Experimental analysis of Delaunay flip algorithms on genus two hyperbolic surfaces

Vincent Despré, Loïc Dubois, Benedikt Kolbe, Monique Teillaud

Experimental analysis of Delaunay flip algorithms on genus two hyperbolic surfaces

Vincent Despré[1], Loïc Dubois[2], Benedikt Kolbe[3], and Monique Teillaud[4]

[1]Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
vincent.despre@loria.fr
[2]LIGM, CNRS, Université Gustave Eiffel, F-77454 Marne-la-Vallée, France *
loic.dubois@ens-lyon.fr
[3]Hausdorff Center for Mathematics, University of Bonn, Germany†
benedikt.kolbe@physik.hu-berlin.de
`https://hyperbolictilings.wordpress.com/`
[4]Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
monique.teillaud@inria.fr
`https://members.loria.fr/Monique.Teillaud`

May 11, 2022

Abstract

Guided by insights on the mapping class group of a surface, we give experimental evidence that the upper bound recently proven on the diameter of the flip graph of a closed oriented hyperbolic surface by Despré, Schlenker, and Teillaud (SoCG'20) is largely overestimated. To this aim, we develop an experimental framework for the storage of triangulations. We show that the computations with algebraic numbers can be overcome by proving a density result on rationally described genus two surfaces, and we propose ways to generate surfaces that are meaningful for the experiments.

Acknowledgements The authors want to thank Vincent Delecroix, Matthijs Ebbens, Hugo Parlier, Jean-Marc Schlenker, and Gert Vegter for helpful discussions over many years.

Source code available at `https://members.loria.fr/Monique.Teillaud/Exp-hyperb-flips/`.

Funding This work was partially supported by grant ANR-17-CE40-0033 of the French National Research Agency ANR (project SoS) `https://SoS.loria.fr/`.

*This work was done while this author was working at Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy

†This work was done while this author was working at Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy

Contents

1 Introduction

It was recently proven that the geometric flip graph of a closed oriented hyperbolic surface is connected [7]. A Delaunay flip algorithm can thus transform any input geometric triangulation T, i.e., a triangulation whose edges are embedded as geodesic segments only intersecting at common endpoints, into a Delaunay triangulation. This is particularly useful in practice as a crucial preprocessing step to computing Delaunay triangulations on a surface: it transforms a "bad" representation of a surface, e.g., by a very elongated fundamental domain, to a "nice" representation by a Delaunay triangulation with only one vertex.

An upper-bound on the number of flips was proven [7, Theorem 19]: $C_h \cdot \Delta(T)^{6g-4} \cdot n^2$, where C_h is a constant, $\Delta(T)$ is the diameter of T, g is the genus of the surface, and n is the number of vertices. The diameter $\Delta(T)$ is the smallest diameter of a fundamental domain that is the union of lifts of the triangles of T in $\mathbb{H}$ (note that this is not the diameter of the surface, which is independent of the representation). Computing it algorithmically seems difficult, however for a triangulation with only one vertex (thus with $4g-2$ triangles) some bounds are easily derived: $L_T \leq \Delta(T) \leq \Delta(F) \leq L_T.(4g-2)$, where L_T denotes the maximal length of an edge of T and $F \subset \mathbb{H}$ is any fundamental domain made of lifts of the triangles of T. From these bounds we see that $\Delta(T)$ cannot differ too much from the diameter of any such F: in the case of a genus two surface they only differ by a factor of at most six. In the experiments, we will thus use the domain that naturally appears.

In this paper, we experimentally study the dependence of the number of flips on $\Delta(T)$ (Section 7), for surfaces of genus two. We suspect that the factor $\Delta(T)^{6g-4}$ is largely overestimated. It is derived from the number of paths of bounded length on a surface. Intuitively, for a length bounded by L, it roughly amounts to the volume of the ball of diameter L, so, it is exponential in L; if only simple paths are considered, this number reduces to L^{6g-4} [7], but there is no reason why the flip algorithm would use all the simple paths shorter than L instead of going straight. More formally, our expectation on the dependence in $\Delta(T)$ is based on insights on the structure of the mapping class group (Section 2.3).

To perform experiments, we set up a framework consisting of various tools. In Section 3, we present a data structure for triangulations of surfaces, which supports flips; it relies on the representation of genus two surfaces by octagons in $\mathbb{H}$ (Section 2.4). Not surprisingly, arithmetic issues quickly arise, as algebraic numbers are involved in the description of the octagons (Section 4.1). We overcome them by proving a density result on rationally described octagons (Section 4.2), which allows us to restrict to rational numbers in our experiments.

The generation of input surfaces and triangulations is far from trivial; it is a non-negligible part of our work (Section 5). We obtain surfaces with a large diameter by twisting the above-mentioned octagons (Section 5.1).

In Section 6, we run experiments comparing strategies on the sequence of edge flips, and conclude that the naive strategy is close to being the best one in practice. We adhere to it for our main experiments that study the dependence of the number of flips on $\Delta(T)$.

The way we conduct these experiments in Section 7 is inspired by previous work by Mark Bell [2] who studied flips in a topological setting. We focus on triangulations having only one vertex, both because the dependence on the number of vertices is easily seen, and because inserting a lot of points would rather be done by Bowyer's incremental algorithm [13, 6], inspired from previous work in the flat case [17]. Quite surprisingly, in practice, we observe a behavior that is only expected asymptotically.

2 Background

2.1 Hyperbolic surfaces

All the surfaces considered in this paper are closed (connected, compact and without boundary), oriented, and hyperbolic. Consider such a hyperbolic surface $\mathcal{S}$ of genus $g \geq 2$ and the underlying topological surface S_g. Given a hyperbolic structure h on $\mathcal{S}$, associated to a metric of constant curvature -1, the surface $\mathcal{S} = (S_g, h)$ is isometric to the quotient $\mathbb{H}/G$, where $\mathbb{H}$ is the hyperbolic plane and G is a (non-Abelian) discrete subgroup of the isometry group of $\mathbb{H}$ isomorphic to the fundamental group $\pi_1(S_g)$.

The universal cover of $\mathcal{S}$ is isometric to $\mathbb{H}$ equipped with a projection $\rho : \mathbb{H} \to \mathcal{S}$ that is a local isometry. The group G acts on $\mathbb{H}$, so that for any $p \in \mathcal{S}$, $\rho^{-1}(p)$ is an orbit under the action of G. A lift $\widetilde{p}$ of a point $p \in \mathcal{S}$ is one of the elements of the orbit $\rho^{-1}(p)$.

We use the Poincaré disk model of $\mathbb{H}$, in which $\mathbb{H}$ is represented as the open unit disk $\mathbb{D}$ of the complex plane $\mathbb{C}$. Every orientation preserving isometry $f : \mathbb{D} \to \mathbb{D}$ can be represented by a matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathbb{C}^{2\times 2}$ such that $f(z) = \dfrac{az+b}{cz+d}$ for any $z \in \mathbb{D}$. Observe that the matrix is not unique. In addition some matrices do not represent an isometry. Given two orientation preserving isometries f and g respectively represented by matrices A and B, the product $A \cdot B$ represents $f \circ g$.

2.2 Triangulations and flips on hyperbolic surfaces

A topological triangulation of a hyperbolic surface $\mathcal{S}$ is any embedding of an undirected graph with a finite number of vertices onto $\mathcal{S}$ such that each resulting face is homeomorphic to an open disk and is bounded by exactly three distinct edge-embeddings. Observe that this graph may have loops or multiple edges, and recall that the terms *embedding* and *embedded* subsume that edges only intersect at common vertices. A *geometric triangulation* is a topological triangulation of $\mathcal{S}$ whose edges are embedded as geodesic segments [7]. All triangulations considered in this paper are geometric, so we just use the term *triangulation*. For any triangulation T of $\mathcal{S}$, the lift $\widetilde{T}$ of T is the (infinite) triangulation of $\mathbb{H}$ whose vertices and edges are the lifts of the vertices and the edges of T. A *Delaunay triangulation* T of $\mathcal{S}$ is a triangulation whose lift $\widetilde{T}$ is a Delaunay triangulation in $\mathbb{H}$. In other words, for each face t of T and any of its lifts $\widetilde{t}$, the open disk in $\mathbb{H}$ circumscribing $\widetilde{t}$ does not contain any vertex of $\widetilde{T}$. Recall that circles in the Poincaré disk model correspond to circles in the complex plane.

Lifting an edge e of T to some $\widetilde{e}$, together with the two triangles incident to $\widetilde{e}$ in the lifted triangulation $\widetilde{T}$, we say that e is *Delaunay-flippable* if the open disks of these two triangles contain the fourth vertex of the quadrilateral formed by the two triangles. In this case, the geodesic segment $\widetilde{e}'$ that is the other diagonal of the quadrilateral is contained in it. The *Delaunay flip* of e in T consists in replacing $\widetilde{e}$ by $\widetilde{e}'$ and projecting it back to $\mathcal{S}$ by ρ.

Every *Delaunay flip algorithm* takes as input a triangulation of $\mathcal{S}$ and flips Delaunay-flippable edges (in any order) until there is none left. Every such algorithm terminates and outputs a Delaunay triangulation [7].

2.3 Mapping class group

We use the same notation as Maher [15] and refer to his paper for details.

The set $\mathrm{Mod}(S_g)$ of all homeomorphims (up to isotopy) of a topological surface S_g is called the *mapping class group* of S_g. Following Thurston's classification [10], $\mathrm{Mod}(S_g)$ contains three kinds of elements: the periodic homeomorphims, which are of finite order and are not useful for our purposes; the reducible ones, which fix at least one curve on S_g; and the so-called *pseudo-Anosov* homeomorphims, also known as the *hyperbolic* elements of $\mathrm{Mod}(S_g)$.

Dehn twists (Figure 1, Left) are typical reducible elements, as they fix all the curves that do not intersect the curve used for twisting. A Dehn twist by a curve c at most adds to the length of a curve a constant that depends on the number of times the curve intersects c. A pseudo-Anosov element at most multiplies the length of a curve by a constant factor.

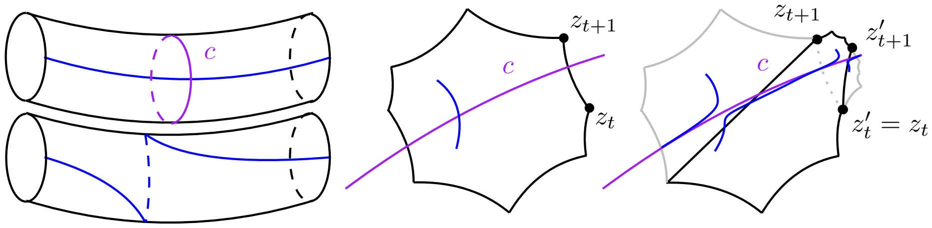


Figure 1: (Left) A Dehn twist along the curve c modifies the blue curve as shown. (Right) A t-twist on an admissible loosely-symmetric octagon.

$\mathrm{Mod}(S_g)$ can be generated by a finite set of Dehn twists [9]. The composition of generators or their inverses in a random order can be interpreted as a random walk in $\mathrm{Mod}(S_g)$: such a walk reaches pseudo-Anosov elements with asymptotic probability 1 [15]. However, this asymptotic result does not *a priori* describe the local structure of $\mathrm{Mod}(S_g)$.

2.4 Admissible symmetric octagons

The Teichmüller space $\mathcal{TM}_2$ of the topological surface S_2 is the set of all the hyperbolic structures (up to isotopy) that can be associated to S_2. It admits various parametrizations. The most commonly used, though not well adapted to our needs, is the set of Fenchel-Nielsen coordinates [12, Section 7.6]. Here, we use a less usual set of parameters introduced by Aigon-Dupuy, Buser *et al.* [1], who proved that any surface of genus 2 has a fundamental domain that is an octagon in $\mathbb{D}$. This versatile representation allows us to easily construct and manipulate such surfaces in our experiments. In this section we recall some definitions and results of the original paper [1], following its notation.

Given $j \geq 3$ and complex numbers $z_1, \ldots, z_j \in \mathbb{D}$ in convex position, $G[z_1, \ldots, z_j]$ denotes the hyperbolic polygon whose vertices are $z_1, \ldots, z_j$ in this order. Given a compact subset $X \subset \mathbb{D}$, $\mathcal{A}(X)$ is the hyperbolic area of X. Given $z \in \mathbb{C}$, we denote by $\mathrm{Re}\,[z]$ and $\mathrm{Im}\,[z]$ its real and imaginary parts, respectively, by $\overline{z}$ its conjugate, and by $|z|$ its modulus; i denotes a root of -1.

Let $\arg z \in [0, 2\pi[$ denote the argument of a point $z \neq 0_{\mathbb{C}}$. Given $z_0, z_1, z_2, z_3 \in \mathbb{D} \setminus \{0_{\mathbb{C}}\}$, the 4-tuple (z_0, z_1, z_2, z_3) is *valid* if $0 = \arg z_0 < \arg z_1 < \arg z_2 < \arg z_3 < \pi$; the hyperbolic octagon $\mathcal{P}[z_0, z_1, z_2, z_3]$ is then defined as $G[z_0, z_1, z_2, z_3, -z_0, -z_1, -z_2, -z_3]$. Such a hyperbolic octagon is called a *symmetric octagon*. The interior angles of a symmetric octagon cannot be greater than π. If moreover $\mathcal{A}(\mathcal{P}[z_0, z_1, z_2, z_3]) = 4\pi$, then $\mathcal{P}[z_0, z_1, z_2, z_3]$ and the 4-tuple (z_0, z_1, z_2, z_3) are called *admissible*. Each surface of genus 2 can be obtained by identifying the opposite sides of an admissible symmetric octagon [1]. Observe that the eight vertices of the octagon correspond to the same point on the surface.

