



**HAL**  
open science

# Computing a Dirichlet domain for a hyperbolic surface

Vincent Despré, Benedikt Kolbe, Hugo Parlier, Monique Teillaud

► **To cite this version:**

Vincent Despré, Benedikt Kolbe, Hugo Parlier, Monique Teillaud. Computing a Dirichlet domain for a hyperbolic surface. 2022. hal-03881015

**HAL Id: hal-03881015**

**<https://hal.science/hal-03881015>**

Preprint submitted on 1 Dec 2022

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

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

# Computing a Dirichlet domain for a hyperbolic surface\*

Vincent Despré<sup>1</sup>, Benedikt Kolbe<sup>2</sup>, Hugo Parlier<sup>3</sup>, and Monique Teillaud<sup>4</sup>

<sup>1</sup>Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France, <http://vdespre.free.fr/>

<sup>2</sup>Hausdorff Center for Mathematics, University of Bonn, Germany , <https://hyperbolictilings.wordpress.com/>

<sup>3</sup>Department of Mathematics, University of Luxembourg , <https://math.uni.lu/parlier/>

<sup>4</sup>Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France , <https://members.loria.fr/monique.teillaud/>

December 1, 2022

## Abstract

The goal of this paper is to exhibit and analyze an algorithm that takes a given closed orientable hyperbolic surface and outputs an explicit Dirichlet domain. The input is a fundamental polygon with side pairings. While grounded in topological considerations, the algorithm makes key use of the geometry of the surface. We introduce data structures that reflect this interplay between geometry and topology and show that the algorithm finishes in polynomial time, in terms of the initial perimeter length and the genus of the surface.

## 1 Introduction and motivation

Hyperbolic surfaces and their moduli spaces play an ubiquitous role in mathematics, namely, through relationships with other areas including Riemannian geometry, number theory, geometric group theory and mathematical physics. Algorithms for surface groups, as combinatorial or topological objects, have a rich history dating back to Dehn. Recently, in part motivated by applications in other sciences [1, 16], there has been a push to understand hyperbolic structures on surfaces from the point of view of computational geometry.

---

\*This work was partially supported by grant ANR-17-CE40-0033 of the French National Research Agency ANR (project SoS) and INTER/ANR/16/11554412/SoS of the Luxembourg National Research fund FNR <https://SoS.loria.fr/>.

31 Dealing with hyperbolic surfaces necessarily involves describing them — or  
 32 even visualizing them — meaningfully. A fundamental domain (in the hyper-  
 33 bolic plane) with a side pairing is one way to determine a hyperbolic metric  
 34 on the surface. Lengths of curves in a pants decomposition and their associ-  
 35 ated pasting parameters (so-called Fenchel-Nielsen coordinates) are another. No  
 36 matter which construction or parameter set used, it is always interesting to know  
 37 to which extent two different constructions output the “same” surface, where  
 38 “same” can take different meanings. However, these representations, either by  
 39 a fundamental domain or a set of Fenchel-Nielsen coordinates, are not unique,  
 40 and determining a canonical representation is challenging for either option. In  
 41 this paper, we tackle this question for fundamental domains, by computing a  
 42 so-called Dirichlet domain.

43 Roughly speaking, a Dirichlet domain of a hyperbolic surface is a funda-  
 44 mental polygon in the hyperbolic plane, with a special point where distances to  
 45 that point in the polygon correspond to distances on the surface. Another way  
 46 of thinking of them is that it is a Voronoi cell associated to a lift of a single  
 47 point of the surface to its universal cover  $\mathbb{H}^2$ . A more precise definition is given  
 48 in the next section. Note that for hyperbolic surfaces any given surface has  
 49 infinitely many Dirichlet domains up to isometry. This is in strong contrast to,  
 50 for example, flat tori. Nonetheless, when describing a surface via fundamental  
 51 domains, the prize for the most relevant geometric domain undoubtedly goes to  
 52 Dirichlet domains because they visualize the distance function for a given point.  
 53 As far as we know, there is only one algorithm in the literature that computes a  
 54 Dirichlet domain for a hyperbolic surface and a given point [17]. Unfortunately,  
 55 the run-time of the algorithm is not studied and an analysis seems complicated.

