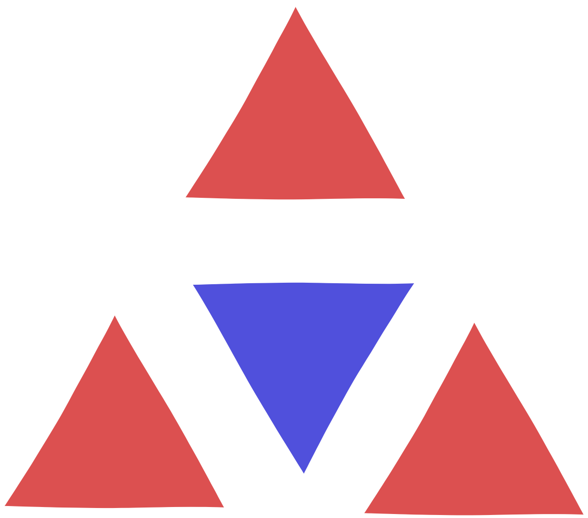
Algorithms for Topological and Metric Surfaces



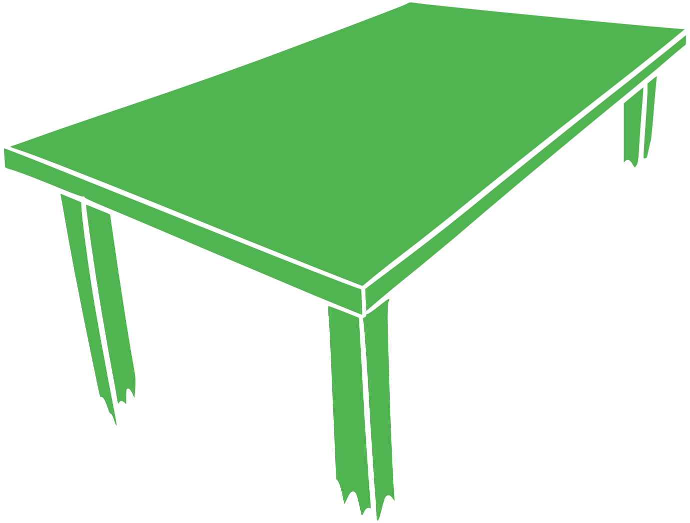
Loïc Dubois

Computational Geometry

Design algorithms for geometric problems

This thesis

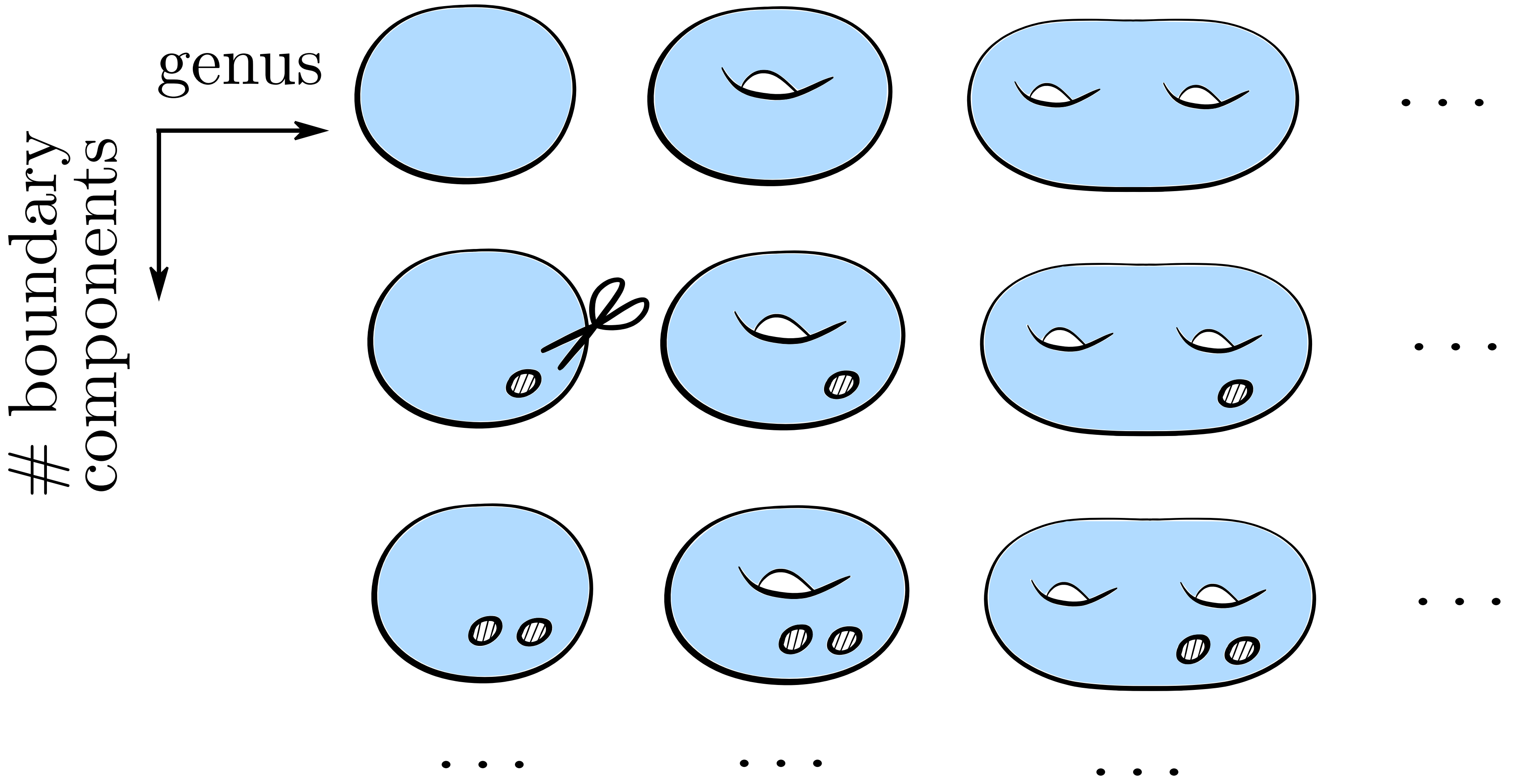
Focus on surfaces



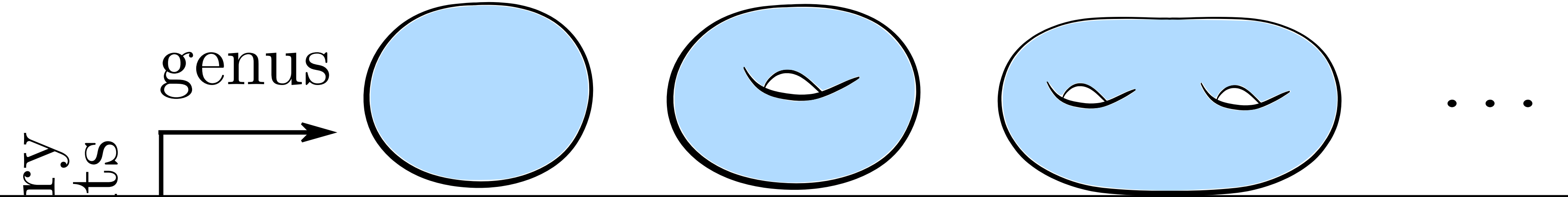
Topology

image by Crane and Segerman

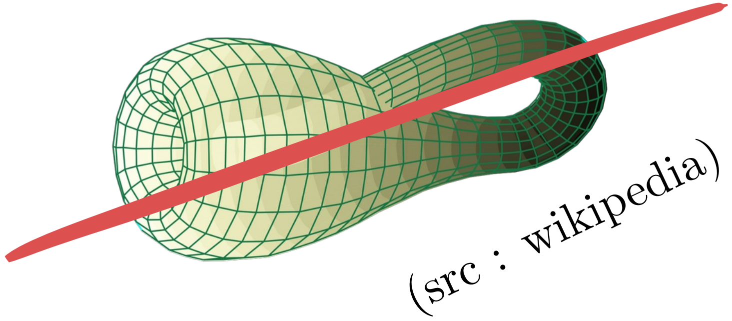
Topological Surfaces

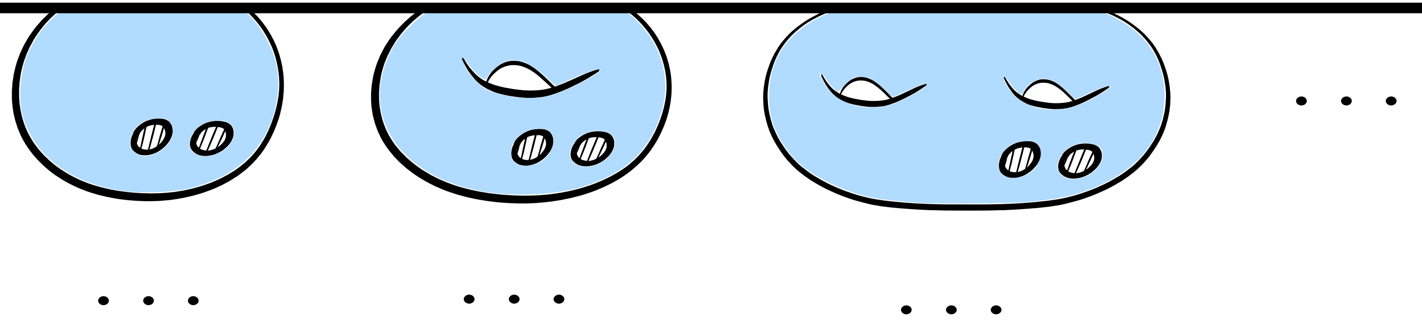


Topological Surfaces

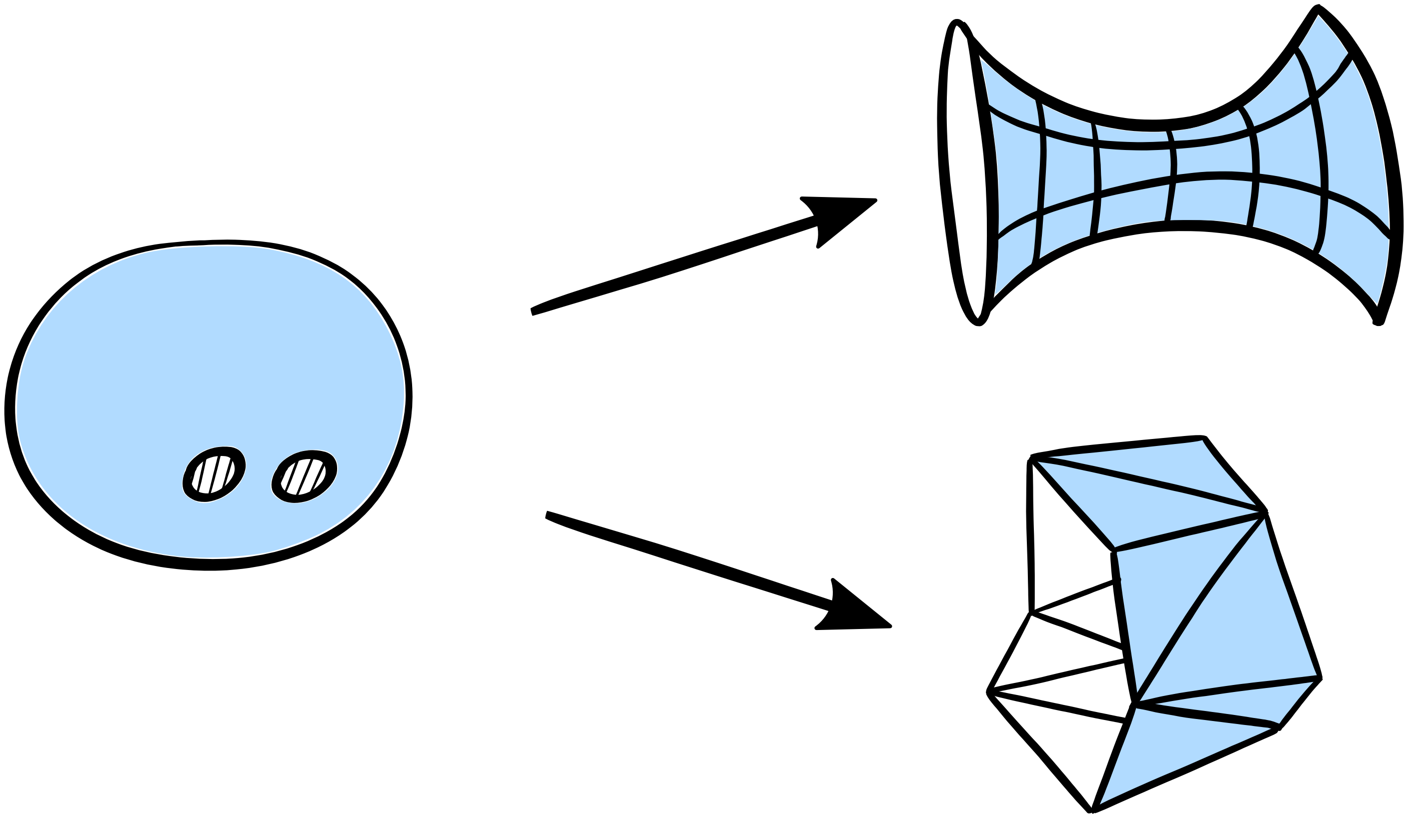


Only orientable surfaces today !





Metrics on surfaces



Untangling Graphs

Computing Delaunay Triangulations

Possible continuations

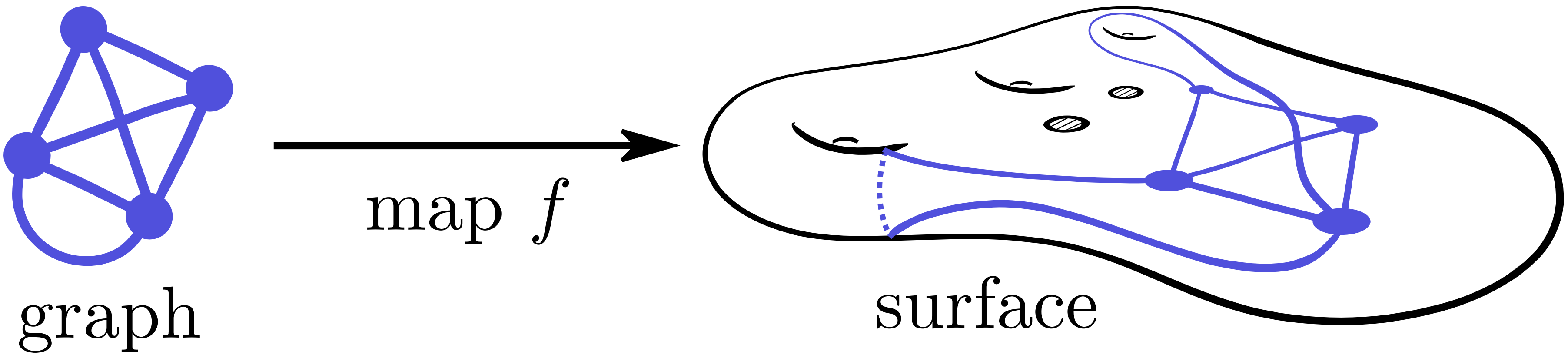
Untangling Graphs

Computing Delaunay Triangulations

Possible continuations

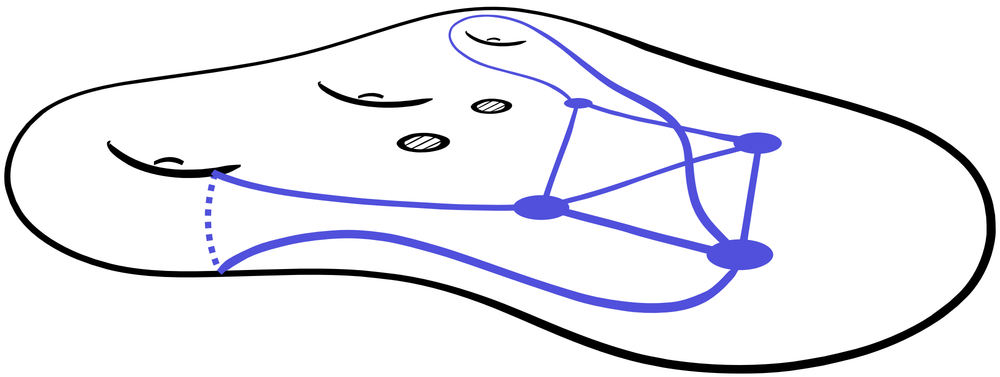
Problem: untangling graphs

Input:

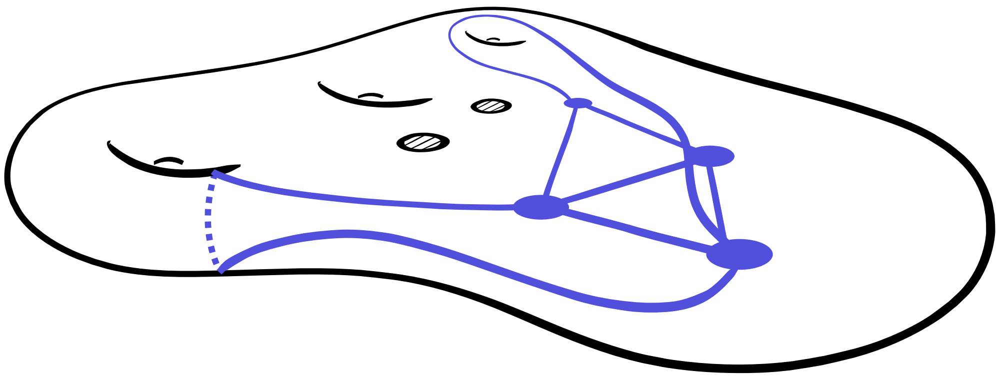


Goal: remove all crossings by deforming f

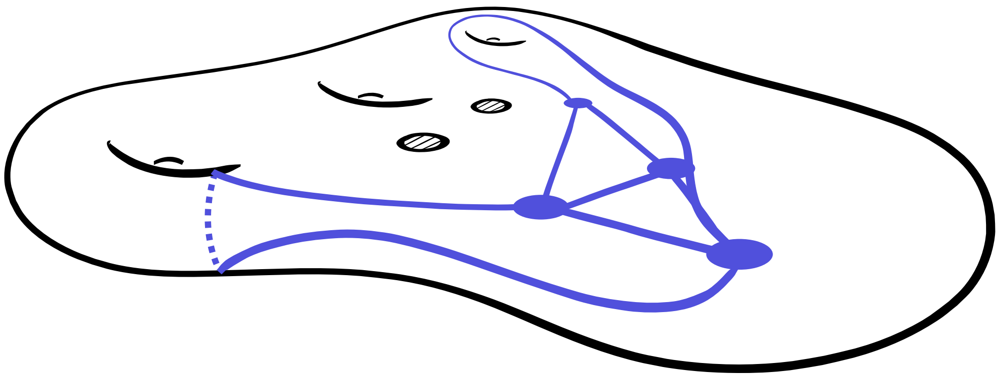
Problem: untangling graphs



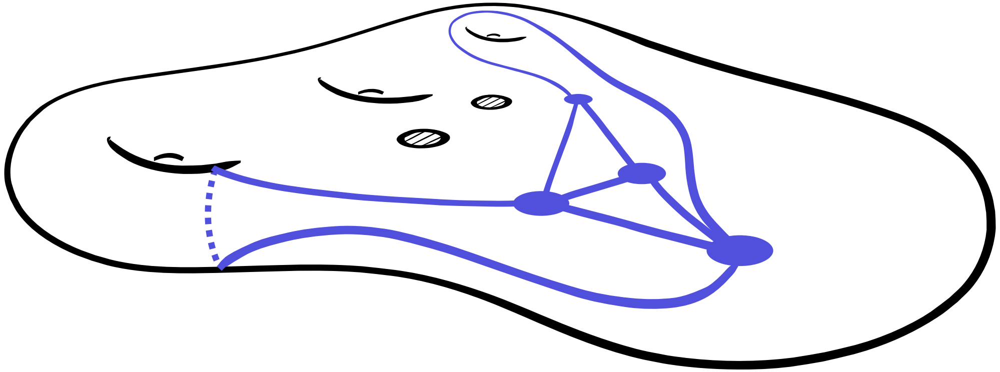
Problem: untangling graphs



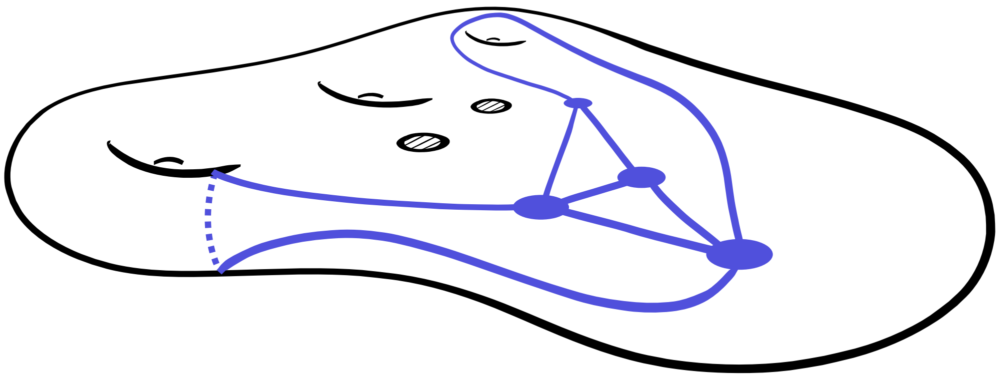
Problem: untangling graphs



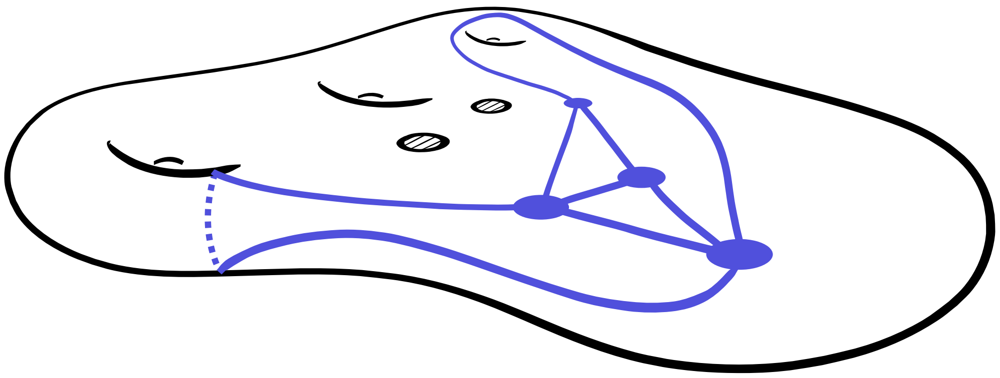
Problem: untangling graphs



Problem: untangling graphs

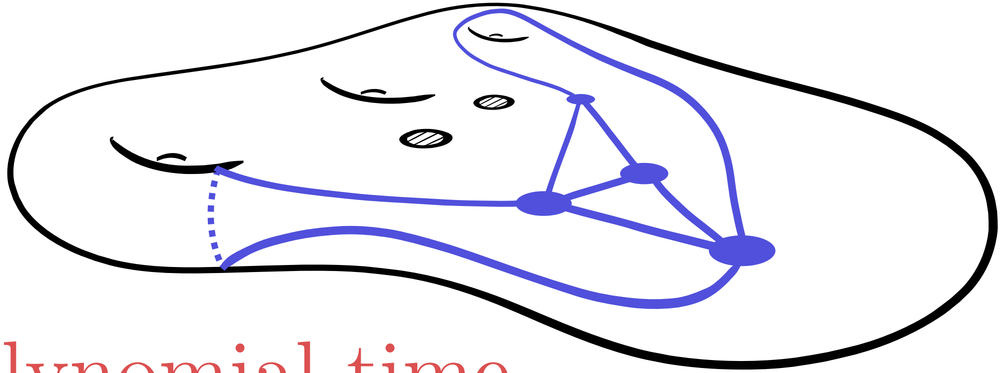


Problem: untangling graphs



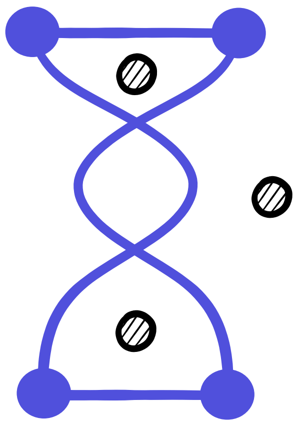
Output: Yes (+ untangled drawing) or No

Problem: untangling graphs

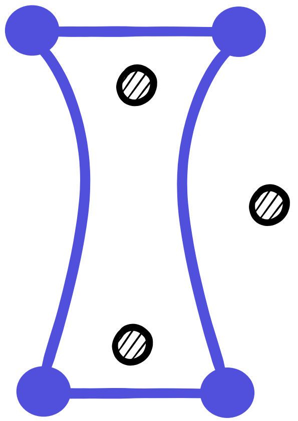


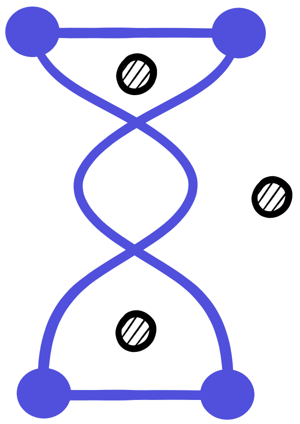
We obtain the first polynomial time algorithms for this problem

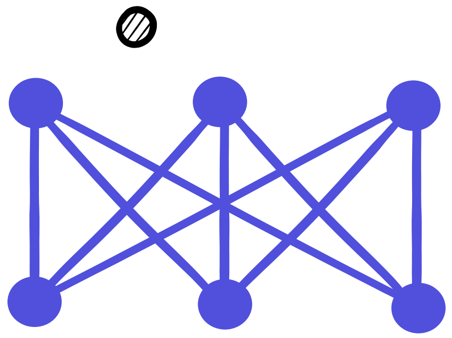
Output: Yes (+ untangled drawing) or No



Yes:

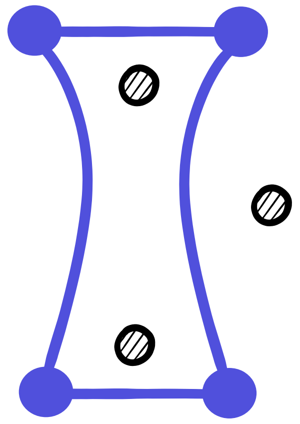


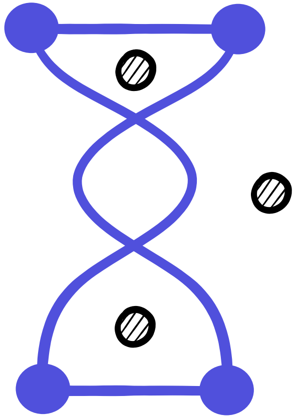


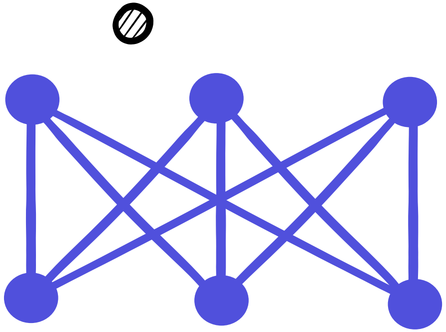


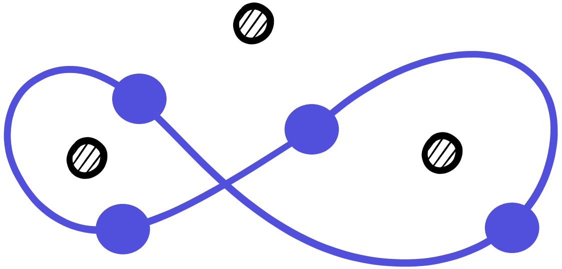
Yes:

No





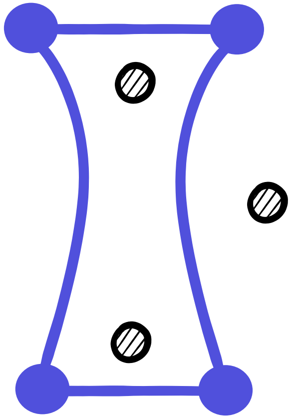




Yes:

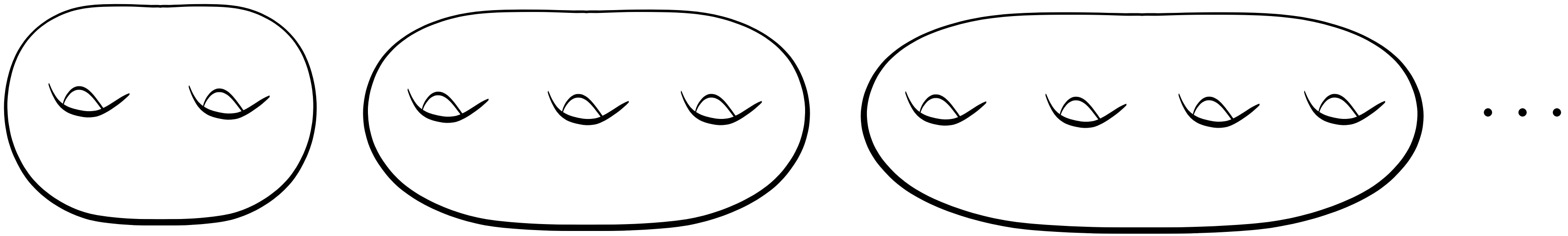
No

No



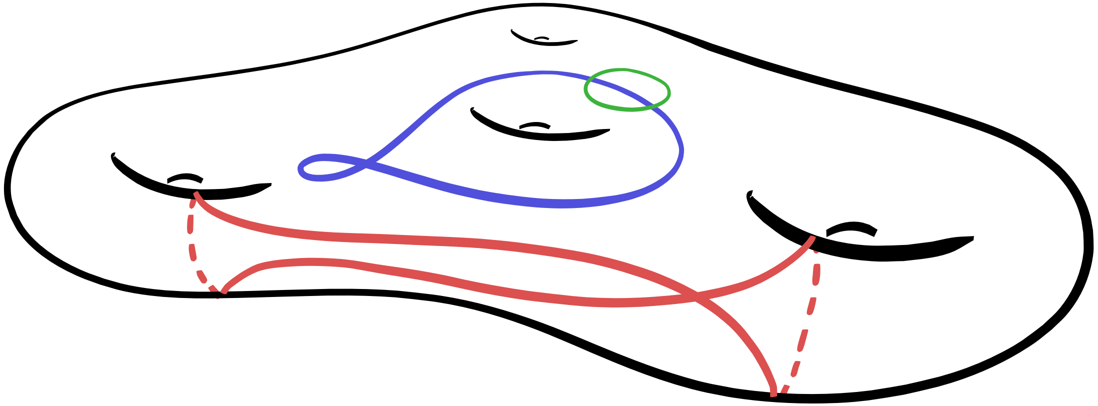
Related works

We focus on surfaces without boundary of genus ≥ 2



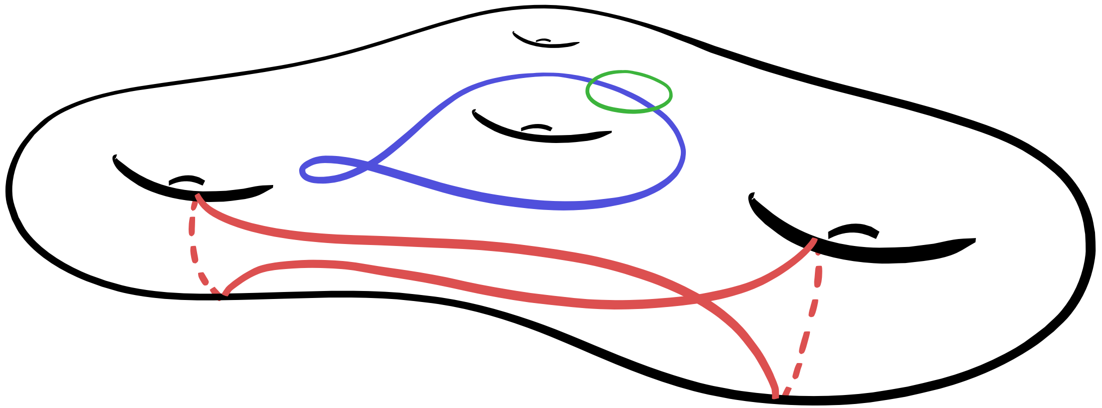
Related problem: making curves cross minimally

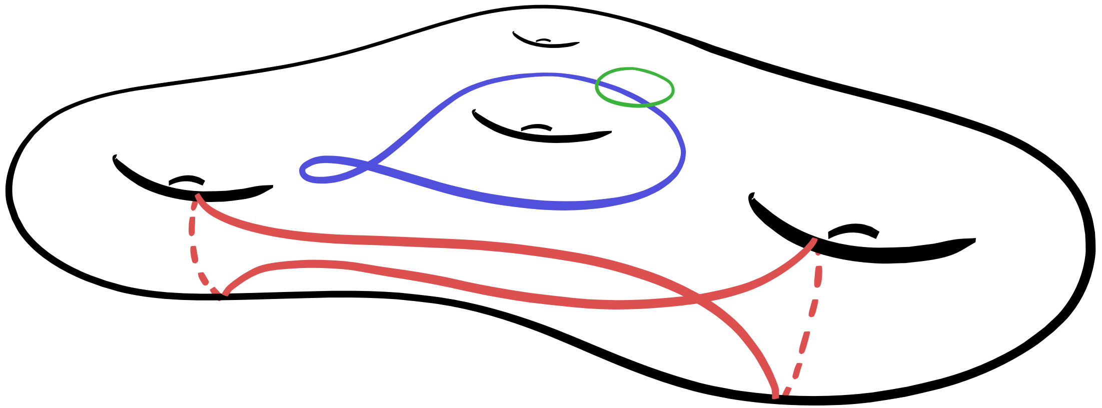
Input: closed curves on a surface

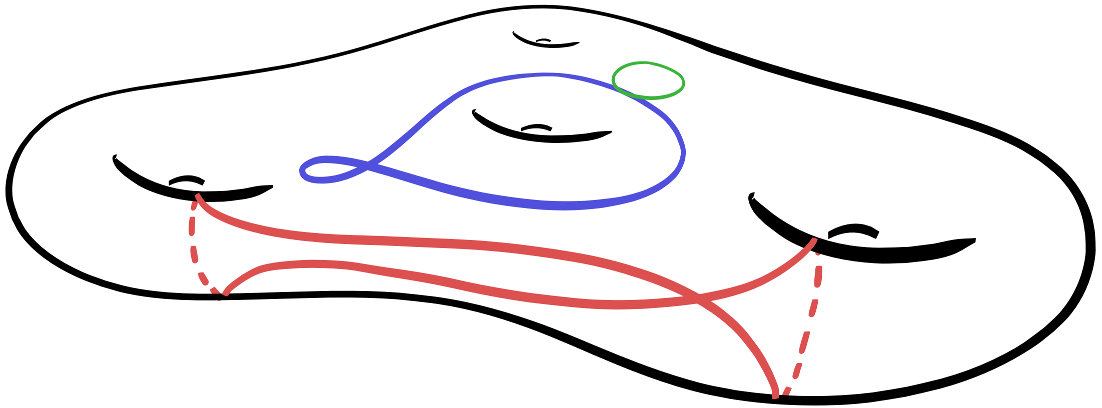


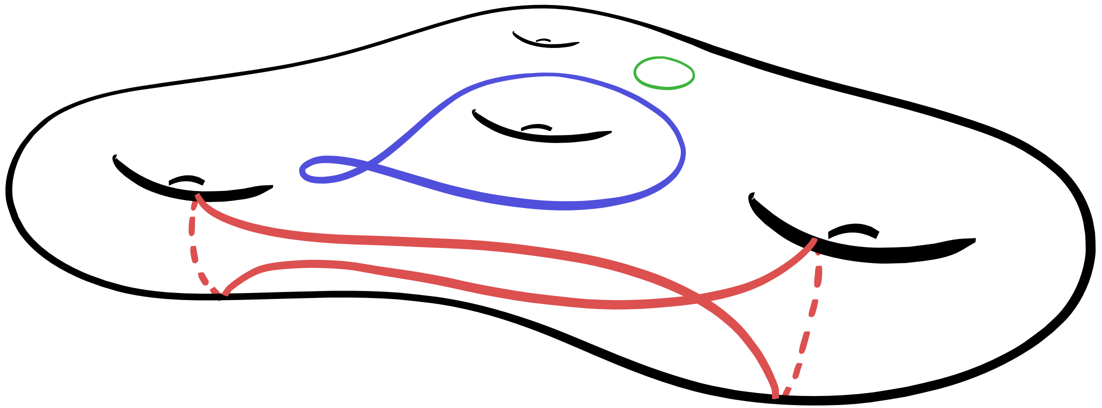
Goal:

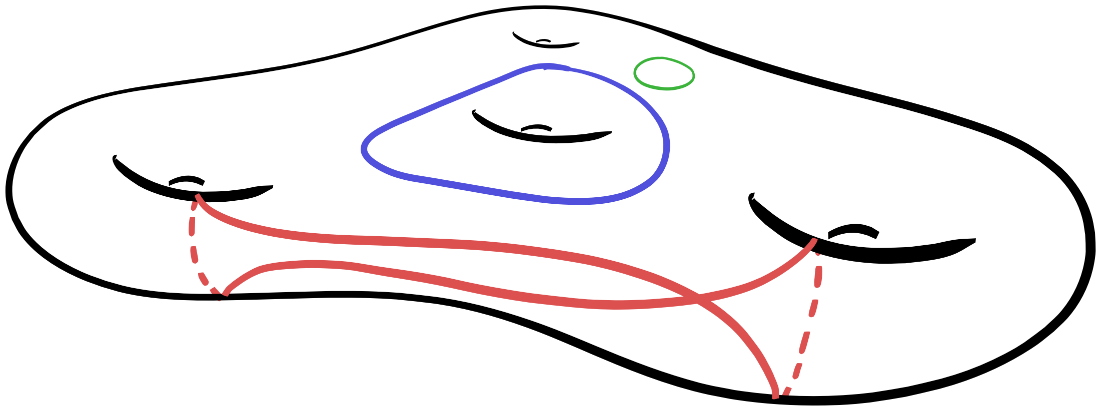
minimize the # crossings by deforming the curves

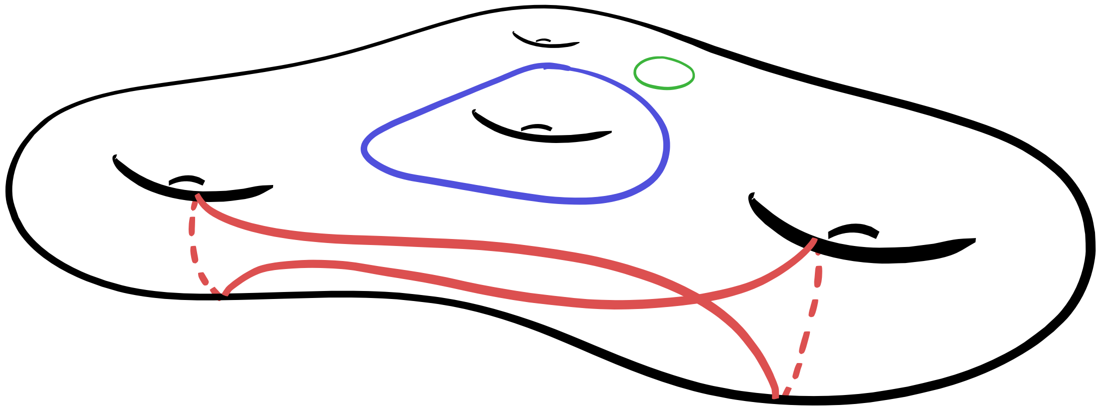












Output: min # of crossings (+ optimal curves)

Many works related to making curves cross minimally!

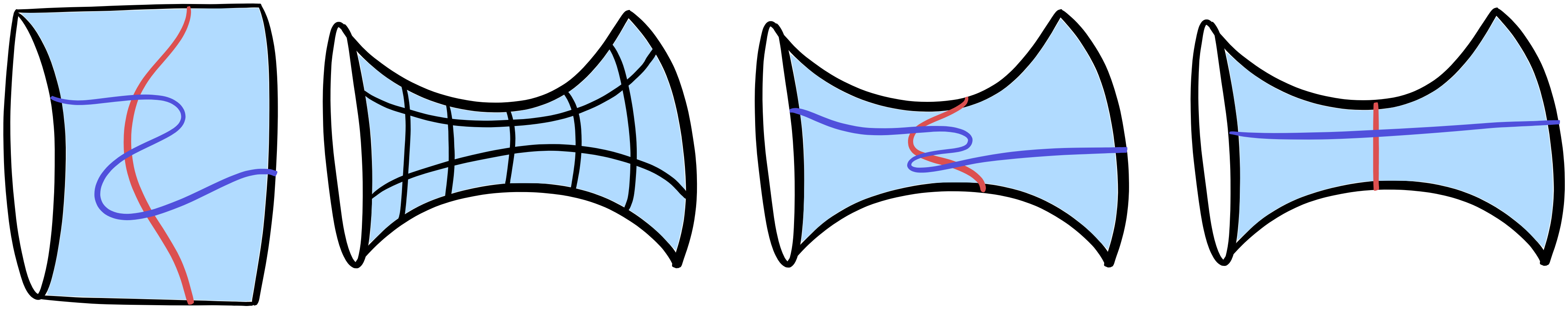
Poincaré, 1905
Dehn, 1911
Dehn, 1912
Reinhart, 1962
Zieschang, 1965
Chillingworth, 1969
Zieschang, 1969
Chillingworth, 1971
Turaev, 1979
Birman and Series, 1984
Cohen and Lustig, 1984
Hass and Scott, 1985
Lustig, 1987

de Graaf and Schrijver, 1987
Dynnikov, 2002
Paterson, 2002
Gonçalves et al., 2005
Schaefer et al., 2008
Lazarus and Rivaud, 2012
Erickson and Whittlesey, 2013
Arettines, 2015
Chang et al., 2018
Despré and Lazarus, 2019
Fulek and Tóth, 2020
Chang and de Mesmay, 2022
Lackenby, 2024

...

Method for making curves cross minimally

Poincaré, 1905



1. give special shape to surface
2. straighten the curves

The special shape

negative curvature:

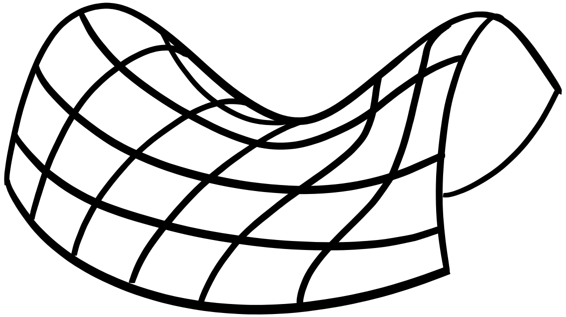


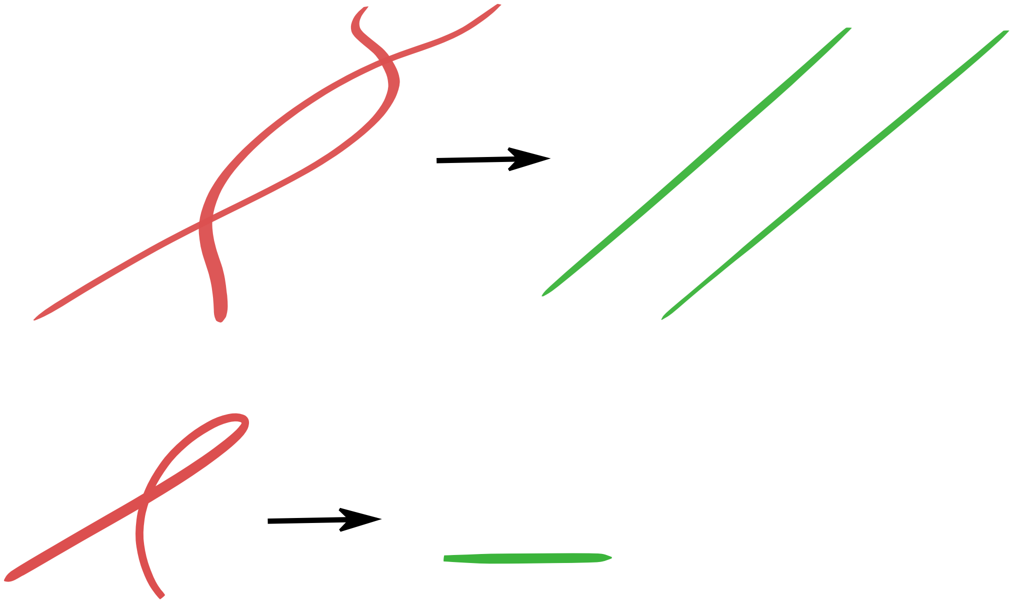
image by Susan Lombardo

almost all surfaces
can be curved negatively

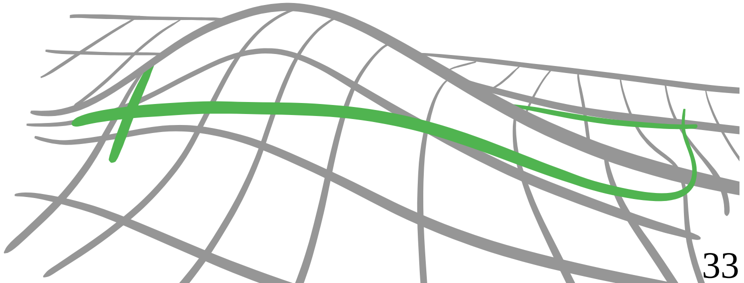
The key property

On a negatively curved surface,

the geodesics cross minimally



(does not hold
on all surfaces)



The key property

On a negatively curved surface,

the geodesics cross minimally

The key property

On a negatively curved surface,

$\boxed{\text{the geodesics cross minimally}}$ $\Longleftarrow$ eve
co

The key property

On a negatively curved surface,

$\boxed{\text{the geodesics cross minimally}} \Longleftarrow$ every homo contains a

The key property

On a negatively curved surface,

he geodesics cross minimally $\Leftarrow$ every homotopy cla
contains a unique

The key property

On a negatively curved surface,

$\boxed{\text{esics cross minimally}}$ $\Longleftarrow$ $\boxed{\text{every homotopy class of pa contains a unique geodesic}}$

The key property

On a negatively curved surface,

$\boxed{\text{oss minimally}}$ $\Longleftarrow$ $\boxed{\text{every homotopy class of paths contains a unique geodesic}}$

The key property

On a negatively curved surface,

$\boxed{\text{mally}} \Longleftarrow \boxed{\text{every homotopy class of paths contains a unique geodesic}}$

The key property

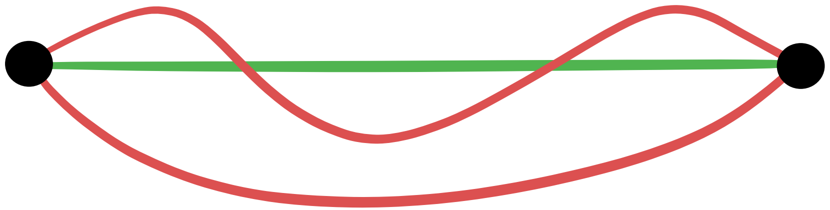
On a negatively curved surface,

every homotopy class of paths contains a unique geodesic

The key property

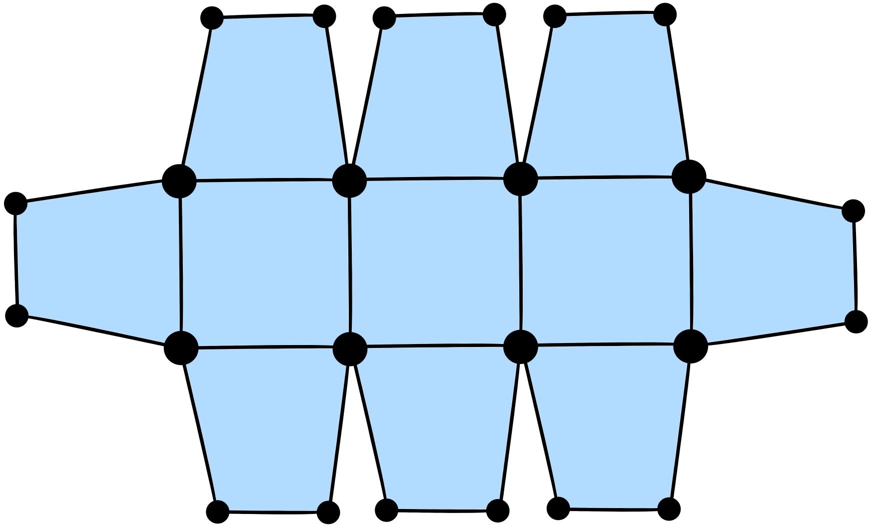
On a negatively curved surface,

every homotopy class of paths contains a unique geodesic

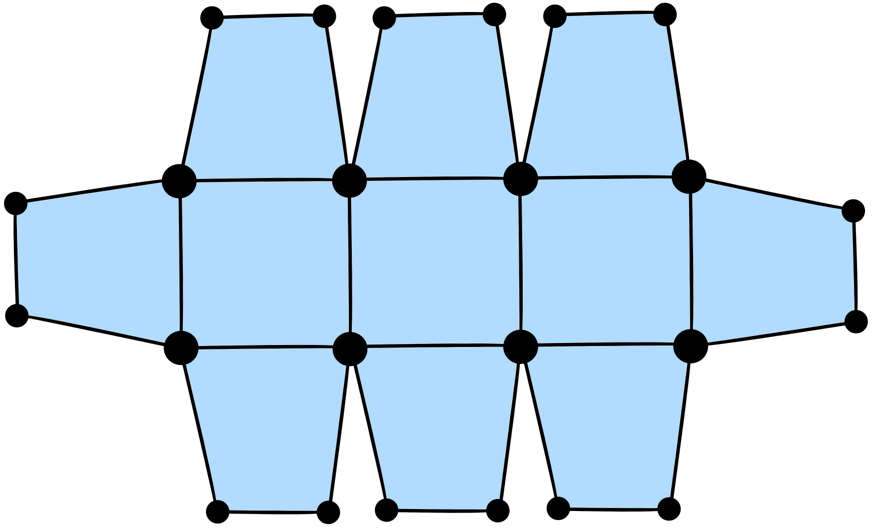


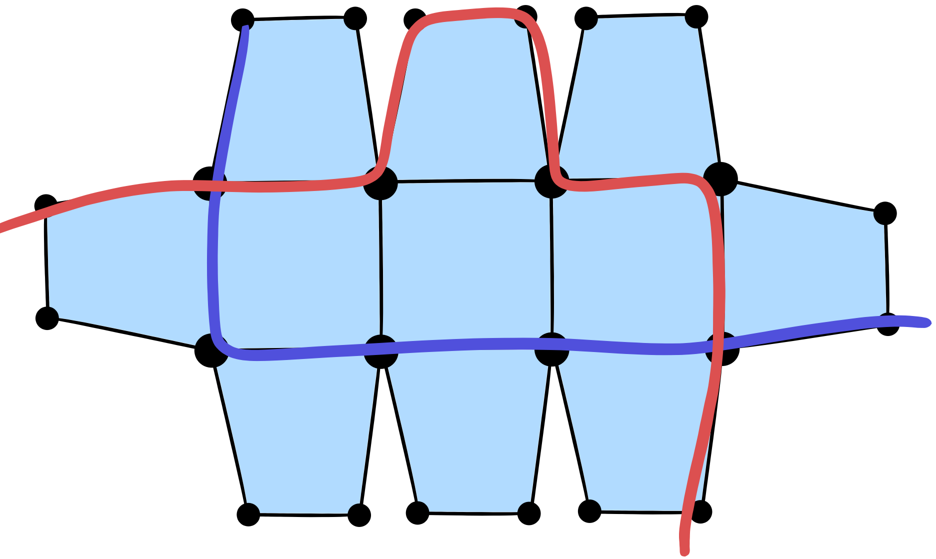
Discrete model of negatively curved surfaces?

Lazarus and Rivaud, 2012
Erickson and Whittlesey, 2013

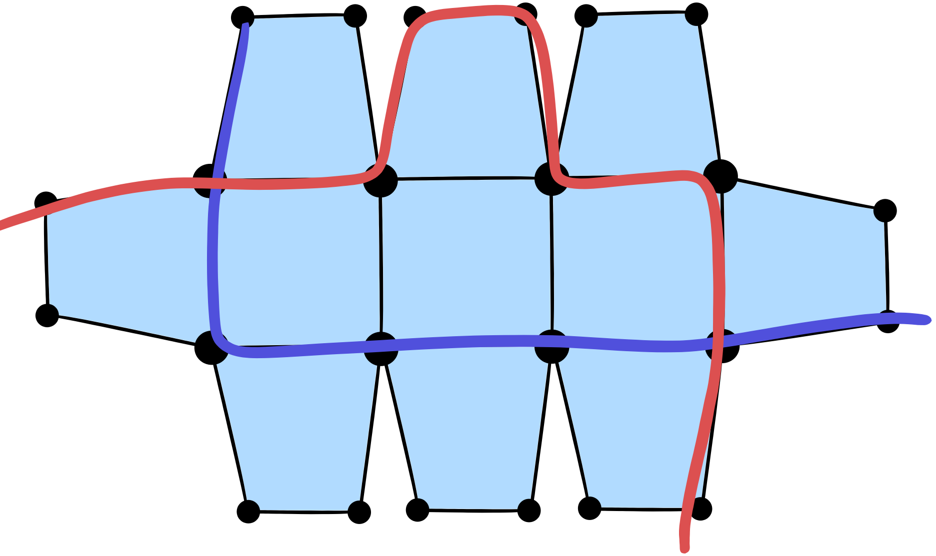


System of quads

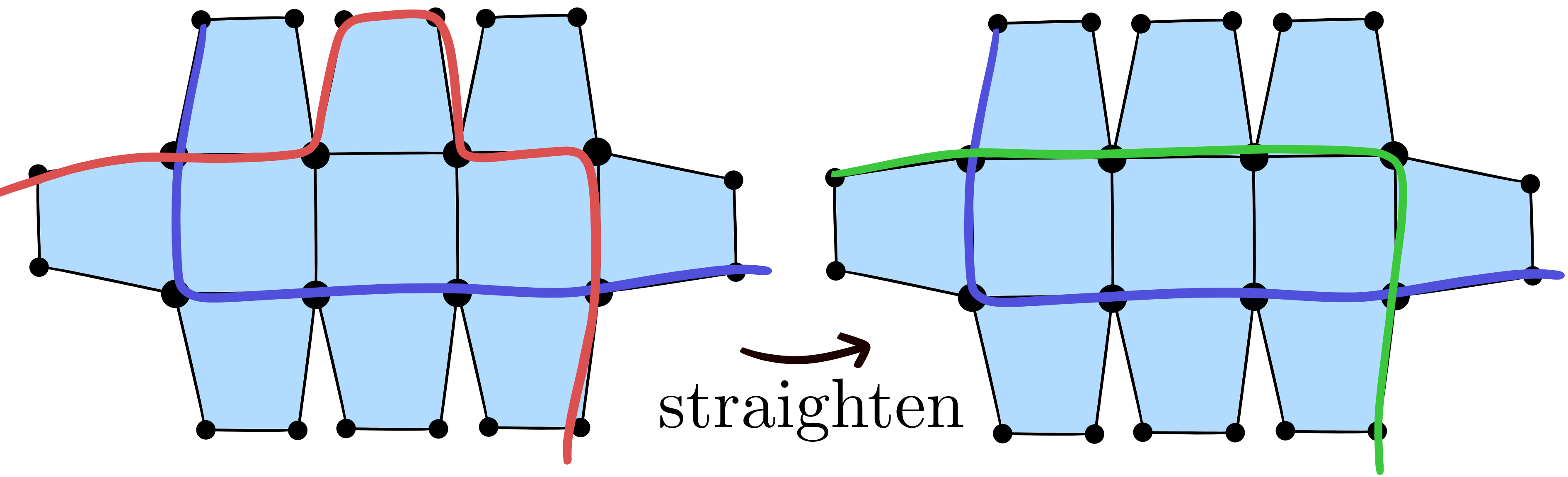




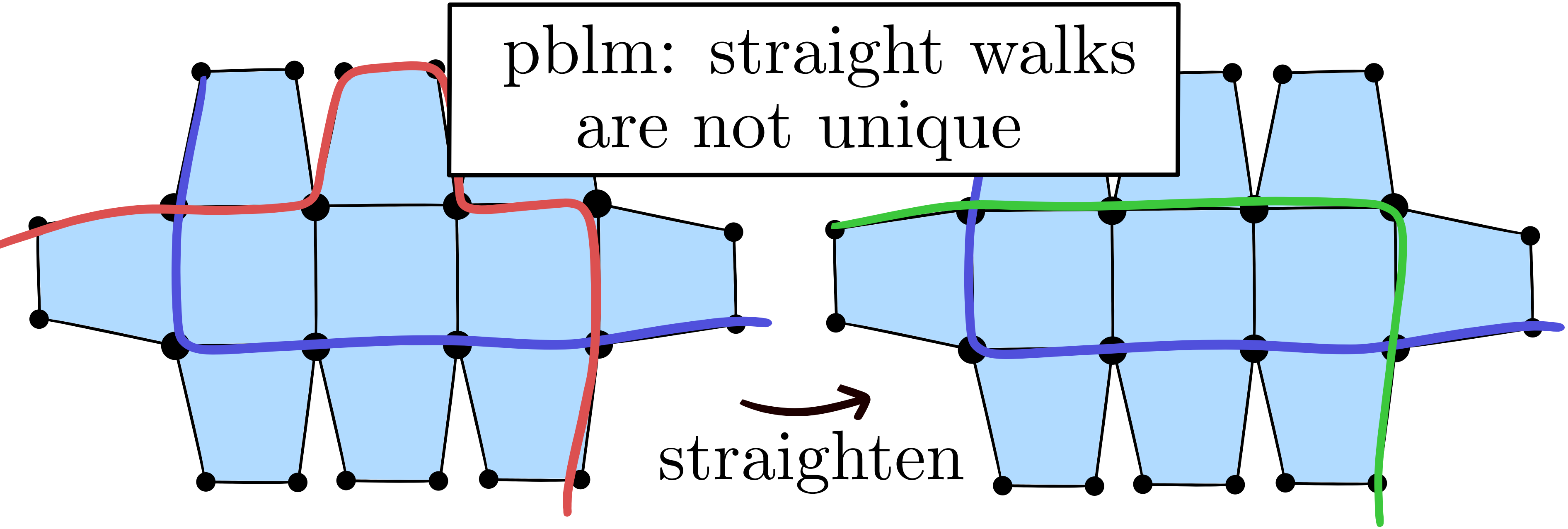
Algo for making curves cross minimally



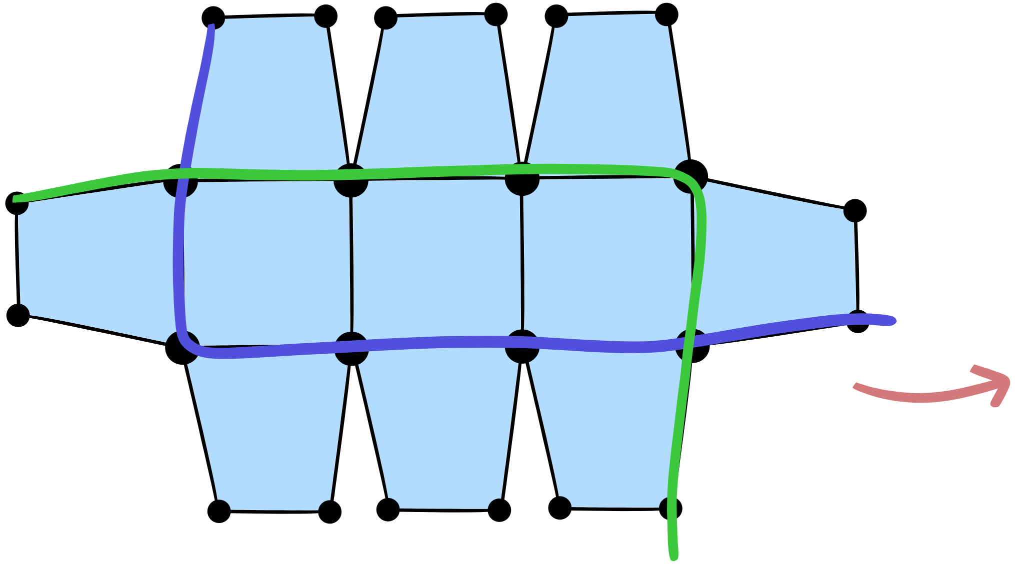
Algo for making curves cross minimally



Algo for making curves cross minimally



Algo for making curves cross minimally



Despré and Lazarus, 2019

What about untangling graphs?

Method for untangling graphs

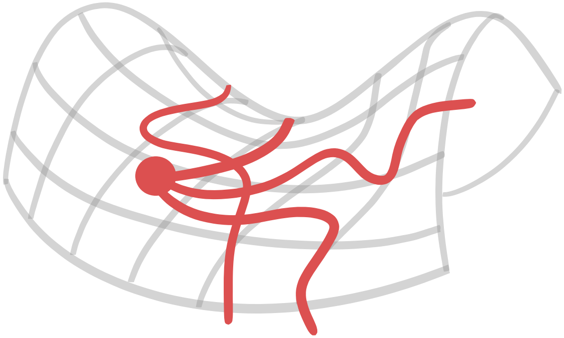
Tutte, 1963

Y. Colin de Verdière, 1991

Method for untangling graphs

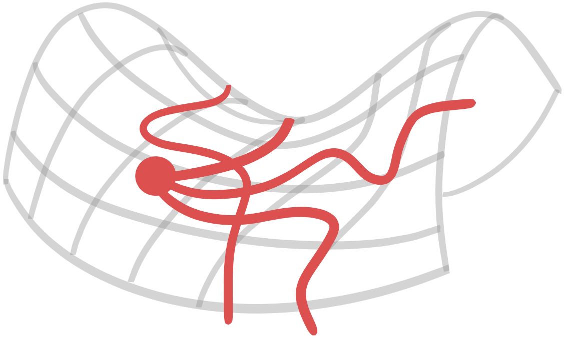
Tutte, 1963

Y. Colin de Verdière, 1991



Method for untangling graphs

Tutte, 1963 Y. Colin de Verdière, 1991

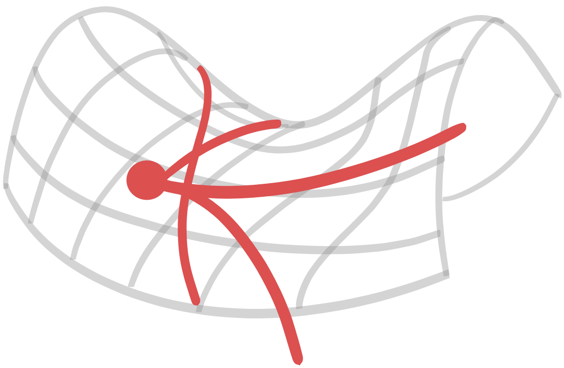


make edges straight

Method for untangling graphs

Tutte, 1963

Y. Colin de Verdière, 1991

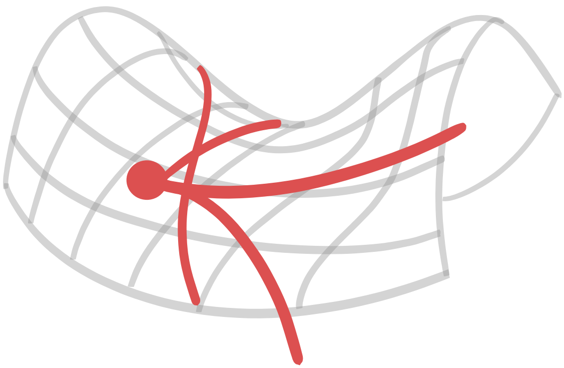


make edges straight

Method for untangling graphs

Tutte, 1963

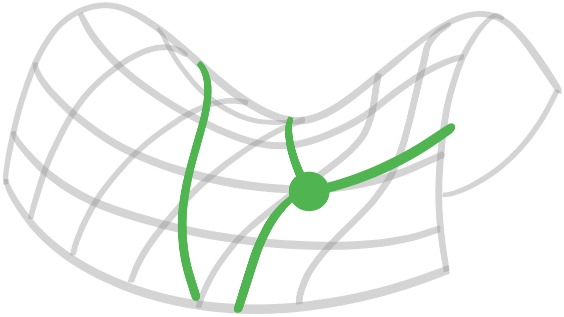
Y. Colin de Verdière, 1991



make edges straight
make vertices barycentric

Method for untangling graphs

Tutte, 1963 Y. Colin de Verdière, 1991

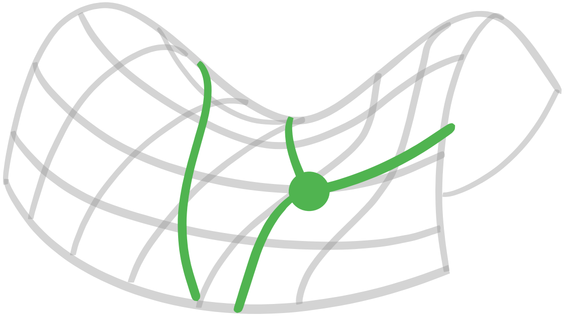


make edges straight
make vertices barycentric

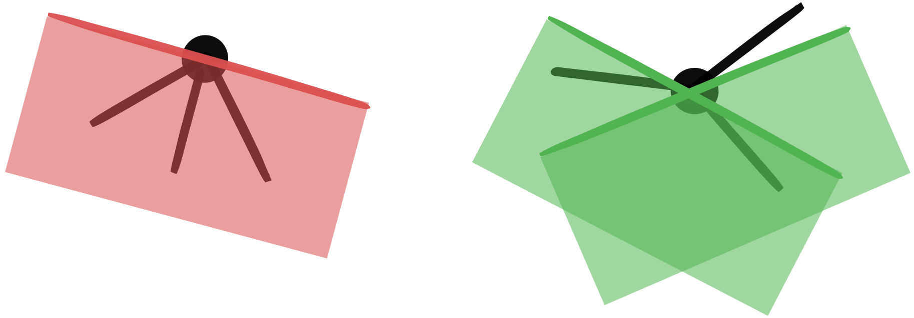
Method for untangling graphs

Tutte, 1963

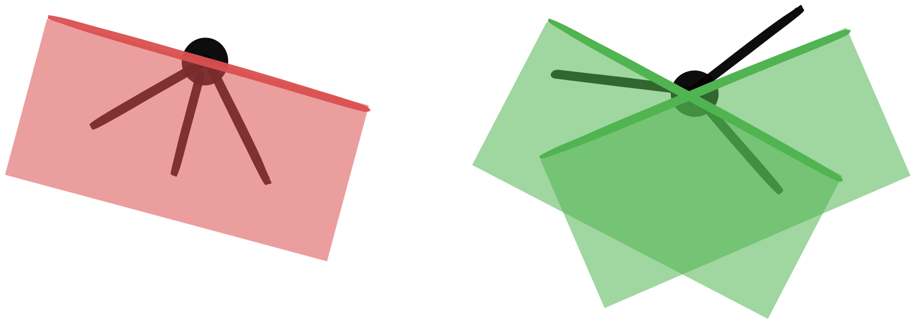
Y. Colin de Verdière, 1991



make edges straight
make vertices barycentric

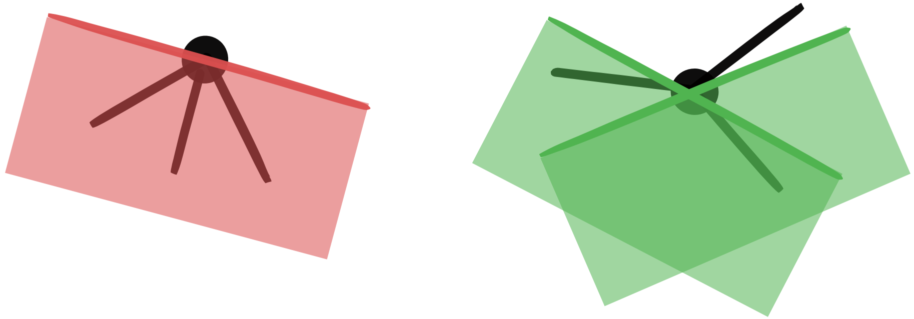


edges straight
vertices barycentric

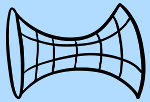
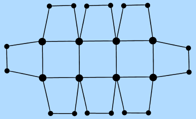


Tutte drawings

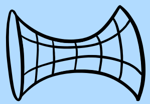
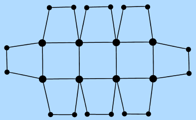
edges straight
vertices barycentric



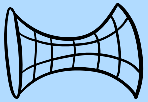
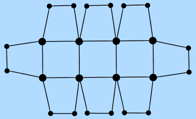
Summary

