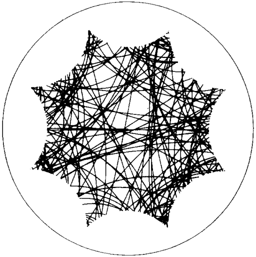
Delaunay triangulations of a family of symmetric hyperbolic surfaces in practice

Iordan Iordanov

Ph.D. defense
12 March 2019 | Nancy, France

Most symmetric surface of genus 2: Bolza

Mathematical physics

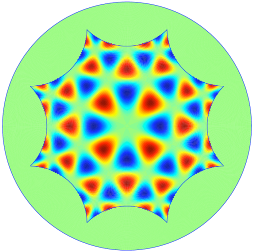


[Balazs, Voros '86]

chaos

[Sausset, Tarjus, Viot '08]

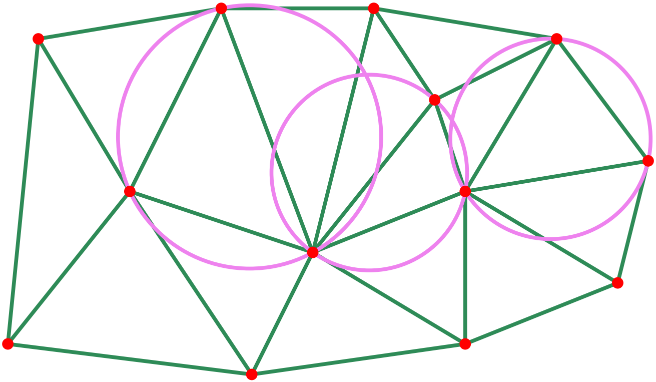
glass forming liquid



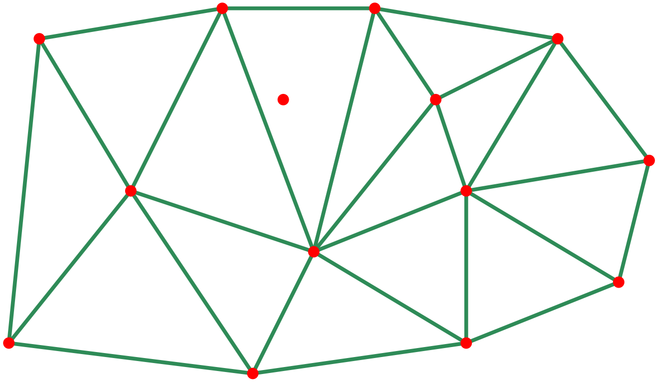
[Chossat, Faye, Faugeras '11]

visual perception of textures

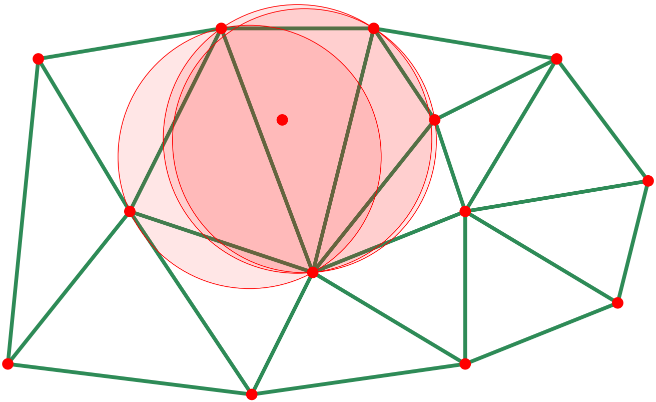
What is a Delaunay triangulation?



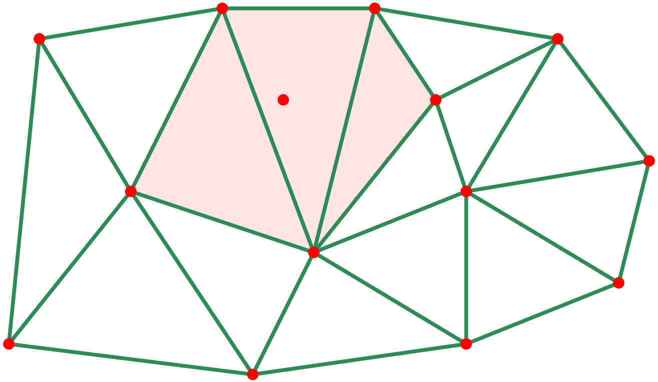
Bowyer's algorithm



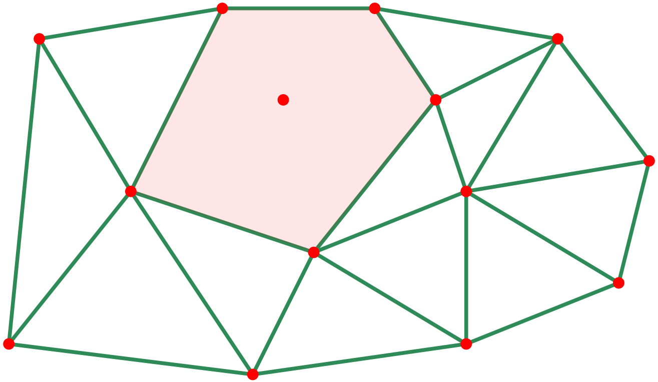
Bowyer's algorithm



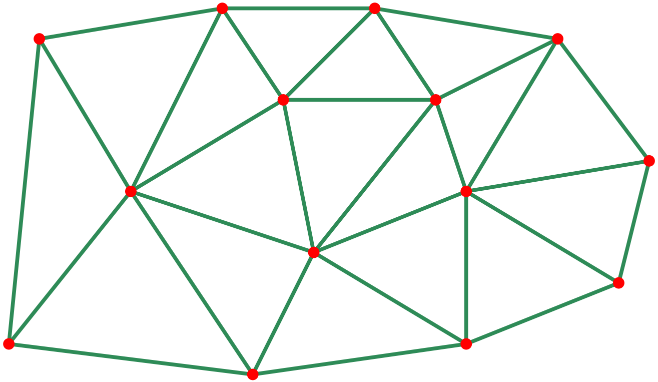
Bowyer's algorithm



Bowyer's algorithm



Bowyer's algorithm



The Computational Geometry Algorithms Library CGAL

- CGAL project: founded in 1996
- CGAL library: open source since 2003
- Today:
  - open-source library for CG
  - over 100 packages (triangulation, mesh, arrangements...)
  - used in GIS, CAD, medical imaging, robotics...

Our goal:

implement triangulations of hyperbolic surfaces in CGAL

State of the art

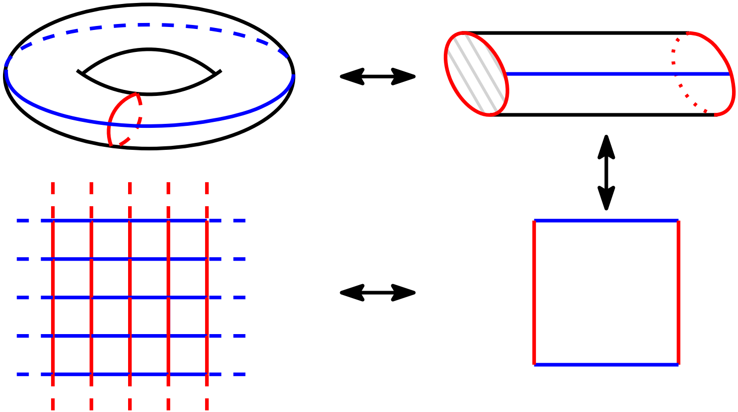
Closed Euclidean manifolds

- Algorithms
  - 2D [Mazón, Recio '97]
  - 3D [Dolbilin, Huson '97]
  - dD [Caroli, Teillaud '16]
- Software (square/cubic flat torus)
  - 2D [Kruithof '13]
  - 3D [Caroli, Teillaud '09]

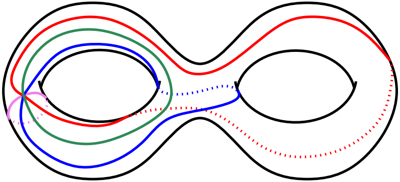
Closed hyperbolic manifolds

- Algorithms
  - 2D, genus 2 [Bogdanov, Teillaud, Vegter, SoCG'16]