56 The contribution of this paper is an algorithm that computes a Dirichlet  
 57 domain efficiently, and its analysis. The point defining the domain is not given  
 58 as input, but it is part of the output. The Dirichlet domain of a given input  
 59 point can then be computed with a complexity that only depends of the genus  
 60 of the surface [10]. Our main result is the following:

61 **Theorem 1.** *Let  $S$  be a closed orientable hyperbolic surface of genus  $g$  given*  
 62 *by a fundamental polygon of perimeter  $L$  and side pairings. A Dirichlet domain*  
 63 *for  $S$  can be computed in  $O((g^2L)^{6g-4})$  time.*

64 A key ingredient is the use of *Delaunay triangulations* on hyperbolic surfaces,  
 65 an area of research that has recently gained traction, both from an experimental  
 66 and a theoretical perspective [3, 7, 4, 14, 8, 12]. Recently, it has been shown  
 67 that the well-known flip algorithm that computes the Delaunay triangulation  
 68 of a set of points in the Euclidean plane  $\mathbb{E}^2$  also works on a hyperbolic surface;  
 69 the complexity result announced in Theorem 1 crucially depends on the only  
 70 known upper bound on the complexity of this Delaunay flip algorithm [11].  
 71 The algorithm subsumes the real RAM model. Studying the algebraic numbers  
 72 involved in the computations goes beyond the scope of this paper.

73 The paper is structured as follows: In Section 3, we give an overview of the  
 74 algorithm and we present the data structure. Sections 4 and 5 explain in detail  
 75 the main two steps of the algorithm, which output a geometric triangulation of  
 76 the surface having only one vertex. Finally, Section 6 builds on the literature  
 77 and concludes the proof of Theorem 1 with the last step of the algorithm.

## 2 Preliminaries

We begin by recalling a collection of facts and setting notations, and we refer to [2, 5, 13] for details. The surfaces studied in this paper are assumed to be closed, orientable, and of genus  $g \geq 2$ . We begin with a topological surface and endow it with a hyperbolic metric to obtain a hyperbolic surface, generally denoted by  $S$ . A hyperbolic surface is locally isometric to its universal covering space, the hyperbolic plane  $\mathbb{H}^2$ . Such surfaces can always be obtained by considering the quotient of  $\mathbb{H}^2$  under the action of  $\Gamma$ , a discrete subgroup of isometries of  $\mathbb{H}^2$  isomorphic to the fundamental group  $\pi_1(S)$ .

Let  $S := \mathbb{H}^2/\Gamma$  be a hyperbolic surface of genus  $g$  and fundamental group  $\Gamma$ . The projection map is denoted as  $\rho : \mathbb{H}^2 \rightarrow S = \mathbb{H}^2/\Gamma$ . We denote by  $\tilde{x} \in \rho^{-1}(x)$  one of the lifts, to  $\mathbb{H}^2$ , of an object  $x$  on  $S$ . More generally, objects in  $\mathbb{H}^2$  are denoted with  $\tilde{\cdot}$ .

A *fundamental domain*  $\mathcal{F}$  for the action of  $\Gamma$  is defined as a closed domain, i.e.,  $\overline{\text{int}(\mathcal{F})} = \mathcal{F}$ , such that  $\Gamma\mathcal{F} = \mathbb{H}^2$  and the interiors of different copies of  $\mathcal{F}$  under  $\Gamma$  are disjoint.

For a point  $\tilde{x} \in \mathbb{H}^2$ , the *Dirichlet domain*  $\mathcal{D}_{\tilde{x}}$  is defined as the Voronoi cell containing  $\tilde{x}$ , of the Voronoi diagram associated to the point set  $\Gamma\tilde{x}$ . In other words,

$$\mathcal{D}_{\tilde{x}} = \{ \tilde{y} \in \mathbb{H}^2 \mid d_{\mathbb{H}^2}(\tilde{x}, \tilde{y}) \leq d_{\mathbb{H}^2}(\tilde{x}, \Gamma\tilde{y}) \} = \{ \tilde{y} \in \mathbb{H}^2 \mid d_{\mathbb{H}^2}(\tilde{x}, \tilde{y}) \leq d_{\mathbb{H}^2}(\tilde{x}, \tilde{y}) \},$$

where the equality is true since  $\Gamma$  acts as isometries w.r.t.  $d_{\mathbb{H}^2}$ . The Dirichlet domain is a compact convex fundamental domain for  $\Gamma$  with finitely many geodesic sides [2, §9.4] and is generally considered a canonical choice of fundamental domain. A property of Dirichlet domains, of interest for the conception of algorithms, is that, by the triangle inequality,

$$\text{diam}(\mathcal{D}_{\tilde{x}}) \leq 2 \text{diam}(S) \leq 2 \text{diam}(\mathcal{D}_{\tilde{x}}),$$

where  $\text{diam}(\cdot)$  denotes the diameter.

### 2.1 Curves, paths, and loops

Recall that a closed curve is the image of  $\mathbb{S}^1$  under a continuous map; a curve is non-trivial (or essential) if it is not freely homotopic to a point. Similarly, a path is a continuous image of the interval  $[0, 1]$ , and the images of 0 and 1 are referred to as its endpoints. A loop is a path whose endpoints are equal; this endpoint is referred to as its basepoint.