A valid 4-tuple (z_0, z_1, z_2, z_3) is admissible if and only if $\mathrm{Im}\left[\prod_{k=0}^{3} (1 - z_k \overline{z_{k+1}})\right] = 0$ [1, Lemma 3.2]. The authors establish this condition after proving a preliminary result that we will reuse: for any two points $z, z' \in \mathbb{D} \setminus \{0_{\mathbb{C}}\}$ if $0 \leq \arg z \leq \arg z' \leq \pi$ then [1, Appendix (A7)]

$$2 \arg(1 - z\overline{z'}) = \mathcal{A}(G[0_{\mathbb{C}}, z, z']). \tag{1}$$

An admissible 4-tuple can be constructed as follows [1, Section 3]. Start with $z_1, z_2, z_3 \in \mathbb{D}$ satisfying $0 < \arg(z_1) < \arg(z_2) < \arg(z_3) < \pi$. Abbreviate $u = (1 - z_1\overline{z_2})(1 - z_2\overline{z_3})$, $a = \mathrm{Im}\,[-u\overline{z_1}z_3]$, $b = \mathrm{Im}\,[u(z_3 - \overline{z_1})]$, and $c = \mathrm{Im}\,[u]$. Assume $a + b + c < 0$ and let $z_0 = \frac{2c}{-b+\sqrt{b^2-4ac}}$. Then (z_0, z_1, z_2, z_3) is an admissible 4-tuple.

From now on indices are modulo 8. Let us consider an admissible 4-tuple (z_0, z_1, z_2, z_3) and define $z_{l+4} = -z_l$ for every $l \in \{0, 1, 2, 3\}$. For $k \in \{0, \ldots, 7\}$, there exists a unique orientation preserving isometry $\tau_k : \mathbb{D} \to \mathbb{D}$ satisfying $\tau_k(z_{k+5}) = z_k$ and $\tau_k(z_{k+4}) = z_{k+1}$: the isometry τ_k maps a side of $\mathcal{P}[z_0, z_1, z_2, z_3]$ to the opposite side of $\mathcal{P}[z_0, z_1, z_2, z_3]$. Define $\omega_k = \dfrac{z_k(1 - |z_{k+1}|^2) + z_{k+1}(1 - |z_k|^2)}{1 - |z_k z_{k+1}|^2}$ and note that $|\omega_k| < 1$; the isometry τ_k is then given by $\tau_k(z) = (z + \omega_k)/(\overline{\omega_k} + 1)$ for every $z \in \mathbb{D}$ [1, Lemma 4.1]. Observe that $\tau_{k+4} = \tau_k^{-1}$.

3 Representation of triangulations

In this section we describe our data structure for representing triangulations (Section 3.1) and we sketch how it is maintained through flips (Section 3.2).

3.1 Data structure

Although an *ad hoc* data structure was previously proposed for flipping triangulations [7], we choose to use *combinatorial maps*, which are commonly used to represent graphs embedded on a surface. We refer the reader to the literature for a formal definition [16, Section 3.3]. The data structure we use offers a representation of the triangulation that intrinsically lies on the surface, while the earlier data structure [7, Section 4.1] stuck to specific representatives of all vertices and faces of the lifted triangulation in the universal cover. See Appendices A and B.1 for details on this section.

For our experiments, we use the flexible implementation of combinatorial maps that is publicly available in CGAL [5]. The *dart*, also known as *flag*, is the central object in a combinatorial map: it gives access to all incidence relations of an edge of the graph (Figure 2). In our setting a combinatorial map can be thought of as a halfedge data structure.

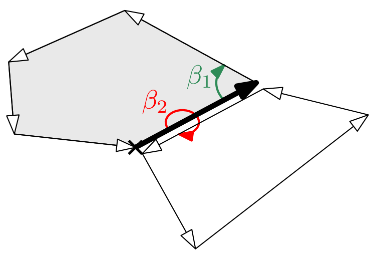


Figure 2: A dart in a combinatorial map (bold).

The geometric information for the triangulation is stored by adding a cross-ratio for each edge. Recall that the cross-ratio of four pairwise-distinct points in $\mathbb{H}$ represented by $z_1, z_2, z_3, z_4 \in \mathbb{D}$ is the complex number $[z_1, z_2, z_3, z_4] = \dfrac{(z_4 - z_2)(z_3 - z_1)}{(z_4 - z_1)(z_3 - z_2)}$ [3]. Cross-ratios are suitable for a flip algorithm, due to their well-known property: assuming that the four points are oriented counterclockwise, $\mathrm{Im}\,[z_1, z_2, z_3, z_4] > 0$ if and only if z_4 lies inside the open disk circumscribing the triangle (z_1, z_2, z_3).

Given an edge e of a triangulation T of $\mathcal{S}$ we consider a lift $\widetilde{e} = (\widetilde{u_1}, \widetilde{u_3})$ of e in $\mathbb{D}$ and the remaining vertices $\widetilde{u_2}$ and $\widetilde{u_4}$ of the two faces incident to $\widetilde{e}$ in $\widetilde{T}$, numbering vertices counterclockwise. The cross-ratio $\mathcal{R}_T(e)$ is defined as $[\widetilde{u_1}, \widetilde{u_2}, \widetilde{u_3}, \widetilde{u_4}]$; it is independent of the choice of the lift of e, as the cross-ratio is invariant under orientation preserving isometries of $\mathbb{D}$. An edge e of T is Delaunay-flippable if and only if $\mathrm{Im}\,[\mathcal{R}_T(e)] > 0$.

Note that in our experiments, the lifts in $\mathbb{D}$ are only used to calculate the cross-ratios of a given input triangulation T; they are ignored during the flips, thus preserving the property that the data structure only considers the embedding of the triangulation on the surface. However, in order to be able to recover a lift in $\mathbb{D}$ in the end, e.g., for drawing a representation in $\mathbb{D}$ of the final Delaunay triangulation, we need to maintain an *anchor* during flips. The anchor $A = (\delta, a_1, a_2, a_3)$ consists in a dart δ, chosen arbitrarily, together with a triple (a_1, a_2, a_3) of the vertices of a lift of the face containing δ.

A triangulation T is thus represented by (M, F, A), where M is the combinatorial map, F the map that associates a cross-ratio to each edge of M, and $A = (\delta, a_1, a_2, a_3)$ is the anchor.

3.2 Flipping an edge

In this section, we quickly sketch how the data structure is maintained through an edge flip. First we modify the combinatorial map, then we update the anchor, and we finally update the cross-ratios. Some details and the pseudo-code are given in Appendix B.2.

Performing a flip in the combinatorial map is a straightforward use of the functionalities given by the CGAL package [5]. The triangulation obtained from T after flipping an edge e is denoted by $T^\star$. By definition, the dart δ of the anchor A belongs to the face t of T represented by a lift $\widetilde{t} = (a_1, a_2, a_3)$ in $\mathbb{D}$. If t does not contain e then A is not modified by the flip. However, if e is an edge of t then t will not belong to $T^\star$ and we must update A. A lift $\widetilde{e}$ of e incident to $\widetilde{t}$ is replaced by $\widetilde{e^\star}$ when e is flipped. The anchor is updated so that it represents one of the two faces incident to $\widetilde{e^\star}$ in $T^\star$.

Finally, the cross-ratios must be retrieved. Only the cross-ratios of the at most 5 edges of the two triangles forming the quadrilateral whose diagonal is to be flipped must be updated. Their values after the flip are expressed in terms of their values before the flip (see Lemma 5 in Appendix A).

4 Solving arithmetic issues

The construction recalled in Section 2.4 shows that the real and imaginary parts of the complex numbers involved when defining surfaces are in general algebraic numbers. Efficiency issues when computing with algebraic numbers have been known for decades. More recently, they appeared when constructing Delaunay triangulations of hyperbolic surfaces [13, 8], showing that the hope to get effective software was restricted to very simple cases. In Section 4.1 we describe a simple experiment on the Bolza surface illustrating that these arithmetic issues are actually prohibitive in practice for the Delaunay flip algorithm in the sense that they imply unreasonable running times.

We show in Section 4.2 that any surface of genus 2 can be approximated by a surface described by rational numbers. It is straightforward to check that the computations made during a Delaunay flip algorithm only use the four basic operations $+, -, \cdot, /$ (see Section 3.2 and Appendix B.2). Thus if the input surface is represented by rational numbers all numbers arising throughout the algorithm stay rational. This fact allows us to run extensive experiments.

4.1 Issues when using algebraic numbers

Let $c_k = \dfrac{\exp\left(i\pi\frac{2k-1}{8}\right)}{2^{1/4}}, k \in \{0, \ldots, 7\}$ be the vertices of a regular hyperbolic octagon in $\mathbb{D}$; identifying the opposite sides of this octagon gives a surface of genus 2 known as the Bolza surface. Consider the triangulation T_0 of the octagon shown in Figure 3. Identifying in T_0 the edges corresponding to opposite sides of the octagon yields a triangulation T of the Bolza surface. Let $e_0, \ldots, e_4$ be the edges of T corresponding to the edges $e'_0, \ldots, e'_4$ of T_0. The algebraic numbers $\cos\left(\dfrac{\pi}{8}\right) = \dfrac{\sqrt{2+\sqrt{2}}}{2}$ and $\sin\left(\dfrac{\pi}{8}\right) = \dfrac{\sqrt{2-\sqrt{2}}}{2}$ naturally appear when computing the cross-ratios $\mathcal{R}_T(e_l) = [c_0, c_{l+1}, c_{l+2}, c_{l+3}]$, $l \in \{0, \ldots, 4\}$.

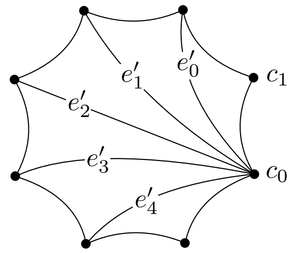


Figure 3: The triangulation T_0 and the edges $e_0', \ldots, e_4'$.

As the points $c_k, k = 0, \ldots, 7$ are concyclic $e_0, \ldots, e_4$ can be flipped although they are strictly speaking not Delaunay-flippable: the situation is degenerate. The experiment consists in computing the new values of the cross-ratios involved during the flips of $e_0, \ldots, e_4$ in this order (see Appendix C.1 for the pseudocode). We used the CGAL wrapper `CORE::Expr` [11] for the algebraic numbers provided by the CORE library [18]. It took minutes to finish on an Intel Core i5-8250u cpu (1.6Ghz, 8 cores) and 16Gb of ram. Such a running time severely restricts the possibility to run heavy experiments with a Delaunay flip algorithm.

4.2 Density of the rationally described surfaces

For any $z \in \mathbb{D}$ and any $\varepsilon > 0$, $B(z, \varepsilon)$ denotes the open ball $\{z' \in \mathbb{D} \,:\, d(z, z') < \varepsilon\}$ where $d(\cdot, \cdot)$ is the hyperbolic distance in $\mathbb{D}$.

Definition 1. We say that a 4-tuple (z_0, z_1, z_2, z_3) is *rational* if $z_k \in \mathbb{Q} + i\mathbb{Q}$ for every $k \in \{0, 1, 2, 3\}$. A *rationally described surface* is a surface obtained from a rational admissible 4-tuple $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$ by identifying the opposite sides of $\mathcal{P}[\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}]$.

Theorem 2. *Let (z_0, z_1, z_2, z_3) be an admissible 4-tuple and $\varepsilon > 0$. There exists a rational admissible 4-tuple $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$ such that $\forall k \in \{0, 1, 2, 3\}, \mathbf{z_k} \in B(z_k, \varepsilon)$.*

Proof. For two reals a and b, define $]a, b[= \{z \in \mathbb{C} \,:\, a < \mathrm{Re}\,[z] < b \text{ and } \mathrm{Im}\,[z] = 0\}$. We first choose for every $k \in \{0, 1, 2, 3\}$ a point $\mathbf{z_k} \in B(z_k, \varepsilon) \cap (\mathbb{Q} + i\mathbb{Q})$, with the additional requirement that $\mathbf{z_0} \in]0, 1[$, but without trying to satisfy the area condition $\mathcal{A}(G[-\mathbf{z_0}, \mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}]) = 2\pi$ (equivalent to the condition $\mathcal{A}(\mathcal{P}[z_0, z_1, z_2, z_3]) = 4\pi$ given in Section 2.4). Consider Figure 4. If ε is small enough, then $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$ is valid. We will now show that if each $\mathbf{z_k}$ is "close enough" to z_k for every k, we can replace $\mathbf{z_3}$ by a point U in $B(z_3, \varepsilon) \cap (\mathbb{Q} + i\mathbb{Q})$ so that the area condition is satisfied. More details on the construction can be found in Appendix C.2.

To do so we first define an isometry $\mathbf{f} : \mathbb{D} \to \mathbb{D}$ in the Poincaré disk: $\mathbf{f}(z) = \dfrac{z + \mathbf{z_0}}{\mathbf{z_0} z + 1}$. Observe that $\mathbf{f}(-\mathbf{z_0}) = 0_{\mathbb{C}}$. Since $\mathbf{f}$ and $\mathbf{f}^{-1}$ both map $\mathbb{D} \cap (\mathbb{Q} + i\mathbb{Q})$ to some subset of $\mathbb{D} \cap (\mathbb{Q} + i\mathbb{Q})$ our problem reduces to replacing $\mathbf{f}(\mathbf{z_3})$ by an element V of $B(\mathbf{f}(z_3), \varepsilon) \cap (\mathbb{Q} + i\mathbb{Q})$ satisfying $\mathcal{A}(G[0_{\mathbb{C}}, \mathbf{f}(\mathbf{z_0}), \mathbf{f}(\mathbf{z_1}), \mathbf{f}(\mathbf{z_2}), V]) = 2\pi$. Indeed, by setting $U = \mathbf{f}^{-1}(V)$ we obtain $U \in B(z_3, \varepsilon) \cap (\mathbb{Q} + i\mathbb{Q})$ and $\mathcal{A}(G[-\mathbf{z_0}, \mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, U]) = \mathcal{A}(G[0_{\mathbb{C}}, \mathbf{f}(\mathbf{z_0}), \mathbf{f}(\mathbf{z_1}), \mathbf{f}(\mathbf{z_2}), V]) = 2\pi$.