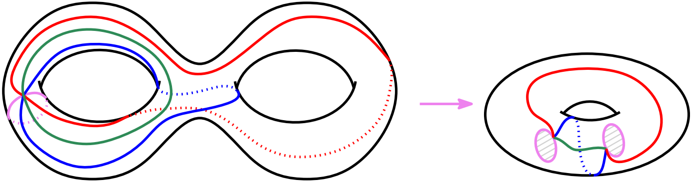
Cutting open a flat torus



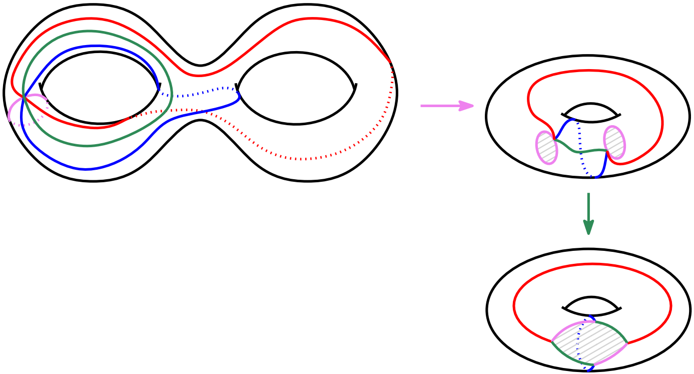
Cutting open a double torus



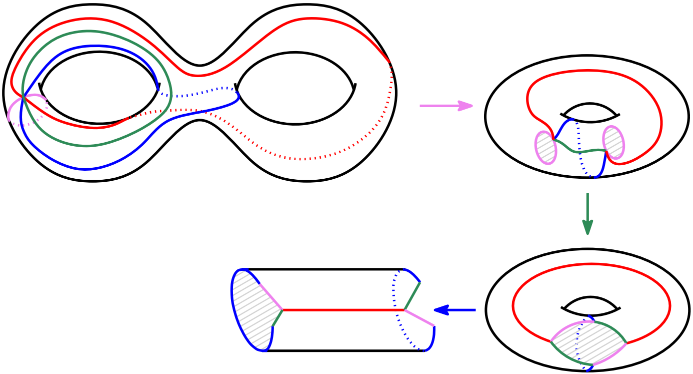
Cutting open a double torus



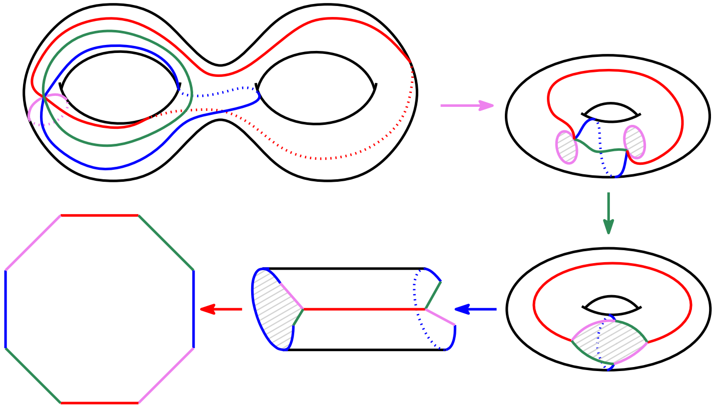
Cutting open a double torus



Cutting open a double torus

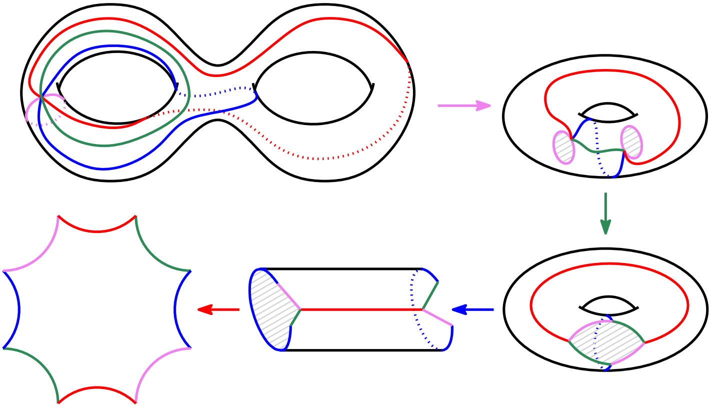


Cutting open a double torus



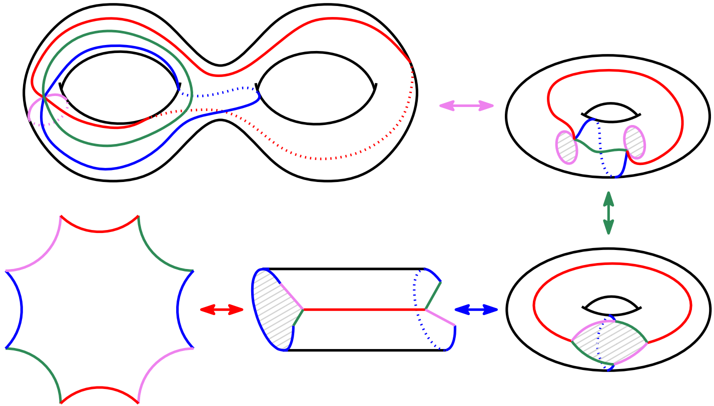
cannot tile Euclidean plane!

Cutting open a double torus

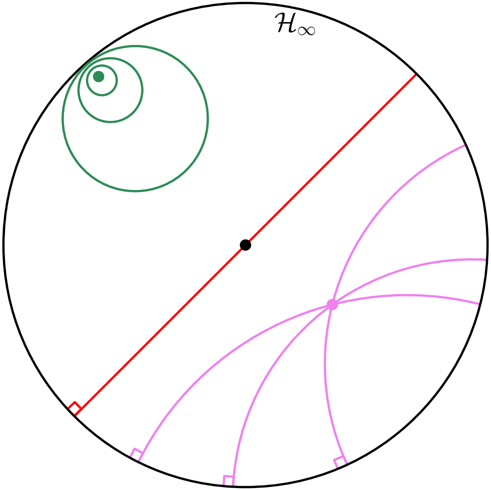


can tile hyperbolic plane!

Cutting open a double torus

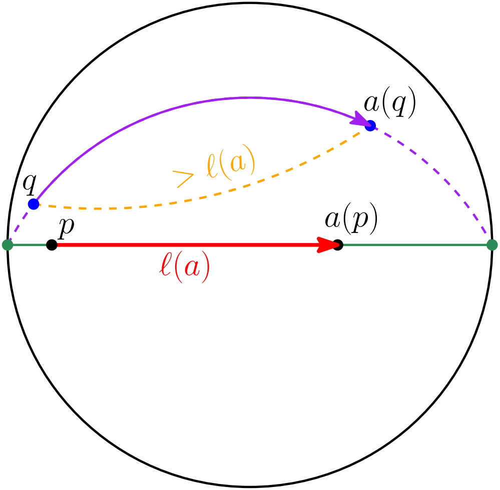


Poincaré model of the hyperbolic plane $\mathbb{H}^2$



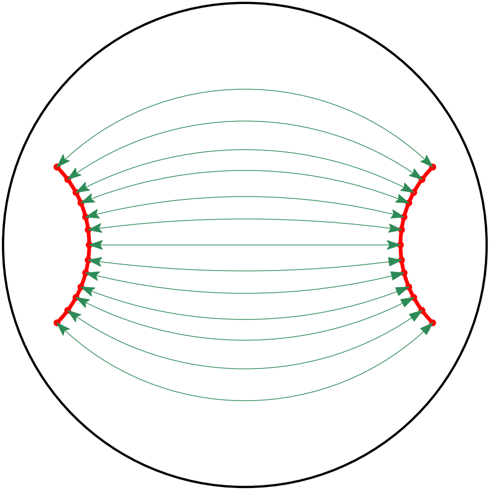
Hyperbolic translations

Special case: axis = diameter



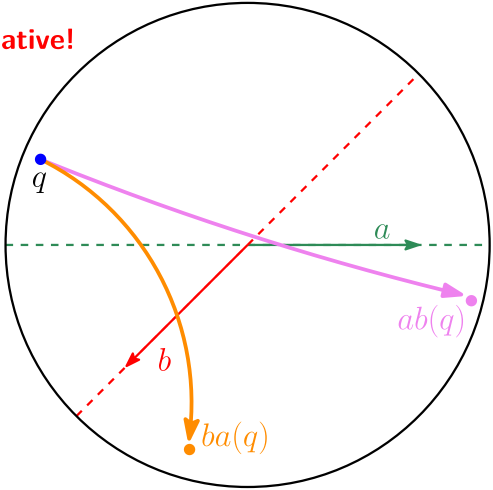
Hyperbolic translations

Side-pairing transformation

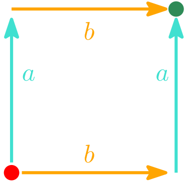


Hyperbolic translations

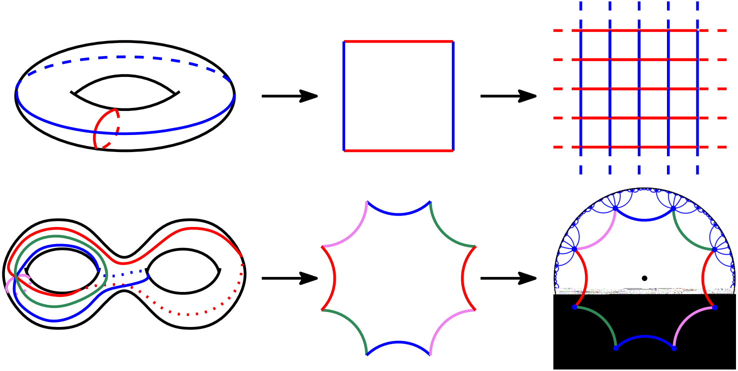
Non-commutative!