	Curves	Graphs
Method	negatively curved surface 	
Algo	system of quads 	

Our results

	Curves	Graphs
Method	negatively curved surface 	
Algo	system of quads 	

Our results

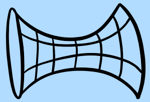
	Curves	Graphs
Method	negatively curved surface 	
Algo	system of quads 	
	reducing triangulations	

Our results

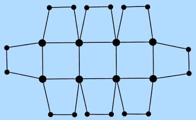
Curves

Graphs

Method

negatively curved surface 

Algo

system of quads 

reducing triangulations

improved algos for making curves cross minimally

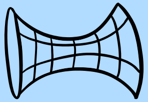
first algos for untangling graphs

Our results

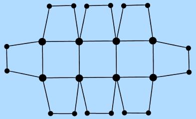
Curves

Graphs

Method

negatively curved surface 

Algo

system of quads 

reducing triangulations

improved algos for making curves cross minimally

first algos for untangling graphs

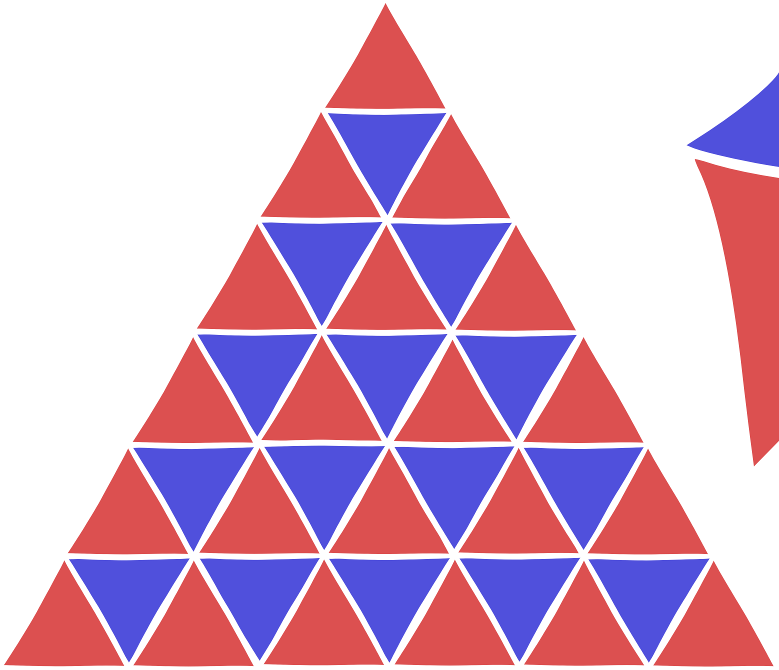
Discrete analogue of Tutte drawings

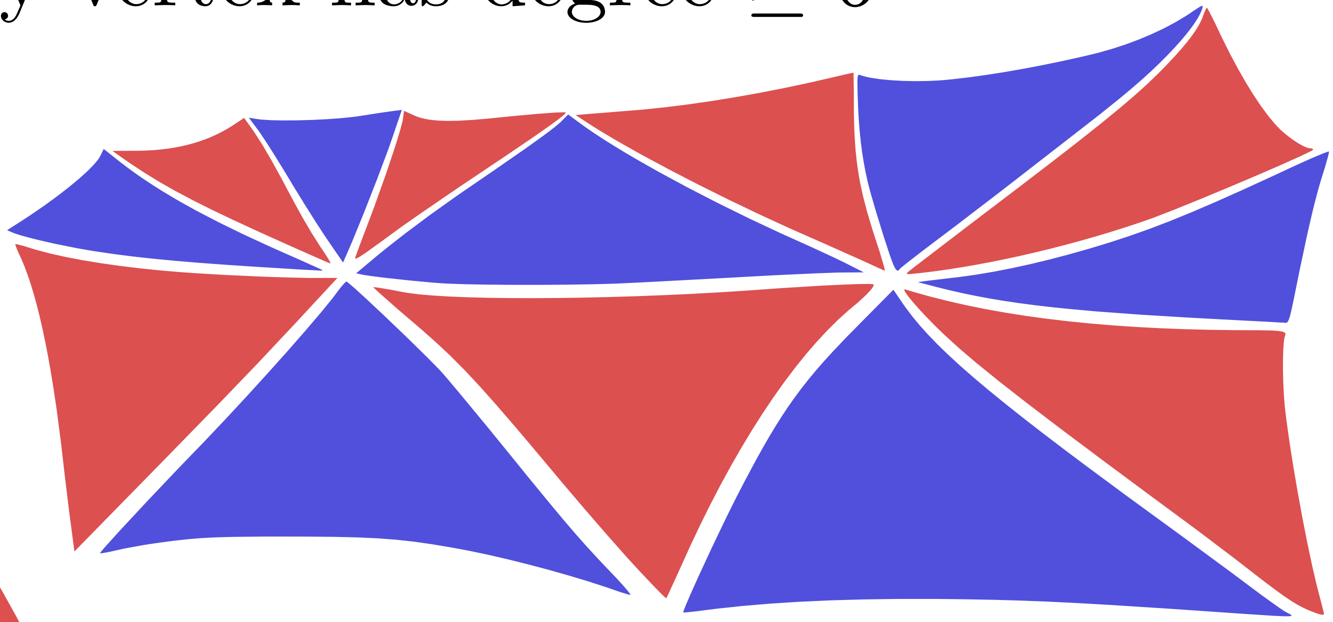
A new tool:

Reducing triangulations

Reducing triangulations

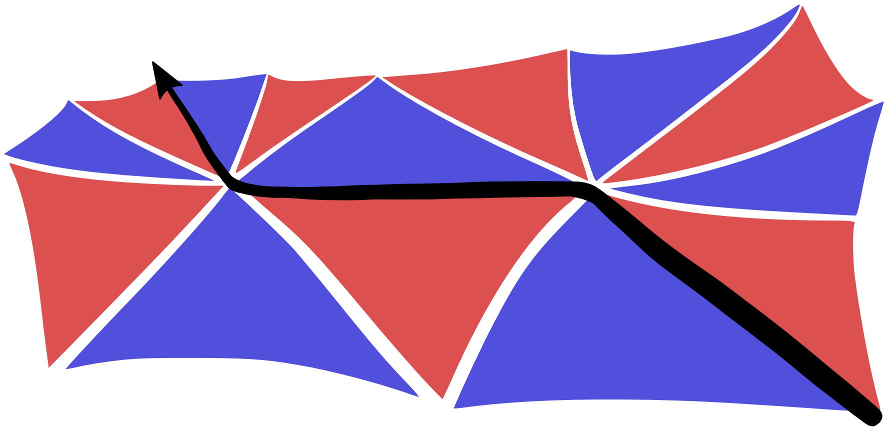
dual is bipartite and
every vertex has degree $\geq 6^*$



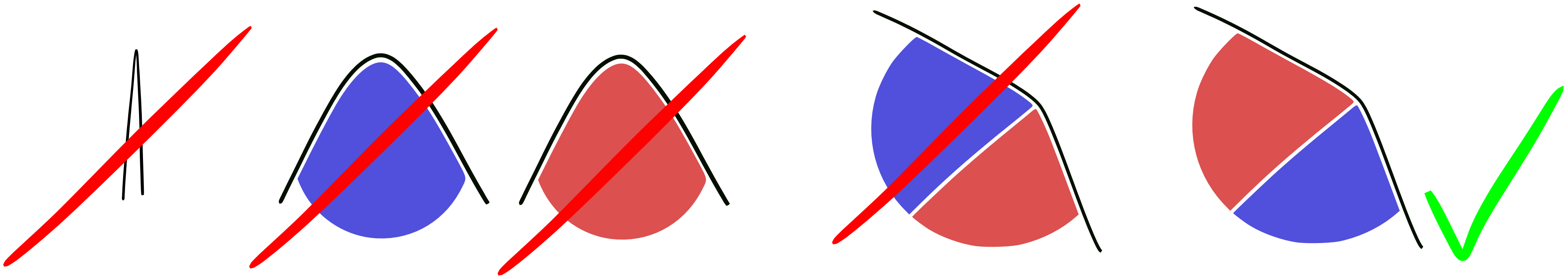


*sometimes 8

Reduced walks

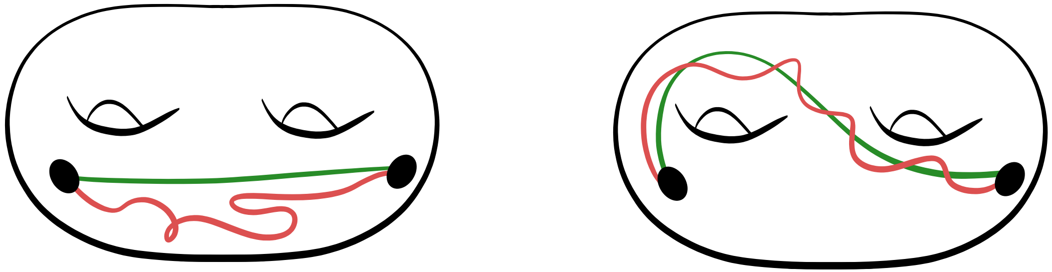


no bad turn



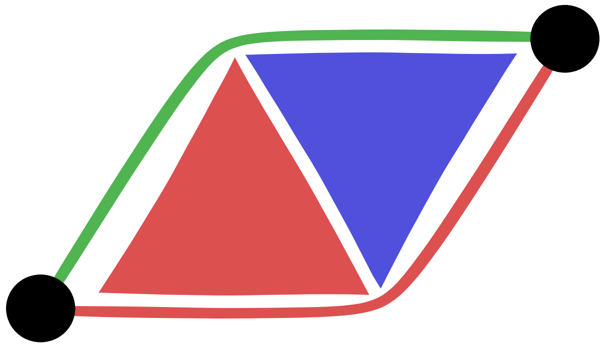
Properties of reduced walks

every walk can be deformed into a unique reduced walk, computable in linear time

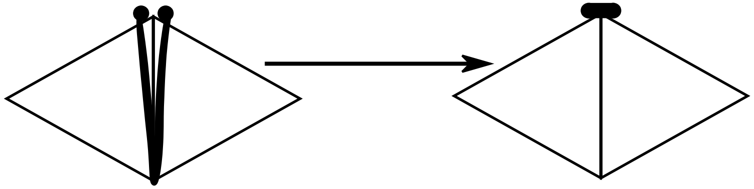


reduced walks are stable upon reversal and subwalk

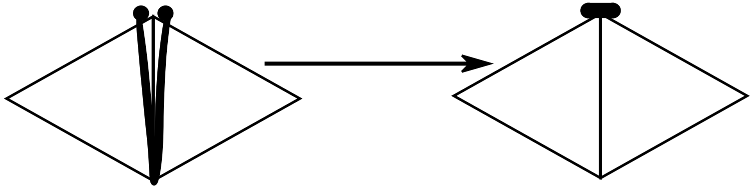
Purpose of the coloring

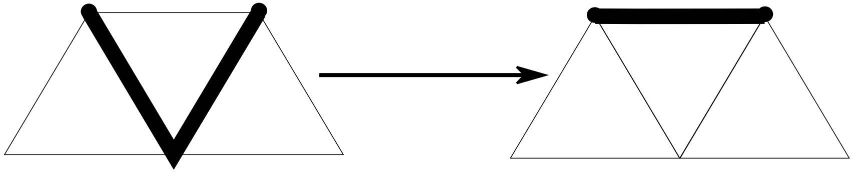


Reducing a walk

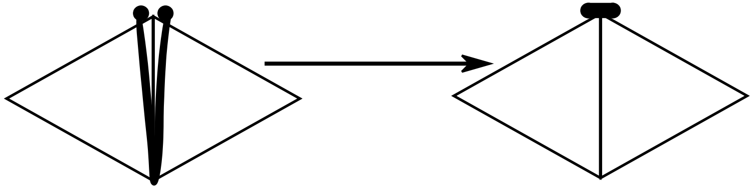


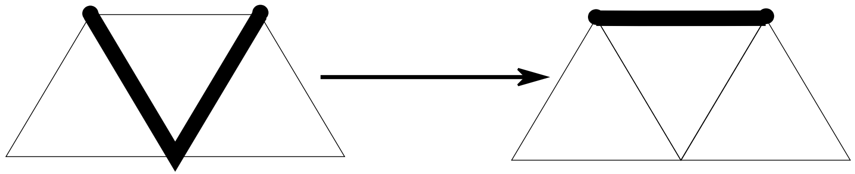
Reducing a walk

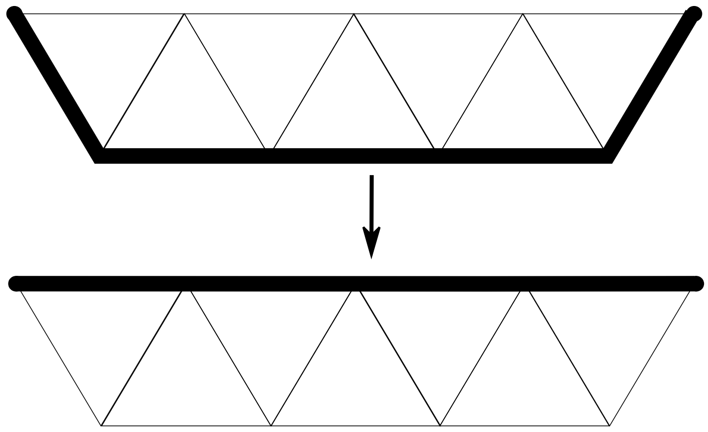




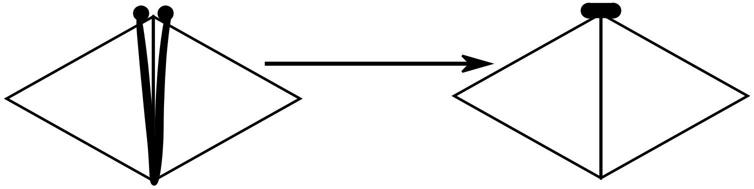
Reducing a walk

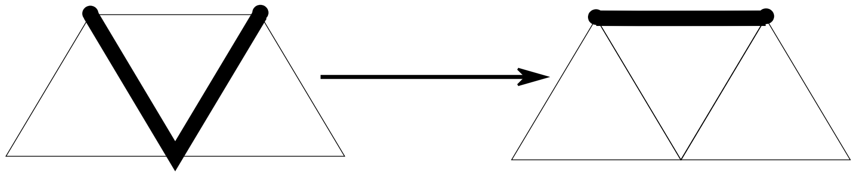


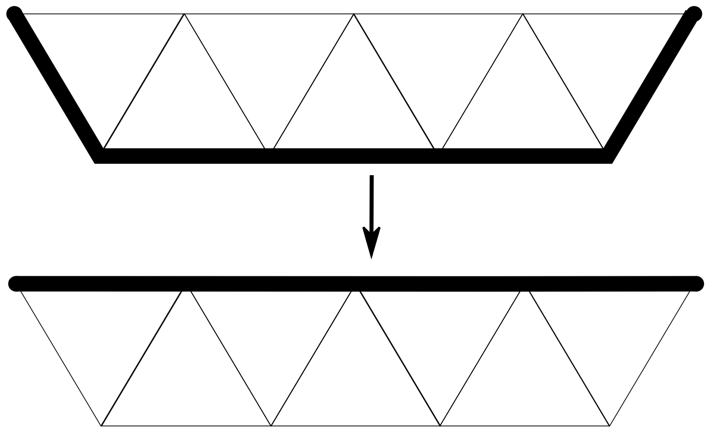


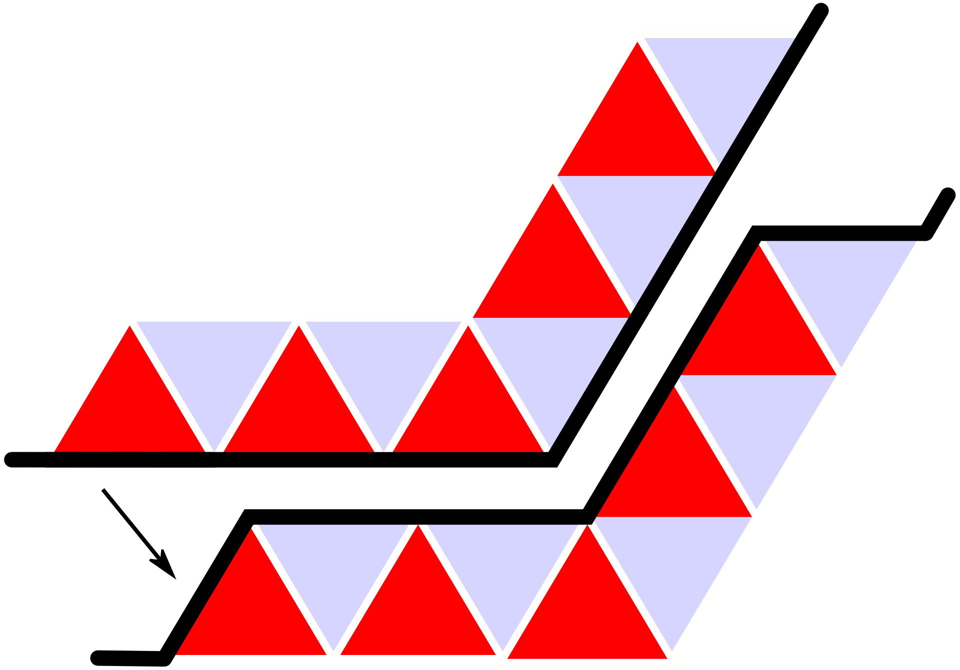


Reducing a walk

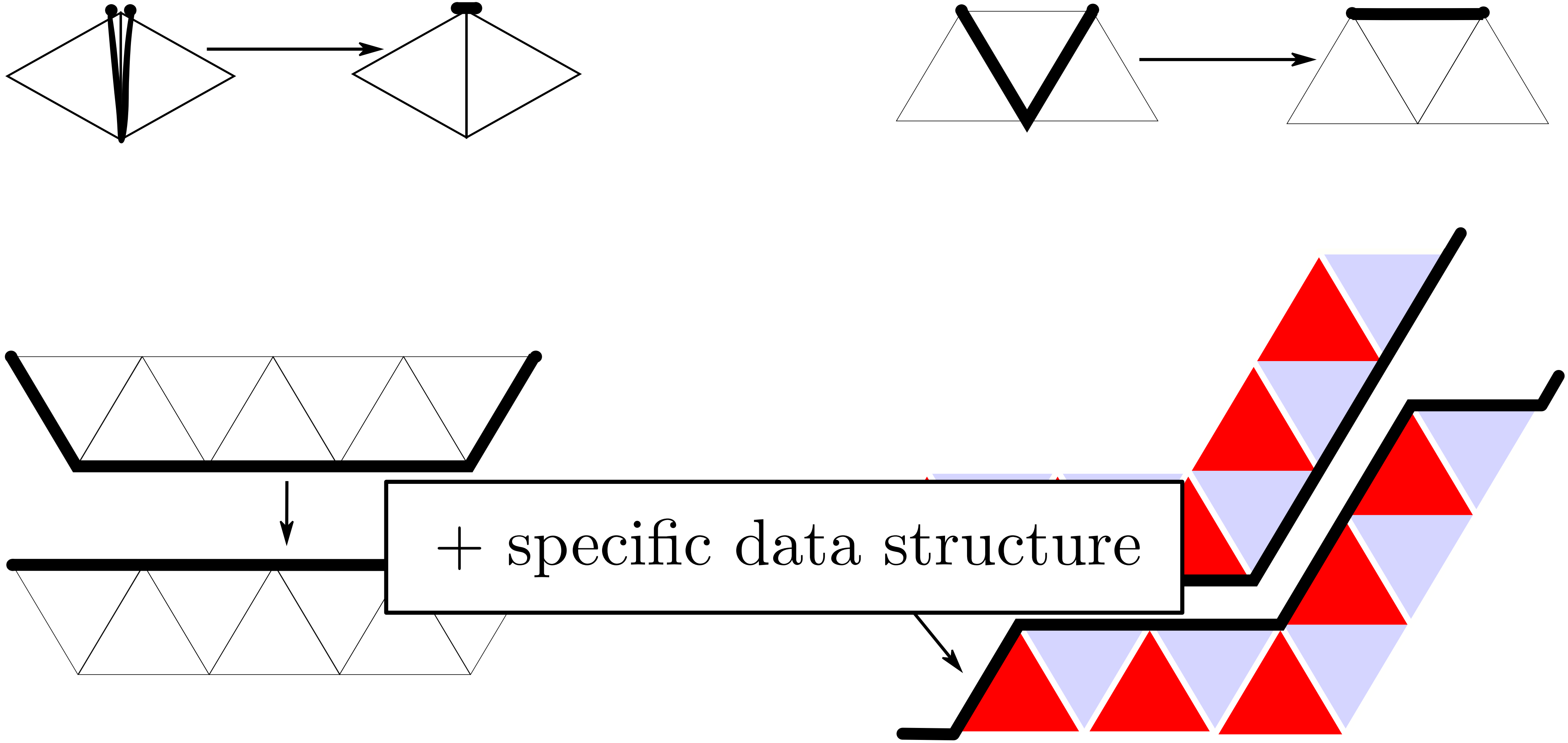






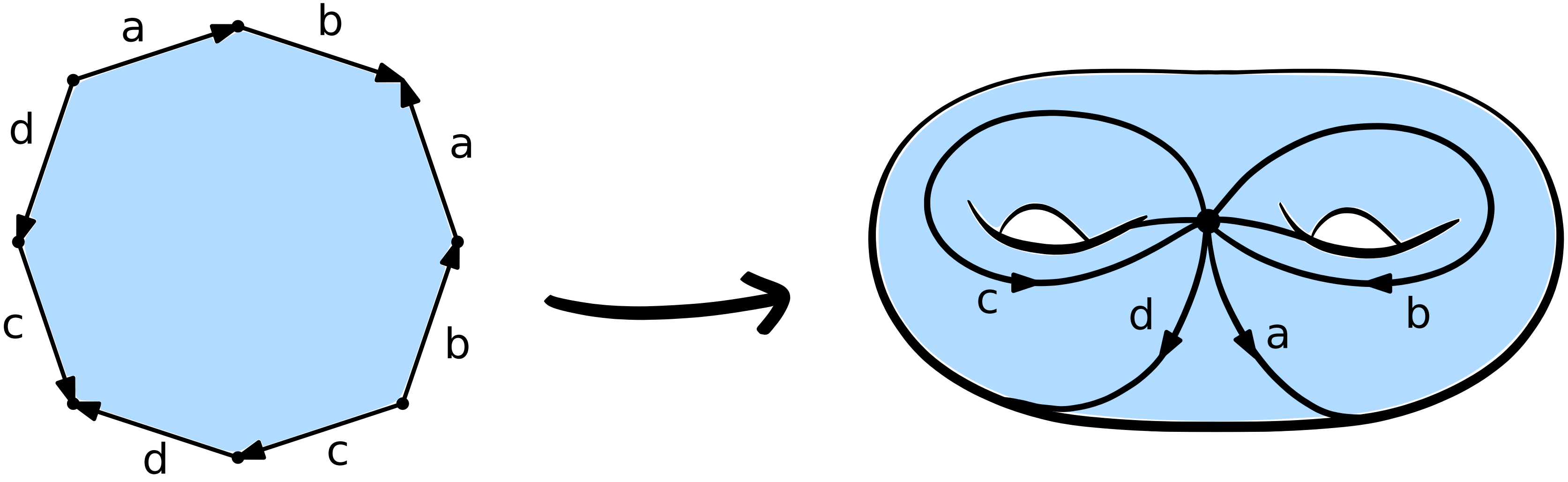


Reducing a walk

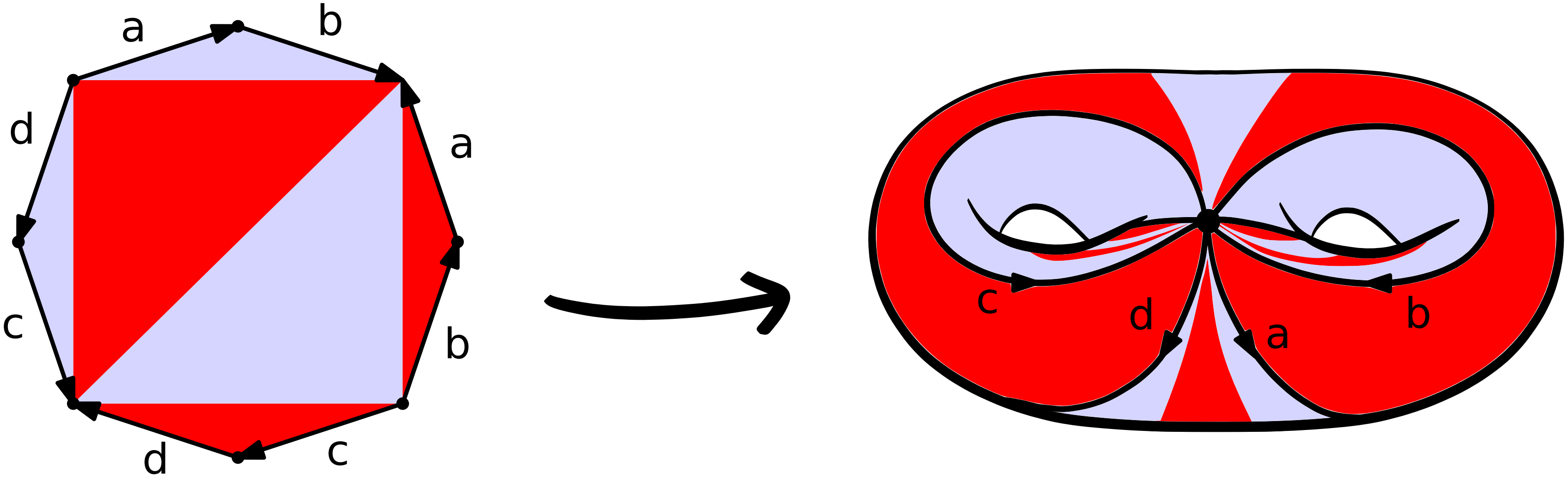


+ specific data structure

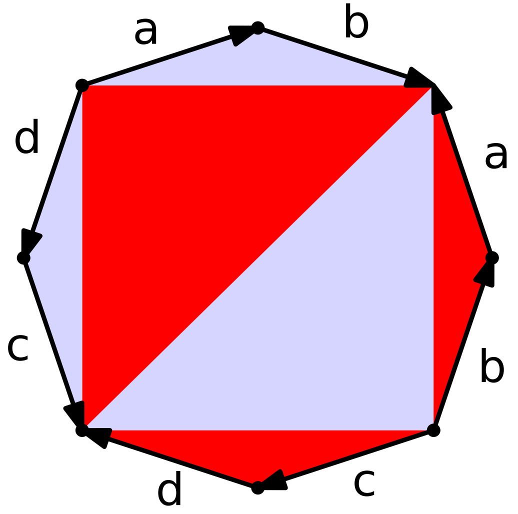
Constructing reducing triangulations



Constructing reducing triangulations

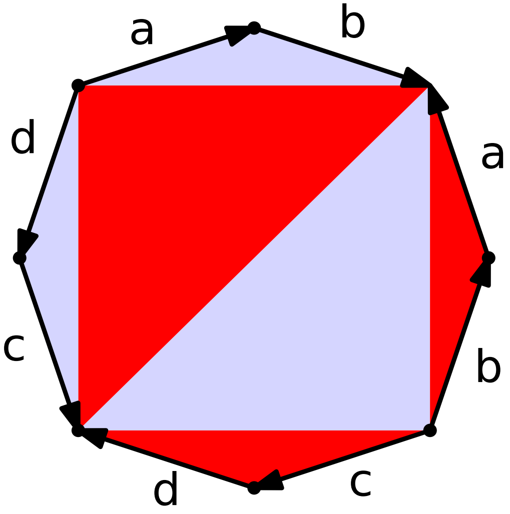


Constructing reducing triangulations



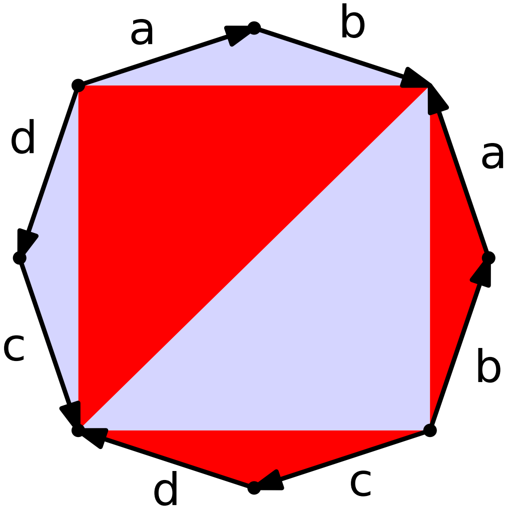
Constructing reducing triangulations

genus 2

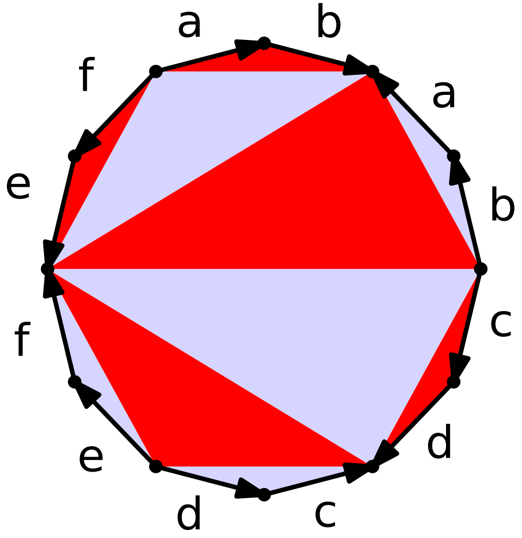


Constructing reducing triangulations

genus 2

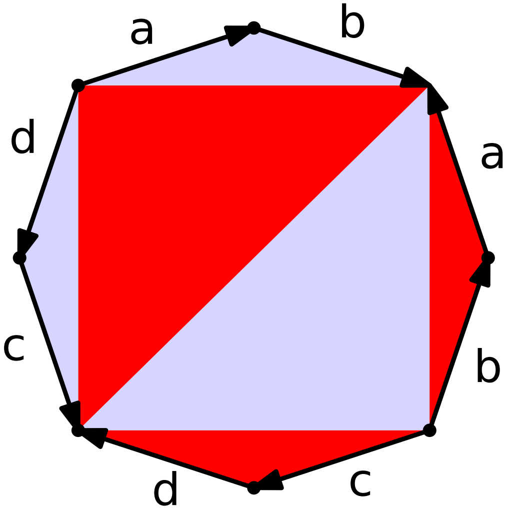


genus 3

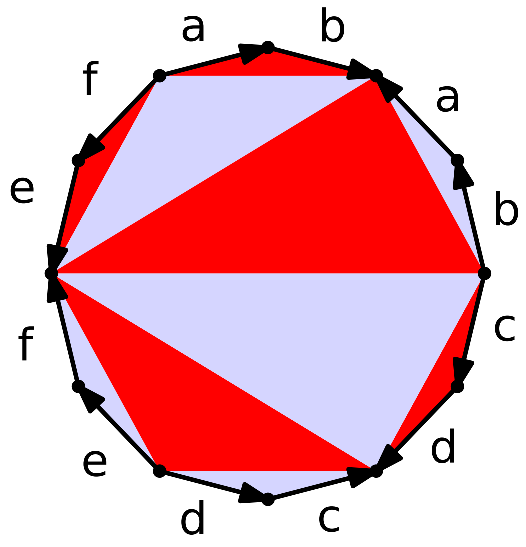


Constructing reducing triangulations

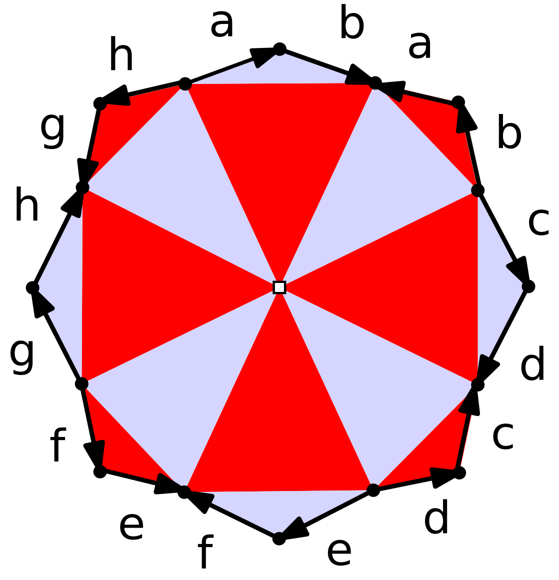
genus 2



genus 3



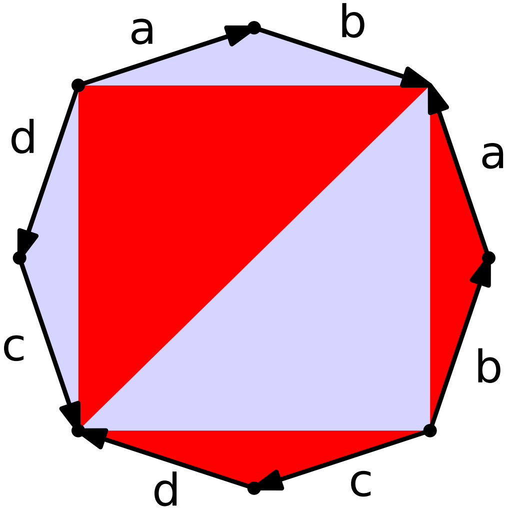
genus 4



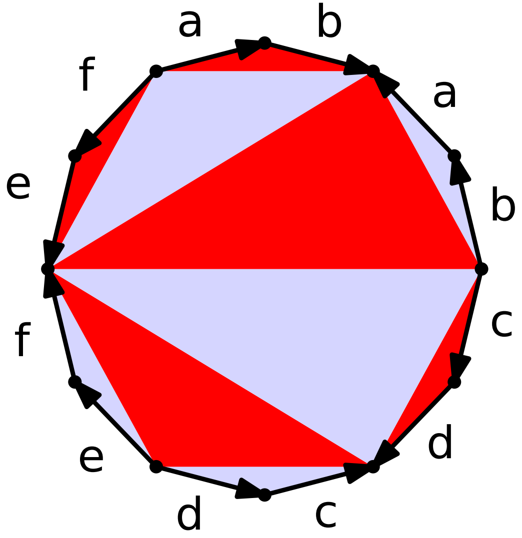
. . .

