Random discrete surfaces and graph exploration processes

Gilles Schaeffer

CNRS / Ecole Polytechnique, Palaiseau, France

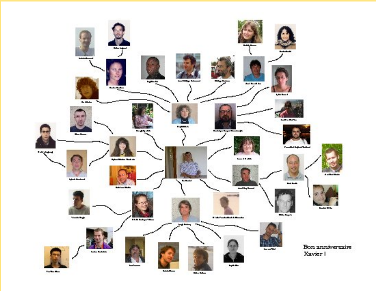
Combinatorics

Combinatorics

Combinatorial objects

Combinatorics

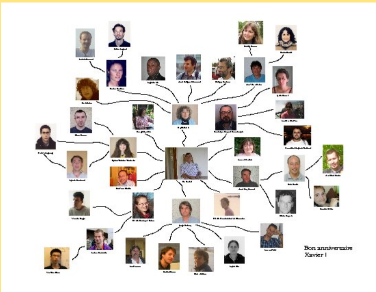
Combinatorial objects

tree like structures

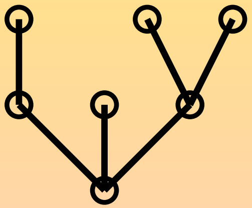
Combinatorics

Combinatorial objects

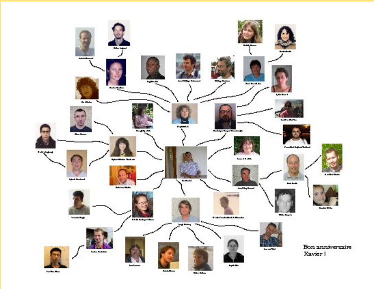

tree like structures

concept of graph



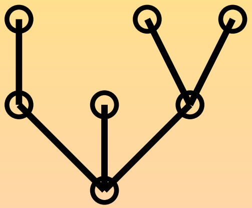
Combinatorics

Combinatorial objects

tree like structures

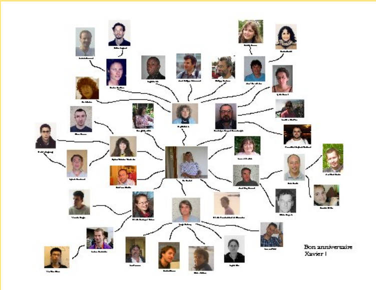
~~concept of graph~~



concept of tree

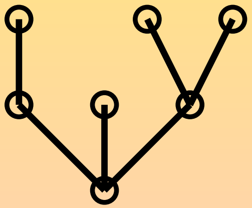
Combinatorics

Combinatorial objects


tree like structures

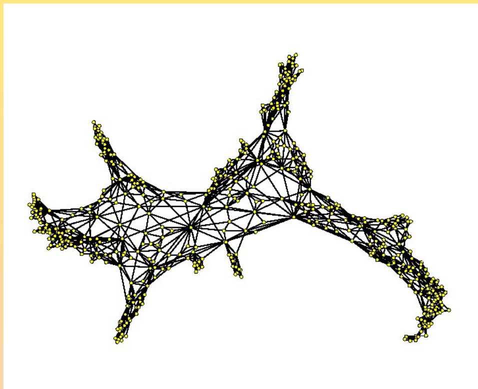
~~concept of graph~~

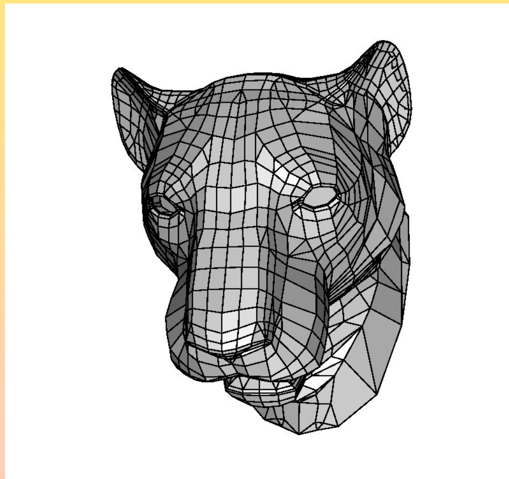


concept of tree

2d discrete structures

(discretized surfaces, meshes,...)



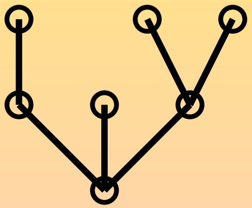
Combinatorics

Combinatorial objects

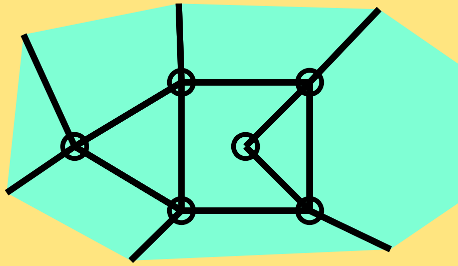


tree like structures

concept of graph



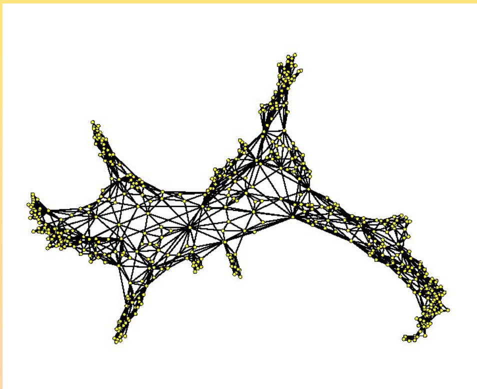
concept of tree

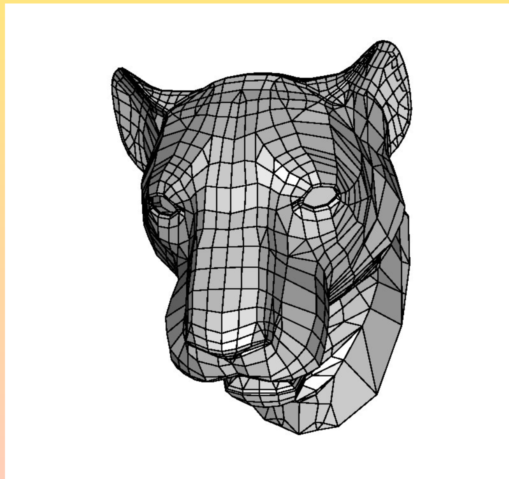


concept of graph

2d discrete structures

(discretized surfaces, meshes,...)

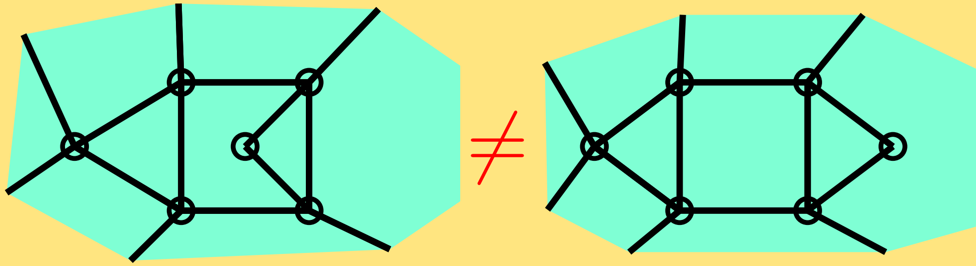


Combinatorics

Combinatorial objects

concept of *map*



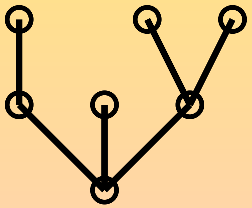
~~concept of graph~~

tree like structures

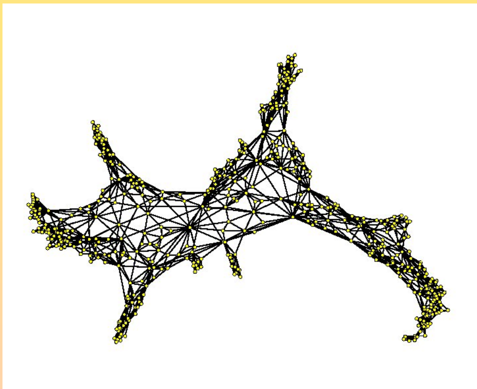
2d discrete structures

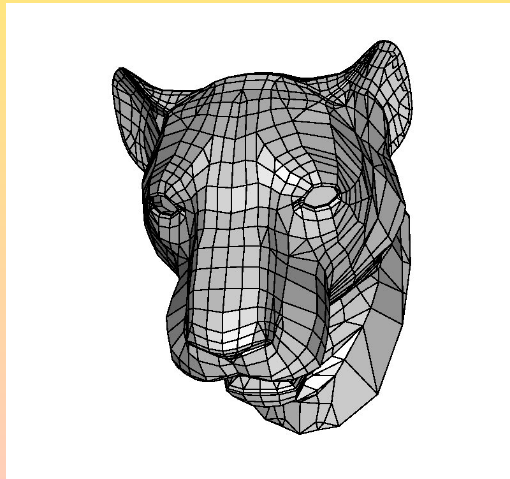
(discretized surfaces, meshes,...)

~~concept of graph~~



concept of tree



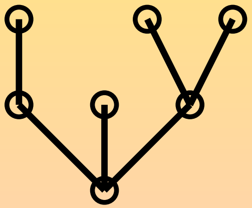


Combinatorics

Combinatorial objects = discrete abstractions of fundamental structures

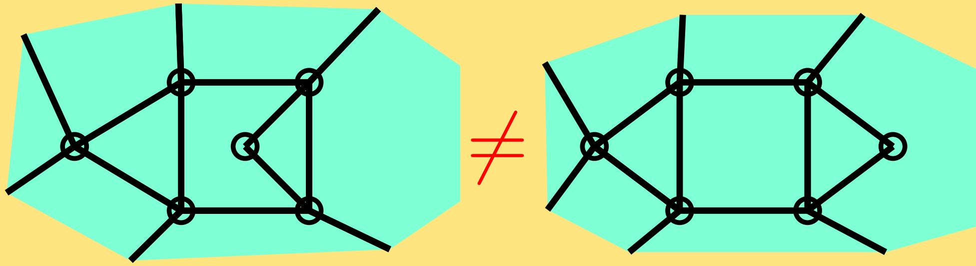
tree like structures

~~concept of graph~~



concept of tree

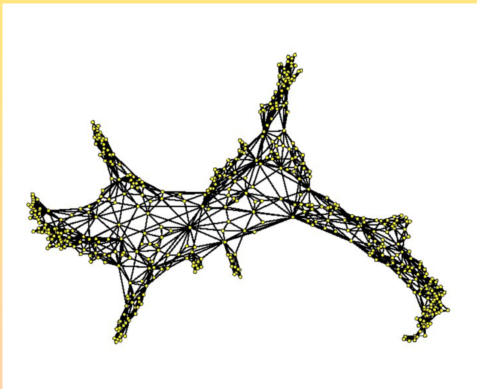
concept of *map*

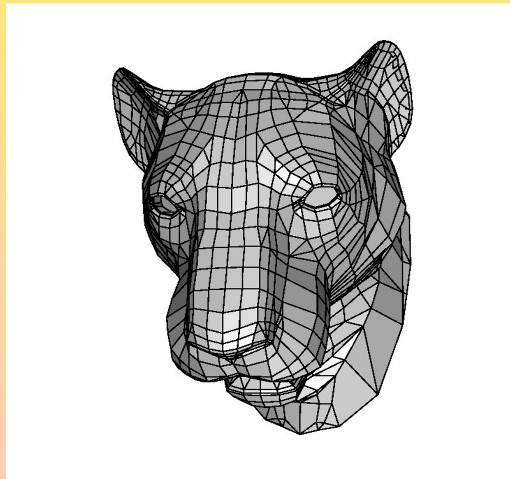


~~concept of graph~~

2d discrete structures

(discretized surfaces, meshes,...)





My idea of combinatorics

Elucidate the properties of those fundamental discrete structures that are common to various scientific fields (CS/math/physics/bio).

Algorithmic combinatorics

My idea of combinatorics

Elucidate the properties of those fundamental discrete structures that are common to various scientific fields (CS/math/physics/bio).

and, more specifically of "algorithmic combinatorics"

concentrate on constructive properties and on the algorithmic point of view on structures

Algorithmic combinatorics

My idea of combinatorics

Elucidate the properties of those fundamental discrete structures that are common to various scientific fields (CS/math/physics/bio).

and, more specifically of "algorithmic combinatorics"

concentrate on constructive properties and on the algorithmic point of view on structures

The example of trees...

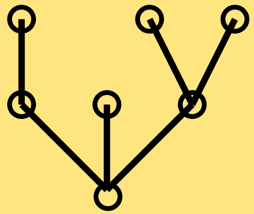
mathematical pt of view: connected graphs without cycle

algorithmic pt of view: recursive description (root; subtrees)
$\Rightarrow$ concept of breadth first or depth first search,
links with context free languages

(... Schützenberger's methodology...)

Exploration algorithms

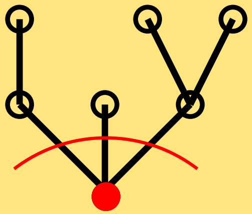
Tree exploration



breadth first

Exploration algorithms

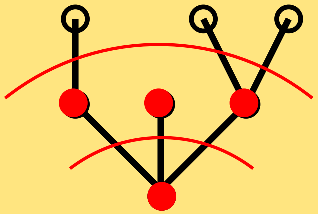
Tree exploration



breadth first

Exploration algorithms

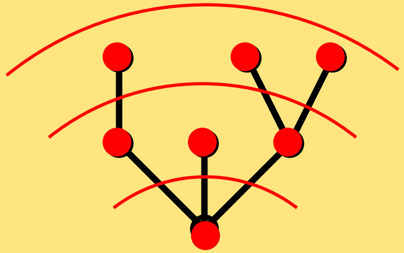
Tree exploration



breadth first

Exploration algorithms

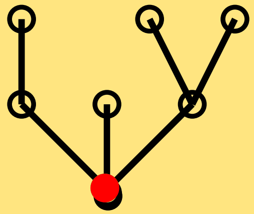
Tree exploration



breadth first

Exploration algorithms

Tree exploration

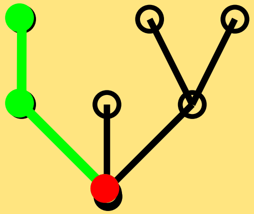


breadth first

depth first

Exploration algorithms

Tree exploration

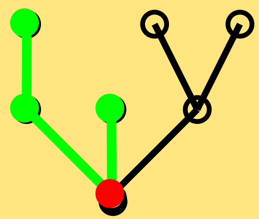


breadth first

depth first

Exploration algorithms

Tree exploration

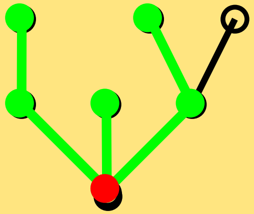


breadth first

depth first

Exploration algorithms

Tree exploration

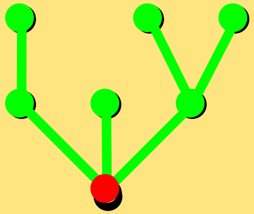


breadth first

depth first

Exploration algorithms

Tree exploration

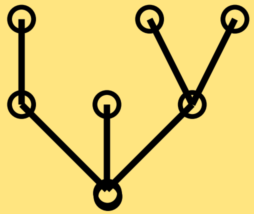


breadth first

depth first

Exploration algorithms

Tree exploration



breadth first

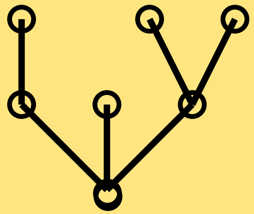
depth first

fundamental tools

for instance to encode trees

Exploration algorithms

Tree exploration



breadth first

depth first

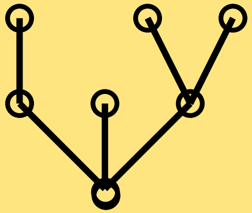
fundamental tools

for instance to encode trees

$\Rightarrow$ the prefix code of a tree

Exploration algorithms

Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

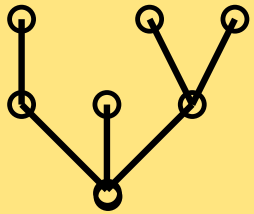
$\Rightarrow$ the prefix code of a tree

3 1 0 2 0 0 0 (breadth first)

3 1 0 0 2 0 0 (depth first)

Exploration algorithms

Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

$\Rightarrow$ the prefix code of a tree

3 1 0 2 0 0 0   (breadth first)

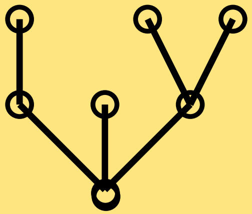
3 1 0 0 2 0 0   (depth first)

Statement. The set of code words is easy to describe.

