



HAL
open science

Triangulations, CGAL, and hyperbolic surfaces

Monique Teillaud

► **To cite this version:**

Monique Teillaud. Triangulations, CGAL, and hyperbolic surfaces. Renormalization, computation and visualization in Geometry, Number Theory and Dynamics, Sep 2023, Marseille (CIRM, Centre International de Rencontres Mathématiques), France. hal-04241550

HAL Id: hal-04241550

<https://inria.hal.science/hal-04241550>

Submitted on 5 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Triangulations, CGAL, and hyperbolic surfaces



Monique Teillaud

Thanks to the organizers!

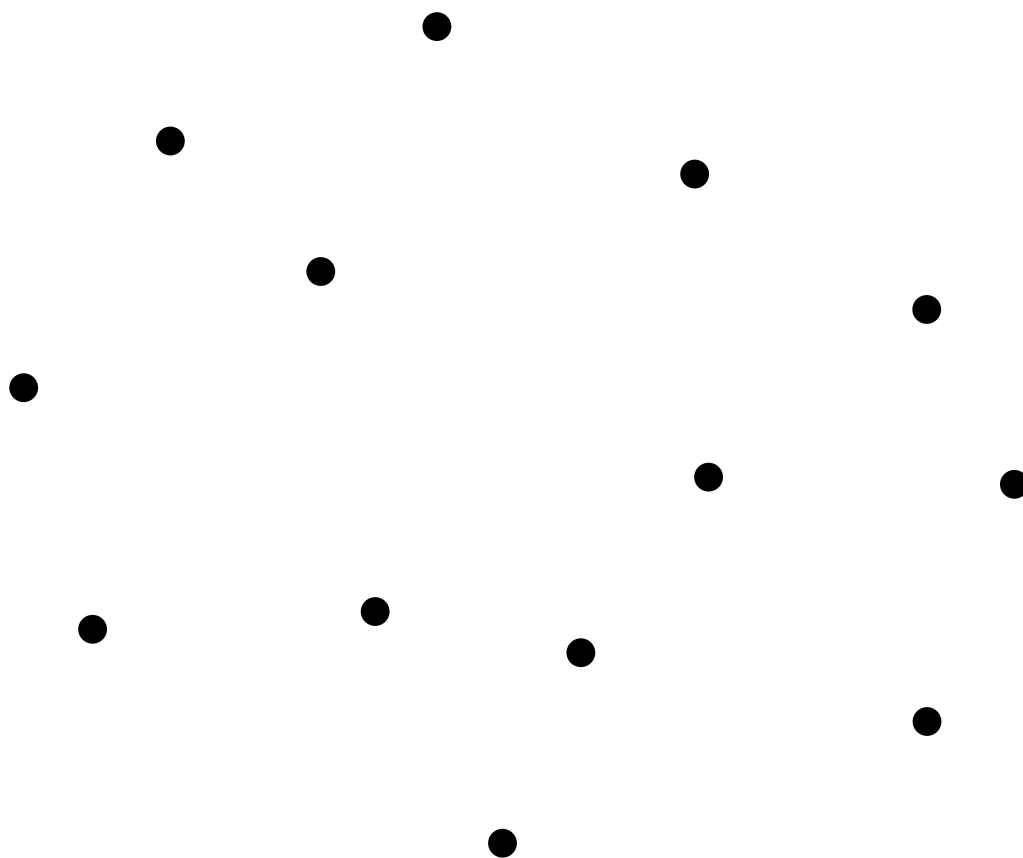
Thanks to the organizers!

I'm leaving on Wednesday afternoon ☹️



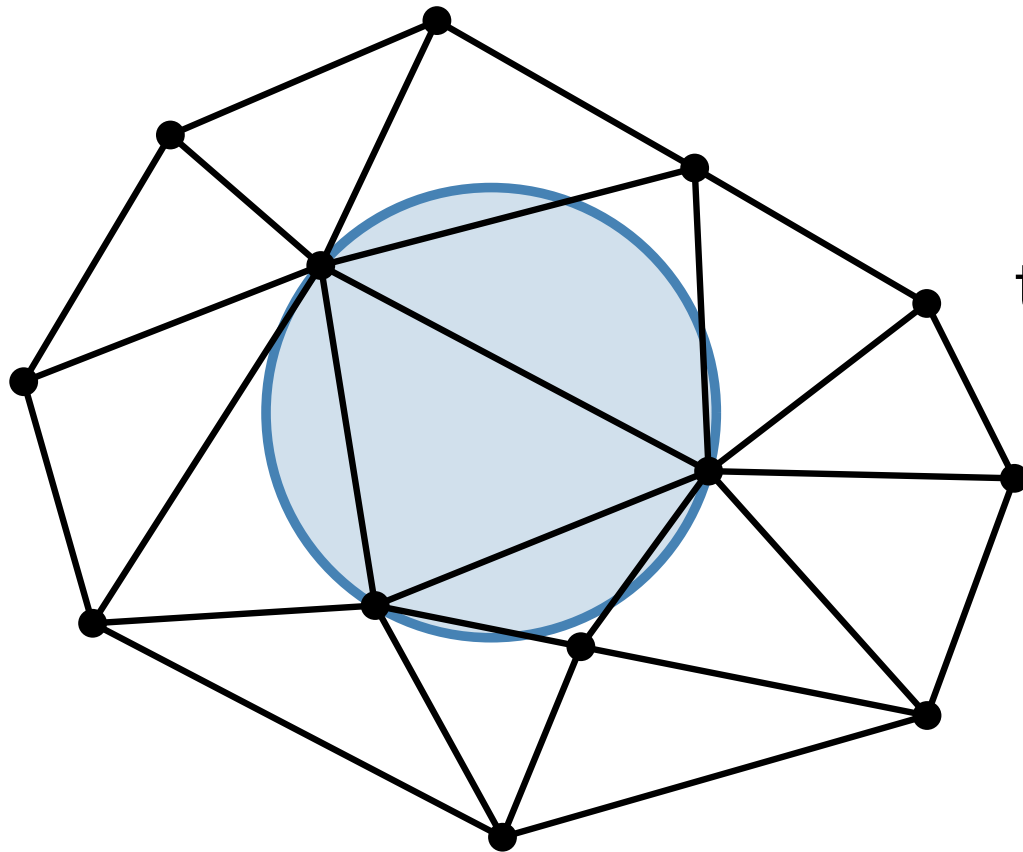
Introduction

Delaunay triangulation in \mathbb{R}^d



set of points \mathcal{P}

Delaunay triangulation in \mathbb{R}^d



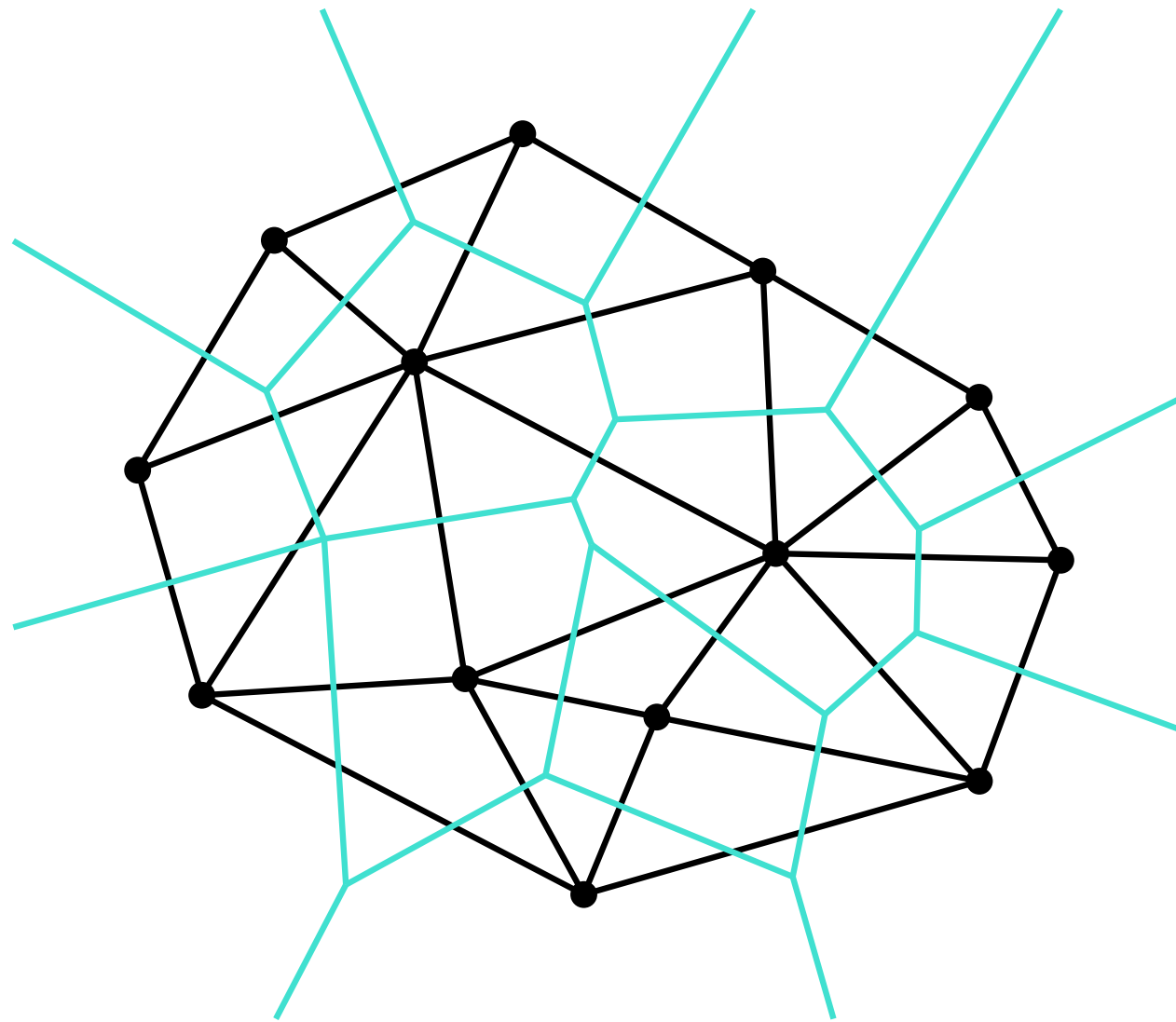
set of points \mathcal{P}

triangulation of $CH(\mathcal{P})$

s.t.

circumscribed balls
are empty

Delaunay triangulation in \mathbb{R}^d

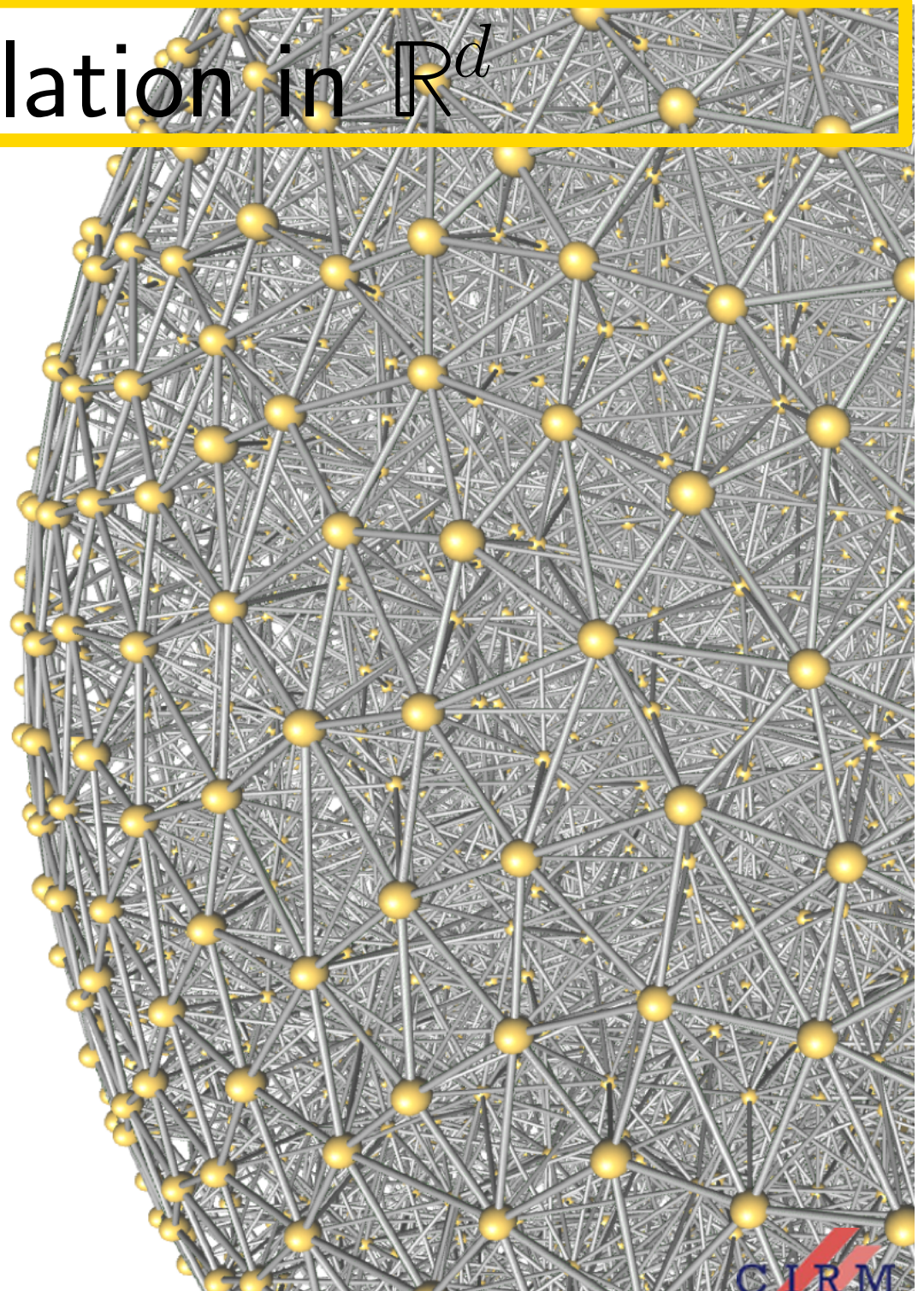


set of points \mathcal{P}

dual
Voronoi diagram

Delaunay triangulation in \mathbb{R}^d

Many applications





www.cgal.org

The Computational Geometry Algorithms Library

Open source project

LGPLv3+ / Commercial



www.cgal.org

The Computational Geometry Algorithms Library

Open source project

LGPLv3+ / Commercial

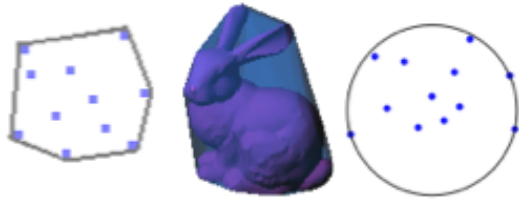
C++

Genericity/flexibility through templates

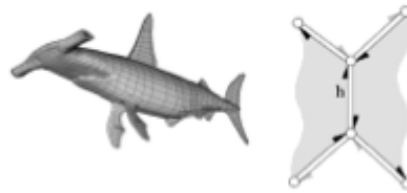
Robustness through Exact Geometric Computation [Yap]

Exact *predicates* \rightsquigarrow exact decisions

> 100 chapters in the manual



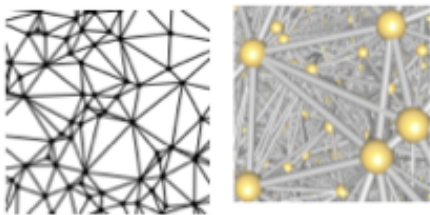
Bounding Volumes



Polyhedral Surface



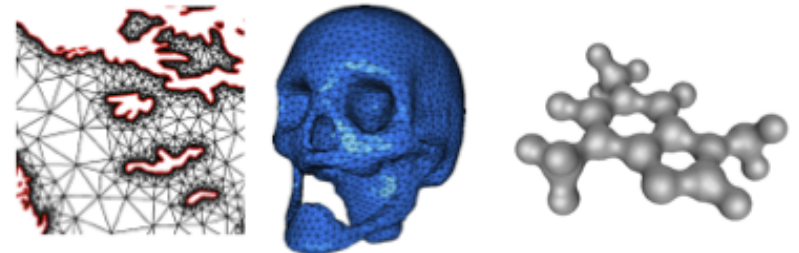
BooleanOperations



Triangulations



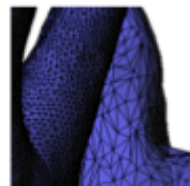
Voronoi Diagrams



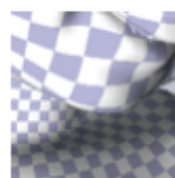
Mesh Generation



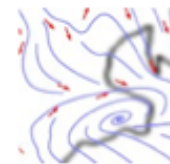
Subdivision



Simplification



Parameterization



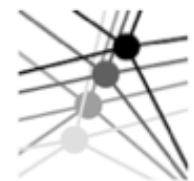
Streamlines



Ridge
Detection



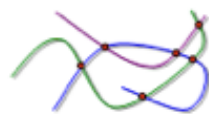
Neighbour
Search



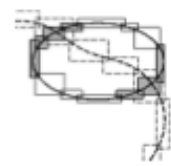
Kinetic
Data structures



Lower Envelope



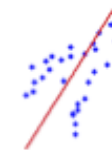
Arrangement



Intersection
Detection



Minkowski
Sum



PCA

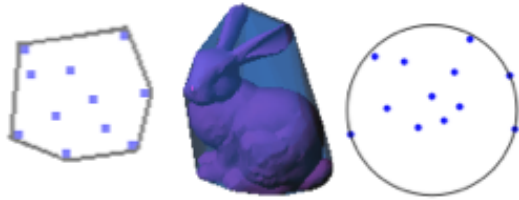


Polytope
distance

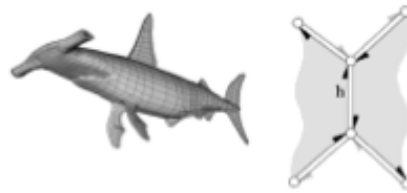


QP Solver

> 100 chapters in the manual



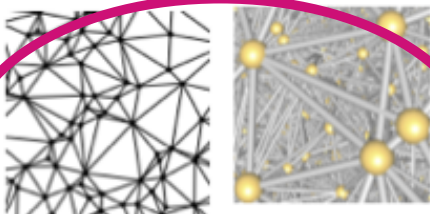
Bounding Volumes



Polyhedral Surface



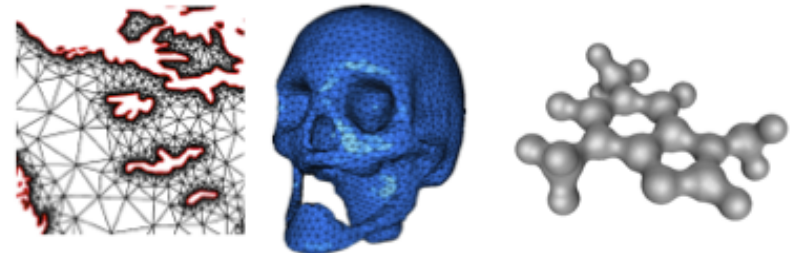
BooleanOperations



Triangulations



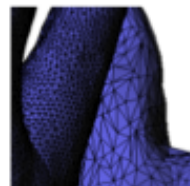
Voronoi Diagrams



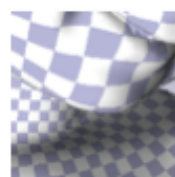
Mesh Generation



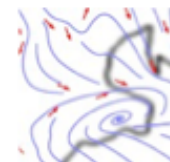
Subdivision



Simplification



Parameterization



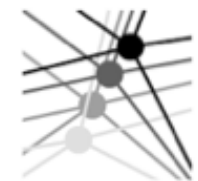
Streamlines



Ridge
Detection



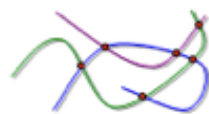
Neighbour
Search



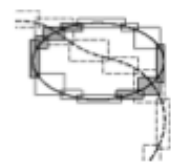
Kinetic
Data structures



Lower Envelope



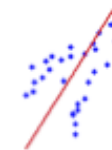
Arrangement



Intersection
Detection



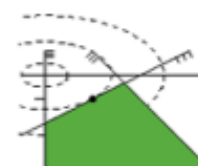
Minkowski
Sum



PCA



Polytope
distance



QP Solver

2D Triangulations (1997)

3D Triangulations (2000) and meshes (2009)

dD Triangulations (2015)

2D Periodic Triangulations (2013)

Flat torus

square/cubic domain
(general in progress)

3D Periodic Triangulations (2009) and meshes (2018)

2D Hyperbolic Triangulations (2019)

Bolza surface

2D Periodic Hyperbolic Triangulations (2019)

Triangulations on the 2D sphere (2021)

install CGAL dependencies + Qt (example MacOS)

```
brew install cgal
```

```
brew install qt5
```

download CGAL sources with examples and demos:

```
https://github.com/CGAL/cgal/releases/  
CGAL-5.6.zip
```

set up environment variables

```
CGAL_DIR=/Users/teillaud/CGAL-5.6
```

```
Qt5_DIR=/opt/homebrew/Cellar/qt@5/5.15.10_1/lib/cmake
```

compile demos

```
cd CGAL_DIR/demo/Your_favorite_package
```

```
mkdir build; cd build
```

```
cmake ..
```

```
make
```

run the demo(s)



www.cgal.org

general design

Triangulation_d < Geom_traits, TDS >

$d = 2, 3$

Triangulation_d < **Geom_traits**, TDS >

$d = 2, 3$

geometry

objects (points, etc)

predicates (e.g., in_circle)

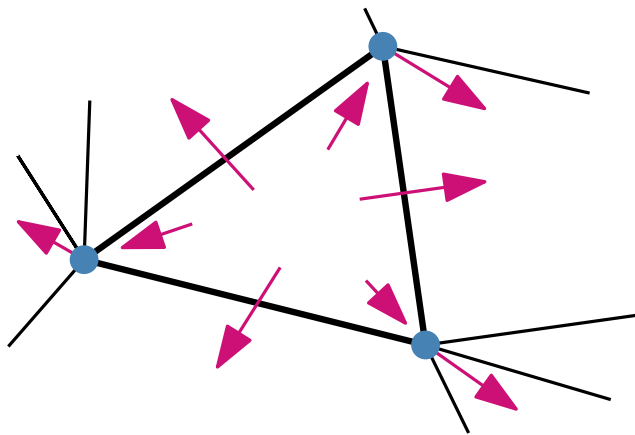
constructions (e.g., circumcenter)

Triangulation_d < **Geom_traits**, **TDS** >

$d = 2, 3$

geometry

objects (points, etc)
 predicates (e.g., in_circle)
 constructions (e.g., circumcenter)



combinatorics

Cell = d -simplex
 → d vertices
 → d adjacent cells

Vertex
 point
 → one incident cell

[Incremental] Algorithms

Bowyer's fill-hole algorithm

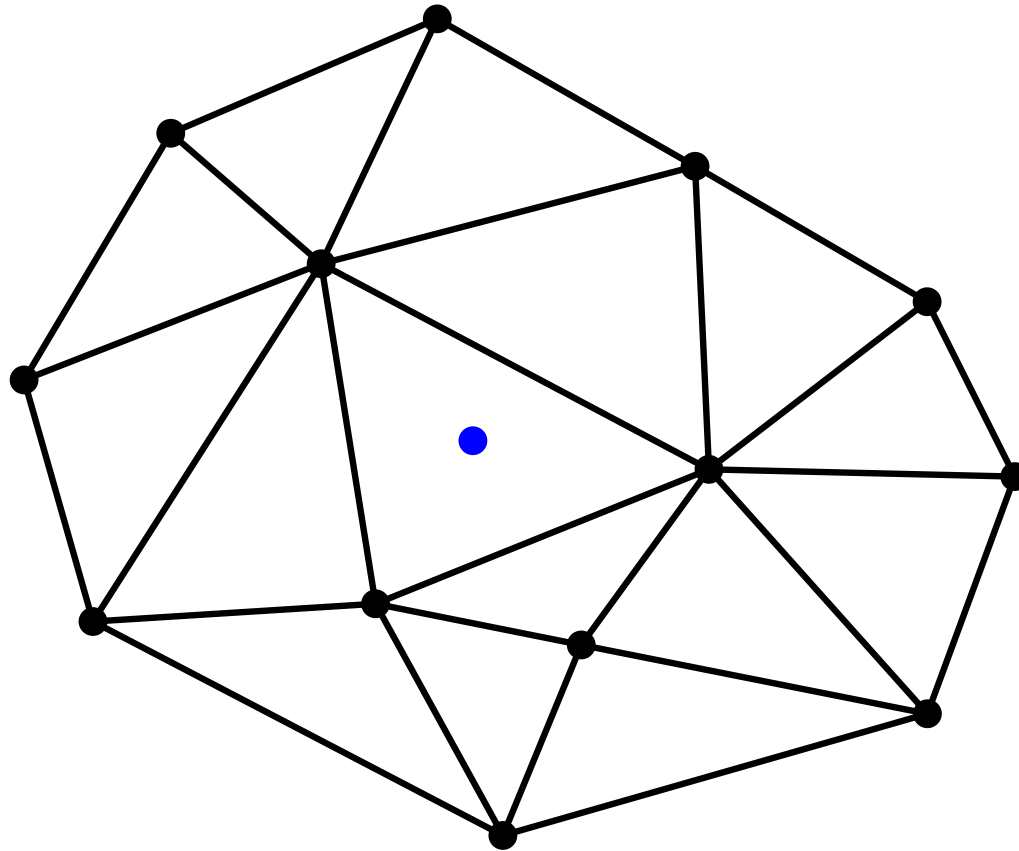
\mathbb{R}^d

[Bowyer 1981] [Watson 1981]

Bowyer's fill-hole algorithm

 \mathbb{R}^d

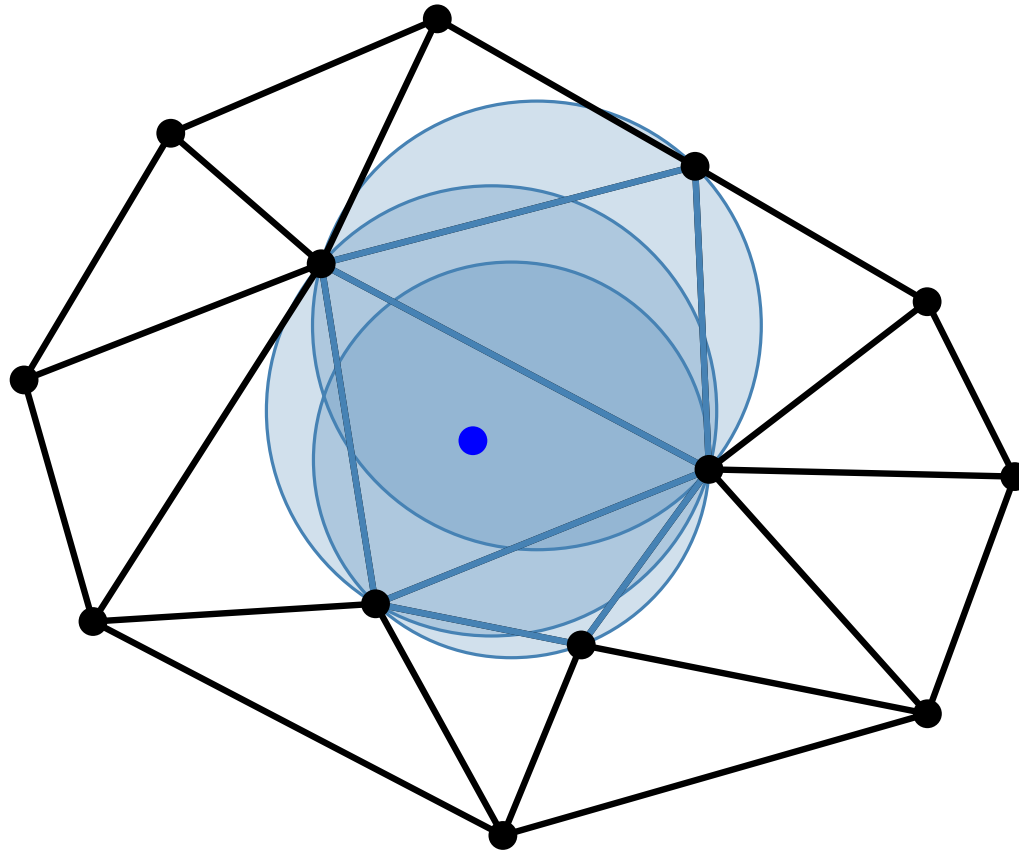
[Bowyer 1981] [Watson 1981]