For a closed curve or loop  $c$ , we will denote by  $[c]$  its free homotopy class, and, if  $c$  is based in a point  $p$ , by  $[c]_p$  its homotopy class of loops based in  $p$ . For a path  $c$  between points  $p$  and  $q$ , we denote by  $[c]_{p,q}$  the homotopy class of the path with fixed endpoints. We will readily make use of the fact that if  $c$  is closed non-trivial curve on a hyperbolic surface, then in  $[c]$  there is a unique closed geodesic. Similarly, if  $c$  is a loop based in  $p$ , in  $[c]_p$  there is a unique closed geodesic loop, and if  $c$  is a path between  $p$  and  $q$ , in  $[c]_{p,q}$  there is a unique geodesic path. If  $c$  is a simple closed curve then the closed geodesic in  $[c]$  is also simple, but this is no longer necessarily the case for loops or paths with basepoints.

119 The intersection number  $i(c, c')$  between homotopy classes of curves  $c$  and  
 120  $c'$  is defined as the minimal intersection among its representatives. Note that  
 121 closed geodesics on a hyperbolic surface always intersect minimally. The situa-  
 122 tion for paths is slightly different. The unique geodesic representatives of paths  
 123 (with fixed end points) might not intersect minimally. This subtlety plays a key  
 124 technical role in our story.

## 125 2.2 Fundamental polygon

126 Let  $S$  be a (closed) hyperbolic surface of genus  $g$  and fundamental group  $\Gamma$ . A  
 127 polygon  $P \subset \mathbb{H}^2$  (i.e., a circular sequence of geodesic edges) bounding a fun-  
 128 damental domain for  $\Gamma$  (as defined in the introduction) is called a fundamental  
 129 polygon. Poincaré's theorem implies that  $\Gamma$  is generated by the side pairings on  
 130  $P$  [2, §9.8]. The edges and vertices of  $P$  project to a graph  $G_P$  on  $S$ ; the region  
 131 enclosed by  $P$  projects to the unique face of  $G_P$ .

132 The numbers  $n_G$  of vertices and  $m_G$  of edges of  $G_P$  satisfy Euler's relation  
 133  $n_G - m_G + 1 = 2 - 2g$ , as there is only one face. It follows that if  $G_P$  only has one  
 134 vertex, then that vertex is incident to the  $m_G = 2g$  edges, which are actually  
 135 all loops. The number of vertices is maximal when they all have degree 3 (then  
 136 there are no loops); in this case  $3n_G = 2m_G$ , so,  $m_G = 6g - 3$  and  $n_G = 4g - 2$ .  
 137 More generally, the number  $2m_G$  of edges and vertices of  $P$  lies between the two  
 138 extreme cases:  $4g \leq 2m_G \leq 12g - 6$ . When  $2m_G < 12g - 6$ , some vertices of  $P$   
 139 project to the same vertex of  $G_P$ , i.e., they belong to the same orbit under  $\Gamma$ .  
 140  $G_P$  has a loop for each edge whose vertices are in the same orbit; then the  
 141 projected point on  $S$  is incident to that loop twice.

## 142 3 Algorithm overview

143 Let  $S$  be a (closed) hyperbolic surface of genus  $g$  and fundamental group  $\Gamma$ .

144 We propose the algorithm sketched below to compute a Dirichlet fundamen-  
 145 tal domain of  $S$ . The output of Step 1 will be denoted with primes; it will be  
 146 used as input for Step 2, whose output will be denoted with double primes.

- 147 1. Construct a system  $\beta'_0, \dots, \beta'_{2g-1}$  of simple topological loops based at the  
 148 same point  $b'$  that cuts  $S$  into a disk (Section 4).
- 149 2. Find a point  $b''$  so that the system of geodesic loops based at  $b''$ , conjugate  
 150 to the ones computed in Step (1), is embedded (Section 5).
- 151 3. Construct the Dirichlet domain of a lift  $\tilde{b}''$  of  $b''$  (Section 6).

152 Obviously, the complexity of the algorithm heavily depends on the data  
 153 structure used to store the objects involved in the constructions. As the algo-  
 154 rithm actually operates in the universal covering space  $\mathbb{H}^2$  of  $S$ , it is natural to  
 155 present the data structure in  $\mathbb{H}^2$ . We assume that, as input, we are given a fun-  
 156 damental polygon  $\Pi \subset \mathbb{H}^2$  for  $\Gamma$ , together with side pairings, as in Section 2.2.  
 157 The data structure described below is actually equivalent to a combinatorial  
 158 map [15, Section 3.3] on  $S$ , enriched with geometric information. In particular,  
 159 for each vertex  $x$  of  $G_\Pi$  (the projection of  $\Pi$  onto  $S$ , as in Section 2.2), the  
 160 sequence of edges around  $x$  is ordered (edges that correspond to a loop appear  
 161 twice).