Constructing reducing triangulations

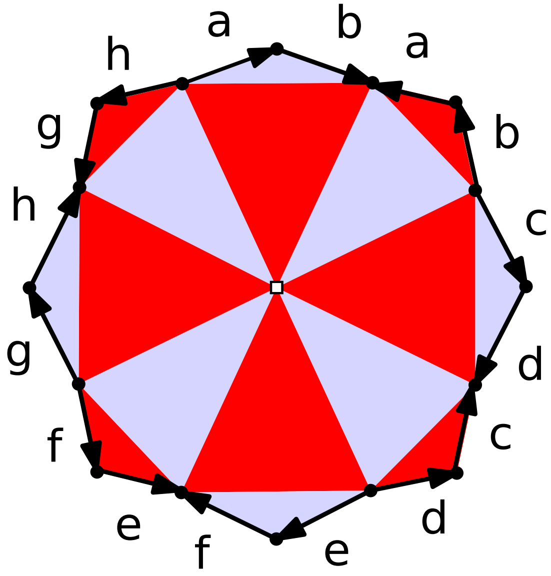
genus 2



genus 3

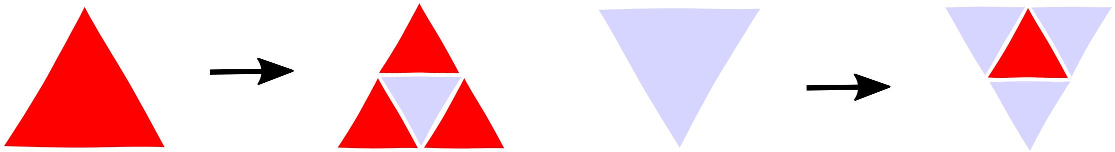


genus 4



. . .

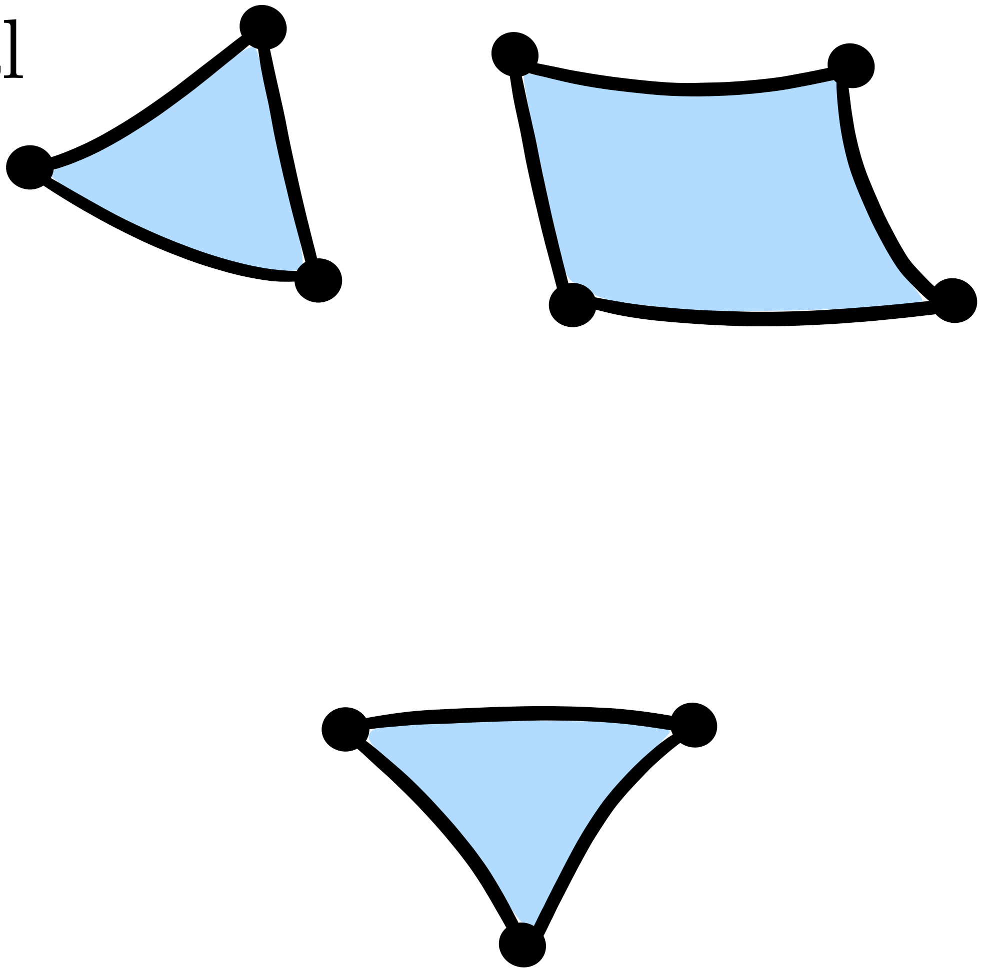
then subdivide at will:



Untangling graphs using reducing triangulations

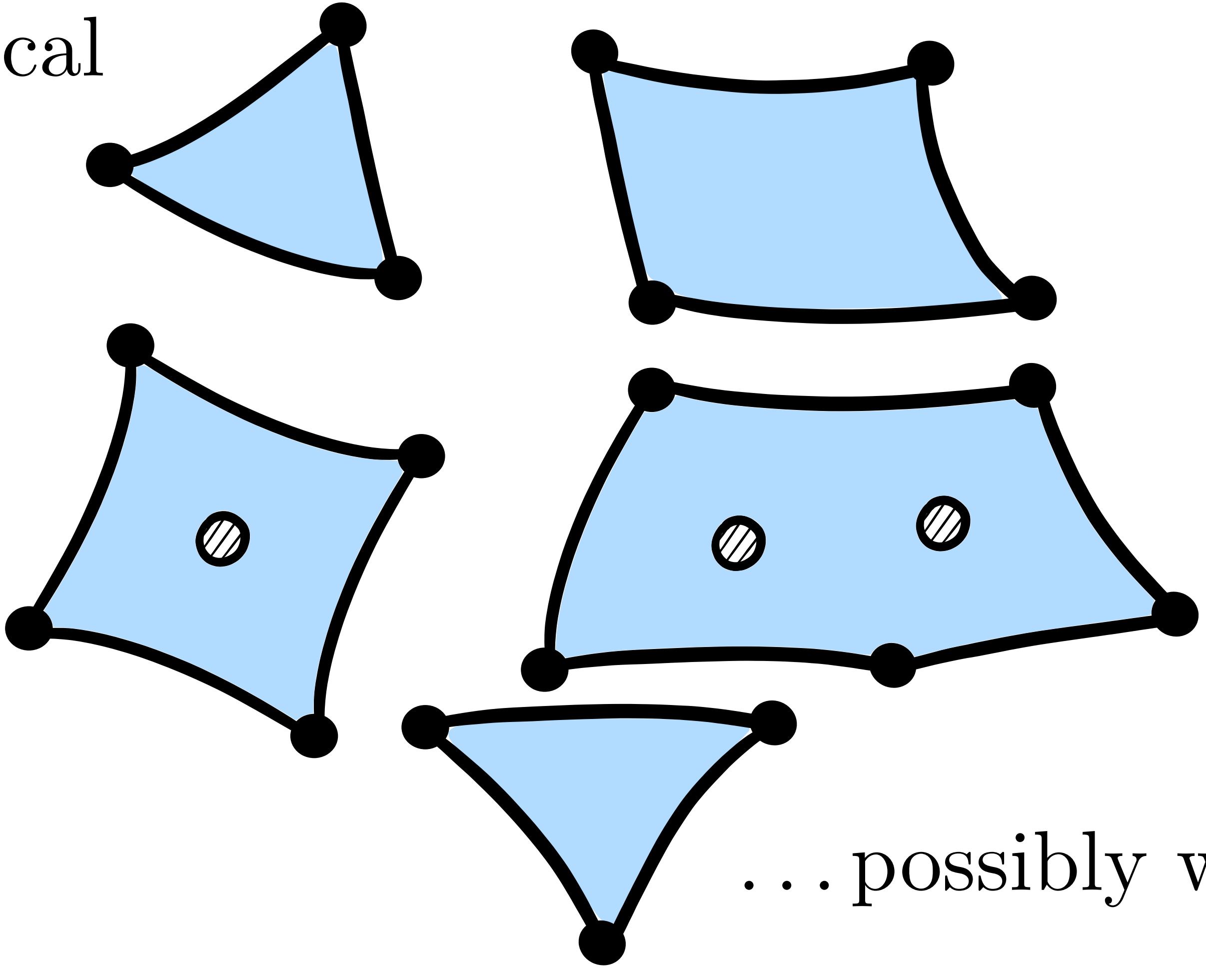
Input

take topological polygons. . .



Input

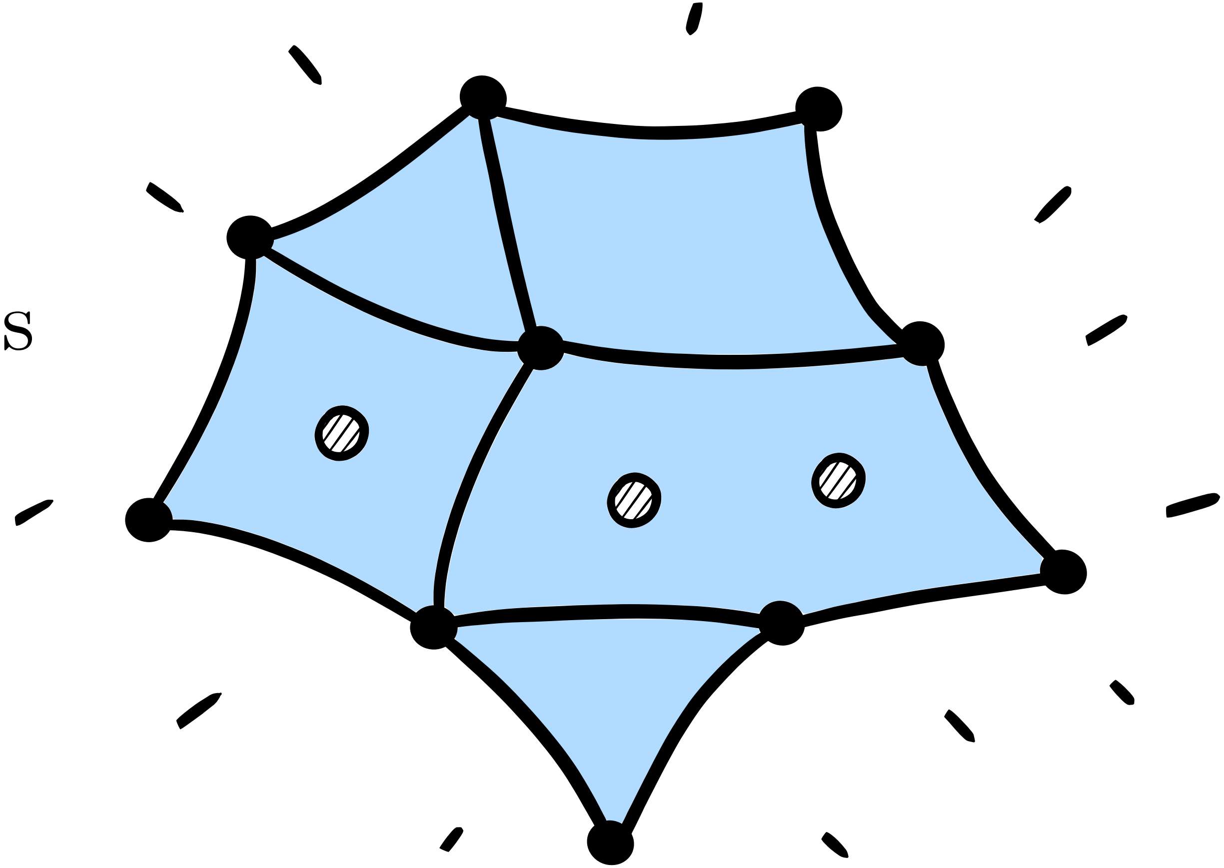
take topological polygons…



…possibly with holes

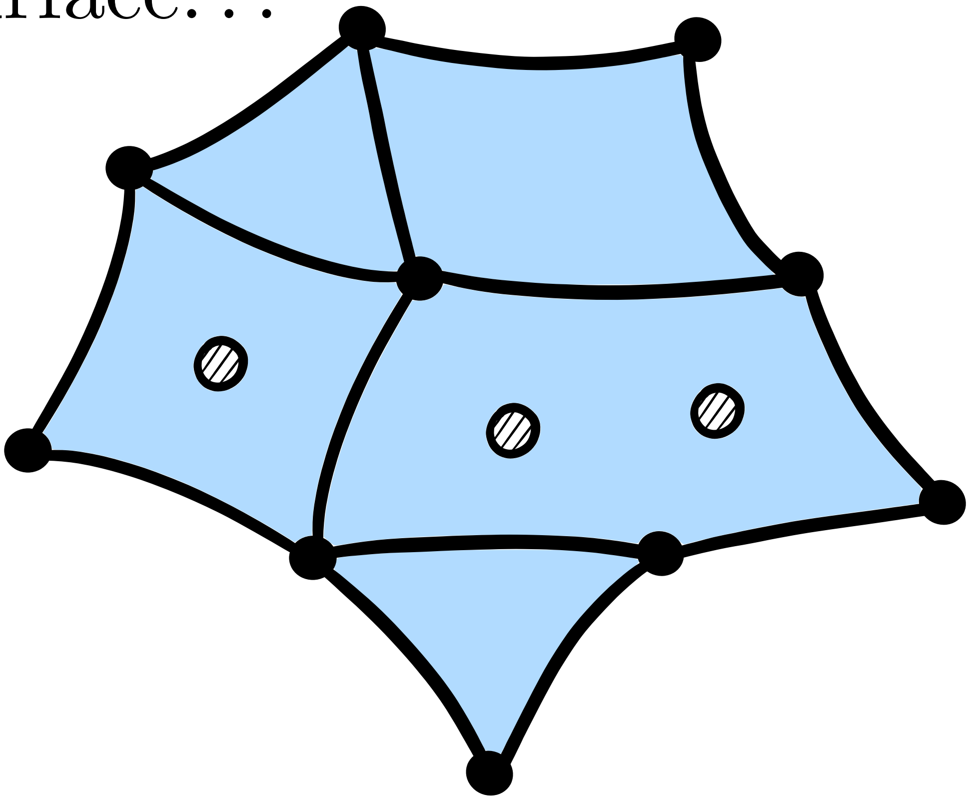
Input

glue edges



Input

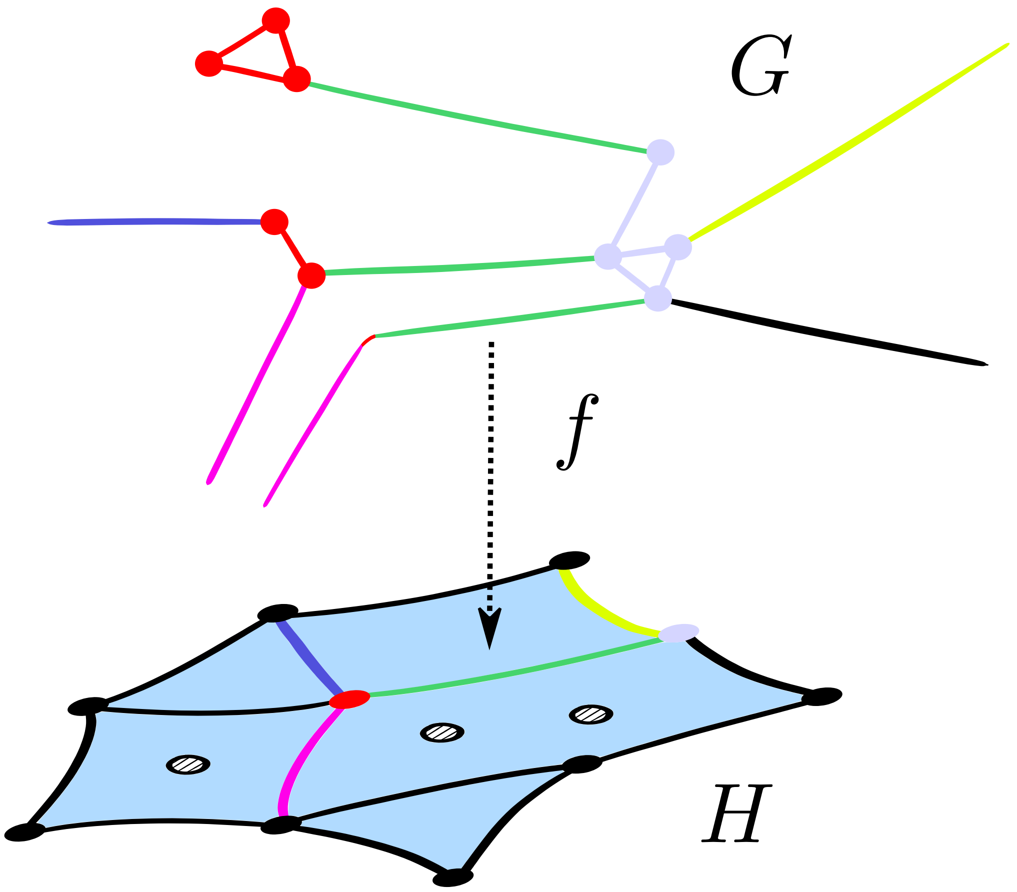
that encodes the surface. . .



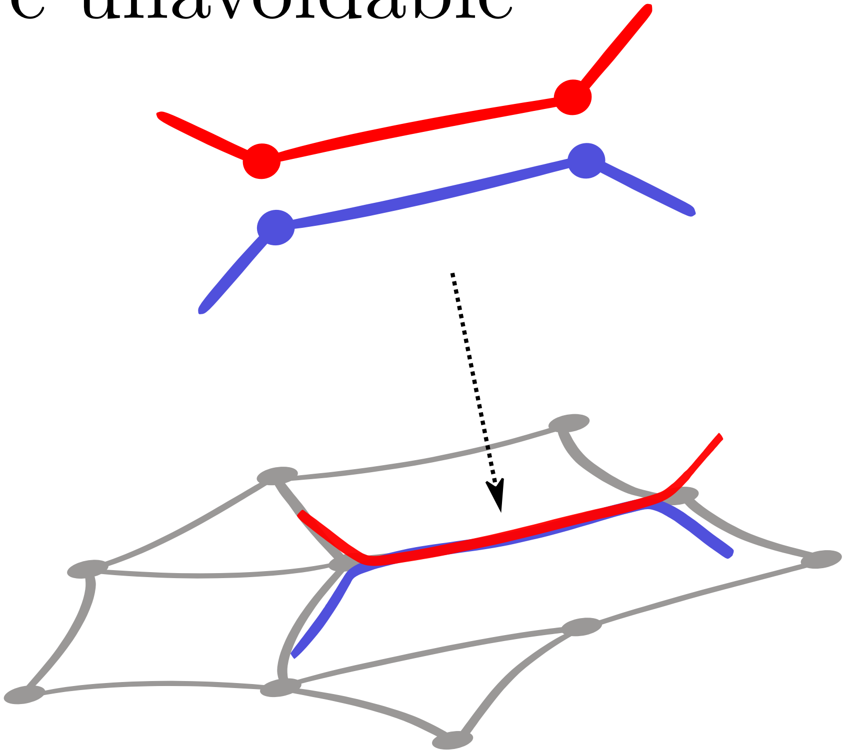
. . . and a graph H on it

Input

draw G in H

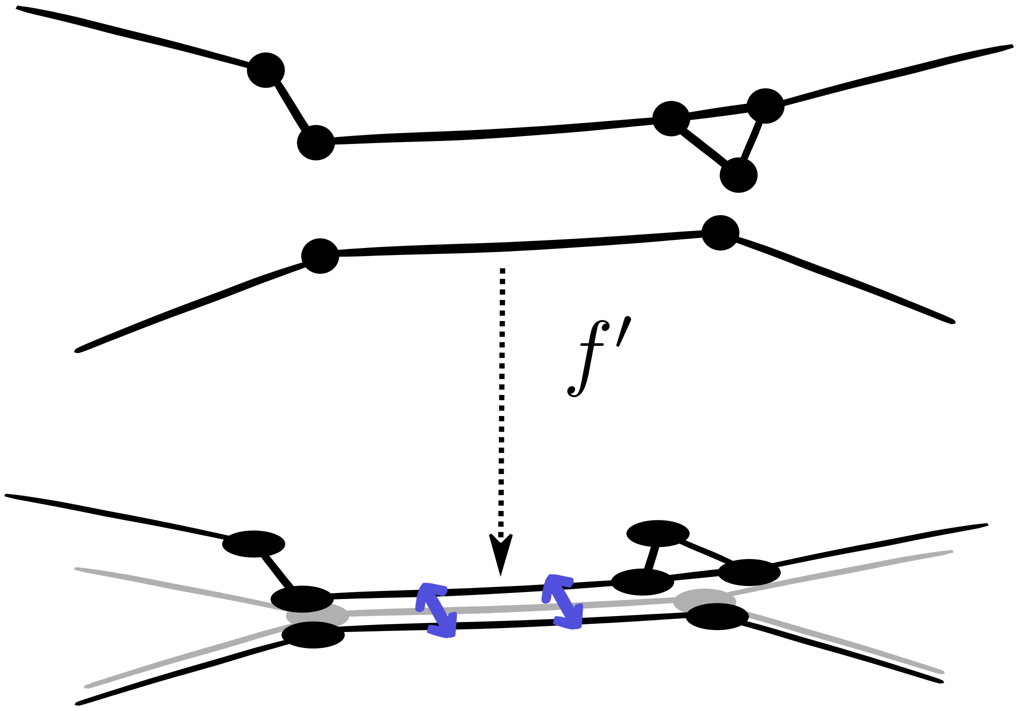


In this model overlaps
are unavoidable



Output

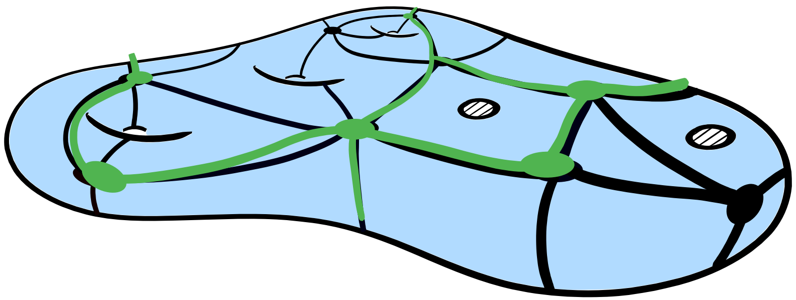
weak embedding: drawing f' that can be untangled by infinitesimal perturbation



Akitaya, Fulek, and Tóth, 2019

algo to determine if f' is weak embedding, and if so to perturb f'

Result



n : # times f uses an edge or vertex of H

m : # vertices and edges of H

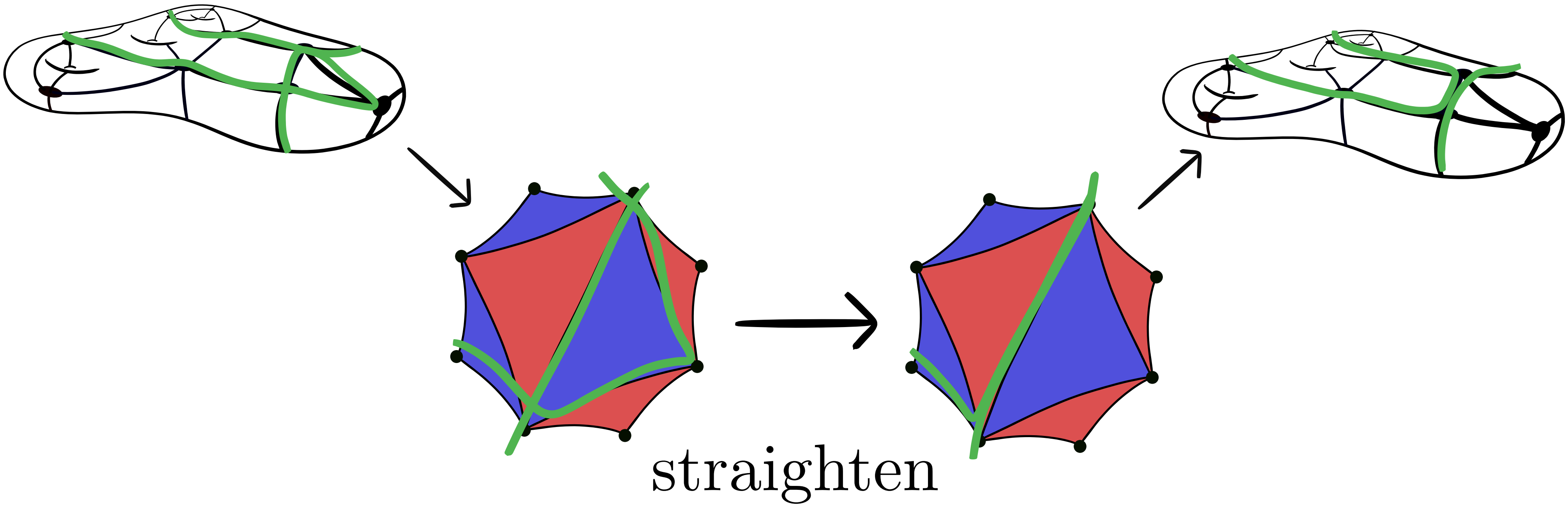
s : genus + # holes

Colin de Verdière, Despré, D., 2023

We can decide if f can be untangled, in $O(m + s^2 n \log(s\, n))$ time.
If so, we can compute a weak embedding homotopic to f in additional $O(s^2 m\, n^2)$ time.

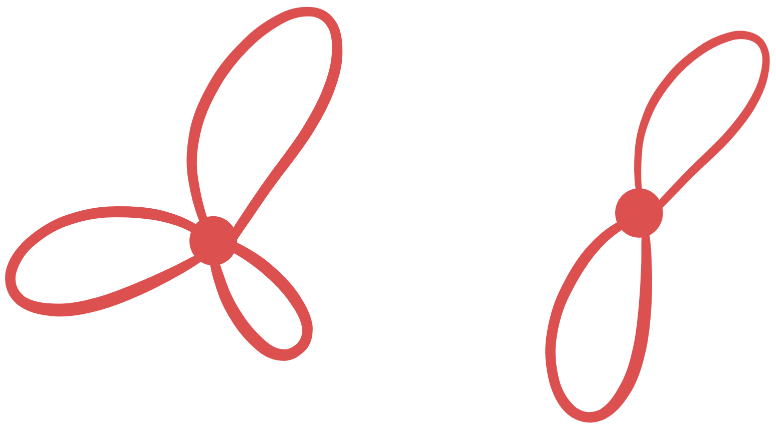
Algorithm overview

(assuming that the surface has no boundary)



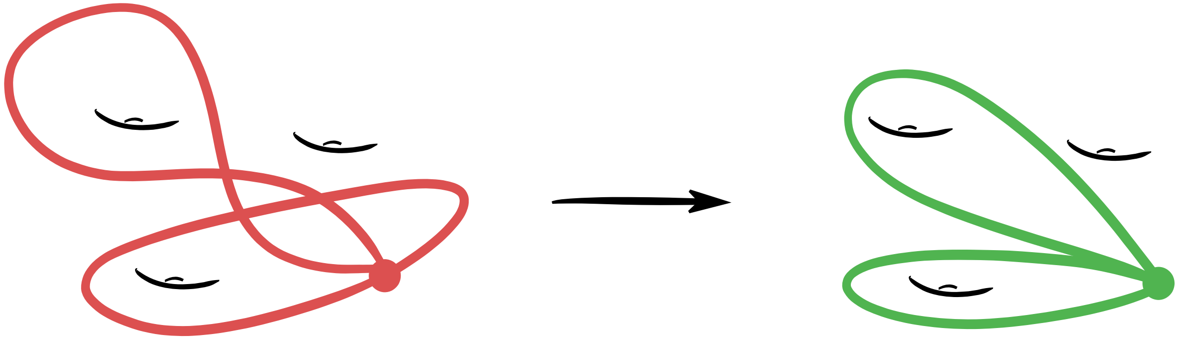
Straightening a graph

Straightening a ~~graph~~ loop graph



Straightening a loop graph

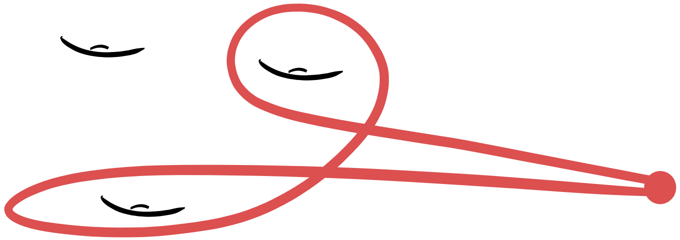
First attempt: reduce the loops, vertex fixed



Straightening a loop graph

First attempt: reduce the loops, vertex fixed

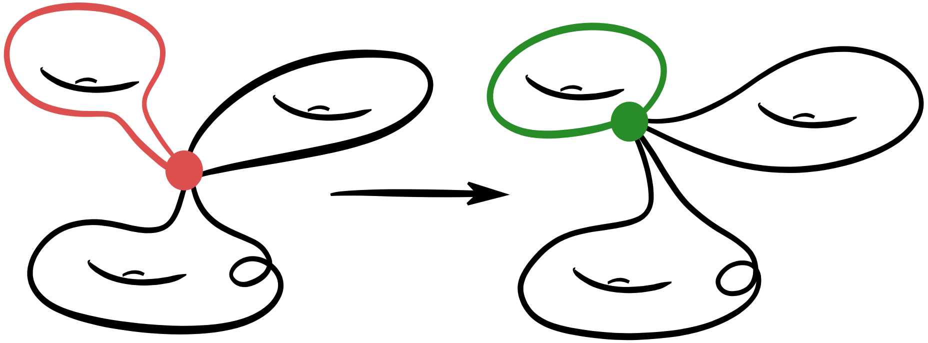
Problem:



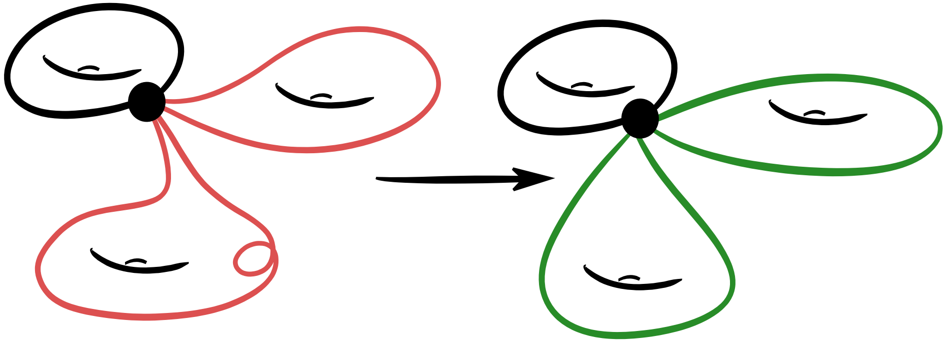
→ vertex must move

Straightening a loop graph

Solution:



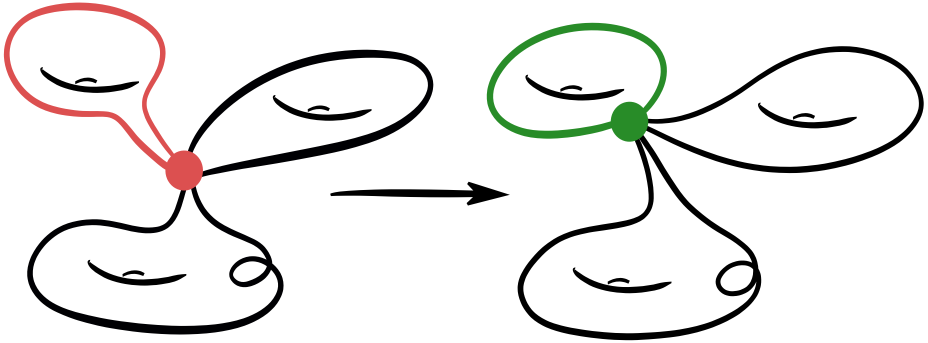
1. Reduce 1 loop cyclically



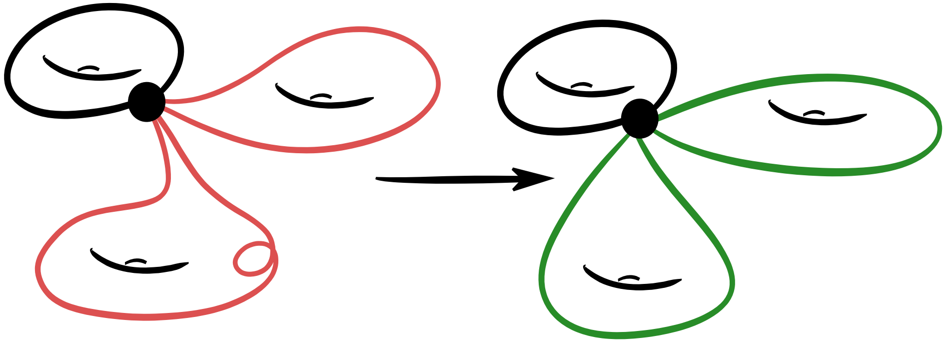
2. Reduce the other loops linearly

Straightening a loop graph

A straightened loop graph is a weak embedding
or cannot be untangled



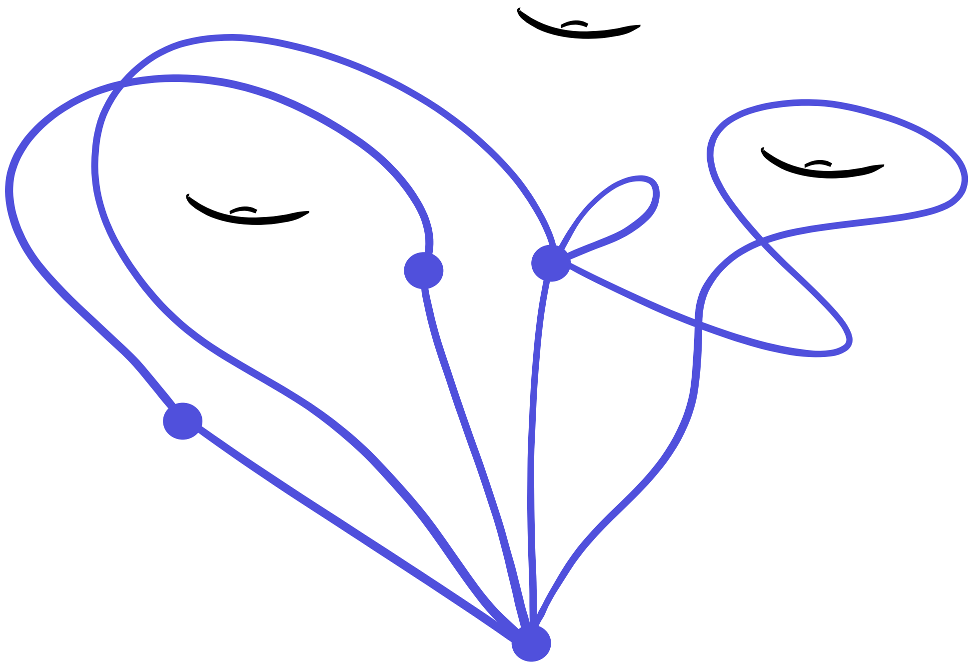
1. Reduce 1 loop cyclically



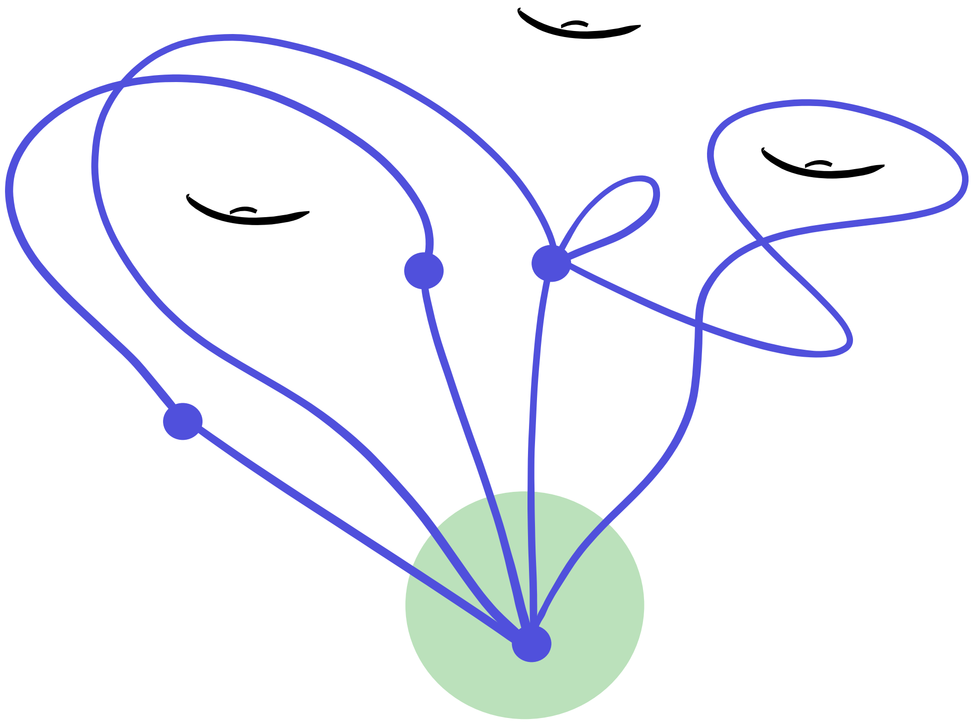
2. Reduce the other loops linearly

Straightening a graph

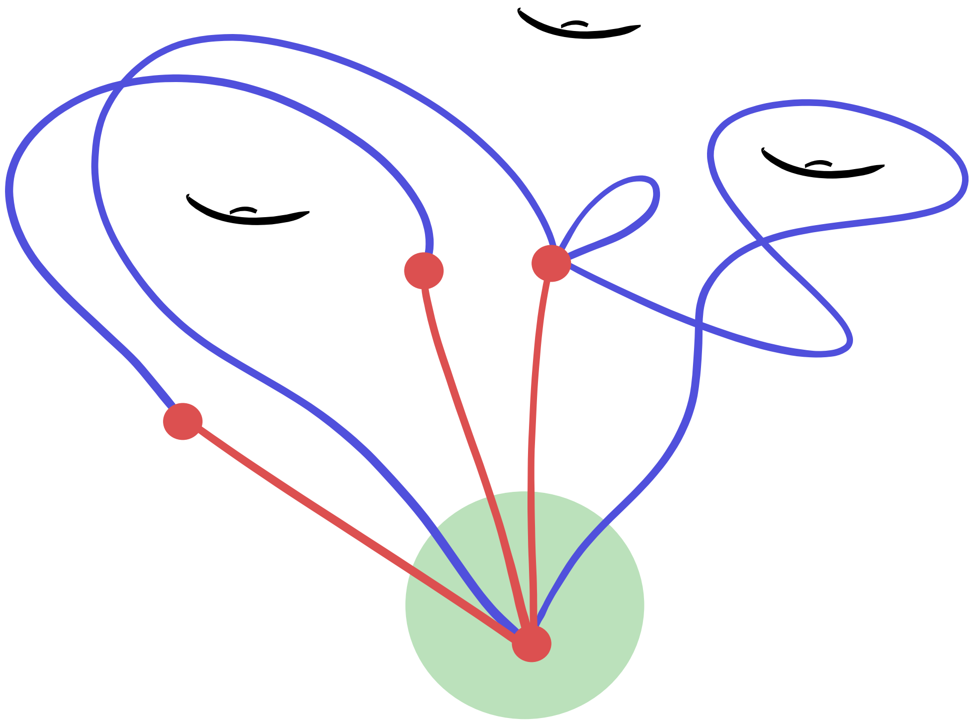
Straightening a graph



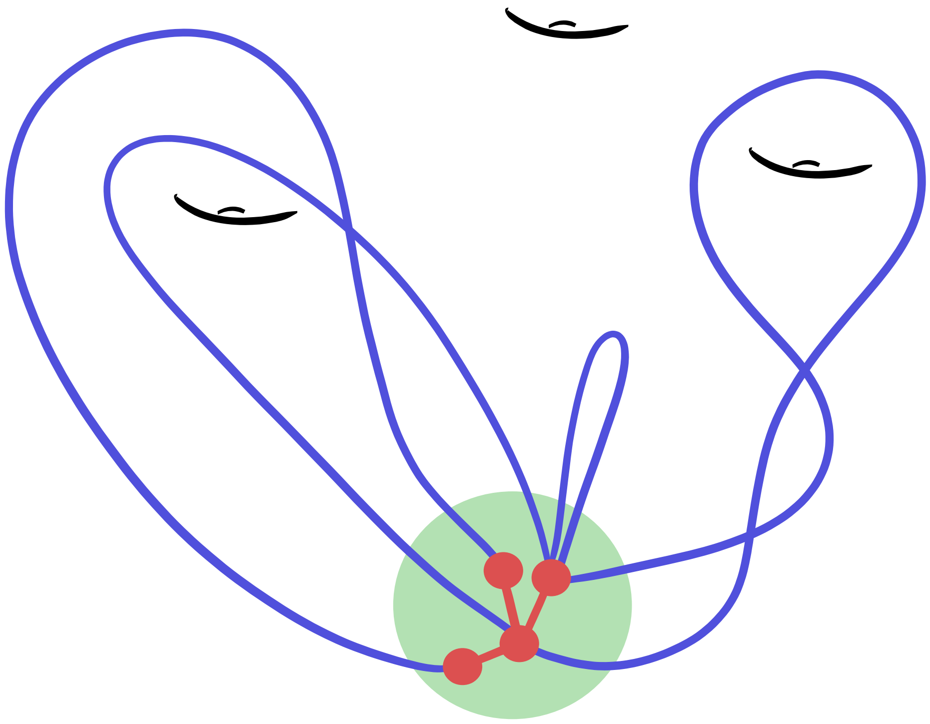
Straightening a graph



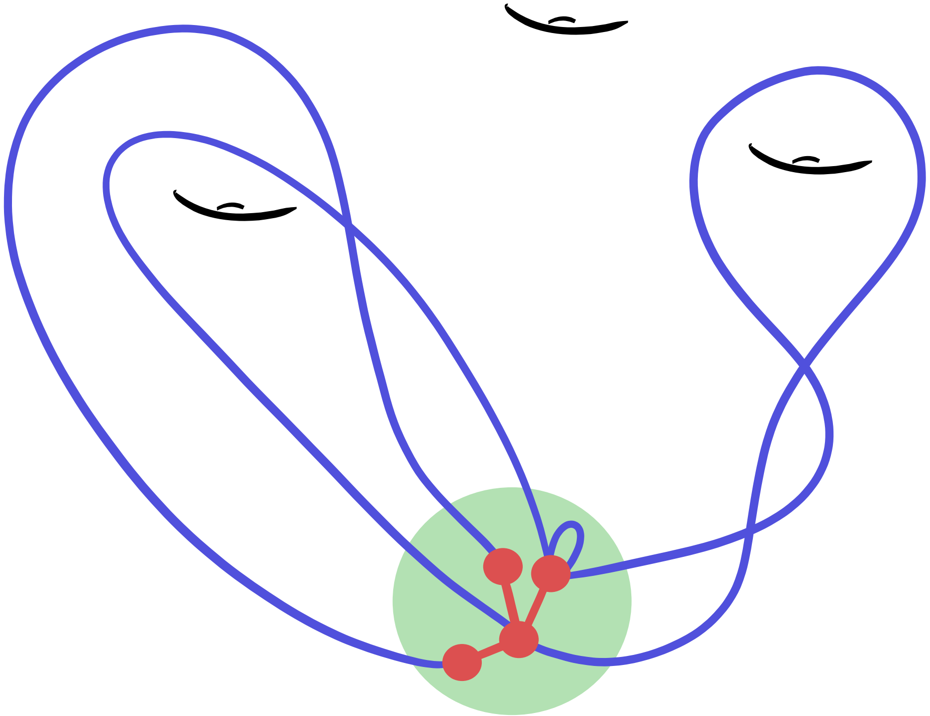
Straightening a graph



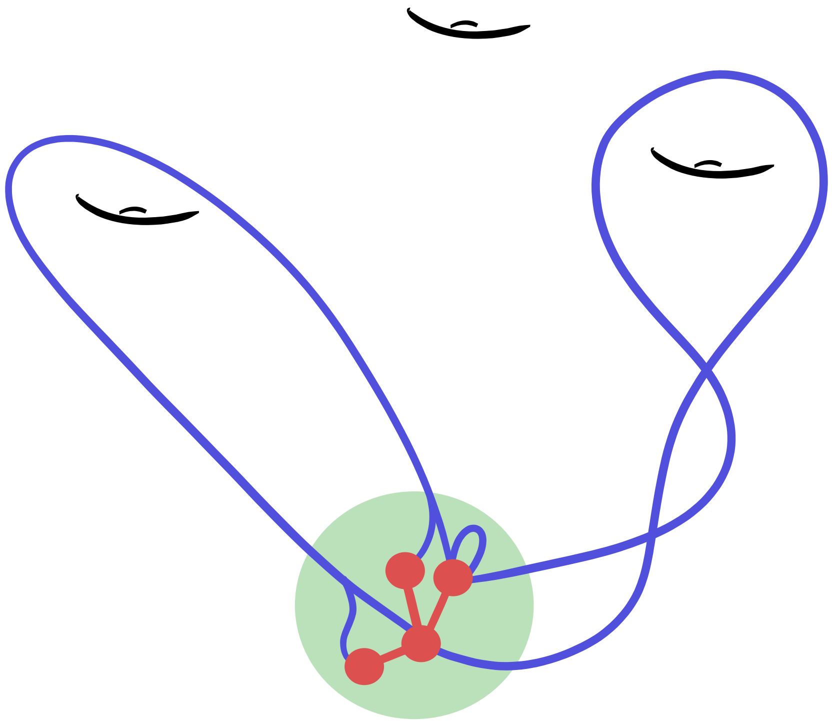
Straightening a graph



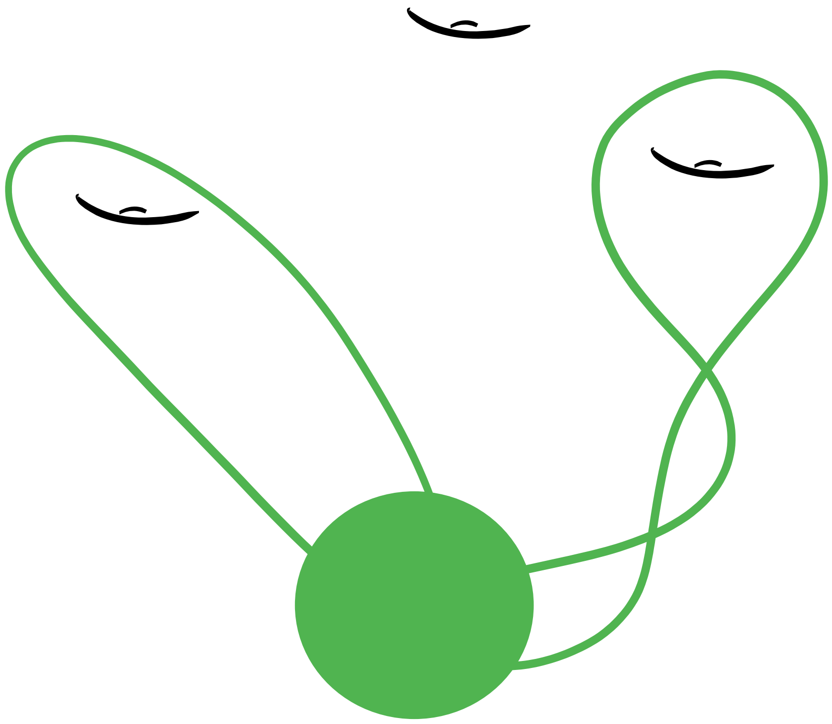
Straightening a graph



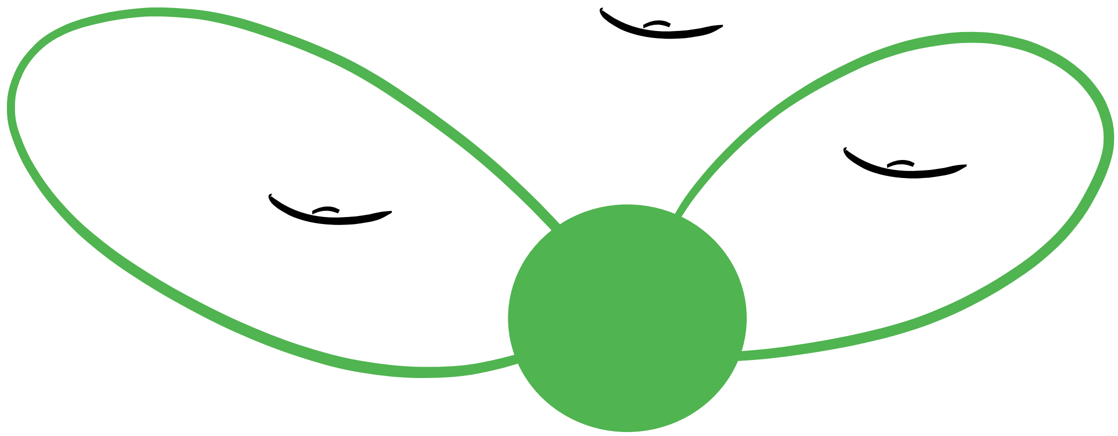
Straightening a graph



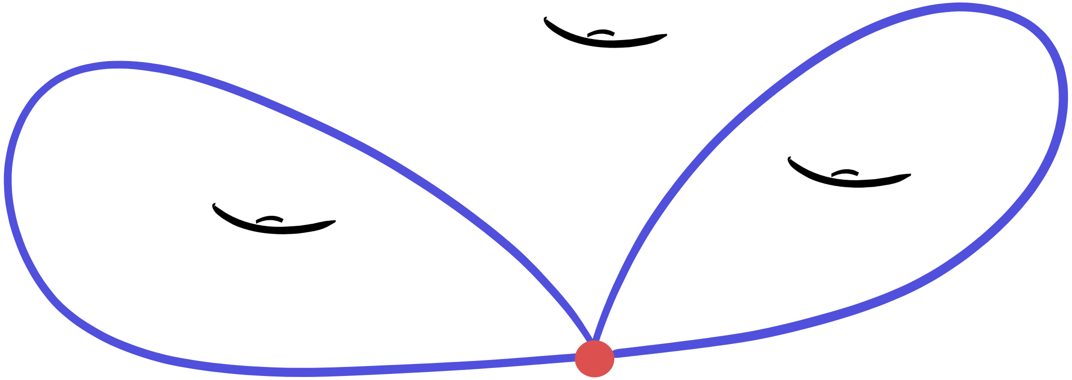
Straightening a graph



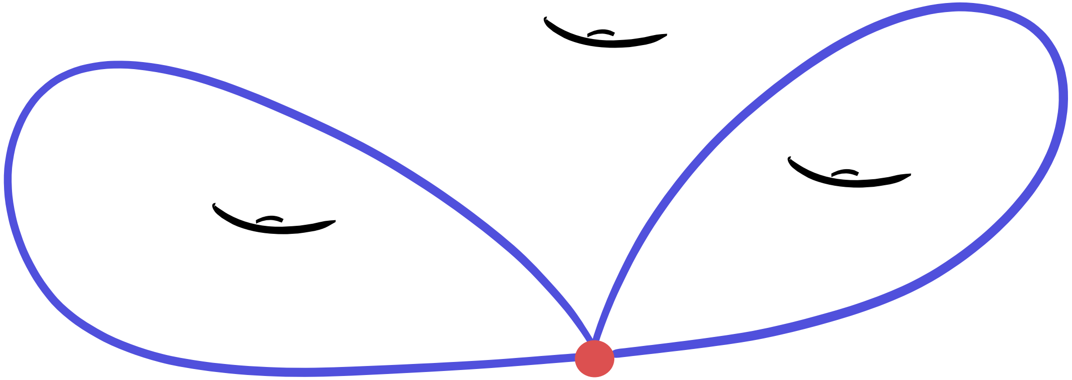
Straightening a graph



Straightening a graph

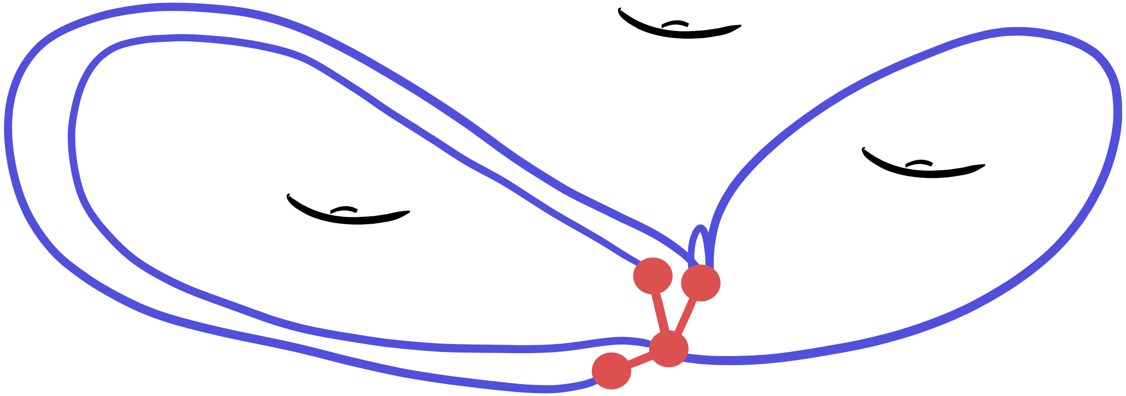


Straightening a graph



A straightened graph is a weak embedding or cannot be untangled

Straightening a graph

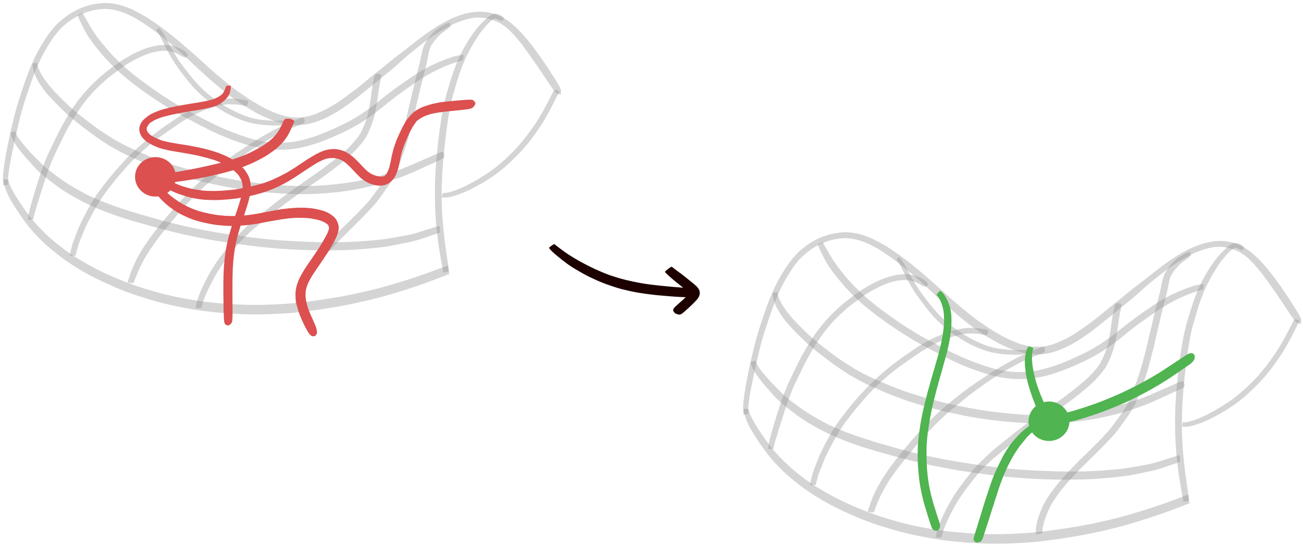


A straightened graph is a weak embedding or cannot be untangled

(+ tricks and data structures for achieving
the announced time complexities)

Discrete analogue of Tutte embeddings

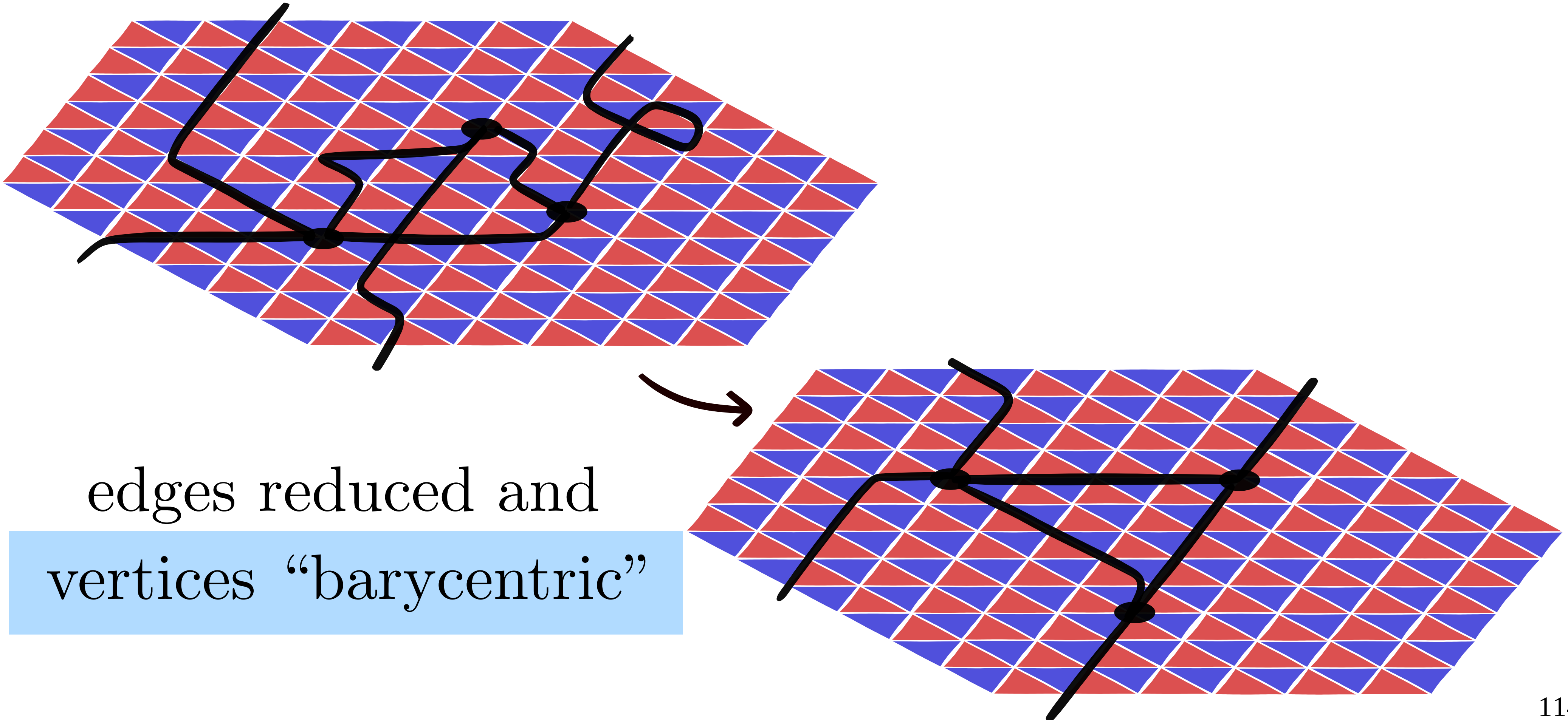
Recall: Tutte drawings



edges straight and

vertices “barycentric”

Harmonious drawings

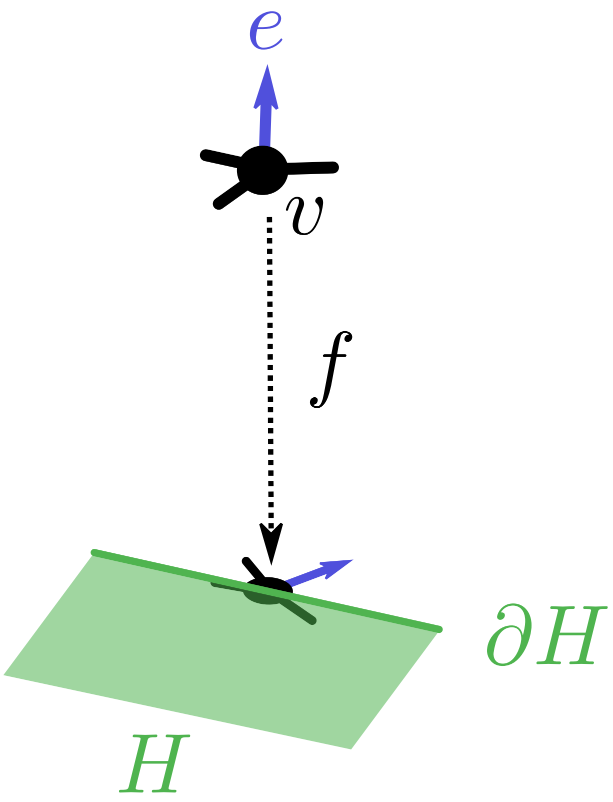


edges reduced and

vertices "barycentric"

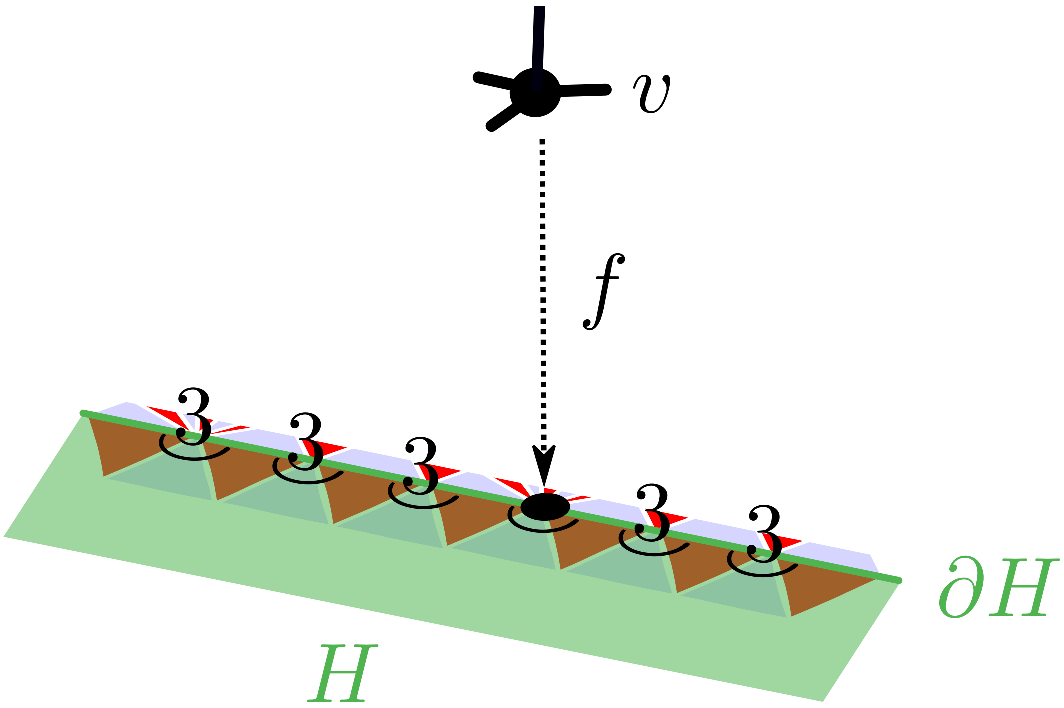
vertices "barycentric"

$$f(v) \in \partial H \\ \Longrightarrow \\ \exists e \; f(e) \text{ escapes } H$$



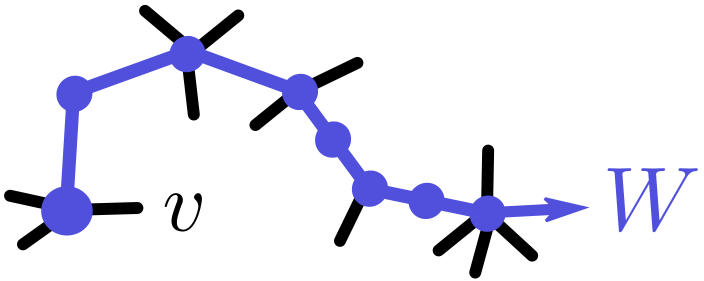
vertices "barycentric"

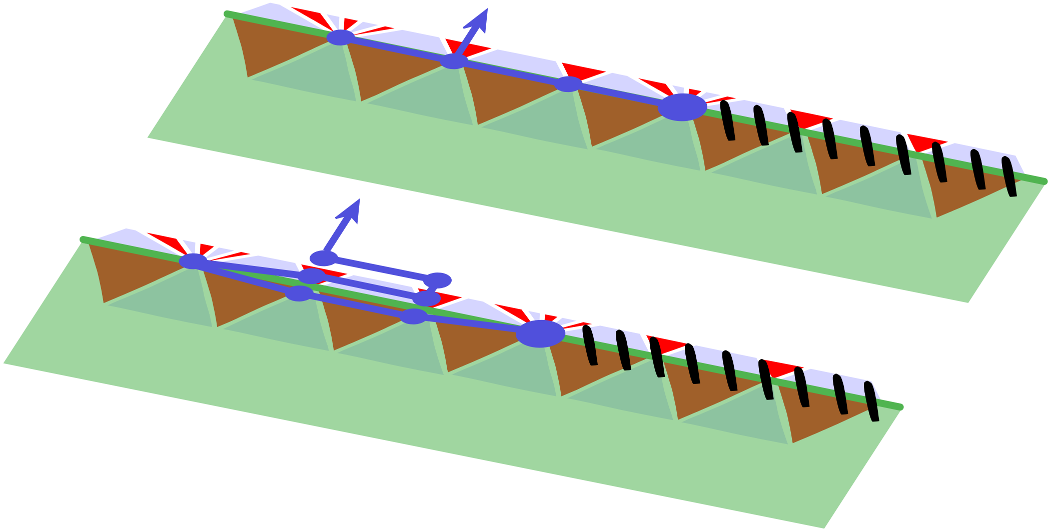
$f(v) \in \partial H$

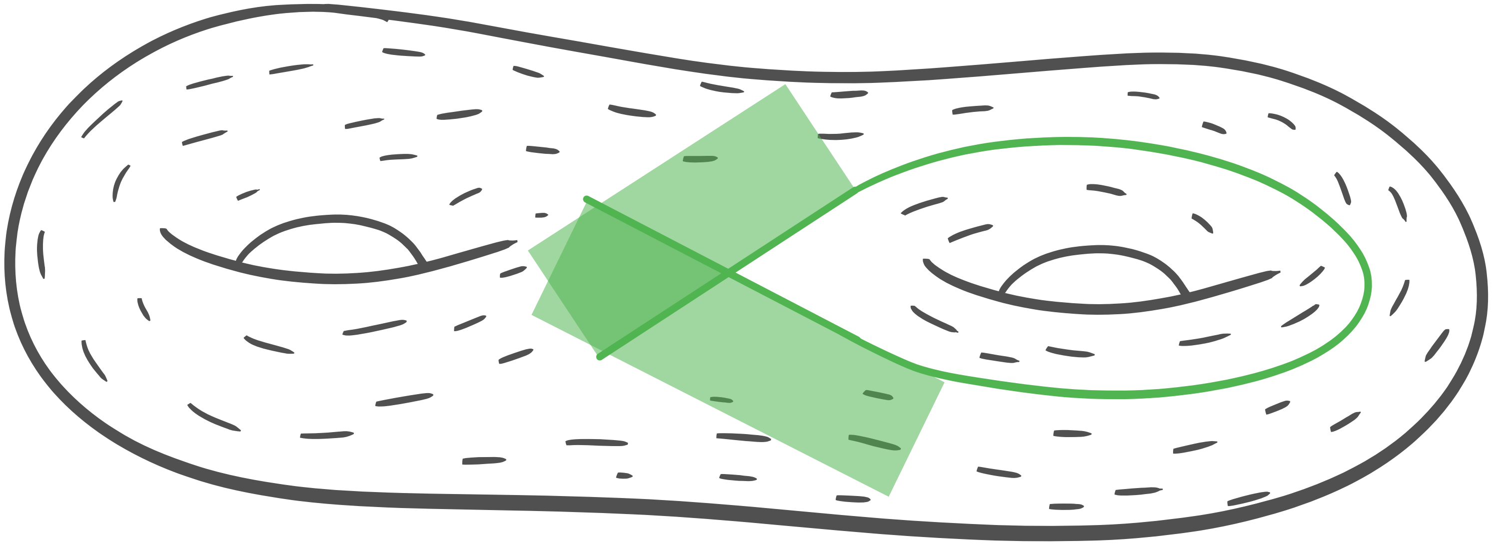


vertices “barycentric”

$$f(v) \in \partial H \implies \exists W \; f \circ W \text{ “escapes” } H$$







this definition generalizes to surfaces

Results

graph G
reducing triangulation T with m edges
$f : G \to T^1$ of size n

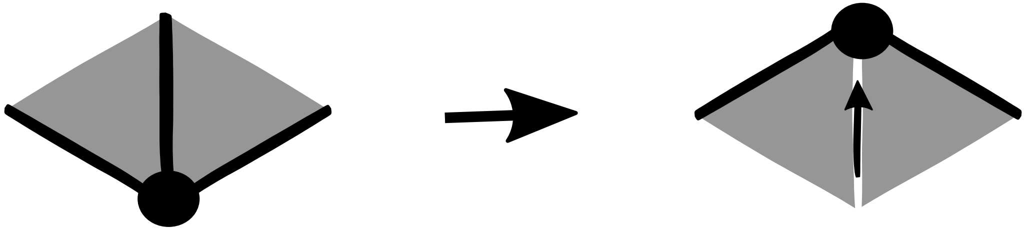
Definition of *harmonious* drawings

f harmonious and f can be untangled
$\Rightarrow f$ weak embedding

Algo to make f harmonious in $O((m+n)^2 n^2)$ time,
without increasing any edge length

Harmonizing a drawing

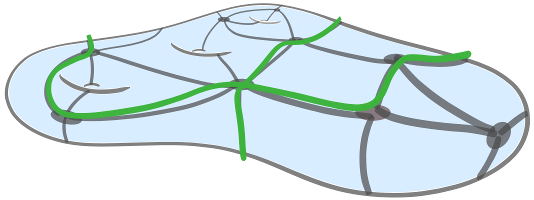
1 we define monotonic moves to apply to f

ex: 

2 some moves do not seem to decrease any potential so we combine the moves carefully

Making curves cross minimally

Related work

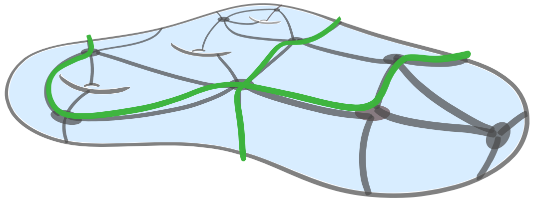


closed walks of total length n
on a graph of size m

Despré, Lazarus, 2019

- Put a single curve in minimal position in $O(m + n^4)$ time
- Compute the min. nb. of crossings in $O(m + n^2)$ time

Result



closed walks of total length n
on a graph of size m

simpler algos and proofs!