Bowyer's fill-hole algorithm

 \mathbb{R}^d

[Bowyer 1981] [Watson 1981]

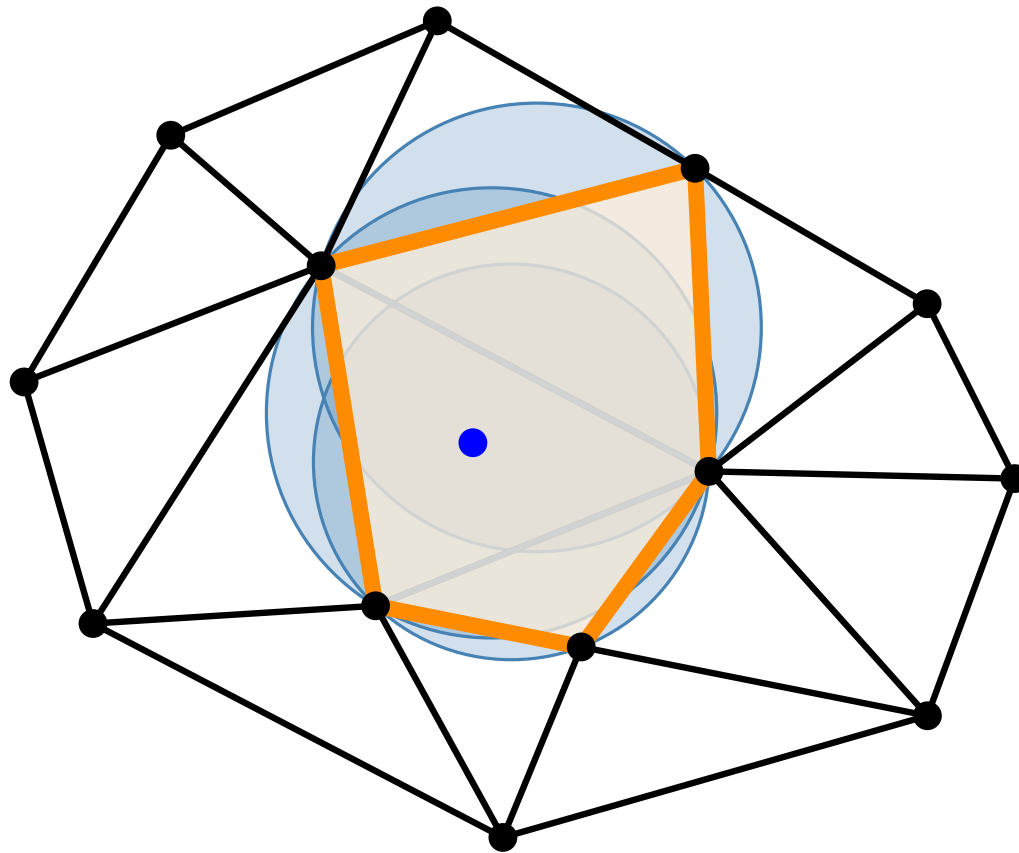


find simplices in conflict

Bowyer's fill-hole algorithm

 \mathbb{R}^d

[Bowyer 1981] [Watson 1981]



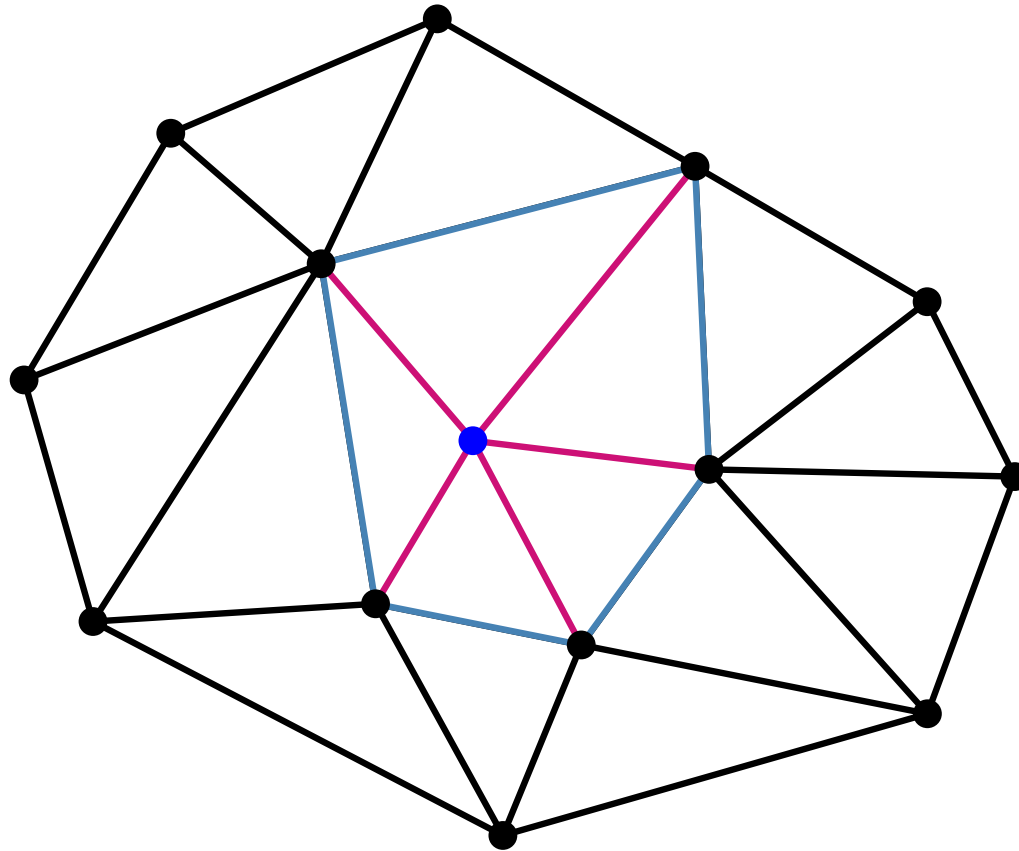
find simplices in conflict

the conflict region forms a topological ball

Bowyer's fill-hole algorithm

\mathbb{R}^d

[Bowyer 1981] [Watson 1981]



create new simplices

Bowyer's fill-hole algorithm

 \mathbb{R}^d The logo for CGAL (Computational Geometry Algorithms Library) is displayed in a stylized, multi-colored font.

Triangulation `< Geom_traits, TDS >`

find simplices in conflict
Triangulation
using Geom_traits

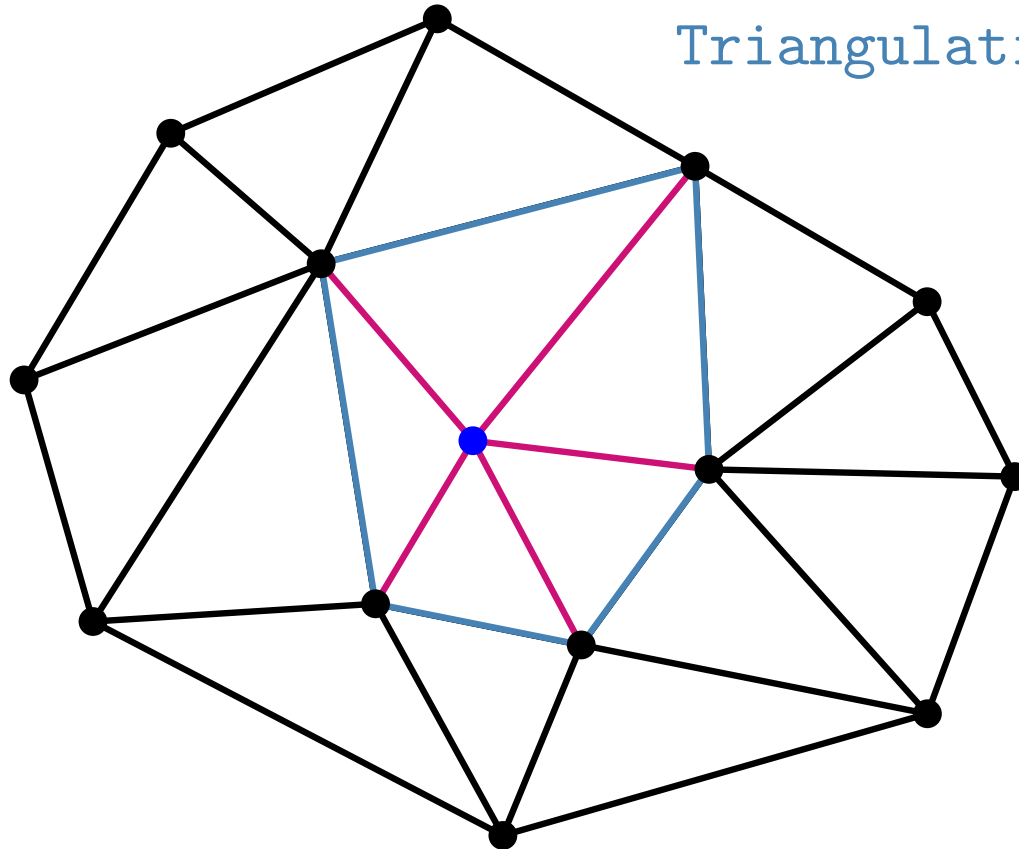
create new simplices
TDS

allows for

- clean
- efficient

implementation

[T. CGAL'00] [Pion, T. CGAL'01*]



Lawson's flip algorithm

\mathbb{R}^d

[Lawson 1977]

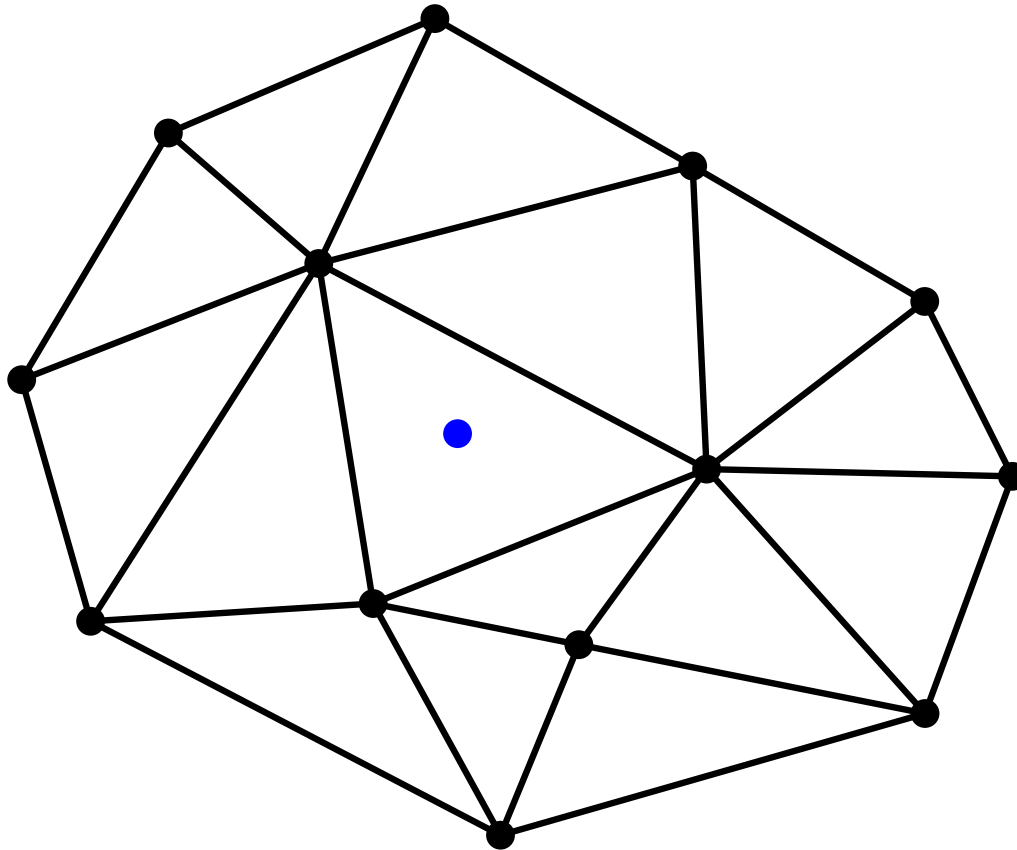
[Joe 1989] [Rajan 1991] [Edelsbrunner Shah 1996]

Lawson's flip algorithm

 \mathbb{R}^d

[Lawson 1977]

[Joe 1989] [Rajan 1991] [Edelsbrunner Shah 1996]

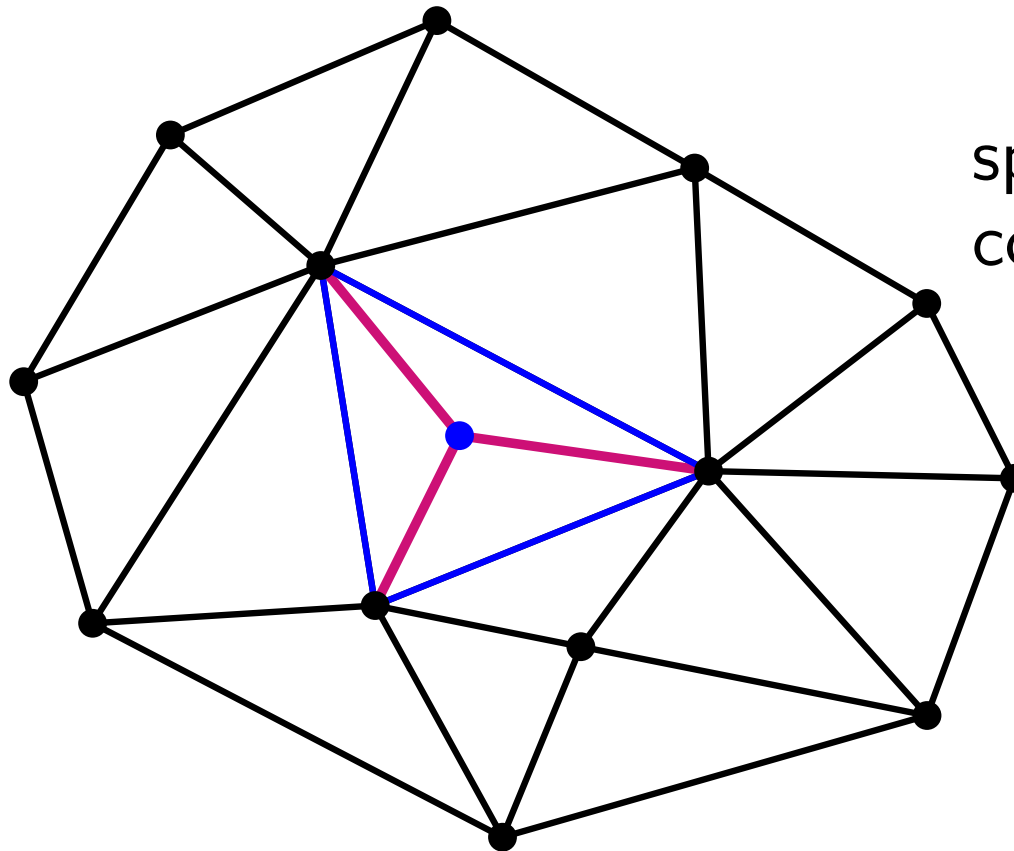


Lawson's flip algorithm

\mathbb{R}^d

[Lawson 1977]

[Joe 1989] [Rajan 1991] [Edelsbrunner Shah 1996]



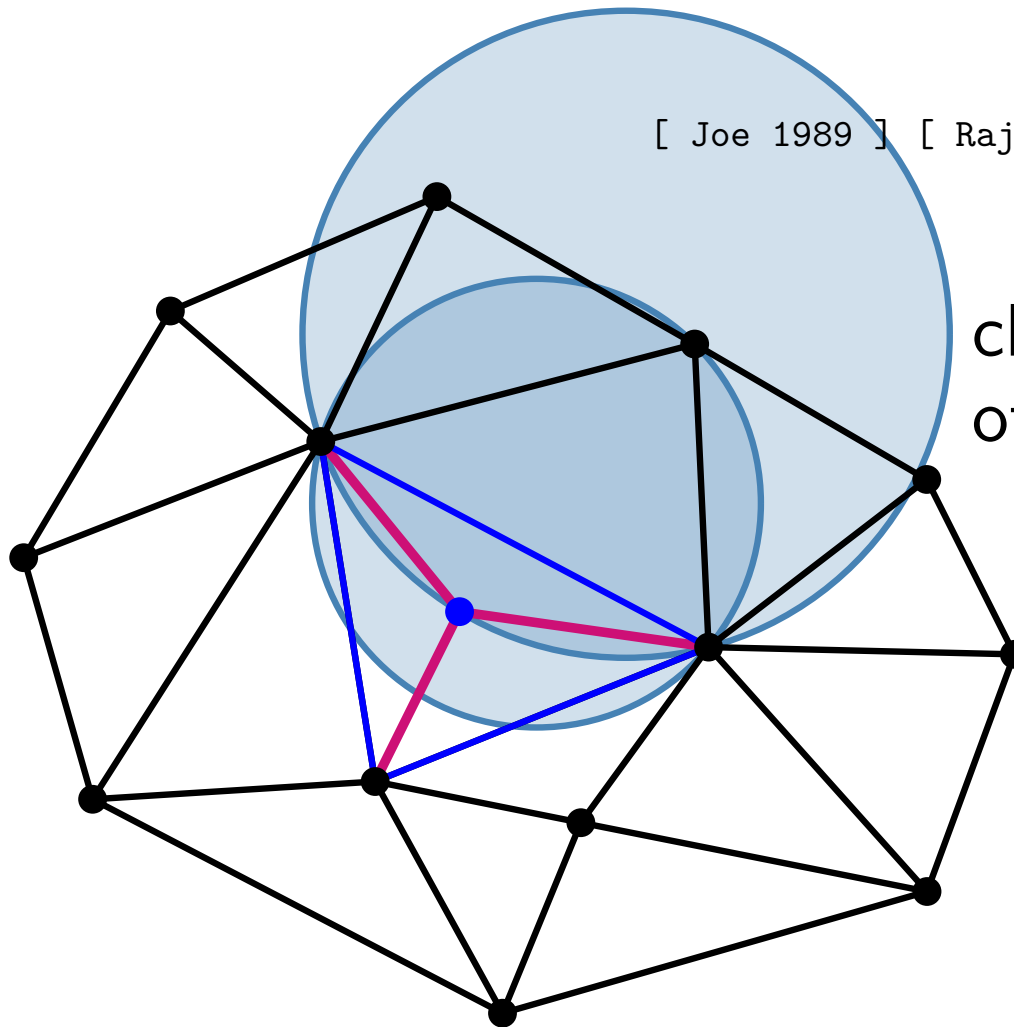
split the triangle
containing the new point

Lawson's flip algorithm

\mathbb{R}^d

[Lawson 1977]

[Joe 1989] [Rajan 1991] [Edelsbrunner Shah 1996]



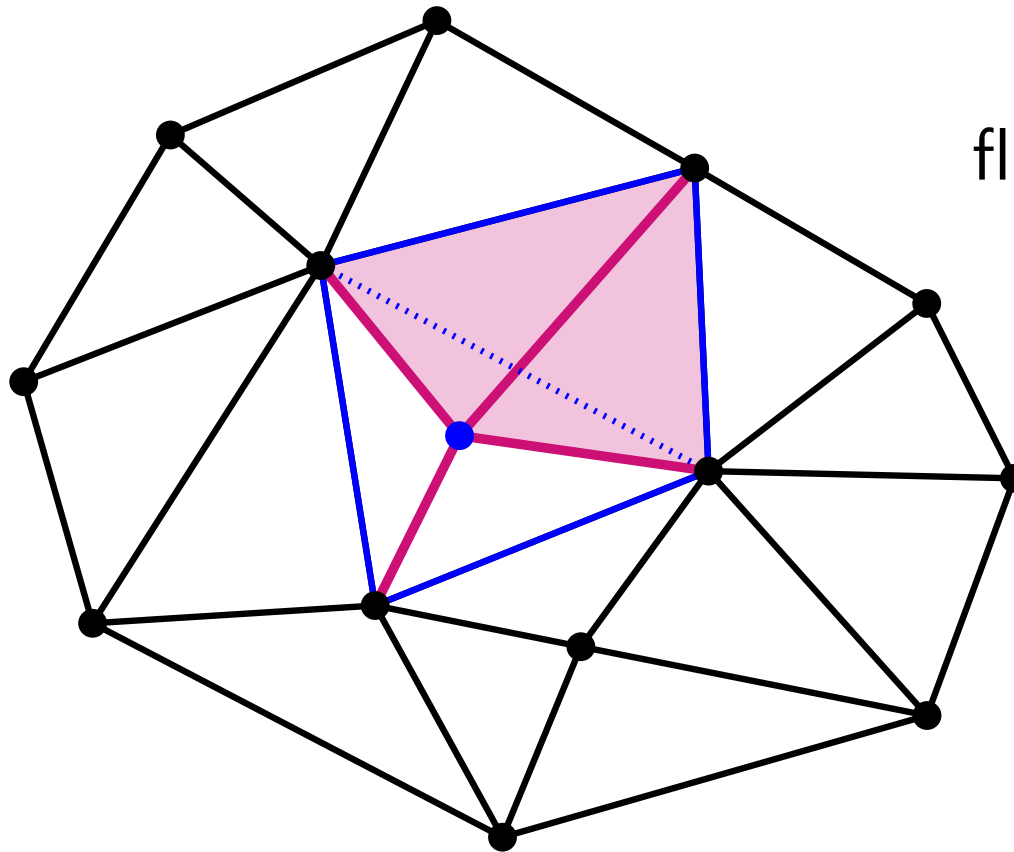
check each edge
of a new triangle

Lawson's flip algorithm

\mathbb{R}^d

[Lawson 1977]

[Joe 1989] [Rajan 1991] [Edelsbrunner Shah 1996]

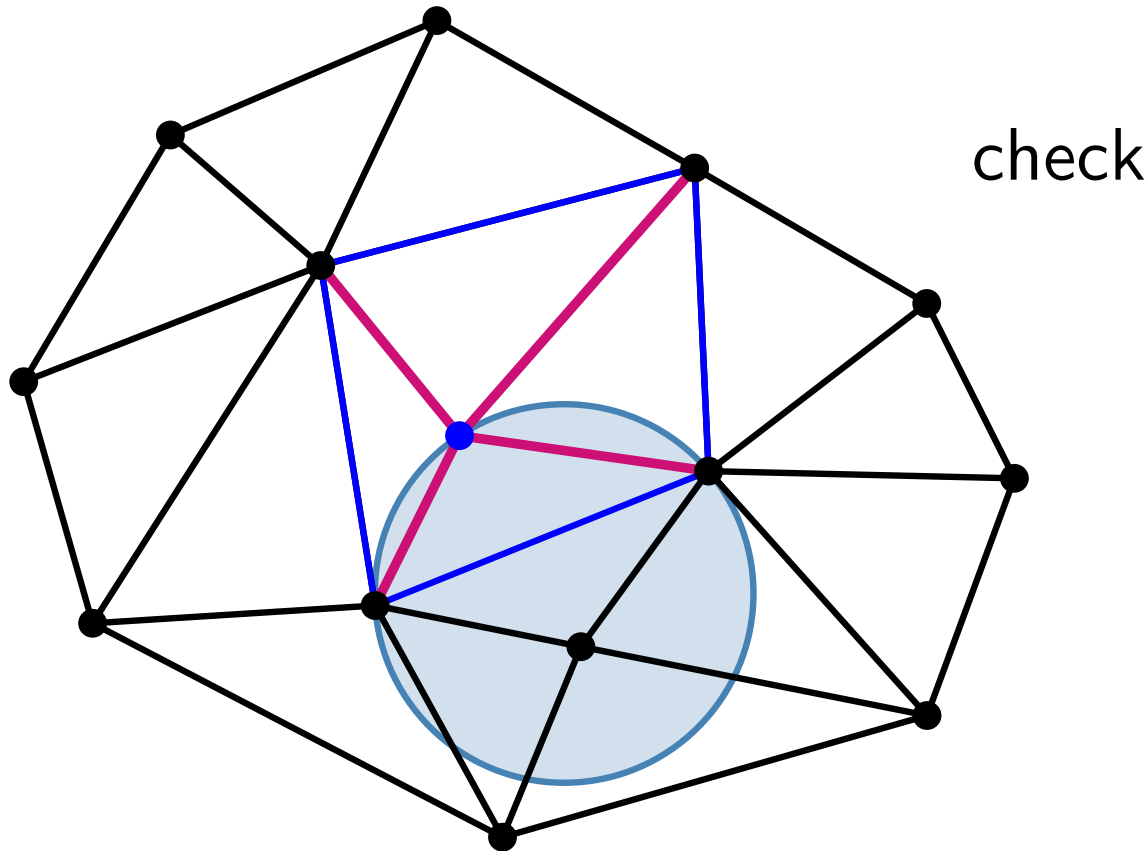


Lawson's flip algorithm

\mathbb{R}^d

[Lawson 1977]

[Joe 1989] [Rajan 1991] [Edelsbrunner Shah 1996]

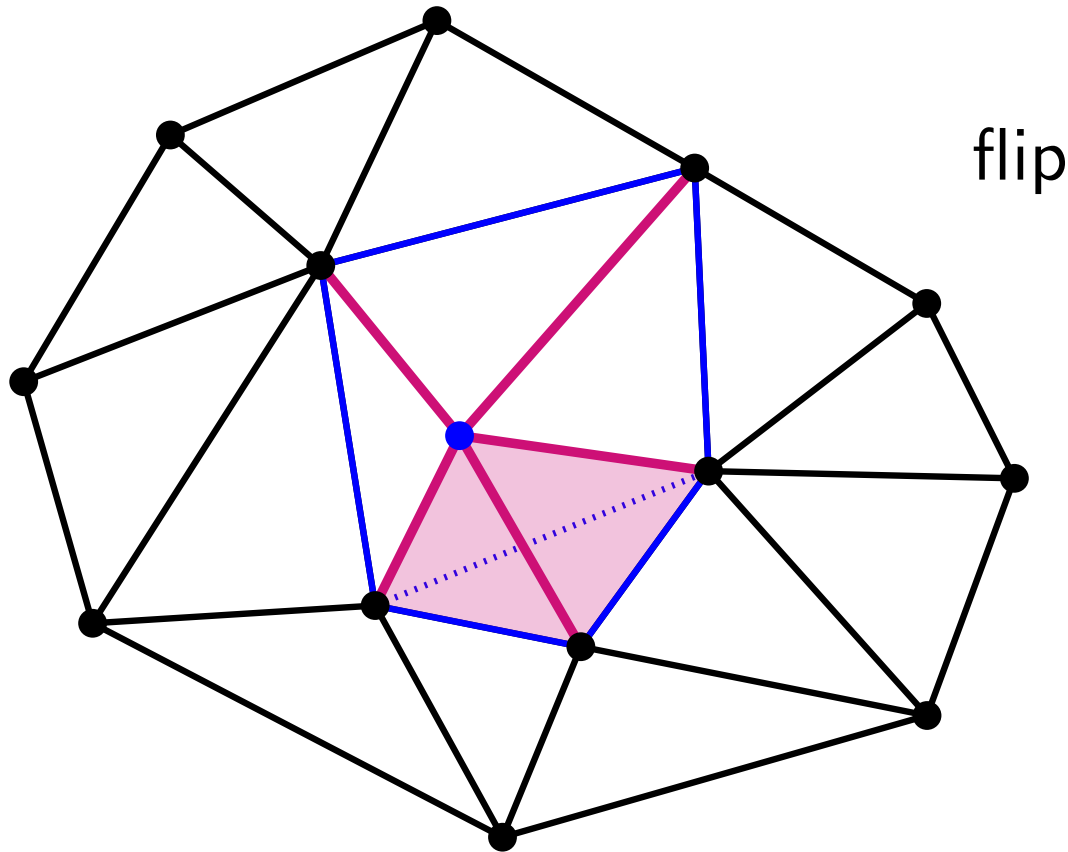


Lawson's flip algorithm

 \mathbb{R}^d

[Lawson 1977]