To find such a point V, we define a polynomial $P \in \mathbb{Q}[X]$ by setting

$$P(X) = \mathrm{Im}\left[(1 - \mathbf{f}(\mathbf{z_0})\overline{\mathbf{f}(\mathbf{z_1})})(1 - \mathbf{f}(\mathbf{z_1})\overline{\mathbf{f}(\mathbf{z_2})})(1 - X\mathbf{f}(\mathbf{z_2})\overline{\mathbf{f}(\mathbf{z_3})})\right].$$

Observe that the degree of P is at most 1, and thus $P(X) = (P(1) - P(0))X + P(0)$. We first show that if we choose $\mathbf{z_k}$ close to z_k for every $k \in \{0, 1, 2, 3\}$, $P(1)$ is close to 0 and $P(0)$ close to $\kappa > 0$. Since $0 = \arg \mathbf{f}(\mathbf{z_0}) < \arg \mathbf{f}(\mathbf{z_1}) < \arg \mathbf{f}(\mathbf{z_2}) < \arg \mathbf{f}(\mathbf{z_3}) < \pi$ we can apply Equality (1)

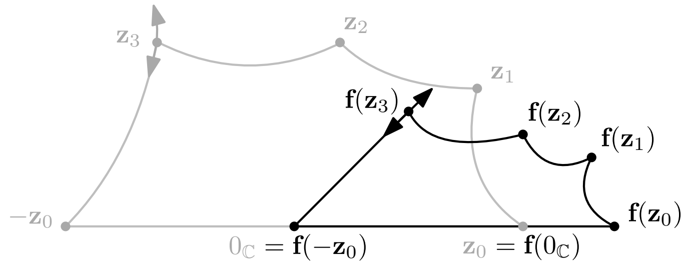


Figure 4: Illustration of the proof of Theorem 2

.

and obtain

$$\begin{aligned}
&\arg\left[\left(1-\mathbf{f(z_0)}\overline{\mathbf{f(z_1)}}\right)\left(1-\mathbf{f(z_1)}\overline{\mathbf{f(z_2)}}\right)\left(1-\mathbf{f(z_2)}\overline{\mathbf{f(z_3)}}\right)\right]\\
&= \arg\left(1-\mathbf{f(z_0)}\overline{\mathbf{f(z_1)}}\right)+\arg\left(1-\mathbf{f(z_1)}\overline{\mathbf{f(z_2)}}\right)+\arg\left(1-\mathbf{f(z_2)}\overline{\mathbf{f(z_3)}}\right)\\
&= \frac{1}{2}\left[\mathcal{A}\left(G[0_{\mathbb{C}},\mathbf{f(z_0)},\mathbf{f(z_1)}]\right)+\mathcal{A}\left(G[0_{\mathbb{C}},\mathbf{f(z_1)},\mathbf{f(z_2)}]\right)+\mathcal{A}\left(G[0_{\mathbb{C}},\mathbf{f(z_2)},\mathbf{f(z_3)}]\right)\right]\\
&= \frac{1}{2}\mathcal{A}\left(G[0_{\mathbb{C}},\mathbf{f(z_0)},\mathbf{f(z_1)},\mathbf{f(z_2)},\mathbf{f(z_3)}]\right)=\frac{1}{2}\mathcal{A}\left(G[-\mathbf{z_0},\mathbf{z_0},\mathbf{z_1},\mathbf{z_2},\mathbf{z_3}]\right).
\end{aligned}$$

By observing that every expression in between the equalities belongs to $[0,2\pi[$ we see that those equalities are indeed equalities and not only congruences modulo 2π. By choosing $\mathbf{z_k}$ close to z_k for every $k\in\{0,1,2,3\}$ we make the last expression approach $\frac{1}{2}\mathcal{A}(G[-z_0,z_0,z_1,z_2,z_3])=\pi$, which makes $P(1)$ tend to 0. Similarly, we obtain

$$\arg\left[\left(1-\mathbf{f(z_0)}\overline{\mathbf{f(z_1)}}\right)\left(1-\mathbf{f(z_1)}\overline{\mathbf{f(z_2)}}\right)\right]=\frac{1}{2}\mathcal{A}\left(G[-\mathbf{z_0},\mathbf{z_0},\mathbf{z_1},\mathbf{z_2}]\right).$$

By choosing $\mathbf{z_k}$ closer and closer to z_k, the last expression tends to $\frac{1}{2}\mathcal{A}(G[-z_0,z_0,z_1,z_2])$ which is not congruent to 0 modulo π. Thus $P(0)$ is close to some constant $\kappa>0$, whence we can assume that $P(1)\neq P(0)$.

To construct V set $\lambda=\frac{P(0)}{P(0)-P(1)}$ and let $V=\lambda\mathbf{f(z_3)}$; we have both $V\in\mathbb{Q}+i\mathbb{Q}$ and $P(\lambda)=0$. We proved that $P(1)$ tends to 0 and that $P(0)$ tends to $\kappa>0$ so λ tends to 1 and V tends to $\mathbf{f(z_3)}$. Finally, observe that $P(\lambda)=0$ implies $\mathcal{A}(G[0_{\mathbb{C}},\mathbf{f(z_0)},\mathbf{f(z_1)},\mathbf{f(z_2)},V])=2\pi$ by Equality (1). ☐

Remark 3. This theorem implies the density of the hyperbolic structures corresponding to rational admissible 4-tuples in $\mathcal{TM}_2$ with its canonical topology. However, a proof would go beyond the scope of this paper and would be quite technical.

5 The generation of the input for the experiments

We generate input for the Delaunay flips algorithms by triangulating admissible loosely-symmetric octagons; the latter are defined in Section 5.1 as a slight extension of the admissible symmetric octagons (Section 2.4). We generate such octagons whose vertices are represented in $\mathbb{D}$ by complex numbers with rational real and imaginary parts. To conduct experiments, we need to generate a large number of triangulations with a large diameter. An effective approach consists in generating octagons with a small diameter and twisting them (Section 5.1) many times to obtain octagons with a very large diameter. This approach has two advantages. Firstly it can generate triangulations with different diameters but lying on a single surface, hence eliminating any dependency on the choice of the surface if needed. Secondly thanks to this approach we will also study the dependency of the number of flips on those twists.

5.1 The twists of admissible loosely-symmetric octagons

We say that a hyperbolic octagon P is *loosely-symmetric* if the opposite sides of P are isometric and the opposite interior angles of P are equal. If moreover $\mathcal{A}(P) = 4\pi$ then we call P *admissible.* Clearly, the symmetric octagons of Aigon-Dupuy *et al.* (Section 2.4) are loosely-symmetric octagons. Identifying the opposite sides of an admissible loosely-symmetric octagon gives a surface of genus 2 [4, Theorem 1.3.5].

Let $G[z_0, \ldots, z_7]$ be an admissible loosely-symmetric octagon. We will consider the Dehn twists along the axes of its side-pairings (Figure 1). These twists generate a subgroup of $\mathrm{Mod}(S_2)$ (Section 2.3), which contains non-reducible elements of $\mathrm{Mod}(S_2)$ since the generators do not all fix a common curve. Thus, this subgroup contains pseudo-Anosov elements [14].

For every $k \in \{0, \ldots, 7\}$ we denote by τ_k the orientation preserving isometry of $\mathbb{D}$ that satisfies $\tau_k(z_{k+5}) = z_k$ and $\tau_k(z_{k+4}) = z_{k+1}$. We fix $t \in \{0, \ldots, 7\}$. For $k \in \{0, \ldots, 7\}$ let

$$z'_k = \begin{cases} \tau_t(z_k) & \text{if } k - t \in \{1, 2, 3, 4\} \mod 8, \\ z_k & \text{otherwise.} \end{cases}$$

By the Gauss-Bonnet formula, the interior angles of $G[z_0, \ldots, z_7]$ sum up to 2π; since opposite interior angles are equal, each interior angle is at most π. Thus the geodesic segment between z_{t+1} and z_{t+5} is contained in $G[z_0, \ldots, z_7]$ and cuts the polygon into the two interior disjoint pentagons $P_1 = G[z_{t+1}, z_{t+2}, z_{t+3}, z_{t+4}, z_{t+5}]$ and $P_2 = G[z_{t+5}, z_{t+6}, z_{t+7}, z_t, z_{t+1}]$; the intersection of P_1 and P_2 is the segment between z_{t+5} and z_{t+1} and the two pentagons are isometric. Similarly, $\tau_t(P_1)$ and P_2 are interior disjoint, they intersect on the segment between $z'_t = z_t$ and $z'_{t+4} = z_{t+1}$ and their union is $G[z'_0, \ldots, z'_7]$. It follows that $G[z'_0, \ldots, z'_7]$ is an admissible loosely-symmetric octagon; the surface that it defines is isometric to the one defined by $G[z_0, \ldots, z_7]$ as both can be obtained by the same identification of the sides of P_1 and P_2 (P_1 and $\tau_t(P_1)$ being isometric). We say that $(z'_0, \ldots, z'_7)$ is obtained by *t-twisting* $(z_0, \ldots, z_7)$. For every point z in the closure of $G[z_0, \ldots, z_7]$ at least z or $\tau_t(z)$ lies in the closure of $G[z'_0, \ldots, z'_7]$.

Let us denote by $(\tau'_k)_{0 \le k \le 7}$ the isometries defined for $z'_0, \ldots, z'_7$, in the same way as $(\tau_k)_{0 \le k \le 7}$ above. By definition of the t-twist, the following holds for every $k \in \{0, \ldots, 7\}$

$$\tau'_k = \begin{cases} \tau_t \circ \tau_k & \text{if } k - t \in \{1, 2, 3\} \mod 8, \\ \tau_k \circ \tau_t^{-1} & \text{if } t - k \in \{1, 2, 3\} \mod 8, \\ \tau_k & \text{if } k = t \mod 4. \end{cases}$$

For a word $t = t_1 \cdots t_m$, we define the t-twist as the composition of the t_k-twists, $k = 1, \ldots, m$, in this order. We pick $t_1, \ldots, t_m$ in $\{0, \ldots, 3\}^m$ instead of $\{0, \ldots, 7\}^m$ to only consider the generators without their inverses and obtain large diameters as quickly as possible.

5.2 The generation of triangulations

The input surfaces and triangulations are constructed in four steps. We refer to Appendix D for details.

[step 1] We construct an initial rational admissible 4-tuple $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$.

[step 2] We choose $n_p \geq 0$ and construct points $(\mathbf{p}_1, \ldots, \mathbf{p}_{n_p}) \in (\mathbb{Q} + i\mathbb{Q})^n$ lying within the closure of $\mathcal{P}[\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}]$.

[step 3] We choose $m \geq 0$ and a sequence $t_1 \cdots t_m$ of twists.

[step 4] From the 4-tuple $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$, the points $(\mathbf{p}_1, \ldots, \mathbf{p}_{n_p})$, and the sequence $t = t_1 \cdots t_m$, we construct a representation (M, F, A) of an input triangulation T. This is (roughly) done by t-twisting $\mathcal{P}[\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}]$, triangulating the octagon resulting from the twists, and inserting the points in the faces of the resulting triangulation. Together with the point corresponding to the vertices of the octagon, the triangulation T has $n = n_p + 1$ vertices.

Sections 6 and 7 refer to these four steps. Step 1 was applied a thousand times to construct the 1,000 rational admissible 4-tuples $Q_1, \ldots, Q_{1{,}000}$; in our experiments we consider the first n_q 4-tuples. In step 3 we constructed 10,000 random sequences of twists denoted by $S_1, \ldots, S_{10{,}000}$, each of length 10, of which some of the experiments use the first n_s sequences. The values of n_p, n_q, and n_s are specified in the description of each experiment.

Technicalities for steps 1, 2, and 4 are deferred to Appendix D. We only elaborate on step 3 here. Consider a sequence of m twists represented by the word $t = t_1 \ldots t_m$ (see Section 5.1). We will study two kinds of sequences in the experiments of Sections 6 and 7:

- A *power sequence* is represented by a word u^m for some $u \in \{0, \ldots, 3\}$.
- In a *random sequence*, $t_1, \ldots, t_m$ are chosen uniformly and independently in $\{0, \ldots, 3\}$.

It appears in practice that the length of a random sequence has a stronger impact on the computations than the length of a power sequence. When twisting, we update an 8-tuple $(\mathbf{z_0}, \ldots, \mathbf{z_7}) \in (\mathbb{Q}+i\mathbb{Q})^8$ corresponding to an admissible loosely-symmetric octagon $G[\mathbf{z_0}, \ldots, \mathbf{z_7}]$, together with the orientation preserving isometries $(\tau_k)_{0\le k\le 7}$ identifying its opposite sides. Both the vertices of the octagon and the isometries are represented by complex numbers: $8(m+1)$ complex numbers for the $8(m+1)$ points and $32(m+1)$ complex numbers for the $8(m+1)$ isometries. Each such complex number is represented by two rational numbers and each such rational number is represented by two integers: its numerator and its denominator. The running time of a sequence of $m \ge 0$ twists thus depends on the sizes of these $160(m+1)$ integers; here the size of an integer is the number of digits of its decimal representation.

Twisting promptly gives rise to big numbers. As an example, take the rational admissible 4-tuple $\mathbf{z_0} = 10/11, \mathbf{z_1} = 1/2 + 1/2i, \mathbf{z_2} = -1/10 + 9/10i, \mathbf{z_3} = -3/5 + 3/5i$. Then twist $(\mathbf{z_0}, \ldots, \mathbf{z_3}, -\mathbf{z_0}, \ldots, -\mathbf{z_3})$ by m twists and get $(\mathbf{z}_{0,m}, \ldots, \mathbf{z}_{7,m}) \in (\mathbb{Q} + i\mathbb{Q})^8$ such that $G[\mathbf{z}_{0,m}, \ldots, \mathbf{z}_{7,m}]$ is an admissible loosely-symmetric octagon. Denote by $\tau_{k,m}$ the orientation preserving isometry of $\mathbb{D}$ mapping $\mathbf{z}_{k+5,m}$ to $\mathbf{z}_{k,m}$ and $\mathbf{z}_{k+4,m}$ to $\mathbf{z}_{k+1,m}$, for $k \in \{0, \ldots, 7\}$.

Consider first the power sequence of twists 0^m (the choice of 0 is without loss of generality) for $m \in \{0, \ldots, 3000\}$. A simple recursion gives the following for every m:

$$\tau_{k,m} = \begin{cases} (\tau_{0,0})^m \circ \tau_{k,0} & \text{if } k \in \{1,2,3\} \mod 8 \\ \tau_{k,0} \circ (\tau_{0,0})^{-m} & \text{if } k \in \{5,6,7\} \mod 8 \\ \tau_{k,0} & \text{if } k = 0 \mod 4 \end{cases}$$

$$\mathbf{z}_{k,m} = \begin{cases} (\tau_{0,0})^m(\mathbf{z}_{k,0}) & \text{if } k \in \{1,2,3\} \mod 8, \\ \mathbf{z}_{k,0} & \text{otherwise.} \end{cases}$$

From that it is easy to see that the sizes of the integers involved in the representations of $(\tau_{k,m})_{0\le k\le 7}$ and $(\mathbf{z}_{k,m})_{0\le k\le 7}$ grow at most linearly in m. See Figure 5.