$m^3 n + mn\log(mn)$

D., 2024

- Put ~~a single curve~~ in minimal position in $O(\cancel{m+n}^4)$ time
- Compute the min. nb. of crossings in $O(\cancel{m+n}^2)$ time

$m^2 + mn\log(mn)$

Untangling Graphs

Computing Delaunay Triangulations

Possible continuations

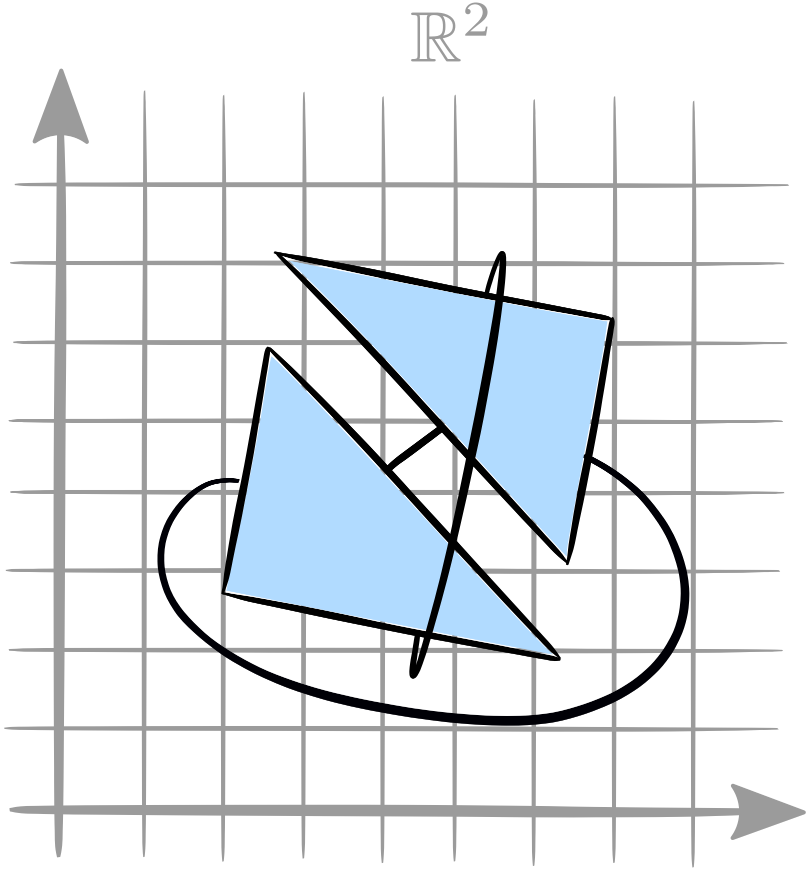
Untangling Graphs

Computing Delaunay Triangulations

Possible continuations

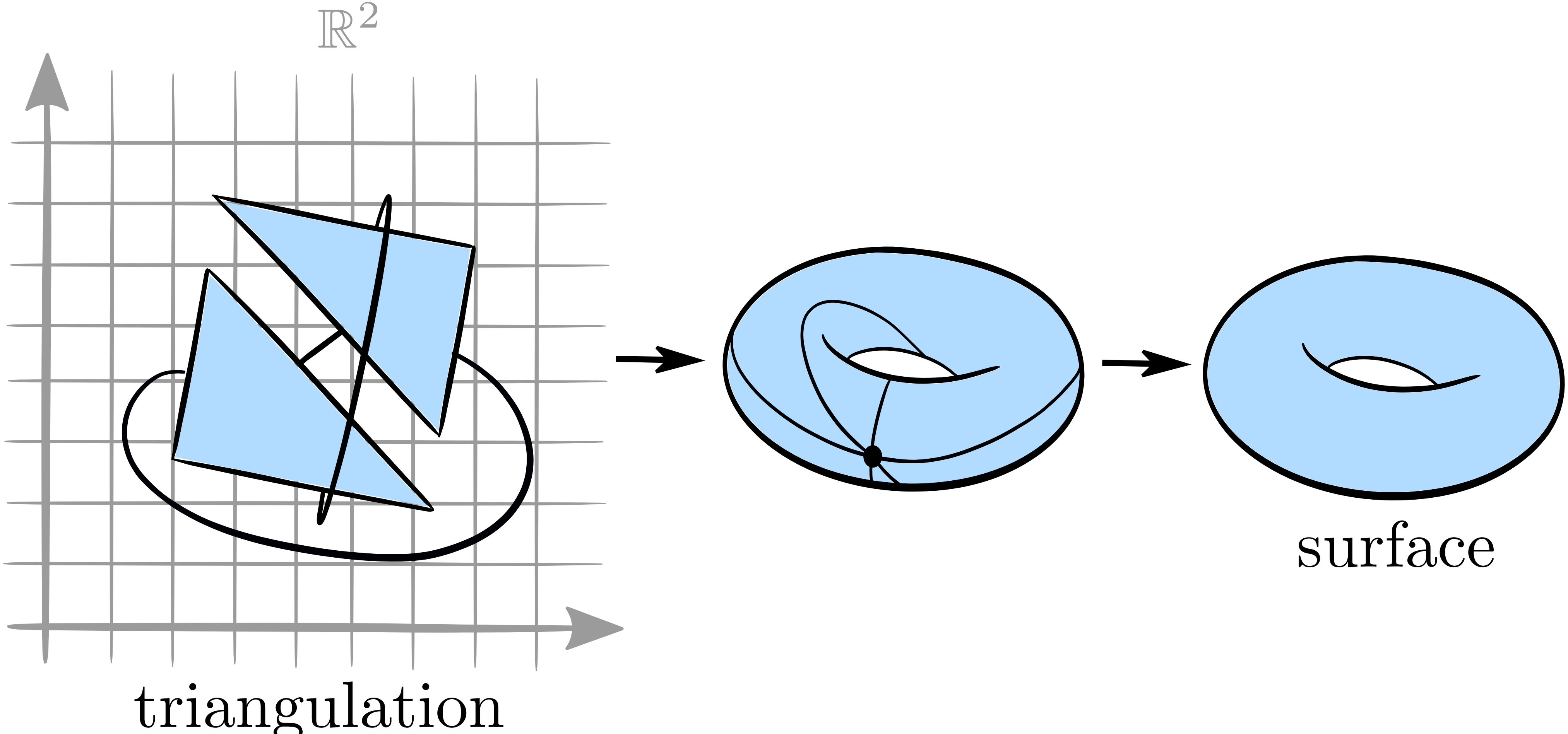
Triangulations of polyhedral surfaces

Triangulation of polyhedral surface

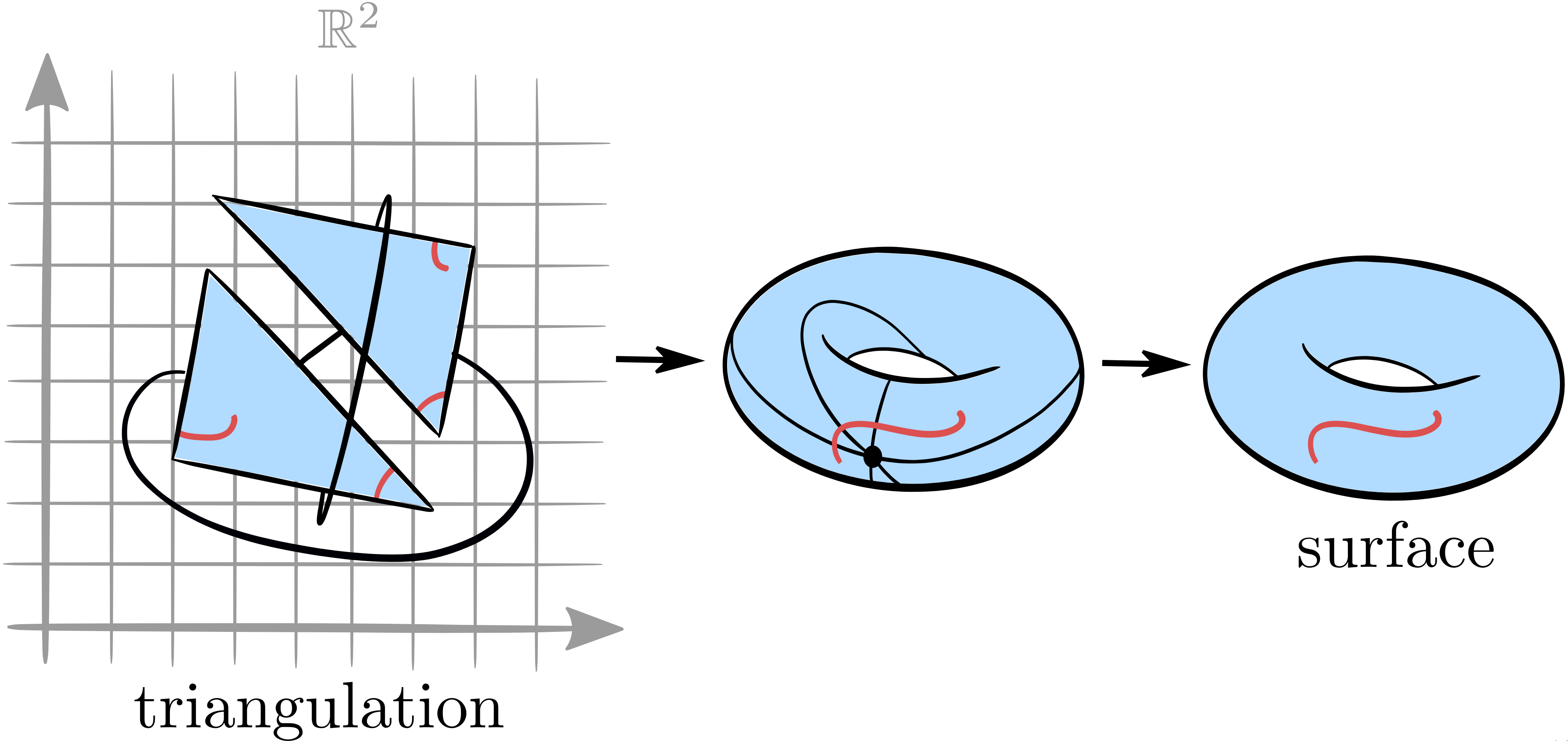


triangulation

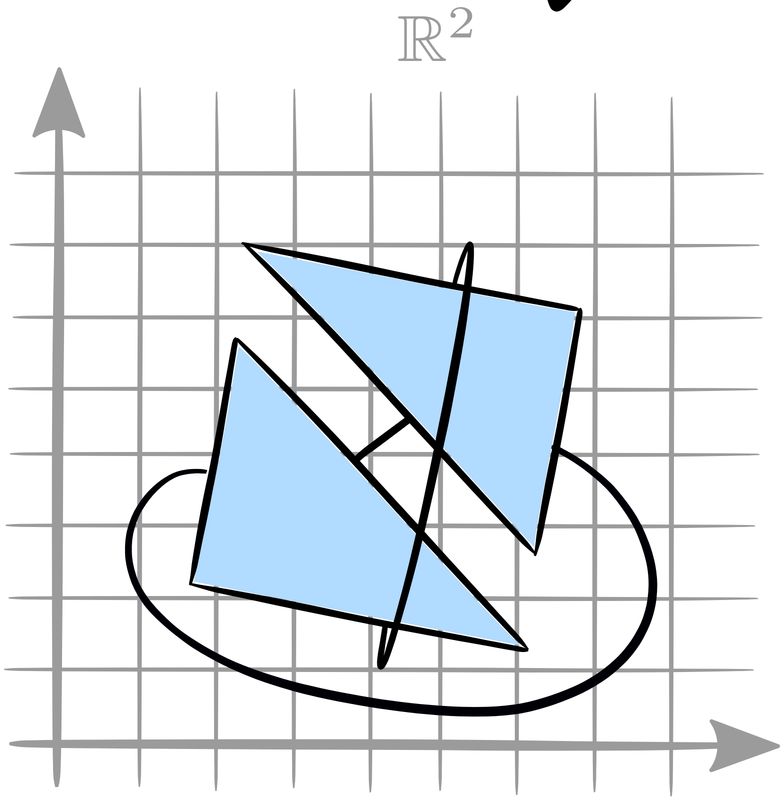
Triangulation of polyhedral surface

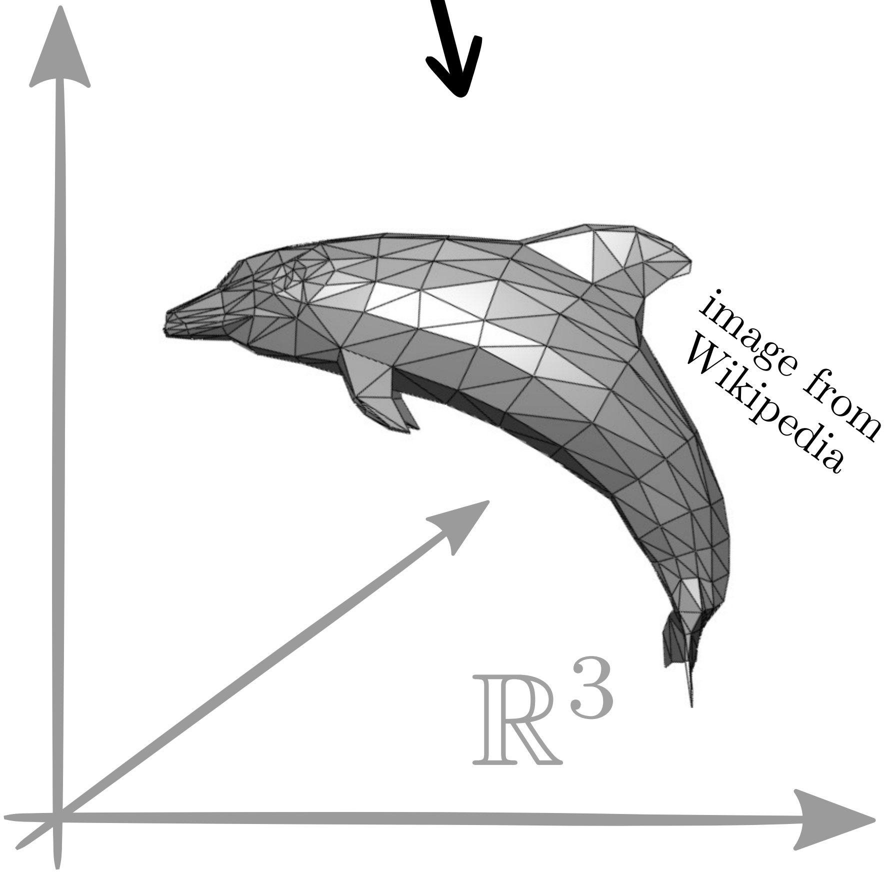


Triangulation of polyhedral surface

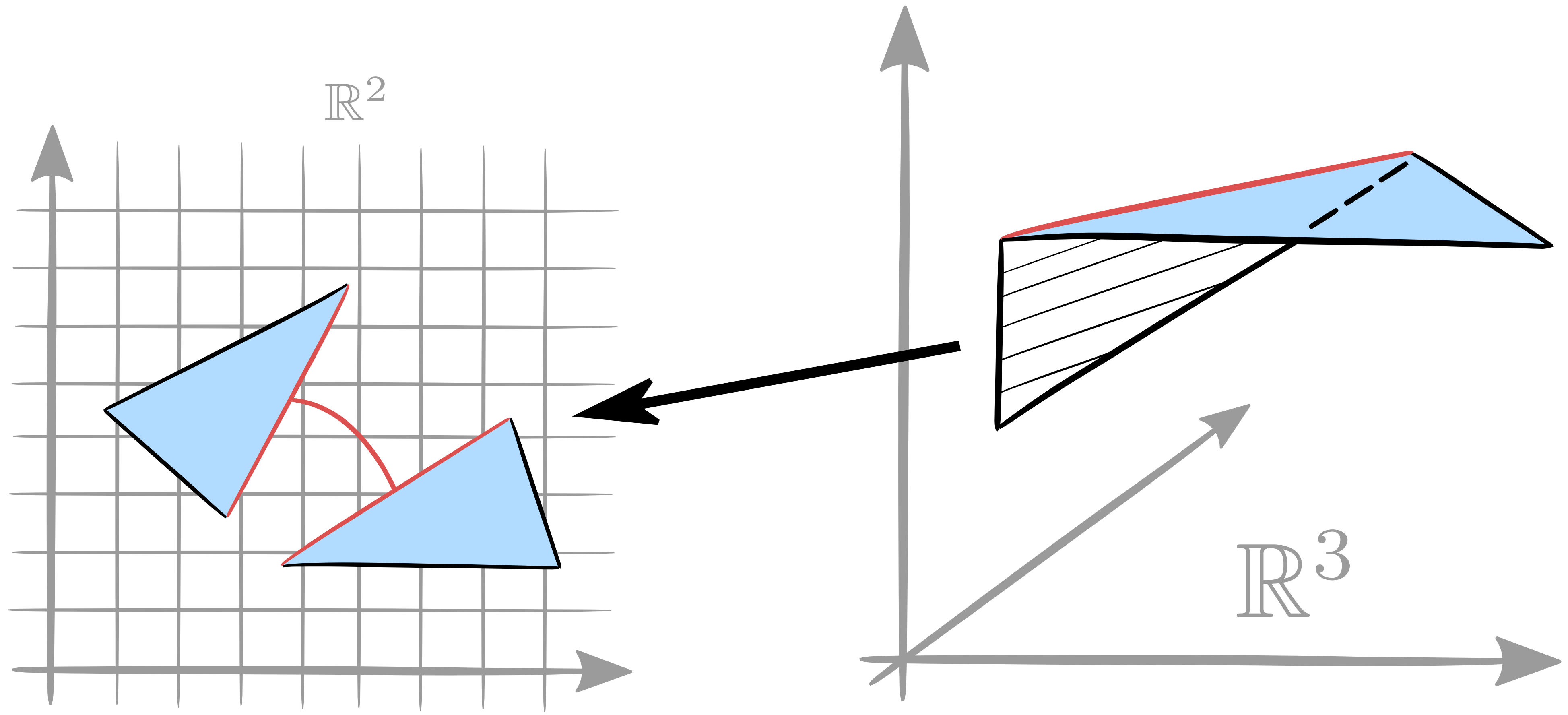


Triangulation vs. Mesh



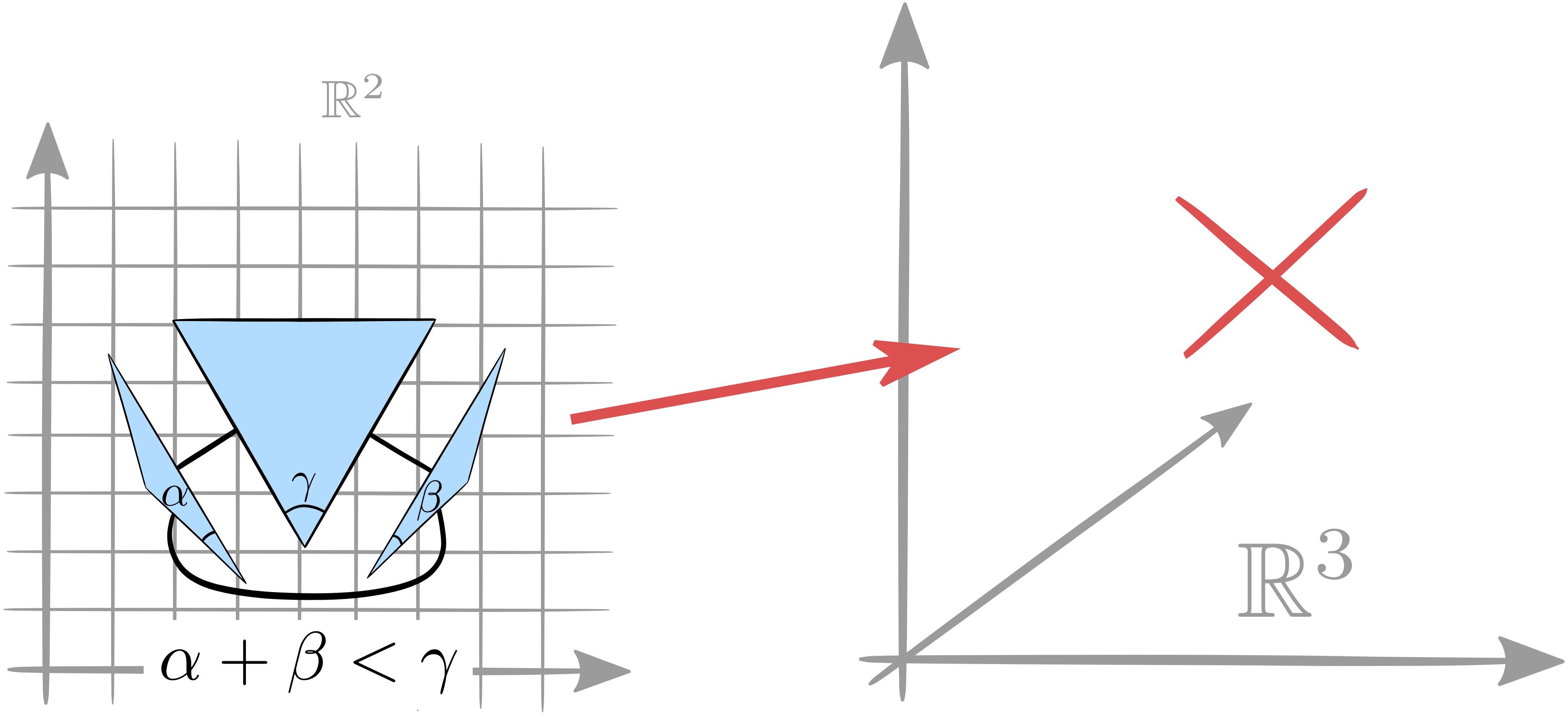


Every mesh gives a triangulation



Every mesh gives a triangulation

but converse is false!

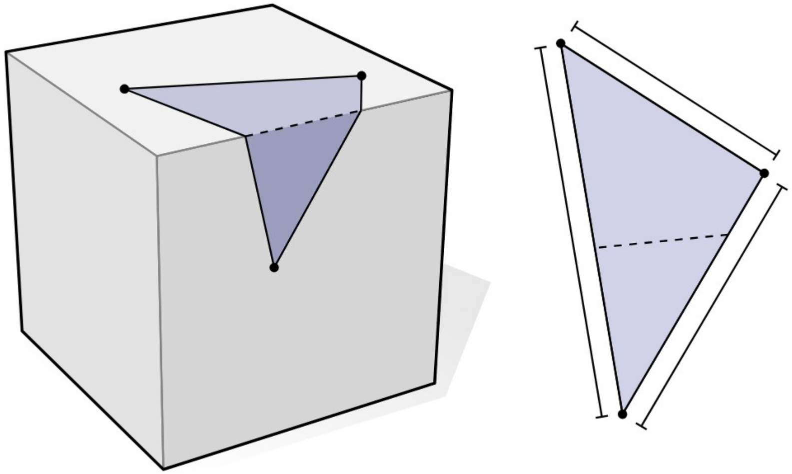


Some recent algos operate on triangulations not issued of a mesh

Liu et al., 2023

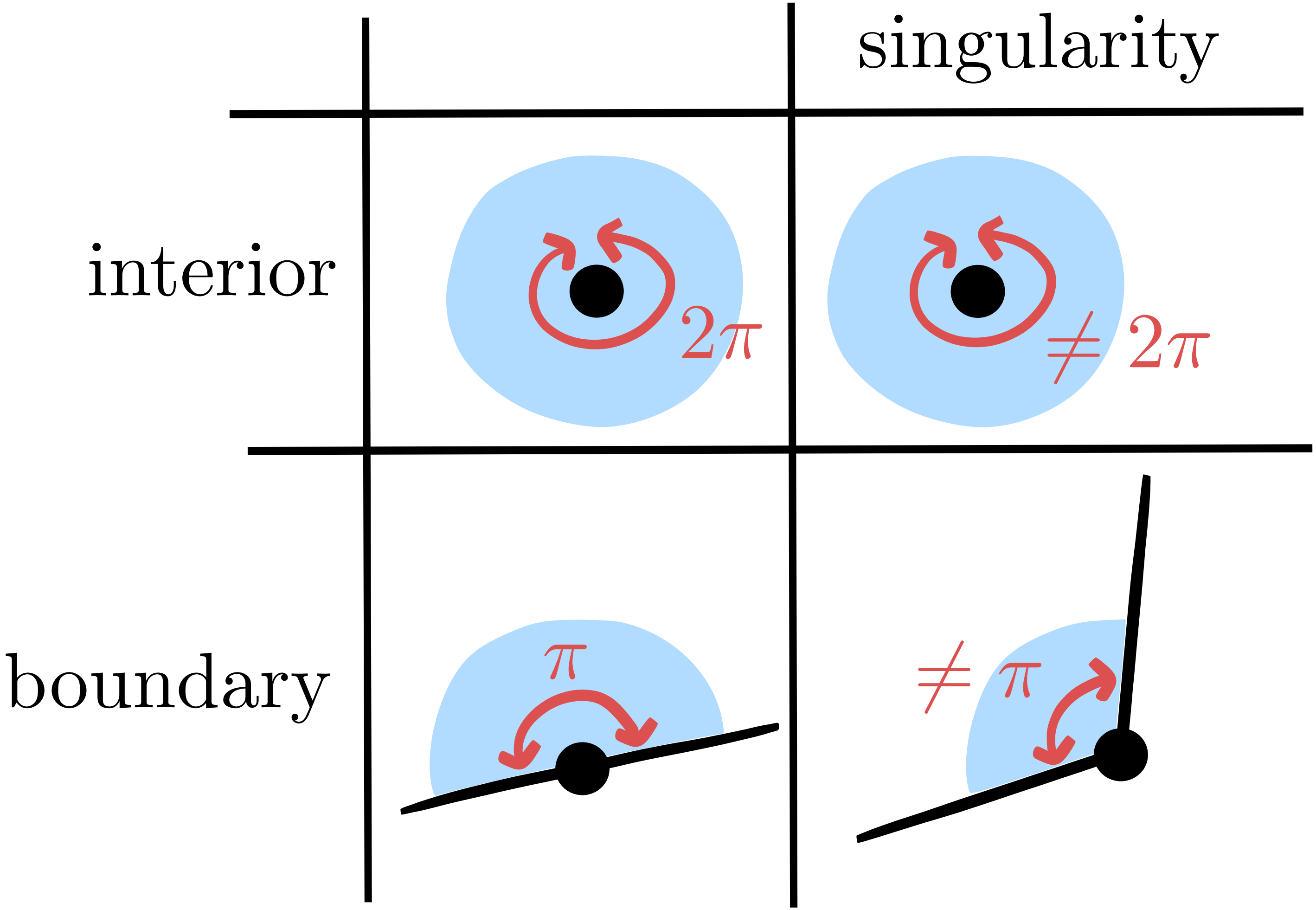
Takayama, 2022

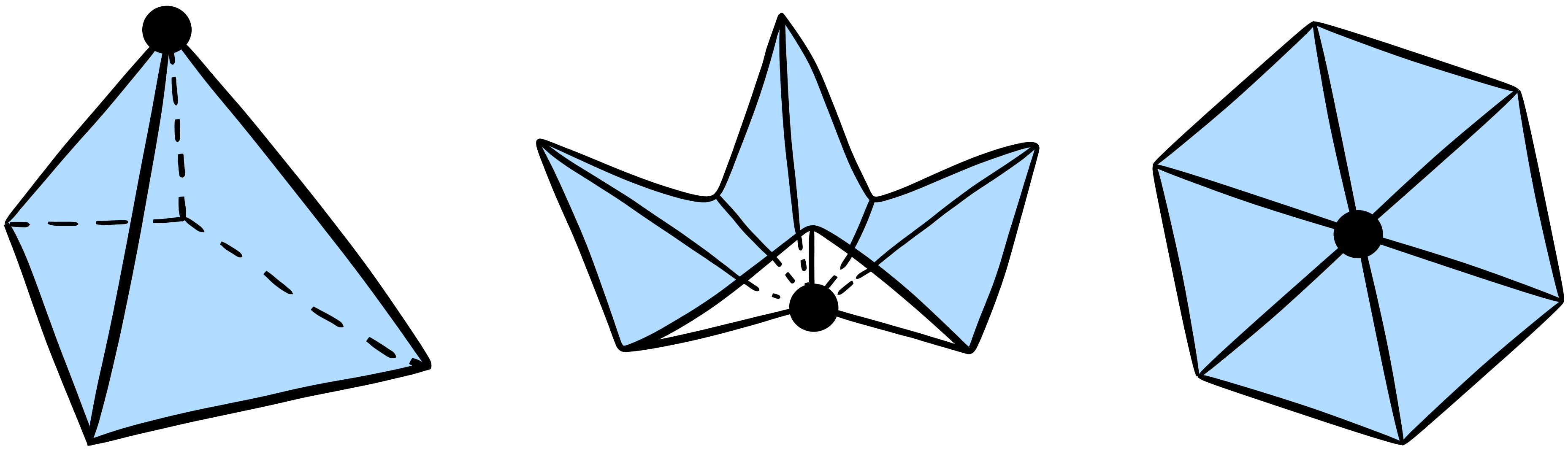
Sharp and Crane, 2020

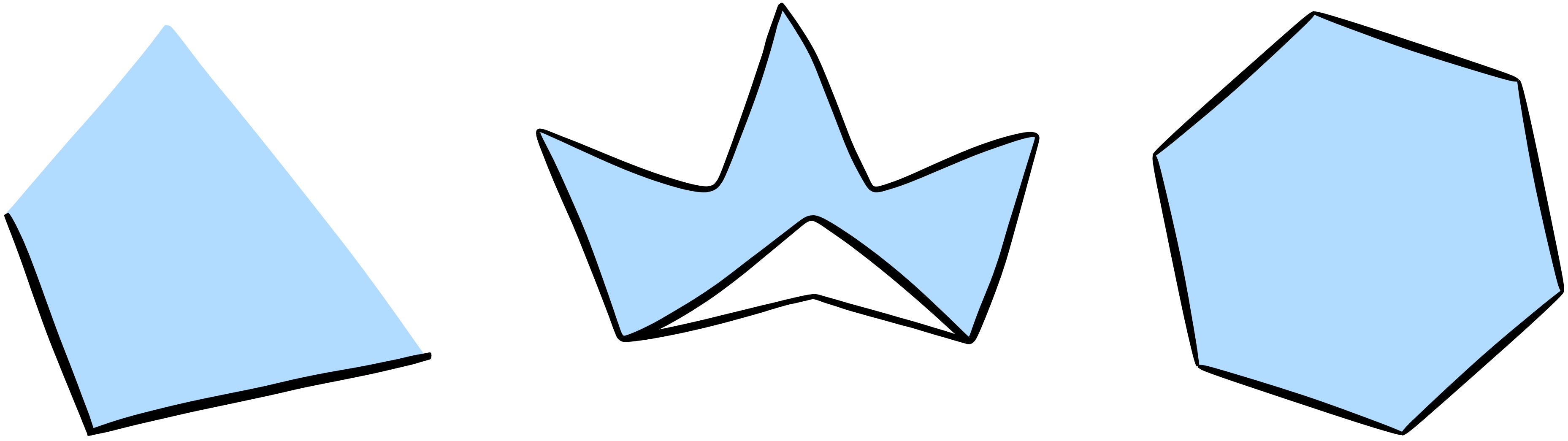


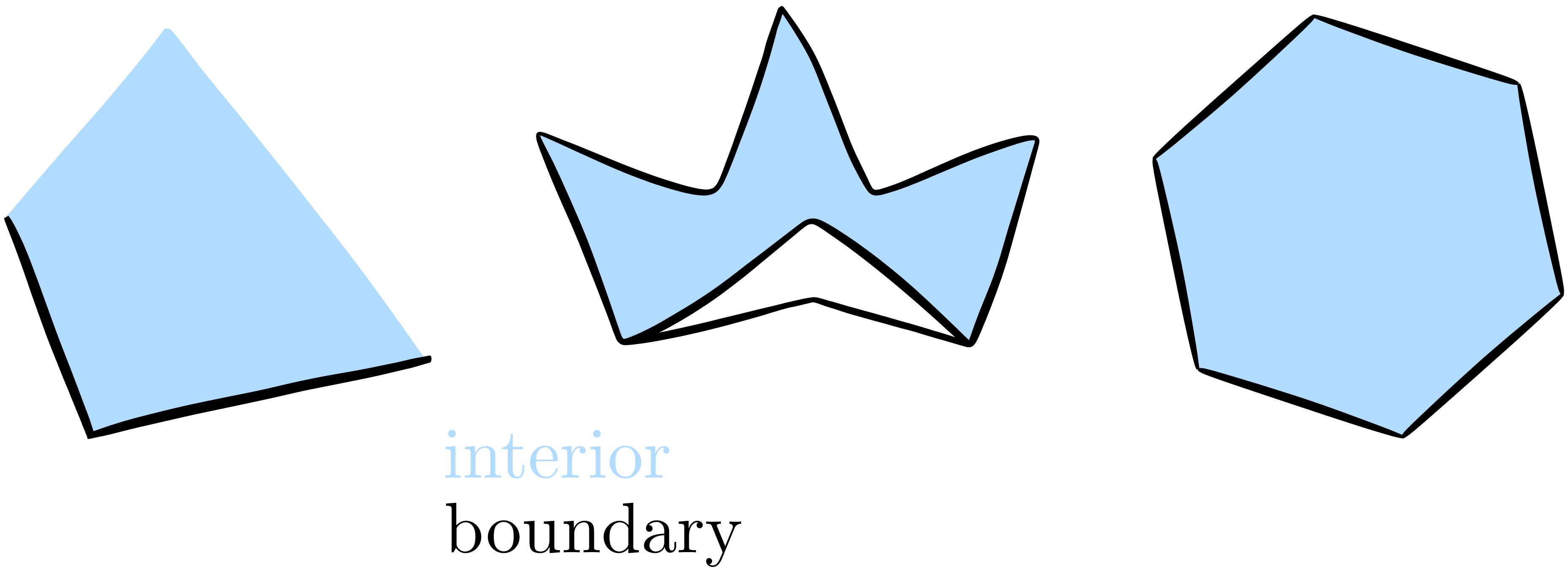
Geometry processing with intrinsic triangulations
Sharp, Gillespie, Krane, 2021