[Joe 1989] [Rajan 1991] [Edelsbrunner Shah 1996]

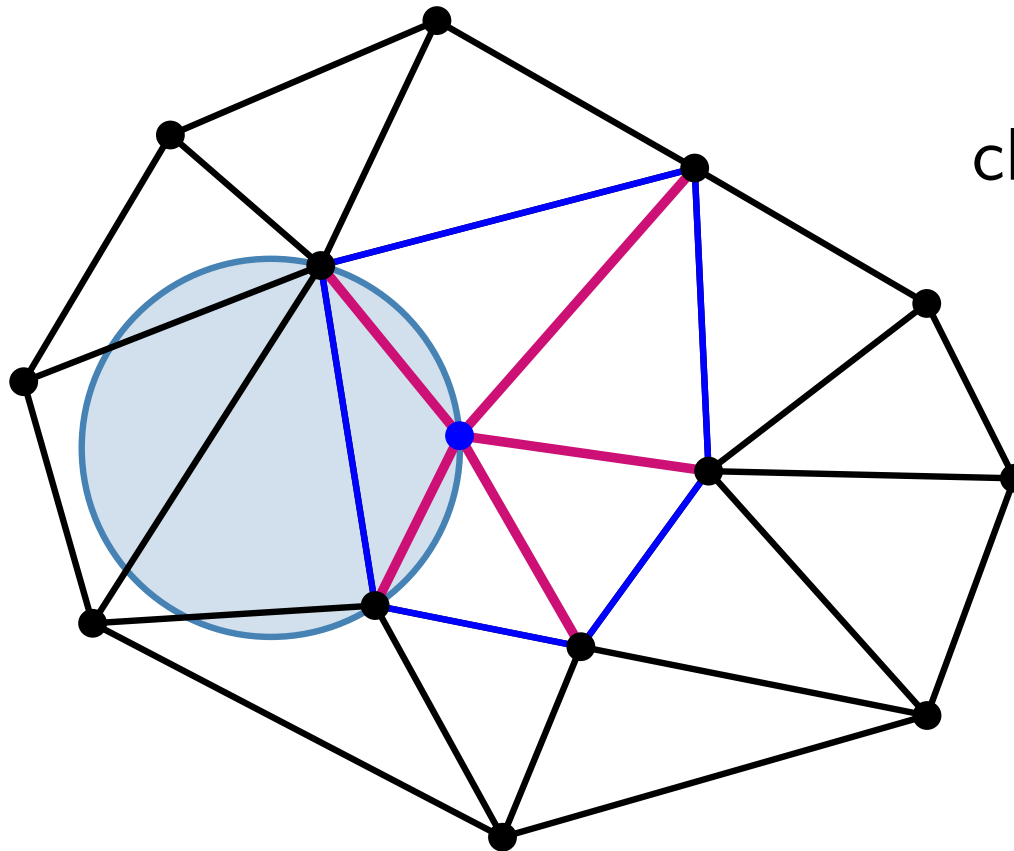


Lawson's flip algorithm

 \mathbb{R}^d

[Lawson 1977]

[Joe 1989] [Rajan 1991] [Edelsbrunner Shah 1996]

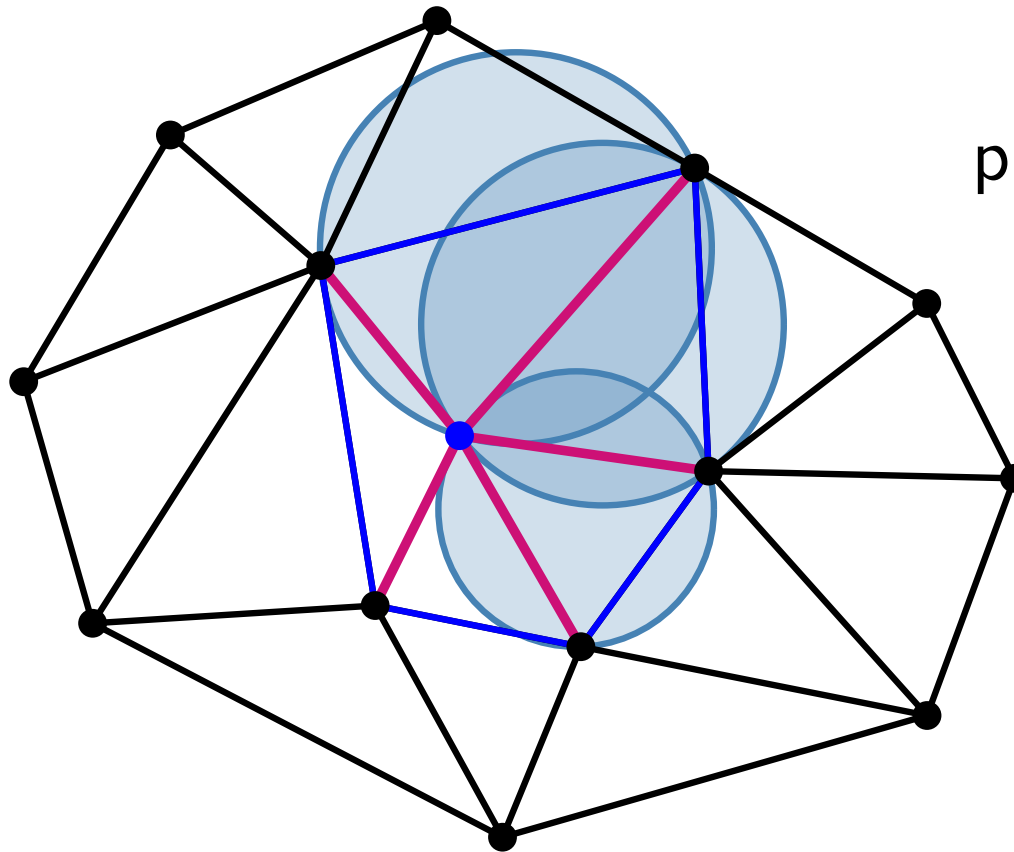


Lawson's flip algorithm

\mathbb{R}^d

[Lawson 1977]

[Joe 1989] [Rajan 1991] [Edelsbrunner Shah 1996]



Lawson's flip algorithm

\mathbb{R}^d

at each step

Triangulation
using Geom_traits

and

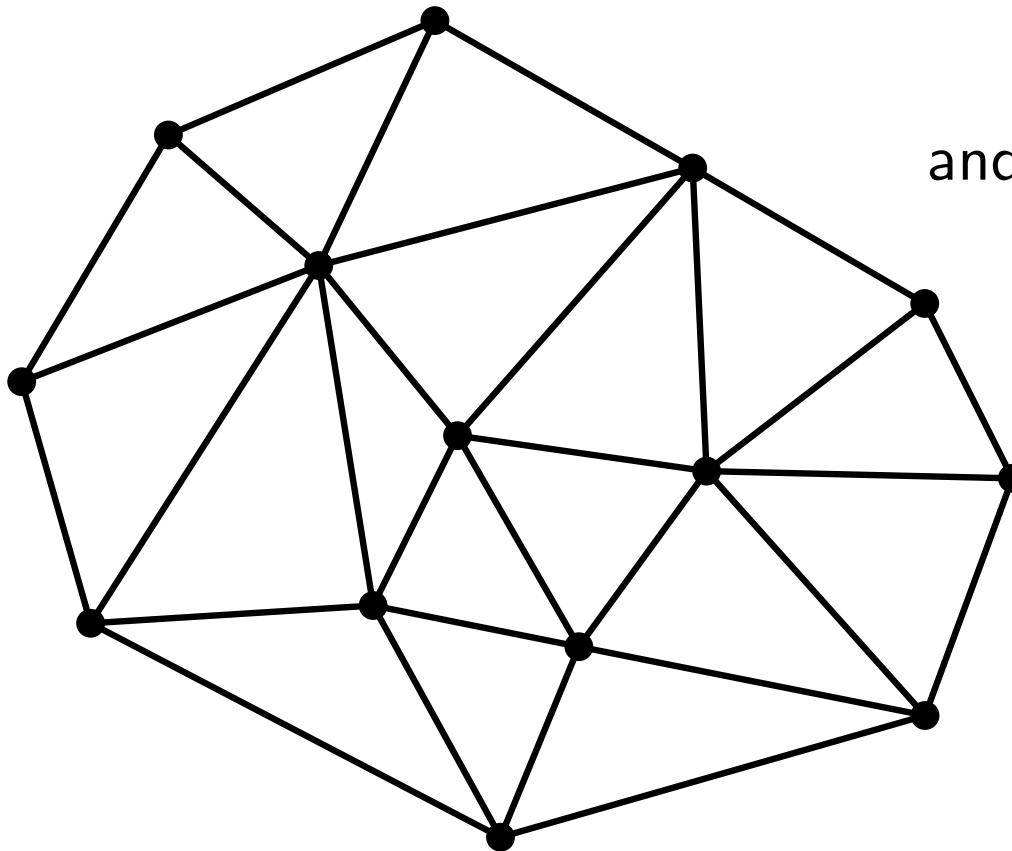
TDS

hard to maintain

early version of

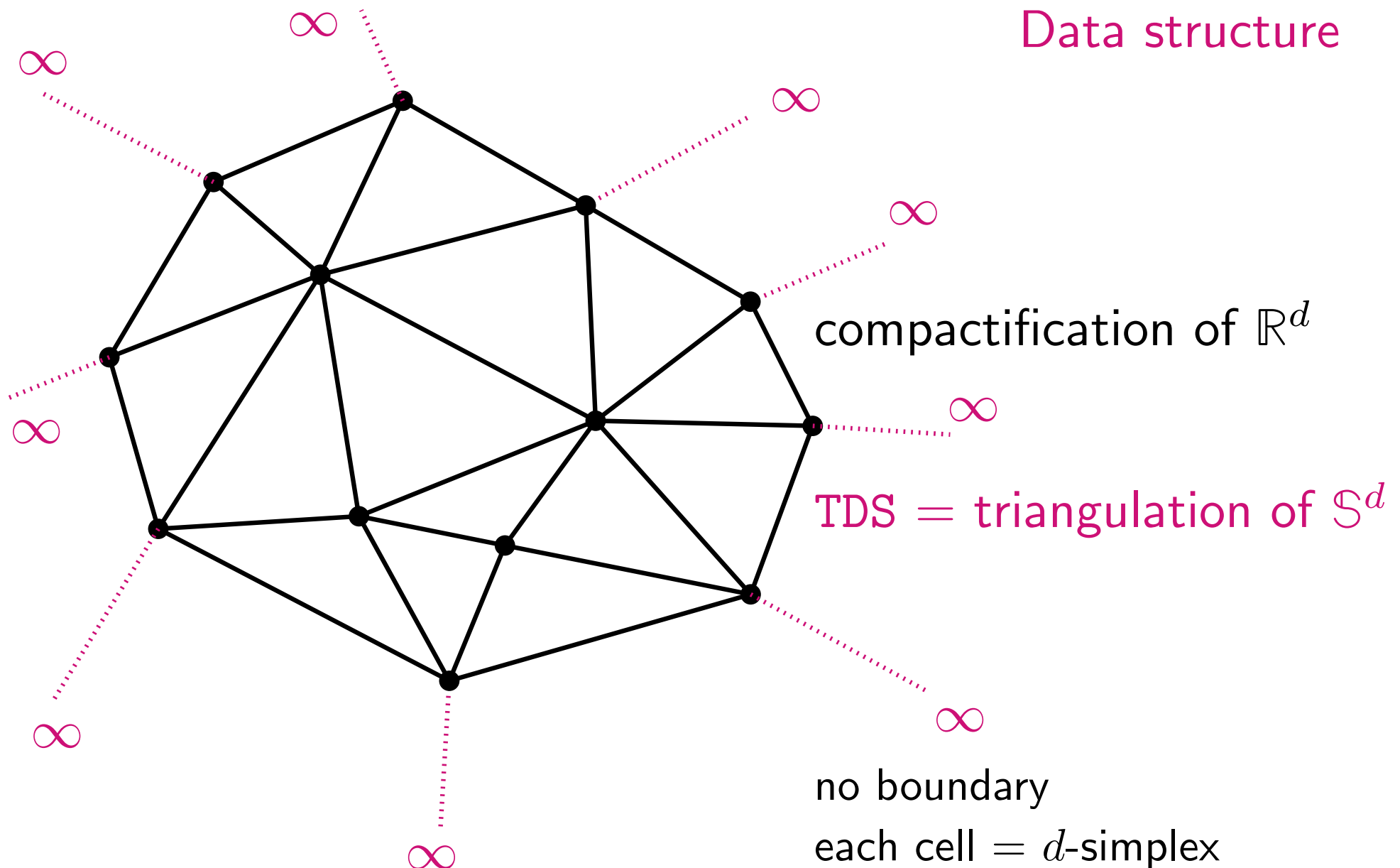
CGAL in \mathbb{R}^2

[Yvinec CGAL'97]



Implementation “details”

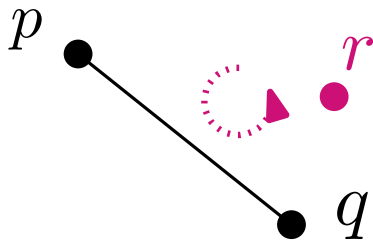
Implementation “details”



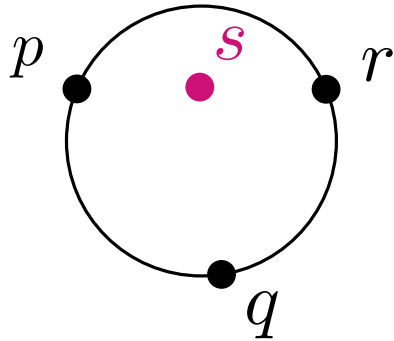
Implementation “details”

Robustness

Predicates



$$\text{orient}(p, q, r) = \text{sign} \begin{vmatrix} 1 & 1 & 1 \\ p_x & q_x & r_x \\ p_y & q_y & r_y \end{vmatrix}$$



$$\text{in_disk}(p, q, r, s) = \text{sign} \begin{vmatrix} 1 & x_p & y_p & x_p^2 + y_p^2 \\ 1 & x_q & y_q & x_q^2 + y_q^2 \\ 1 & x_r & y_r & x_r^2 + y_r^2 \\ 1 & x_s & y_s & x_s^2 + y_s^2 \end{vmatrix}$$

*center and radius are **not** computed*

sign of polynomial expression

rational input points

Implementation “details”

Robustness

Arithmetic issues

Exact Geometric Computation paradigm [Yap]

exact predicates \neq exact arithmetics

Filtering

easy cases are more frequent

\implies cost \simeq cost of approximate (double) computation

Implementation “details”

Robustness

Arithmetic issues

Approximate evaluation $P^a(x)$
+ Error ϵ

Filtering

$$P^a(x) > \epsilon$$

?

Yes

No

$$\text{Sign}(P(x)) = \text{Sign}(P^a(x))$$

Exact computation

Implementation “details”

Robustness

Degenerate cases

symbolic perturbation

only perturbs the `in_sphere` predicate

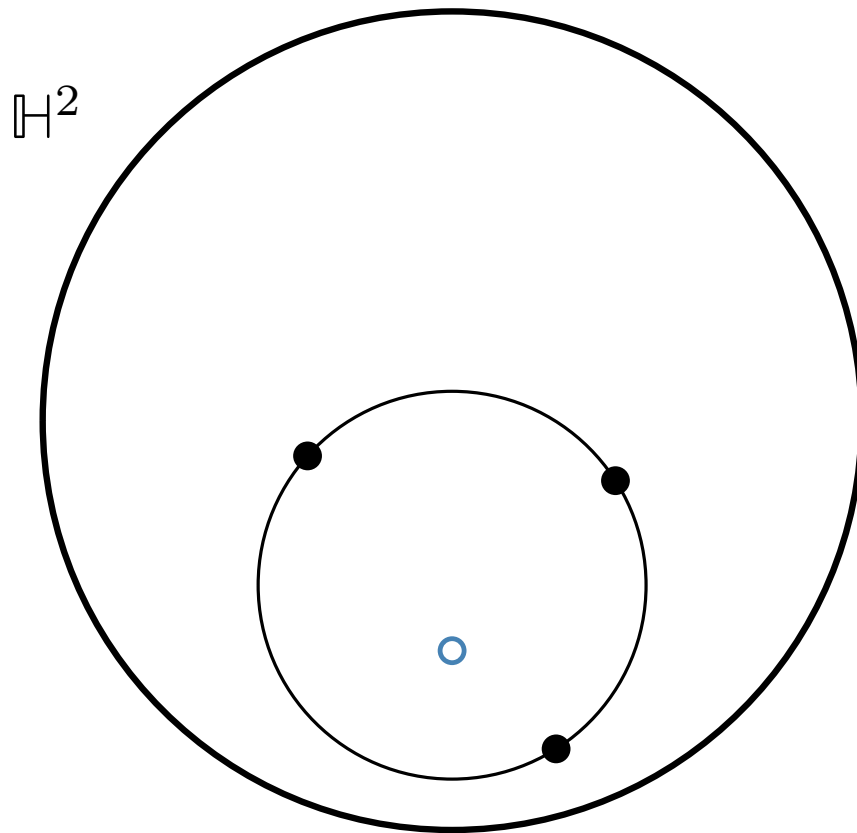
[Devillers, T. SODA'03, CGTA'11]

Implementation “details”

Algorithmic tricks to improve efficiency

Delaunay triangulation in \mathbb{H}^2

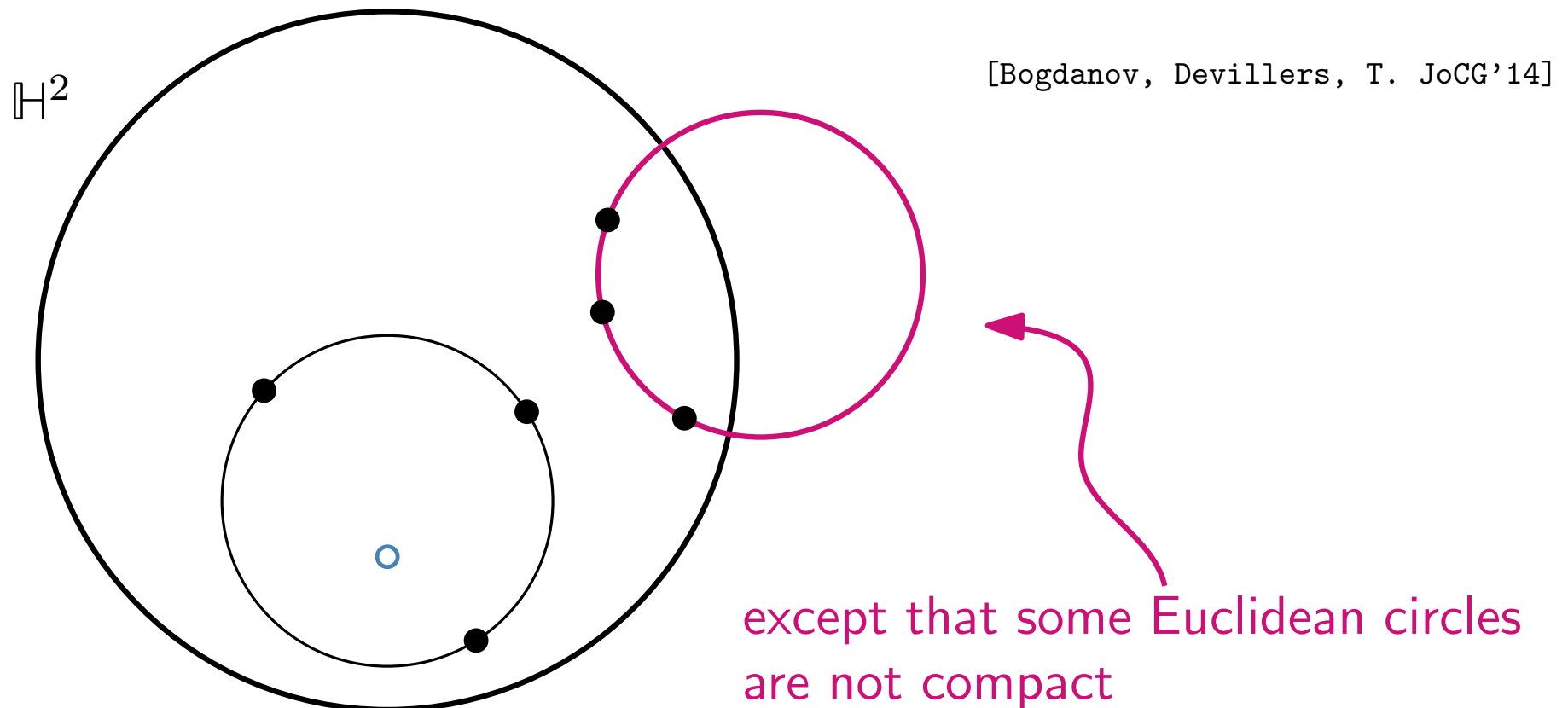
Hyperbolic triangulations



hyperbolic circles are Euclidean circles

↳ same combinatorics as Euclidean Delaunay triangulation

Hyperbolic triangulations



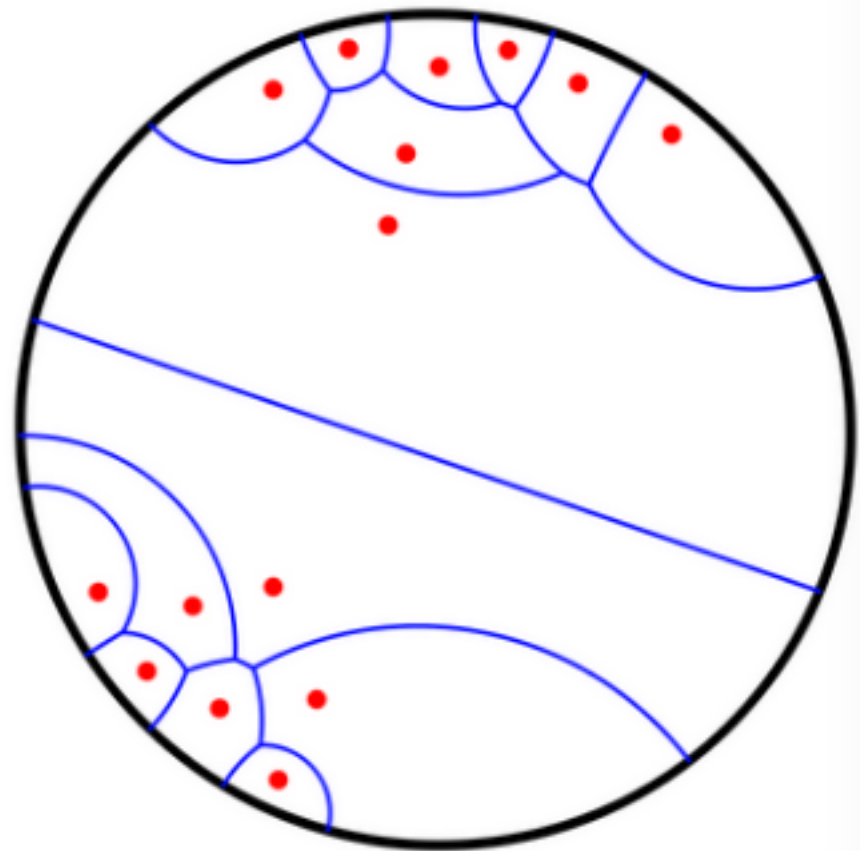
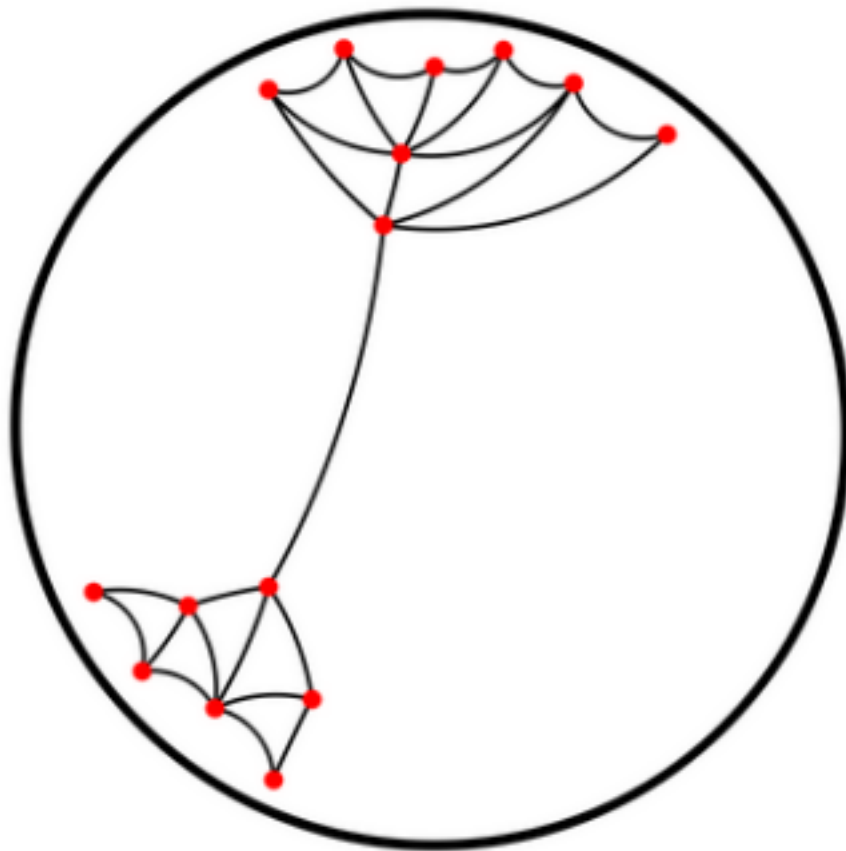
hyperbolic circles are Euclidean circles

↳ same combinatorics as Euclidean Delaunay triangulation

Hyperbolic triangulations

Hyperbolic_Delaunay_triangulation_2
inherits from
Delaunay_triangulation_2

[Bogdanov, Iordanov, T. CGAL'19]



Surfaces

Closed oriented surfaces

Closed oriented surfaces

genus 1

$$\mathbb{T}_u^2 \sim \mathbb{R}^2 / (\mathbb{Z}e_1 + \mathbb{Z}u) \quad \text{flat torus}$$

$$e_1 = (1, 0), u = (u_x, u_y)$$

Closed oriented surfaces

genus 1

$$\mathbb{T}_u^2 \sim \mathbb{R}^2 / (\mathbb{Z}e_1 + \mathbb{Z}u) \quad \text{flat torus}$$

$$e_1 = (1, 0), u = (u_x, u_y)$$

genus ≥ 2

$$S \sim \mathbb{H}^2 / G$$

G Fuchsian group

= discrete group of
orientation preserving isometries of \mathbb{H}^2

isometric to fundamental group $\pi_1(S)$

Closed oriented surfaces

genus 1

$$\mathbb{T}_u^2 \sim \mathbb{R}^2 / (\mathbb{Z}e_1 + \mathbb{Z}u) \quad \text{flat torus}$$

$$e_1 = (1, 0), u = (u_x, u_y)$$

genus ≥ 2

$$S \sim \mathbb{H}^2 / G \quad \text{non-Abelian}$$

Poincaré disk
model

G Fuchsian group

= discrete group of
orientation preserving isometries of \mathbb{H}^2

isometric to fundamental group $\pi_1(S)$

Closed oriented surfaces

genus 1

universal cover

$$\mathbb{T}_u^2 \sim \mathbb{R}^2 / (\mathbb{Z}e_1 + \mathbb{Z}u) \quad \text{flat torus}$$
$$e_1 = (1, 0), u = (u_x, u_y)$$

genus ≥ 2

universal cover

$$S \sim \mathbb{H}^2 / G \quad \text{non-Abelian}$$

Poincaré disk model

G Fuchsian group
= discrete group of orientation preserving isometries of \mathbb{H}^2
isometric to fundamental group $\pi_1(S)$