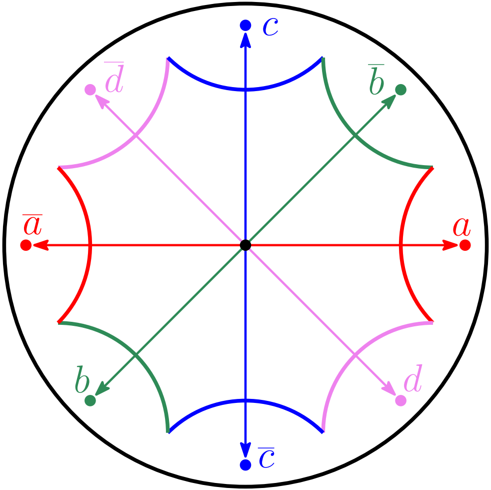
Compare with Euclidean:



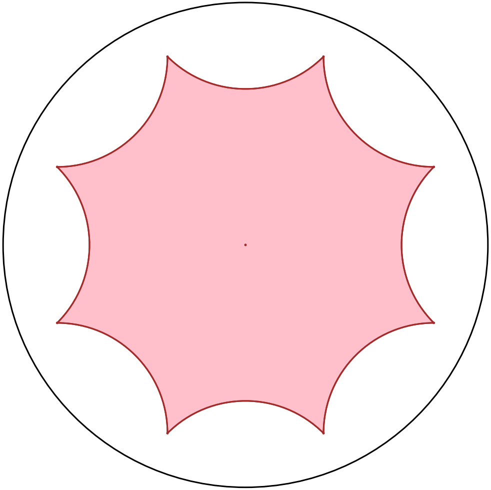
Tilings of the Euclidean and hyperbolic planes



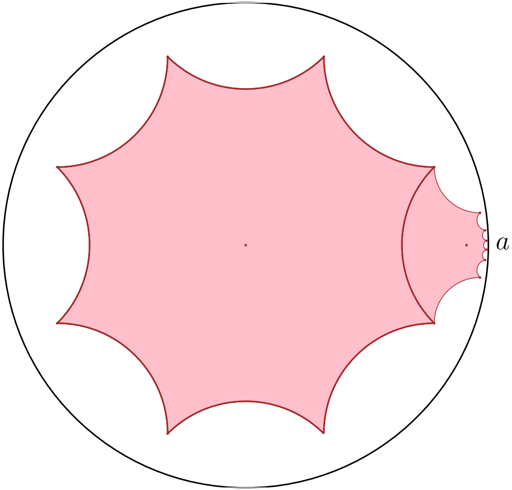
Tiling of the hyperbolic plane with octagons



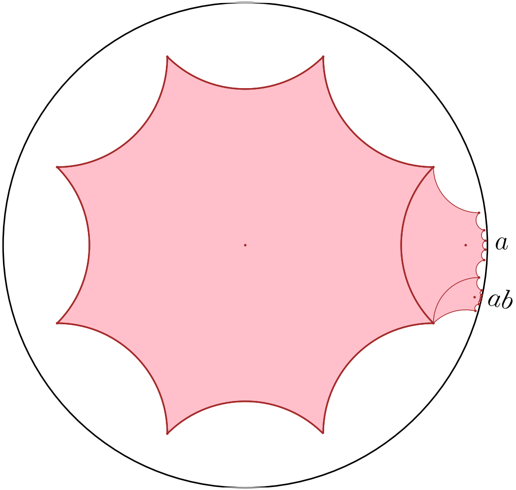
Tiling of the hyperbolic plane with octagons



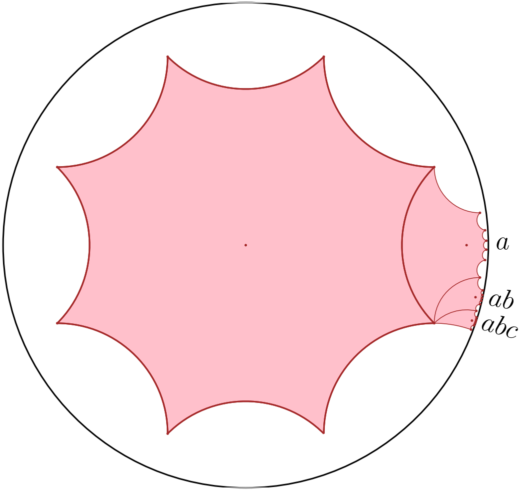
Tiling of the hyperbolic plane with octagons



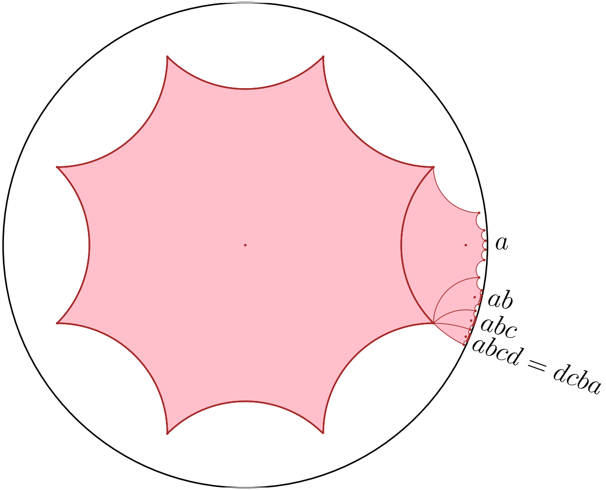
Tiling of the hyperbolic plane with octagons



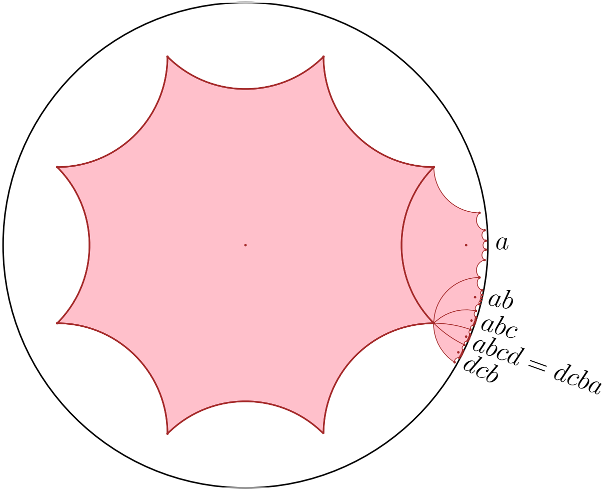
Tiling of the hyperbolic plane with octagons



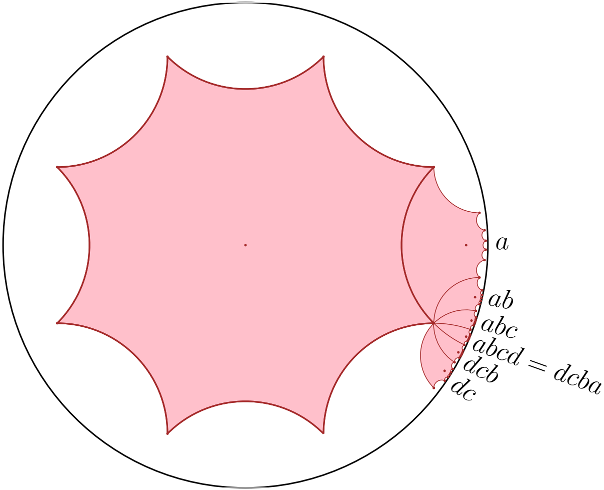
Tiling of the hyperbolic plane with octagons



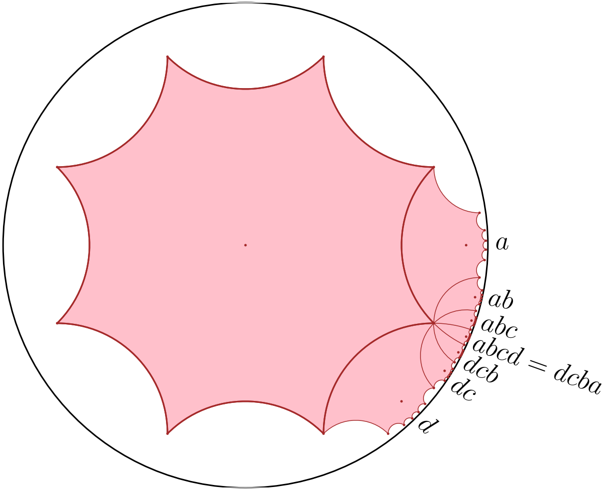
Tiling of the hyperbolic plane with octagons



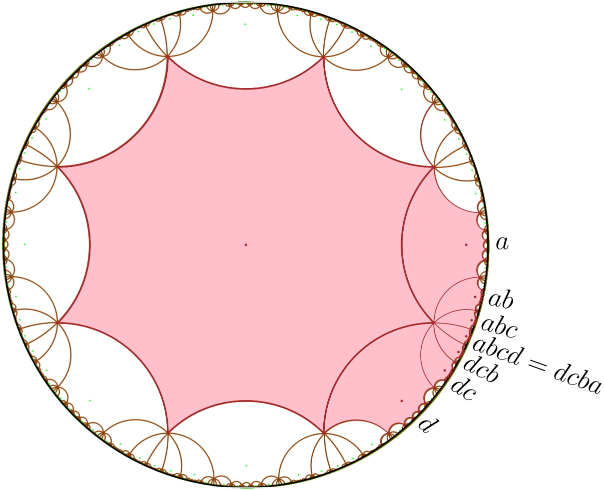
Tiling of the hyperbolic plane with octagons



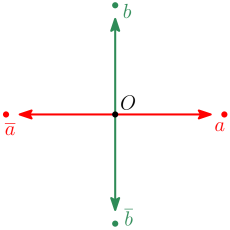
Tiling of the hyperbolic plane with octagons



Tiling of the hyperbolic plane with octagons



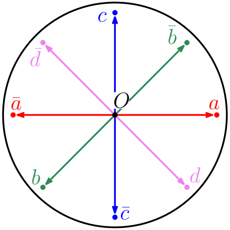
The square flat torus and the Bolza surface



