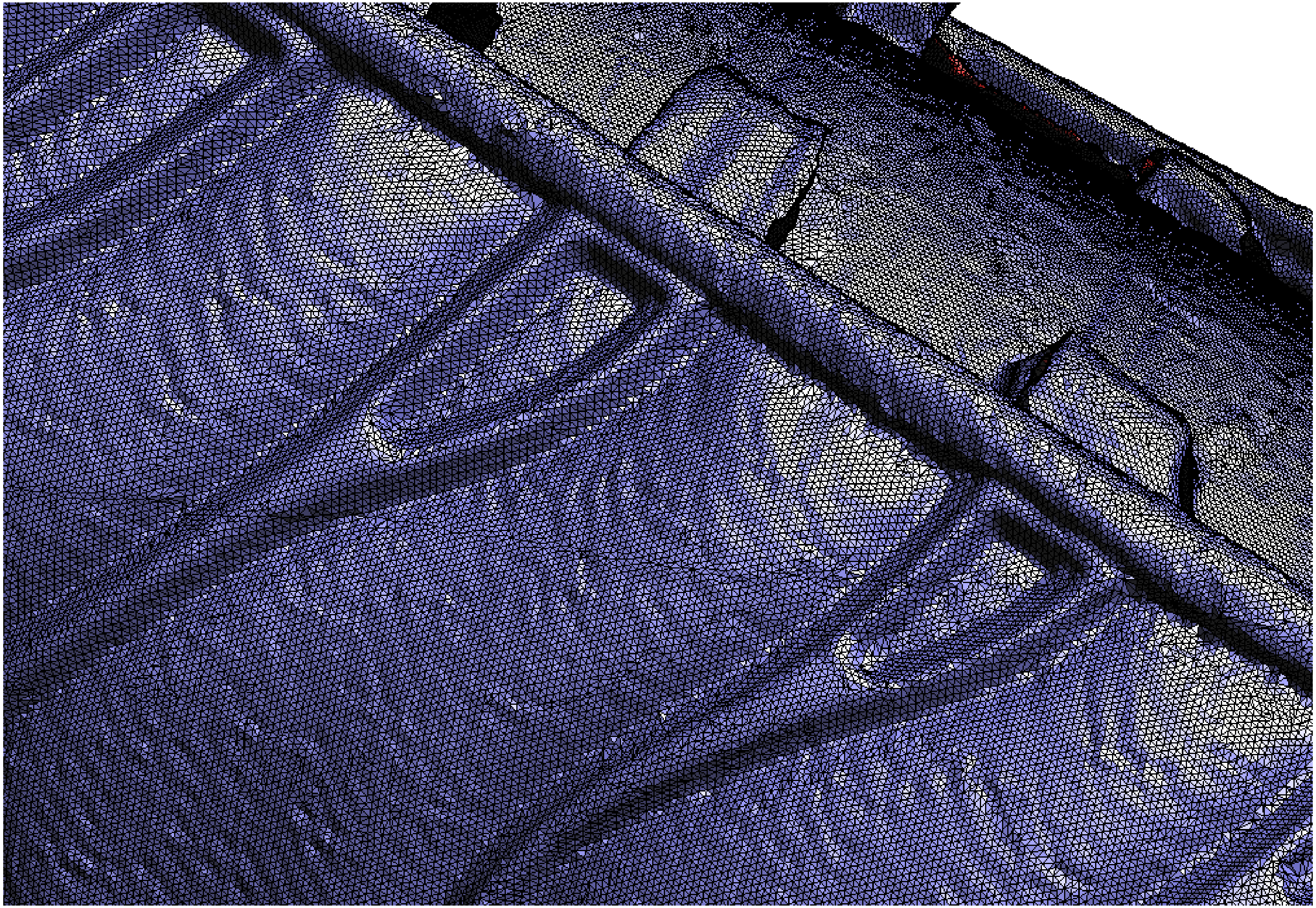
Vorpaline mesh, 100K vertices

Part. 6 Some results – Porshe (Distene)

Part. 6 Some results – Plane (Distene)