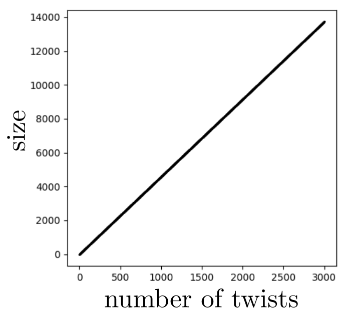


Figure 5: Size of the numerator of $\mathrm{Re}\,[z_{1,m}]$ as a function of the length m of a power sequence of twists.

When twisting with a random sequence, the growth in size of the integers involved does not appear to be linear in the number of twists. Examples are shown in Figure 6 for the random sequence 23330132013121032301 of $m = 20$ twists. Observe that the numerator of the real part of the top-left coefficient of the matrix representing $\tau_{0,20}$ contains more than $200,000$ digits in its decimal representation. In general the bottleneck for such computations seems to be the size of such coefficients of the isometries $(\tau_{k,m})_{0\leq k\leq 7}$.

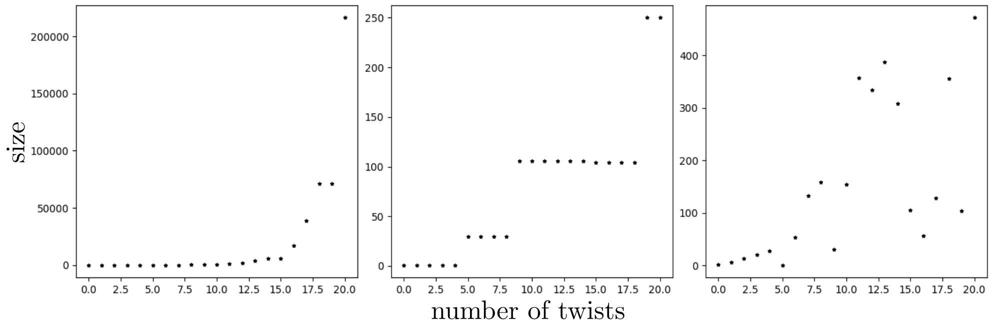


Figure 6: The size of integers as a function of the length m of a random sequence of twists. Left: the numerator of the real part of the top-left coefficient of the matrix representation of $\tau_{0,m}$. Middle: the numerator of $\mathrm{Re}\,[\mathbf{z}_{1,m}]$. Right: the numerator of $\mathrm{Re}\,[\mathbf{z}_{4,m}]$.

6 Comparison of flip strategies

As recalled in Section 2.2, a Delaunay flip algorithm can flip Delaunay-flippable edges in any order. In this section, we consider six strategies:

- *naive* strategy: choose the first Delaunay-flippable edge given by the CGAL combinatorial map iterator `DartRange::iterator`.
- *random* strategy: choose uniformly among all the Delaunay-flippable edges.
- *minimag* and *maximag* strategies: choose the edge e whose cross-ratio $\mathcal{R}_T(e)$ minimizes (resp. maximizes) $\mathrm{Im}\,[\mathcal{R}_T(e)]$ among the Delaunay-flippable edges.
- *minratio* and *maxratio* strategies: choose the edge e whose cross-ratio $\mathcal{R}_T(e)$ minimizes (resp. maximizes) the quotient $|\mathrm{Im}\,[\mathcal{R}_T(e)]| \,/\, |\mathcal{R}_T(e)|$.

We present eight experiments A, B, C, D, E, F, G, and H, allowing us to compare the number of flips that the six strategies induce on a variety of inputs. The notation Q_k, S_k and the parameters n_q, n_s, and n_p are defined in Section 5.

exp.	A	B	C	D
n_q	50	30	10	1
n_s	50	30	10	10
n_p	0	10	100	$1,000$

exp.	E	F	G	H
n_q	100	30	10	10
Ω	$0, 30, 60, 90, 120$	$0, 30, 60, 90, 120$	$0, 10, 20, 30$	$0, 5, 10$
n_p	0	10	100	$1,000$

Table 1: Parameters for experiments A to H.

Let us first check that the strategy actually has an influence on the number of flips. Experiments A, B, C, and D use random sequences of twists. The values of n_q, n_s, and n_p are shown in Table 1. We first construct the set X containing the 11 prefixes of the sequence of twists S_l (including the empty sequence) for every $l \in \{1, \ldots, n_s\}$: X contains at most $10n_s + 1$ sequences whose sizes vary between 0 and 10. Then for every $k \in \{1, \ldots, n_q\}$ and every $t \in X$, we perform steps 2 and 4 with $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}) = Q_k$, n_p interior vertices, and $t_1 \cdots t_m = t$. In each case, we run the Delaunay flip algorithm six times: one for each strategy, and count the number of flips that were needed for the algorithm to terminate. Among those six integers we denote by $\alpha_{k,t}$ the minimum and by $\beta_{k,t}$ the maximum. Figure 7 shows that choosing a strategy has an impact on the number of flips. A point lying far from the diagonal $y = x$ represents a computation where one of the strategies clearly requires more flips than another, while a point lying close to the diagonal represents a computation where the strategies were essentially equivalent in the number of flips they induced.

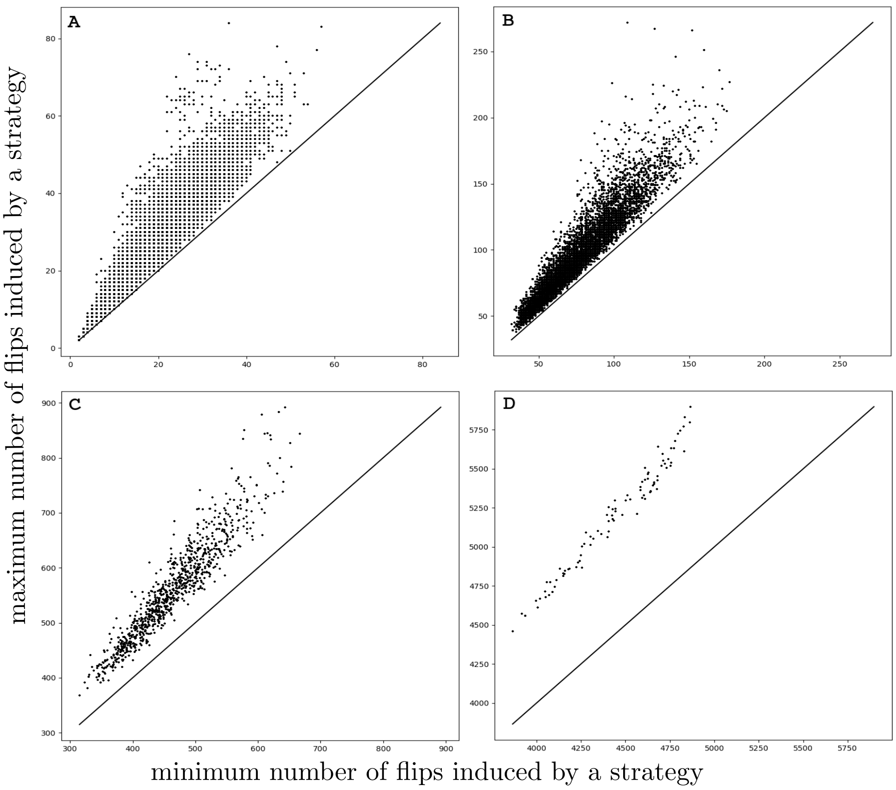


Figure 7: Experiments A, B, C, and D: points $(\alpha_{k,t}, \beta_{k,t})$ for $k \in \{1, \ldots, n_q\}$ and $t \in X$.

Experiments E, F, G, and H use power sequences of twists. They are parameterized by n_q, n_p, and a set Ω of integers giving the lengths of the considered twists, see Table 1. For every $k \in \{1, \ldots, n_q\}$, every $m \in \Omega$, and every $u \in \{0, 1, 2, 3\}$, we perform steps 2 and 4 with $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}) = Q_k$, n_p interior vertices, and $t_1 \ldots t_m = u^m$. Then we run the Delaunay flip algorithm for each of the six strategies. Here, the minimum and maximum number of flips are respectively denoted by $\alpha_{k,m,u}$ and $\beta_{k,m,u}$. Figure 8 shows a stronger impact of the strategy on the number of flips than experiments A, B, C, D.

To compare the six strategies, we count for each experiment and each strategy the number of times (i.e., the number of pairs $(\alpha_{k,t}, \beta_{k,t})$ for experiments A, …, D or pairs $(\alpha_{k,m,u}, \beta_{k,m,u})$ for experiments E, …, H) when the strategy induced the minimum/maximum number of flips among the other strategies. Figure 9 summarizes the results. Overall the minratio and the maxratio strategies seem to regularly achieve the maximum and the minimum (respectively). Observe in particular that in experiments D and H the minratio and the maxratio strategies *always* induced more and fewer flips, respectively, than any other strategy.

The naive strategy seems to rarely achieve the minimum or the maximum number of flips among the six strategies. In Figures 10 and 11, the y-coordinate is the number of flips induced by the naive strategy (instead of the maximum among the six strategies); the x-coordinate is

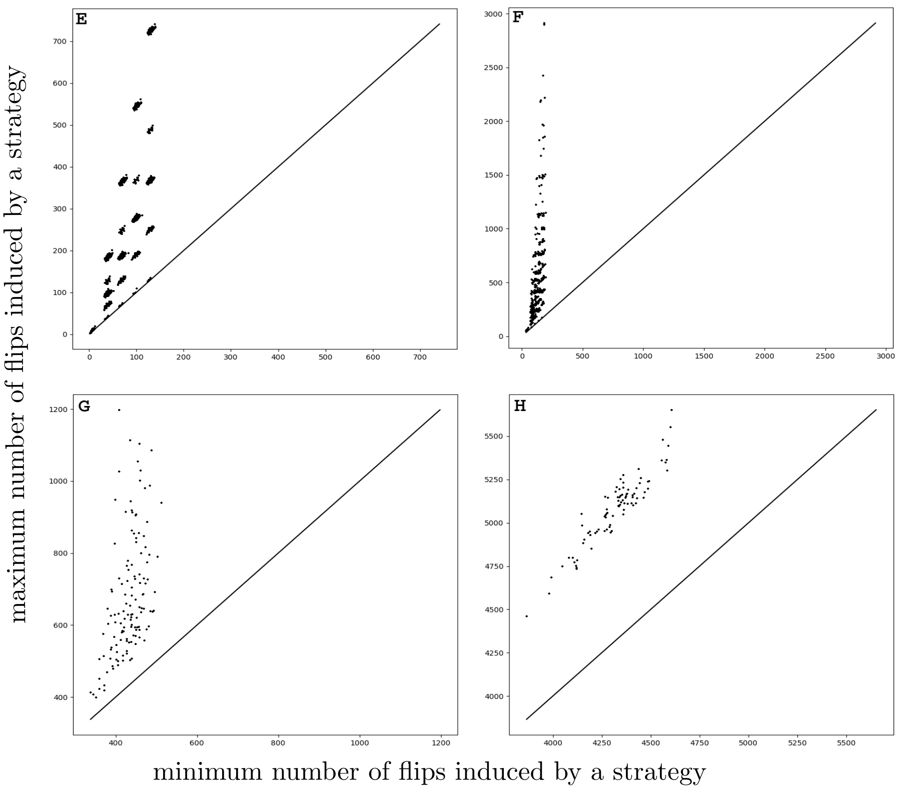


Figure 8: Experiments E, F, G, and H: points $(\alpha_{k,m,u}, \beta_{k,m,u})$, $k \in \{1, \ldots, n_q\}, m \in \Omega, u \in \{0, 1, 2, 3\}$.

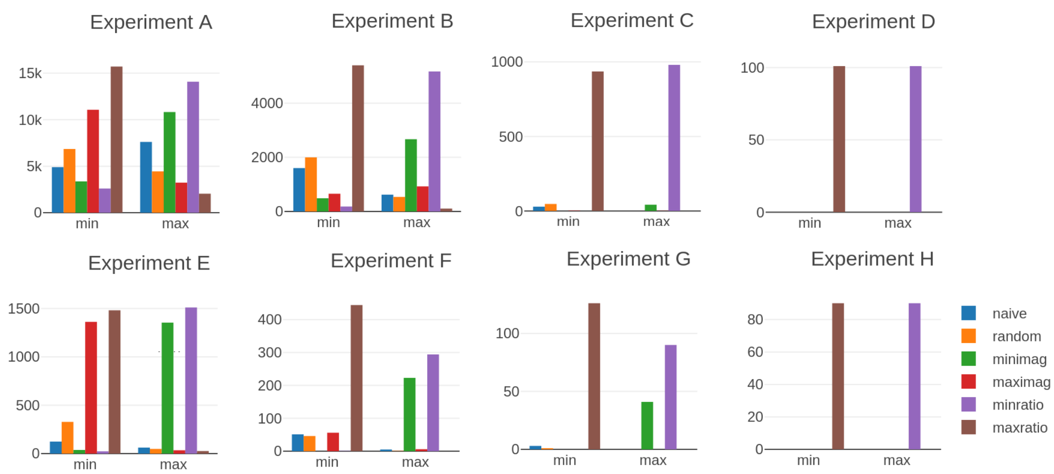


Figure 9: The number of times when each strategy induced the minimum/maximum number of flips.

still the minimum number of flips among the six strategies. The figures show that the number of flips required by the naive strategy is close to the minimum. As it runs much faster than all other strategies, we stick to the naive strategy for the experiments of Section 7.