162 **Description of the input.** Let a representative  $\tilde{e}_i, i = 0, \dots, m-1$  be chosen  
163 for each couple of paired edges of  $\Pi$  and denote as  $\gamma_0, \dots, \gamma_{m-1}$  the correspond-  
164 ing side pairings in  $\Pi$ : the other edge of the couple is  $\gamma_i^{-1}\tilde{e}_i$ , where  $\gamma_i^{-1}$  is the  
165 inverse of  $\gamma_i$ . We denote the set of the  $2m$  edges of  $\Pi$  as  $E_\Pi$  and the set of its  
166  $2m$  vertices as  $V_\Pi$ . We choose a representative  $\tilde{v}_j, j = 0, \dots, n-1$  for each orbit  
167 of vertices of  $\Pi$ ;  $n$  is the number of vertices of  $G_\Pi$ .

168 Each element of  $\Gamma$  can be represented as a word on the alphabet  $\mathcal{A}_\Gamma =$   
169  $\{\mathbb{1}, \gamma_0, \dots, \gamma_{m-1},$   
170  $\gamma_0^{-1}, \dots, \gamma_{m-1}^{-1}\}$ , where  $\mathbb{1}$  denotes the identity in  $\Gamma$ . Here, letters of  $\mathcal{A}_\Gamma$  and  
171 the corresponding generators in  $\Gamma$  are denoted by the same symbol; this should  
172 not cause any confusion.

173 The data structure is roughly a doubly linked list of edges of  $\Pi$ , which stores  
174 the combinatorial information. Additional information is necessary to store the  
175 geometry (i.e., the positions of the vertices of  $\Pi$  in  $\mathbb{H}^2$ ) and the side pairings.  
176 The data stored for each edge and vertex is constant, so the size of the data  
177 structure is  $O(g)$  (we do not try to shave constants in the  $O()$ ).

178 Concretely, for each edge  $\tilde{x} \in E_\Pi$ , the data structure stores:

- 179 • two pointers  $\text{prev}(\tilde{x})$  and  $\text{next}(\tilde{x})$  that give access to the previous and next  
180 edges in  $\Pi$ , respectively (in counterclockwise order);
- 181 • two pointers  $\text{source}(\tilde{x})$  and  $\text{target}(\tilde{x})$  that give access to the source and  
182 target of  $\tilde{x}$  in  $\Pi$ , respectively (in counterclockwise order); when  $\rho\tilde{x}$  is a  
183 loop in  $G_\Pi$ ,  $\text{source}(\tilde{x})$  and  $\text{target}(\tilde{x})$  lie in the same orbit under  $\Gamma$ ;
- 184 • a pointer to the paired edge  $\text{pair}(\tilde{x})$  in  $\Pi$ ;
- 185 • a letter  $w(\tilde{x}) \in \mathcal{A}_\Gamma$  that encodes the relation between  $\tilde{x}$  and  $\text{pair}(\tilde{x})$ :  
186 
$$w(\tilde{x}) = \begin{cases} \mathbb{1} & \text{if } \tilde{x} = \tilde{e}_i \\ \gamma_i & \text{if } \tilde{x} = \gamma_i^{-1}\tilde{e}_i \text{ for some } i \in \{0, \dots, m-1\}. \end{cases}$$

187 By definition,  $\text{pair}(\tilde{x}) = \begin{cases} \gamma_i^{-1}\tilde{x} & \text{when } w(\tilde{x}) = \mathbb{1} (\tilde{x} = \tilde{e}_i) \\ \gamma_i\tilde{x} & \text{when } w(\tilde{x}) = \gamma_i \end{cases}$ .

188 For each vertex  $\tilde{y} \in V_\Pi$ , the data structure stores:

- 189 •  $\text{point}(\tilde{y})$ , which is the representative point of its orbit:  $\text{point}(\tilde{y}) = \tilde{v}_j$  for  
190 some  $j \in \{0, \dots, n-1\}$ ;
- 191 • a word  $\gamma_{\tilde{y}}$  on  $\mathcal{A}_\Gamma$  (equivalently,  $\gamma_{\tilde{y}} \in \Gamma$ ), which specifies the precise position  
192  $\gamma_{\tilde{y}} \text{point}(\tilde{y})$  of  $\tilde{y}$  in  $\mathbb{H}^2$ .

193 The graph  $G_\Pi$  lifts in the universal covering space  $\mathbb{H}^2$  to the (infinite) graph  
194  $\rho^{-1}G_\Pi = \Gamma\Pi$ . In particular, the sequence of edges of  $\Gamma\Pi$  incident to a given  
195 vertex  $\tilde{v} \in \rho^{-1}v$  is a sequence of lifts of the edges incident to  $v$  in  $G_\Pi$ . Each  
196 of these lifts is the image by an element of  $\Gamma$  of an edge of  $\Pi$  (Figure 1). The  
197 following result is straightforward from the data structure. We still prove it for  
198 completeness.