Euclidean: translation group

$$\Gamma_1 = \left\langle a, b \mid ab\bar{a}\bar{b} = \mathbb{1} \right\rangle$$

Flat torus: $\mathbb{M}_1 = \mathbb{E}^2/\Gamma_1$
with projection map $\pi_1 : \mathbb{E}^2 \to \mathbb{M}_1$

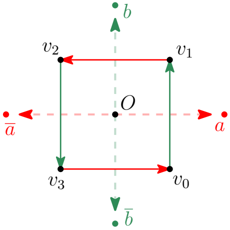


Hyperbolic: Fuchsian group

$$\Gamma_2 = \left\langle a, b, c, d \mid abcd\bar{a}\bar{b}\bar{c}\bar{d} = \mathbb{1} \right\rangle$$

Bolza surface: $\mathbb{M}_2 = \mathbb{H}^2/\Gamma_2$
with projection map $\pi_2 : \mathbb{H}^2 \to \mathbb{M}_2$

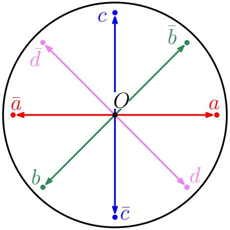
The square flat torus and the Bolza surface



Euclidean: translation group

$$\Gamma_1 = \left\langle a, b \mid ab\bar{a}\bar{b} = \mathbb{1} \right\rangle$$

Flat torus: $\mathbb{M}_1 = \mathbb{E}^2/\Gamma_1$
with projection map $\pi_1 : \mathbb{E}^2 \to \mathbb{M}_1$

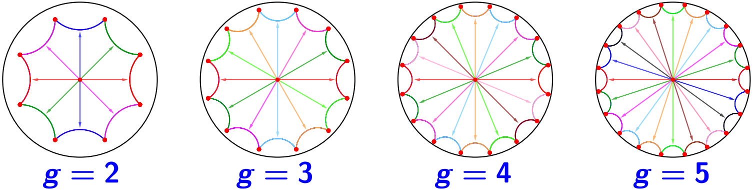


Hyperbolic: Fuchsian group

$$\Gamma_2 = \left\langle a, b, c, d \mid abcd\bar{a}\bar{b}\bar{c}\bar{d} = \mathbb{1} \right\rangle$$

Bolza surface: $\mathbb{M}_2 = \mathbb{H}^2/\Gamma_2$
with projection map $\pi_2 : \mathbb{H}^2 \to \mathbb{M}_2$

Symmetric hyperbolic surfaces of genus $g \geq 2$



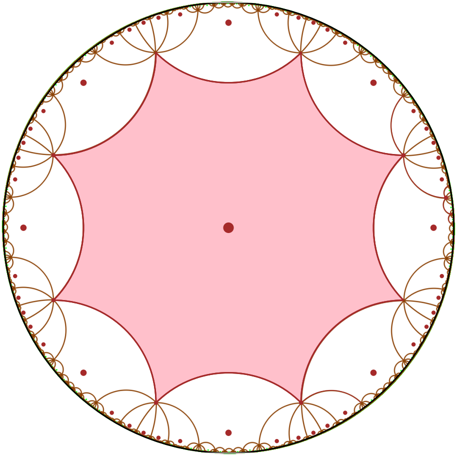
angle sum $= 2\pi$ for all $4g$-gons!

Let Γ_g: Fuchsian group with finite presentation similar to Bolza
$\rightarrow$ $2g$ generators, single relation

Symmetric hyperbolic surface: $\mathbb{M}_g = \mathbb{H}^2 / \Gamma_g$, $\quad g \geq 2$

with natural projection mapping $\pi_g : \mathbb{H}^2 \rightarrow \mathbb{M}_g$

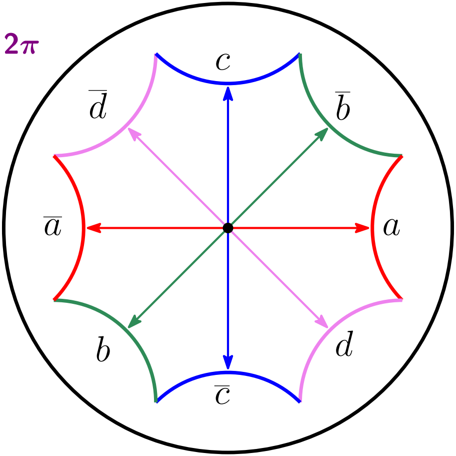
Dirichlet regions



Voronoi diagram of $\Gamma_g O$ for $g = 2$

Dirichlet regions

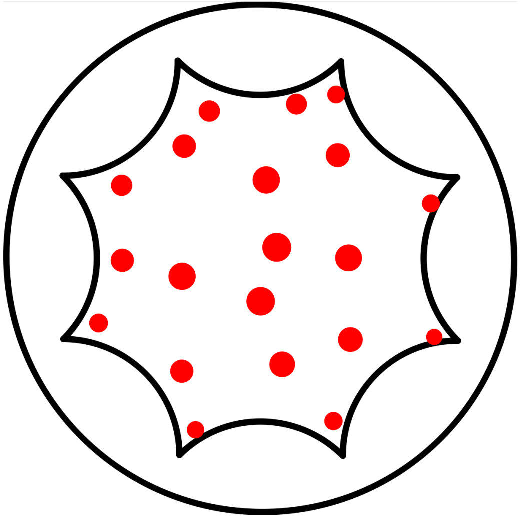
angle sum $= 2\pi$



Fundamental domain D_g = Dirichlet region of O for Γ_g

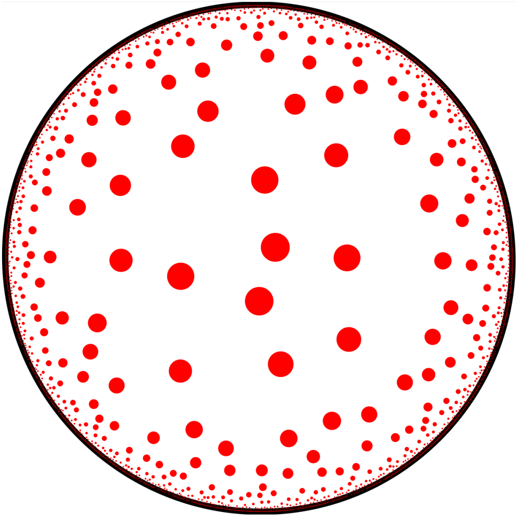
here for $g = 2$

Delaunay triangulation



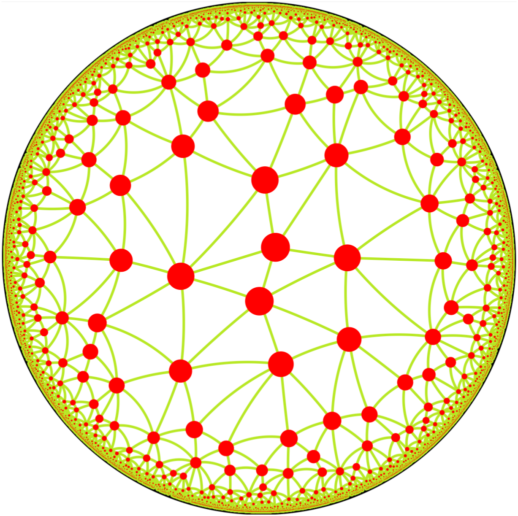
S set of points in D_g

Delaunay triangulation



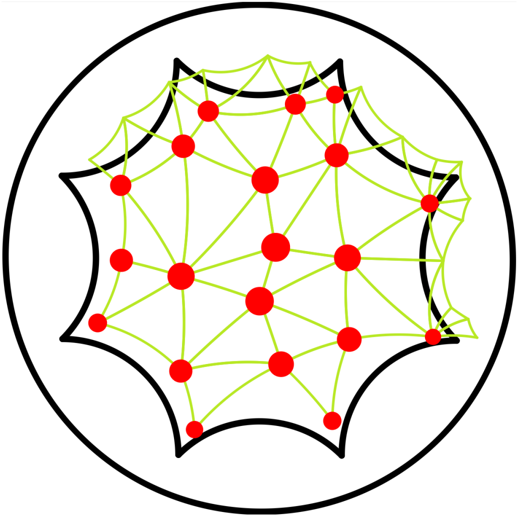
orbits $\Gamma_g S$ in $\mathbb{H}^2$

Delaunay triangulation



Delaunay triangulation in $\mathbb{H}^2$
$DT_{\mathbb{H}}(\Gamma_g S)$

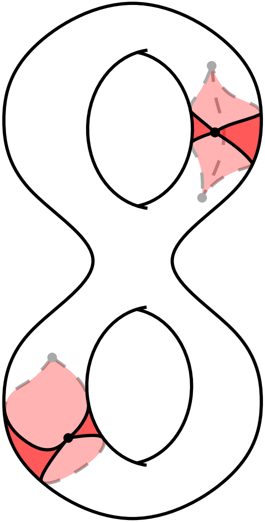
Delaunay triangulation



Delaunay triangulation of $\mathbb{M}_g$

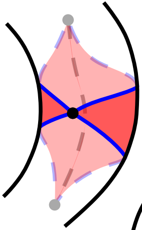
$DT_{\mathbb{M}_g}(S)$

Delaunay triangulation

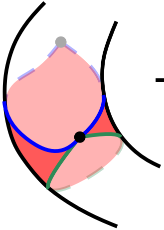


projection of $DT_{\mathbb{H}}(\Gamma_g S)$ on the surface $\mathbb{M}_g$

→ not necessarily a simplicial complex!