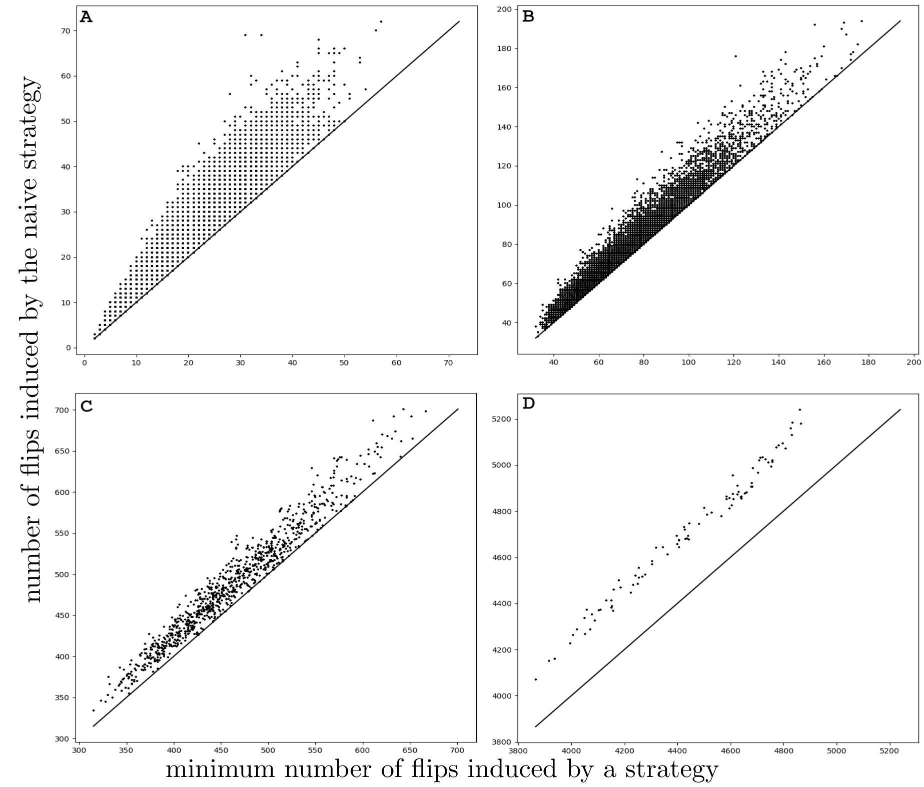


Figure 10: The number of flips induced by the naive strategy with respect to the minimum among the six strategies in experiments A, B, C, and D.

Figure 12 illustrates a run of the program. The diameter of the initial domain is about 139 and the diameter of the final domain is smaller than 5.

7 Exploring the relationship between number of flips and diameter

7.1 Rationale for the experiments

Mark Bell [2] showed that the structure of the mapping class group has a very interesting effect on the flip graph of topological triangulations. In this topological setting, the objective is to reduce the number of intersections k of the input triangulation with a fixed curve. The main theorem of Bell's paper states that one can always find a flip or a power of a Dehn twist that reduces the number of crossings by a fixed percentage. This result can be seen as follows: either a pseudo-Anosov transformation allows the number of crossings to decrease in a single application, or there exists a power of a Dehn twist that reduces the number of crossings. This gives an algorithm to compute the optimal triangulation using $O(\log(k))$ operations.

Our problem is different from Mark Bell's: in his study, the number of crossings is an explicit measure of the distance to the goal, while there is no way to know in advance how far the input triangulation is from being Delaunay, and we do not know the homotopy classes of final edges. However, asymptotically, combinatorial intersection metrics are very similar to the hyperbolic metrics on surfaces of genus $g \geq 2$. If a triangulation has very long edges (in terms of the number of crossings for the topological version, or in terms of the hyperbolic length in our

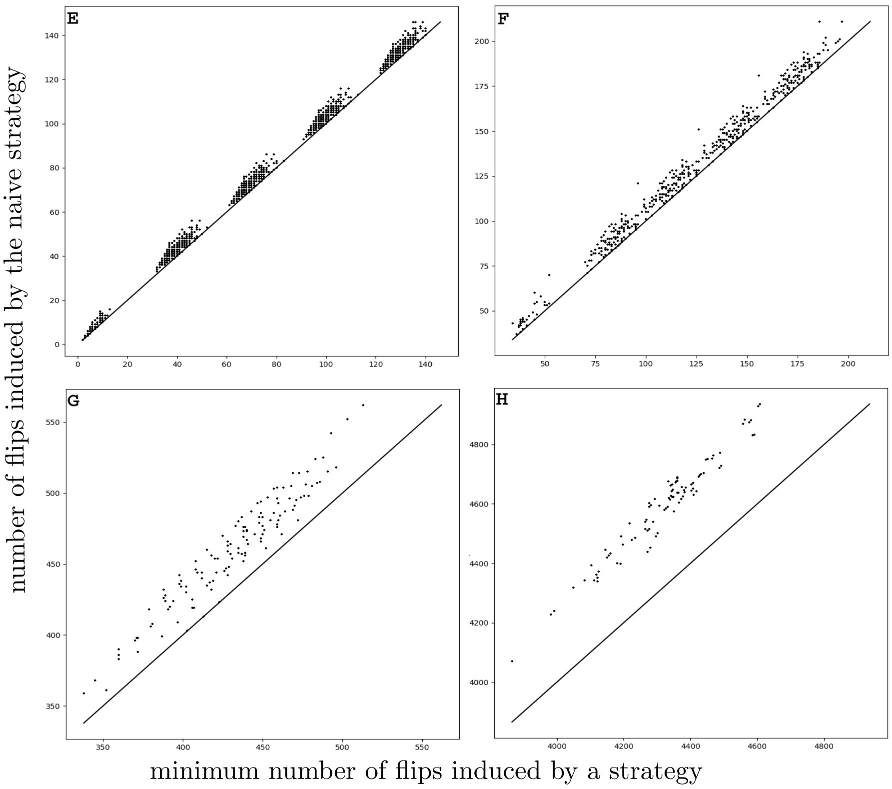


Figure 11: Same as Figure 10, for experiments E, F, G, and H.

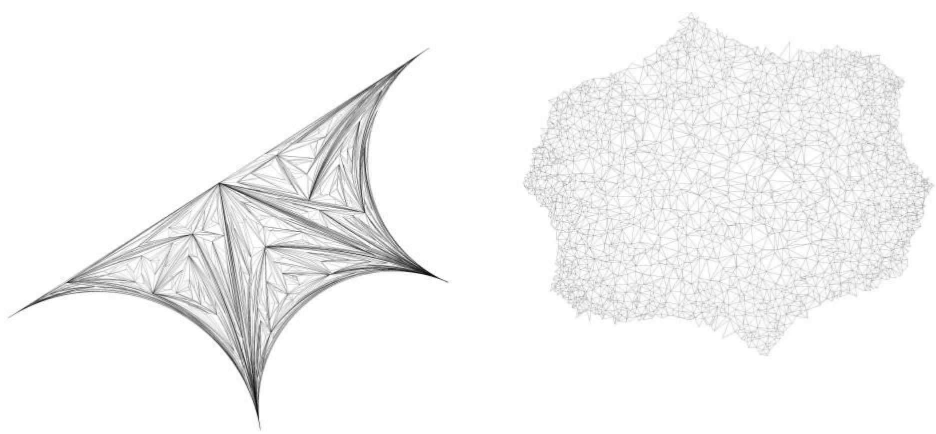


Figure 12: Triangulation with 3001 vertices before (left) and after (right) the flips.

geometric setting), then in the first stage both strategies aim at reducing edge lengths. Thus the two problems might have a similar asymptotic efficiency.

This raises two questions:

- Is there any hope to experimentally observe such similarities in the efficiency? It looks *a priori* unpromising as the above only holds asymptotically.
- Can Mark Bell's result be transposed to the number of flips?

We carry out two sets of experiments. The first set constructs the input triangulation by twisting the initial octagon in one direction only; as these twists correspond to reducible elements of $\mathrm{Mod}(S_2)$ (Section 2.3) we expect to observe a linear number of flips. The second set of experiments twists the octagon in a random way; asymptotically, we should obtain pseudo-Anosov elements of the mapping class group and an asymptotic logarithmic behavior.

We present five experiments named I, J, K, L, and M, all using the naive strategy (see Section 6). We use again the same notation as in Section 5. We follow steps 3 and 4 and keep track of the loosely-symmetric octagon $G[\mathbf{z_0}', \dots, \mathbf{z_7}']$ obtained in step 4 after the twists; we compute (an approximation represented by a C++ `double` of) the hyperbolic diameter of $G[\mathbf{z_0}', \dots, \mathbf{z_7}']$ (see Appendix E). As we are only interested in the influence of the diameter, we do not run step 2 (i.e., we set $n_p = 0$) and the triangulation thus has only one vertex.

7.2 Exploring with power sequences

Experiments I and J are parameterized by the number n_q of 4-tuples: $n_q = 1$ in I and $n_q = 1,000$ in J. We perform step 4 with $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}) = Q_k$, $n_p = 0$, and $t_1 \dots t_m = u^{3m}$ for $k \in \{1, \dots, n_q\}$, $u \in \{0, 1, 2, 3\}$, and $m \in \{0, \dots, 50\}$, and we compute the approximate hyperbolic diameter $\varnothing_{k,m,u}$ of $G[\mathbf{z_0}', \dots, \mathbf{z_7}']$. We run the Delaunay flip algorithm, counting the number $\alpha_{k,m,u}$ of flips needed for the algorithm to terminate. Figure 13 shows the result.

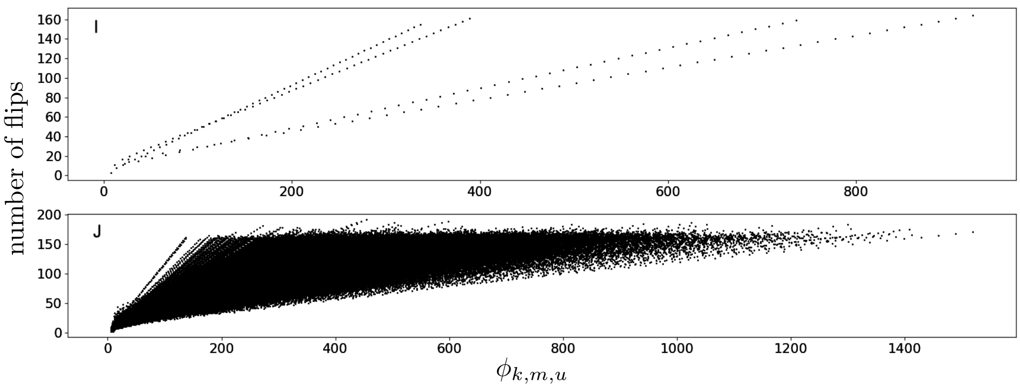


Figure 13: Experiments I and J: the number of flips $\alpha_{k,m,u}$ with respect to the (approximate) diameter $\varnothing_{k,m,u}$, $k \in \{1, \dots, n_q\}, m \in \{0, \dots, 50\}, u \in \{0, 1, 2, 3\}$.

7.3 Exploring with random sequences

In the following experiments the values of n_q and n_s are respectively $n_q = 1, n_s = 10,000$ (experiment K), $n_q = 10, n_s = 1,000$ (experiment L), and $n_q = 1,000, n_s = 100$ (experiment M). We first construct the set X containing the 11 prefixes of S_l (including the empty sequence) for every $l \in \{1, \dots, n_s\}$. Then for every $k \in \{1, \dots, n_q\}$ and every $t \in X$, we perform step 4 with $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}) = Q_k$, $n_p = 0$, and $t_1 \dots t_m = t$. We compute the approximate diameter $\varnothing_{k,t}$ of $G[\mathbf{z_0}', \dots, \mathbf{z_7}']$. We run the Delaunay flip algorithm and count the number $\alpha_{k,t}$ of flips needed for the algorithm to terminate. Figure 14 shows $\alpha_{k,t}$ as a function of $10 \ln(\varnothing_{k,t})$ for $k \in \{1, \dots, n_q\}$ and $t \in X$. Here ln denotes the natural logarithm (base e).

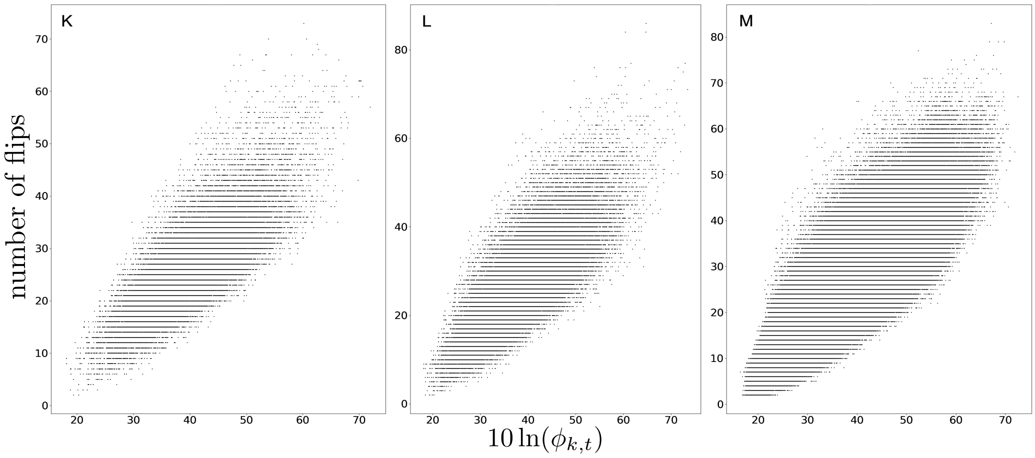


Figure 14: Experiments K, L, and M: the number of flips with respect to $10\ln(\varnothing_{k,t})$, $k \in \{1,\ldots,n_q\}, t \in X$; the maximum diameter is about 1500.

7.4 Interpretation of the results

Our experiments show that controlling the elements of the mapping class group $\mathrm{Mod}(S_2)$ used for twisting actually allows us to control the number of flips needed by the flip algorithm. Indeed, in the case of power sequences, we observe that the number of flips is linear in the diameter of the input triangulation: Delaunay flips untwist the triangulation by performing a constant number of flips per iteration of the twist. For random sequences, we observe that the number of flips is logarithmic in the diameter of the input triangulation. In practice the Delaunay flip algorithm actually realizes a strategy that is as efficient as Mark Bell's.

Surprisingly, the asymptotic behavior of random walks in the mapping class group can be observed in practice with relatively small sequences of twists: even rather short random sequences reach pseudo-Anosov homeomorphisms, yielding the logarithmic behavior.

Some of the experiments use a single input surface while other experiments use up to 1,000 different input surfaces. The behaviors observed do not depend on the surface.

In light of our experimental results, we conjecture that the complexity of the Delaunay flip algorithm is worst-case linear in the diameter of the triangulation, and logarithmic on average. It should *a priori* not depend on the genus, as Mark Bell's and Maher's results hold for any genus $g \geq 2$.

References

[1] Aline Aigon-Dupuy, Peter Buser, Michel Cibils, Alfred F. Künzle, and Frank Steiner. Hyperbolic octagons and Teichmüller space in genus 2. *Journal of mathematical physics*, 46(3):033513, 2005. doi:10.1063/1.1850177.