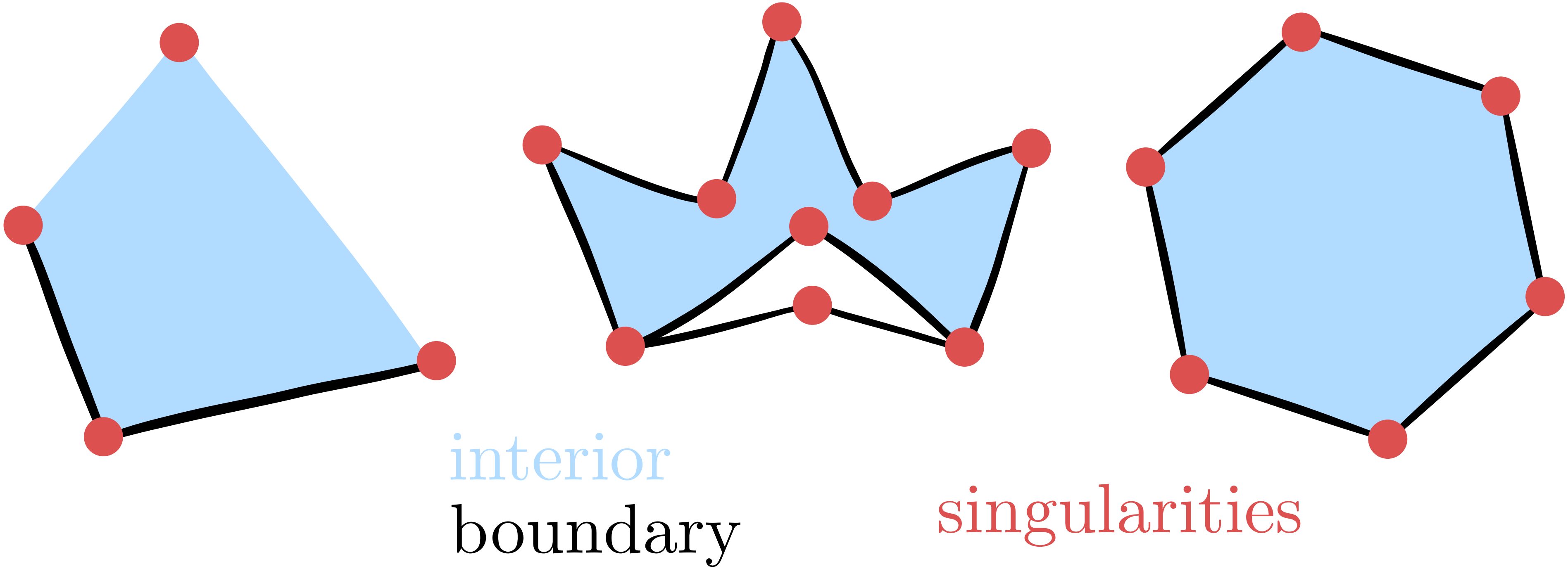
Types of points on the surface







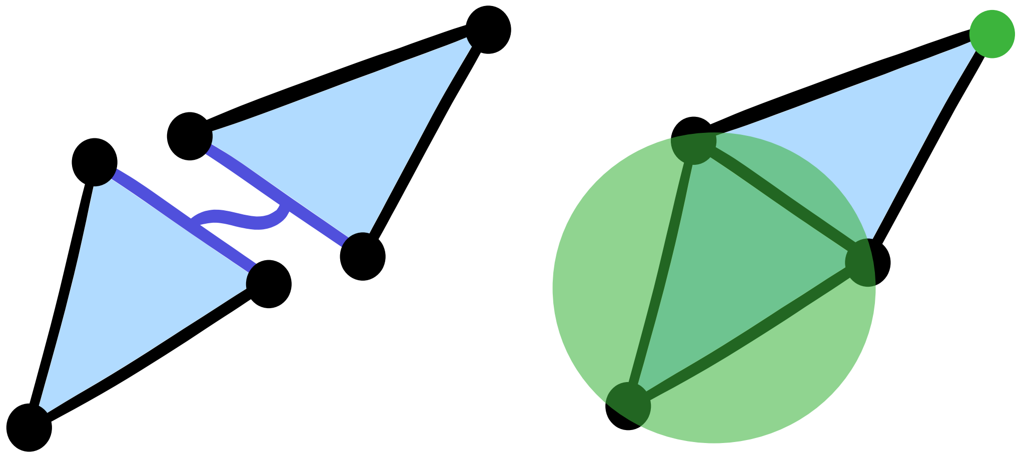

interior
boundary


interior
boundary
singularities

Problem

Delaunay triangulation

triangulation in which
every edge is Delaunay



The Delaunay triangulation

Generically, every surface has a unique Delaunay triangulation whose vertices are the singularities

The Delaunay triangulation

Generically, every surface has a unique
Delaunay triangulation
whose vertices are the singularities

Problem

Given triangulation T, compute
"the" Delaunay triangulation
of the surface of T

Motivations

Motivations

- isometry testing

Motivations

- isometry testing
- shortest paths

- shortest paths

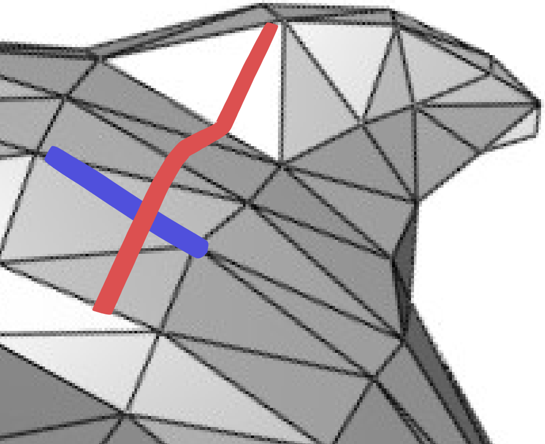
Shortest paths on meshes

On a mesh M with n triangles...

$\boxed{\text{a shortest path cannot cross an edge twice}}$

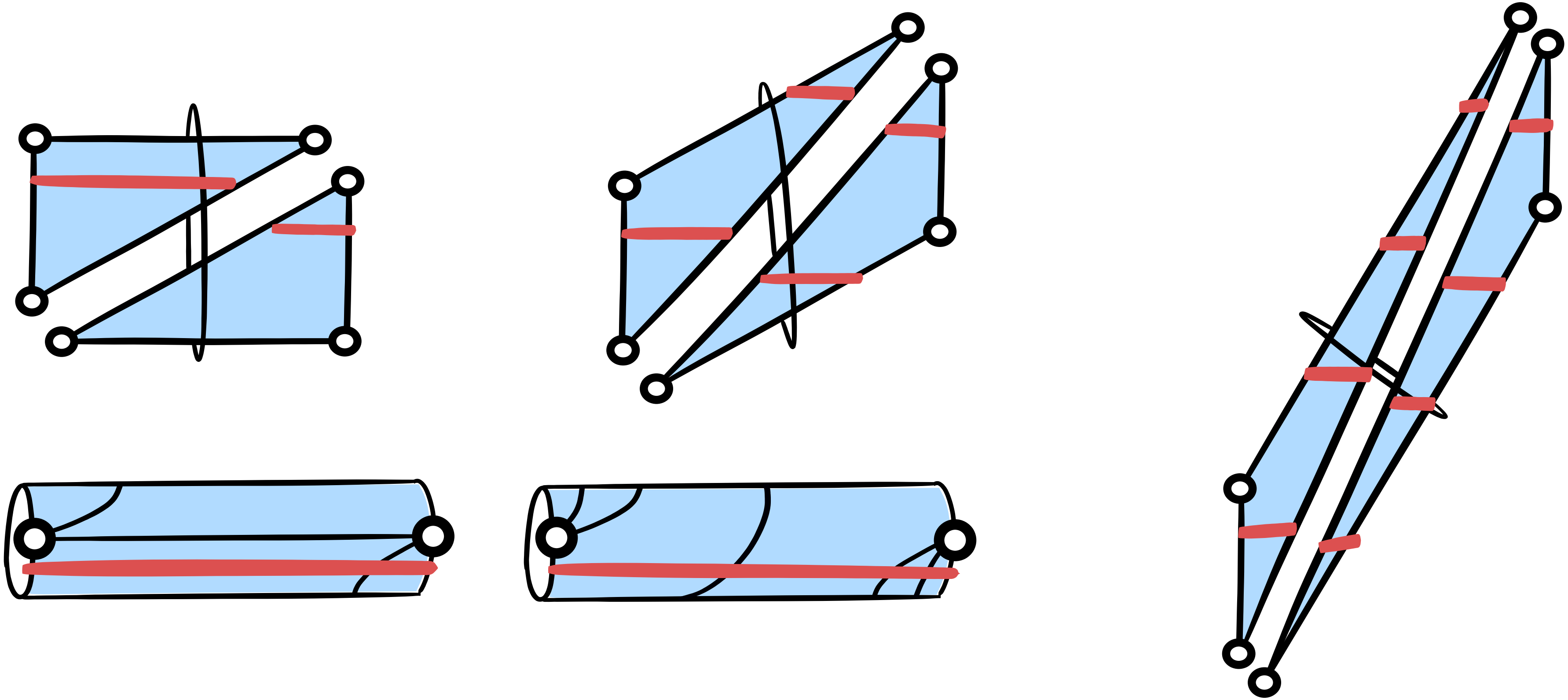
$\rightarrow$ shortest path can be computed in $O^*(n^2)$ time

Mitchel, Mount, Papadimitriou, 1987



Shortest paths on triangulations

they can cross edges arbitrarily many times

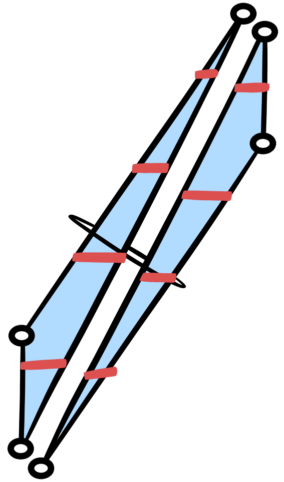


Erickson, 2006

Shortest paths on triangulations

Löffler, Ophelders, Staals, Silveira, 2023

happiness h: max number of times a shortest path visits a triangle



$\rightarrow$ shortest path can be computed in $O^*(n^2\, h)$ time

Shortest paths on triangulations

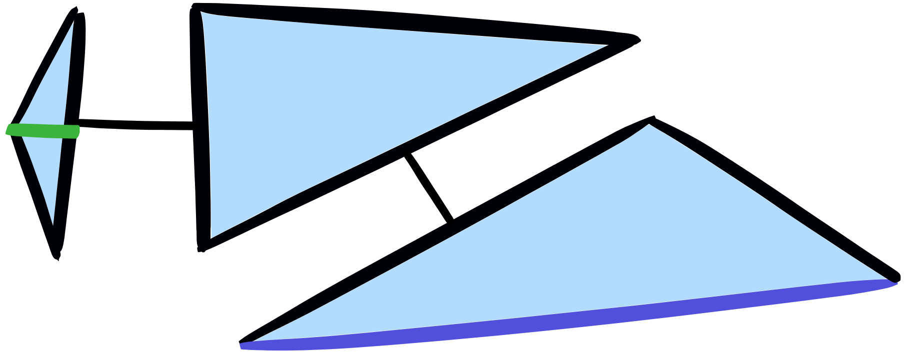
Löffler, Ophelders, Staals, Silveira, 2023

Delaunay triangulations have happiness $O(1)$

Result

Result

aspect ratio = $\dfrac{\text{maximum side length}}{\text{minimum height}}$

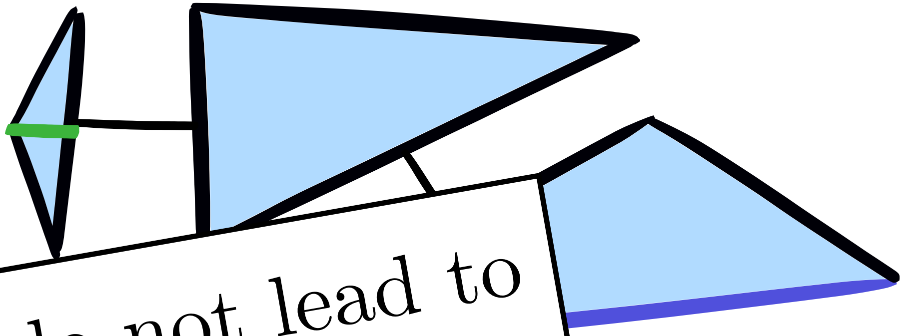


D. 2025

Given triangulation T of n triangles, of aspect ratio r, whose surface has no boundary, we can compute Delaunay in $O(n^3 \log^2(n) \cdot \log^4(r))$ time

Result

$$\text{aspect ratio} = \frac{\text{maximum side length}}{\text{minimum height}}$$



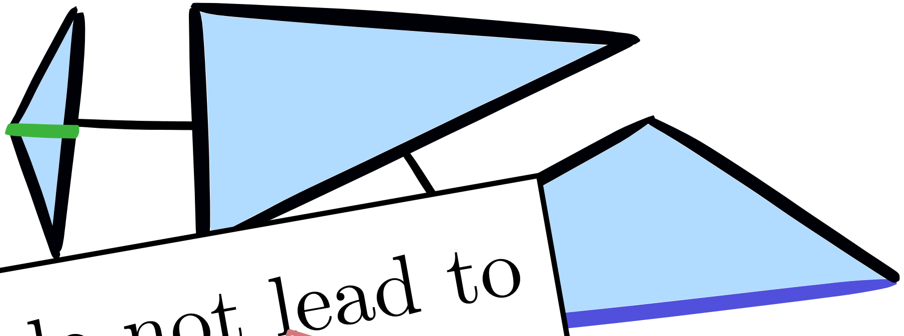
Previous techniques do not lead to better than $O(\mathrm{Poly}(n, r))$

D. 202

Given t … n triangles, of aspect ratio r, whose … surface has no boundary, we can compute Delaunay in $O(n^3 \log^2(n) \cdot \log^4(r))$ time

Result

aspect ratio $=$ $\dfrac{\text{maximum side length}}{\text{minimum height}}$



D. 202

Given t… of n triangles, of …
whose surface has no boundary, we can compute…
Delaunay in $O(n^3 \log^2(n) \cdot \log^4(r))$ time

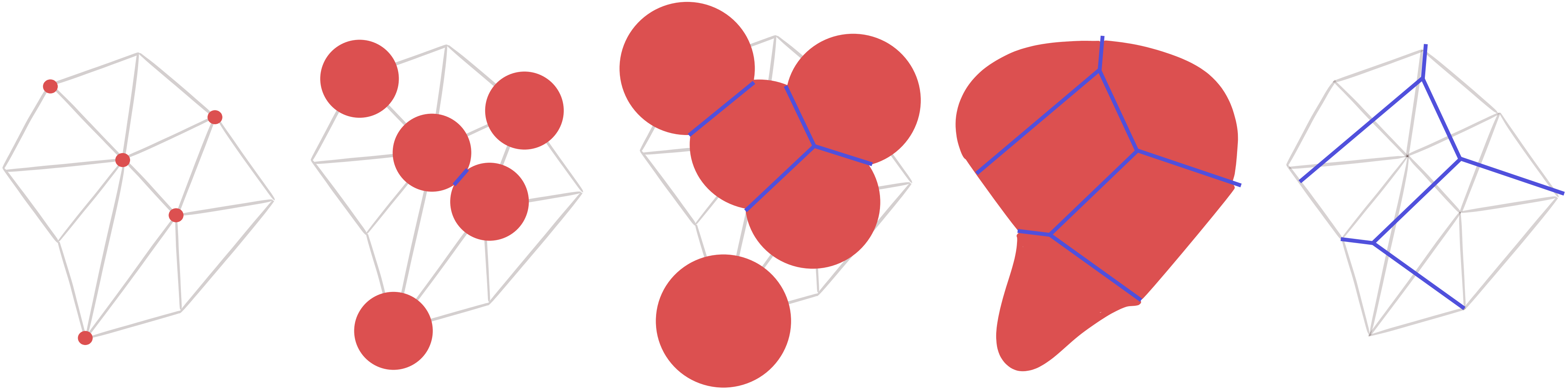
Previous techniques do not lead to better than $O(\mathrm{Poly}(n, r$…

Now backed by a lower bound!

Algorithm overview

Classical method

compute the Voronoi diagram
by propagating waves



then derive Delaunay from it

Algorithm

T → compute classically from Voronoi → Delaunay

Algorithm

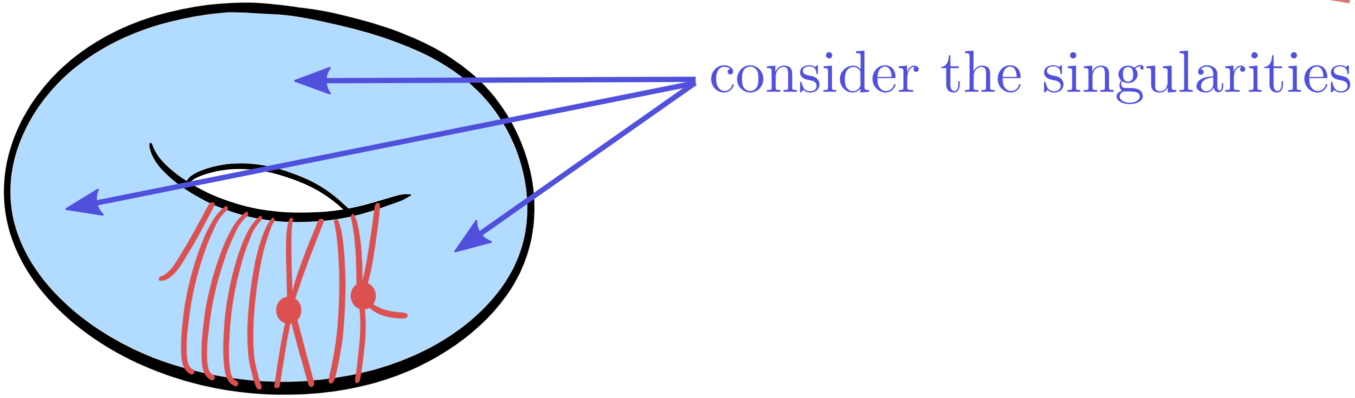
D. 2025

T → lower happiness → T' → compute classically from Voronoi → Delaunay

Algorithm

D. 2025

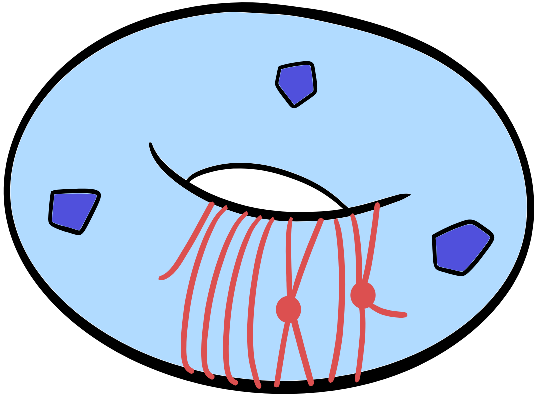
T → **lower happiness** → T' → **compute classically from Voronoi** → Delaunay