More precisely: the language of prefix codes of ordered trees is *context-free*.

Exploration algorithms

Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

⇒ the prefix code of a tree

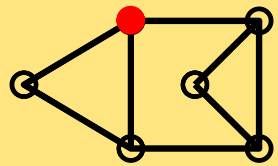
3 1 0 2 0 0 0 (breadth first)

3 1 0 0 2 0 0 (depth first)

Statement. The set of code words is easy to describe.

More precisely: the language of prefix codes of ordered trees is *context-free*.

Graph exploration

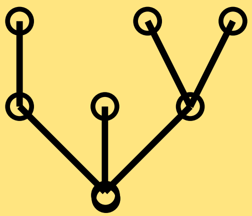


breadth first

depth first

Exploration algorithms

Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

⇒ the prefix code of a tree

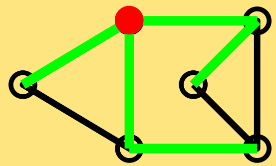
3 1 0 2 0 0 0 (breadth first)

3 1 0 0 2 0 0 (depth first)

Statement. The set of code words is easy to describe.

More precisely: the language of prefix codes of ordered trees is *context-free*.

Graph exploration



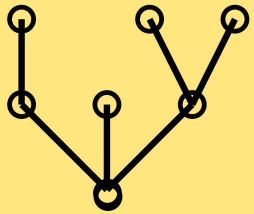
breadth first

depth first

construct a tree along the exploration

Exploration algorithms

Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

⇒ the prefix code of a tree

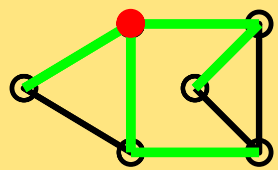
3 1 0 2 0 0 0 (breadth first)

3 1 0 0 2 0 0 (depth first)

Statement. The set of code words is easy to describe.

More precisely: the language of prefix codes of ordered trees is *context-free*.

Graph exploration



breadth first

depth first

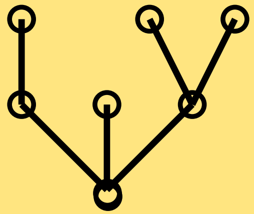
construct a tree along the exploration

+ extra info for external edges

⇒ encode graphs by tree-like structures

Exploration algorithms

Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

⇒ the prefix code of a tree

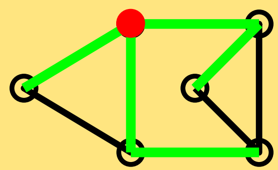
3 1 0 2 0 0 0 (breadth first)

3 1 0 0 2 0 0 (depth first)

Statement. The set of code words is easy to describe.

More precisely: the language of prefix codes of ordered trees is *context-free*.

Graph exploration



breadth first

depth first

construct a tree along the exploration

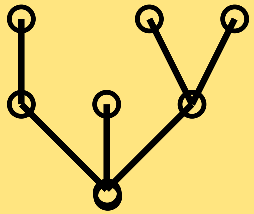
$+$ extra info for external edges

⇒ encode graphs by tree-like structures

but the set of "coding" trees is not easy to describe (for classic families of graphs like planar, 3-connected,...)

Exploration algorithms

Tree exploration



breadth first

depth first

fundamental tools

for instance to encode trees

⇒ the prefix code of a tree

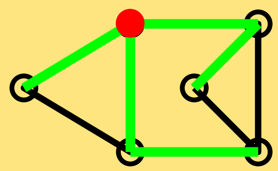
3 1 0 2 0 0 0 (breadth first)

3 1 0 0 2 0 0 (depth first)

Statement. The set of code words is easy to describe.

More precisely: the language of prefix codes of ordered trees is *context-free*.

Graph exploration



breadth first

depth first

construct a tree along the exploration

+ extra info for external edges

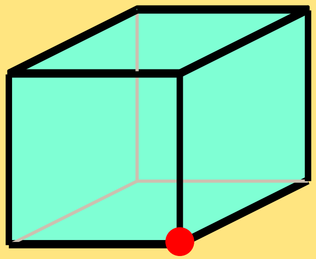
⇒ encode graphs by tree-like structures

but the set of "coding" trees is not easy to describe (for classic families of graphs like planar, 3-connected,...)

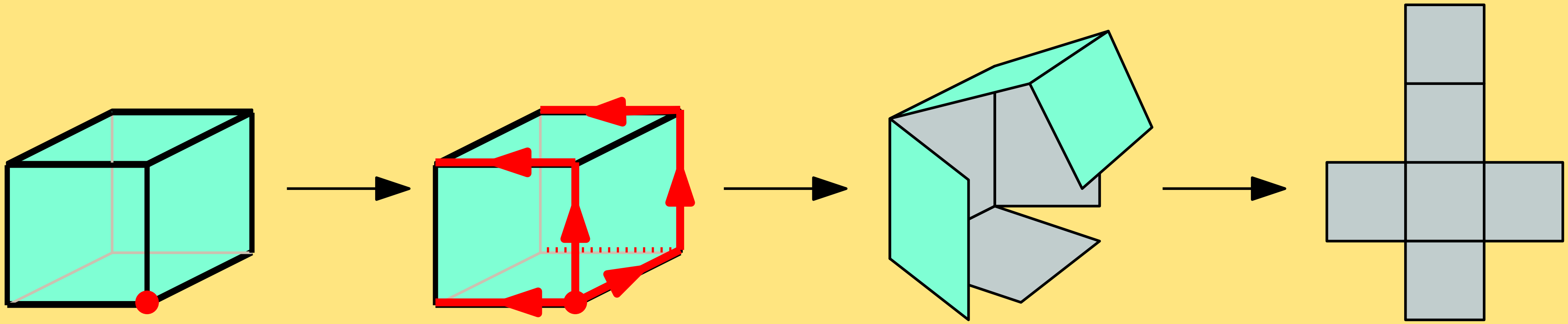
No good analog of the previous "statement".

Exploration of a map and surface surgery

Exploration of a map and surface surgery

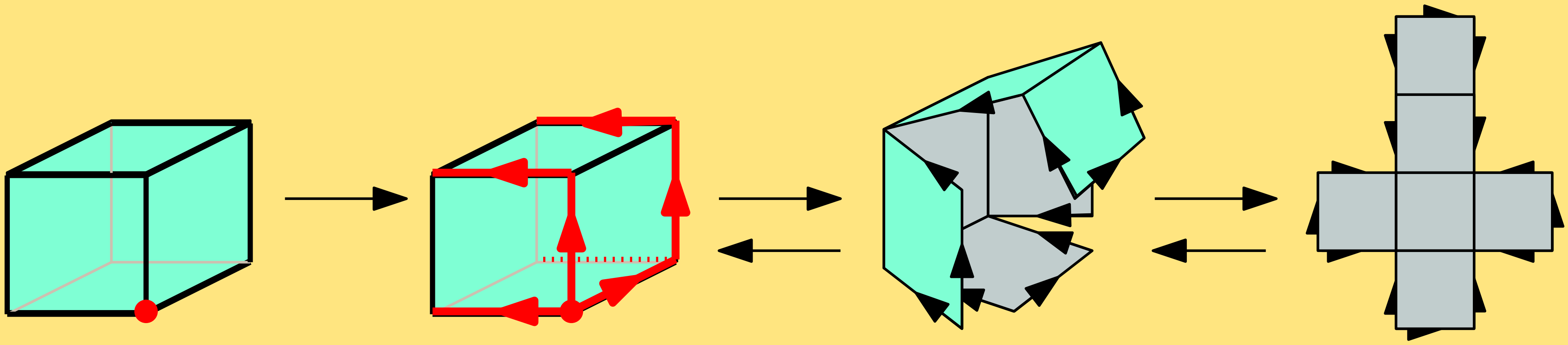


Exploration of a map and surface surgery



Exploration + cut ⇒ a "net" of the map

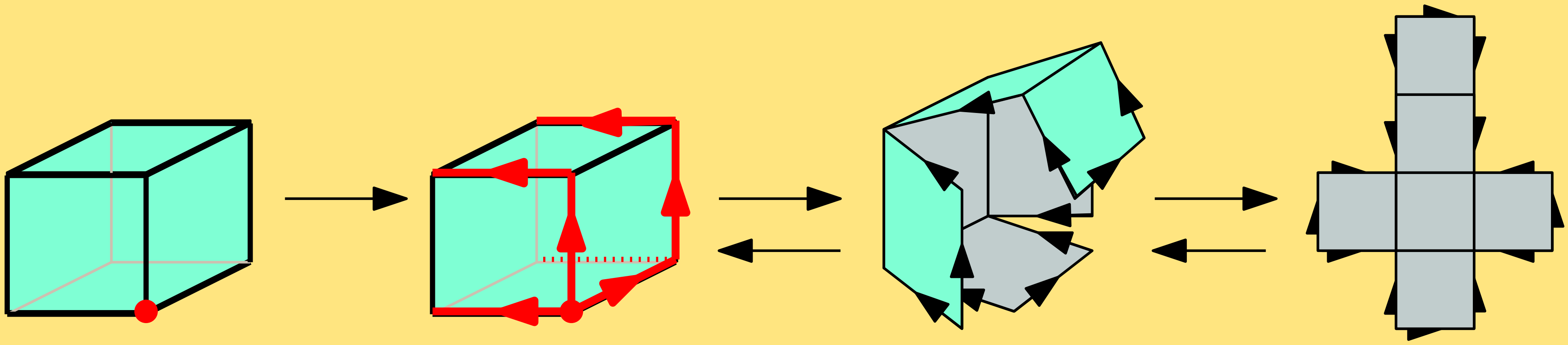
Exploration of a map and surface surgery



Exploration + cut ⇒ a "net" of the map

in order to reconstruct the surface, the orientation of cuts is enough: merge adjacent converging sides + iterate

Exploration of a map and surface surgery

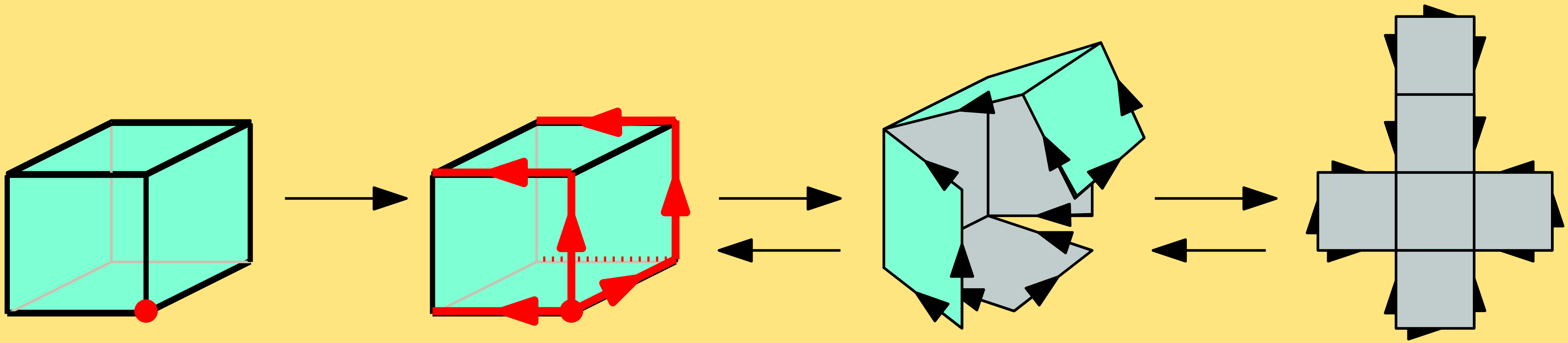


Exploration + cut ⇒ a "net" of the map

in order to reconstruct the surface, the orientation of cuts is enough: merge adjacent converging sides + iterate

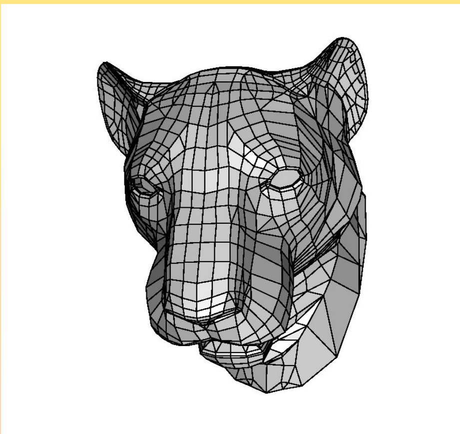
Exploration algorithms

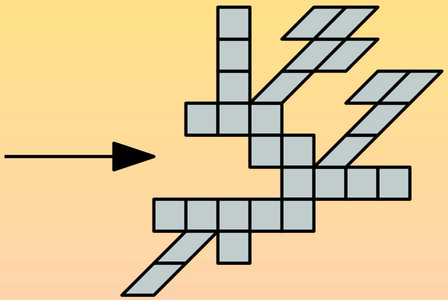
Exploration of a map and surface surgery



Exploration + cut ⇒ a "net" of the map

in order to reconstruct the surface, the orientation of cuts is enough: merge adjacent converging sides + iterate