Closed oriented surfaces

genus 1

universal cover

$$\mathbb{T}_u^2 \sim \mathbb{R}^2 / (\mathbb{Z}e_1 + \mathbb{Z}u) \quad \text{flat torus} \quad \pi : \mathbb{R}^2 \rightarrow \mathbb{T}^2$$

$$e_1 = (1, 0), u = (u_x, u_y)$$

genus ≥ 2

universal cover

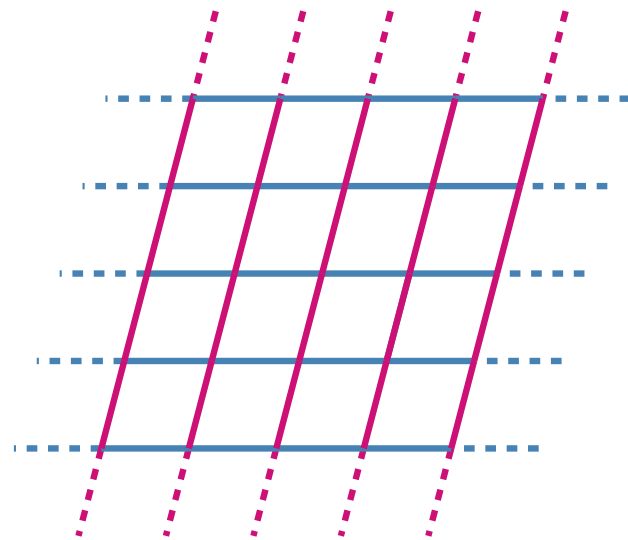
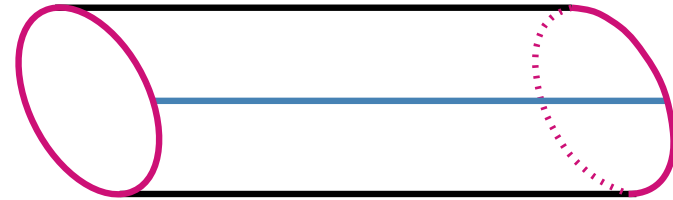
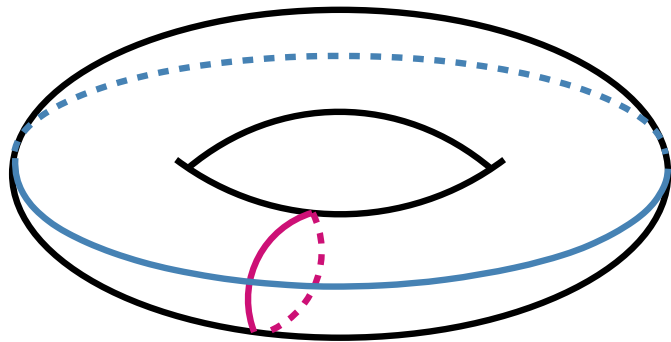
$$S \sim \mathbb{H}^2 / G \quad \text{non-Abelian} \quad \pi : \mathbb{H}^2 \rightarrow S$$

Poincaré disk model

G Fuchsian group
 = discrete group of orientation preserving isometries of \mathbb{H}^2
 isometric to fundamental group $\pi_1(S)$

Delaunay triangulation

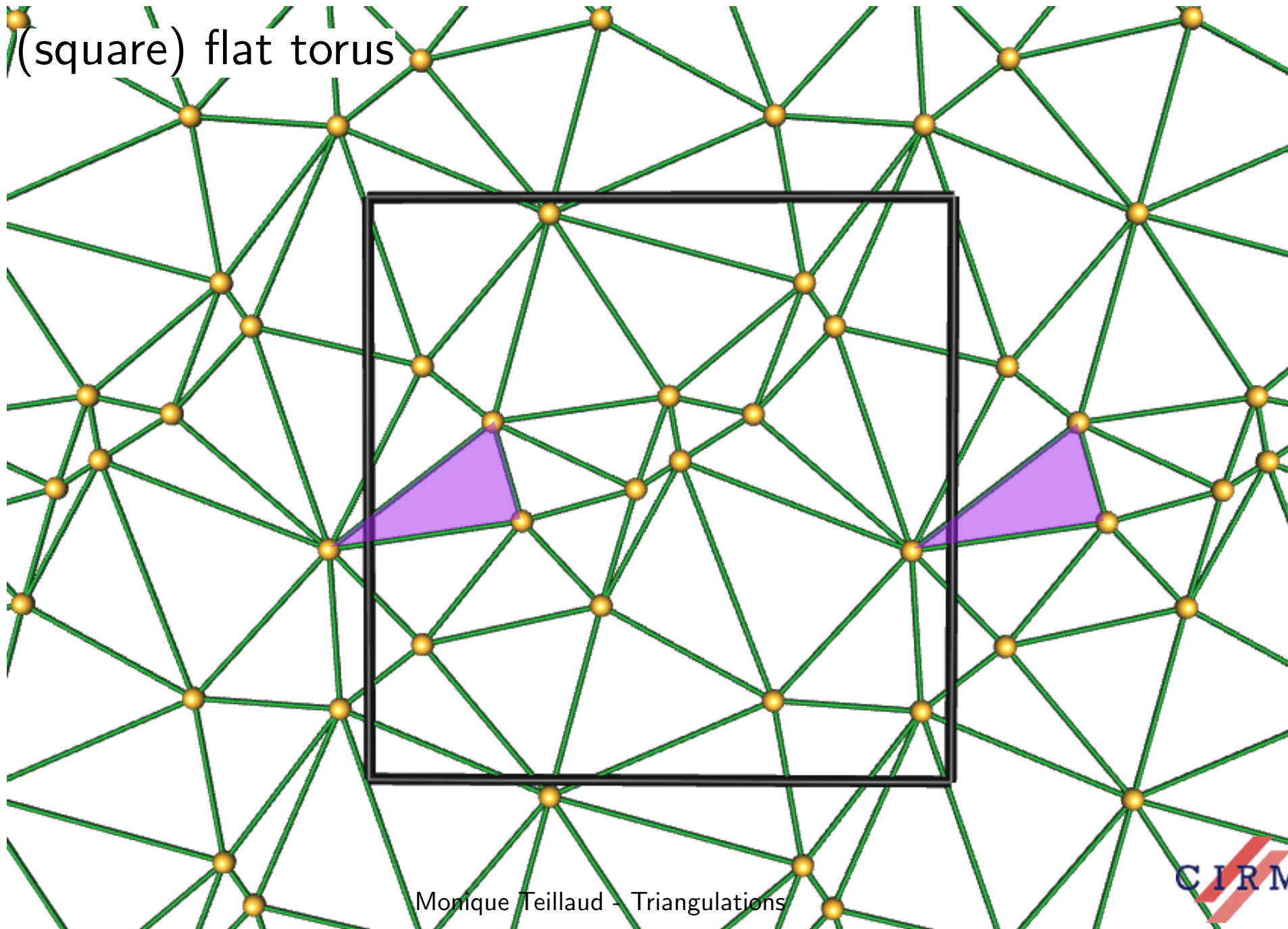
flat torus



universal covering space

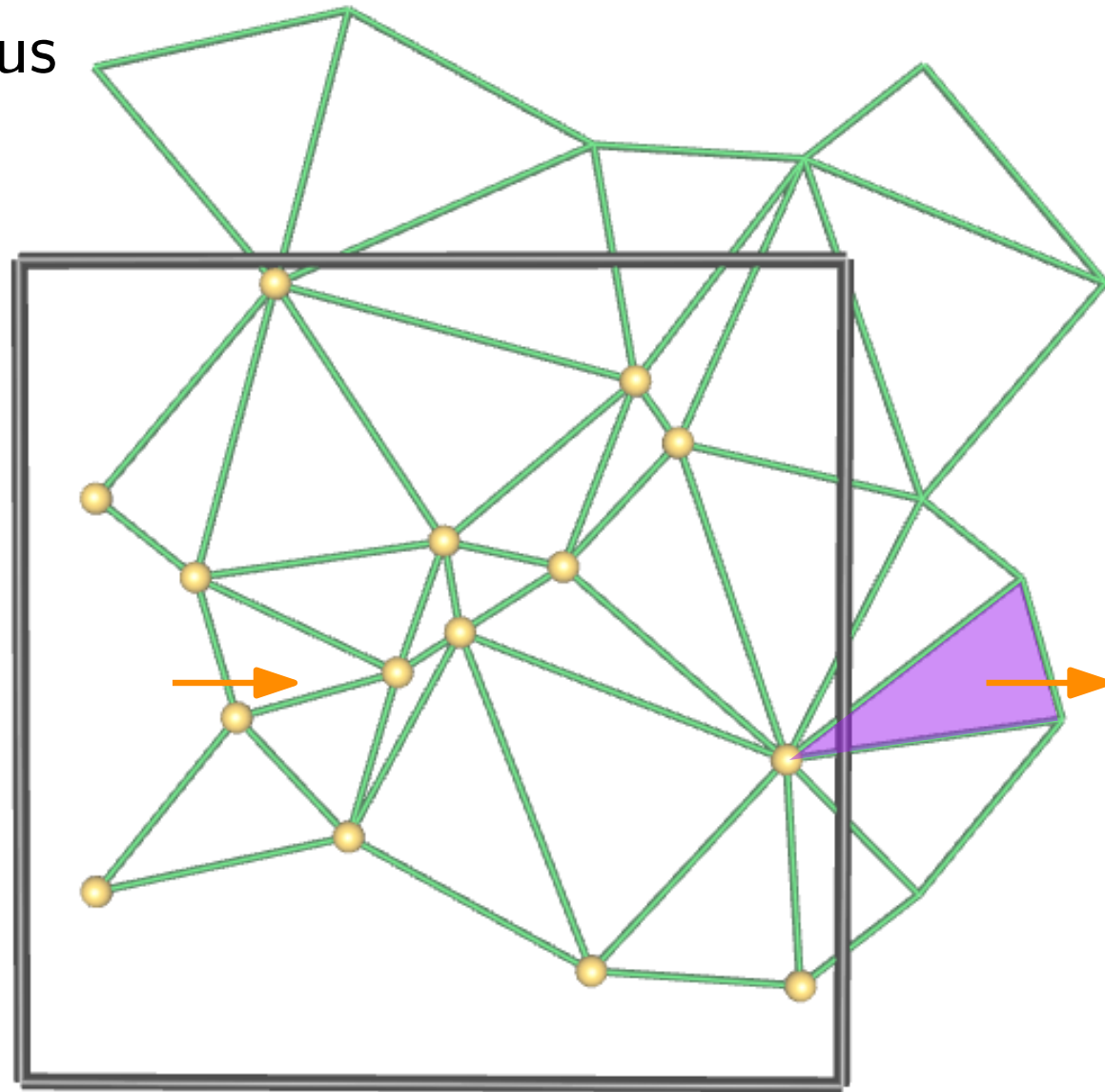
Delaunay triangulation

(square) flat torus



Delaunay triangulation

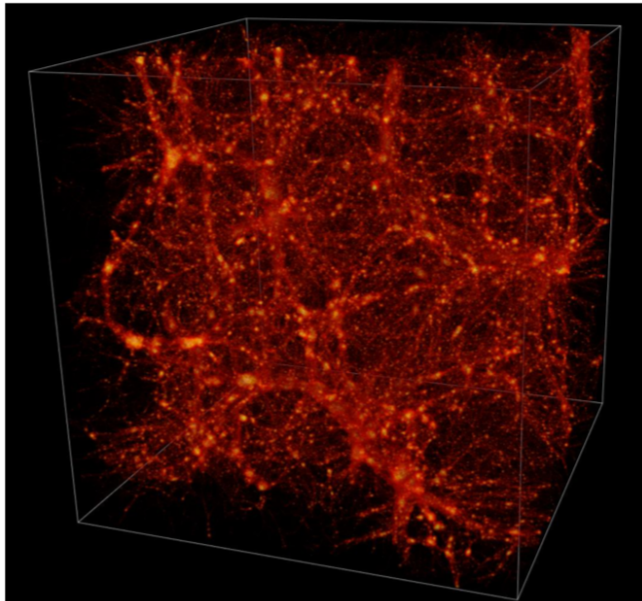
(square) flat torus



Delaunay triangulation

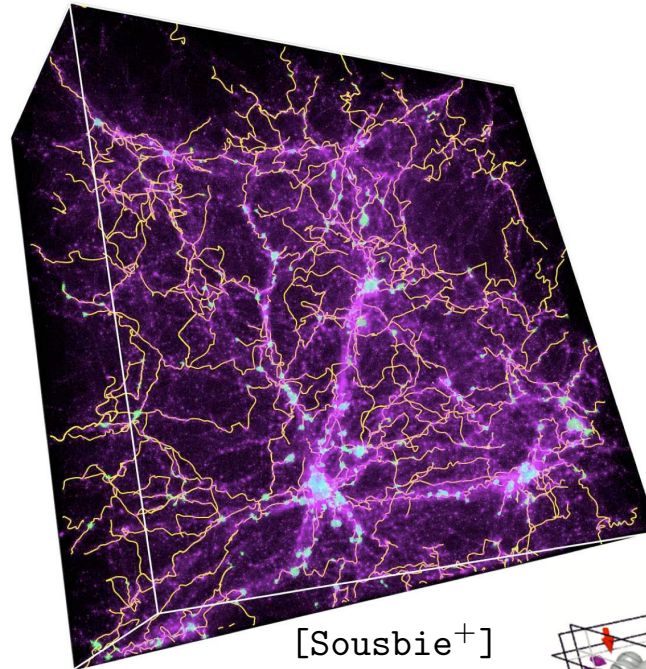
(cubic) 3D flat torus

Astrophysics



[v.d. Weijgaert⁺]

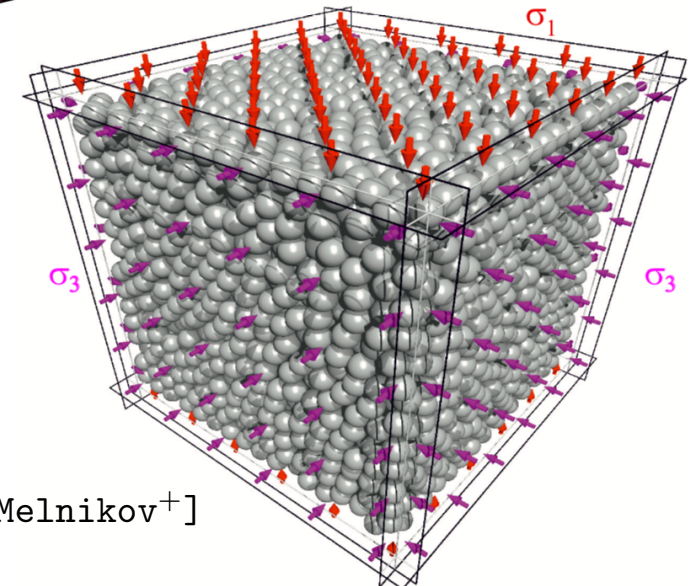
[SoCG'12 video]



[Sousbie⁺]

Particle physics
Nanostructures
etc

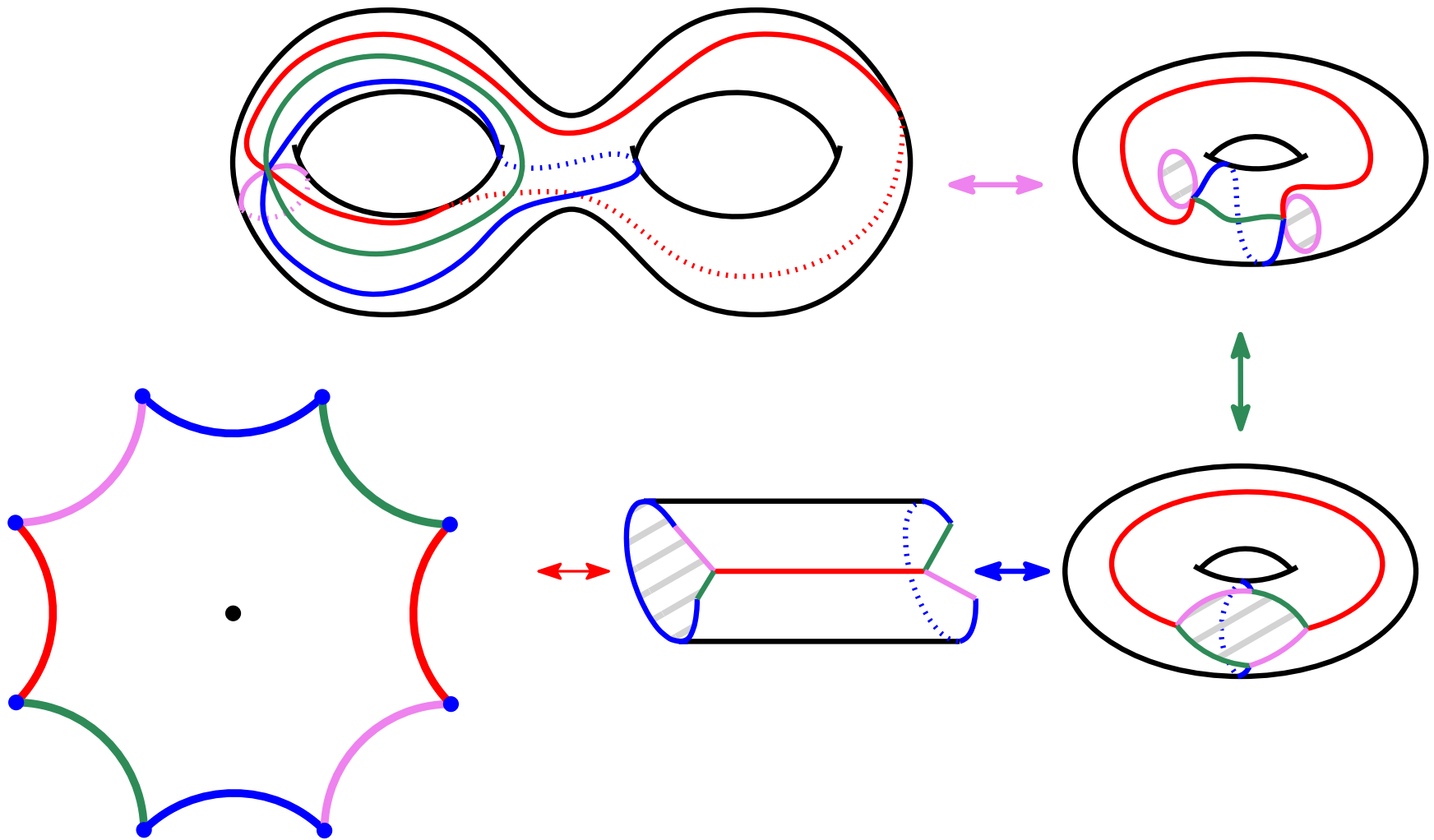
Granular materials



[Melnikov⁺]