→ **double edges**

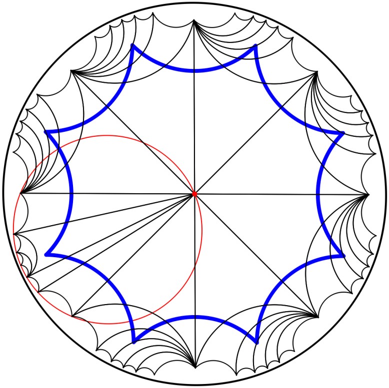


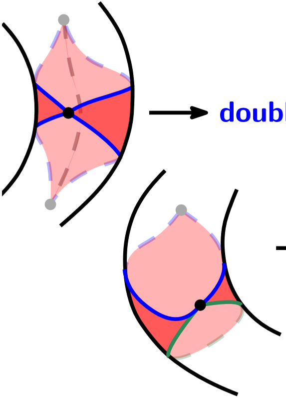
→ **double edges and/or loops**

Delaunay triangulation

projection of $DT_{\mathbb{H}}(\Gamma_g S)$ on the surface $\mathbb{M}_g$

→ not necessarily a simplicial complex!



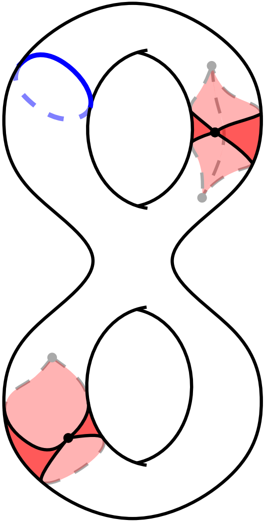


→ **double edges**

→ **double edges and/or loops**

Validity condition

[BTV16]



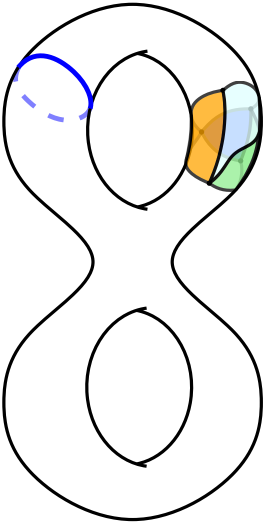
projection of $DT_{\mathbb{H}}(\Gamma_g S)$ on the surface $\mathbb{M}_g$

$\rightarrow$ not necessarily a simplicial complex!

Systole of a surface = minimum length of a non-contractible loop on the surface

Validity condition

[BTV16]



projection of $DT_{\mathbb{H}}(\Gamma_g S)$ on the surface $\mathbb{M}_g$

$\rightarrow$ **is a simplicial complex, if**

$$\delta_S < \frac{1}{2}\text{sys}(\mathbb{M}_g), \quad \text{where}$$

δ_S = diameter of largest disks in $\mathbb{H}^2$ not containing any point of $\Gamma_g S$

$$\boldsymbol{DT_{\mathbb{M}_g}(S) := \pi_g(DT_{\mathbb{H}}(\Gamma_g S))}$$

Computing Delaunay triangulations of $\mathbb{M}_g$

Use set of *dummy points Q_g* that satisfies the validity condition:

$$S := Q_g \bigcup P \Longrightarrow \delta_S < \frac{1}{2}\mathrm{sys}(\mathbb{M}_g) \quad \text{always}$$

Computing Delaunay triangulations of $\mathbb{M}_g$

Use set of *dummy points* Q_g that satisfies the validity condition:

$$S := Q_g \bigcup P \Longrightarrow \delta_S < \frac{1}{2}\text{sys}(\mathbb{M}_g) \quad \text{always}$$

Algorithm for Delaunay triangulations of $\mathbb{M}_g$ [BTV16]

1. initialize $DT_{\mathbb{M}_g}$ with a set of dummy points Q_g
2. insert input points P in the triangulation
3. remove points of Q_g from the triangulation, if possible

→ condition preserved with insertion of new points
→ final triangulation might contain dummy points

Problem statement

To compute $DT_{\mathbb{M}_g}(S)$, we need to *choose* **what** to store.

Requirement: all input points lie in D_g
$\rightarrow$ unique representative in $D_g \subset \mathbb{H}^2$ for each point on $\mathbb{M}_g$

Question: How to choose a unique representative for each *face*?

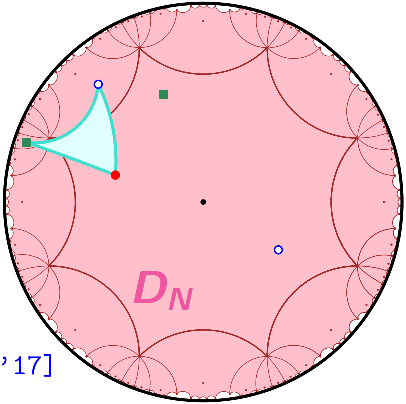
Inclusion property

Let $S \subset D_g$ be a point set such that

$$\delta_S < \frac{1}{2}\,\text{sys}(\mathbb{M}_g).$$

Let σ be a face of $DT_{\mathbb{H}}(\Gamma_g S)$ with at least one vertex in D_g

$\Rightarrow$ σ is contained in D_N

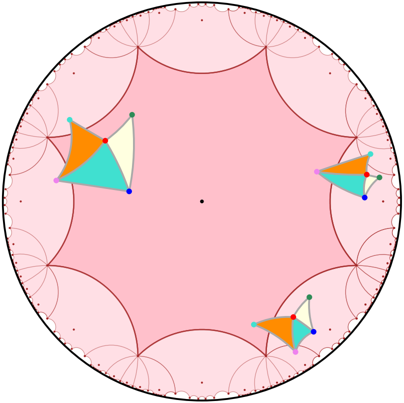


Proof:

- for $g = 2$ $\rightarrow$ [IT, SoCG '17]
- for $g \geq 2$ $\rightarrow$ [EITV]

Canonical representatives of faces

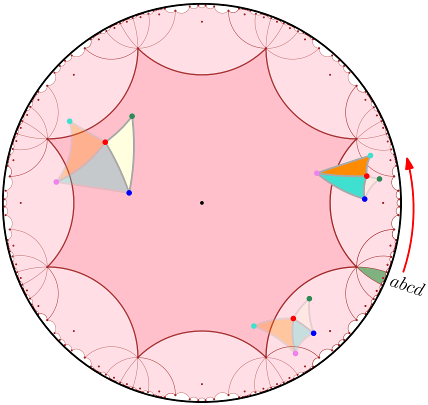
Canonical representative: face with at least one vertex in D_g
$\rightarrow$ other vertices will be in D_N



Canonical representatives of faces

Canonical representative: face with at least one vertex in D_g
$\rightarrow$ other vertices will be in D_N

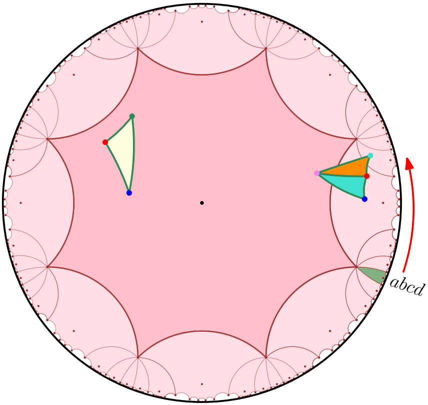
To make it unique:
$\rightarrow$ choose the face "closest" to the first Dirichlet neighbor



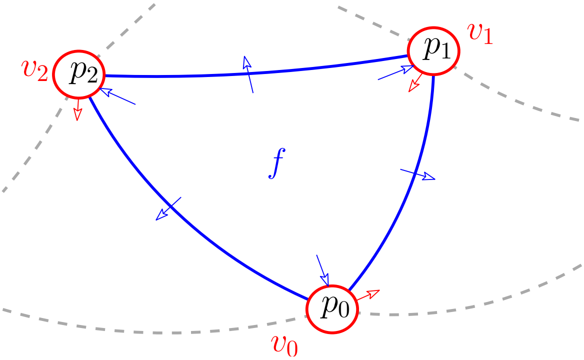
Canonical representatives of faces

Canonical representative: face with at least one vertex in D_g
$\rightarrow$ other vertices will be in D_N

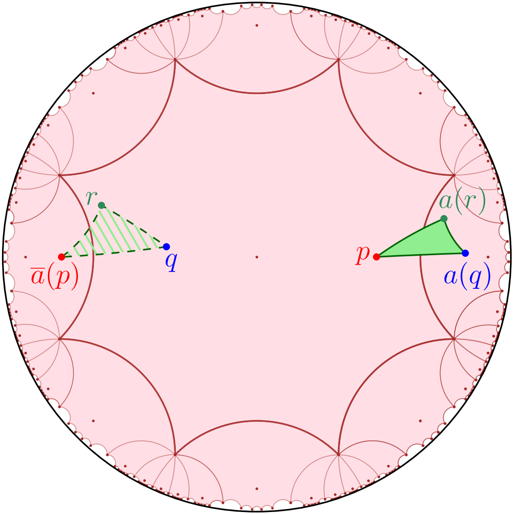
To make it unique:
$\rightarrow$ choose the face "closest" to the first Dirichlet neighbor