199 **Lemma 2.** *Let  $e$  be an edge of  $G_\Pi$  and  $v$  a vertex of  $e$ . The sequence of edges*  
200 *of  $G_\Pi$  incident to  $v$  can be found in time  $O(g)$ .*

201 *Proof.* Without loss of generality (this can always be achieved by renaming edges  
202 and vertices of  $\Pi$ )  $\tilde{v} = \text{source}(\tilde{e})$  (as in Figure 1 for  $e = e_0$ ), or  $\tilde{v} = \text{target}(\tilde{e})$ .  
203 Consider the first case. After  $e$ , the next edge incident to  $v$  in counterclockwise  
204 order in  $G_\Pi$  is given in  $\Pi$  by  $\tilde{x} = \text{prev}(\tilde{e})$ , whose vertex  $\text{target}(\tilde{x})$  is  $\tilde{v}$ . The next  
205 edge incident to  $v$  in  $G_\Pi$  is given by  $\text{prev}(\text{pair}(\tilde{x}))$ , whose target vertex lies in  
206 the same orbit as  $\tilde{v}$  under  $\Gamma$ . And so on: a sequence of accesses to  $\text{pair}(\cdot)$  and  
207  $\text{prev}(\cdot)$  allows us to find the edges of  $\Gamma\Pi$  incident to  $\tilde{v}$  in counterclockwise order.  
208 The process performs a constant number of accesses for each edge incident to  $\tilde{v}$ ,  
209 and the number of such edges is linear in  $g$  as recalled above. The case when  
210  $\tilde{v} = \text{target}(\tilde{e})$  is similar:  $\text{next}(\cdot)$  is used instead of  $\text{prev}(\cdot)$ .  $\square$

211 In addition, the precise positions in  $\mathbb{H}^2$  of all vertices of  $\Pi$  in the orbit  $\rho^{-1}v$   
212 can be computed along the process using the information  $\text{point}(\cdot)$  and  $\text{w}(\cdot)$  stored  
213 in the data structure, without changing the complexity.

214 Relations in the finitely presented group [6, Chapter 5.5]  $\Gamma$  can be deduced  
215 by comparing the two sequences of edges —clockwise and counterclockwise—  
216 around each vertex.

## 217 4 Constructing the initial system of simple loops

218 The combinatorial part of Step 1 of the algorithm is quite common in the topol-  
219 ogy literature: it consists in computing a spanning tree  $\mathcal{T}$  of  $G_\Pi$ , then the edges  
220 of  $\mathcal{T}$  are contracted, so that each vertex of  $\mathcal{T}$  is merged into the root, and each  
221 edge of  $G_\Pi$  that is not an edge of  $\mathcal{T}$  is transformed into a loop based at the  
222 root. This is illustrated in genus 2 by Figure 2(Top). In this example,  $\mathcal{T}$  has  
223 three edges  $e_1, e_5$ , and  $e_6$ . If  $v$  is chosen as the root, edge  $e_0$  transforms into a  
224 loop based at  $v$  when  $e_1$  and  $e_6$  are contracted.

225 However, topology is not enough in this work. We actually compute the  
226 geometry of each loop that is obtained from the contraction of  $\mathcal{T}$  by precisely  
227 computing a lift.

228 The main result of this section is as follows:

229 **Proposition 3.** *Let  $S$  be a closed orientable surface of genus  $g$  and  $\Pi$  a funda-*  
230 *mental polygon of  $S$  with  $2m$  edges and side pairings as described in Section 3.*  
231 *A system of loops based at a common point on  $S$ , given by a circular list of*  
232 *geodesic segments in  $\mathbb{H}^2$  and side pairings, can be constructed in time  $O(g^3)$ .*  
233 *The total length of this system of loops is  $O(gL)$ , where  $L$  denotes the perimeter*  
234 *of  $\Pi$ .*

235 The construction algorithm proceeds in three phases:

- 236 (i). Compute a spanning tree  $\mathcal{T}$  of  $G_\Pi$ . A root  $b$  is chosen for  $\mathcal{T}$ , together  
237 with an edge  $e_0$  incident to  $b$  in  $G_\Pi \setminus \mathcal{T}$  and lifts  $\tilde{b}$  and  $\tilde{e}_0$  in  $\Pi$ . Up to a  
238 renaming of representatives in orbits, we can assume that  $\tilde{b} = \text{source}(\tilde{e}_0)$ .
- 239 (ii). Construct a new fundamental domain  $\Pi'$ , as a polygon whose edges are  
240 paths consisting of  $O(g)$  geodesic segments in  $\Gamma\Pi$ : in each such path, one  
241 segment is a lift of an edge of  $G_\Pi \setminus \mathcal{T}$ , and the other segments are lifts  
242 of edges of  $\mathcal{T}$ ; the endpoints of each path lie in the orbit of  $\tilde{v}$ . The side  
243 pairings in  $\Pi'$  are also computed.