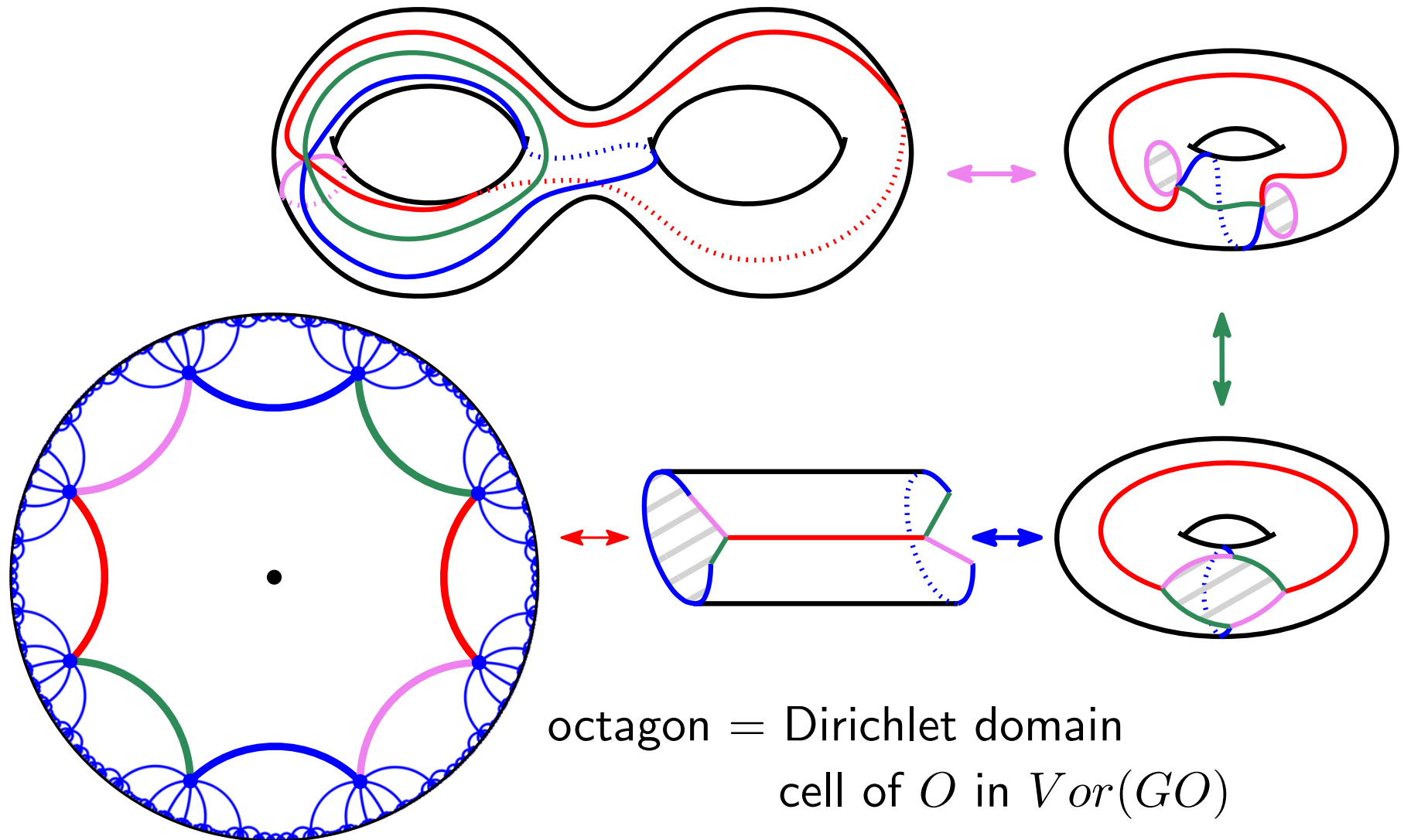
Delaunay triangulation

(Bolza) hyperbolic surface



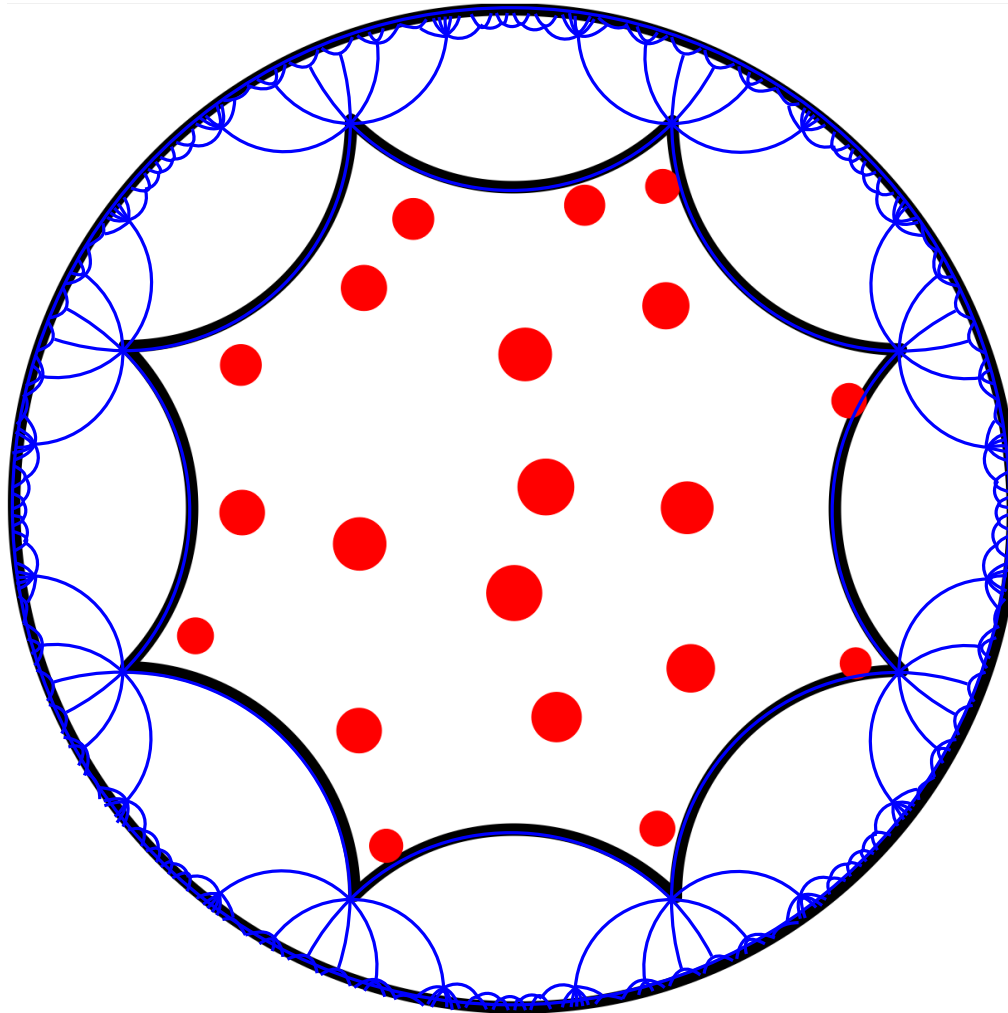
Delaunay triangulation

(Bolza) hyperbolic surface



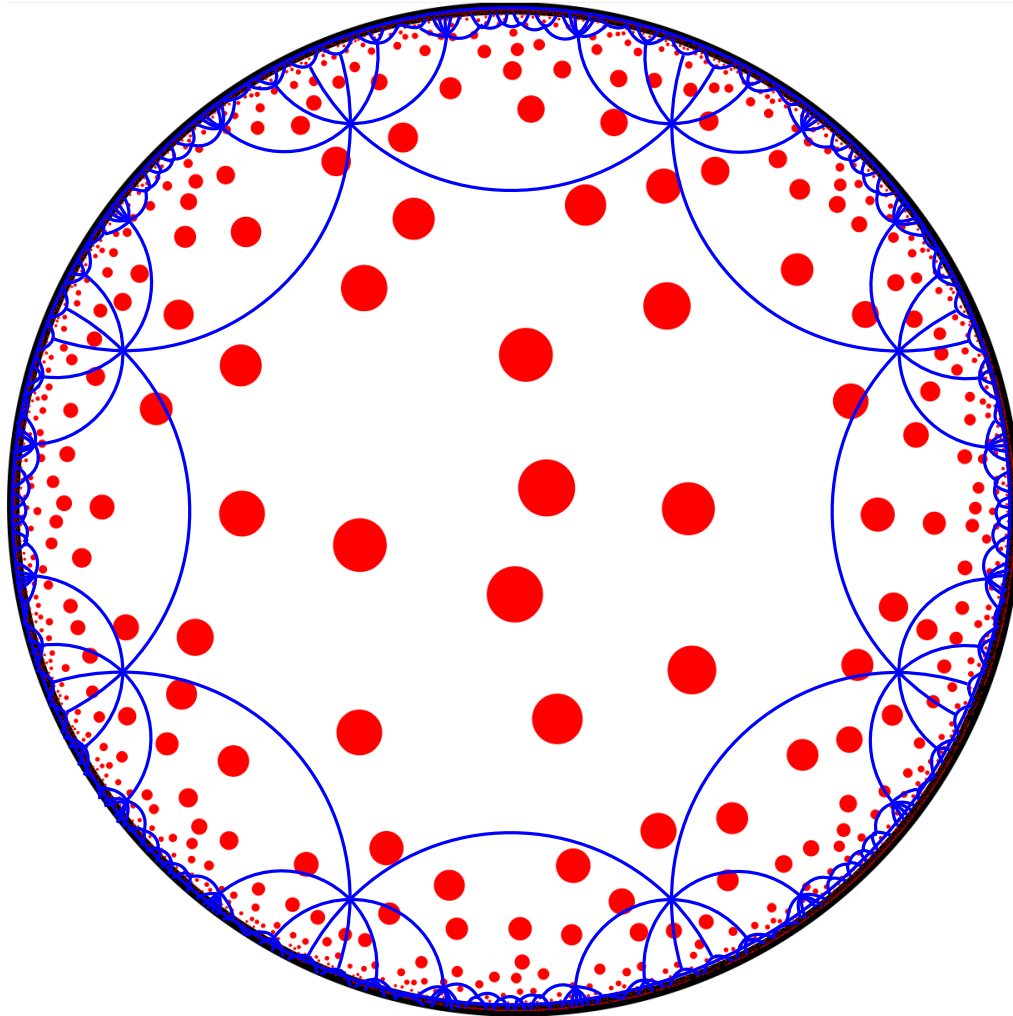
Delaunay triangulation

(Bolza) hyperbolic surface



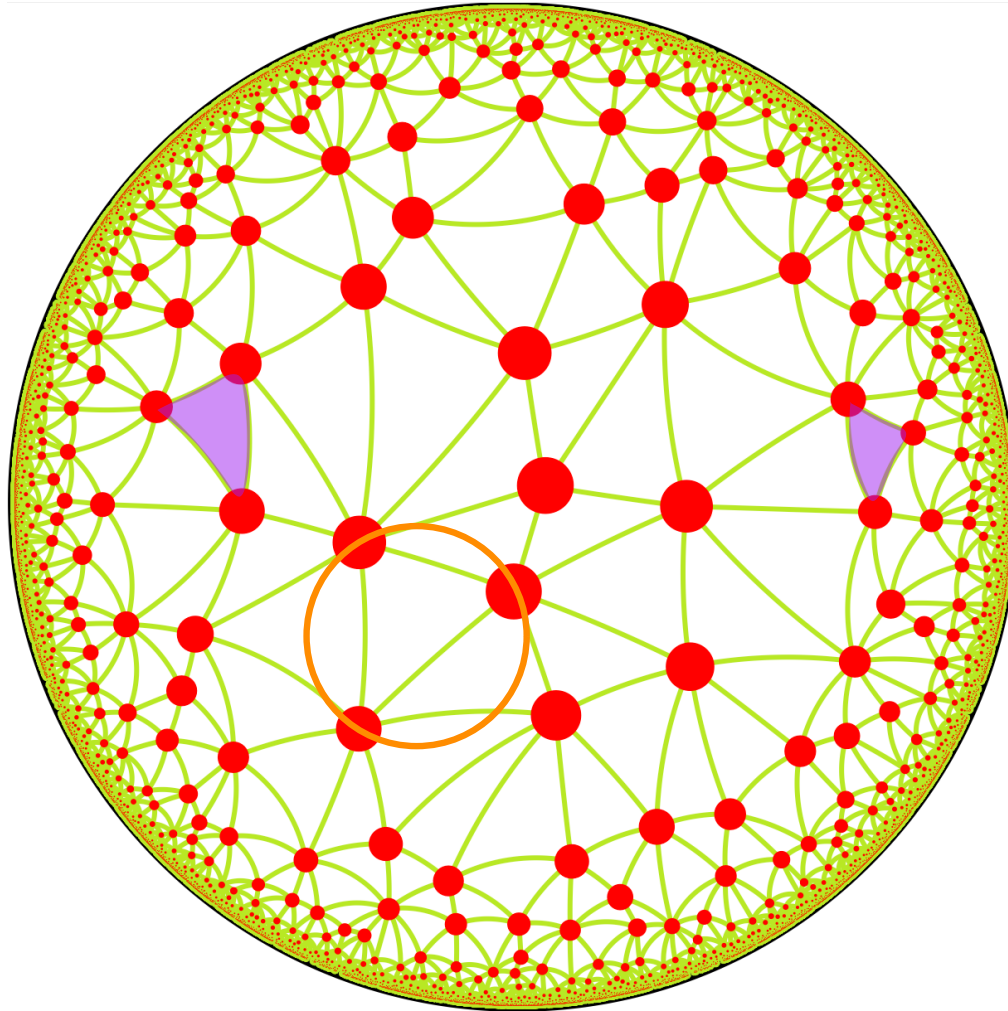
Delaunay triangulation

(Bolza) hyperbolic surface



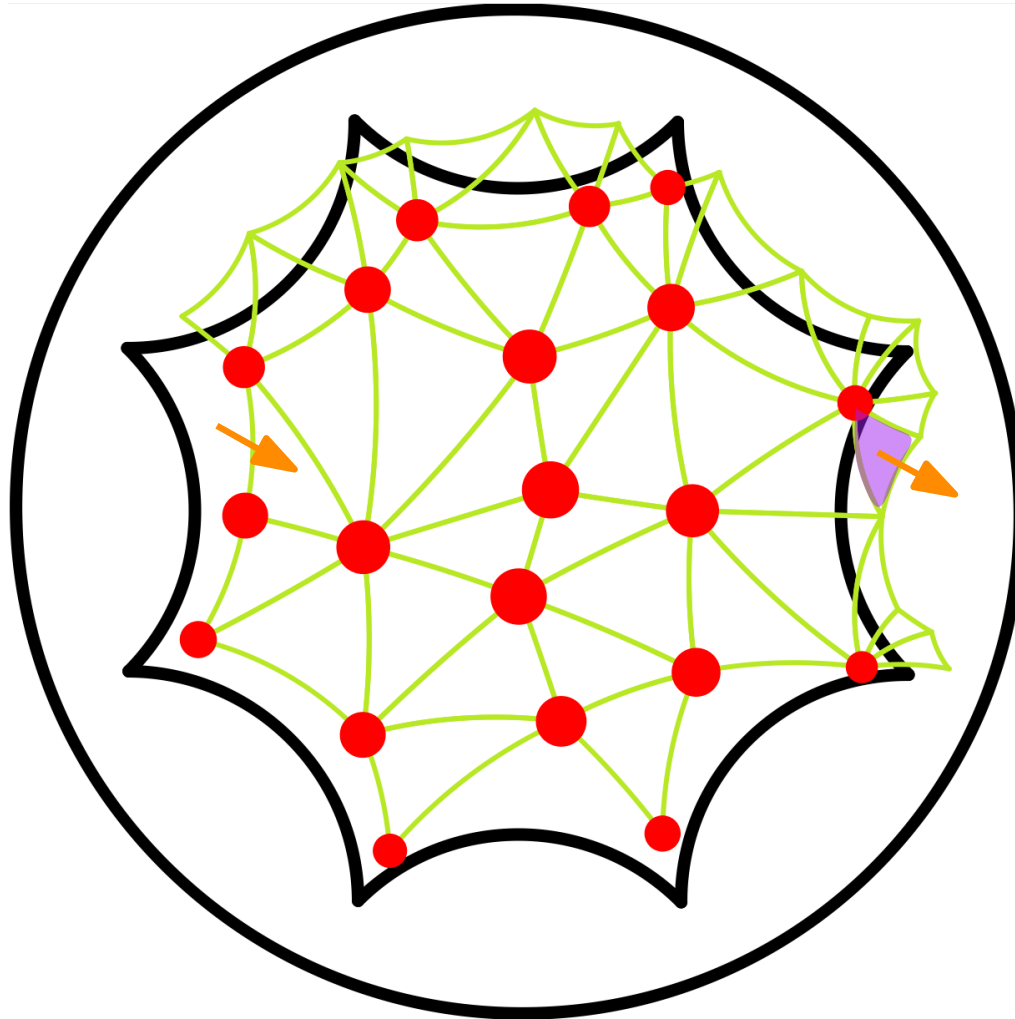
Delaunay triangulation

(Bolza) hyperbolic surface



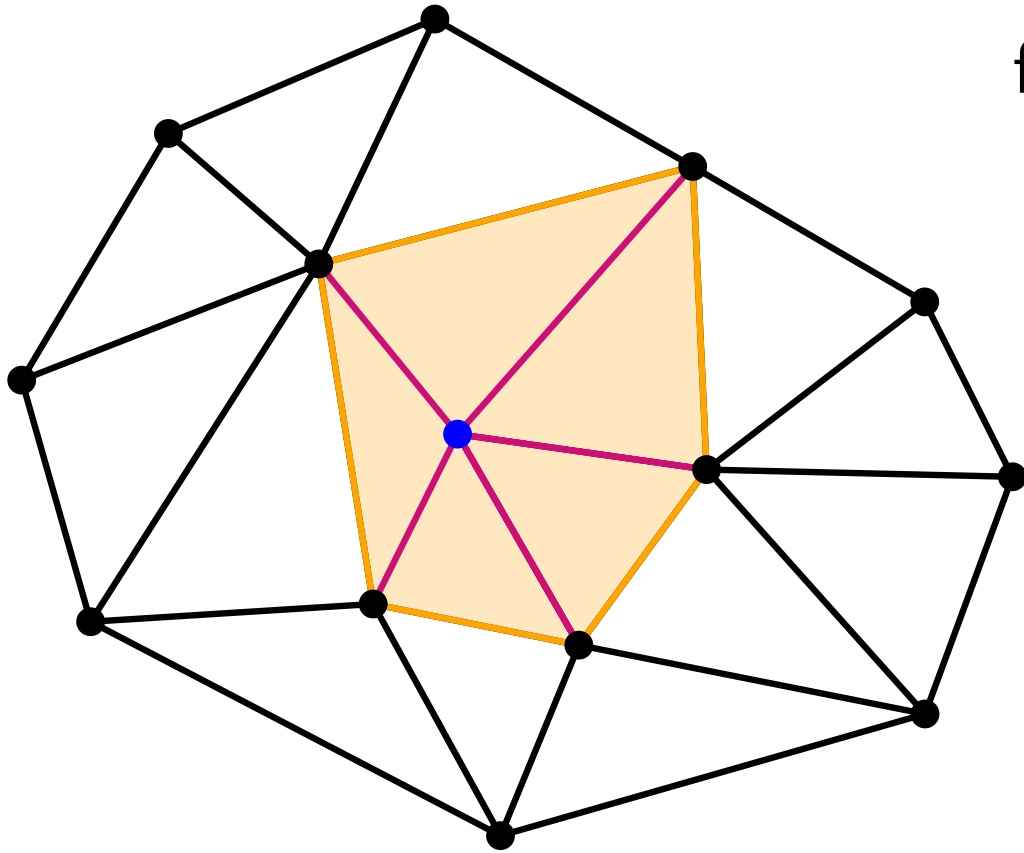
Delaunay triangulation

(Bolza) hyperbolic surface



Bowyer's algorithm on a surface

Back to \mathbb{R}^2



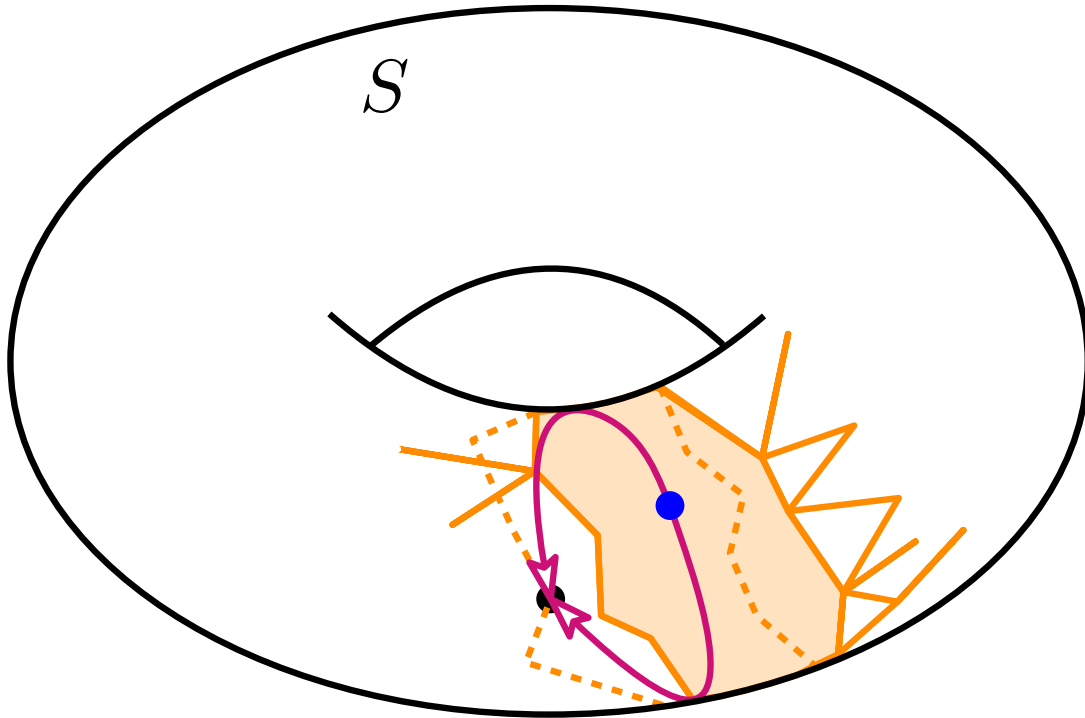
fill the hole

*the hole is
a topological disk*

[Bowyer 1981] [Watson 1981]

On a surface

Validity condition



the hole is **not** always
a topological disk

multiple geodesics between two points

If

$$\mathit{sys}(S) > 2\Delta(Q)$$

then

the hole is
a topological disk

when inserting points in any set $\mathcal{P} \supseteq Q$

On a surface

Validity condition

do not contain any point in $\pi^{-1}(Q)$

If diameter of the largest empty disks
in the universal cover

$$\text{sys}(S) > 2\Delta(Q)$$

then

systole = length of a shortest
non-contractible curve on S

**the hole is
a topological disk**

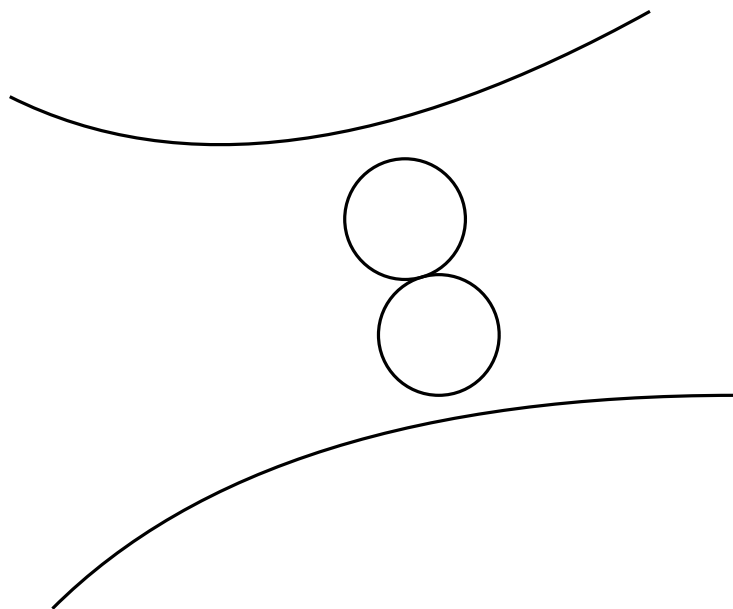
when inserting points in any set $\mathcal{P} \supseteq Q$

On a surface

Validity condition

$$\text{sys}(S) > 2\Delta(Q)$$

Proof



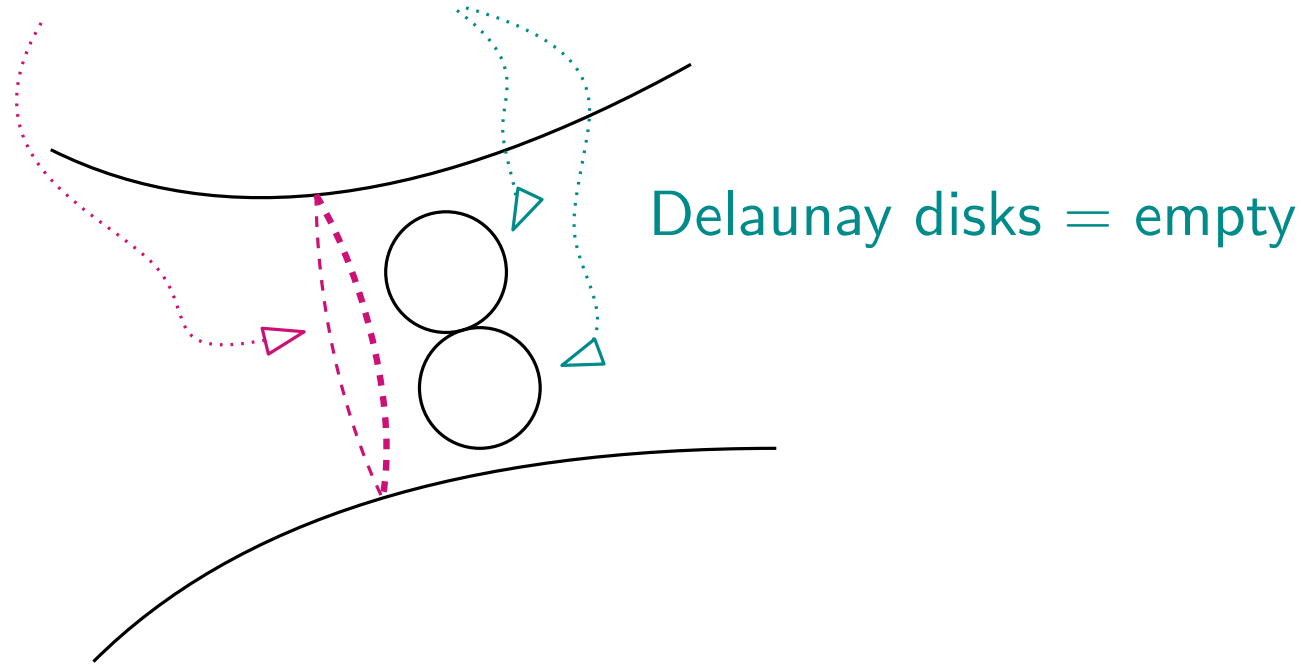
[Caroli T. ESA'09 DCG'16] [Bogdanov T. Vegter SoCG'16]

On a surface

Validity condition

$$\text{sys}(S) > 2\Delta(Q)$$

Proof



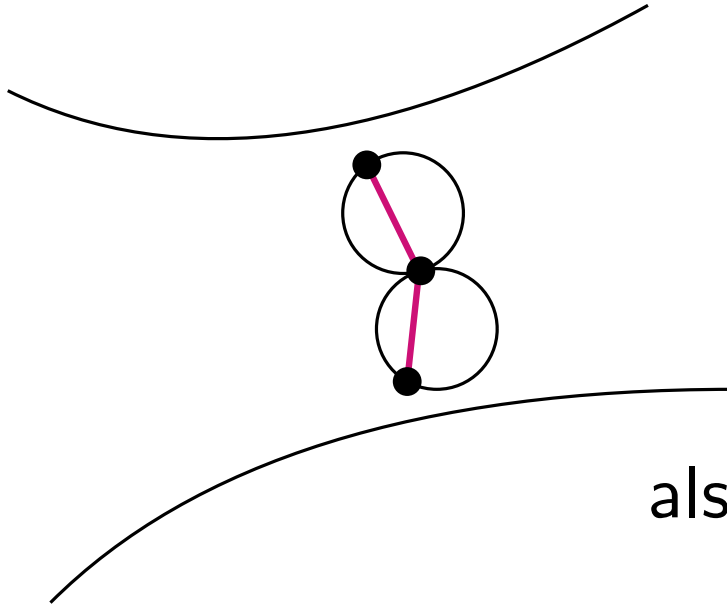
[Caroli T. ESA'09 DCG'16] [Bogdanov T. Vegter SoCG'16]

On a surface

Validity condition

$$\text{sys}(S) > 2\Delta(Q)$$

Proof



two Delaunay edges
cannot form a cycle

also in $DT(\mathcal{P})$ for $\mathcal{P} \supseteq Q$

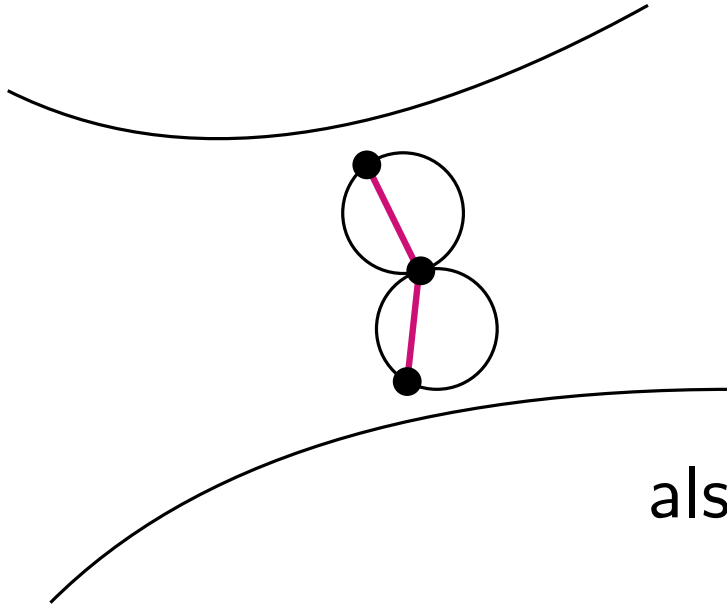
$DT(\mathcal{P})$ is a simplicial complex

On a surface

Validity condition

$$\text{sys}(S) > 2\Delta(Q)$$

Proof



two Delaunay edges
cannot form a cycle

also in $DT(\mathcal{P})$ for $\mathcal{P} \supseteq Q$

$DT(\mathcal{P})$ is a simplicial complex

the hole is
a topological disk



Bowyer's algorithm

“dummy” points

Initialize the Delaunay triangulation in \mathbb{H}^2 with \mathcal{Q} s.t.

$$\text{sys}(S) > 2\Delta(\mathcal{Q})$$

Compute the DT directly on the surface

Insert input points

Remove the points of \mathcal{Q} if possible

Bowyer's algorithm

"dummy" points

Initialize the Delaunay triangulation in \mathbb{H}^2 with \mathcal{Q} s.t.

$$\text{sys}(S) > 2\Delta(\mathcal{Q})$$

Compute the DT directly on the surface

Insert input points

Remove the points of \mathcal{Q} if possible



3D cubic flat torus [Caroli T. CGAL'09]

2D square flat torus [Kruithof CGAL'13]

Bolza surface [Iordanov T. CGAL'19]

Bowyer's algorithm

"dummy" points

Initialize the Delaunay triangulation in \mathbb{H}^2 with \mathcal{Q} s.t.

$$\text{sys}(S) > 2\Delta(\mathcal{Q})$$

Compute the DT directly on the surface

Insert input points

Remove the points of \mathcal{Q} if possible

Questions

$\text{sys}(S)$?

Algorithms computing \mathcal{Q} ?

High degree algebraic numbers

Number of points in \mathcal{Q} ?

Bounds for *generalized* Bolza surfaces

[Ebbens Iordanov T. Vegter JoCG'22]

Another approach

Bowyer

~~Initialize the Delaunay triangulation in \mathbb{H}^2 with \mathcal{Q}~~

$$sys(S) > 2\Delta(\mathcal{Q})$$

Compute the DT ~~directly on the surface~~ in \mathbb{H}^2

Insert input points

~~Remove the points of \mathcal{Q} if possible~~

Project to the surface

Another approach

Bowyer

~~Initialize the Delaunay triangulation in \mathbb{H}^2 with \mathcal{Q}~~

$$sys(S) > 2\Delta(\mathcal{Q})$$

Compute the DT ~~directly on the surface~~ in \mathbb{H}^2

Insert input points

~~Remove the points of \mathcal{Q} if possible~~

Project to the surface

Questions

How far do we need to go?

How many copies of points do we need?

Another approach

Bowyer

knowing a Dirichlet domain

flat torus: ≤ 13

lattice reduction

[Dolbilin Huson 1997]

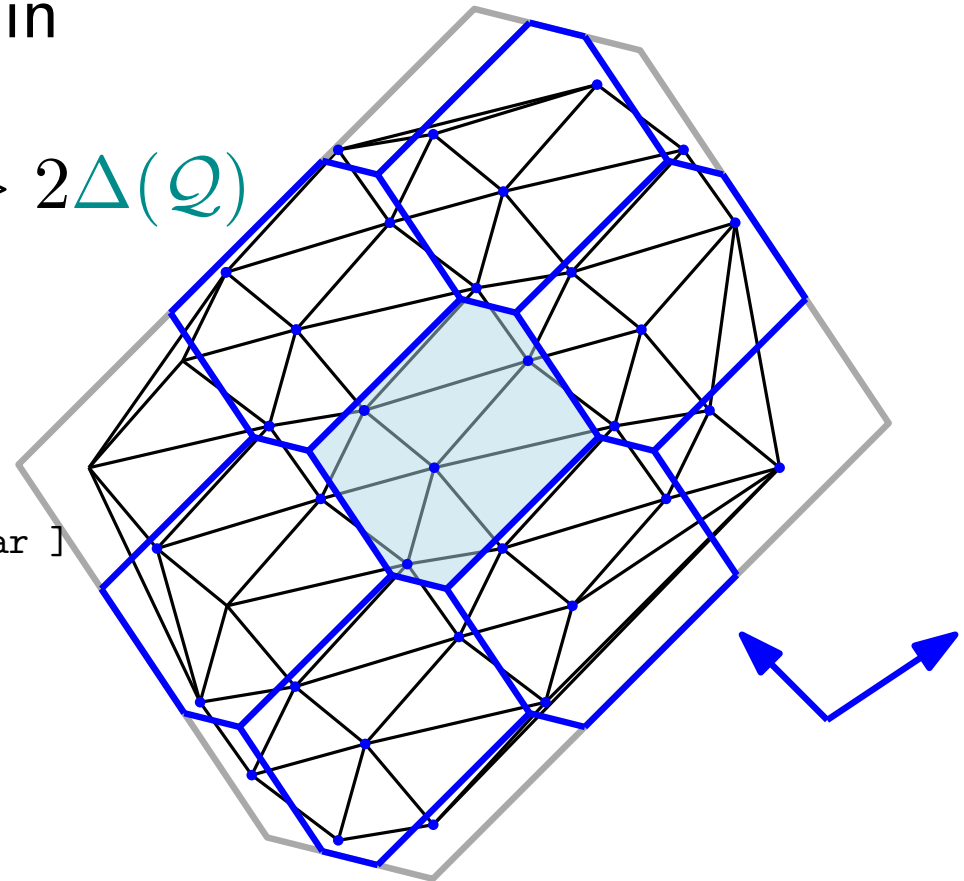
project as soon as

validity condition holds

[Osang Rouxel-Labbé T. ESA'20]

[Osang Rouxel-Labbé T. CGAL To Appear]

$$\text{sys}(S) > 2\Delta(Q)$$



Questions

How far do we need to go?

How many copies of points do we need?