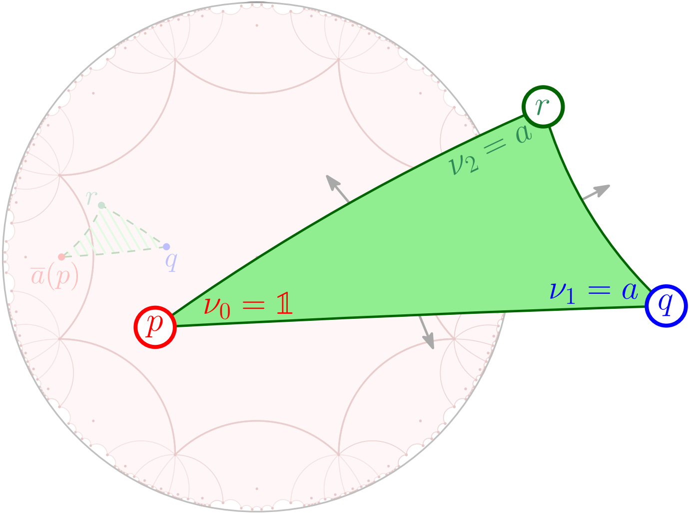
CGAL triangulation data structure



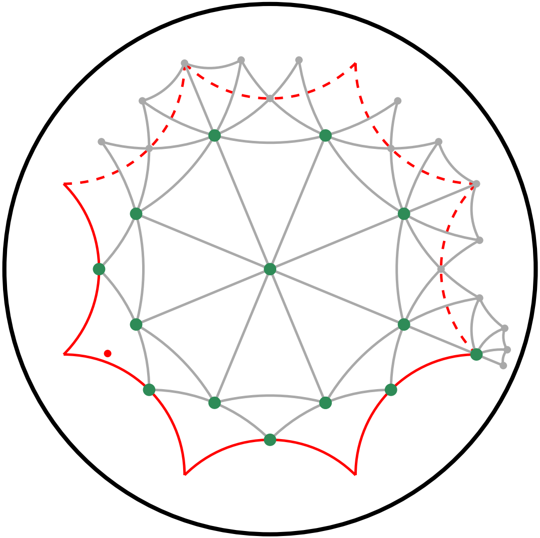
Canonical representatives can cross the boundary



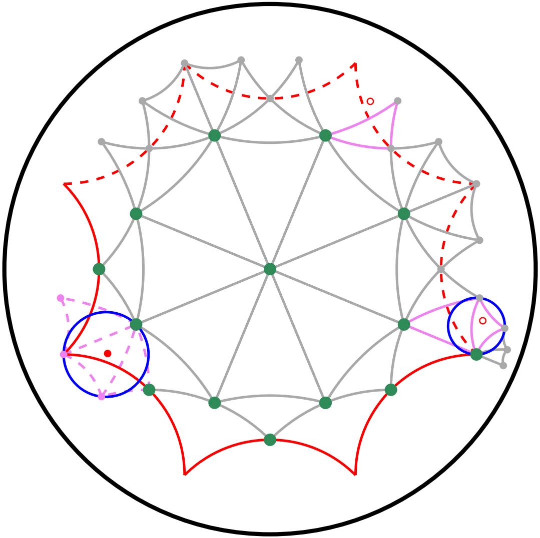
CGAL extended triangulation data structure



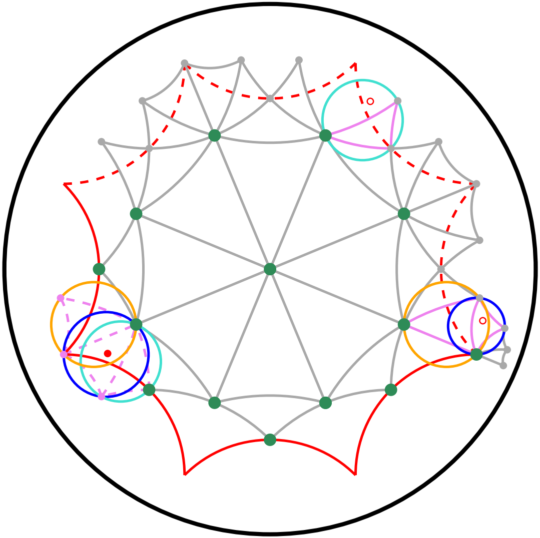
Point Location



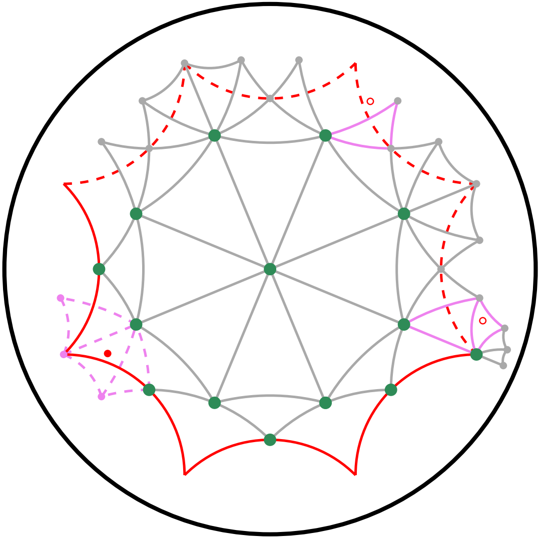
Point Location



Point Location

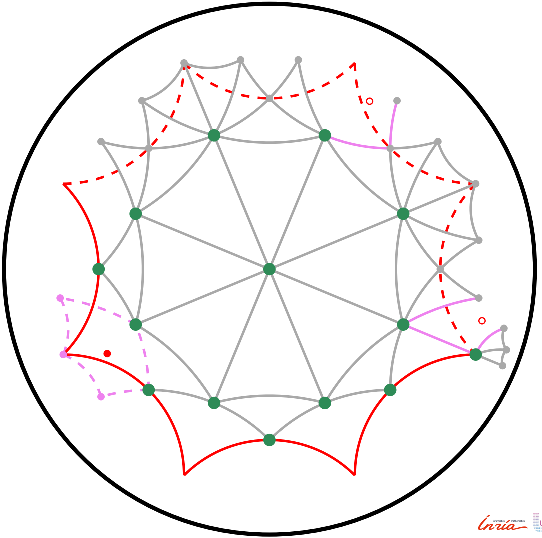


Point Location



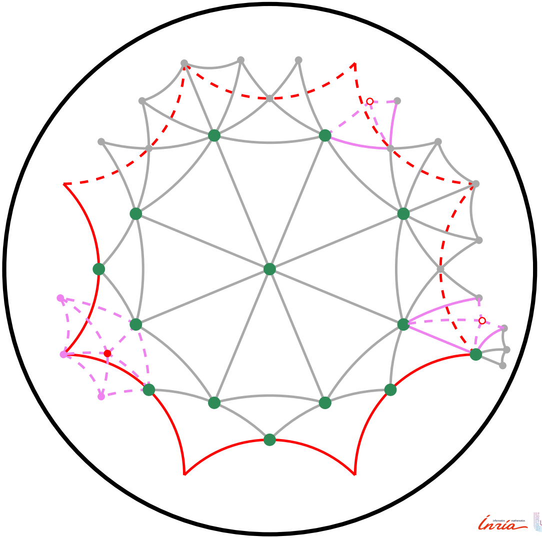
Point Insertion

recall:
"hole" = topological disk



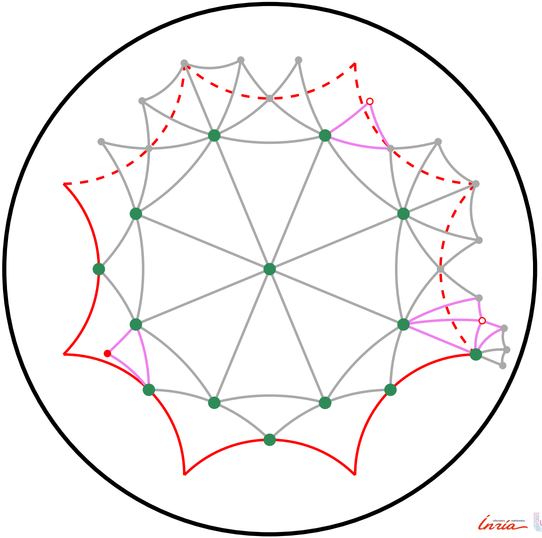
Point Insertion

recall:
“hole” = topological disk



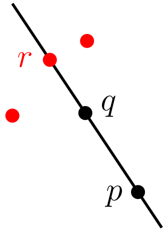
Point Insertion

Computations
on translations

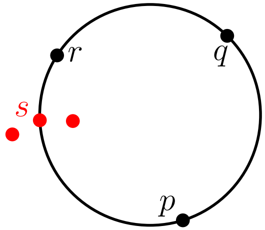


Predicates

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



$$InCircle(p, q, r, s) = \text{sign} \begin{vmatrix} p_x & p_y & p_x^2 + p_y^2 & 1 \\ q_x & q_y & q_x^2 + q_y^2 & 1 \\ r_x & r_y & r_x^2 + r_y^2 & 1 \\ s_x & s_y & s_x^2 + s_y^2 & 1 \end{vmatrix}$$



Predicates

Suppose that the points in S are rational.