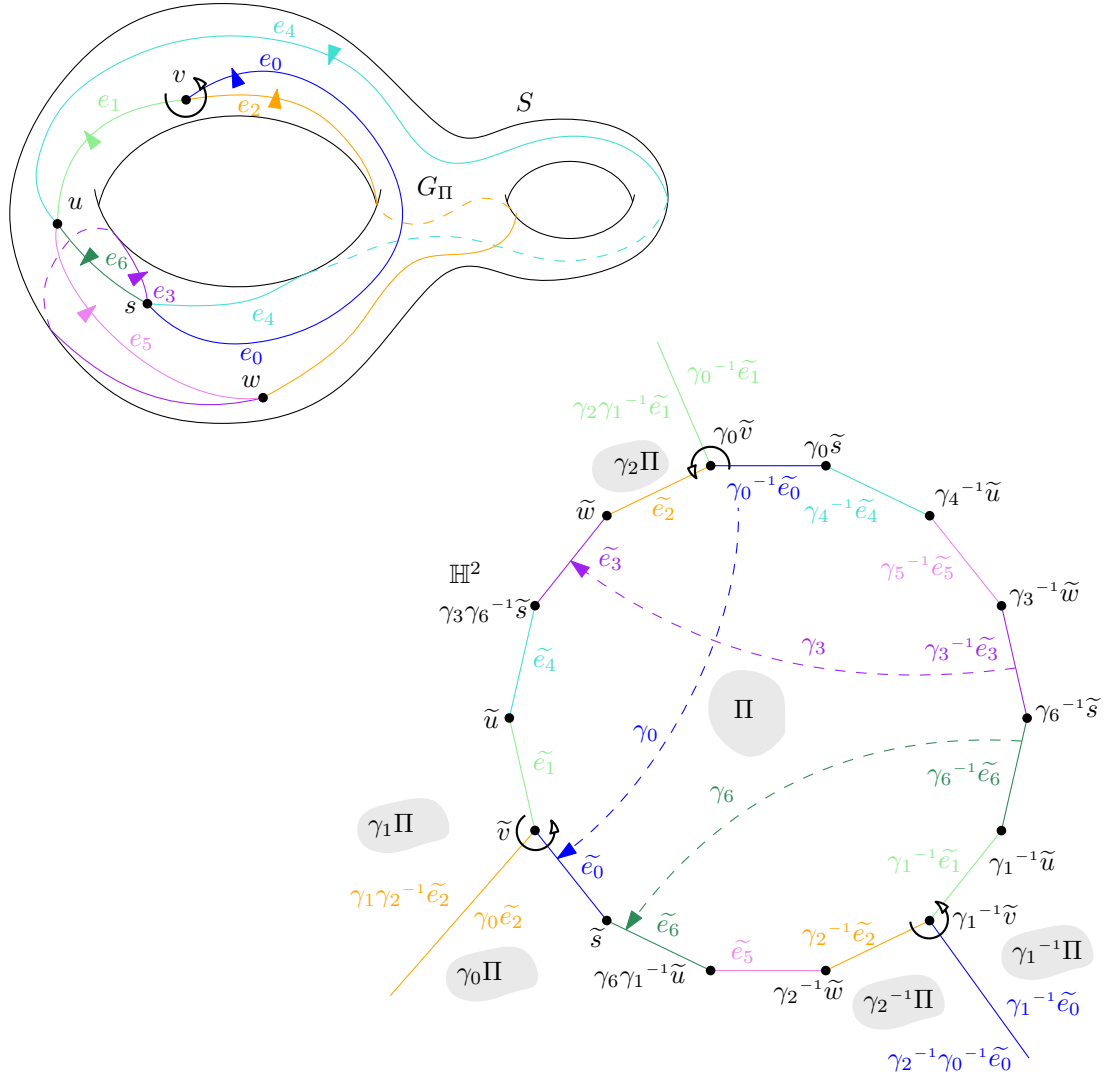


Figure 1: (Top) The graph  $G_\Pi$ . The arrow around vertex  $v$  shows its incident edges. (Bottom) The fundamental polygon  $\Pi$ . Vertices  $\tilde{s}, \tilde{u}, \tilde{v}$ , and  $\tilde{w}$  of  $\Pi$  are chosen as representatives of the orbits of  $s, u, v$ , and  $w$ , respectively. The arrows show the combinatorics of the tiling  $\Gamma\Pi$  at the three vertices of  $\Pi$  in the orbit  $\rho^{-1}v$ :  $\tilde{v}, \gamma_0\tilde{v} = \gamma_2\gamma_1^{-1}\tilde{v}$ , and  $\gamma_1^{-1}\tilde{v} = \gamma_2^{-1}\gamma_0$ .



244 (iii). Replace each path computed in the previous stage by the geodesic segment  
 245 between its vertices and keep the side pairings.