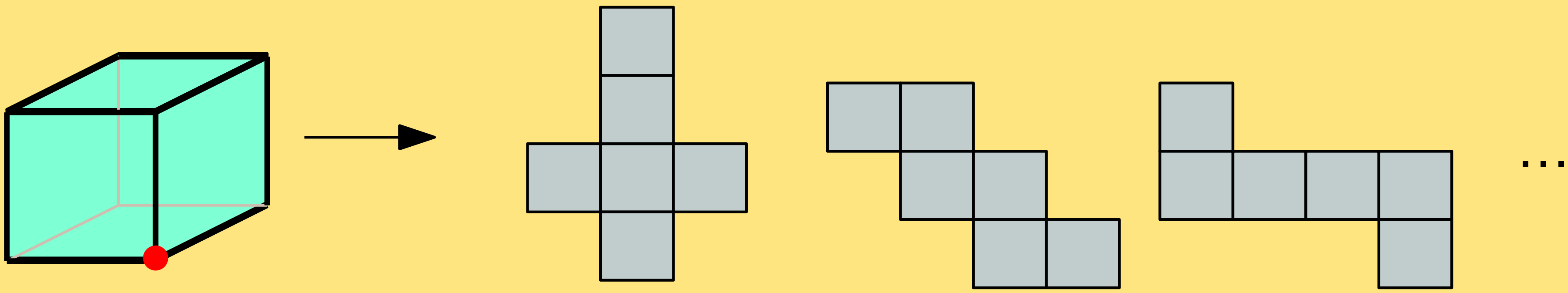


Nets are always trees of polygons

(as long as the surface has no handle)

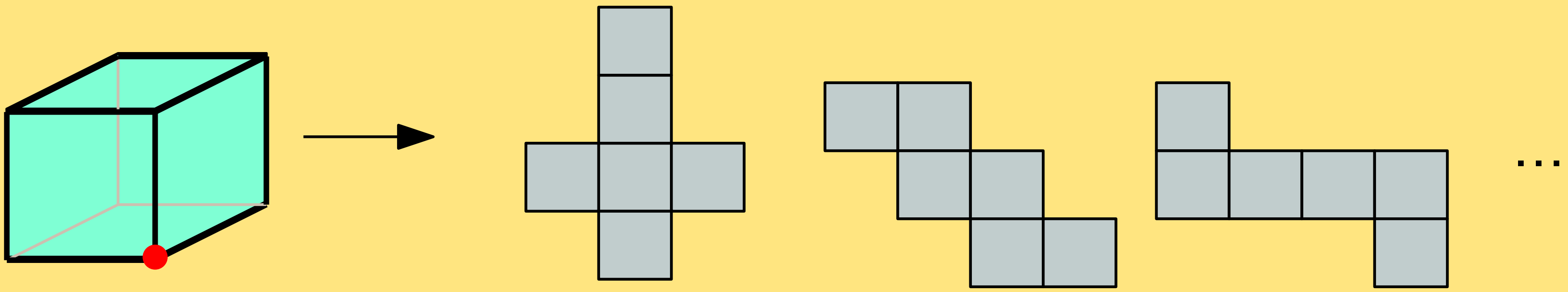
Exploration algorithms

To a map are associated many different nets

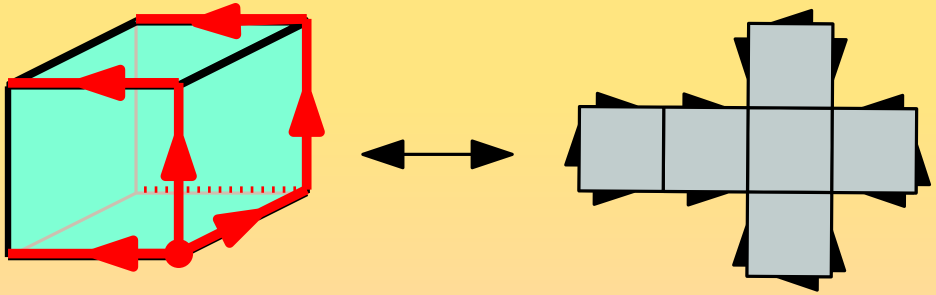


Exploration algorithms

To a map are associated
many different nets

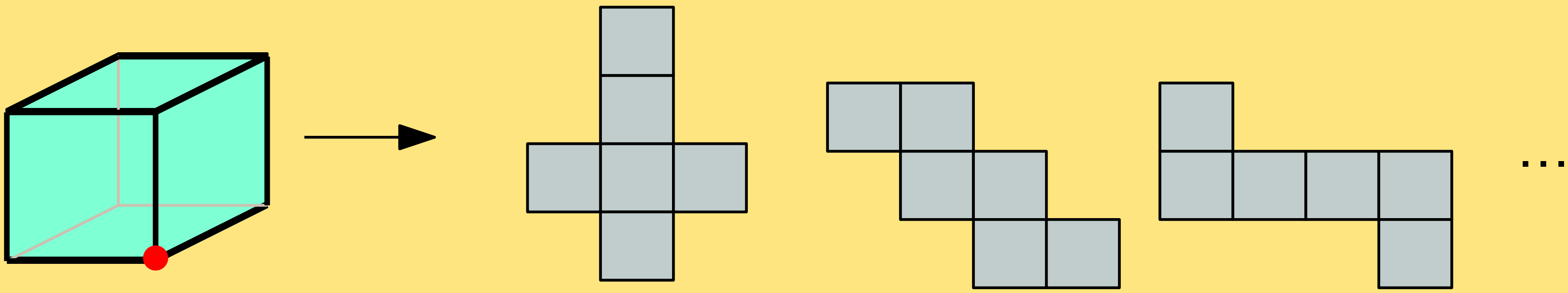


but a given exploration algorithm associates a canonical net to each map

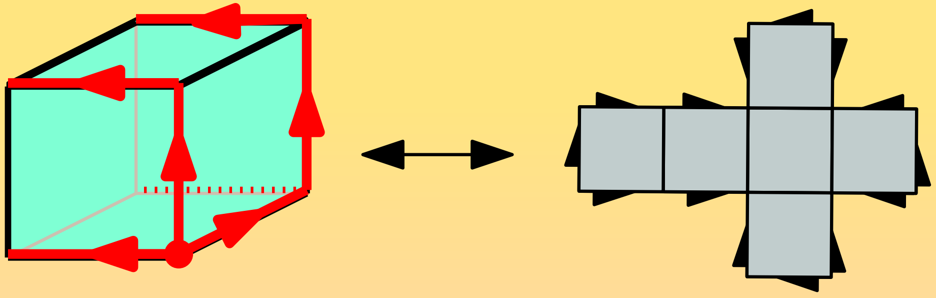


Exploration algorithms

To a map are associated many different nets



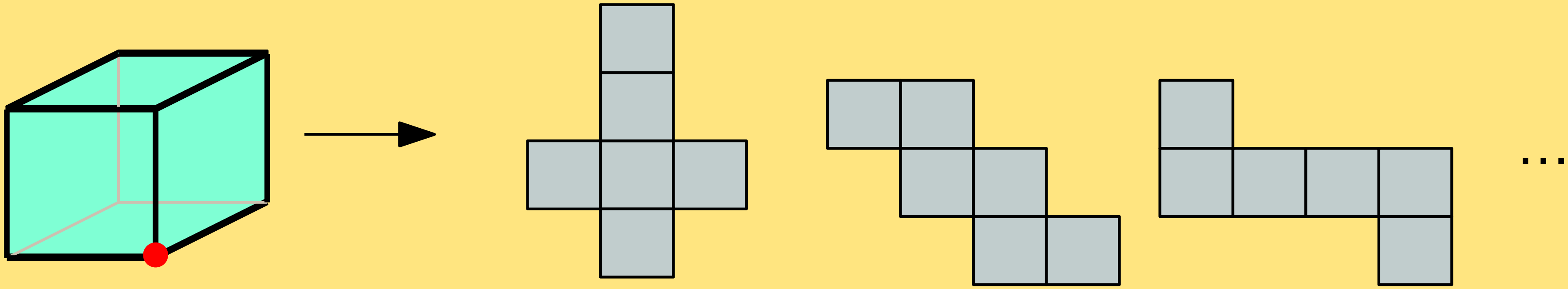
but a given exploration algorithm associates a canonical net to each map



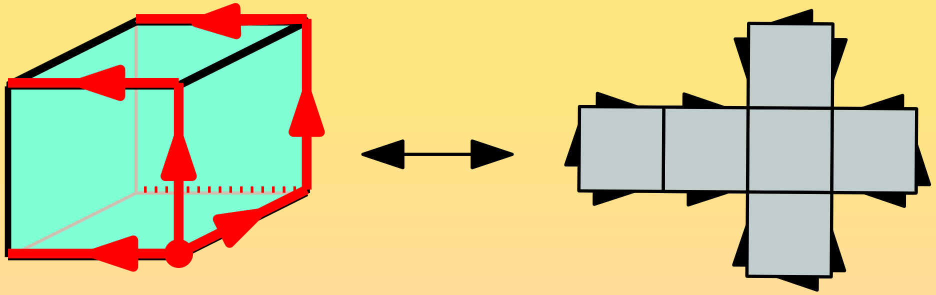
Represent again a map by a tree like structure!

Exploration algorithms

To a map are associated
many different nets



but a given exploration algorithm associates a canonical net to each map

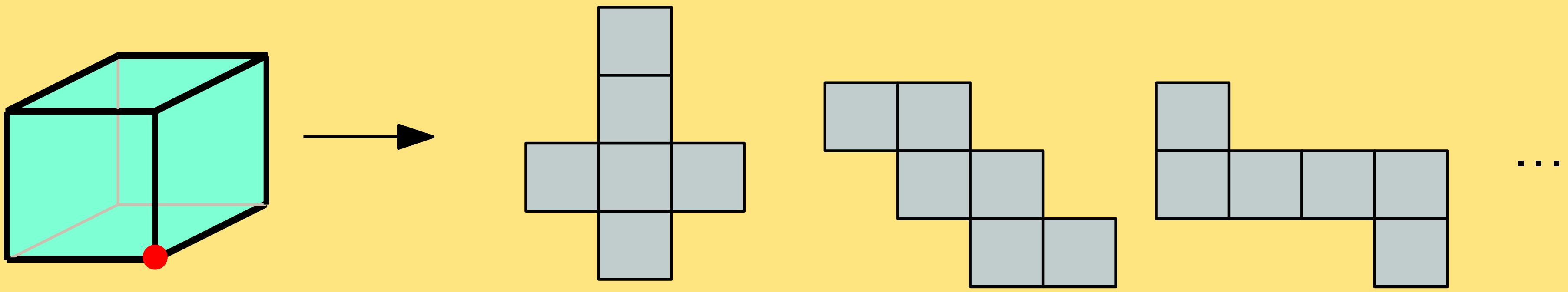


Represent again a map by a tree like structure!

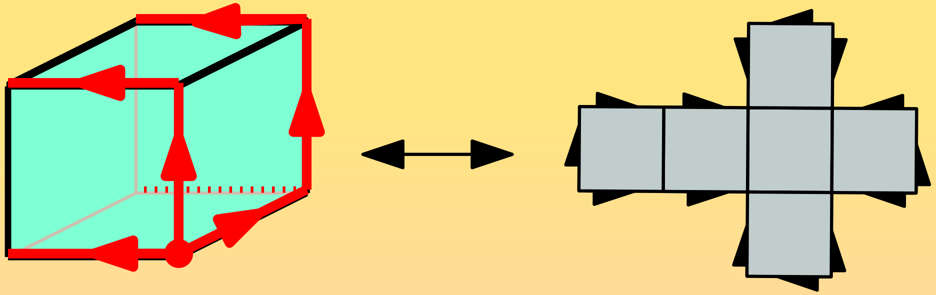
Each exploration algo $\Rightarrow$ a bijection, but what is the set of valid nets?

Exploration algorithms

To a map are associated
many different nets



but a given exploration algorithm associates a canonical net to each map



Represent again a map by a tree like structure!

Each exploration algo $\Rightarrow$ a bijection, but what is the set of valid nets?

Valid nets are easier to describe than exploration trees!

Exploration algorithms

Statement

To many natural families of maps is associated a standard exploration algorithms (breadth first, depth first, Schnyder,...) such that the cut yields *context-free* nets.

Exploration algorithms

Statement

To many natural families of maps is associated a standard exploration algorithms (breadth first, depth first, Schnyder,...) such that the cut yields *context-free* nets.

this statment covers a series of "coherent" theorems

- Cori-Vauquelin 1984, S. 1997, Marcus-S. 1998, Bousquet-Mélou-S. 1999, Poulalhon-S. 2003, Bouttier-di Francesco-Guitter 2004, Fusy-Poulalhon-S. 2005, Bernardi 2006

Exploration algorithms

Statement

To many natural families of maps is associated a standard exploration algorithms (breadth first, depth first, Schnyder,...) such that the cut yields *context-free* nets.

this statment covers a series of "coherent" theorems

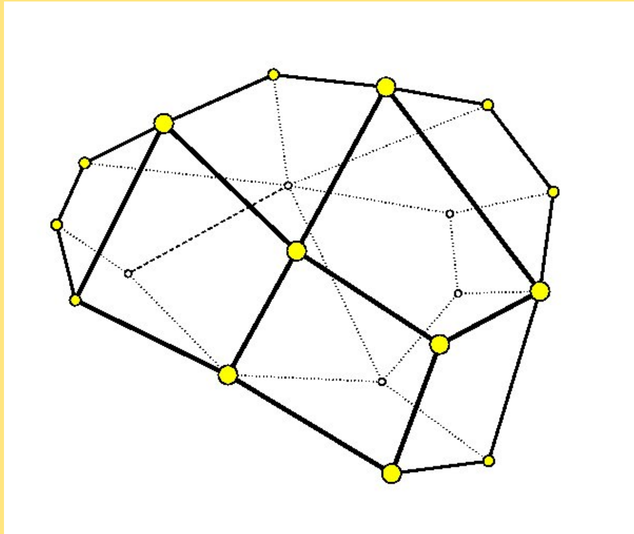
- Cori-Vauquelin 1984, S. 1997, Marcus-S. 1998, Bousquet-Mélou-S. 1999, Poulalhon-S. 2003, Bouttier-di Francesco-Guitter 2004, Fusy-Poulalhon-S. 2005, Bernardi 2006

with various types of applications

- optimal encodings and compact data structures for meshes
- random sampling and automatic drawing of graph and map
- enumeration: maps, ramified coverings, alternating knots...
- random discrete surfaces