[2] Mark C. Bell. Simplifying triangulations. *Discrete & Computational Geometry*, 66(1):1–11, 2021. doi:10.1007/s00454-021-00309-0.

[3] M. Berger. *Geometry (vols. 1-2)*. Springer-Verlag, 1987.

[4] Peter Buser. *Geometry and Spectra of Compact Riemann Surfaces*. Birkhäuser, Boston, 1992. doi:10.1007/978-0-8176-4992-0.

[5] Guillaume Damiand. Combinatorial maps. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.2.1 edition, 2021. URL: https://doc.cgal.org/5.2.1/Manual/packages.html#PkgCombinatorialMaps.

[6] Vincent Despré, Benedikt Kolbe, and Monique Teillaud. Representing infinite hyperbolic periodic Delaunay triangulations using finitely many Dirichlet domains. Research report, INRIA, July 2021. URL: https://hal.inria.fr/hal-03045921.

[7] Vincent Despré, Jean-Marc Schlenker, and Monique Teillaud. Flipping geometric triangulations on hyperbolic surfaces. In *Proceedings of the 36th International Symposium on Computational Geometry (SoCG'20)*, pages 35:1–35:16, 2020. doi:10.4230/LIPIcs.SoCG.2020.35.

[8] Matthijs Ebbens, Iordan Iordanov, Monique Teillaud, and Gert Vegter. Delaunay triangulations of generalized Bolza surfaces. Research report, INRIA, 2021. URL: https://hal.inria.fr/hal-03080125.

[9] Benson Farb and Dan Margalit. *A primer on mapping class groups*, volume 49 of *Princeton Mathematical Series*. Princetown University Press, Princetown, 2012. doi:10.1515/9781400839049.

[10] Albert Fathi, François Laudenbach, and Valentin Poénaru. *Thurston's Work on Surfaces (MN-48)*. Princeton University Press, 2012.

[11] Michael Hemmer, Susan Hert, Sylvain Pion, and Stefan Schirra. Number types. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.3 edition, 2021. URL: https://doc.cgal.org/5.3/Manual/packages.html#PkgNumberTypes.

[12] John H. Hubbard. *Teichmüller Theory and Applications to Geometry, Topology, and Dynamics*, volume 1. Matrix Editions, 2006.

[13] Iordan Iordanov and Monique Teillaud. Implementing Delaunay triangulations of the Bolza surface. In *Proceedings of the 33rd International Symposium on Computational Geometry (SoCG'17)*, pages 44:1–44:15, 2017. doi:10.4230/LIPIcs.SoCG.2017.44.

[14] Nikolai V. Ivanov. *Subgroups of Teichmüller modular groups*, volume 115 of *Translations of Mathematical Monographs*. American Mathematical Soc., 1992. doi:10.1090/mmono/115.

[15] Joseph Maher. Random walks on the mapping class group. *Duke Mathematical Journal*, 156(3):429–468, 2011. doi:10.1215/00127094-2010-216.

[16] Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins University Press, Baltimore, 2001.

[17] Georg Osang, Mael Rouxel-Labbé, and Monique Teillaud. Generalizing CGAL periodic Delaunay triangulations. In *Proceedings 28th European Symposium on Algorithms*, pages 75:1–75:17, 2020. Best Paper Award (Track B: Engineering and Applications). `doi:10.4230/LIPIcs.ESA.2020.75`.

[18] Chee Yap *et al.* The CORE library project. URL: `http://cs.nyu.edu/exact/core_pages/`.

A Cross-ratios and Delaunay flips

We define the map $\phi : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$ by $\phi(x, y) = 1 - (1 - x) \cdot y$ for every $(x, y) \in \mathbb{C}^2$.

To prove Lemma 5, we first give a straightforward lemma. Here, the triangulation $\mathcal{T}$ may be infinite or finite.

Lemma 4. *Consider a triangulation $\mathcal{T}$ of $\mathbb{H}$ and an edge e of $\mathcal{T}$. Denote by f, g, h, k the edges, oriented counter-clockwise, of the quadrilateral formed by the two triangles of $\mathcal{T}$ that are incident to e (hence assuming that e is incident to two bounded faces). Assume that e is Delaunay-flippable and let $\mathcal{T}^\star$ be the triangulation obtained from $\mathcal{T}$ when replacing e by the other diagonal $e^\star$ of the quadrilateral. Then:*

- $\mathcal{R}_{\mathcal{T}^\star}(e^\star) = \mathcal{R}_{\mathcal{T}}(e)/(\mathcal{R}_{\mathcal{T}}(e) - 1)$.
- $\mathcal{R}_{\mathcal{T}^\star}(w) = \phi(\mathcal{R}_{\mathcal{T}}(w), \mathcal{R}_{\mathcal{T}}(e))$ *for* $w \in \{f, h\}$.
- $\mathcal{R}_{\mathcal{T}^\star}(w) = \phi(\mathcal{R}_{\mathcal{T}}(w)), 1/\mathcal{R}_{\mathcal{T}^\star}(e^\star))$ *for* $w \in \{g, k\}$.

It is clear that the cross-ratio of any edge of $\mathcal{T}$ other than $\{e, f, g, h, k\}$ remains unchanged after the flip.

Proof. Consider the notation defined by Figure 15.

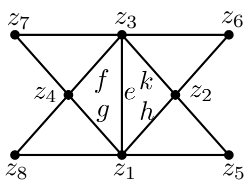


Figure 15: Notation for the proof of Lemma 4 (geodesic edges are represented by straight line segments).

A straightforward computation gives:

$$\begin{aligned}
[z_1, z_2, z_5, z_3] \cdot [z_1, z_2, z_3, z_4] &= [z_1, z_2, z_5, z_4] \\
[z_2, z_3, z_6, z_4] \cdot [z_2, z_3, z_4, z_1] &= [z_2, z_3, z_6, z_1] \\
[z_3, z_4, z_7, z_1] \cdot [z_1, z_2, z_3, z_4] &= [z_3, z_4, z_7, z_2] \\
[z_4, z_1, z_8, z_2] \cdot [z_2, z_3, z_4, z_1] &= [z_4, z_1, z_8, z_3].
\end{aligned}$$

The result follows. ☐

Let us now state the result on $\mathcal{S}$.

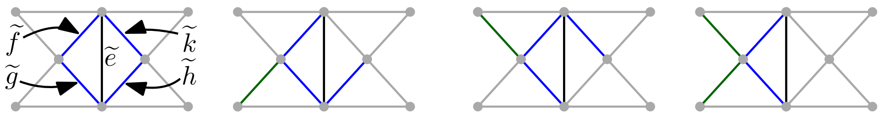
Lemma 5. *Consider a triangulation T of $\mathcal{S}$ and an edge e of T. Let f, g, h, k be the edges of T such that e, f, g and e, h, k (oriented counter-clockwise) bound the triangles incident to e in T. Assume that e is Delaunay-flippable and let $T^\star$ be the triangulation obtained from T after the flip of e and $e^\star$ be the new edge replacing e. Then the following holds:*

- $\mathcal{R}_{T^\star}(e^\star) = \mathcal{R}_T(e)/(\mathcal{R}_T(e) - 1)$.
- *If* $f \neq h$ *then* $\mathcal{R}_{T^\star}(w) = \phi(\mathcal{R}_T(w), \mathcal{R}_T(e))$ *for every* $w \in \{f, h\}$.
- *If* $f = h$ *then* $\mathcal{R}_{T^\star}(f) = \phi(\phi(\mathcal{R}_T(f), \mathcal{R}_T(e)), \mathcal{R}_T(e))$.
- *If* $g \neq k$ *then* $\mathcal{R}_{T^\star}(w) = \phi(\mathcal{R}_T(w), 1/\mathcal{R}_{T^\star}(e^\star))$ *for every* $w \in \{g, k\}$.
- *If* $g = k$ *then* $\mathcal{R}_{T^\star}(g) = \phi(\mathcal{R}_T(g), \phi(\mathcal{R}_T(g), 1/\mathcal{R}_{T^\star}(e^\star)), 1/\mathcal{R}_{T^\star}(e^\star))$.

Before going to the proof, note that some edges in $X = \{e, f, g, h, k\}$ may be equal. However, the edges e, f, g are pairwise-distinct and so are the edges e, h, k as they bound faces of $\widetilde{T}$. Also, $f \neq k$ and $g \neq h$ because the interior angles of the faces of T are all less than π. Hence the only two possible equalities in X are between f and h, and between g and k.

One easily sees that the cross-ratio of any edge $w \notin X$ remains unchanged after the flip.

Proof. Consider the lift $\widetilde{T}$ of T. Choose a fixed lift $\widetilde{e}$ of e and let $\widetilde{f}, \widetilde{g}, \widetilde{h}, \widetilde{k}$ be the edges of $\widetilde{T}$ such that $\widetilde{e}, \widetilde{f}, \widetilde{g}$ and $\widetilde{e}, \widetilde{h}, \widetilde{k}$ bound the two faces incident to $\widetilde{e}$ in $\widetilde{T}$, oriented counter-clockwise. By renaming $\widetilde{f}, \widetilde{g}$ to $\widetilde{h}, \widetilde{k}$ and *vice versa* if needed we can also assume that each $w \in X$ is lifted by $\widetilde{w}$. We define $\widetilde{X_1}$ as $\{\widetilde{f}, \widetilde{h}\}$ if $f \neq h$ or as $\{\widetilde{f}\}$ if $f = h$. We define $\widetilde{X_2}$ similarly for g and k. Then we set $\widetilde{X} = \{\widetilde{e}\} \cup \widetilde{X_1} \cup \widetilde{X_2}$. This way $\widetilde{X}$ contains exactly one lift of each element of X. Define $\widetilde{E}$ as the set of all lifts of e that are incident to one of the faces of $\widetilde{T}$ having an edge in $\widetilde{X}$. The possible configurations are summarized in Figure 16. Consider the infinite triangulation $\widetilde{T'}$



$f \neq h$ and $g \neq k$ $\quad$ $f \neq h$ and $g = k$ $\quad$ $f = h$ and $g \neq k$ $\quad$ $f = h$ and $g = k$

Figure 16: The possible configurations:$\widetilde{e}$ is the black segment, $\widetilde{X} \setminus \{\widetilde{e}\} = \widetilde{X_1} \cup \widetilde{X_2}$ is in blue, and $\widetilde{E} \setminus \{\widetilde{e}\}$ is in green.

of the hyperbolic plane obtained from $\widetilde{T}$ after flipping each element of $\widetilde{E}$. We denote by $\widetilde{e^\star}$ the edge of $\widetilde{T'}$ resulting from the flip of $\widetilde{e}$. Then for every $\widetilde{w} \in \widetilde{X} \setminus \{\widetilde{e}\}$ we have $\mathcal{R}_{T^\star}(w) = \mathcal{R}_{\widetilde{T'}}(\widetilde{w})$ and $\mathcal{R}_T(w) = \mathcal{R}_{\widetilde{T}}(\widetilde{w})$. Also, $\mathcal{R}_{T^\star}(e^\star) = \mathcal{R}_{\widetilde{T'}}(\widetilde{e^\star})$ and $\mathcal{R}_T(e) = \mathcal{R}_{\widetilde{T}}(\widetilde{e})$. The result follows by computing $\mathcal{R}_{\widetilde{T'}}(\widetilde{w})$ and $\mathcal{R}_{\widetilde{T'}}(\widetilde{e^\star})$ using Lemma 5. ☐

B Details for the representation of triangulations (Section 3)

B.1 On combinatorial maps and the anchor (Section 3.1)

A 2-dimensional combinatorial map can be described as a finite set whose elements are called darts together with three permutations β_0, β_1, and β_2 of this set of darts. The permutations β_0 and β_1 are the inverse of each other while the permutation β_2 is an involution. We use 2-dimensional combinatorial maps to describe graphs cellularly embedded on surfaces as follows. For each face of a graph we constitute a cycle of darts such that given a dart d the next dart in the cycle is $\beta_1(d)$ (and thus the previous one is $\beta_0(d)$). The darts of the cycle represent the edges bordering the face. We "glue" faces along their borders by pairing darts: given two darts d and d' we set $\beta_2(d) = \beta_2(d')$. It is possible to identify two darts that belong to a single face. We refer to the literature for a formal definition [16, Section 3.3].

Now we explain the role played by the anchor A in the data structure (M, F, A) described in Section 3.1. If z_1, z_2, z_3, and z_4 denote 4 distinct complex numbers then the number z_4 can be deduced from z_1, z_2, and z_3 and from the cross-ratio $[z_1, z_2, z_3, z_4]$. This fact has a consequence useful to us: given an infinite triangulation $\mathcal{T}$ of the hyperbolic plane if one knows the coordinates in $\mathbb{D}$ of the 3 vertices of some face of $\mathcal{T}$ together with the cross-ratio of every edge of $\mathcal{T}$ then one can recursively compute the coordinates of any point of $\mathcal{T}$. In our setting $\mathcal{T}$ is the lift $\widetilde{T}$ of a triangulation T of a surface and for every edge e of T we have $\mathcal{R}_{\widetilde{T}}(\widetilde{e}) = \mathcal{R}_T(e)$ by definition. In the data structure (M, F, A) representing the triangulation T the cross-ratios of $\widetilde{T}$ are given by F and the anchor A provides the coordinates of the 3 vertices of some face of $\widetilde{T}$. Consequently given (M, F, A) one can construct the coordinates of the vertices of $\widetilde{T}$. That enables drawing some part of $\widetilde{T}$ for example (this is our use of the anchor).

B.2 Flipping edges (Section 3.2)

In this section, we explain how the flip of an edge e of a triangulation T is encoded on the data structure (M, F, A) defined in Section 3.1 and representing T. We may not make the distinction between the edges of the triangulation T and those of the combinatorial map M. We are given as input a dart d_e of the combinatorial map M belonging to the edge e to be flipped. The edge flip is performed in three steps. Algorithms 1, 2, and 3 are performed in this order. In particular Algorithms 2 and 3 use the variables $d_f, d_g, \ldots$ defined in Algorithm 1. We use the notation of Section 3.1. We recall here that F denotes the function that maps each edge of the combinatorial map M to its cross-ratio and $A = (\delta, a_1, a_2, a_3)$ is the anchor.

Algorithm 1 performs operations in the combinatorial map M (Figure 17), and is implemented using the CGAL package [5].