246 Note that  $\Pi'$  is a fundamental domain, but not a fundamental polygon in the  
 247 sense of Section 2.2: its edges are paths consisting of several geodesic segments;  
 248 the geodesic segments between its vertices (i.e., endpoints of these paths) will  
 249 intersect in general, so they do not bound a fundamental domain. We will call  
 250 such a polygon a *topological polygon*. Section 5 will present the construction of  
 251 a fundamental polygon from this topological polygon (Step 2 of the algorithm).

252 The rest of this section is devoted to proving Proposition 3, by detailing the  
 253 construction.

254 *Proof.* As in Section 3,  $n$  denotes the number of vertices of  $G_\Pi$ . Phase (i) is per-  
 255 formed by a standard construction of a minimum spanning tree  $\mathcal{T}$  in  $O(m \log n)$   
 256 or  $O(m + n \log n)$ , i.e.,  $O(g \log g)$ . The tree has  $n - 1$  edges.

257 Phase (ii) consists of walking along the edges of  $\Gamma_\Pi$ . The walk constructs  
 258 the new fundamental domain  $\Pi'$  in counterclockwise order and stores it in a  
 259 data structure that is very similar to the data structure defined in Section 3  
 260 for  $\Pi$ . However some of its elements have a different meaning, which will be  
 261 detailed in the sequel; in particular, the elements  $\text{pair}(\cdot)$  are actually not yet  
 262 side pairings in this phase, but temporary elements of  $\Gamma$ .

263 As a preprocessing step, for each edge  $x$  of  $G_\Pi \setminus \mathcal{T}$ , we find the path  $\mathbf{p}(x)$   
 264 on  $S$  whose homotopy class contains the loop that will eventually replace  $x$ : it  
 265 is given by the (unique and simple) path in  $\mathcal{T}$  from the root to a first vertex  
 266 of  $x$ , followed by  $x$ , and finally by the path in  $\mathcal{T}$  from the second vertex of  $x$   
 267 to the root. In the example of Figure 2(Top),  $e_0$  is replaced by a loop based  
 268 at  $b = v$  that is homotopic to the sequence  $\mathbf{p}(e_0) = e_0 \cdot e_6 \cdot e_1$ , where  $\cdot$  denotes  
 269 concatenation of paths. The path for edge  $e_4$  is  $\mathbf{p}(e_4) = e_1 \cdot e_4 \cdot e_6 \cdot e_1^{-1}$ ; here,  
 270 edge  $e_1$  is traversed in both directions.

271 The walk starts at  $\tilde{b}$  and first considers edge  $\tilde{e}_0$  chosen in stage (i). For each  
 272 considered edge  $\tilde{x}$  not in  $\rho^{-1}\mathcal{T}$ , by the pre-processing we have just mentioned,  
 273 we look for lifts of edges of  $\mathbf{p}(x)$  in order in  $\Gamma_\Pi$ . This is easily done by a sequence  
 274 of operations  $\text{next}(\cdot)$ ,  $\text{prev}(\cdot)$ , and  $\text{pair}(\cdot)$  on edges of  $\Pi$ , and turning around their  
 275 vertices  $\text{source}(\cdot)$  and  $\text{target}(\cdot)$  as in Lemma 2 until a lift of the next element of  
 276  $\mathbf{p}(x)$  is found. On the way, the elements  $w(\cdot)$  of  $\Gamma$  found in the data structure  
 277 are collected so that the precise lift of each edge or vertex of  $\Pi'$  is known.

278 Each time a lift of a path  $\mathbf{p}(x)$ , i.e., an edge of  $\Pi'$ , has been found, the  
 279 algorithm proceeds to the next one. Note that edges (i.e., paths) appear on  $\Pi'$   
 280 in the same order as the order in which the corresponding edges appear on  $\Pi$ :  
 281 indeed, contracting the edges of  $\mathcal{T}$  does not change the order in which edges on  
 282  $S$  are traversed to describe the boundary of the face of  $G_\Pi$ .

283 This is illustrated in Figure 2(Bottom). The walk starts from  $\tilde{b}$  and follows  
 284  $\tilde{e}_0$ . The next edge in  $\Pi$  is  $\text{next}(\tilde{e}_0) = \tilde{e}_6$ , which projects onto the next edge in  
 285  $\mathbf{p}(e_0)$ . Then we must look for a lift of  $e_1$  incident to  $\text{target}(\tilde{e}_6)$ . This is done by  
 286 going to  $\text{pair}(\tilde{e}_6) = \gamma_6^{-1}\tilde{e}_6$  and turning around its source vertex. The first edge  
 287 in counterclockwise order is  $\gamma_1^{-1}\tilde{e}_1$ ; the source vertex of its image  $\gamma_6\gamma_1^{-1}\tilde{e}_1$  is  
 288 the target vertex of  $\tilde{e}_6$  and the walk traverses it. Its target vertex is  $\gamma_6\gamma_1^{-1}\tilde{b}$ ,  
 289 which is in the orbit of  $\tilde{b}$ . We have now found the lift of  $\mathbf{p}(e_0)$  in  $\Gamma_\Pi$  that forms  
 290 the first edge of  $\Pi'$ : it is the sequence  $\tilde{e}_0 \cdot \tilde{e}_6 \cdot \gamma_6\gamma_1^{-1}\tilde{e}_1$ . From vertex  $\gamma_6\gamma_1^{-1}\tilde{b}$  we  
 291 will now construct the edge of  $\Pi'$  corresponding to  $\tilde{e}_2$ , as  $\tilde{e}_2$  is the edge following