Application to discrete random surfaces

Planar quadrangulations (quads) as a model of discretized spheres

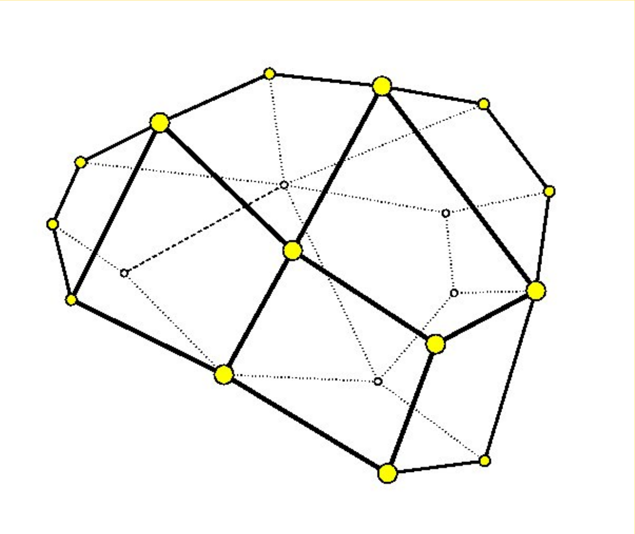


Let $|Q_n|$ be the set of quads with n faces and X_n be a uniform random quad of Q_n:

$$\Pr(X_n = q) = \frac{1}{|Q_n|}, \quad \forall q \in Q_n$$

Application to discrete random surfaces

Planar quadrangulations (quads) as a model of discretized spheres



Let $|Q_n|$ be the set of quads with n faces and X_n be a uniform random quad of Q_n:

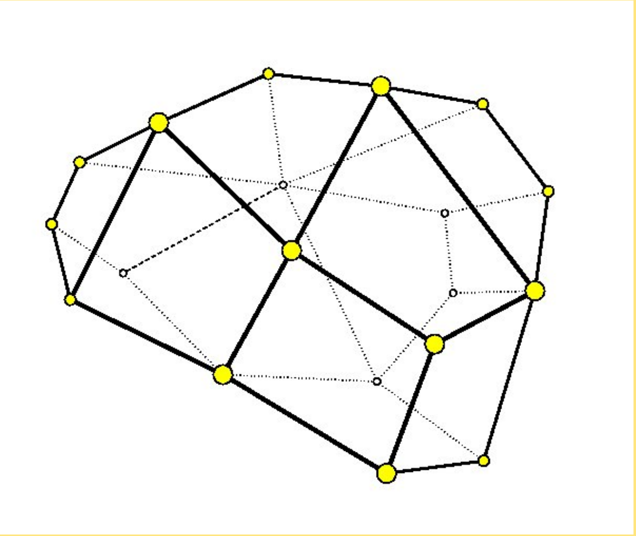
$$\Pr(X_n = q) = \frac{1}{|Q_n|}, \quad \forall q \in Q_n$$

This model of random geometries is called *2d discrete quantum gravity* in statistical φ.

Lots of results via the celebrated method of topological expansion of matrix integrals (Brezin, Itzykson, Parisi, Zuber, 72).

Application to discrete random surfaces

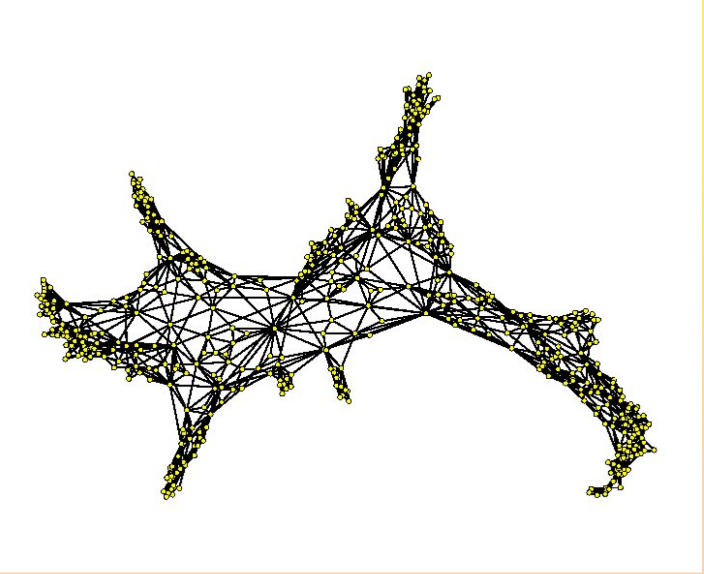
Planar quadrangulations (quads) as a model of discretized spheres



Let $|Q_n|$ be the set of quads with n faces and X_n be a uniform random quad of Q_n:

$$\Pr(X_n = q) = \frac{1}{|Q_n|}, \quad \forall q \in Q_n$$

This model of random geometries is called *2d discrete quantum gravity* in statistical φ.

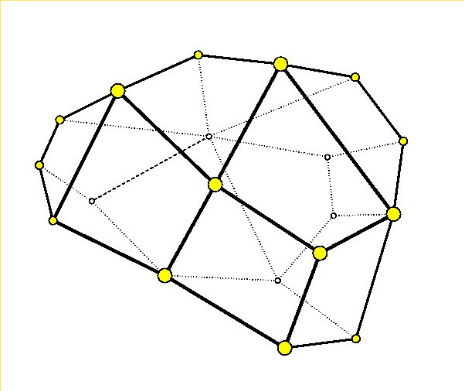


Lots of results via the celebrated method of topological expansion of matrix integrals (Brezin, Itzykson, Parisi, Zuber, 72).

But this approach does not allow to study the intrinsec geometry of these surface!

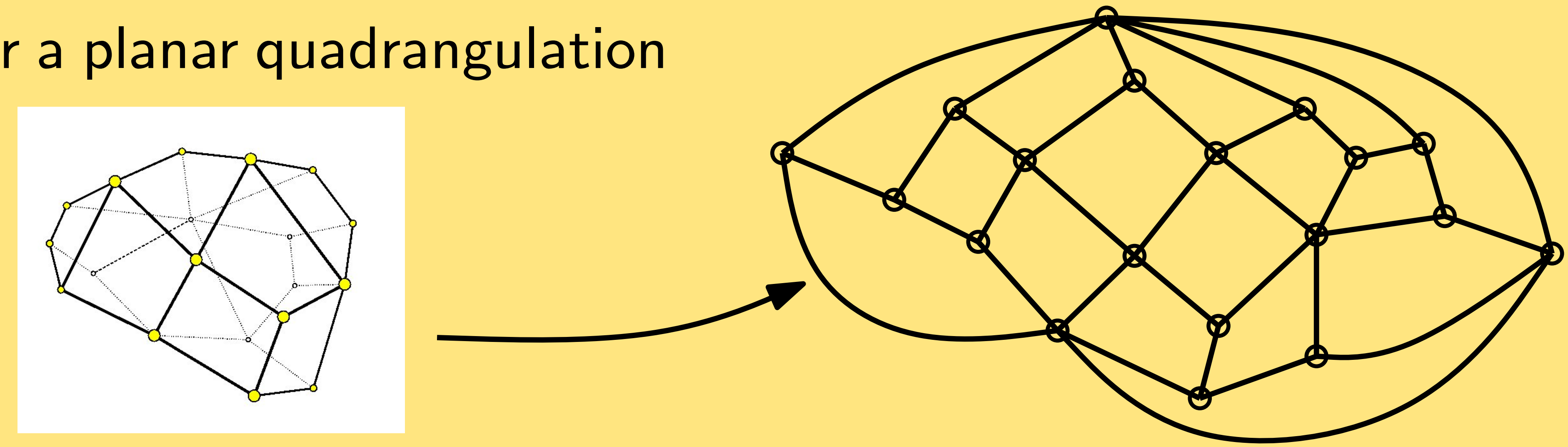
Quadrangulations via breadth first search

Consider a planar quadrangulation



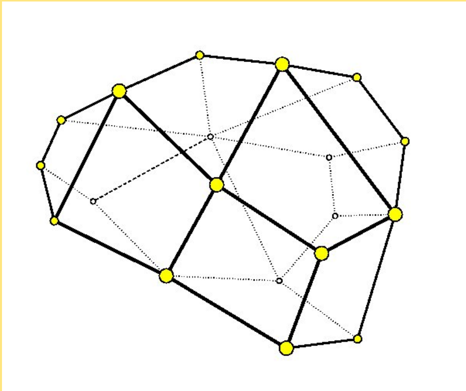
Quadrangulations via breadth first search

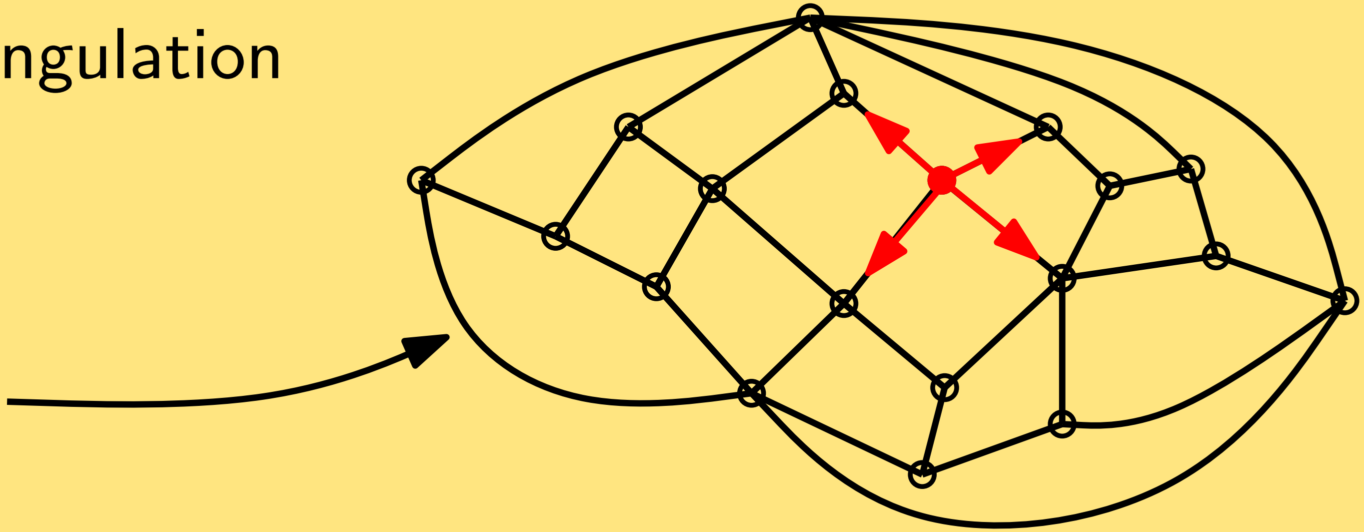
Consider a planar quadrangulation



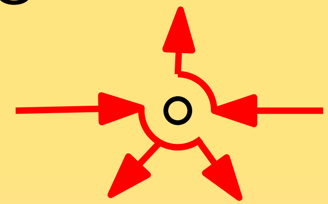
Quadrangulations via breadth first search

Consider a planar quadrangulation



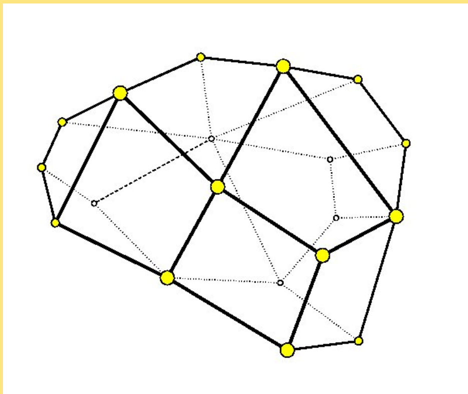


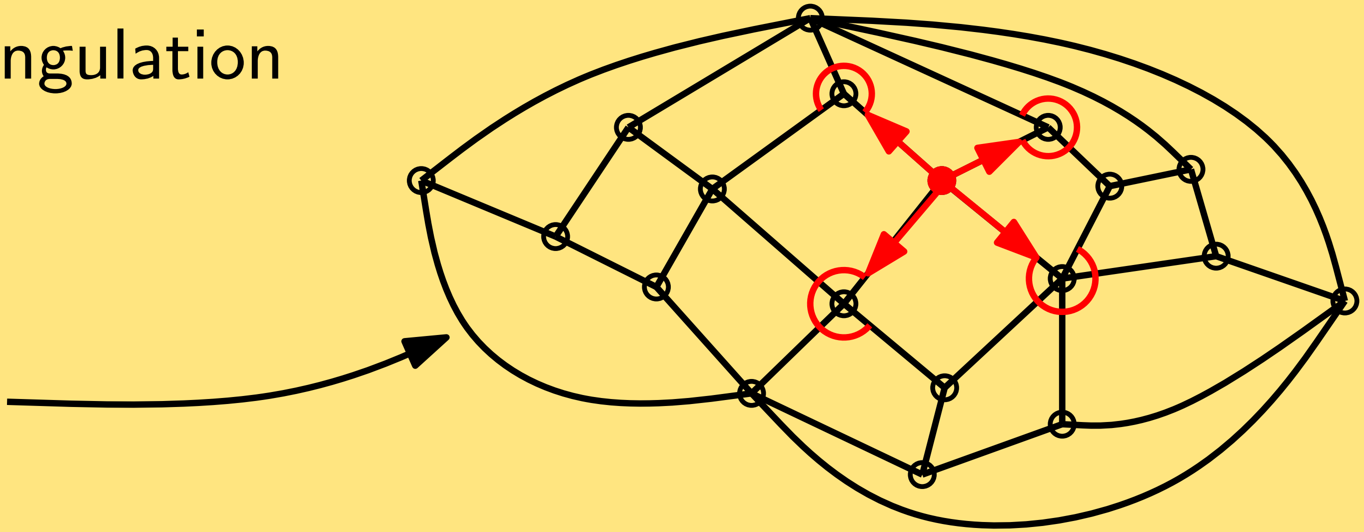
Apply bfs with the rotatoria rule and cut along the flow



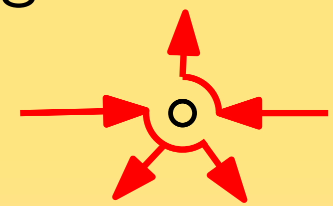
Quadrangulations via breadth first search