$d_f \leftarrow \beta_1(d_e)$;
$d_g \leftarrow \beta_1(d_f)$;
$d'_e \leftarrow \beta_2(d_e)$;
$d_h \leftarrow \beta_1(d'_e)$;
$d_k \leftarrow \beta_1(d_h)$;
$\beta_1(d_f), \beta_1(d_e), \beta_1(d_k) \leftarrow d_e, d_k, d_f$;
$\beta_1(d_g), \beta_1(d_h), \beta_1(d'_e) \leftarrow d_h, d'_e, d_g$;
$\beta_0(d_e), \beta_0(d_k), \beta_0(d_f) \leftarrow d_f, d_e, d_k$;
$\beta_0(d_h), \beta_0(d'_e), \beta_0(d_g) \leftarrow d_g, d_h, d'_e$;

Algorithm 1: Flipping the edge containing a dart d_e in a combinatorial map.

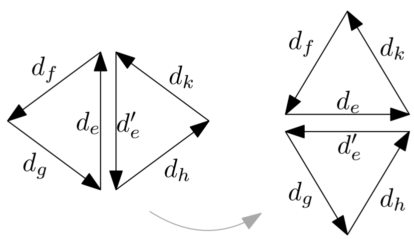


Figure 17: Illustration of Algorithm 1

In Algorithms 2 and 3 given a dart d of the combinatorial map M we denote by $[d]$ the edge that contains d. Algorithm 2 computes an anchor for the triangulation $T^\star$ obtained after the flip. Recall that the lift $\widetilde{T^\star}$ of $T^\star$ is precisely the infinite triangulation of $\mathbb{H}$ obtained by flipping the lifts of the edge e (there are infinitely many of them) in the lift $\widetilde{T}$ of T. Before the flip the anchor A represents a face $\widetilde{t}$ of $\widetilde{T}$. We assume that $\widetilde{t}$ is adjacent to a lift $\widetilde{e}$ of the edge e to be flipped (otherwise Algorithm 2 does nothing and the anchor is correctly not modified). Let $e^\star$ be the edge obtained after the flip of e in T and $\widetilde{e^\star}$ be the lift of $e^\star$ obtained after the flip of $\widetilde{e}$ in $\widetilde{T}$. We claim that after the execution of Algorithm 2 the new anchor represents one of the two faces of $\widetilde{T^\star}$ that are incident to $\widetilde{e^\star}$. First observe that such a face, name it $\widetilde{t^\star}$, shares two vertices with $\widetilde{t}$ and that the vertex of $\widetilde{t^\star}$ that is not shared with $\widetilde{t}$ can be computed from the three vertices of $\widetilde{t}$ and the cross-ratio of e in T: this computation is done by a function ϕ that we now define. Let $\Omega \subset \mathbb{C}^4$ be the set of 4-tuples $(x, y, z, r) \in \mathbb{C}^4$ such that x, y, z are pairwise-distinct and $r(z-y) \neq (z-x)$. Then $\phi : \Omega \to \mathbb{C}$ is defined by $\phi(x, y, z, r) = (xr(z-y) + y(x-z))/(r(z-y) + x - z)$ on every $(x, y, z, r) \in \Omega$. The map ϕ is well-defined. Now we briefly explain why ϕ computes this third vertex of $\widetilde{t^\star}$ and why Algorithm 2 always gives to ϕ inputs that are in Ω. Consider an infinite triangulation $\mathcal{T}$ of $\mathbb{H}$ and an edge w of $\mathcal{T}$. Denote by u_1 and u_3 the vertices of w and by u_2 and u_4 the two other vertices of the two faces of $\mathcal{T}$ containing w: assume that u_1, u_2, u_3, u_4 are in counter-clockwise order. A simple computation shows that $(u_1, u_2, u_3, \mathcal{R}_{\mathcal{T}}(w)) \in \Omega$ and $u_4 = \phi(u_1, u_2, u_3, \mathcal{R}_{\mathcal{T}}(w))$. The correctness of Algorithm 2 follows by case analysis.

switch δ **do**
- **case** d_e **do**
  - $\delta \leftarrow d_h$;
  - $a_2 \leftarrow \phi(a_2, a_3, a_1, F([d_e]))$;
- **end**
- **case** d'_e **do**
  - $\delta \leftarrow d_f$;
  - $a_2 \leftarrow \phi(a_2, a_3, a_1, F([d_e]))$;
- **end**
- **case** d_f *or* d_h **do**
  - $a_3 \leftarrow \phi(a_1, a_2, a_3, F([d_e]))$;
- **end**
- **case** d_g *or* d_k **do**
  - $a_3 \leftarrow \phi(a_3, a_1, a_2, F([d_e]))$;
- **end**

end

Algorithm 2: Updating the anchor $A = (\delta, a_1, a_2, a_3)$. The dart δ is modified if $\delta \in \{d_e, d_f, d_g, d'_e, d_h, d_k\}$.

The update of the cross-ratios encoded in the map F is done by Algorithm 3. Algorithm 3 is a straightforward implementation of Lemma 5.

$F([d_f]) \leftarrow 1 - (1 - F([d_f])) \cdot F([d_e])$;
if $\beta_2(d_f) = d_h$ **then**
- $F([d_f]) \leftarrow 1 - (1 - F([d_f])) \cdot F([d_e])$;

else
- $F([d_h]) \leftarrow 1 - (1 - F(d_h])) \cdot F([d_e])$;

end
$F([d_e]) \leftarrow F([d_e])/(F([d_e]) - 1)$;
$F([d_g]) \leftarrow 1 - (1 - F([d_g]))/F([d_e])$;
if $\beta_2(d_g) = d_k$ **then**
- $F([d_g]) \leftarrow 1 - (1 - F([d_g]))/F([d_e])$;

else
- $F([d_k]) \leftarrow 1 - (1 - F([d_k]))/F([d_e])$;

end

Algorithm 3: Updating the cross-ratios.

C Details for solving arithmetic issues (Section 4)

C.1 Experiment on algebraic numbers (Section 4.1)

Algorithm 4 updates the cross-ratios through the sequence of 5 flips described in Section 4.1. It is is a straightforward implementation of Lemma 5.

Input: The cross-ratios $R_0, \dots, R_4$
for $k = 0, \dots, 4$ **do**
  $R_k \leftarrow R_k/(R_k - 1)$;
  if $k \geq 1$ **then**
    $R_{k-1} \leftarrow 1 - (1 - R_{k-1})/R_k$;
  end
  if $k \leq 3$ **then**
    $R_{k+1} \leftarrow 1 - (1 - R_{k+1})/R_k$;
  end
end

Algorithm 4: Updating $R_0, \dots, R_4$ along the sequence of 5 flips.

C.2 Approximation algorithm (Section 4.2)

This section gives additional details on the construction of the rational admissible 4-tuple shown in the proof of Theorem 2 in Section 4.2.

Definition 6. Let $z_0, z_1, z_2, z_3 \in \mathbb{D} \setminus \{0_{\mathbb{C}}\}$ and $\varepsilon > 0$. We say that (z_0, z_1, z_2, z_3) is *ε-valid* if for any $k \in \{0, 1, 2, 3\}$ and $1 \leq l < m \leq 3$ the following properties are satisfied:

- $\arg z_0 = 0$
- $0_{\mathbb{C}} \notin B(z_k, \varepsilon)$
- $\forall x \in B(z_l, \varepsilon), \forall y \in B(z_m, \varepsilon), 0 < \arg x < \arg y < \pi$.

Now let $\mu > 0$. If moreover $|\mathcal{A}(G[-z_0, z_0, z_1, z_2, z_3]) - 2\pi| < \mu$ then we call the tuple (z_0, z_1, z_2, z_3) *(ε, μ)-admissible.*

In what follows we consider some $\varepsilon, \mu > 0$ and a rational (ε, μ)-admissible 4-tuple $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$. Algorithm 5 returns a rational admissible 4-tuple $(\mathbf{z'_0}, \mathbf{z'_1}, \mathbf{z'_2}, \mathbf{z'_3})$ such that $\mathbf{z'_k} \in B(\mathbf{z_k}, \varepsilon)$ for every $k \in \{0, 1, 2, 3\}$. We prove the correctness of the algorithm in Proposition 8 under certain assumptions on ε, μ, and on the input 4-tuple. Before describing the algorithm we state a preliminary Lemma.

Lemma 7. *At least one of the 2 triangles $G[-\mathbf{z_0}, \mathbf{z_2}, \mathbf{z_3}]$ and $G[\mathbf{z_0}, \mathbf{z_2}, \mathbf{z_1}]$ has a hyperbolic area bigger than $\frac{\pi}{2} - \frac{\mu}{2}$.*

Proof. This is clear since $G[-\mathbf{z_0}, \mathbf{z_0}, \mathbf{z_2}]$ is a triangle so its hyperbolic area is at most π. □

Proposition 8 (Correctness of Algorithm 5)**.** *Assume $\varepsilon \in]0, 1[$ and $\mu \in]0, \pi/6[$. We introduce the following parameter:*

$$R = \max_{0 \leq k \leq 3} d(0_{\mathbb{C}}, \mathbf{z_k}).$$

If the following assumption is satisfied:

$$\varepsilon > 12\mu e^{6R} \tag{2}$$

then Algorithm 5 is well-defined and correct.

Input : Reals $\varepsilon, \mu > 0$ and a rational (ε,μ)-admissible 4-tuple $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$
Output: A rational admissible 4-tuple $(\mathbf{z'_0}, \mathbf{z'_1}, \mathbf{z'_2}, \mathbf{z'_3})$ s.t. $\mathbf{z'_k} \in B(\mathbf{z_k}, \varepsilon)$, $\forall k$
if $\mathcal{A}(G[-\mathbf{z_0}, \mathbf{z_2}, \mathbf{z_3}]) > \frac{\pi}{2} - \frac{\mu}{2}$ **then**
  $\mathbf{f} : z \mapsto \frac{z+\mathbf{z_0}}{\mathbf{z_0}z+1}$;
  $P_0 \leftarrow \mathrm{Im}\left[(1 - \mathbf{f}(\mathbf{z_0})\overline{\mathbf{f}(\mathbf{z_1})})(1 - \mathbf{f}(\mathbf{z_1})\overline{\mathbf{f}(\mathbf{z_2})})\right]$;
  $P_1 \leftarrow \mathrm{Im}\left[(1 - \mathbf{f}(\mathbf{z_0})\overline{\mathbf{f}(\mathbf{z_1})})(1 - \mathbf{f}(\mathbf{z_1})\overline{\mathbf{f}(\mathbf{z_2})})(1 - \mathbf{f}(\mathbf{z_2})\overline{\mathbf{f}(\mathbf{z_3})})\right]$;
  $\lambda \leftarrow P_0/(P_0 - P_1)$;
  $V \leftarrow \lambda \mathbf{f}(\mathbf{z_3})$;
  return $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{f}^{-1}(V))$;
else
  $\mathbf{f} : z \mapsto \frac{z-\mathbf{z_0}}{-\mathbf{z_0}z+1}$;
  $P_0 \leftarrow \mathrm{Im}\left[(1 - \mathbf{f}(\mathbf{z_3})\overline{\mathbf{f}(-\mathbf{z_0})})(1 - \mathbf{f}(\mathbf{z_2})\overline{\mathbf{f}(\mathbf{z_3})})\right]$;
  $P_1 \leftarrow \mathrm{Im}\left[(1 - \mathbf{f}(\mathbf{z_3})\overline{\mathbf{f}(-\mathbf{z_0})})(1 - \mathbf{f}(\mathbf{z_2})\overline{\mathbf{f}(\mathbf{z_3})})(1 - \mathbf{f}(\mathbf{z_1})\overline{\mathbf{f}(\mathbf{z_2})})\right]$;
  $\lambda \leftarrow P_0/(P_0 - P_1)$;
  $V \leftarrow \lambda \mathbf{f}(\mathbf{z_1})$;
  return $(\mathbf{z_0}, \mathbf{f}^{-1}(V), \mathbf{z_2}, \mathbf{z_3})$;
end

Algorithm 5: The approximation algorithm.

Proof. We only consider the case $\mathcal{A}(G[-\mathbf{z_0}, \mathbf{z_2}, \mathbf{z_3}]) > \frac{\pi}{2} - \frac{\mu}{2}$ as the same arguments hold for the other case after application of Lemma 7. Using the notations introduced in the algorithm we bound P_0 and P_1 and then deduce that λ is well-defined. Only then we prove that $V \in B(\mathbf{f}(\mathbf{z_3}), \varepsilon)$. We have $P_0 = \mathrm{Im}\,[Z_0]$ and $P_1 = \mathrm{Im}\,[Z_1]$ with

$$\begin{aligned} Z_0 &= (1 - \mathbf{f}(\mathbf{z_0})\overline{\mathbf{f}(\mathbf{z_1})})(1 - \mathbf{f}(\mathbf{z_1})\overline{\mathbf{f}(\mathbf{z_2})}) \\ Z_1 &= (1 - \mathbf{f}(\mathbf{z_0})\overline{\mathbf{f}(\mathbf{z_1})})(1 - \mathbf{f}(\mathbf{z_1})\overline{\mathbf{f}(\mathbf{z_2})})(1 - \mathbf{f}(\mathbf{z_2})\overline{\mathbf{f}(\mathbf{z_3})}). \end{aligned}$$

We already proved that $\arg Z_1 = \frac{1}{2}\mathcal{A}(G[-\mathbf{z_0}, \mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}])$. Also we have $|\arg Z_1 - \pi| < \frac{\mu}{2}$ since $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$ is (ε,μ)-admissible. Since $\mu < \pi$, $|P_1| < \sin(\frac{\mu}{2}) < \frac{\mu}{2}$. In addition,

$$\begin{aligned} \arg Z_0 &= \frac{1}{2}\mathcal{A}(G[-\mathbf{z_0}, \mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}]) \\ &= \frac{1}{2}(\mathcal{A}(G[-\mathbf{z_0}, \mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}]) - \mathcal{A}(G[-\mathbf{z_0}, \mathbf{z_2}, \mathbf{z_3}])). \end{aligned}$$