Algorithm

D. 2025

T → lower happiness → T' → compute classically from Voronoi → Delaunay

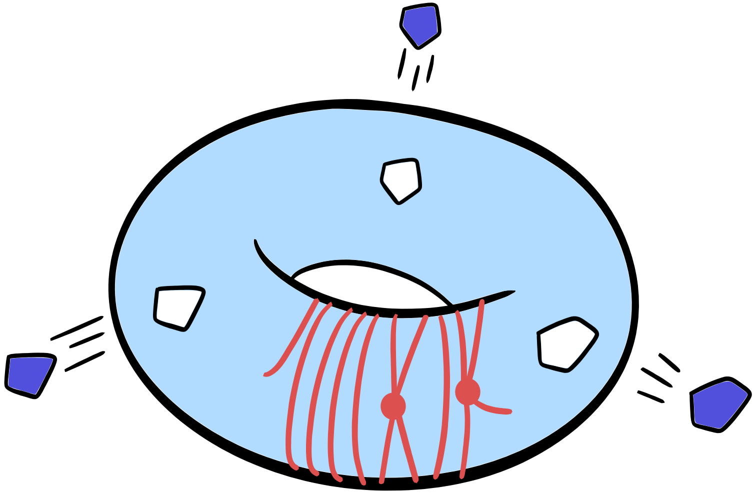


cut out caps around the singularities

Algorithm

D. 2025

T → lower happiness → T' → compute classically from Voronoi → Delaunay

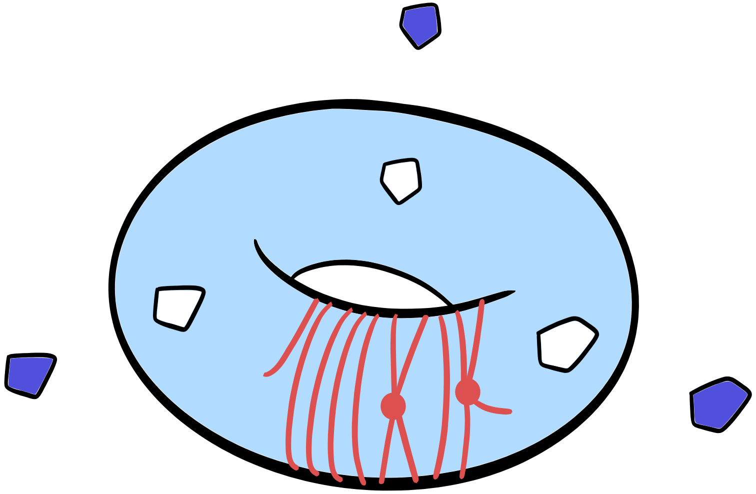


cut out caps around the singularities

Algorithm

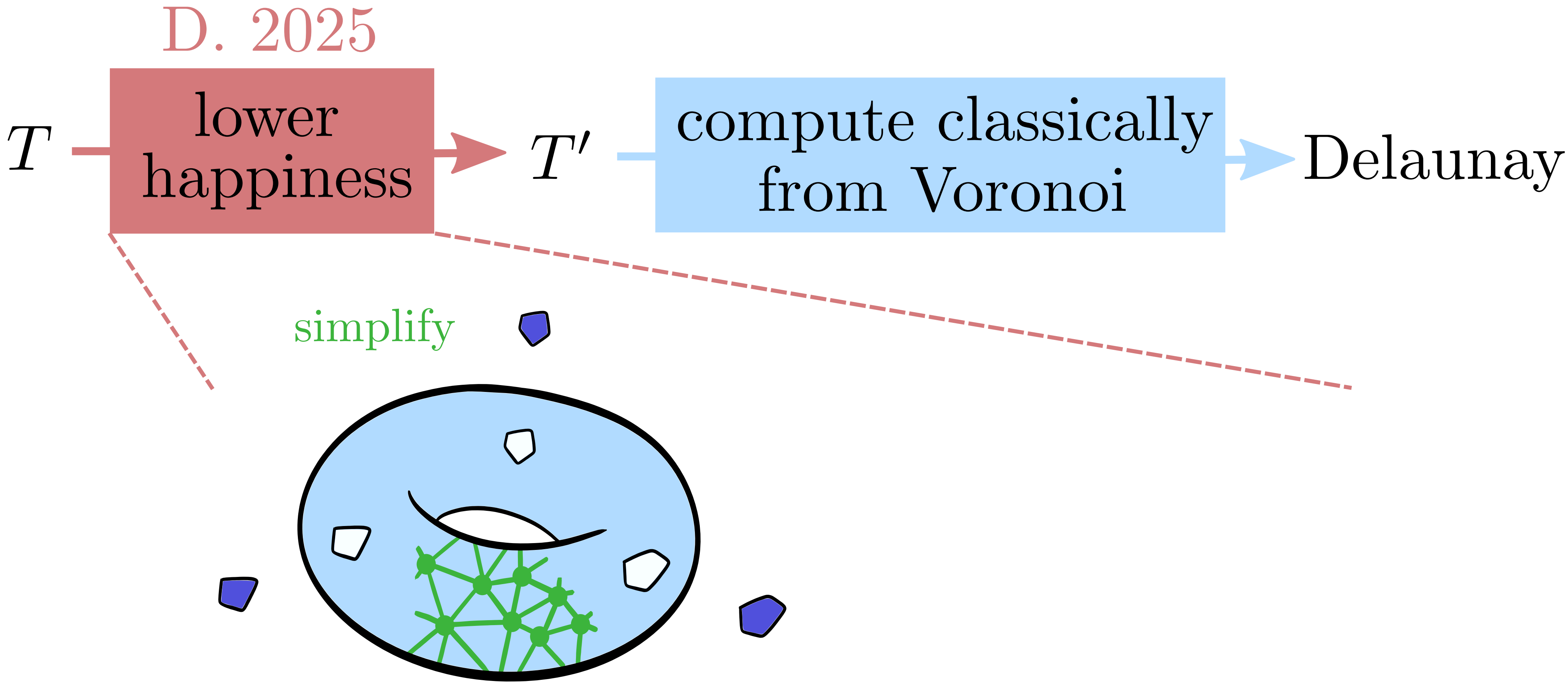
D. 2025

T → **lower happiness** → T' → **compute classically from Voronoi** → Delaunay



cut out caps around the singularities

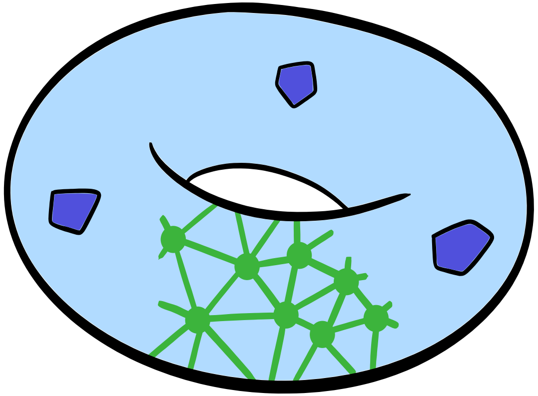
Algorithm


D. 2025
T
lower happiness
T'
compute classically from Voronoi
Delaunay
simplify

Algorithm

D. 2025

T → lower happiness → T' → compute classically from Voronoi → Delaunay



put the caps back

Simplification algorithm

Tuned combination of elementary operations, like

inserting vertices in edges

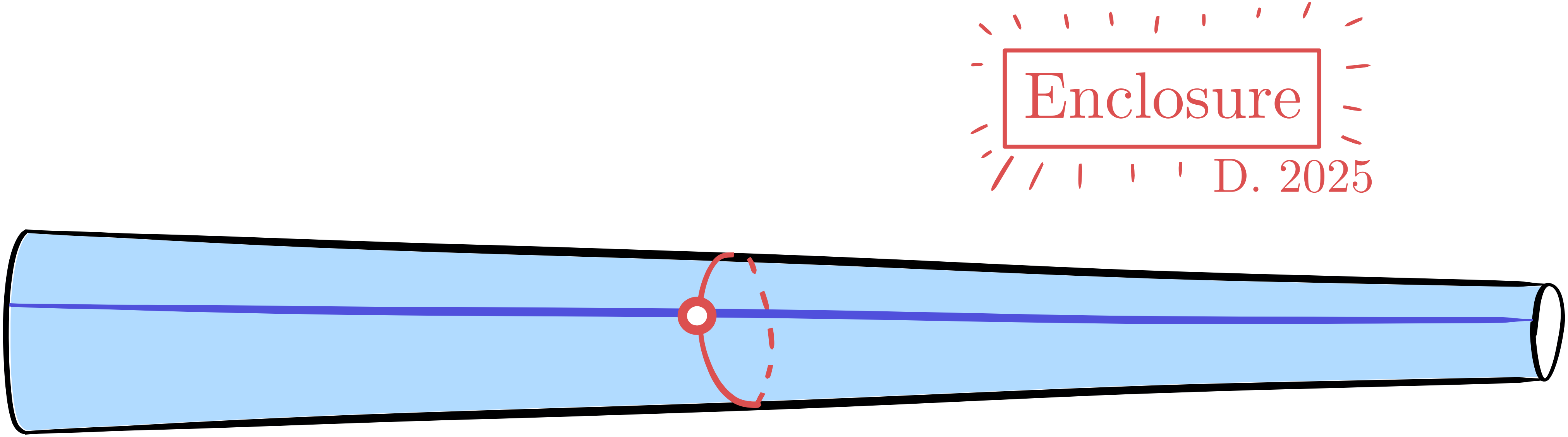
inserting edges in faces

deleting vertices

repeated many times

some simplify the geometry,

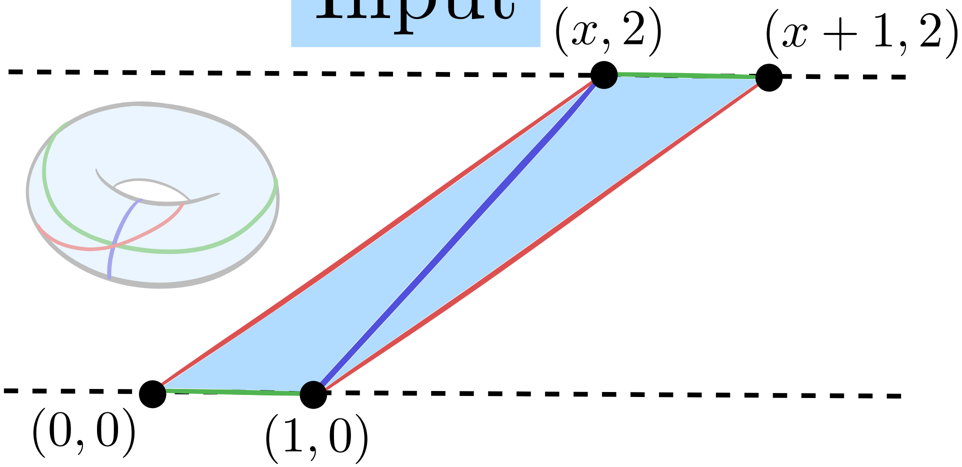
others decrease # vertices


Enclosure
D. 2025

Lower bound

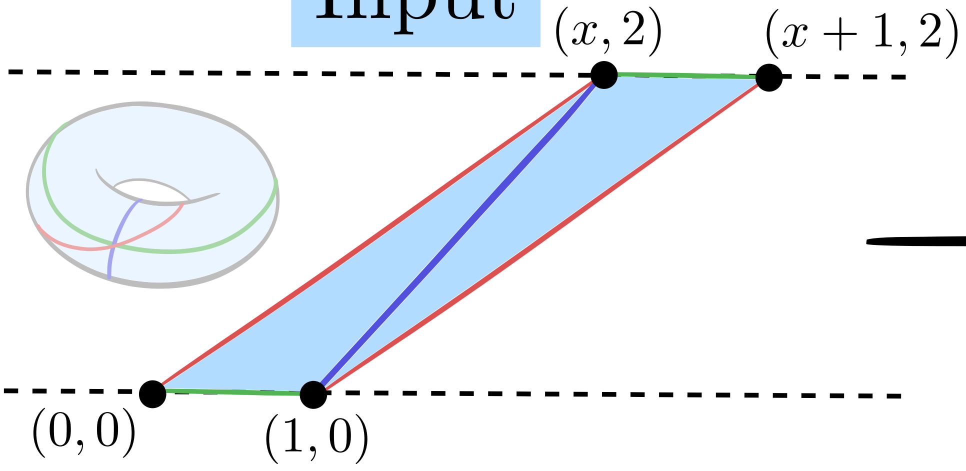
Lower bound

Input

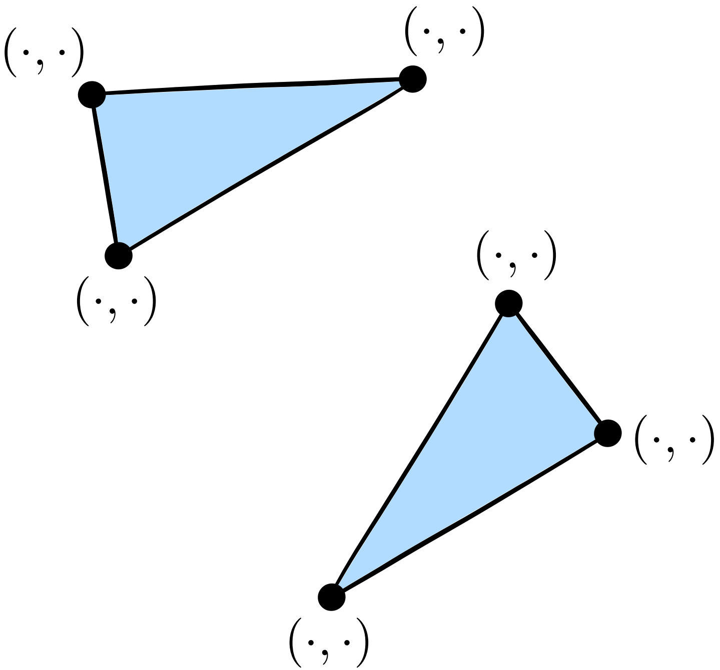


Lower bound

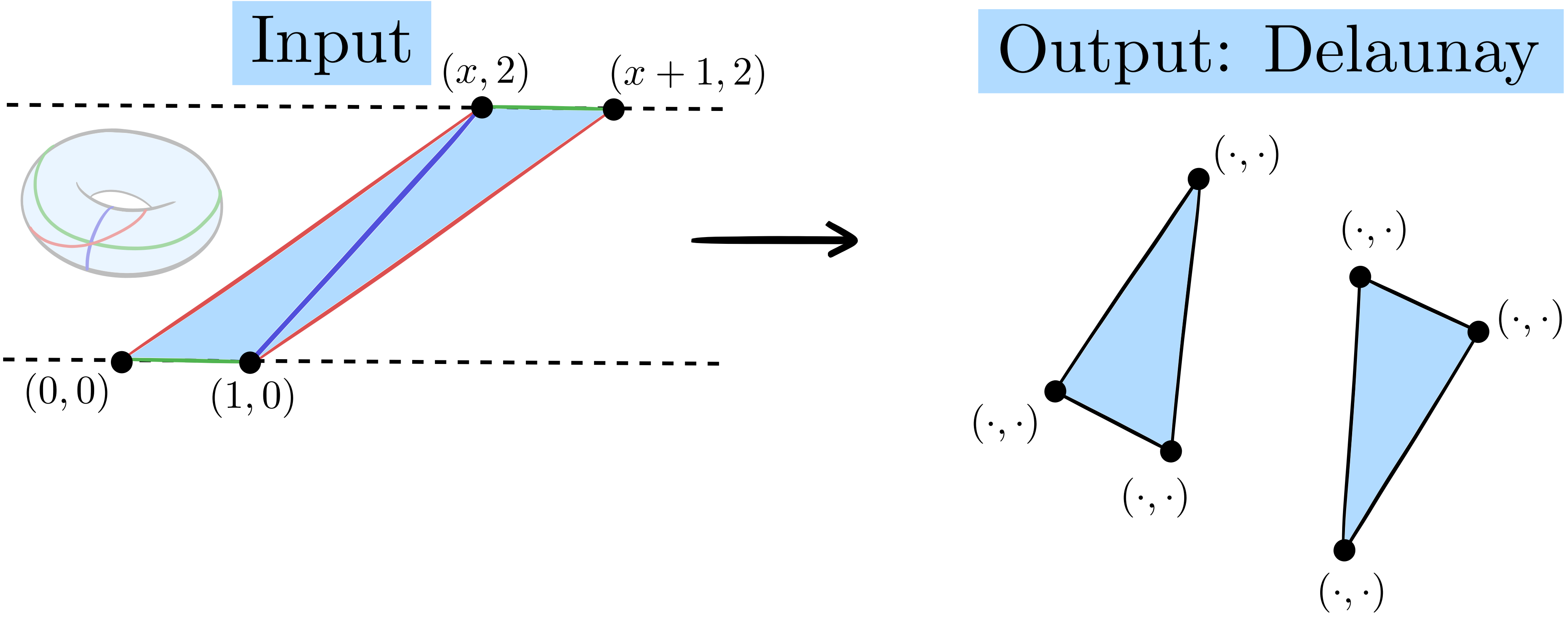
Input



Output: Delaunay

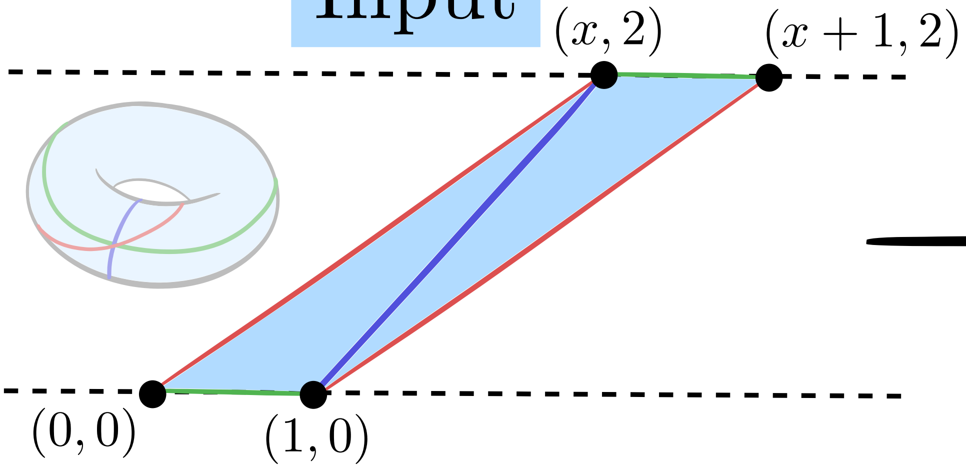


Lower bound

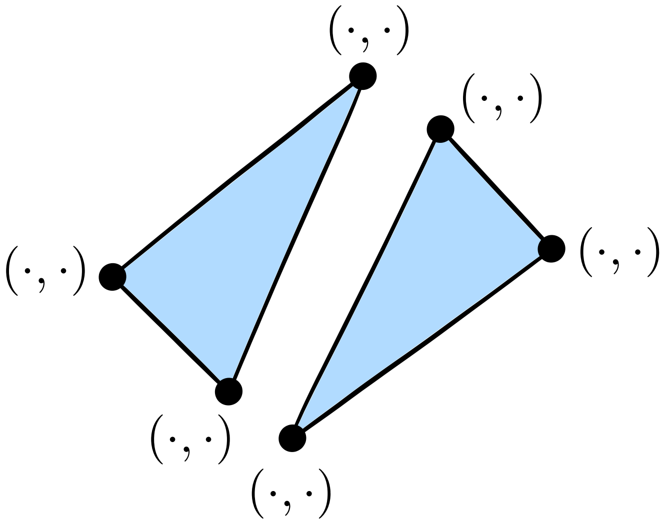

Input
$(x, 2)$
$(x+1, 2)$
$(0, 0)$
$(1, 0)$
Output: Delaunay
$(\cdot, \cdot)$
$(\cdot, \cdot)$
$(\cdot, \cdot)$
$(\cdot, \cdot)$
$(\cdot, \cdot)$
$(\cdot, \cdot)$

Lower bound

Input

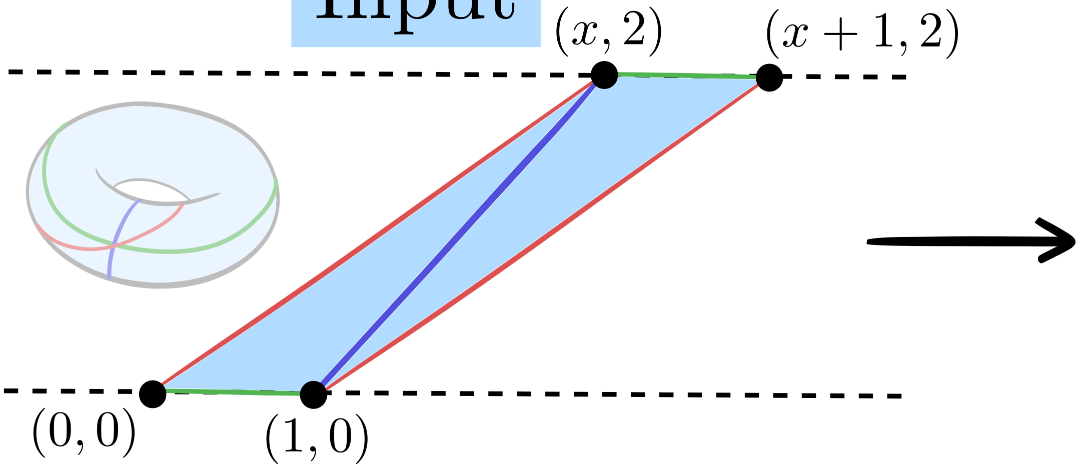


Output: Delaunay

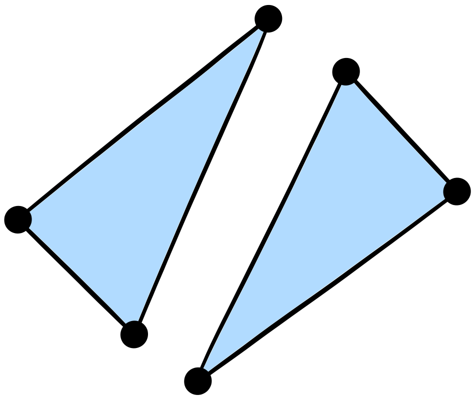


Lower bound

Input

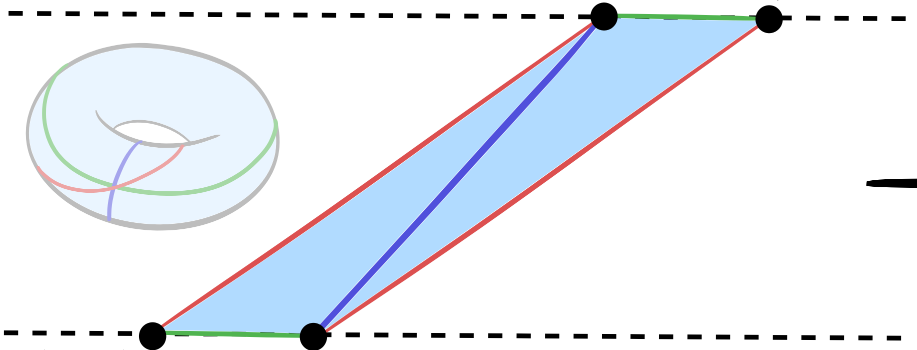


Output: Delaunay

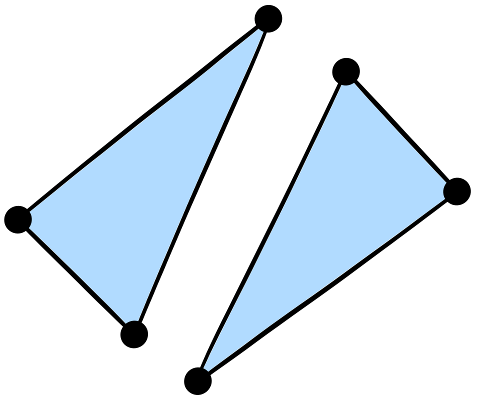


Lower bound

Input



Output: Delaunay

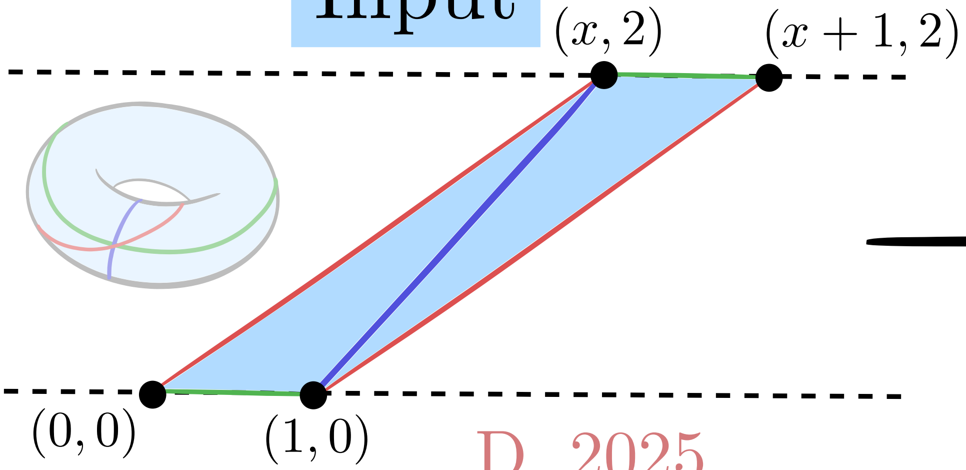


D. 2025

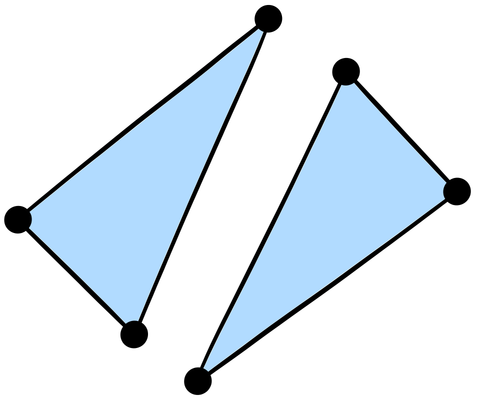
No Real RAM algo can compute Delaunay from x in $o(\log x)$ time

Lower bound

Input



Output: Delaunay



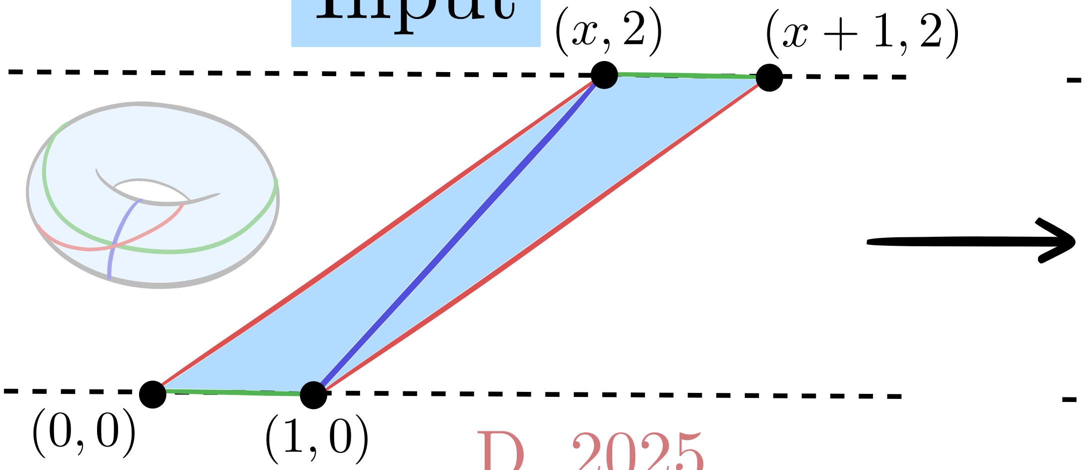
D. 2025

No Real RAM algo can compute Delaunay from x in $o(\log x)$ time

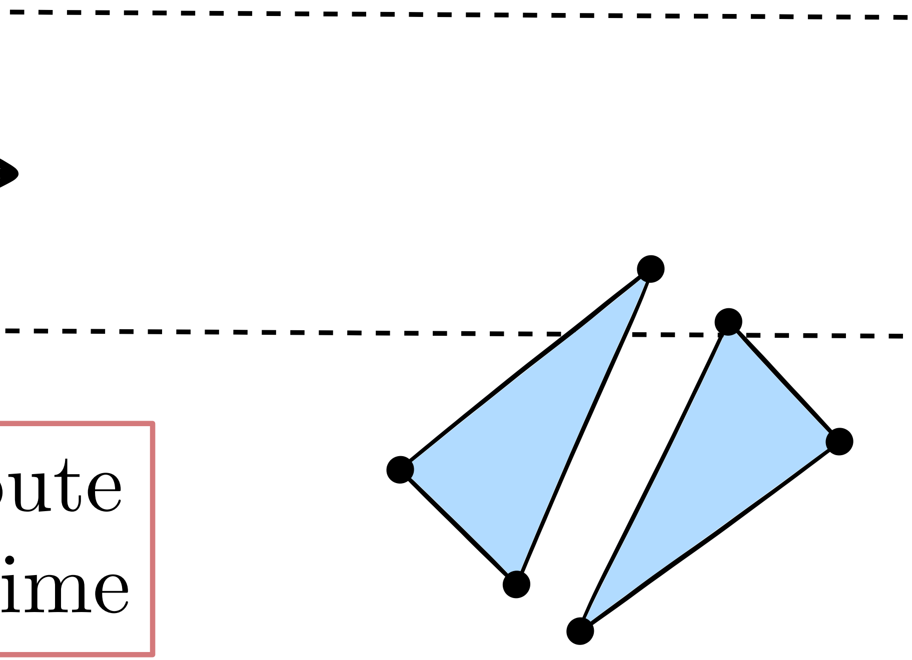
Otherwise we could compute $\lfloor x \rfloor$ from x in $o(\log x)$ time, which is impossible [Blum, Shub, and Smale, 1989]

Lower bound

Input



Output: Delaunay



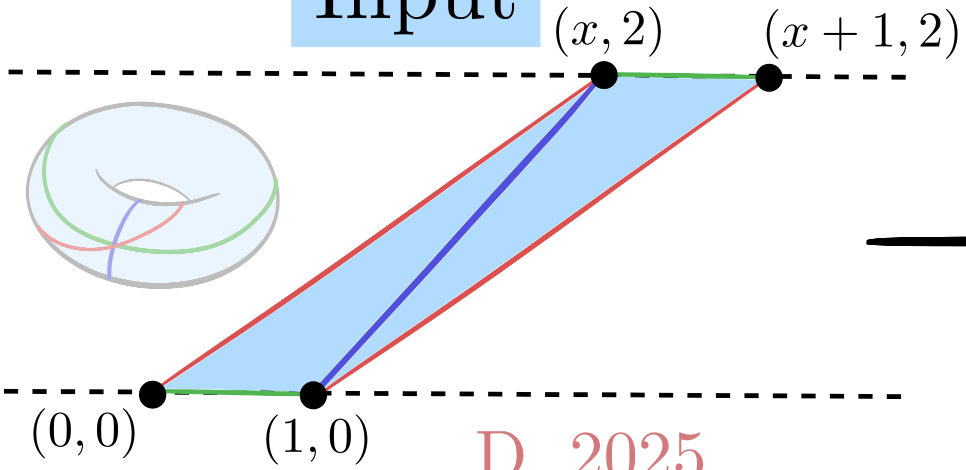
D. 2025

No Real RAM algo can compute Delaunay from x in $o(\log x)$ time

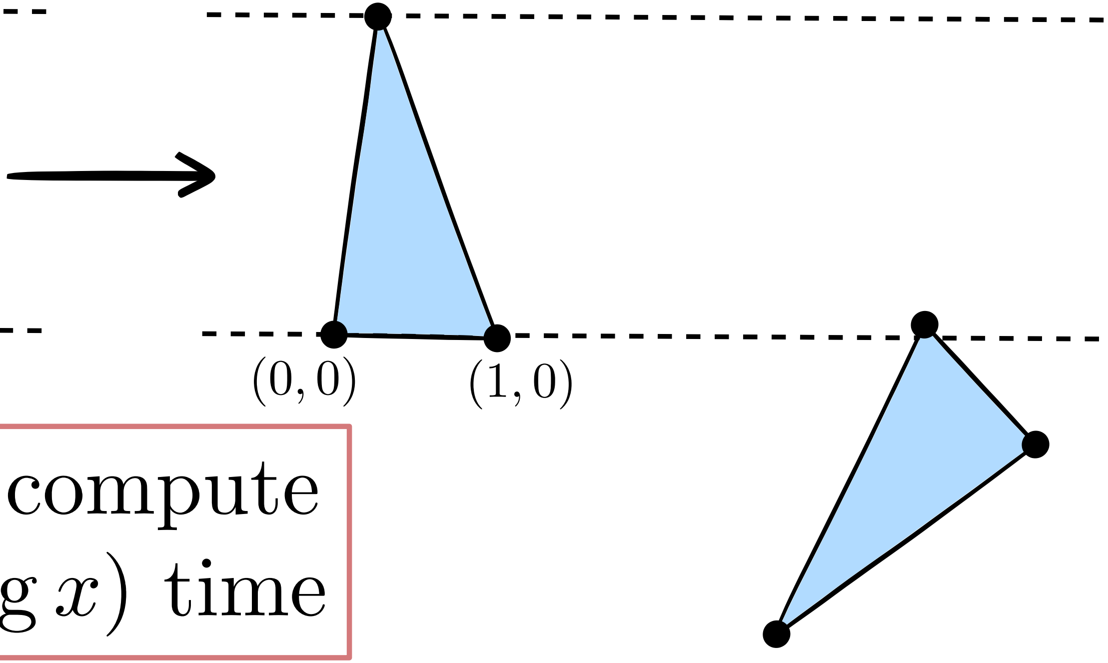
↪ Otherwise we could compute $\lfloor x \rfloor$ from x in $o(\log x)$ time, which is impossible [Blum, Shub, and Smale, 1989]

Lower bound

Input



Output: Delaunay

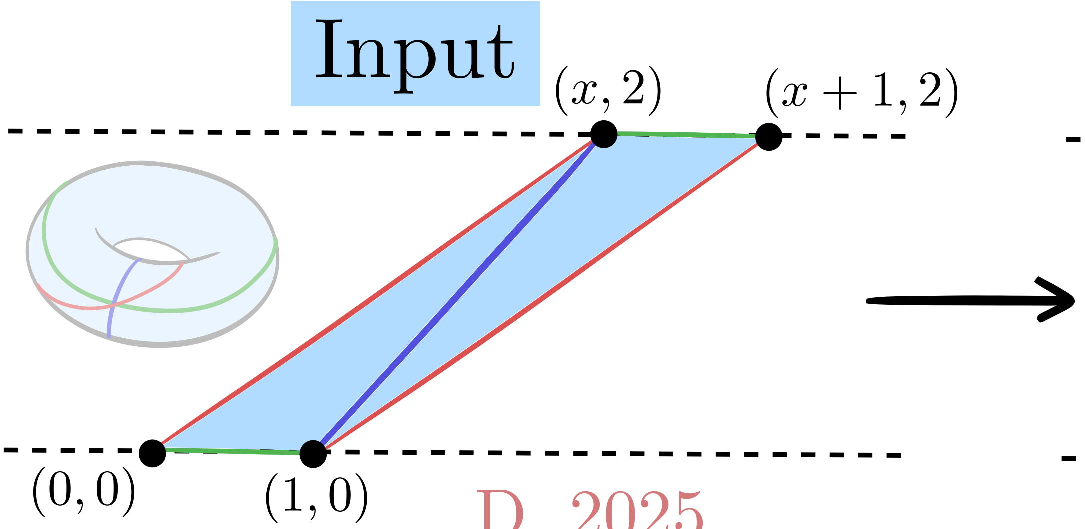


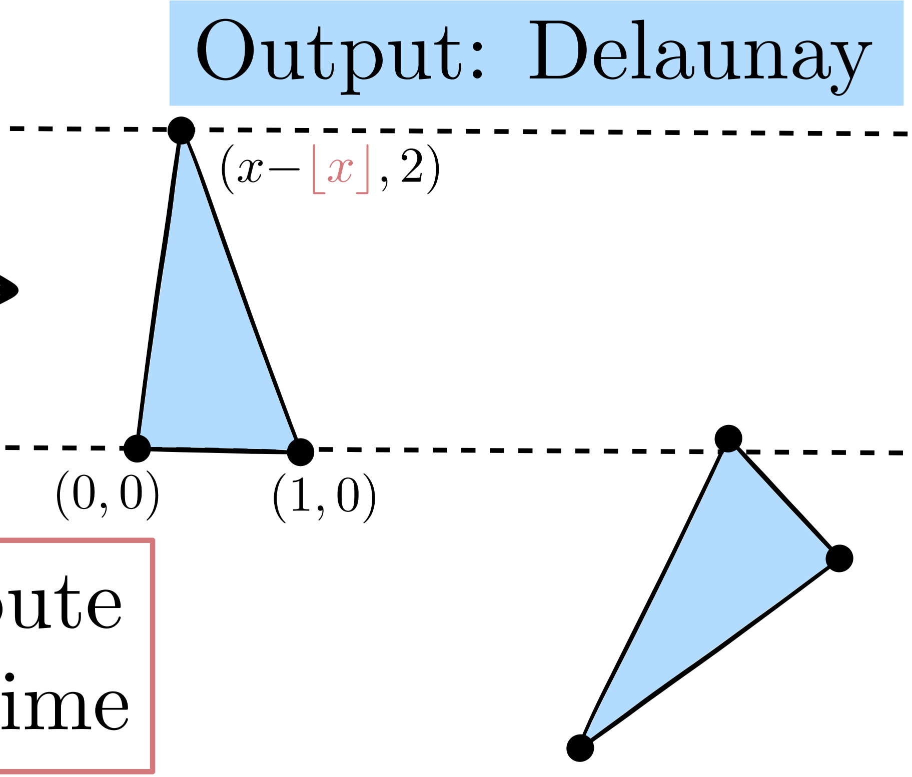
D. 2025

No Real RAM algo can compute Delaunay from x in $o(\log x)$ time

Otherwise we could compute $\lfloor x \rfloor$ from x in $o(\log x)$ time, which is impossible [Blum, Shub, and Smale, 1989]

Lower bound





D. 2025

No Real RAM algo can compute Delaunay from x in $o(\log x)$ time

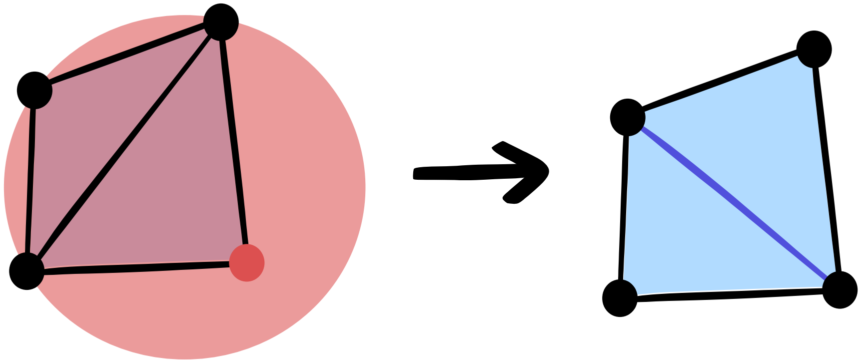
↪ Otherwise we could compute $\lfloor x \rfloor$ from x in $o(\log x)$ time, which is impossible [Blum, Shub, and Smale, 1989]

Other results

Other results

D., 2022-23

Upper bound for # Delaunay flips on flat tori,
tight up to constant factor



(open for higher genus surfaces)

Other results

Despré, D., Pouget, and Teillaud, 2025

CGAL package for computing with hyperbolic surfaces

generation (genus 2 only)

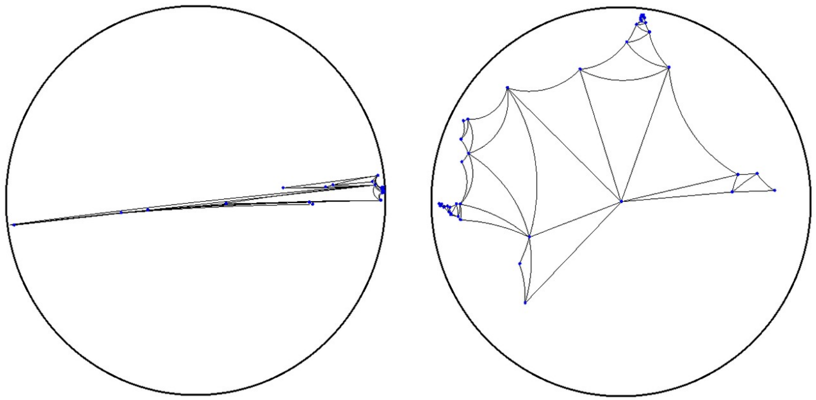
↓

Triangulated hyperbolic surface

↓

visualization

Delaunay flip



Ot results

Exact computations!

Despré, D., Pouget, and Te

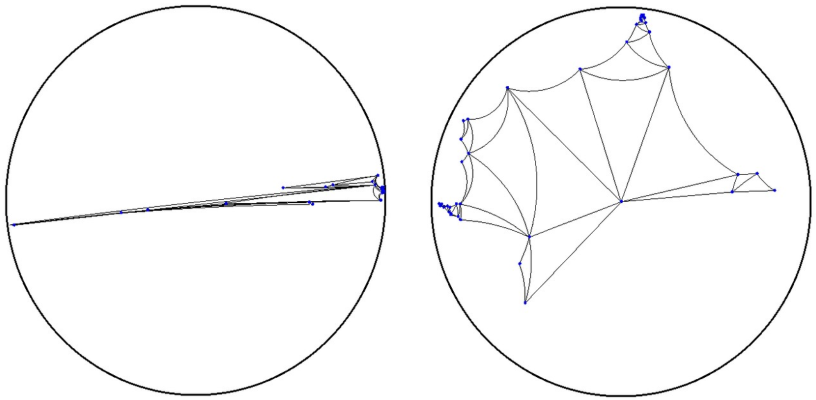
CGAL package for computing hyperbolic surfaces

generation (genus 2 only)

Triangulated hyperbolic surface

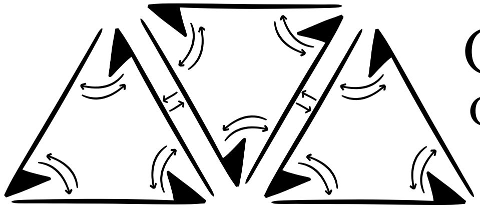
Delaunay flip

visualization



Triangulated hyperbolic surface

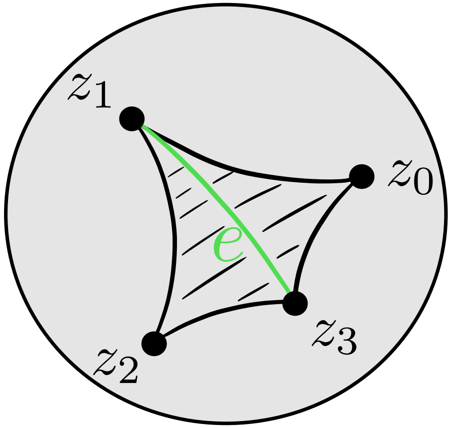
Combinatorics



CGAL
combinatorial maps

Geometry

- each edge is decorated
  with a complex number
  (cross ratio)



$$e \to \frac{(z_3 - z_1)(z_4 - z_2)}{(z_3 - z_2)(z_4 - z_1)}$$

Triangulated hyperbolic surface

Untangling Graphs

Computing Delaunay Triangulations

Possible continuations

Untangling Graphs

Computing Delaunay Triangulations

Possible continuations

Improving the untangling algos

Improving the untangling algos

- if the drawing can be untangled, return a sequence of untangling moves

Improving the untangling algos

- if the drawing can be untangled, return a sequence of untangling moves

- otherwise, return a smallest sub-drawing that can't be untangled

Improving the untangling algos

- if the drawing can be untangled, return a sequence of untangling moves
- otherwise, return a smallest sub-drawing that can't be untangled
- improve the complexity using systems of quads? or by better conversion of model?

Related problems

Related problems

- is computing the minimum number of crossings of a graph up to homotopy FPT or $W[1]$-hard?

Related problems

- is computing the minimum number of crossings of a graph up to homotopy FPT or $W[1]$-hard?

- how unique reducing triangulations are?

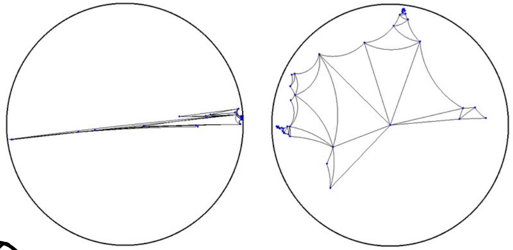
Related problems

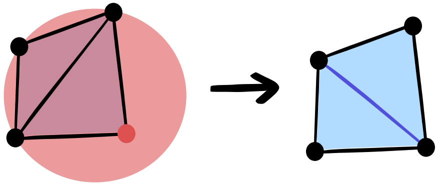
- is computing the minimum number of crossings of a graph up to homotopy FPT or $W[1]$-hard?

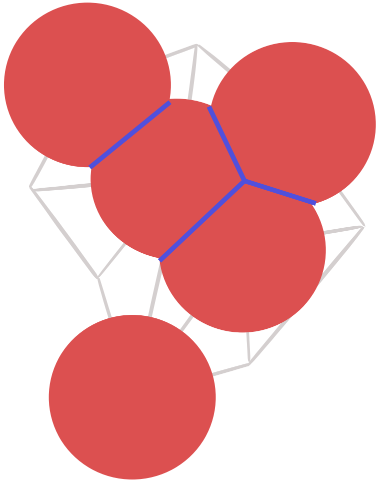
- how unique reducing triangulations are?

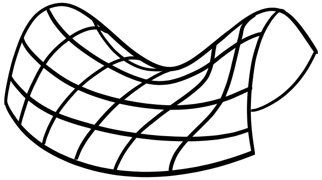
- extension to non orientable surfaces

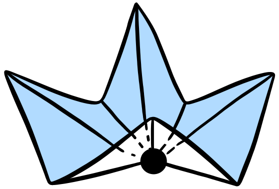
CGAL

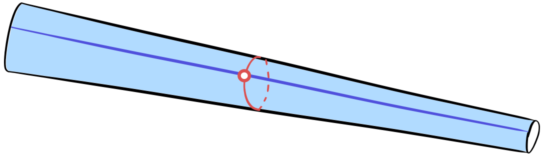


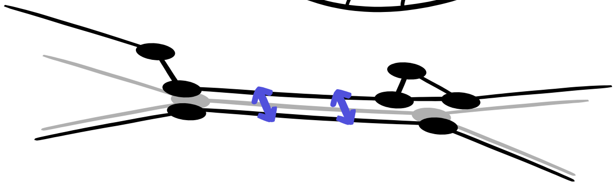


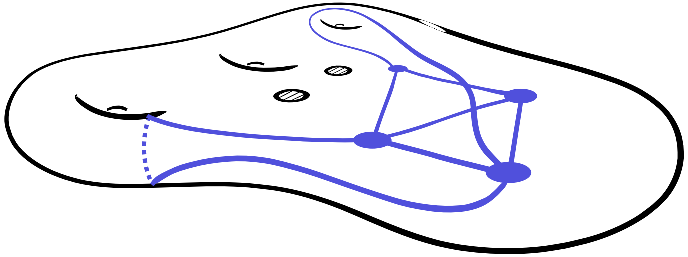


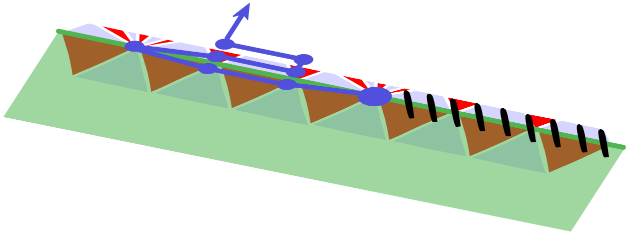


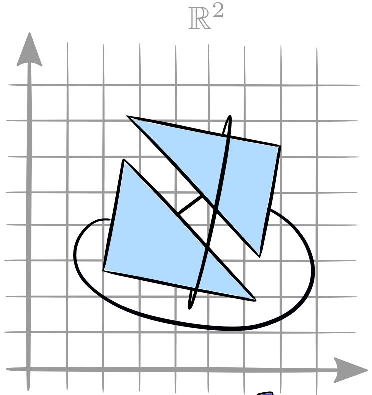












Thank you