Consider a planar quadrangulation



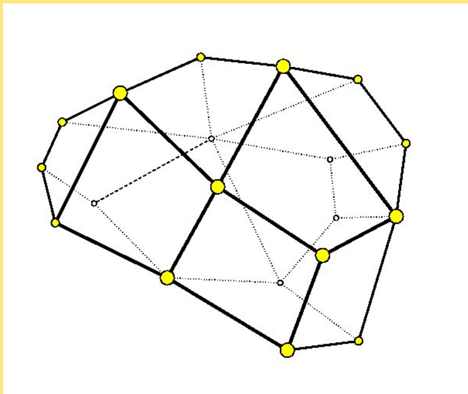


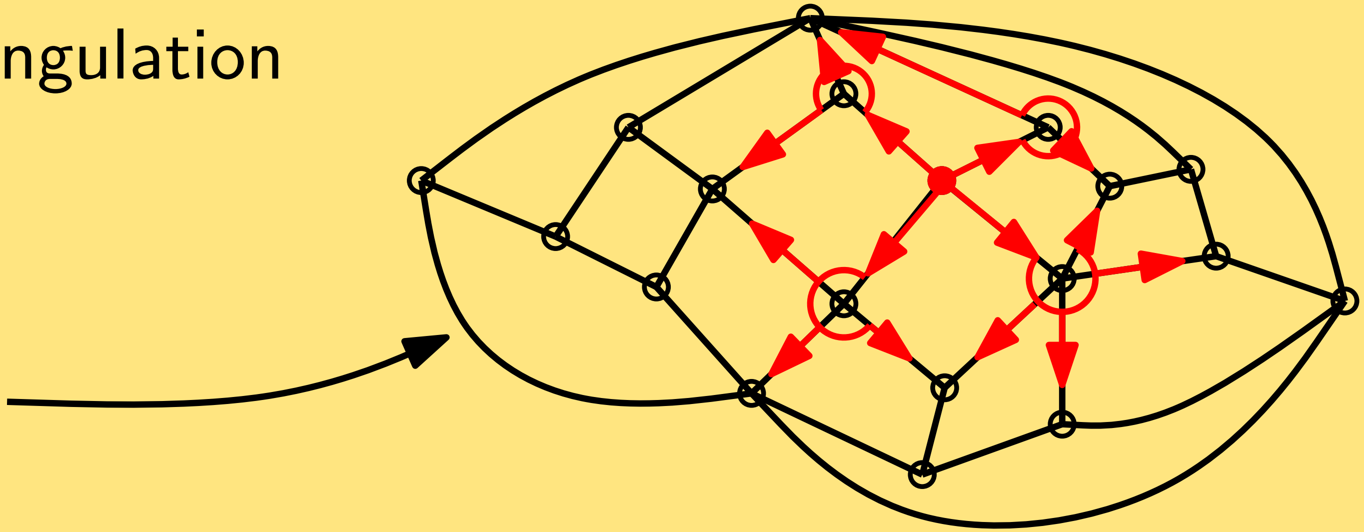
Apply bfs with the rotatoria rule and cut along the flow



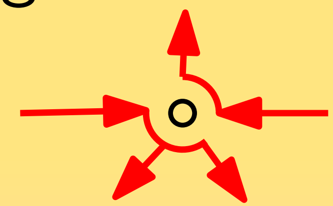
Quadrangulations via breadth first search

Consider a planar quadrangulation



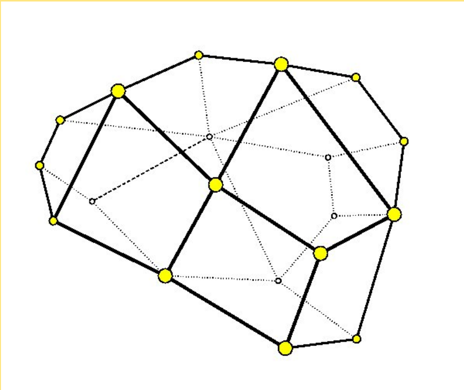


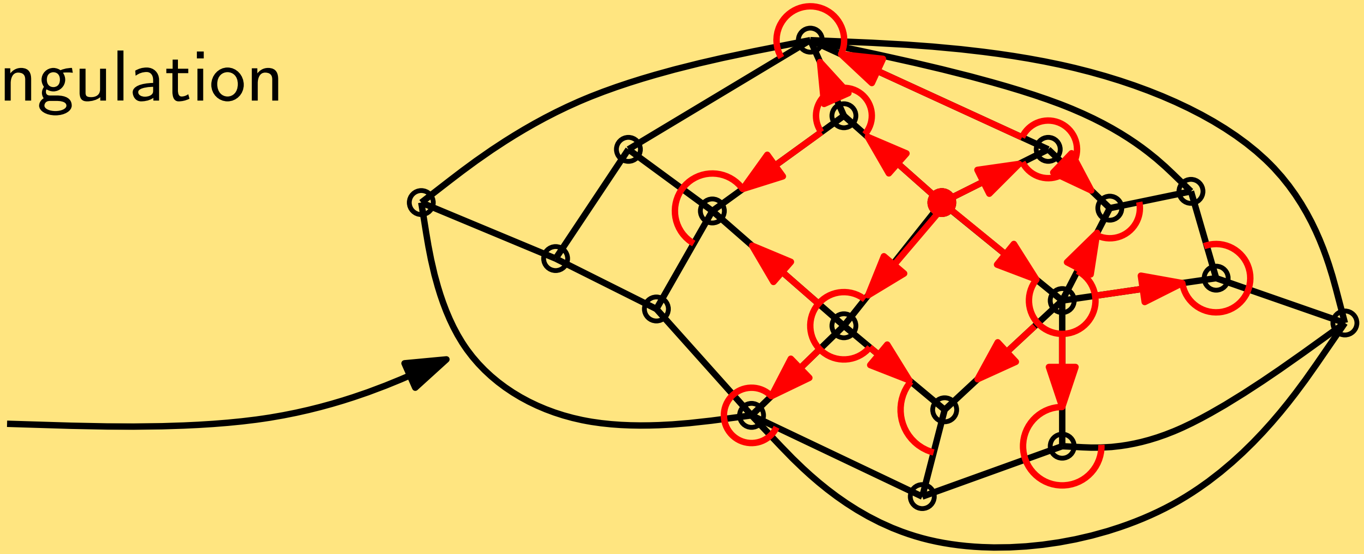
Apply bfs with the rotatoria rule and cut along the flow



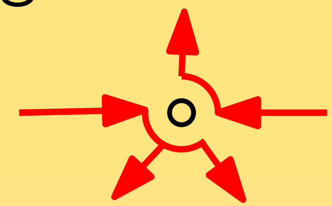
Quadrangulations via breadth first search

Consider a planar quadrangulation



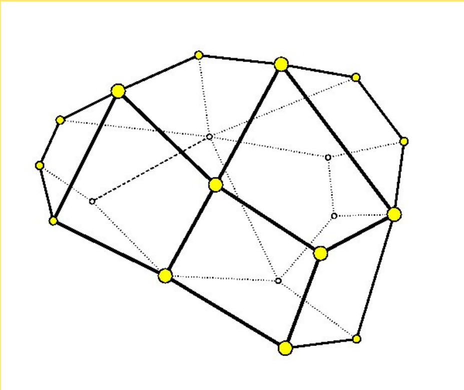


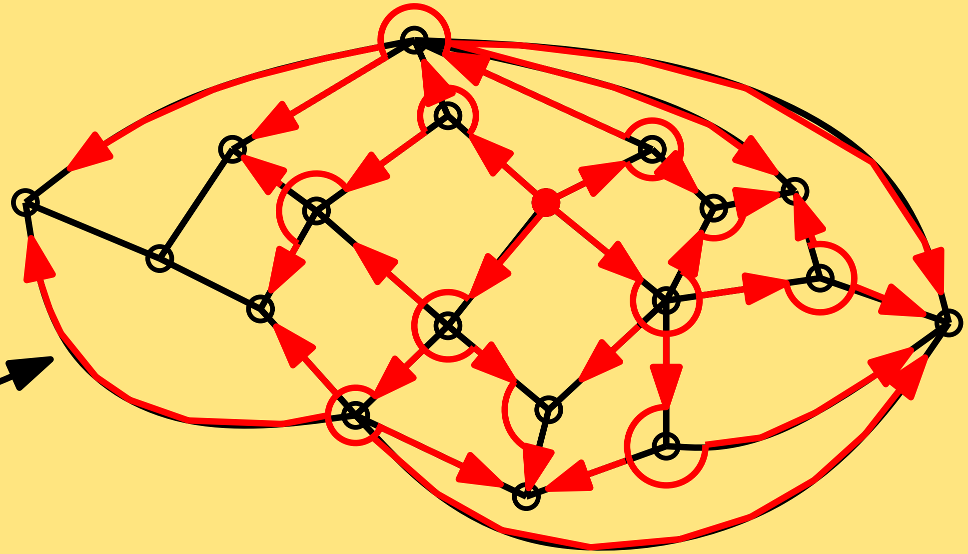
Apply bfs with the rotatoria rule and cut along the flow



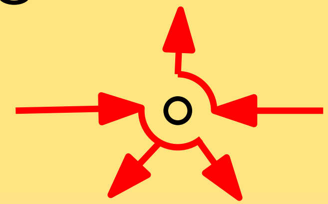
Quadrangulations via breadth first search

Consider a planar quadrangulation



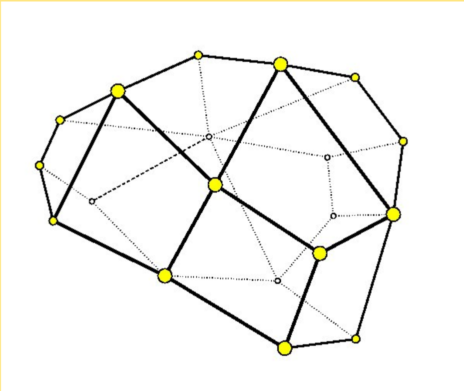


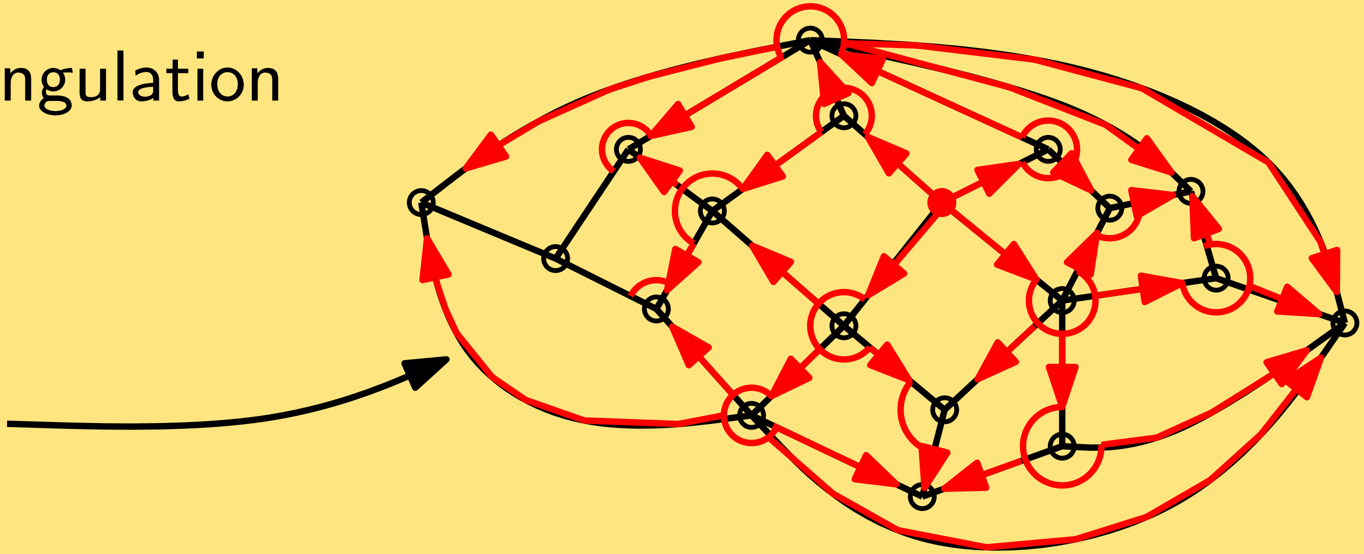
Apply bfs with the rotatoria rule
and cut along the flow



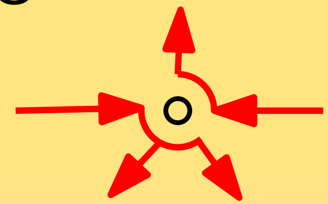
Quadrangulations via breadth first search

Consider a planar quadrangulation



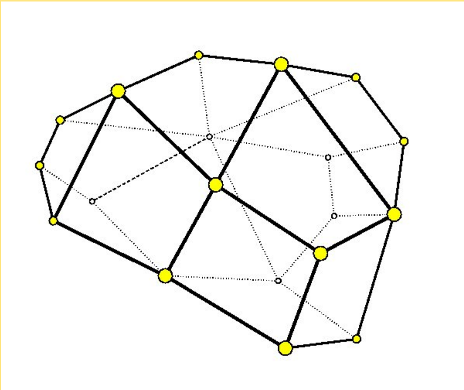


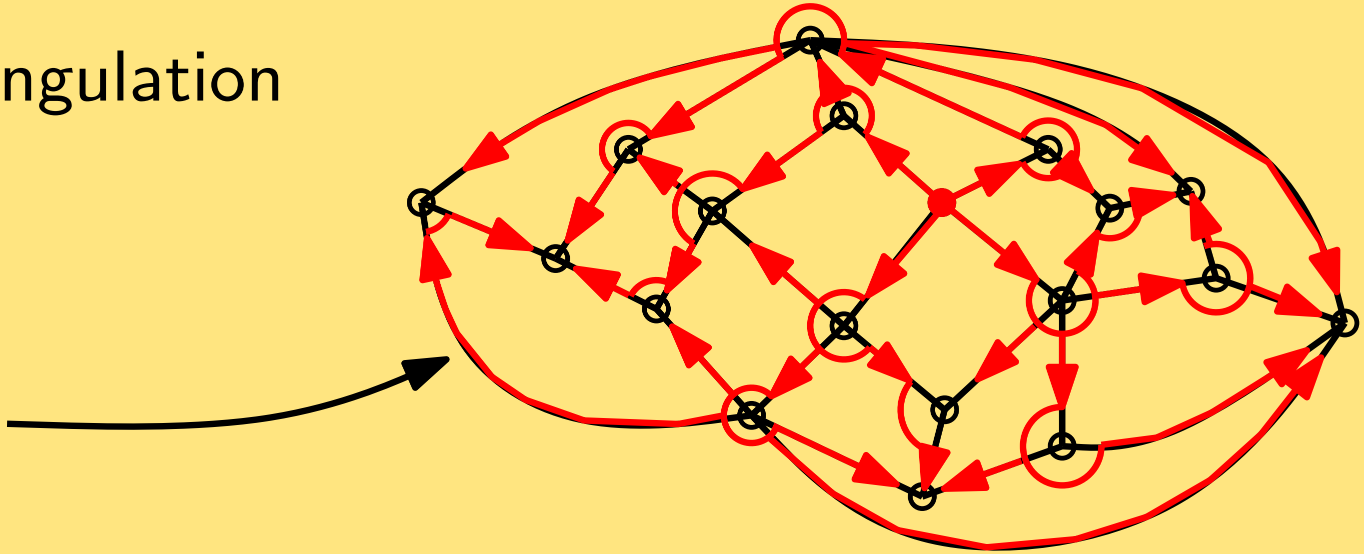
Apply bfs with the rotatoria rule
and cut along the flow



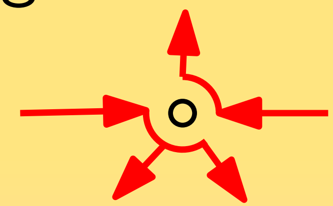
Quadrangulations via breadth first search