Part. 6 Some results

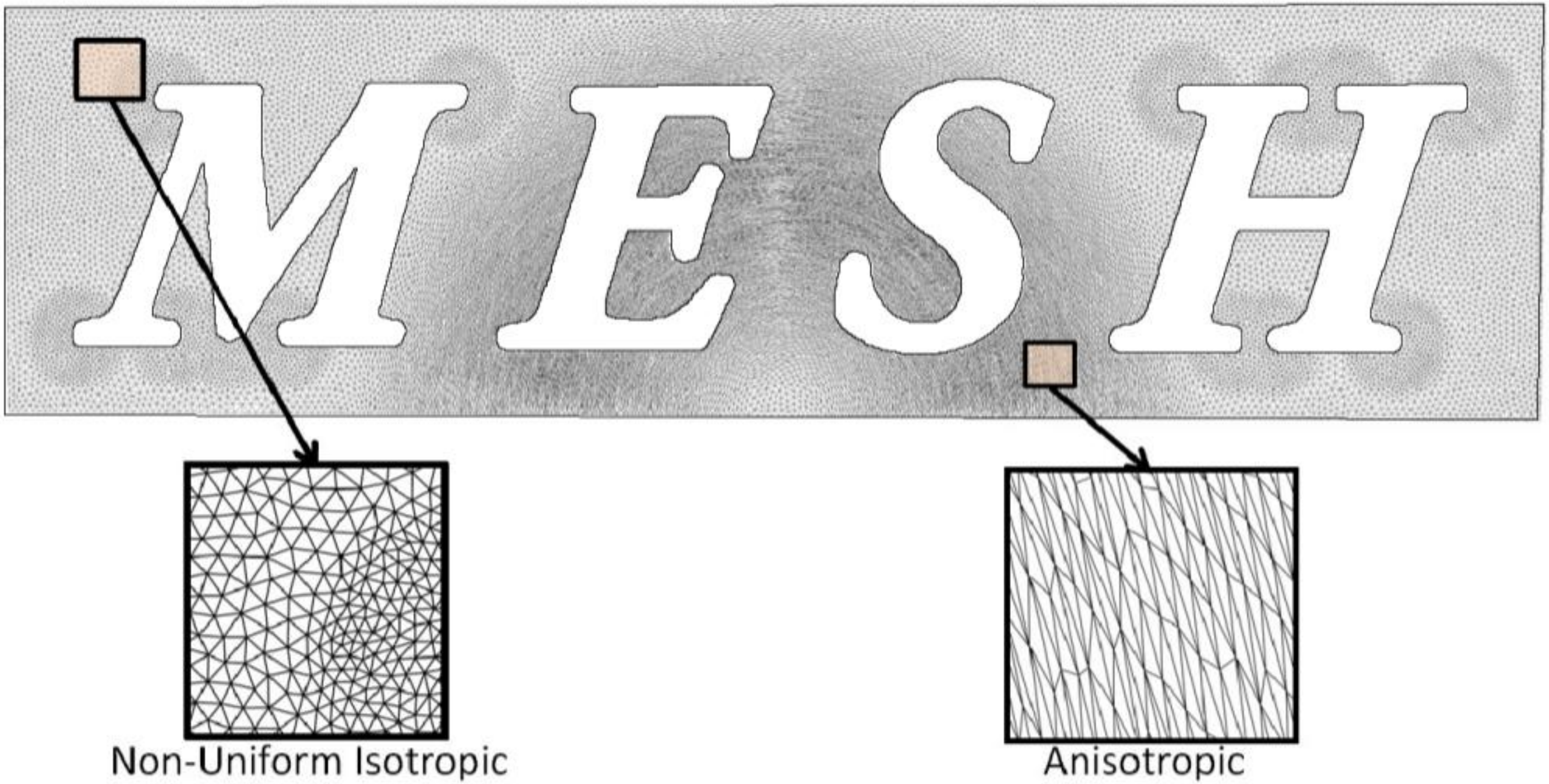
Part. 6 Some results



Part. 6 Some results

Part. 6 What about arbitrary anisotropy ?

Particle-based Anisotropic Surface Meshing
[Zhong, Guo, Wang, **L**, Sun, Liu and Mao, ACM SIGGRAPH 2013]



Part. 6 Reconstruction – Co3Ne (Concurrent Co Cone)

Thank you for your attention

Acknowledgements

Wenping Wang (Hong-Kong U.)

Nicolas Bonneel (Harvard U.)

Funding

GOODSHAPE ERC StG 205693

VORPALINE ERC-PoC 334829 *pre-industrialization*

ANR MORPHO, ANR BECASIM