Another approach

Bowyer

knowing a **Dirichlet** domain

flat torus: ≤ 13

lattice reduction

[Dolbilin Huson 1997]

project as soon as

validity condition holds

[Osang Rouxel-Labbé T. ESA'20]

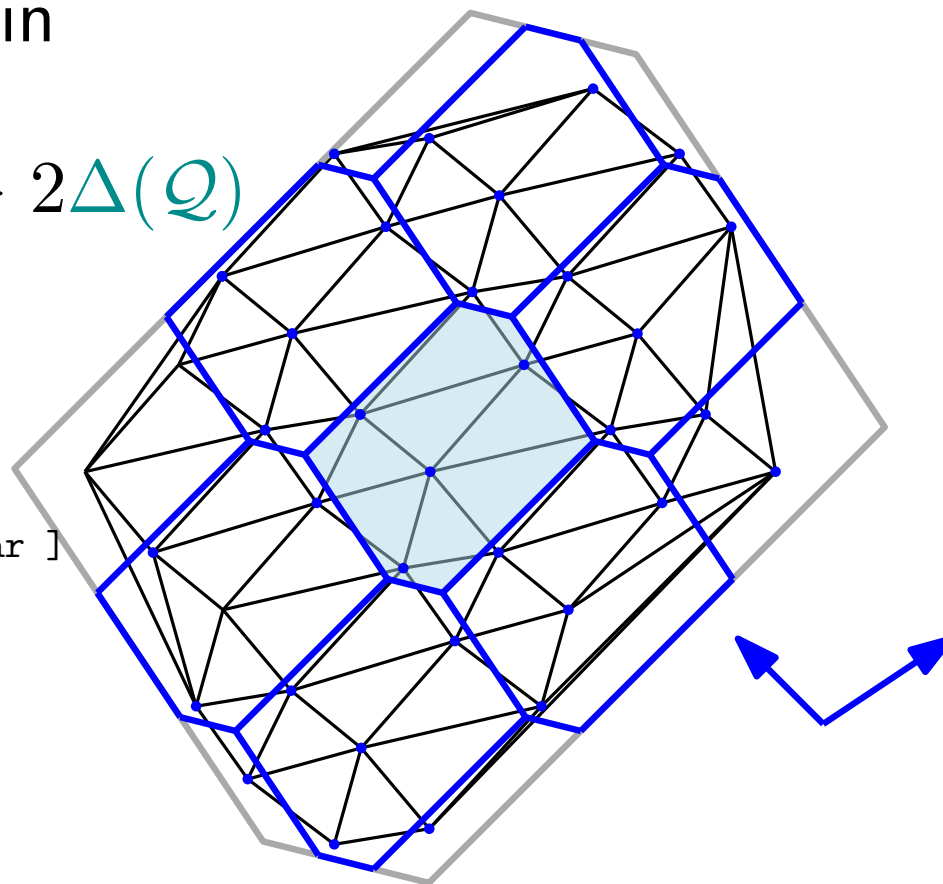
[Osang Rouxel-Labbé T. CGAL To Appear]

general hyperbolic surface:

$$O((12g)^{12g-6})$$

[Despré Kolbe T. HAL'20]

$$\text{sys}(S) > 2\Delta(Q)$$



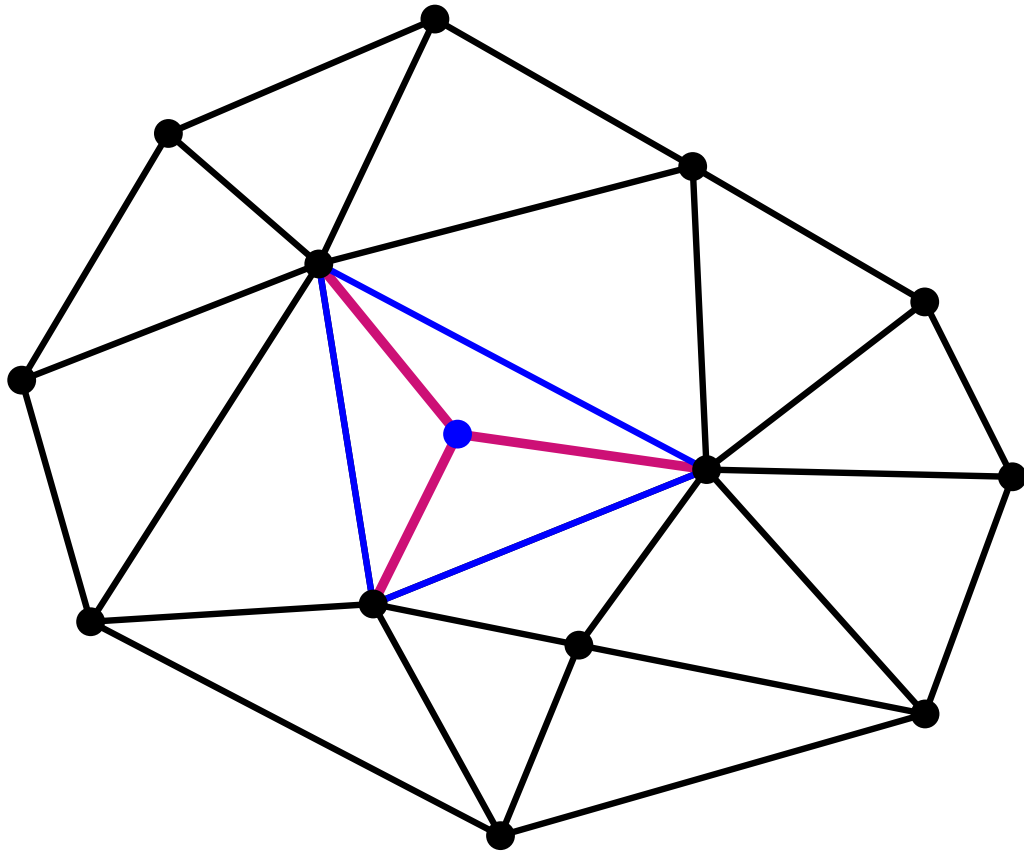
Questions

How far do we need to go?

How many copies of points do we need?

Flip algorithm on a surface

Back to \mathbb{R}^2



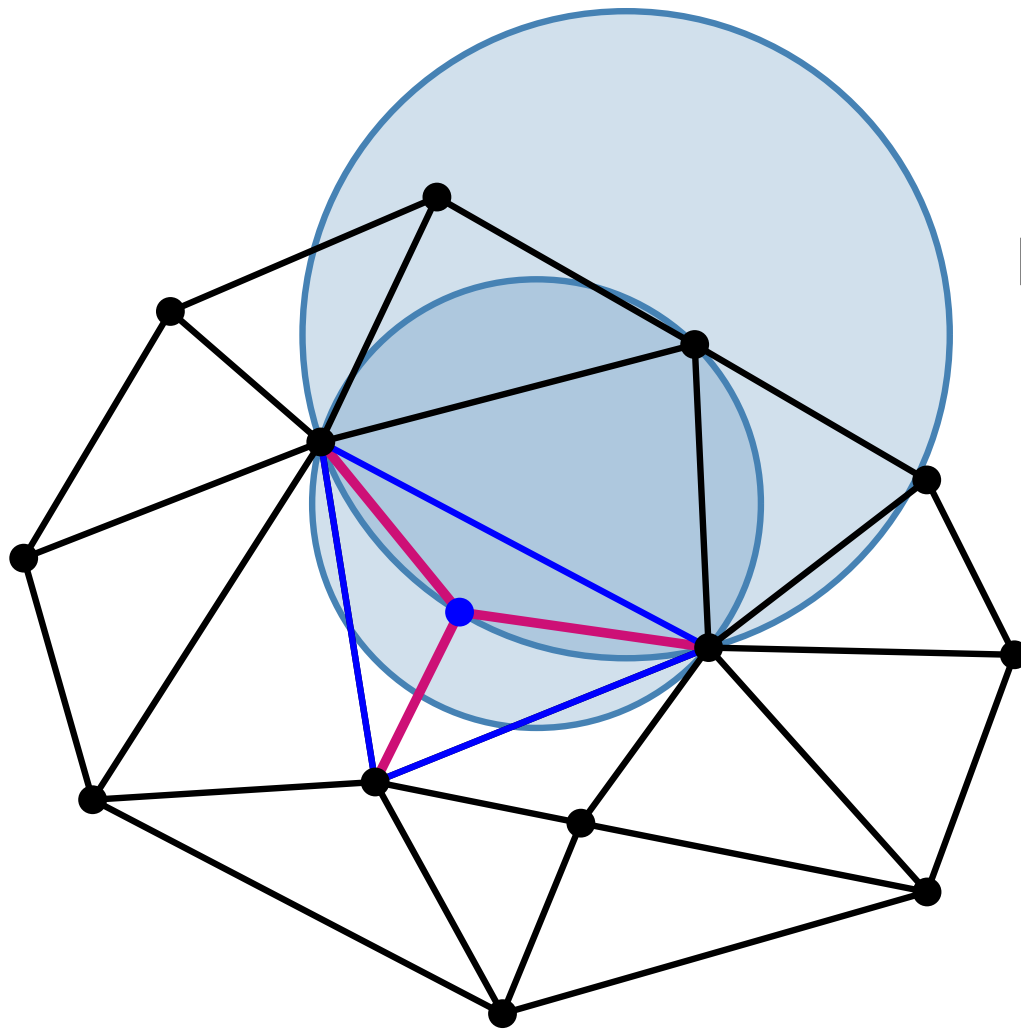
[Lawson 1977]

21 - 1

Monique Teillaud - Triangulations



Back to \mathbb{R}^2



propagate flips

[Lawson 1977]

21 - 2

Monique Teillaud - Triangulations

A stronger result

n points in \mathbb{R}^2

The flip graph of triangulations is connected

Triangulation T $\xrightarrow{O(n^2) \text{ flips}}$ Delaunay triangulation

[Hurtado *et al.* DCG'99]

A stronger result

n points on a surface

the edges of the equivalent geodesic triangulation do not intersect

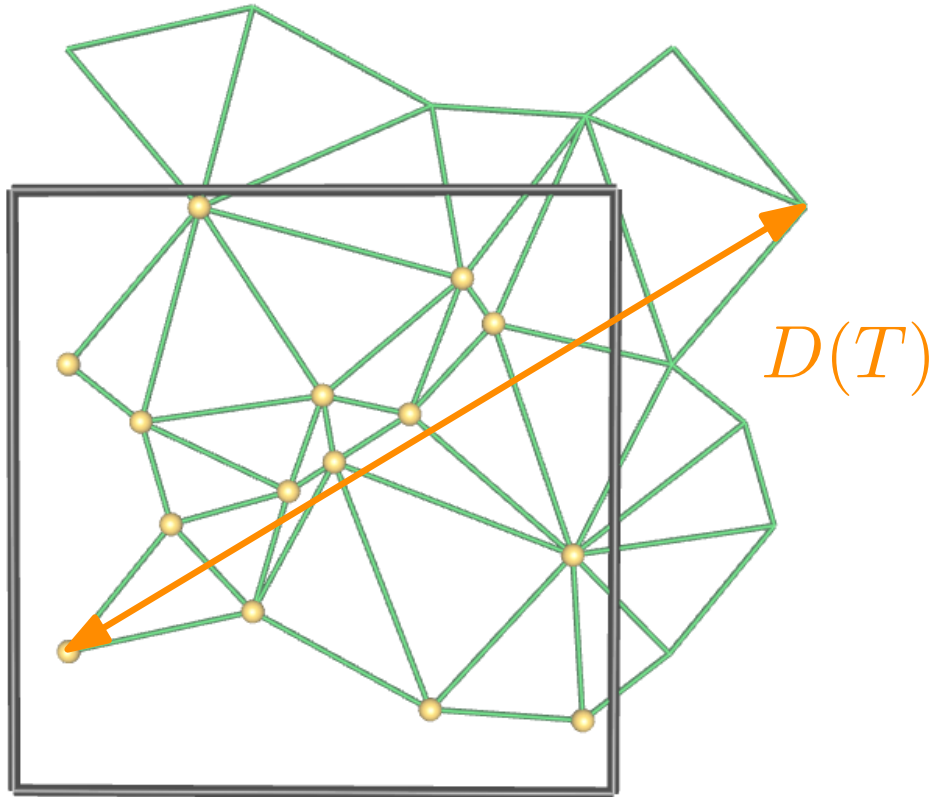
The flip graph of geometric triangulations is connected

Triangulation T $\xrightarrow{O(f(g, T) n^2) \text{ flips}}$ Delaunay triangulation

The Delaunay triangulation is geometric

[Despré Schlenker T. SoCG'20]

Diameter



smallest diameter
of a connected
union of lifts
 $\tilde{t} \in \pi^{-1}(t), t \in T$

Ω fundamental domain

Diameter

A flip can never create an edge longer than $2D(T)$

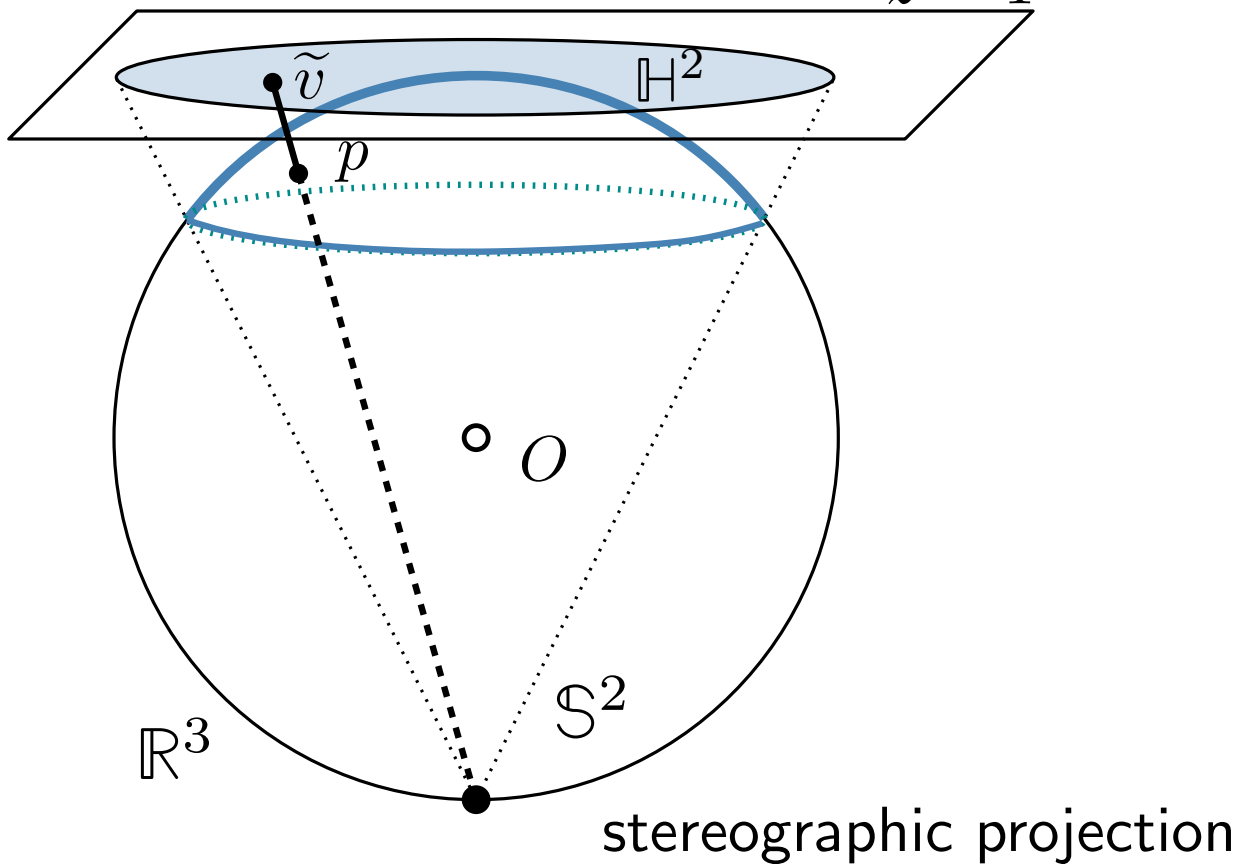
Proof

Diameter

A flip can never create an edge longer than $2D(T)$

Proof

v on the surface, $\tilde{v} \in \pi^{-1}(v)$ $z = 1$



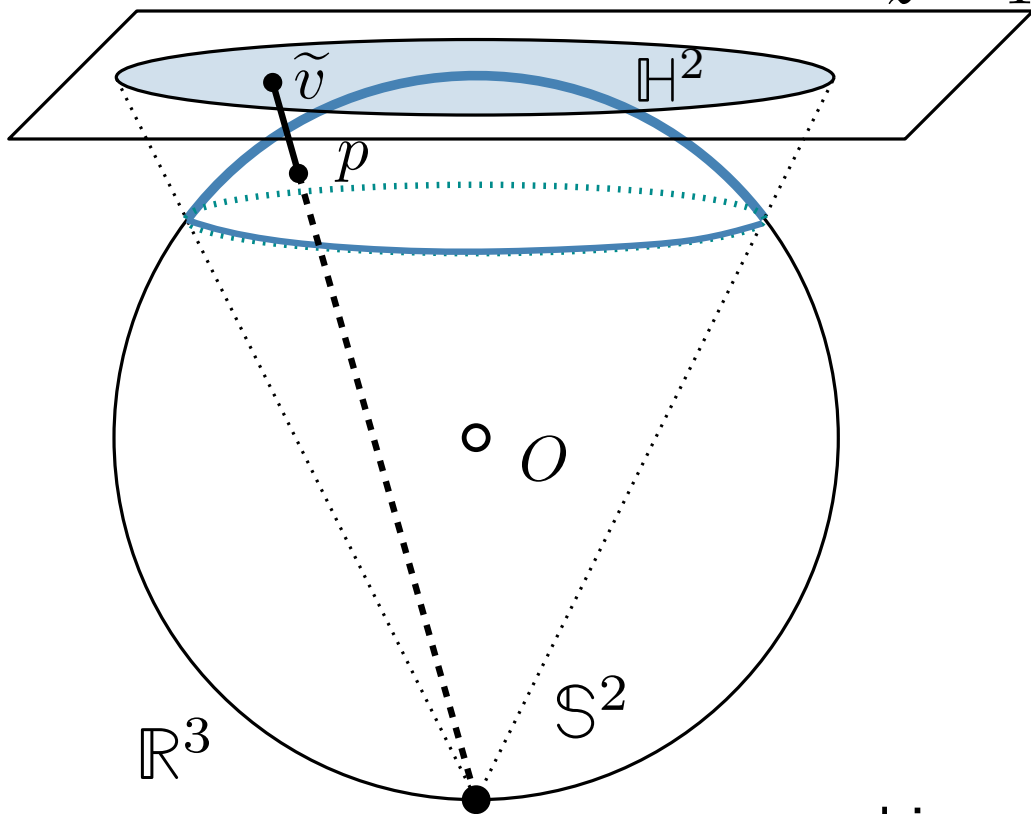
[Despré Schlenker T. SoCG'20]

Diameter

A flip can never create an edge longer than $2D(T)$

Proof

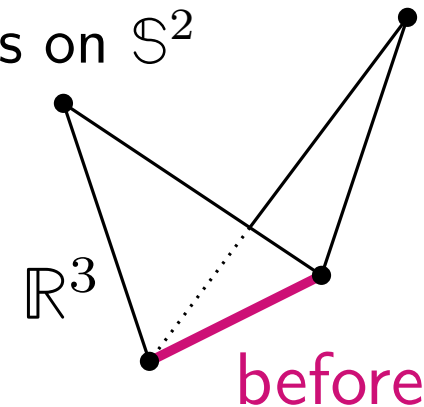
v on the surface, $\tilde{v} \in \pi^{-1}(v)$ $z = 1$



stereographic projection

flip in \mathbb{H}^2

4 points on S^2



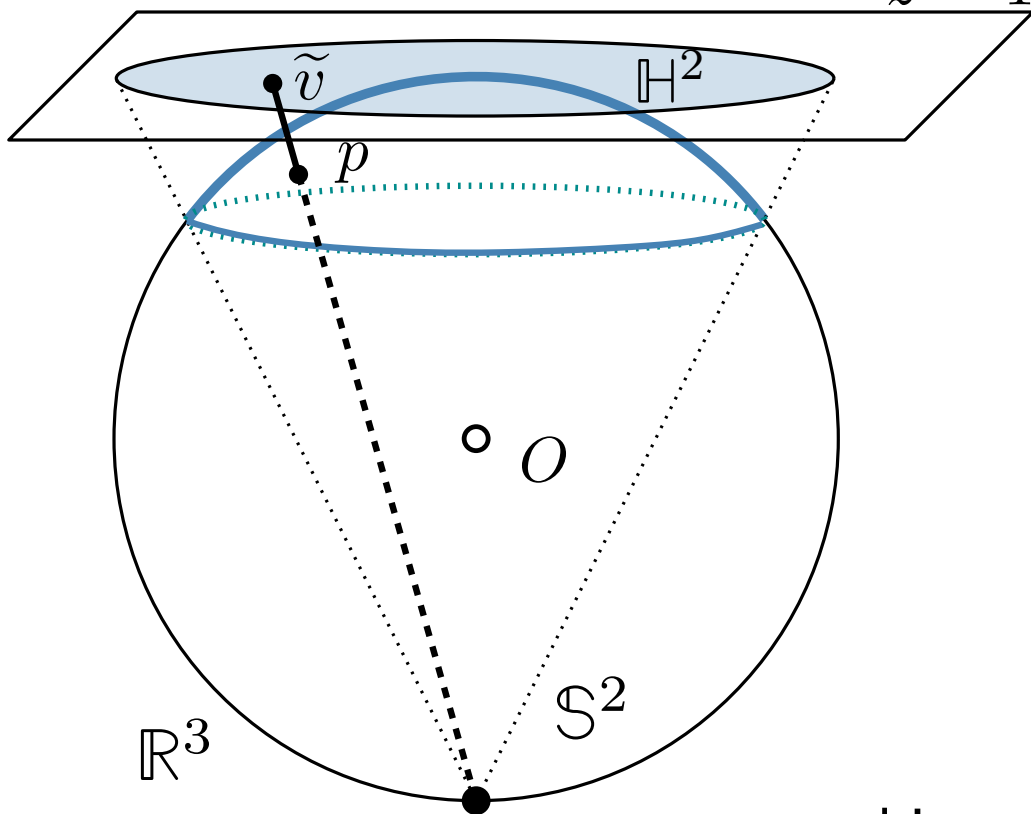
[Despré Schlenker T. SoCG'20]

Diameter

A flip can never create an edge longer than $2D(T)$

Proof

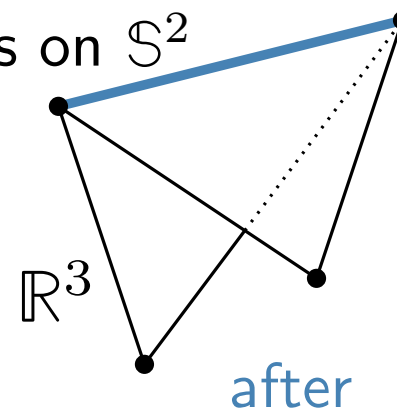
v on the surface, $\tilde{v} \in \pi^{-1}(v)$ $z = 1$



stereographic projection

flip in \mathbb{H}^2

4 points on S^2



Delaunay \leftrightarrow convex polyhedron
an edge can never reappear

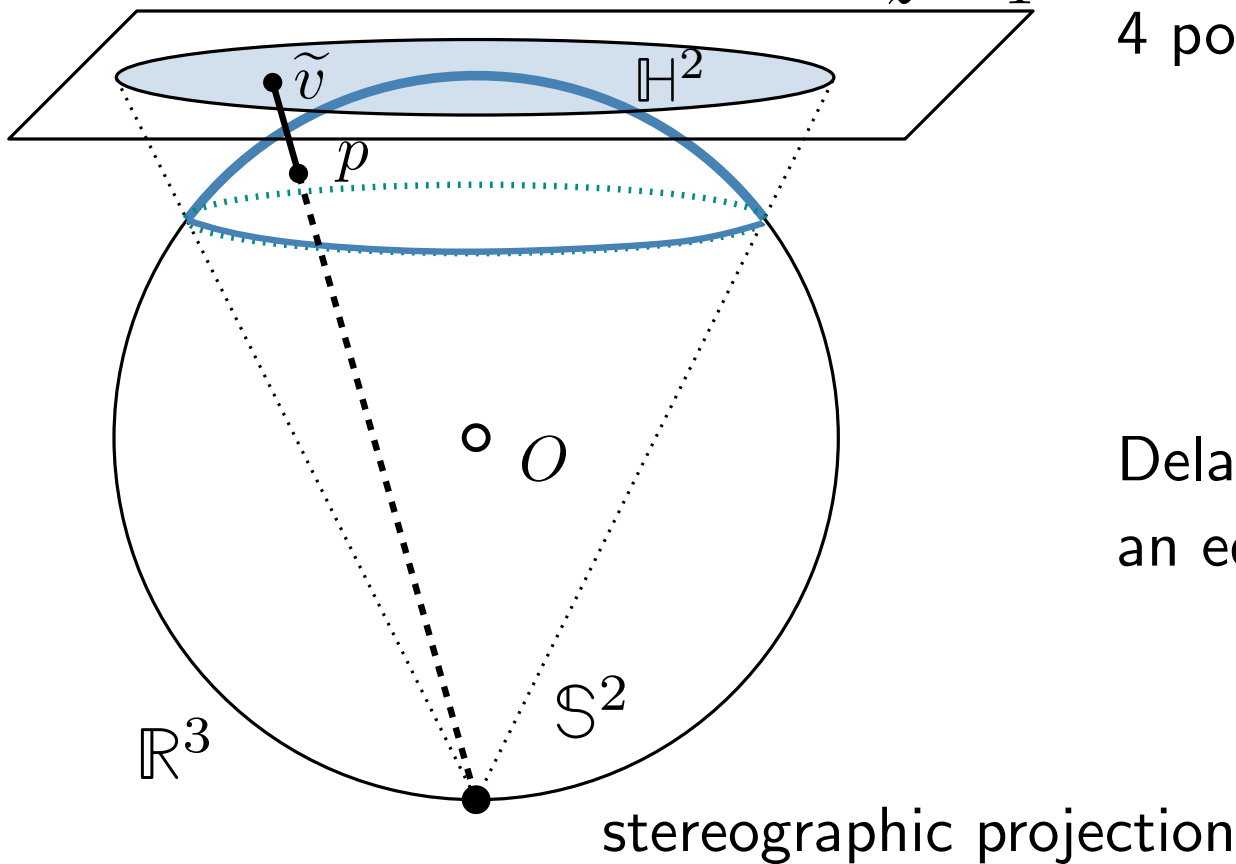
[Despré Schlenker T. SoCG'20]

Diameter

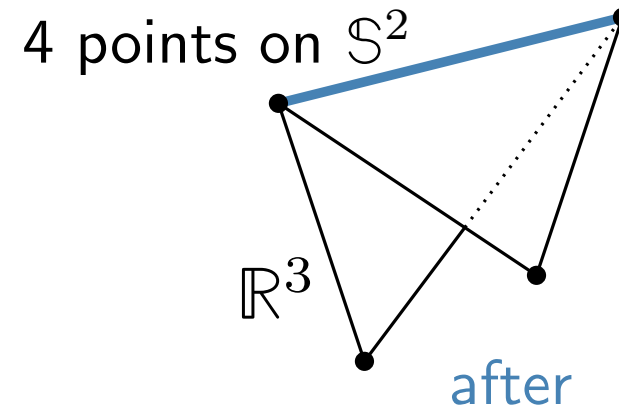
A flip can never create an edge longer than $2D(T)$

Proof

v on the surface, $\tilde{v} \in \pi^{-1}(v)$ $z = 1$



flip in \mathbb{H}^2



Delaunay \leftrightarrow convex polyhedron
an edge can never reappear

sequence of flips

\rightsquigarrow sequence of surfaces

$$\Sigma_0 \subset \dots \subset \Sigma_k \subset \dots$$

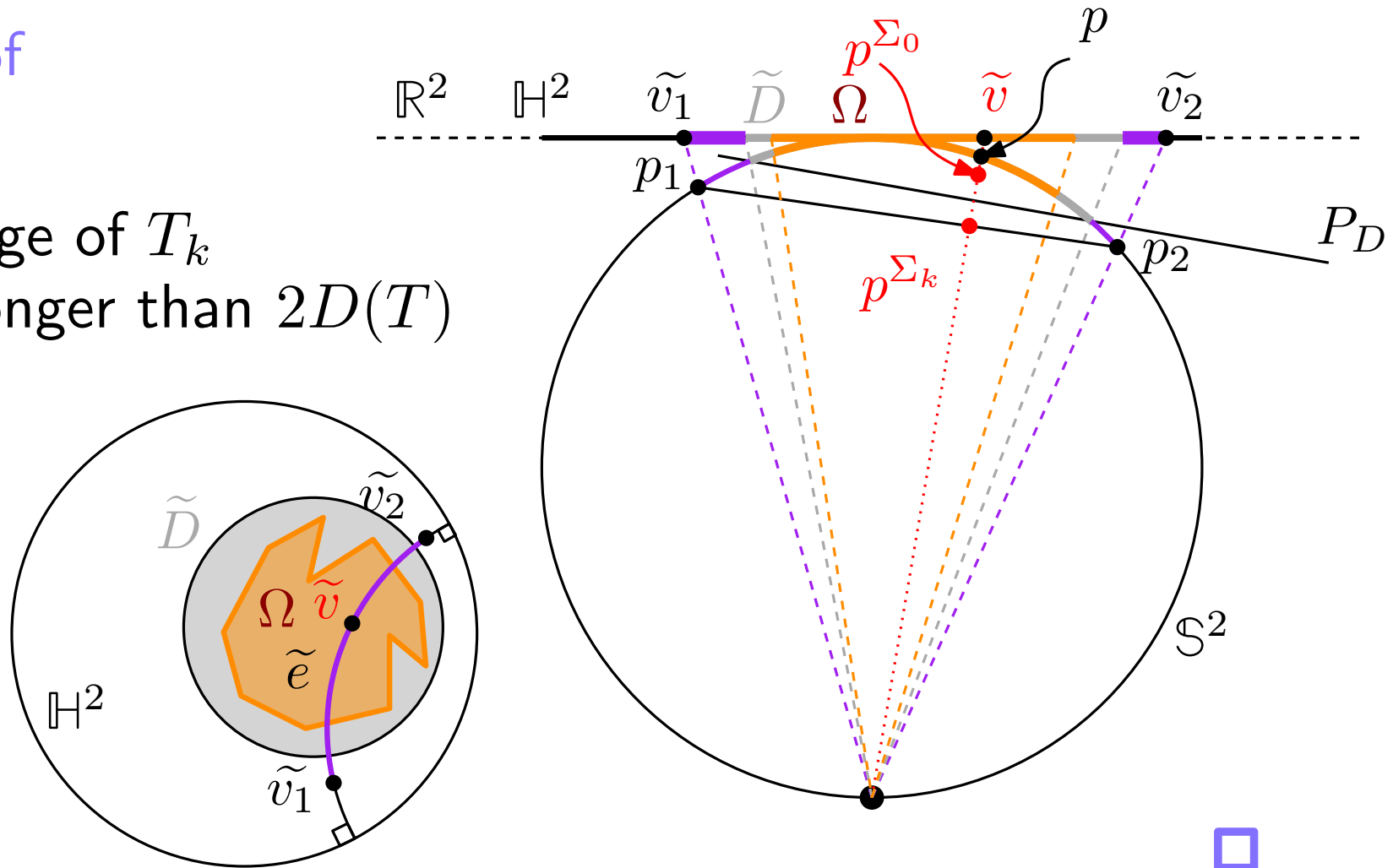
[Despré Schlenker T. SoCG'20]