Input of the predicates can be their images under translations, e.g.,

$$\overline{b} : z \mapsto \frac{z \cdot \left(1+\sqrt{2}\right) + e^{\frac{i\pi}{4}}\sqrt{2}\sqrt{1+\sqrt{2}}}{z \cdot e^{-\frac{i\pi}{4}}\sqrt{2}\sqrt{1+\sqrt{2}} + \left(1+\sqrt{2}\right)}.$$

Orientation:

Degree	16	20
# cases	28	42

InCircle:

Degree	32	40	48	56	64	72
# cases	13	57	140	21	6	1

Point coordinates represented with `CORE::Expr`
→ (filtered) exact evaluation of predicates

Experiments

Fully dynamic implementation

1 million rational random points

- CGAL Euclidean DT (double) $\sim$ 1 sec.
- CGAL Euclidean DT (CORE::Expr) $\sim$ 22 sec.
- Hyperbolic periodic DT (double) $\sim$ 13 sec.
- Hyperbolic periodic DT (CORE::Expr) $\sim$ 48 sec.

Experiments

Fully dynamic implementation

1 million rational random points

- CGAL Euclidean DT (double) ~ 1 sec.
- CGAL Euclidean DT (CORE::Expr) ~ 22 sec.
- Hyperbolic periodic DT (double) ~ 13 sec.
- Hyperbolic periodic DT (CORE::Expr) ~ 48 sec.

Predicates

- 0.76% calls to predicates involve non-identity translations
- responsible for 36% of total time spent in predicates

No dummy points left after insertion of > 200 random points.

Demo

- Implementation (open source) is available on Github:
  `https://imiordanov.github.io/code/`
  To appear in `CGAL v.4.14` (March 2019)
- YouTube video of CGAL demo shows hyperbolic free motion:
  `https://tinyurl.com/bolza-free-motion`
- We will see the live demo right now!

An initial set of dummy points [EITV]

For $\mathbb{M}_2$, a set of dummy points was given [BTV16]. **In general?**

An initial set of dummy points [EITV]

For $\mathbb{M}_2$, a set of dummy points was given [BTV16]. **In general?**

The idea is to **generate** dummy points:

1. Start with a set W_g for $\mathbb{M}_g$ (called *Weierstrass points*)
   $\rightarrow$ origin, one vertex, and midpoints of sides of the $4g$-gon
2. Compute the images of these points in D_N
3. Compute their hyperbolic Delaunay triangulation in $\mathbb{H}^2$
4. Apply Delaunay refinements to satisfy condition $\leftarrow$ strategies!

An initial set of dummy points [EITV]

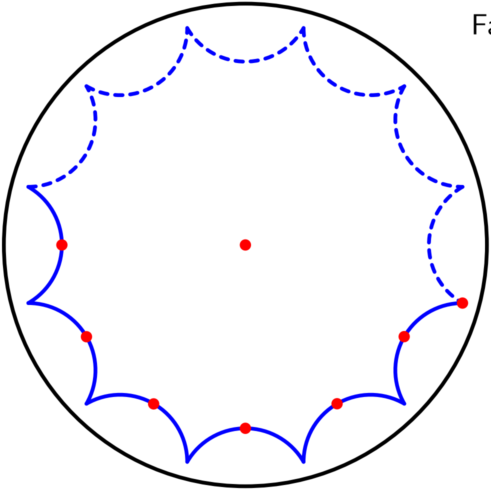
For $\mathbb{M}_2$, a set of dummy points was given [BTV16]. **In general?**

The idea is to **generate** dummy points:

1. Start with a set W_g for $\mathbb{M}_g$ (called *Weierstrass points*)
   $\rightarrow$ origin, one vertex, and midpoints of sides of the $4g$-gon
2. Compute the images of these points in D_N
3. Compute their hyperbolic Delaunay triangulation in $\mathbb{H}^2$
4. Apply Delaunay refinements to satisfy condition $\leftarrow$ strategies!

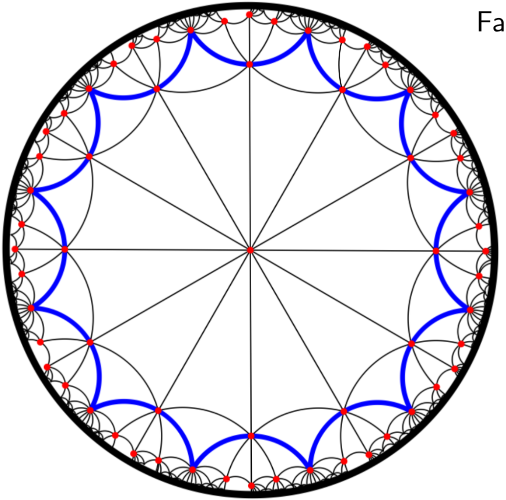
- $\mathrm{sys}(\mathbb{M}_g) = 2\,\mathrm{arcosh}\left(1 + 2\cos\left(\frac{\pi}{2g}\right)\right)$ [Ebbens, 2018]
- triangulation of sets including W_g: contained in D_N

Triangulations of Weierstrass points



Faces with a vertex in the polygon
$\rightarrow$ contained in $\boldsymbol{D_N}$

Triangulations of Weierstrass points

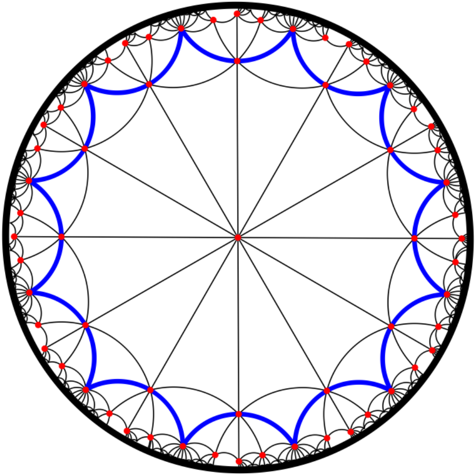


Faces with a vertex in the polygon
$\rightarrow$ contained in $\boldsymbol{D_N}$

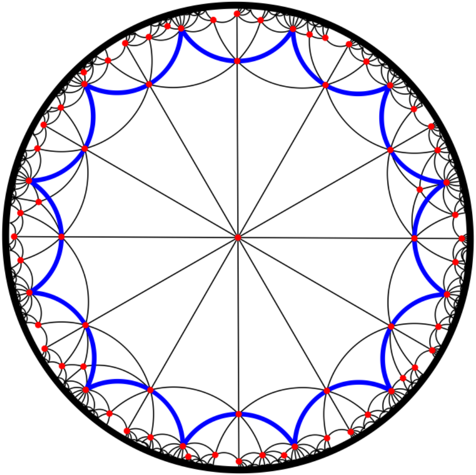
Compute dummy points:

1. Get triangulation in $\boldsymbol{D_N}$
2. Refine triangulation
3. Take points in $4g$-gon

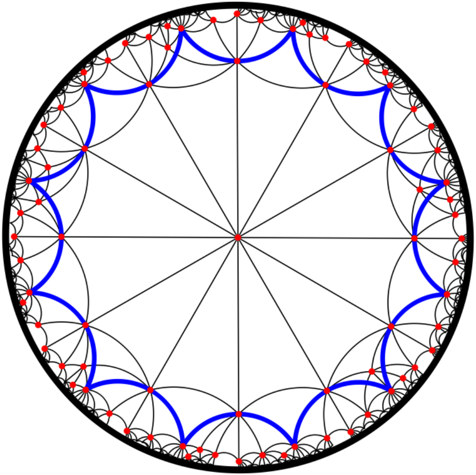
Sequential strategy



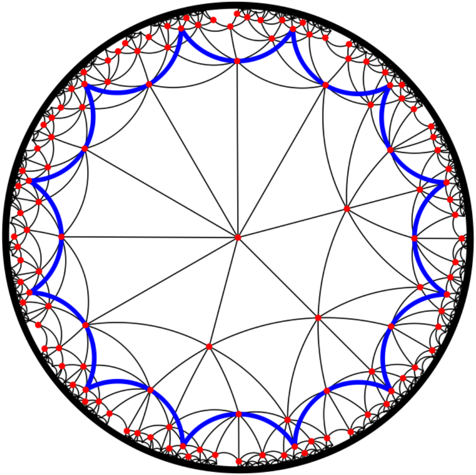
Sequential strategy



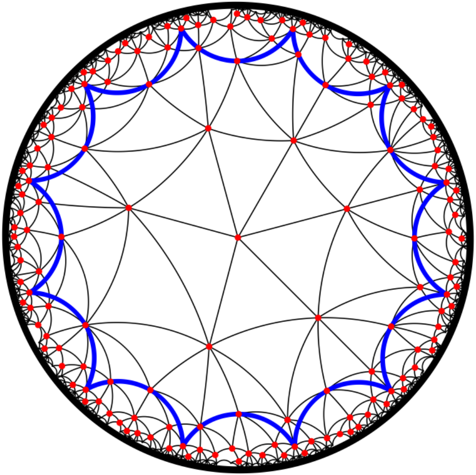
Sequential strategy



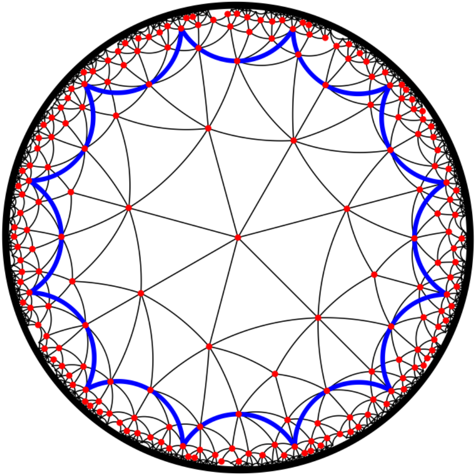
Sequential strategy



Sequential strategy

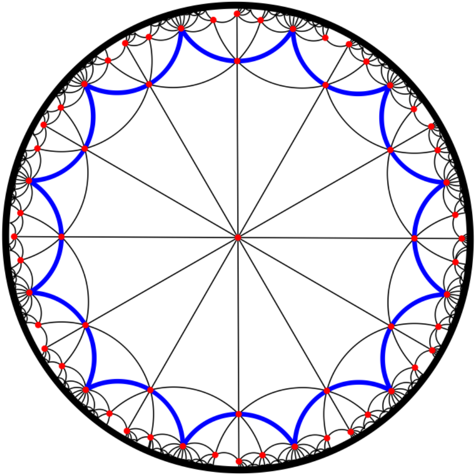


Sequential strategy

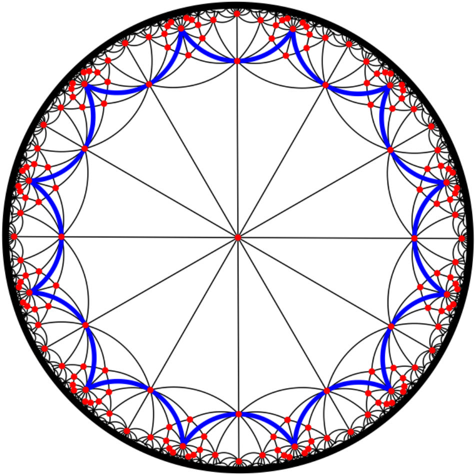


set size $\mathcal{O}(g)$
[EITV]

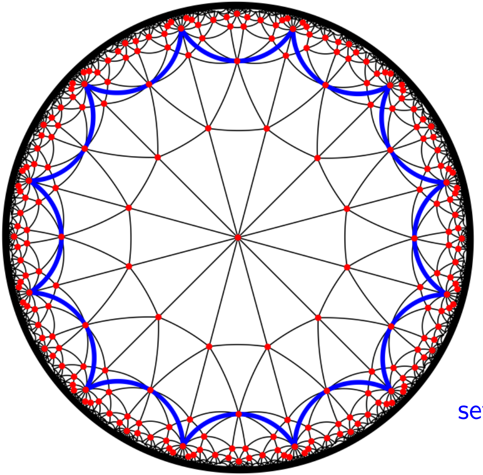
Sequential strategy with symmetries



Sequential strategy with symmetries



Sequential strategy with symmetries



set size $\mathcal{O}(g \log g)$
[EITV]

Implementation

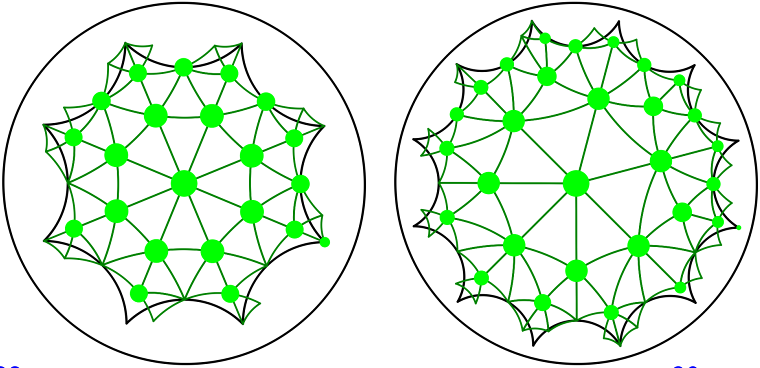
- Preliminary code on Github, but **not** public
- What is implemented:
  - generation of dummy points (first two strategies)
  - initialization of periodic triangulation
  - location, insertion, removal: as for Bolza

Implementation

- Preliminary code on Github, but **not** public
- What is implemented:
  - generation of dummy points (first two strategies)
  - initialization of periodic triangulation
  - location, insertion, removal: as for Bolza
- Problems in practice
  - Recall: exact predicates; now with more complex expressions!
  - Comparison of two numbers: non-conclusive! (even for $g = 3$)
  - Idea: use limited accuracy, validate *a posteriori*
    - obtained preliminary results for $g = 3$
    - up to $2048 \times g$ bits: `CORE` crashes for $g > 3$ (generating Q_g)

Experimental results

sequential

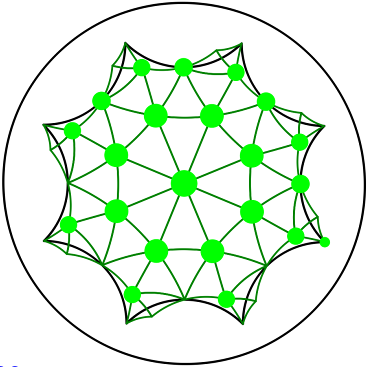


22 pts.

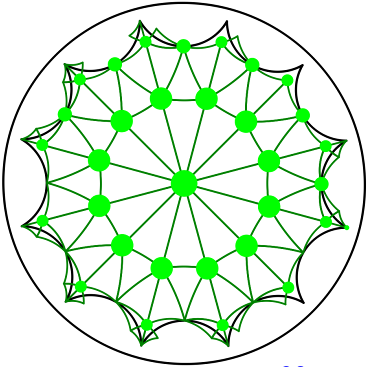
30 pts.

Experimental results

sequential w/ symmetries

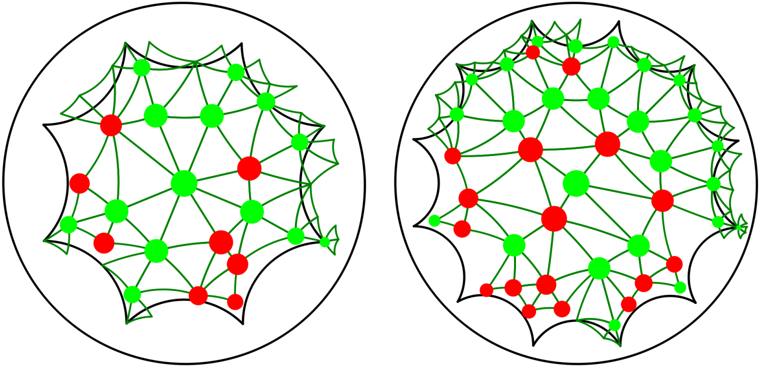


22 pts.

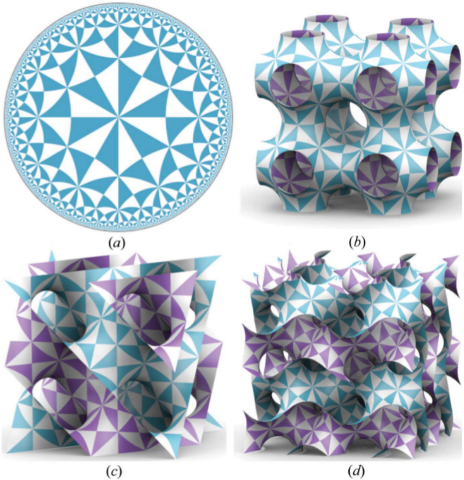


32 pts.

Experimental results insertion/removal

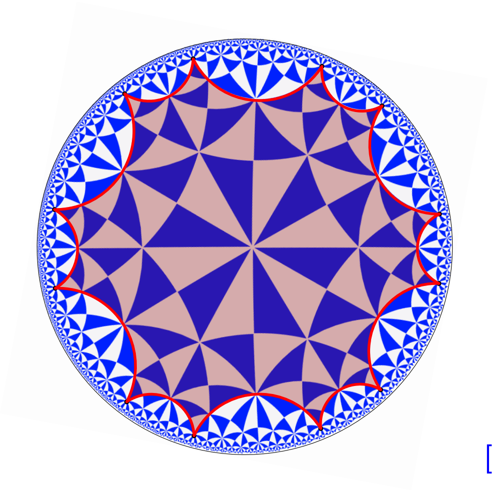


Triply Periodic Minimal Surfaces (TPMS)



[Evans *et al.*, 2013]

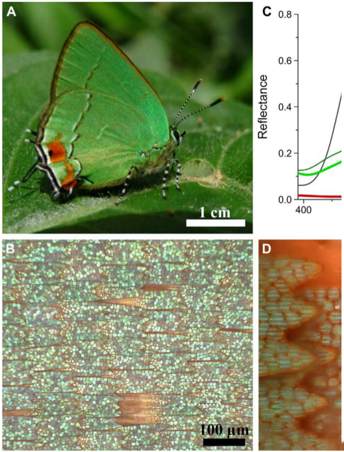
Triply Periodic Minimal Surfaces (TPMS)

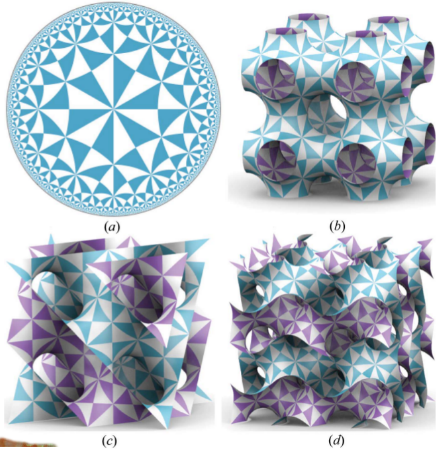


[Evans *et al.*, 2013]

Example of hyperbolic surface: gyroid

[Evans *et al.*, 2013]





[Schröder-Turk *et al.*, 2017]