Consider a planar quadrangulation



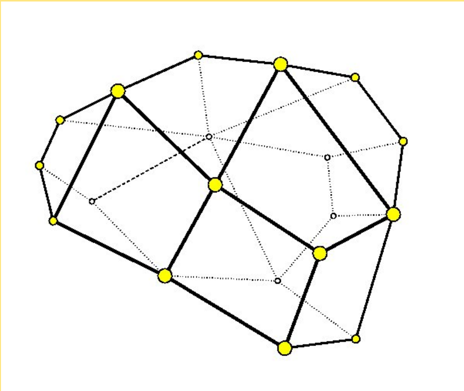


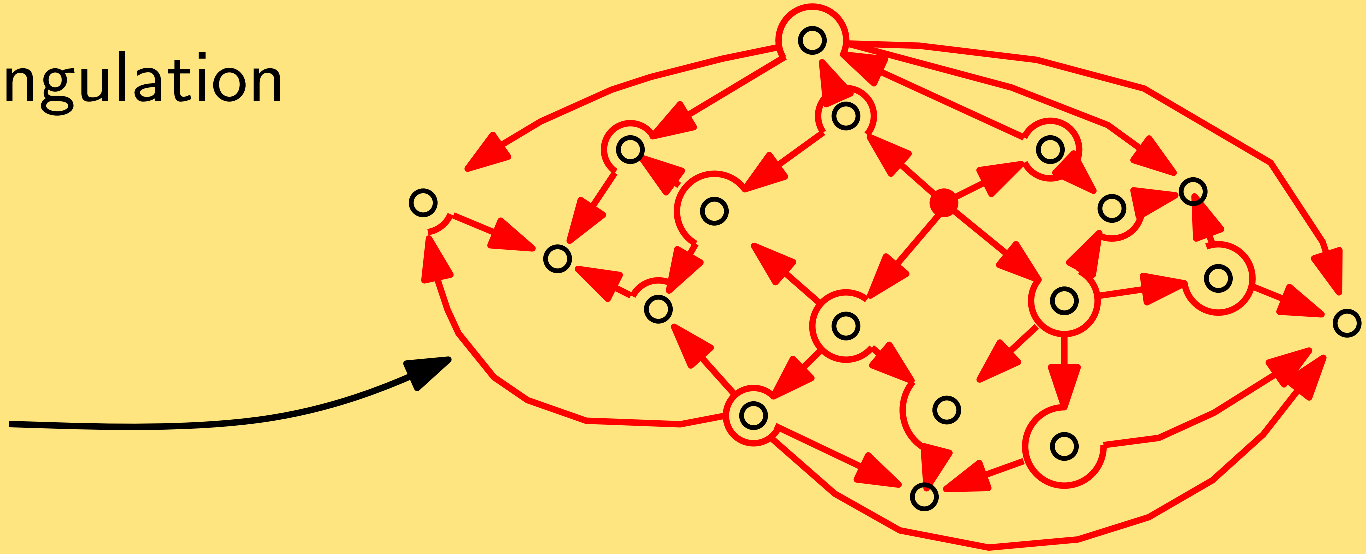
Apply bfs with the rotatoria rule
and cut along the flow



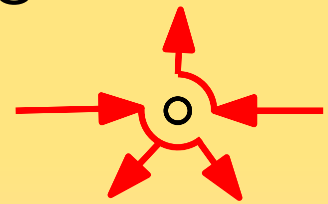
Quadrangulations via breadth first search

Consider a planar quadrangulation



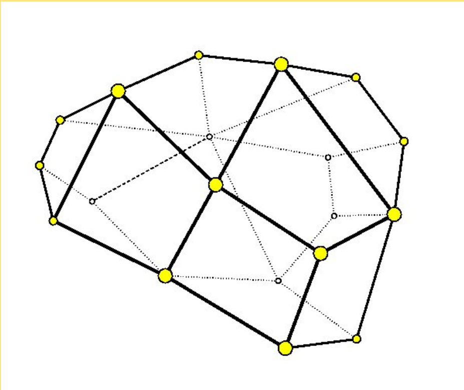


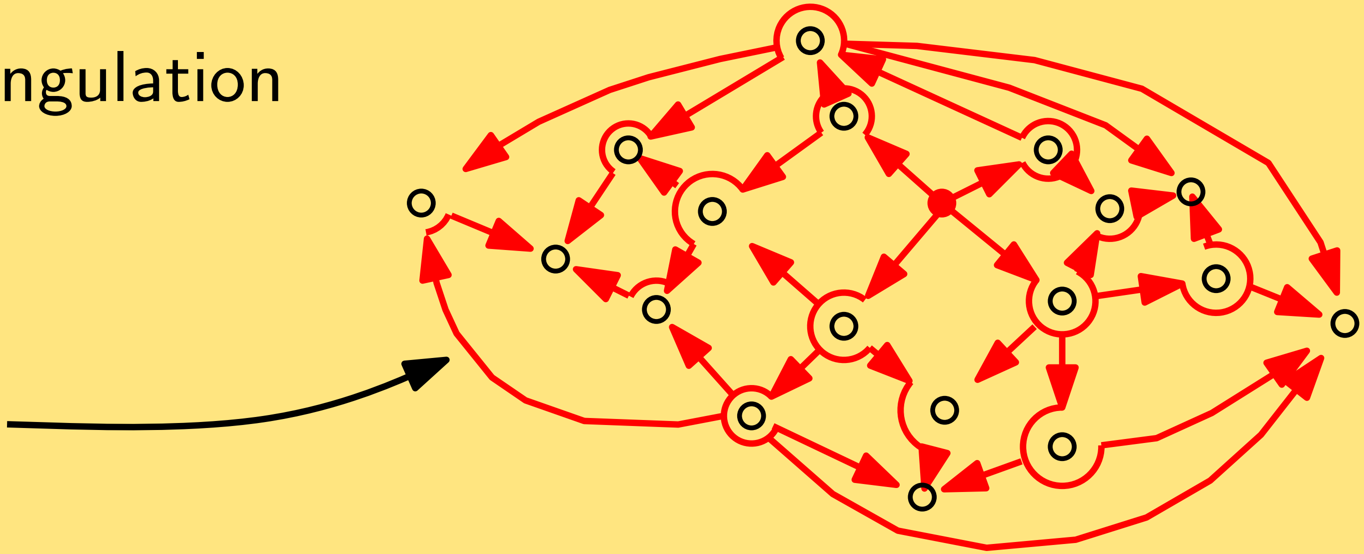
Apply bfs with the rotatoria rule
and cut along the flow



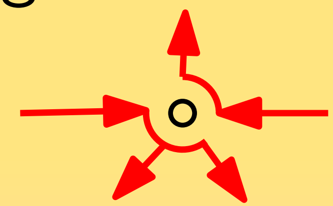
Quadrangulations via breadth first search

Consider a planar quadrangulation

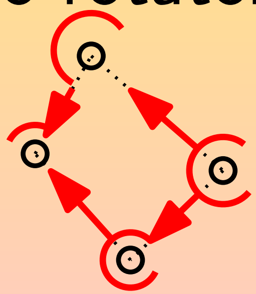




Apply bfs with the rotatoria rule
and cut along the flow

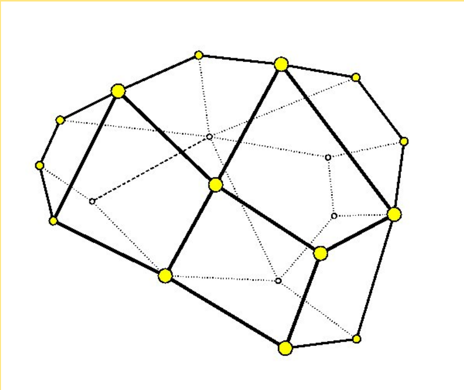


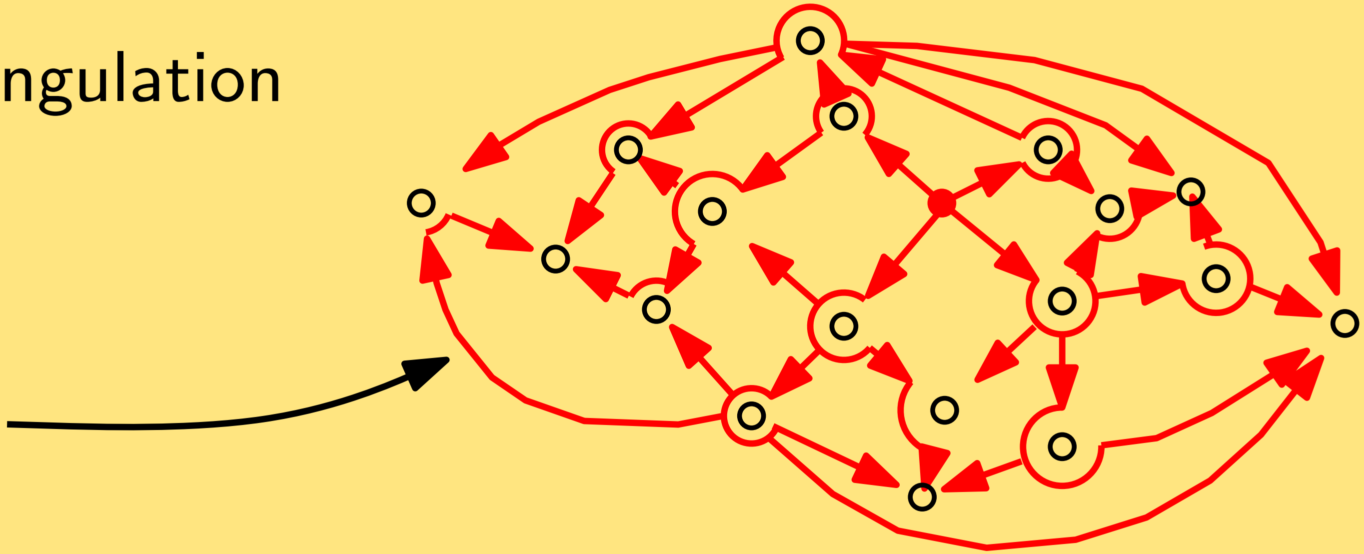
Each face sees exactly two rotatoria



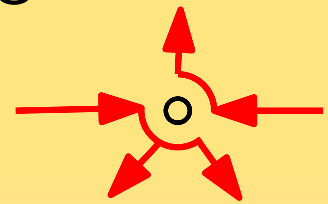
Quadrangulations via breadth first search

Consider a planar quadrangulation



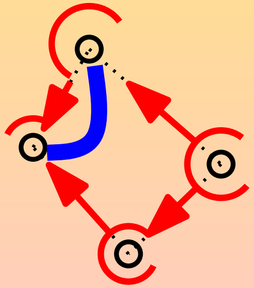


Apply bfs with the rotatoria rule and cut along the flow



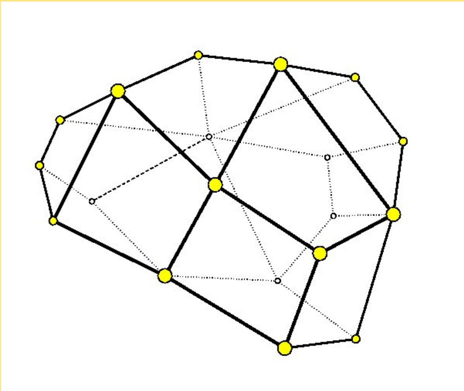
Each face sees exactly two rotatoria

Join these 2 rotatoria!

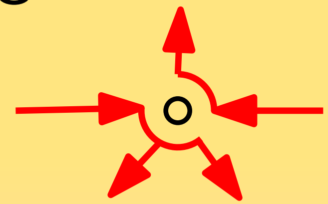


Quadrangulations via breadth first search

Consider a planar quadrangulation

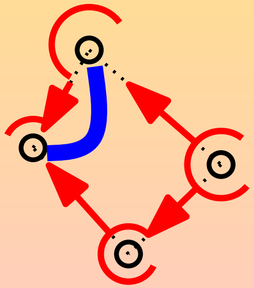


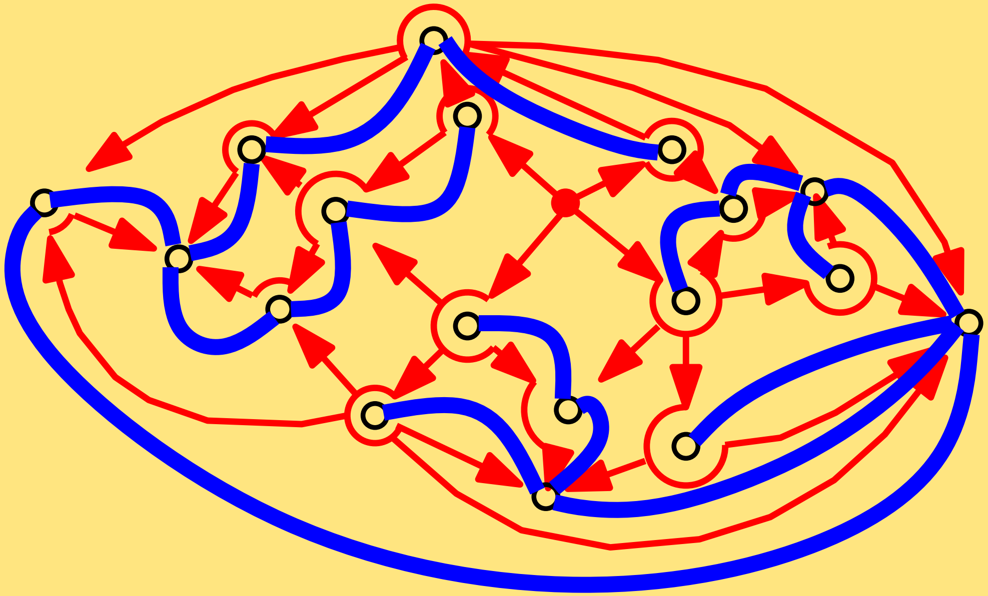
Apply bfs with the rotatoria rule and cut along the flow



Each face sees exactly two rotatoria

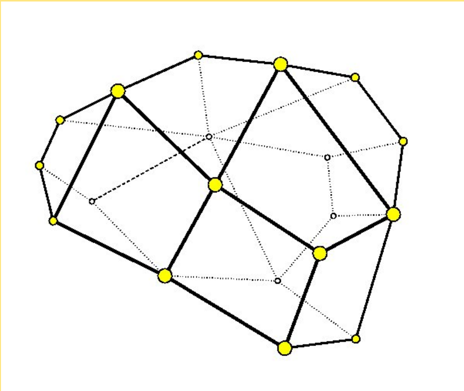
Join these 2 rotatoria!