Diameter

A flip can never create an edge longer than $2D(T)$

Proof

\tilde{e} edge of T_k
longer than $2D(T)$



[Despré Schlenker T. SoCG'20]

Flat torus

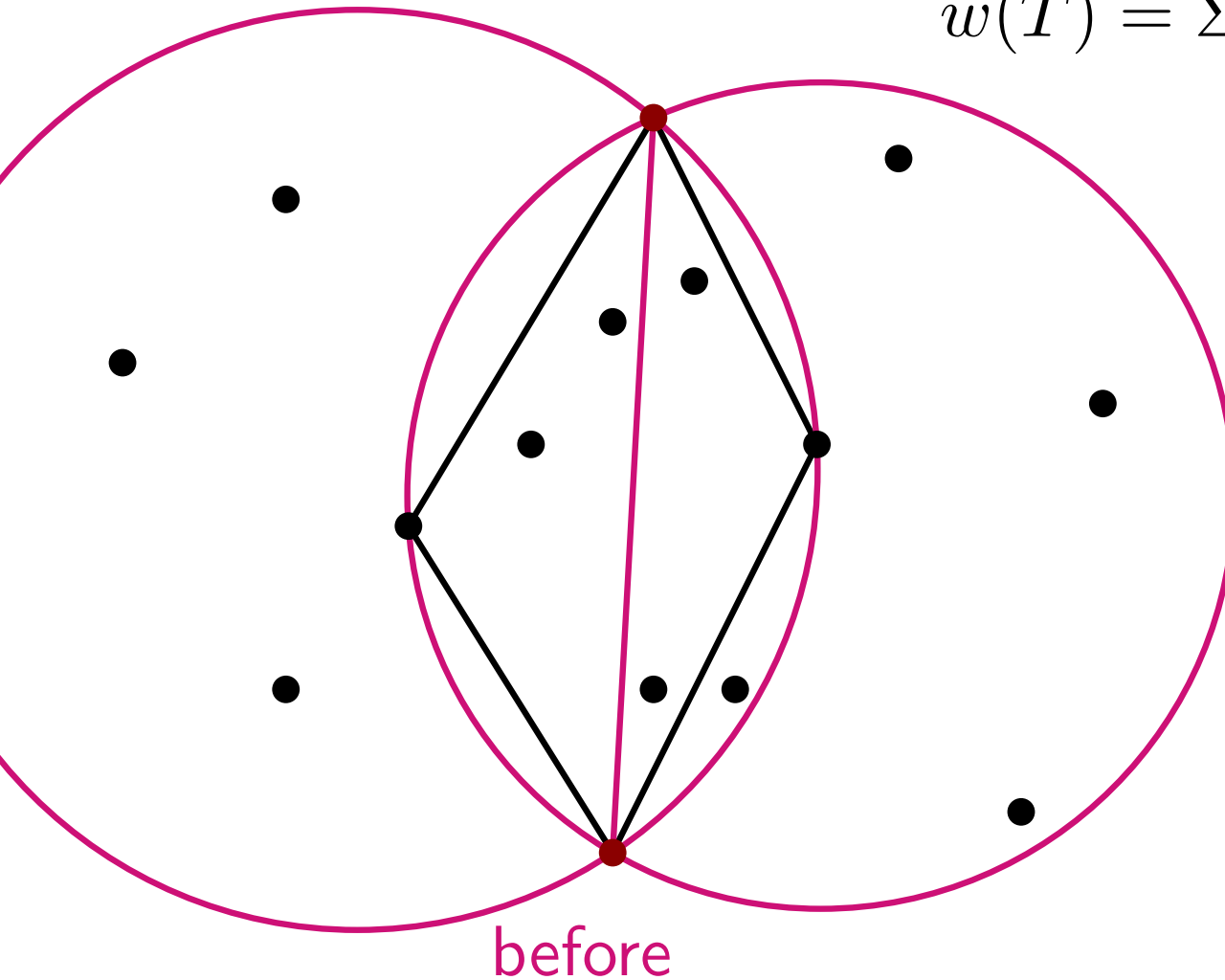
Connectivity

Proof

in \mathbb{R}^2

$$w(T) = \sum_{t \in T} (\text{number of points in } \text{open_disk}(t))$$

all disks are compact



[Despré Schlenker T. SoCG'20]

Flat torus

Connectivity

Proof

in \mathbb{R}^2

$$w(T) = \sum_{t \in T} (\text{number of points in } \text{open_disk}(t))$$

all disks are compact

$$w(T') \leq w(T) - 2$$

\implies termination

\implies connectivity

after



[Despré Schlenker T. SoCG'20]

\mathbb{T}_u , n points, triangulation T

$C_u \cdot D(T) \cdot n^2$ flips to the Delaunay triangulation

[Dubois CCCG'22]

Hyperbolic surface

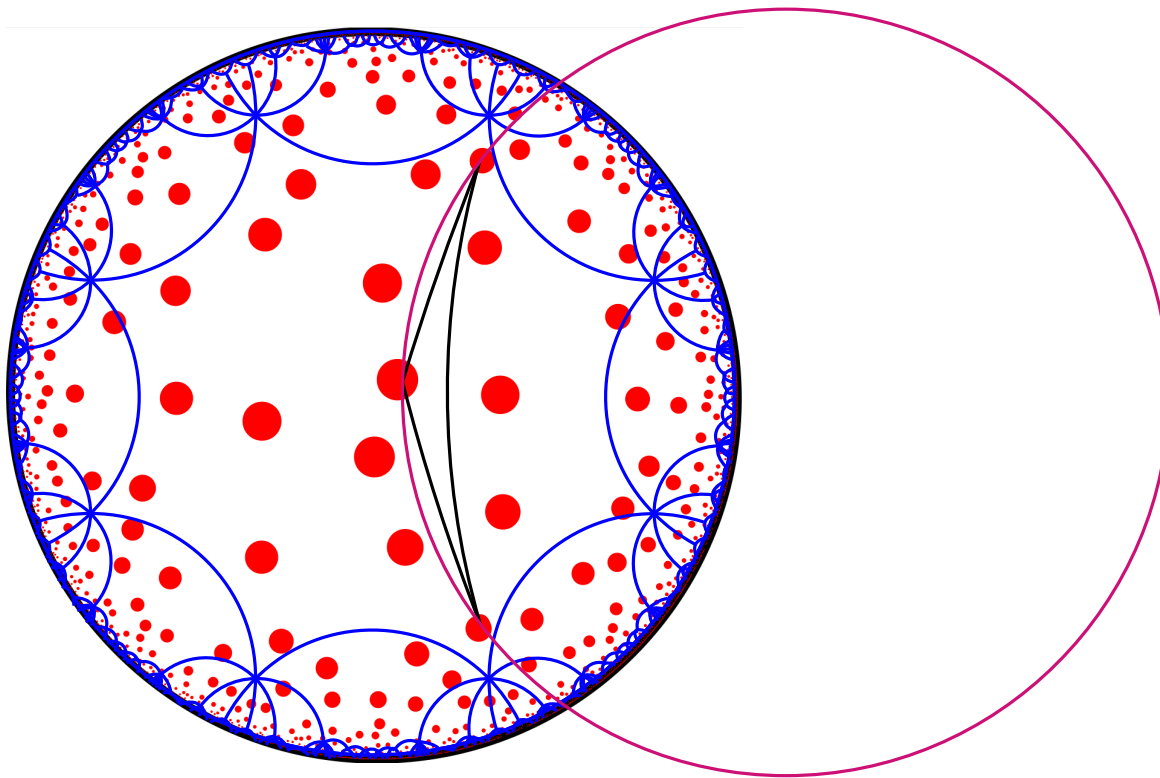
Connectivity

Proof

in \mathbb{H}^2

$$w(T) = \sum_{t \in T} (\text{number of points in disk}(t))$$

disks not always compact



[Despré Schlenker T. SoCG'20]

Hyperbolic surface

Connectivity

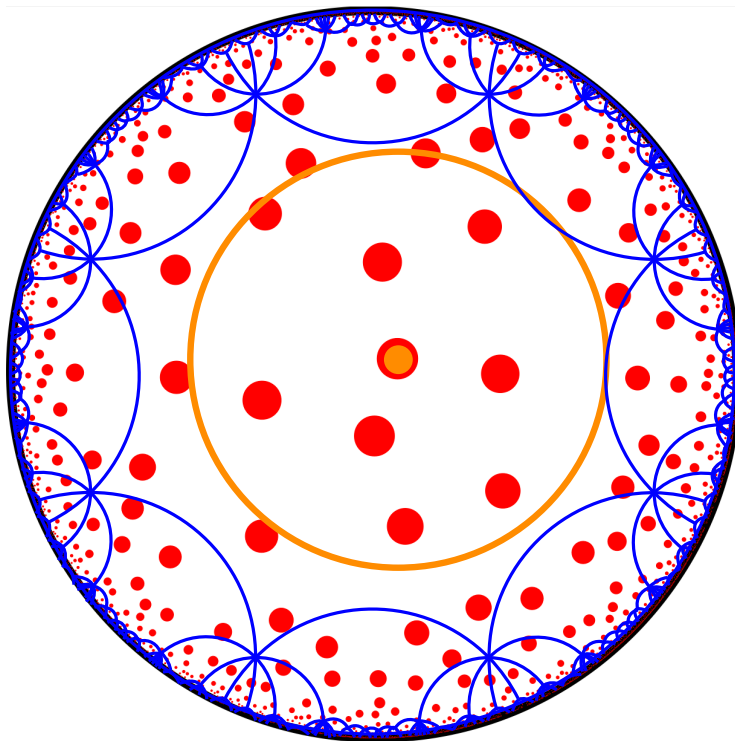
Proof

in \mathbb{H}^2

$$w(T) = \sum_{t \in T} (\text{number of points in disk}(t))$$

disks not always compact

edges are shorter than $2D(T)$



compact disk

(but area is exponential)

finite number of possible edges

\implies termination

\implies connectivity



[Despré Schlenker T. SoCG'20]

Hyperbolic surface

Complexity

surface S , n points, triangulation T
 $\leq C_{metric} \cdot D(T)^{6g-4} \cdot n^2$ flips to the Delaunay triangulation

Hyperbolic surface

Complexity

surface S , n points, triangulation T
 $\leq C_{metric} \cdot D(T)^{6g-4} \cdot n^2$ flips to the Delaunay triangulation

Sketch of proof

count non-homotopic simple paths

known: closed curves

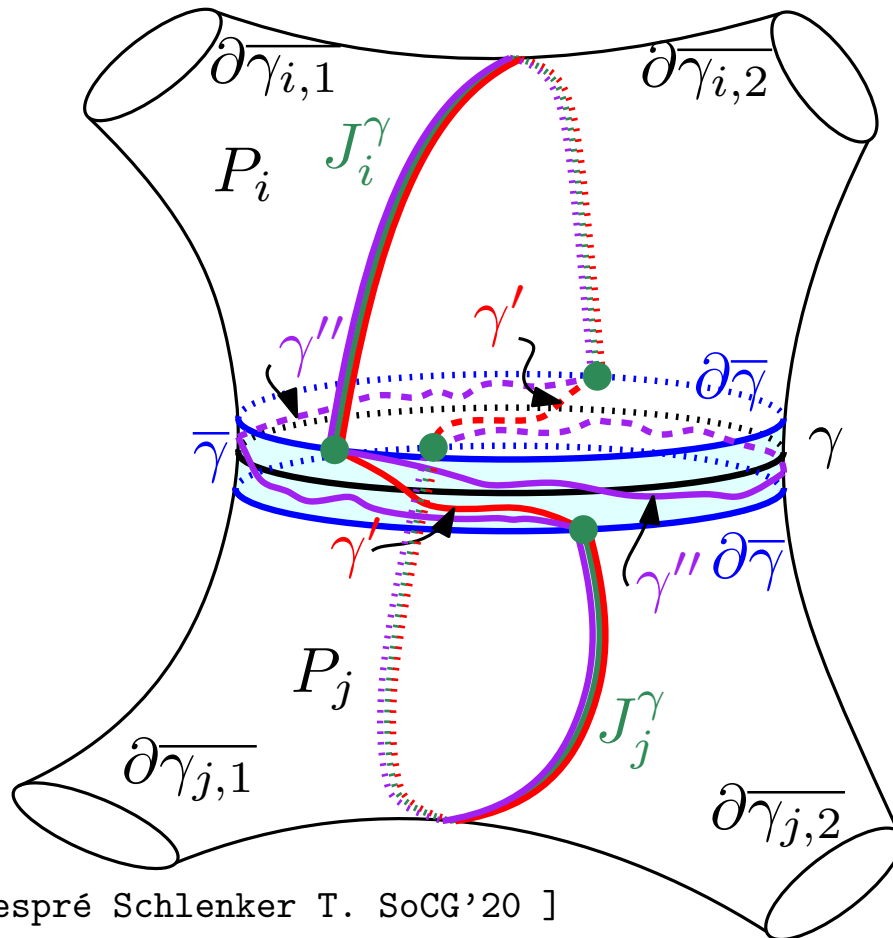
[Fathi Laudenbach Poénaru SMF'79]

Hyperbolic surface

Complexity

surface S , n points, triangulation T
 $\leq C_{metric} \cdot D(T)^{6g-4} \cdot n^2$ flips to the Delaunay triangulation

Sketch of proof



count non-homotopic simple paths

known: closed curves

[Fathi Laudenbach Poénaru SMF'79]

path $f = (v_1, v_2)$

normal form homotopic to f

$f = (v_1, w_1) \cup (w_1, w_2) \cup (w_2, v_2)$

w_i on annulus boundary

“push” twists in annuli

“normal” coordinates

w.r.t. short pants

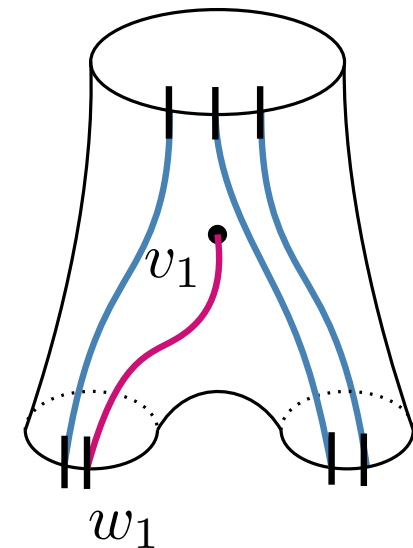
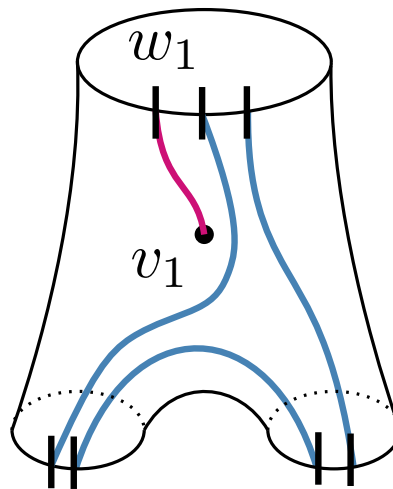
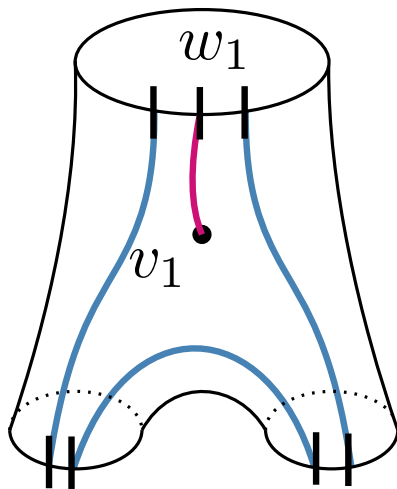
[Despré Schlenker T. SoCG'20]

Hyperbolic surface

Complexity

surface S , n points, triangulation T
 $\leq C_{metric} \cdot D(T)^{6g-4} \cdot n^2$ flips to the Delaunay triangulation

Sketch of proof



bound on length

$= 6g - 4$ Dehn Thurston coordinates

count possible coordinates and possible paths



[Despré Schlenker T. SoCG'20]

Hyperbolic surface

Complexity

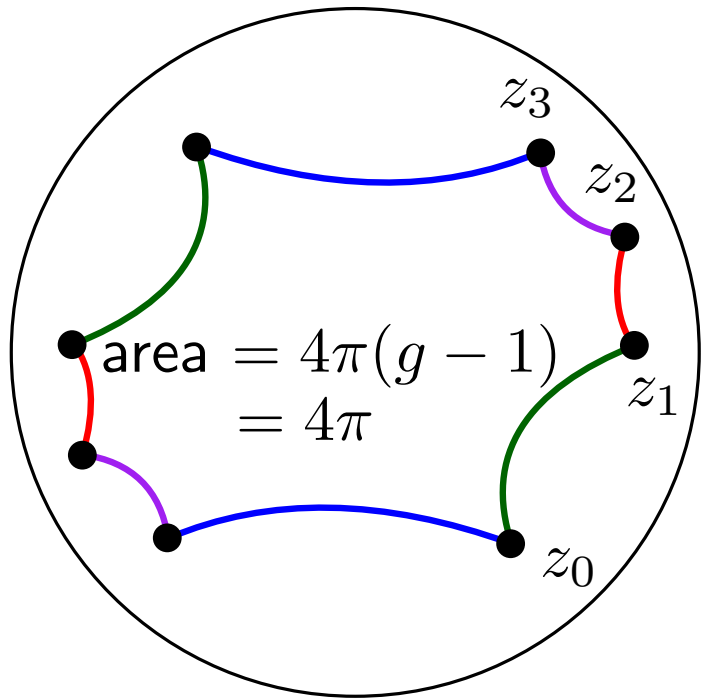
surface S , n points, triangulation T
 $\leq C_{metric} \cdot D(T)^{6g-4} \cdot n^2$ flips to the Delaunay triangulation

Is the bound (close to) tight?

Experimental analysis

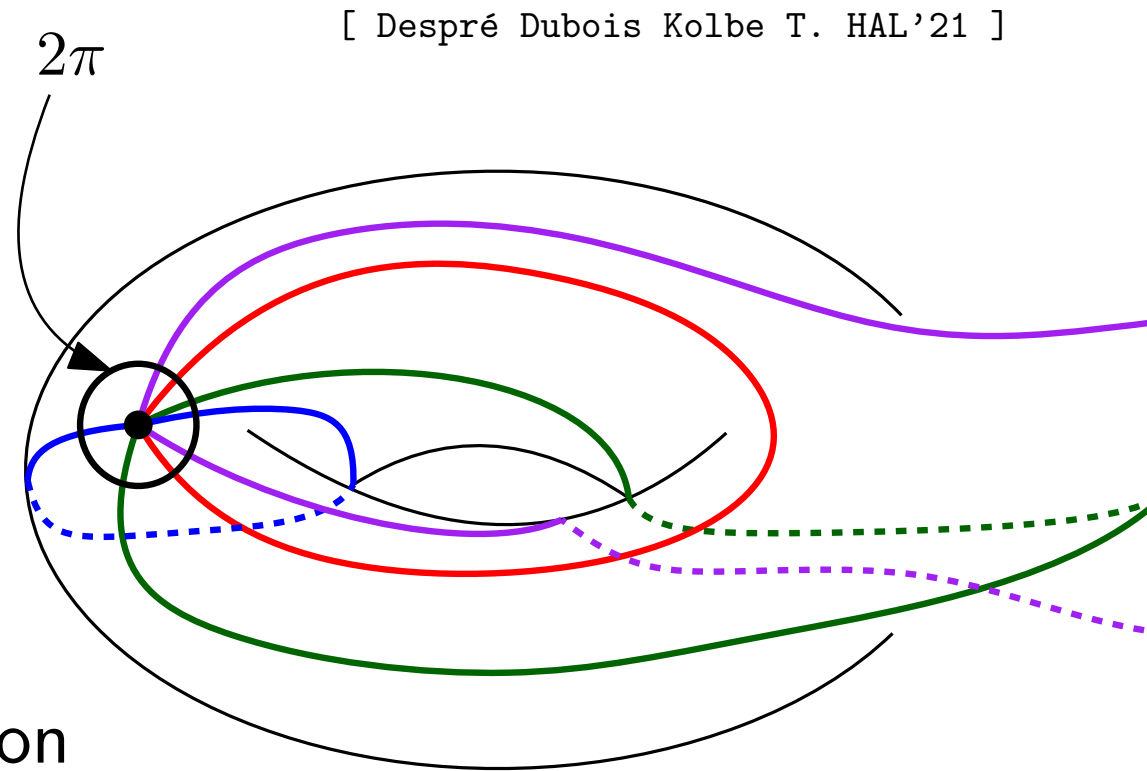
Representation

genus 2 surface



admissible symmetric octagon

[Aigon-Dupuy, Buser, etc Math. Phys. '05]

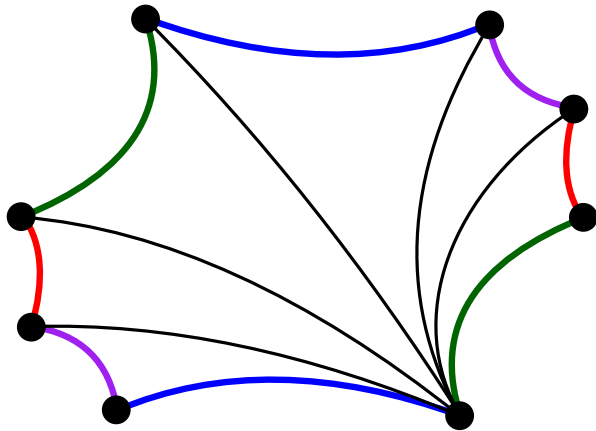


[Despré Dubois Kolbe T. HAL'21]

Representation

genus 2 surface

[Despré Dubois Kolbe T. HAL'21]

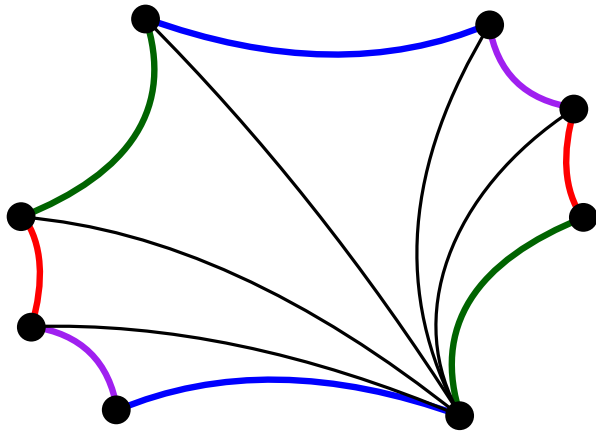


triangulation

Representation

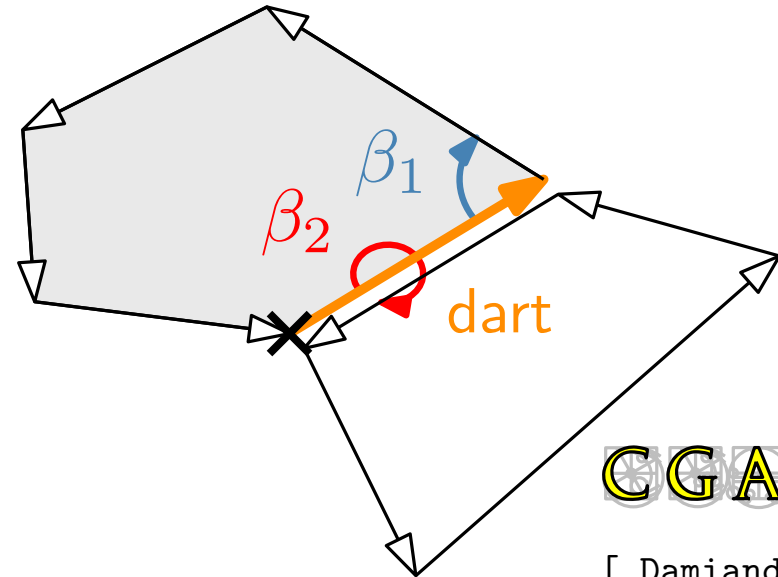
genus 2 surface

[Despré Dubois Kolbe T. HAL'21]



triangulation

combinatorial map

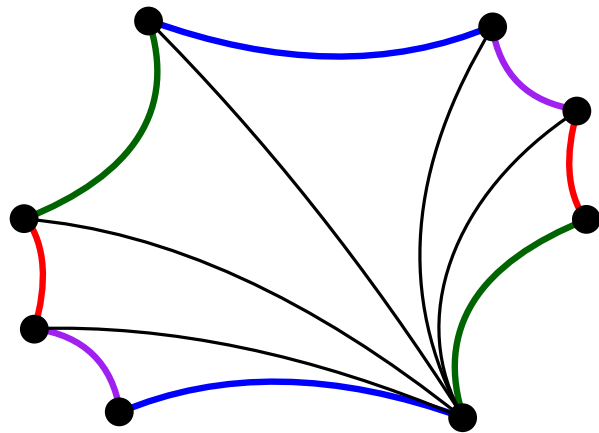


[Damiand]

Representation

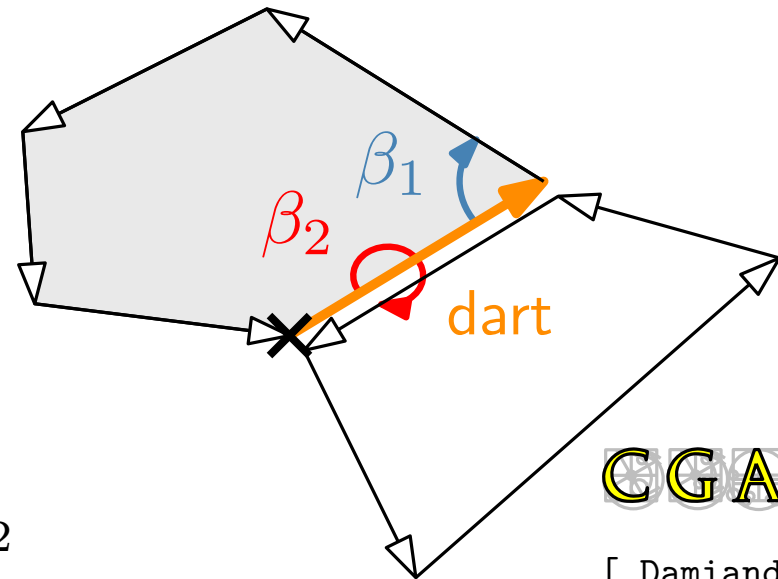
genus 2 surface

[Despré Dubois Kolbe T. HAL'21]