By Definition 6, and since $\frac{\pi}{2} - \frac{\mu}{2} < \mathcal{A}(G[-\mathbf{z_0}, \mathbf{z_2}, \mathbf{z_3}]) < \pi$,

$$\frac{\pi}{2} - \frac{\mu}{2} < \arg Z_0 < \frac{3\pi}{4} + \frac{3\mu}{4}.$$

Moreover for every $k \in \{0,1,2,3\}$ one has $d(0_{\mathbb{C}}, \mathbf{f}(\mathbf{z_k})) \leq d(0_{\mathbb{C}}, \mathbf{f}(0_{\mathbb{C}})) + d(\mathbf{f}(0_{\mathbb{C}}), \mathbf{f}(\mathbf{z_k})) = d(0_{\mathbb{C}}, \mathbf{z_0}) + d(0_{\mathbb{C}}, \mathbf{z_k}) \leq 2R$. Writing the Euclidean norm in term of the hyperbolic distance d (in the Poincaré disk model) the latter becomes $|\mathbf{f}(\mathbf{z_k})| \leq \tanh(R)$. That proves $1 \geq |Z_0| \geq (1 - \tanh(R))^2$. Together with the bound on $\arg Z_0$ we obtain a bound on P_0:

$$\begin{aligned} 1 > P_0 &> \left(1 - \tanh(R)^2\right)^2 \cdot \sin\left(\frac{3\pi}{4} + \frac{3\mu}{4}\right) \\ &\geq e^{-4R} \cdot \sin\left(\frac{3\pi}{4} + \frac{3\mu}{4}\right) \\ &> \frac{1}{3}e^{-4R}, \end{aligned}$$

since $\mu < \frac{\pi}{6}$ and $\sin(\frac{7\pi}{8}) > \frac{1}{3}$. From the bounds on P_0 and P_1, and using Assumption (2) we deduce that $P_0 > P_1$, so λ is well defined. Also, we get that

$$1 < \lambda < \frac{1}{1 - \frac{3}{2}\mu e^{4R}}.$$

It remains to prove that $V \in B(\mathbf{f}(\mathbf{z_3}), \varepsilon)$. For the sake of clarity we denote by D the hyperbolic distance $d(0_{\mathbb{C}}, \mathbf{f}(\mathbf{z_3}))$. It is enough to show the following:

$$\lambda \tanh\left(\frac{D}{2}\right) < \tanh\left(\frac{D+\varepsilon}{2}\right).$$

We first observe that $x \mapsto \tanh(x) - x/2$ is increasing on $[0, 1/2[$ and maps 0 to 0. Thus, since $\varepsilon/2 \in]0, 1/2[$, $\tanh(\varepsilon/2) \geq \varepsilon/4$. From that and by applying Assumption (2) we obtain

$$\mu e^{4R} < \tanh(\varepsilon/2)e^{-2R}.$$

We conclude with the following implications:

$$\begin{aligned}
& 3\mu e^{4R} < \tanh(\varepsilon/2)e^{-2R} \\
\implies\ & \frac{3}{2}\mu e^{4R} < \frac{1}{2}\tanh(\varepsilon/2)(1 - \tanh(R)^2) \\
\implies\ & \frac{\tanh(D/2) + \tanh(D/2)^2\tanh(\varepsilon/2)}{\tanh(D/2) + \tanh(\varepsilon/2)} < 1 - \frac{3}{2}\mu e^{4R} \\
\implies\ & \frac{\tanh(\frac{D}{2})}{\tanh(\frac{D+\varepsilon}{2})} < 1 - \frac{3}{2}\mu e^{4R} \\
\implies\ & \lambda \tanh\left(\frac{D}{2}\right) < \tanh\left(\frac{D+\varepsilon}{2}\right).
\end{aligned}$$

That concludes the proof. □

D Details for the generation of input (Section 5)

D.1 Generating an initial rational 4-tuple (step 1)

We follow the construction of 4-tuples [1, Section 3] recalled in Section 2.4 but only compute rational approximations of the algebraic numbers involved. Then we apply Algorithm 5. The generation process described below has the following advantage: the size of the integers involved in the output 4-tuple $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$ are controlled by the parameter `N` defined below. Providing small fractions is important as the process described in this section is performed at the very beginning of the experiments.

We first construct for $k \in \{1, 2, 3\}$ the real and imaginary parts x_k and y_k of a complex number z_k; they are represented as `float` numbers in python and constructed in $[-1, 1]$ using the `random` method of the `random` package: x_k `= 2*random.random()-1`. That simulates a uniform distribution. The construction fails if one of the points $\{z_1, z_2, z_3\}$ lies outside $\mathbb{D}$, or if the condition $a + b + c < 0$ with a, b, and c as defined in Section 2.4 is not satisfied.

Then we construct the `float` numbers x_0 and y_0 representing the real and imaginary parts of z_0 as described in Section 2.4. From that we construct for each $k \in \{0, 1, 2, 3\}$ the real and imaginary parts $\mathbf{x_k}$ and $\mathbf{y_k}$ of $\mathbf{z_k}$ as rational approximations of x_k and y_k: we set $\mathbf{x_k}$ = `int(N *` x_k`) / N`, where the parameter $\mathtt{N} \in \mathbb{N} \setminus \{0_{\mathbb{N}}\}$ determines the quality of the approximation and `int` is native in python. We arbitrarily chose $\mathtt{N} = 100$ in each computation. The construction fails if $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$ is not valid.

The rational 4-tuple $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$ is not necessarily admissible. However, by the construction method, it can be seen as a rational approximation of some admissible 4-tuple and it satisfies the hypothesis of Proposition 8. We can thus compute an admissible 4-tuple using Algorithm 5 (see also Section 4.2).

To simplify notation, we still denote the rational admissible 4-tuple that we obtain by $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$.

D.2 Generating points in an admissible symmetric octagon (step 2)

Consider the rational admissible 4-tuple $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$ obtained after step 1. In this section, we describe our method to construct a point $\mathbf{p} \in \mathbb{Q} + i\mathbb{Q}$ in the closure of the admissible symmetric octagon $\mathcal{P}[\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}]$, simulating a uniform distribution with respect to the hyperbolic metric.

The method uses inexact computation so it can fail especially if the Euclidean area of $\mathcal{P}[\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}]$ is close to 0. This is also why we do not generate such points in the admissible loosely-symmetric octagons resulting from the twists in step 4.

We start by dividing $\mathcal{P}[\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}]$ into 6 hyperbolic triangles $\Delta_1, \dots, \Delta_6$. We compute the hyperbolic area of each triangle as a C++ (native) `double` number using Equality (1). Then we choose the triangle Δ_k that will contain $\mathbf{p}$ with probability $\dfrac{\mathcal{A}(\Delta_k)}{\sum_{l=1}^{6} \mathcal{A}(\Delta_l)}, k \in \{1, \dots, 6\}$. By a translation we can assume that $0_{\mathbb{C}}$ is a vertex of Δ_k. We construct as `double` numbers the real and imaginary parts of a complex number $p \in \mathbb{D}$, simulating a uniform choice within the closure of Δ_k. To construct $\mathbf{p}$ from p we cast the real and imaginary parts of p into `CGAL::Gmpq` numbers [11]. Then we check using Lemma 9 whether $\mathbf{p}$ actually belongs to the closure of Δ_k; if this is the case we return $\mathbf{p}$.

Lemma 9. *Consider pairwise-distinct points $z_1, z_2, z_3 \in \mathbb{D}$ and the oriented geodesic l containing z_1 and z_2, oriented from z_1 to z_2. The oriented geodesic l separates $\mathbb{D}$ into 2 open regions and we consider the region R on the left of l. We define $\tau : \mathbb{D} \to \mathbb{D}$ by*

$$\tau(z) = \frac{z - z_1}{1 - \overline{z_1} z}$$

for every $z \in \mathbb{D}$. Then $z_3 \in R$ if and only if

$$Im \left[\frac{\tau(z_3)}{\tau(z_2)} \right] > 0$$

and the above expression is an equality if and only if $z_3 \in l$.

Proof. The result follows from observing that τ is an orientation preserving isometry of $\mathbb{D}$ sending z_1 to $0_{\mathbb{C}}$. □

D.3 Constructing the data structure (step 4)

After step 1, step 2, and step 3, we are given a rational admissible 4-tuple $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$, points $(\mathbf{p_1}, \dots, \mathbf{p_{n_p}}) \in (\mathbb{Q} + i\mathbb{Q})^{n_p}$ lying in the closure of $\mathcal{P}[\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}]$ and a sequence $t_1, \dots, t_m$ of twists. The rational admissible 4-tuple $(\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3})$ defines the surface $\mathcal{S}$.

Applying the procedure in Section 5.1 we construct the vertices $\mathbf{z_0}', \dots, \mathbf{z_7}'$ of the rational admissible loosely-symmetric octagon resulting from twisting $\mathcal{P}[\mathbf{z_0}, \mathbf{z_1}, \mathbf{z_2}, \mathbf{z_3}]$ according to the sequence $t_1, \dots, t_m$. From $\mathbf{p_1}, \dots, \mathbf{p_{n_p}}$ we also construct new points $(\mathbf{p_1}', \dots, \mathbf{p_{n_p}}') \in (\mathbb{Q} + i\mathbb{Q})^{n_p}$ lying in the closure of $G[\mathbf{z_0}', \dots, \mathbf{z_7}']$. The latter is done twist by twist : when performing a twist on an octagon O we obtain a new octagon O' and we update the list of points so that they lie in the closure of the octagon O' after the twist. When a point p is replaced by a new point p' we make sure that p and p' are two lifts of the *same* point on the surface represented by O and O'. In the end we also compute the orientation preserving isometries $(\tau'_k)_{0 \le k \le 7}$ pairing the opposite sides of $G[\mathbf{z_0}', \dots, \mathbf{z_7}']$ (see Section 5.1).

Recursively, we construct a sequence $T_0, \ldots, T_{n_p}$ of triangulations of the octagon $G[\mathbf{z_0}', \ldots, \mathbf{z_7}']$. We start with the triangulation T_0 whose edges are the eight sides of $G[\mathbf{z_0}', \ldots, \mathbf{z_7}']$, and the five geodesic segments between $\mathbf{z_0}'$ and $\mathbf{z_2}', \mathbf{z_3}', \mathbf{z_4}', \mathbf{z_5}', \mathbf{z_6}'$. The triangulation T_0 is represented by a combinatorial map M_0 and a map P_0 associating to each vertex v of M_0 its position $P_0(v)$ in $\mathbb{D}$. For $k \in \{1, \ldots, n_p\}$ the triangulation T_k is obtained from T_{k-1} by splitting the triangle containing $\mathbf{p_k}'$ into three triangles. In the end we get a triangulation T_{n_p} together with its combinatorial map M_{n_p} and the map P_{n_p} giving the position of each vertex in $\mathbb{D}$. By identifying the edges of T_{n_p} that are the opposite sides of $G[z_0', \ldots, z_7']$ we obtain a triangulation T of $\mathcal{S}$.

We finally construct the triple (M, F, A) representing the triangulation T from the combinatorial map M_{n_p} and the map P_{n_p} (see Section 3.1). The combinatorial map M is easily obtained from M_{n_p} by setting $\beta_2(d) = d'$ and $\beta_2(d') = d$ (see Figure 2) for any 2 distinct darts d and d' of M_{n_p} supporting 2 edges corresponding to opposite sides of $G[z_0', \ldots, z_7']$. The anchor $A = (\delta, a_1, a_2, a_3)$ is defined by choosing δ in M_{n_p}: the dart δ belongs to a face (v_1, v_2, v_3) of M_{n_p} and is based at v_1; we set $a_k = P_{n_p}(v_k)$ for every $k \in \{1, 2, 3\}$. Now consider some edge e of M. There are 2 cases. If e results from an edge of M_{n_p} that was not a side of $G[\mathbf{z_0}', \ldots, \mathbf{z_7}']$ then computing its cross-ratio in T_{n_p} or equivalently in T is straightforward. If e results from the identification of 2 edges e_1 and e_2 of M_{n_p} then we compute $F(e)$ as follows. We denote the vertices of e_1 in M_{n_p} by a, b and the vertices of e_2 by c, d such that $P_{n_p}(a), P_{n_p}(b), P_{n_p}(c), P_{n_p}(d)$ appear in counter-clockwise order on the boundary of $G[\mathbf{z_0}', \ldots, \mathbf{z_7}']$: when identifying e_1 and e_2 to construct M from M_{n_p} the vertex a is identified with d, and the vertex b is identified with c. We consider $k \in \{0, \ldots, 7\}$ such that orientation preserving isometry τ_k' maps $P_{n_p}(d)$ to $P_{n_p}(a)$ and maps $P_{n_p}(c)$ to $P_{n_p}(b)$. The edge e_1 belongs to a unique face f_1 of M_{n_p} and we denote the vertex of f_1 that is neither a nor b by u_1. Similarly, the edge e_2 belongs to a unique face f_2 of M_{n_p} and we denote the vertex of f_2 that is neither c nor d by u_2. Then $F(e) = [P_{n_p}(a), \tau_k(P_{n_p}(u_2)), P_{n_p}(b), P_{n_p}(u_1)]$.

E Computation of the approximation of the diameter (Section 7)

Consider a rational admissible loosely-symmetric octagon O given by the 16 rational numbers representing the real and imaginary parts of its 8 vertices. The hyperbolic diameter of O is the maximum of the hyperbolic distances between any two of its vertices. For every pair z_1, z_2 of two such distinct vertices we compute an approximation represented by a C++ `double` D of the hyperbolic distance between z_1 and z_2. The maximum (obtained using `std::max`) of these $\binom{8}{2}$ values is an approximation of the hyperbolic diameter of O.

We compute every such D as follows. The isometry $f : z \mapsto (z - z_1)/(1 - \bar{z_1} z)$ maps $\mathbb{Q} \cap \mathbb{D}$ to a subset of $\mathbb{Q}$ and maps z_1 to 0. We compute the exact rational value r_2 of the square of the modulus of $f(z_2)$. Then we convert r_2 to a `CORE::Expr` r_2' and set $x = (1 + \texttt{CGAL::sqrt}(r_2'))/(1 - \texttt{CGAL::sqrt}(r_2'))$. The number D is an approximation of the natural logarithm $\ln(x)$ of x obtained by first casting x to a string s. The string s contains the string representation s_1 of the lower integer rounding k of $\log_{10}(x)$. Also s contains the string representation s_2 of an approximation of $x \cdot 10^{-k}$. The value of D is calculated as $\texttt{std::stoi}(s_1) * \texttt{std::log}(10) + \texttt{std::log}(\texttt{std::stod}(s_2))$.