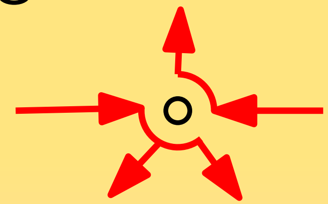


Quadrangulations via breadth first search

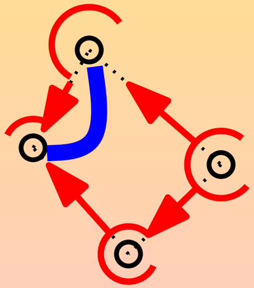
Consider a planar quadrangulation



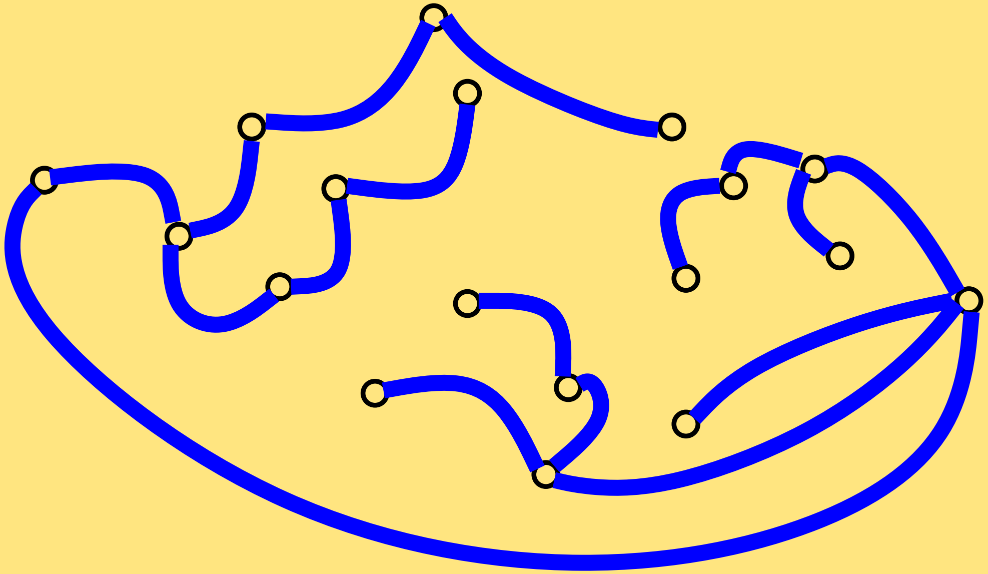
Apply bfs with the rotatoria rule and cut along the flow



Each face sees exactly two rotatoria



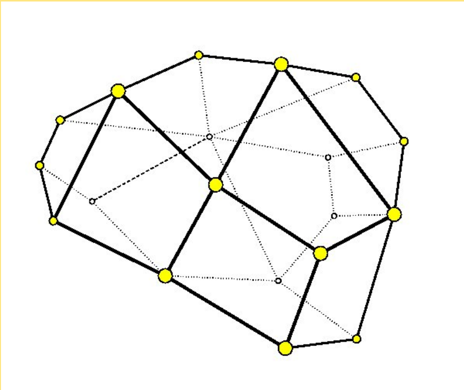
Join these 2 rotatoria!



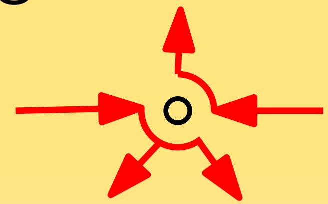
The result is **tree**.

Quadrangulations via breadth first search

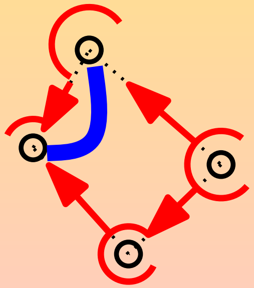
Consider a planar quadrangulation



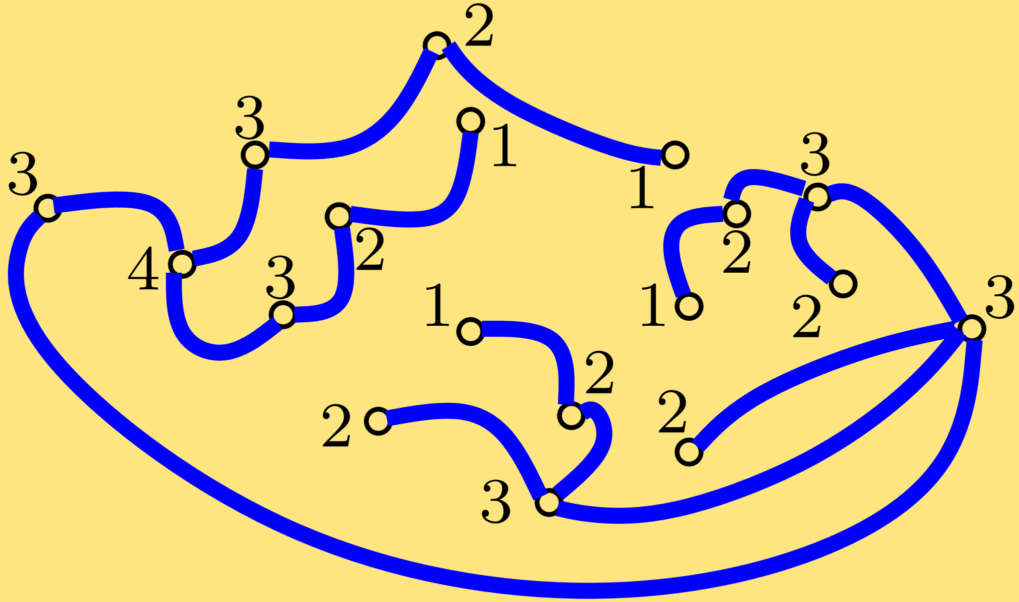
Apply bfs with the rotatoria rule and cut along the flow



Each face sees exactly two rotatoria



Join these 2 rotatoria!

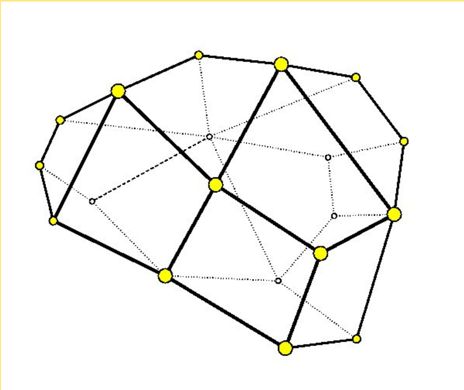


The result is **tree**.

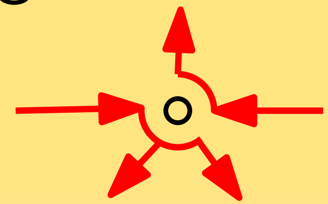
Label vertices by the round at which they were visited by bfs.

Quadrangulations via breadth first search

Consider a planar quadrangulation

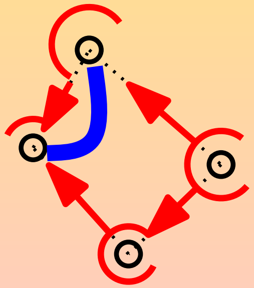


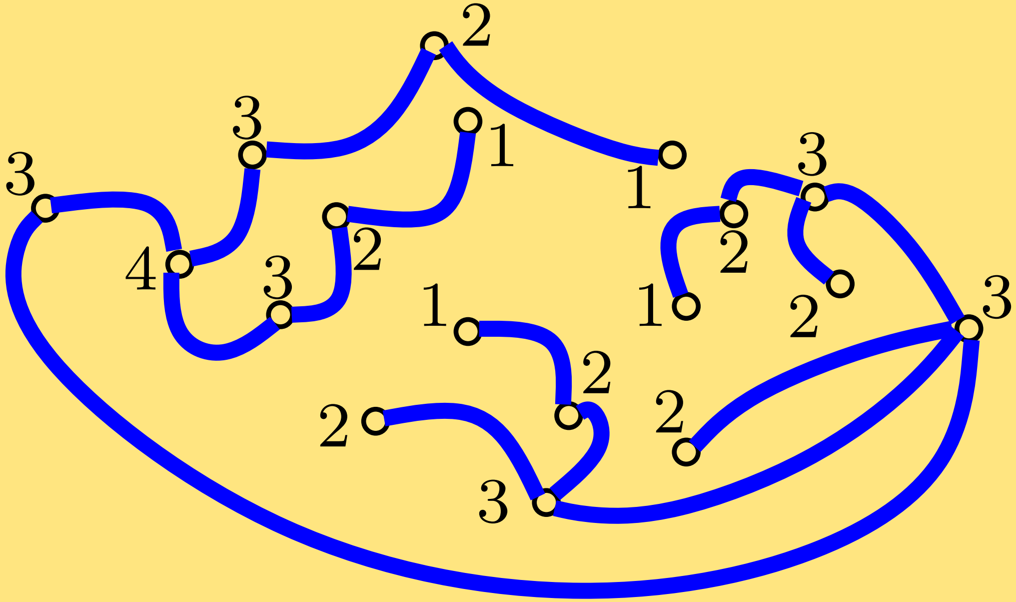
Apply bfs with the rotatoria rule and cut along the flow



Each face sees exactly two rotatoria

Join these 2 rotatoria!



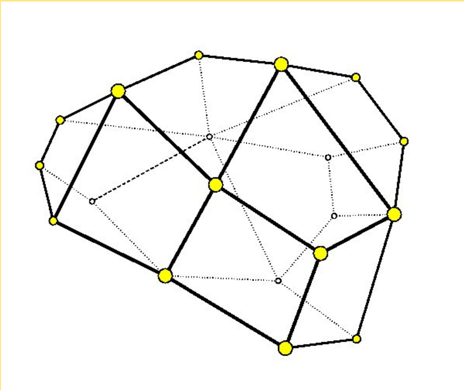


The result is a well labeled **tree**.

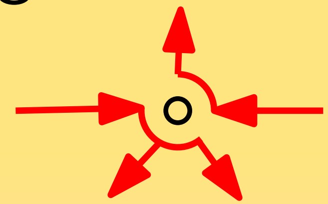
Label vertices by the round at which they were visited by bfs.

Quadrangulations via breadth first search

Consider a planar quadrangulation

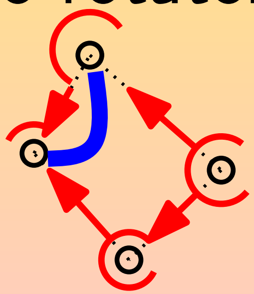


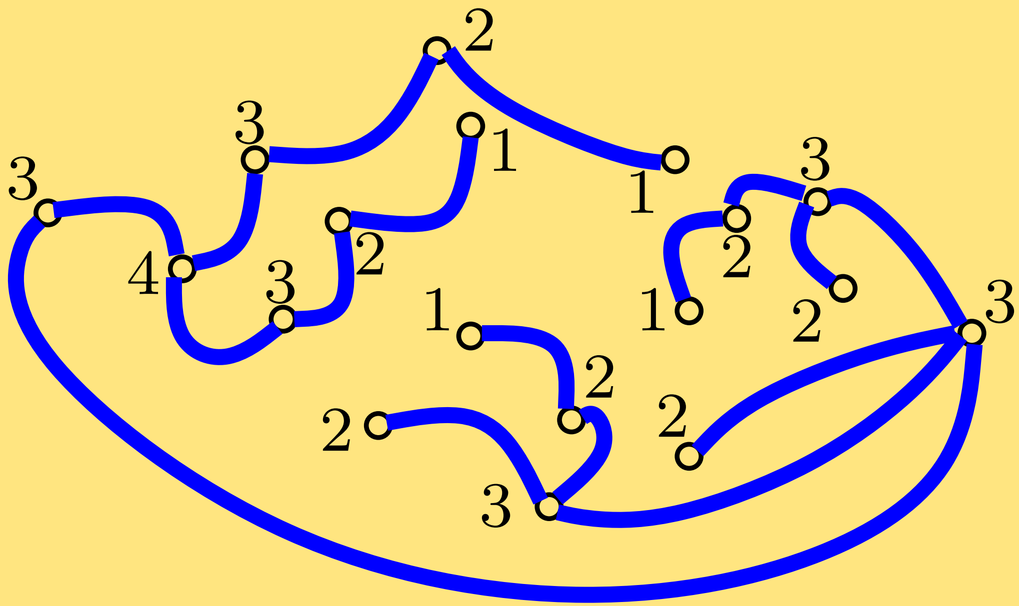
Apply bfs with the rotatoria rule and cut along the flow



Each face sees exactly two rotatoria

Join these 2 rotatoria!





The result is a well labeled **tree**.

Label vertices by the round at which they were visited by bfs.

Theorem. This is a bijection.

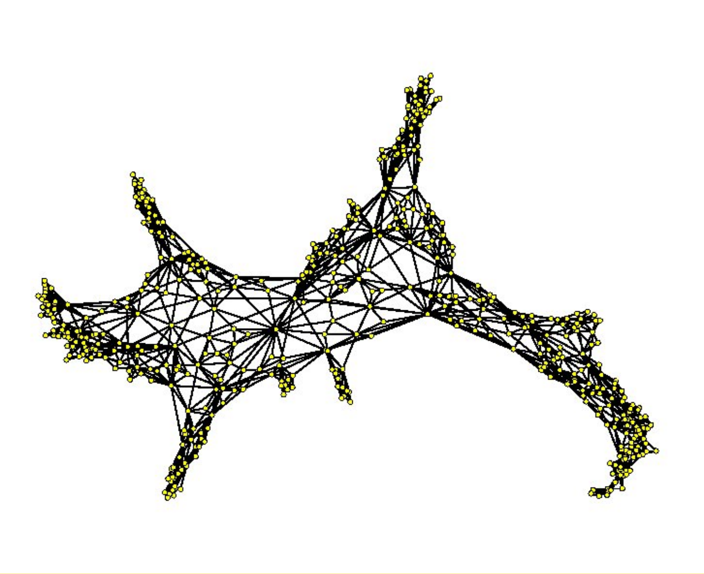
X_n: pointed quads, n faces

$\wr$

T_n: well labeled trees, n vtx

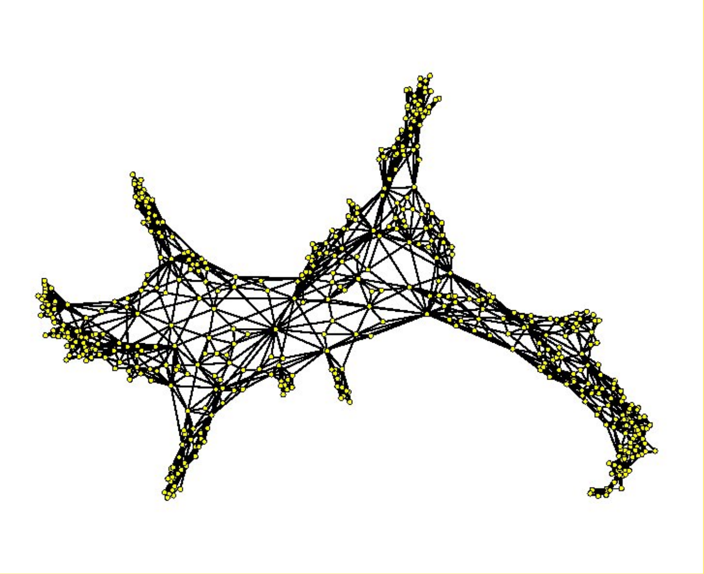
Quadrangulations via breadth first search

use breadth first search to study the geometry



distance between 2 pts = nb of edges on a path

Quadrangulations via breadth first search



use breadth first search to study the geometry

distance between 2 pts = nb of edges on a path

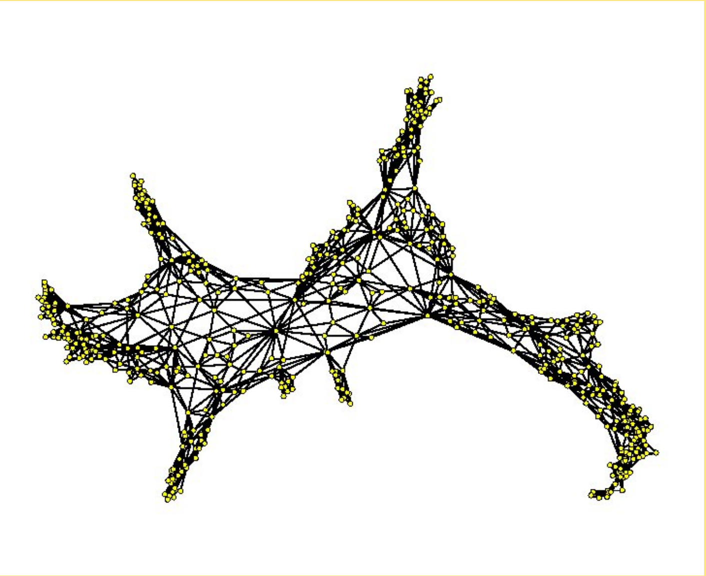
distance from basepoint

= round of exploration by bfs

⇒ breadth first search computes distances:

Quadrangulations via breadth first search

use breadth first search to study the geometry



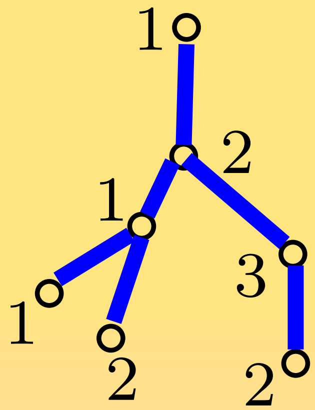
distance between 2 pts = nb of edges on a path

distance from basepoint

= round of exploration by bfs

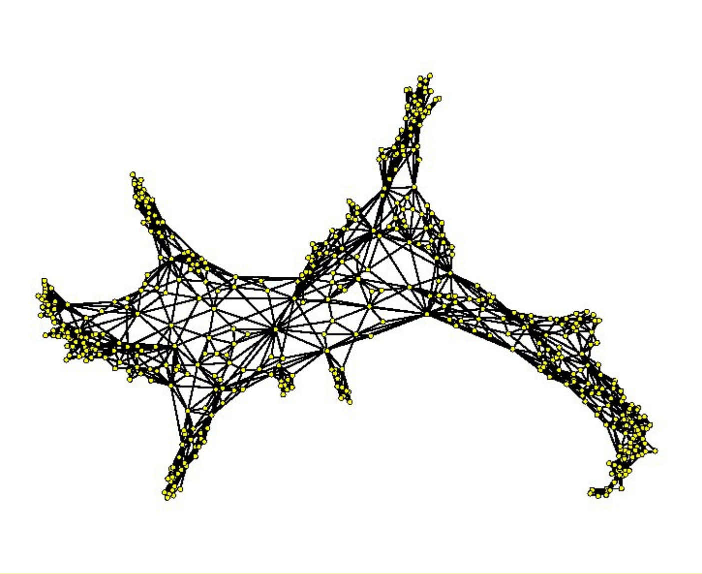
⇒ breadth first search computes distances:

- labels of the tree record distances from the basepoint



Quadrangulations via breadth first search

use breadth first search to study the geometry

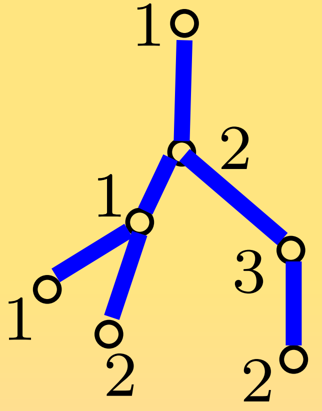


distance between 2 pts = nb of edges on a path

distance from basepoint

= round of exploration by bfs

$\Rightarrow$ breadth first search computes distances:

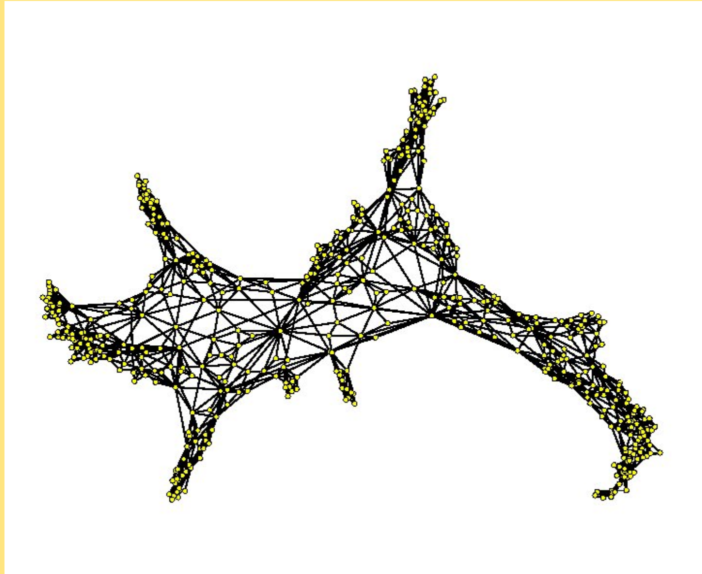


- labels of the tree record distances from the basepoint
- the height of a random tree of size n is $n^{1/2}$
- the random walk of labels on a branch of length ℓ has max about $\ell^{1/2}$

$\Rightarrow$ typical labels are of order $n^{1/4}$.

Quadrangulations via breadth first search

use breadth first search to study the geometry

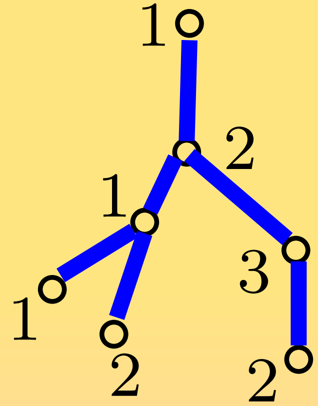


distance between 2 pts $=$ nb of edges on a path

distance from basepoint

$=$ round of exploration by bfs

$\Rightarrow$ breadth first search computes distances:



- labels of the tree record distances from the basepoint
- the height of a random tree of size n is $n^{1/2}$
- the random walk of labels on a branch of length ℓ has max about $\ell^{1/2}$

$\Rightarrow$ typical labels are of order $n^{1/4}$.

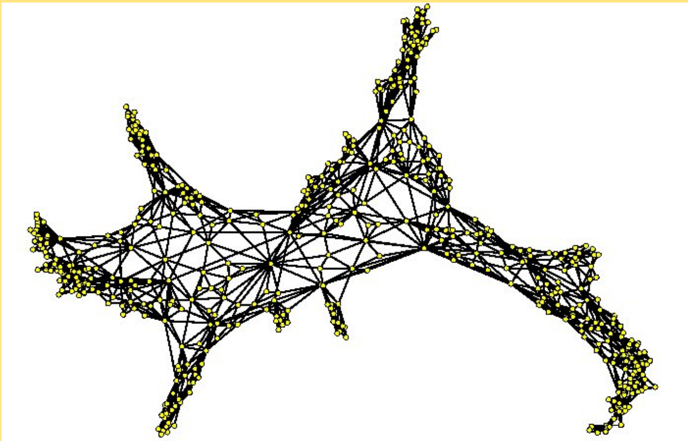
Theorem (Chassaing-S, 2004).
The distance between 2 random vertices of X_n is of order $n^{1/4}$.

This approach was pursued by Chassaing-Durhuus (2005), Marckert-Mokkadem (2004), Miermond (2005), Weill (2006)... culminating with

Some properties of random discrete surfaces

This approach was pursued by Chassaing-Durhuus (2005), Marckert-Mokkadem (2004), Miermond (2005), Weill (2006)... culminating with

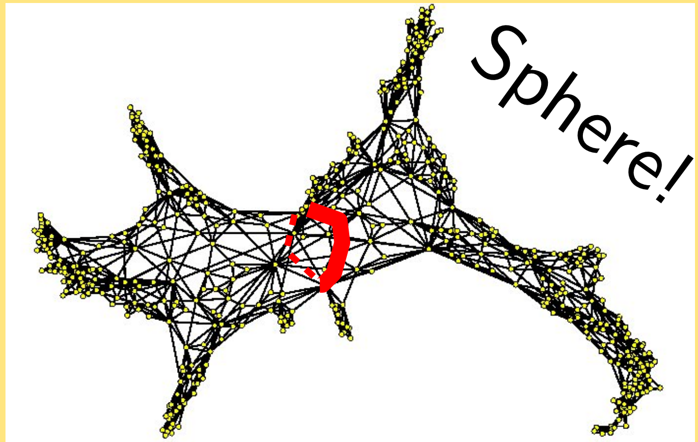
Theorem (Le Gall, 2006). Rescaled planar quadrangulations converge in the large size limit to a *random continuum planar map* that has spherical topology.



Some properties of random discrete surfaces

This approach was pursued by Chassaing-Durhuus (2005), Marckert-Mokkadem (2004), Miermond (2005), Weill (2006)... culminating with

Theorem (Le Gall, 2006). Rescaled planar quadrangulations converge in the large size limit to a *random continuum planar map* that has spherical topology.

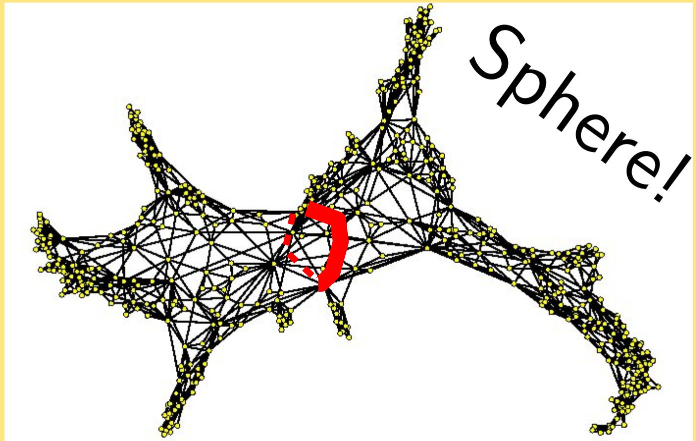


In particular there exists no separating cycle of size $\ll n^{1/4}$.

Some properties of random discrete surfaces

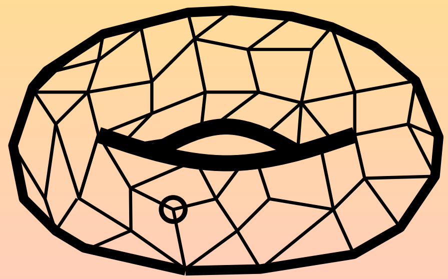
This approach was pursued by Chassaing-Durhuus (2005), Marckert-Mokkadem (2004), Miermond (2005), Weill (2006)... culminating with

Theorem (Le Gall, 2006). Rescaled planar quadrangulations converge in the large size limit to a *random continuum planar map* that has spherical topology.



In particular there exists no separating cycle of size $\ll n^{1/4}$.

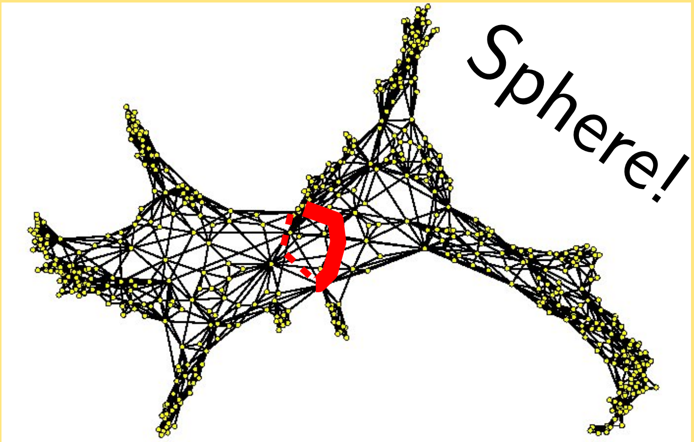
The bfs exploration works also for higer genus surfaces:
Theorem (Chapuy-Marcus-S. 2006) The distance between 2 random vertices of a random quad X_n^g of genus g is of order $n^{1/4}$.



Some properties of random discrete surfaces

This approach was pursued by Chassaing-Durhuus (2005), Marckert-Mokkadem (2004), Miermond (2005), Weill (2006)... culminating with

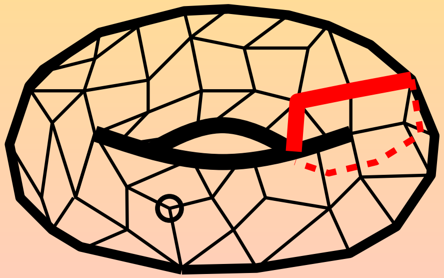
Theorem (Le Gall, 2006). Rescaled planar quadrangulations converge in the large size limit to a *random continuum planar map* that has spherical topology.



In particular there exists no separating cycle of size $\ll n^{1/4}$.

The bfs exploration works also for higer genus surfaces:
Theorem (Chapuy-Marcus-S. 2006) The distance between 2 random vertices of a random quad X_n^g of genus g is of order $n^{1/4}$.

Conjectures.
There is no non-contractible cycles with size $\ll n^{1/4}$.
The rescaled continuum limit exists and has genus g.



A conjecture on random graphs with low genus

Let Y_n^g be a uniform random connected labelled graphs with n vertices that can be embedded on a surface of genus g.

For instance Y_n^0 is a random connected planar graph with n vertices.

A conjecture on random graphs with low genus

Let Y_n^g be a uniform random connected labelled graphs with n vertices that can be embedded on a surface of genus g.

For instance Y_n^0 is a random connected planar graph with n vertices.

Conjecture. The graph Y_n^g is a.s. composed of a 3-connected graph $Core(Y)$ of size $\Theta(n)$ with edges replaced by small planar networks and with small pending planar components.

Moreover $Core(Y)$ a.s. has minimal genus g and has a unique minimal embedding. The small parts have size $O(n^{2/3})$.

In the rescaled limit, Y_n^g converge to the same continuum random map of genus g as X_n^g.

Cf. McDiarmid, Noy, Steger's talks for proofs...

Many thanks for your attention !

Many thanks to my collaborators!