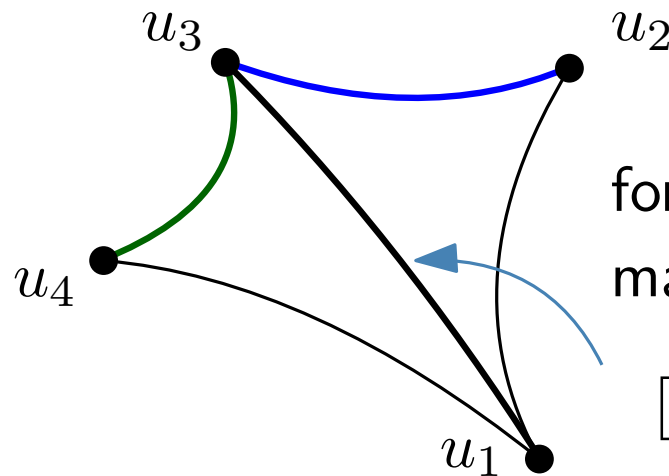
triangulation

combinatorial map



CGAL

[Damiand]



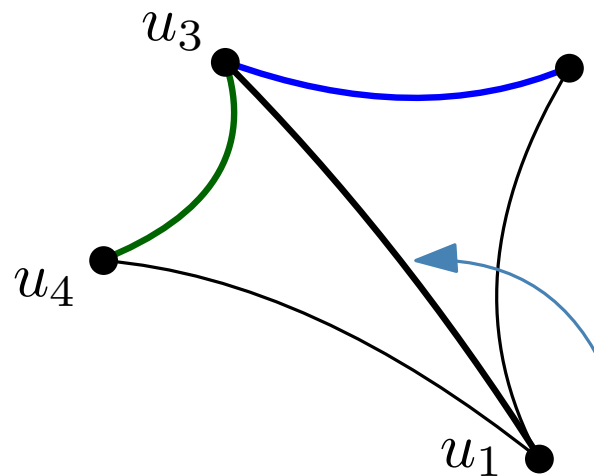
forget points
maintain **cross-ratios**

$$[u_1, u_2, u_3, u_4] = \frac{(u_4 - u_2)(u_3 - u_1)}{(u_4 - u_1)(u_3 - u_2)}$$

[Despré Dubois Kolbe T. HAL'21]

Practical issue

high degree algebraic numbers



forget points
maintain **cross-ratios**

$$[u_1, u_2, u_3, u_4] = \frac{(u_4 - u_2)(u_3 - u_1)}{(u_4 - u_1)(u_3 - u_2)}$$

[Despré Dubois Kolbe T. HAL'21]

Practical issue

high degree algebraic numbers

Density result

Given: admissible (z_0, z_1, z_2, z_3) and $\varepsilon > 0$

\exists rational admissible $(\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3)$
 $\forall k, \mathbf{z}_k \in B(z_k, \varepsilon)$

Dependence on diameter

surface S , n points, triangulation T [Despré Dubois Kolbe T. HAL'21]
 $\leq C_{metric} \cdot D(T)^{6g-4} \cdot n^2$ flips to the Delaunay triangulation

need triangulations with large diameter

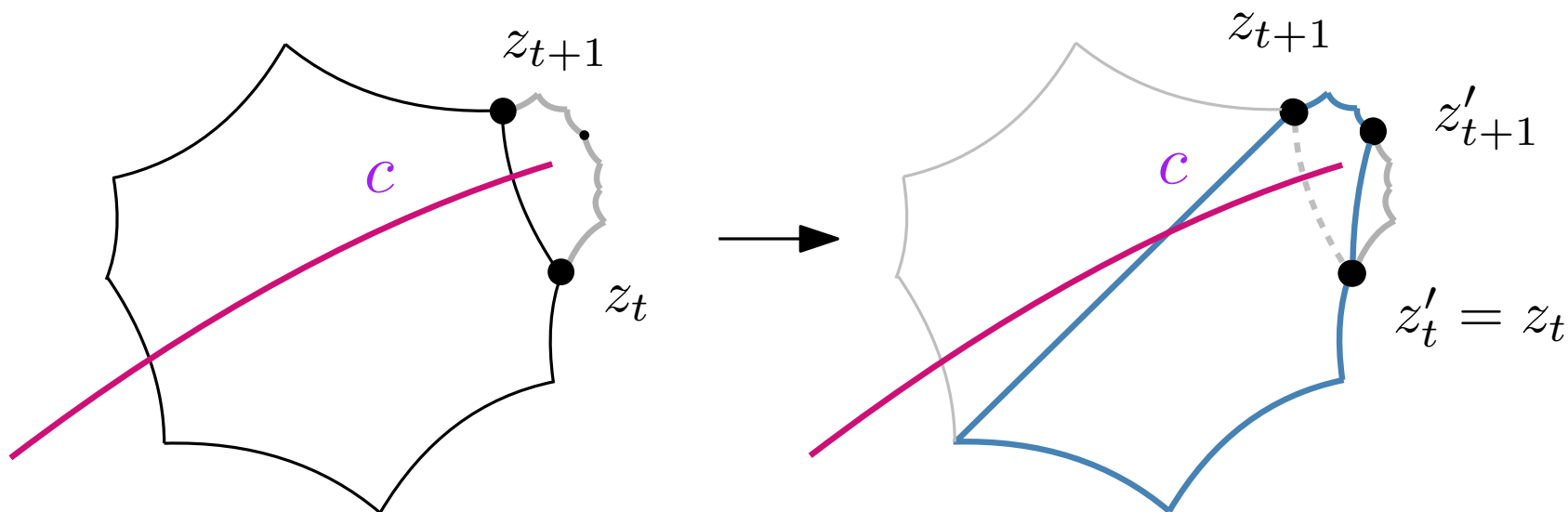
 use Dehn twists

Dependence on diameter

surface S , n points, triangulation T [Despré Dubois Kolbe T. HAL'21]
 $\leq C_{metric} \cdot D(T)^{6g-4} \cdot n^2$ flips to the Delaunay triangulation

need triangulations with large diameter

use Dehn twists along side pairings



same surface, larger diameter

Experiments

[Despré Dubois Kolbe T. HAL'21]

2 types

Power sequences of twists

Random sequences of twists

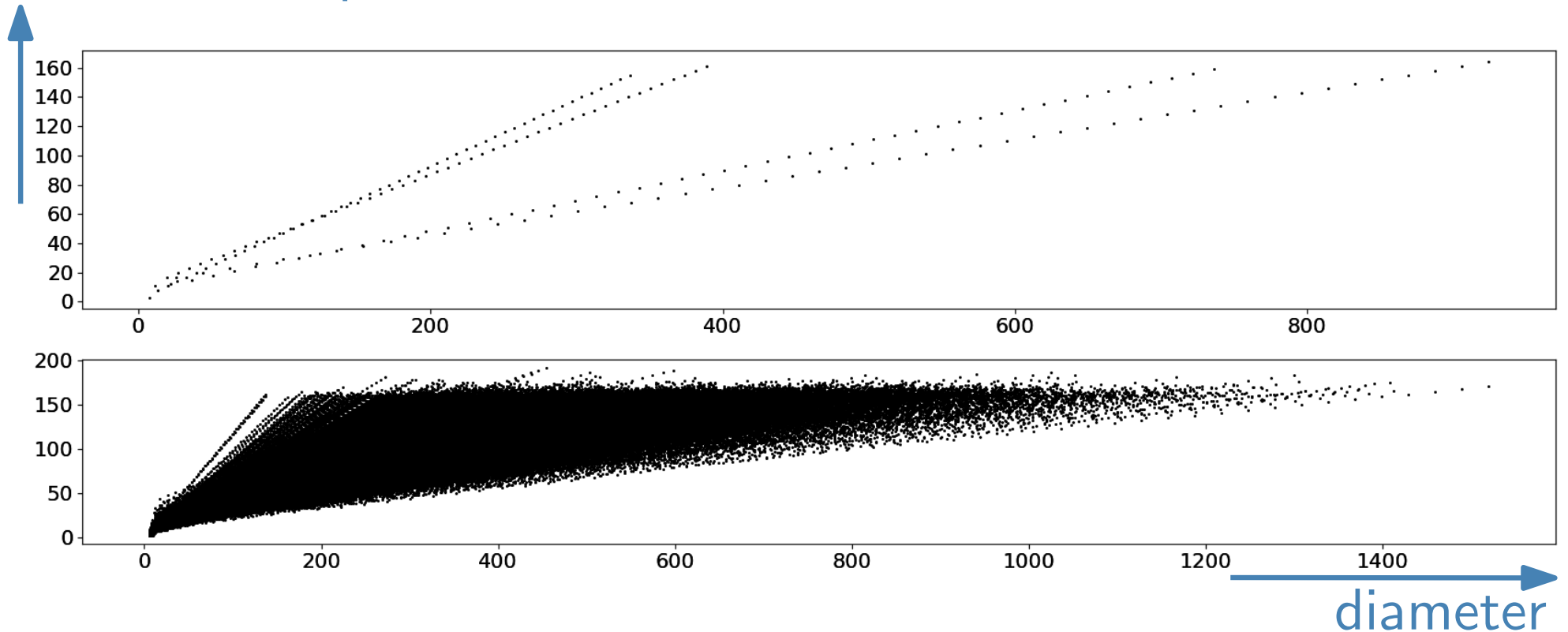
on a large number of octagons

Experiments

power

[Despré Dubois Kolbe T. HAL'21]

number of flips



linear behavior:

flips untwist the triangulation

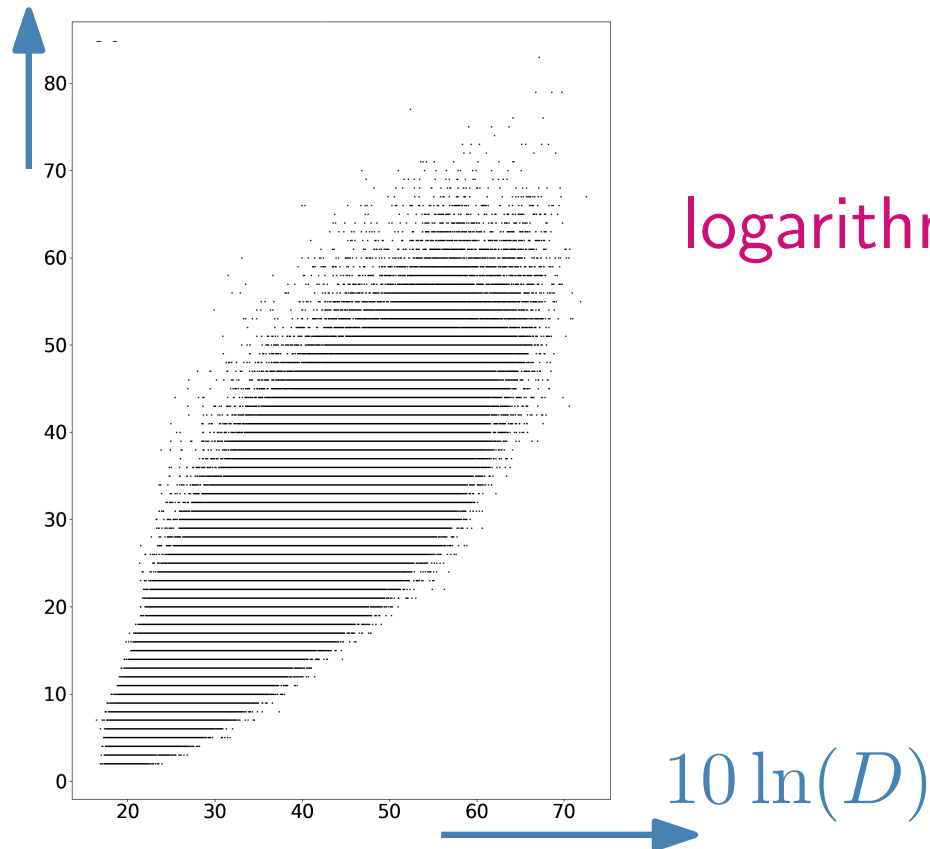
by a constant number of flips per iteration of the twist

Experiments

random

[Despré Dubois Kolbe T. HAL'21]

number of flips



logarithmic behavior

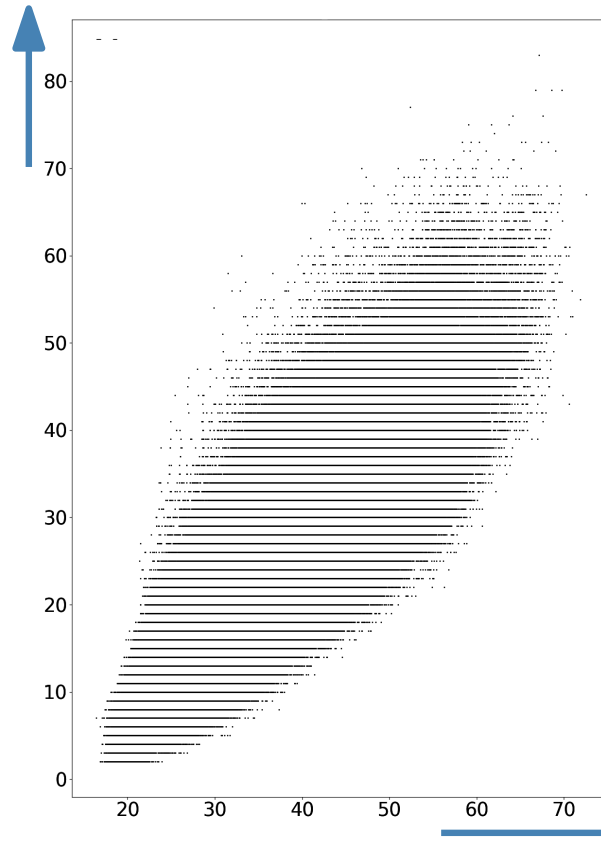
maximum diameter $\simeq 1500$

Experiments

random

[Despré Dubois Kolbe T. HAL'21]

number of flips



logarithmic behavior

“looks like” M. Bell’s results
(topological triangulations)

$10 \ln(D)$

maximum diameter $\simeq 1500$

Questions

S_g topological surface

$Mod(S_g)$ = mapping class group

= set of homeomorphisms (up to isotopy) of S_g

Questions

S_g topological surface

$Mod(S_g)$ = mapping class group

= set of homeomorphisms (up to isotopy) of S_g

[Disarlo Parlier T.AMS'19]

Flip graph quasi isometric to $Mod(S_g)$

Questions

S_g topological surface

$Mod(S_g) =$ mapping class group

$=$ set of homeomorphisms (up to isotopy) of S_g

[Thurston]

reducible: fix at least one curve

pseudo-Anosov (hyperbolic)

periodic (finite order)

Questions

S_g topological surface

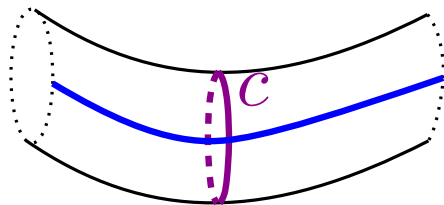
$Mod(S_g) =$ mapping class group

$=$ set of homeomorphisms (up to isotopy) of S_g

[Thurston]

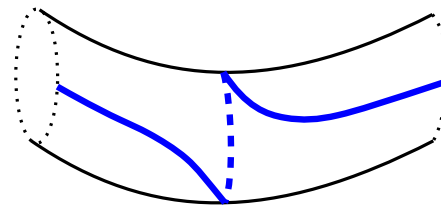
reducible: fix at least one curve

Dehn twists



asymptotically

add a constant to the length



pseudo-Anosov (hyperbolic)

multiply the length by a constant

periodic (finite order)

Questions

S_g topological surface

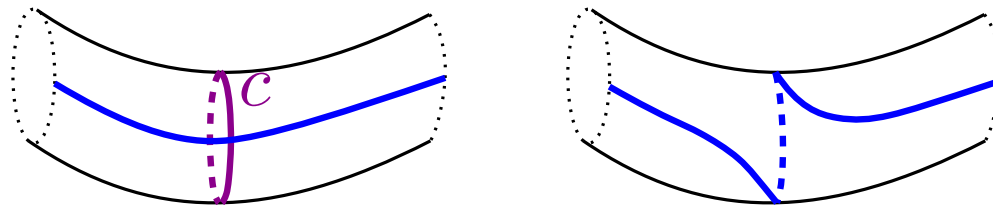
$Mod(S_g)$ = mapping class group

= set of homeomorphisms (up to isotopy) of S_g

$Mod(S_g)$ generated by a finite set of

Dehn twists

[Farb Margalit '12]



random composition  random walk in $Mod(S_g)$

pseudo-Anosov with asymptotic probability 1

[Maher Duke '11]

Questions

S_g topological surface

$Mod(S_g) =$ mapping class group

$=$ set of homeomorphisms (up to isotopy) of S_g

[Bell DCG'21]

effect on the flip graph of topological triangulations

a pseudo-Anosov or a power of a Dehn twist

reduces the **number of crossings** with a **fixed** curve

different problem here

unknown goal (Delaunay)

how far from the goal?

homotopy class of goal edges?

surface S , n points, triangulation T [Despré Dubois Kolbe T. HAL'21]

flips to the Delaunay triangulation?

On our experiments ($g = 2$)

$$\leq C_{metric} \cdot D(T) \cdot n^2$$

average (?) $\log(D(T))$



surface S , n points, triangulation T [Despré Dubois Kolbe T. HAL'21]

flips to the Delaunay triangulation?

On our experiments ($g = 2$)

$$\leq C_{metric} \cdot D(T) \cdot n^2$$

average (?) $\log(D(T))$

so...?

surface S , n points, triangulation T [Despré Dubois Kolbe T. HAL'21]

flips to the Delaunay triangulation?

On our experiments ($g = 2$)

$$\leq C_{metric} \cdot D(T) \cdot n^2$$

average (?) $\log(D(T))$

so...?

other input surfaces ?

Thank you !

(Slides just in case)

History of the CGAL project

Before CGAL, in Europe

PlaGeo, SpaGeo (Utrecht)

XYZ GeoBench (ETH Zürich)

LEDA (MPII Saarbrücken)

C++GAL
(INRIA Sophia Antipolis)

...



Jodelet/Lépinay - CC Attribution-Share Alike 2.0 France

History of the CCGAL project

1995 - Start: Gather efforts

Academic consortium



History of the project

1995 - Start: Gather efforts

Academic consortium

2003

Open Source project

Double-license: GPLv3+ and commercial
new contributors

Creation of  **Geometry
Factory** *initially as an INRIA startup*

sells commercial licenses and pays royalties to owners
responsible for the infrastructure
and long-term evolutive maintenance

Goals

Promote research in CG

“ make the large body of geometric algorithms developed in the field of computational geometry available for industrial applications ”

⇒ high quality
review process, ...

⇒ robust code

Goals

Promote research in CG

Reward structure for implementations in academia

⇒ high quality
review process, . . .

⇒ robust code

Becoming a user

<http://www.cgal.org>

Many platforms

Linux, MacOS, Windows
g++, VC++, clang, . . .

Becoming a user

<http://www.cgal.org>

Many platforms

Linux, MacOS, Windows
g++, VC++, clang, . . .

Download from `github`

or use

Linux distribution (Debian, . . .)

Mac distribution (macports, brew, . . .)

Euclidean 2D and 3D triangulations

- in Matlab
- in Python through CGAL bindings

Becoming a user

<http://www.cgal.org>

Follow “Getting started”

easier than ever!

- “ Since CGAL version 5.0, CGAL is header-only by default, which means that there is no need to build CGAL before it can be used. ”
- Simple viewer

Becoming a user

<http://www.cgal.org>

Follow “Getting started”

Read the **User manual** of your favorite package

Read, compile and run **examples**

Becoming a user

<http://www.cgal.org>

Follow “Getting started”

Read the **User manual** of your favorite package

Read, compile and run **examples**

Write your own code!

looking at the **Reference manual**

Obey license GPLv3+ (or buy a commercial license)

Becoming a user

<http://www.cgal.org>

Molecular Modeling

Particle Physics, Fluid Dynamics, Microstructures

Medical Modeling and Biophysics

Geographic Information Systems

Games

Motion Planning

Sensor Networks

Architecture, Buildings Modeling, Urban Modeling

Astronomy

2D and 3D Modelers

Mesh Generation and Surface Reconstruction

Geometry Processing

Computer Vision, Image Processing, Photogrammetry

Computational Topology and Shape Matching

Computational Geometry and Geometric Computing

Becoming a user

Ipelets!

ipe.otfried.org/

Becoming a CGAL user

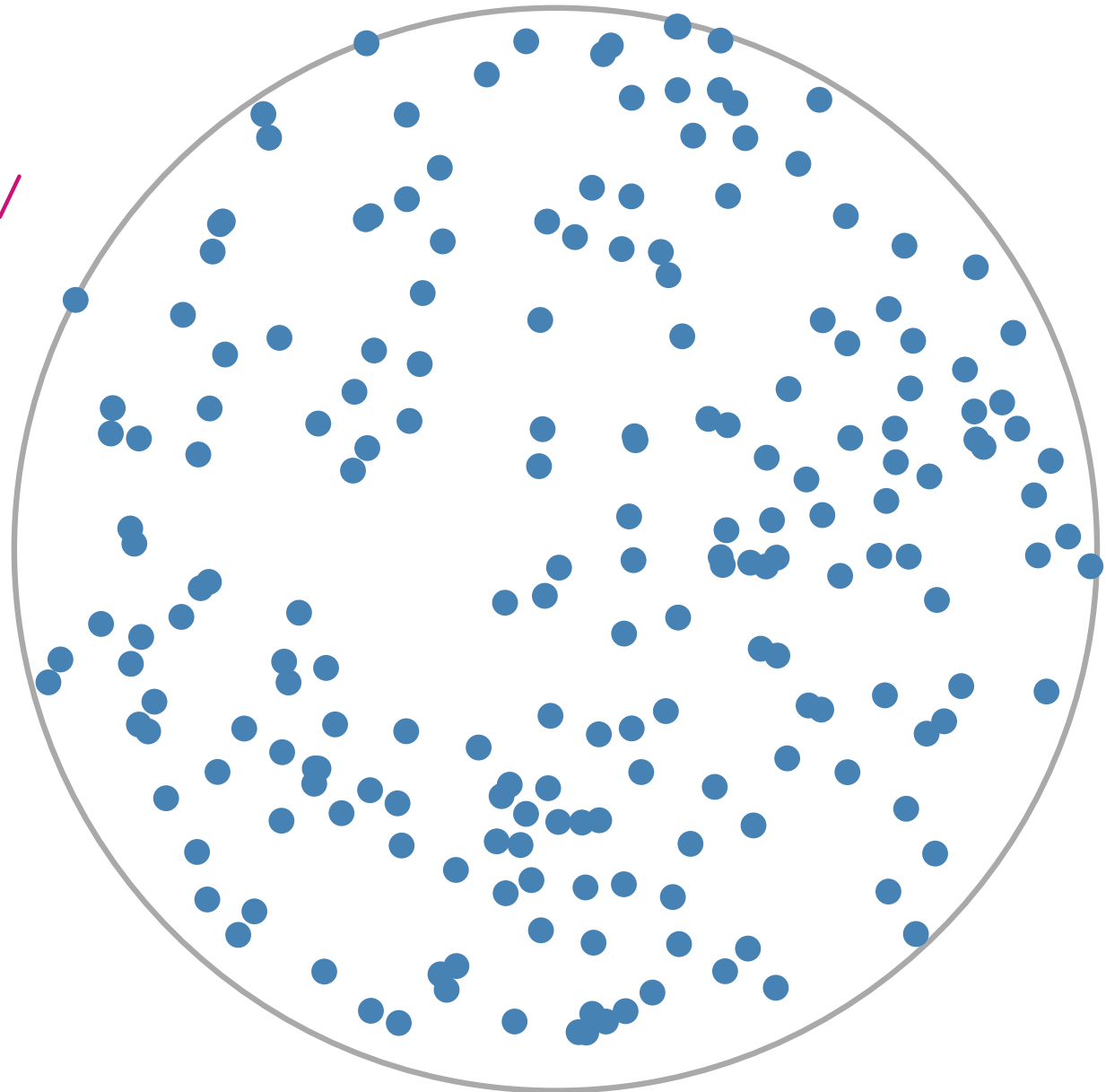
Ipelets!

ipe.otfried.org/

Generators

e.g. 200 points in a disk

\mathbb{R}^2



Becoming a CGAL user

Ipelets!

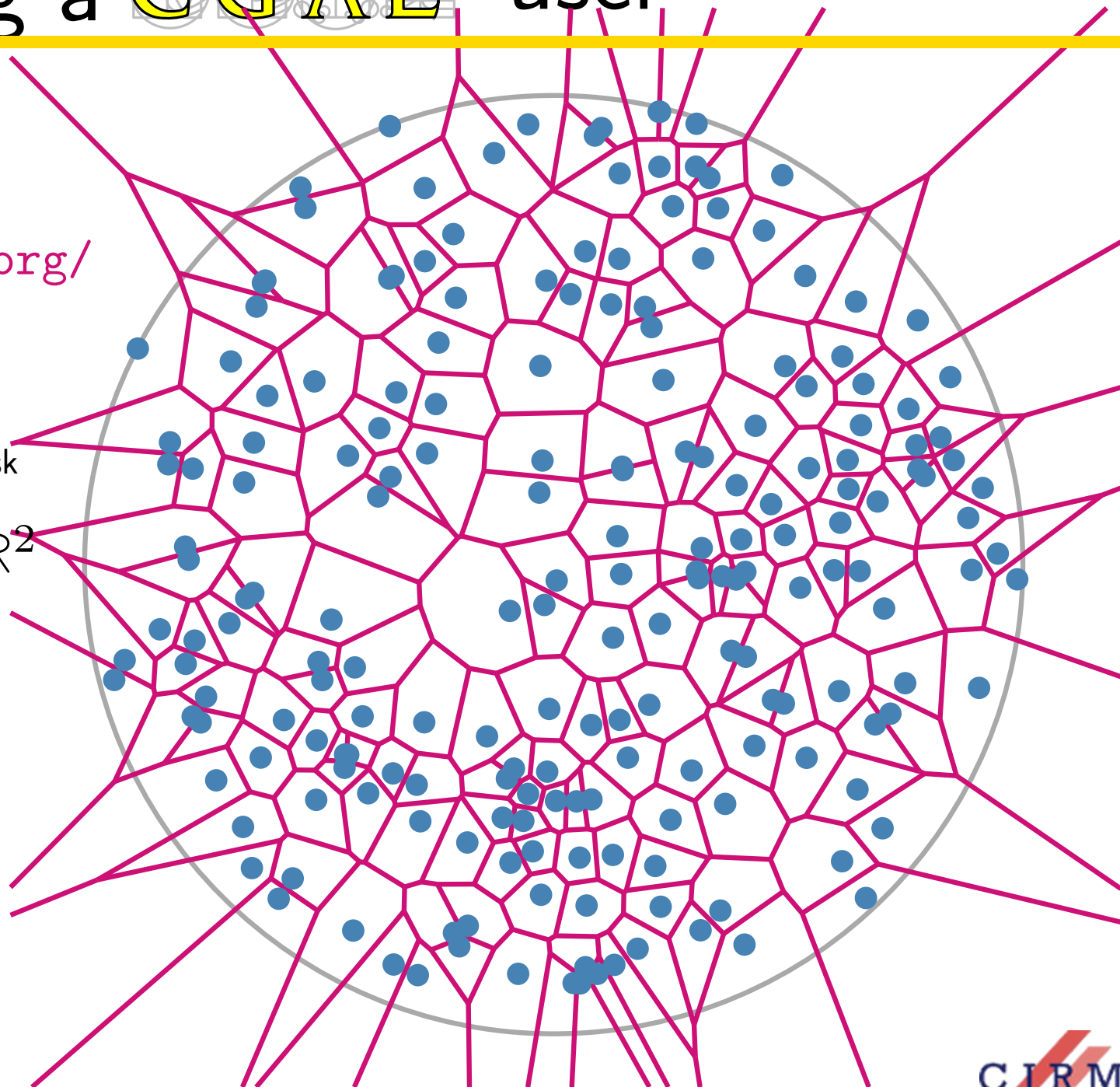
ipe.otfried.org/

Generators

e.g. 200 points in a disk

\mathbb{R}^2

“Diagrams”
→ Voronoi



Becoming a CGAL user

Ipelets!

ipe.otfried.org/

Generators

e.g. 200 points in a disk

\mathbb{R}^2

“Triangulations”
→ Delaunay