292  $\gamma_0^{-1}\tilde{e}_0$  in  $\Pi \setminus \rho^{-1}\mathcal{T}$ . We know that  $\mathbf{p}(e_2) = e_2 \cdot e_5 \cdot e_1$ . Here, turning around  
 293 the source vertex of  $\gamma_1^{-1}\tilde{e}_1$  gives  $\gamma_2^{-1}\tilde{e}_2$ , and the walk continues with  $\gamma_6\gamma_2^{-1}\tilde{e}_2$ ,  
 294 then turning around the source of  $\gamma_2^{-1}\tilde{e}_2$  we find  $\tilde{e}_5$ , and the walk follows  $\gamma_6\tilde{e}_5$ .  
 295 So far we have only followed edges of  $\gamma_6\Pi$ , as the edge that we were looking for  
 296 when turning around vertices was always the first one. However, this is not the  
 297 case after  $\gamma_6\tilde{e}_5$ . The target of the representative  $\gamma_5^{-1}\tilde{e}_5$  is  $\gamma_4^{-1}\tilde{u}$ , around which  
 298 we must turn until we find a lift of  $e_1$ ; the next edge of  $\Gamma\Pi$  that we follow is thus  
 299  $\gamma_6\gamma_5\gamma_5^{-1}\gamma_6\gamma_1^{-1}\tilde{e}_1 = \gamma_6\gamma_6\gamma_1^{-1}\tilde{e}_1$ , which finishes the edge of  $\Pi'$  corresponding to  
 300  $\tilde{e}_2$ . Next, we would continue with  $\tilde{e}_3$  in the same vein. And so on.

301 Note that, as we are constructing the fundamental domain  $\Pi'$ , following the  
 302 order of the edges of  $\Pi$ , each edge  $\tilde{e}'_i$  of  $\Pi'$  defines a topological loop  $\beta'_i$  based at  
 303  $b$  on  $S$ , which represents the homotopy class  $[e'_i]$ . Such an edge  $\tilde{e}'_i$  is formed by  
 304 a sequence of edges of  $\Gamma\Pi$  that corresponds to the path  $\mathbf{p}(e_i)$ , for  $e_i \in G_\Pi \setminus \mathcal{T}$ ,  
 305 and will naturally be paired with another sequence for the same  $\mathbf{p}(e_i)$  (traversed  
 306 in the opposite direction around  $\Pi'$ ). The words associated with the edges in  
 307 the two sequences differ by an element of  $\Gamma$ , which gives the side pairing  $\gamma'_i \in \Gamma$   
 308 for  $\Pi'$ .

309 Phase (iii) is easy. It consists in replacing each edge of  $\Pi'$  by the geodesic  
 310 segment between its two vertices, and keeping the associated side pairings. As  
 311 mentioned above, the corresponding geodesic loops may intersect on  $S$ , though  
 312 the topological loops that we choose to represent their homotopy classes only  
 313 intersect at their common basepoint.

314 As the edges of  $\Pi'$  project by construction to loops, all based at the same  
 315 point, there are  $2g$  such loops on  $S$  and  $\Pi'$  has  $4g$  edges, each consisting of  $O(g)$   
 316 edges (and vertices) of  $\Gamma\Pi$ . By Lemma 2,  $O(g)$  operations are performed at  
 317 each vertex. This shows the complexity announced in Proposition 3. The bound  
 318 on the sum of the lengths of the geodesic loops also follows directly.  $\square$

319 Note that during the traversal detailed in the proof, we have computed for  
 320 each vertex  $\tilde{x}$  of  $\Pi'$  the element  $\gamma \in \Gamma$  such that  $\tilde{x} = \gamma\tilde{b}$ . We store these elements  
 321 of  $\Gamma$  in a table  $\mathfrak{t}$ , which will be used in the sequel, in addition to the data main  
 322 data structure.

323 We denote the output of this step 2 as follows: we re-index the sides of the  
 324 topological polygon  $\Pi'$  (which has  $4g \leq 2m$  edges) so that the side pairings  
 325 are denoted as  $\gamma'_0, \dots, \gamma'_{2g-1}$  and the corresponding  $2g$  topological loops on  $S$   
 326 are  $\beta'_0, \dots, \beta'_{2g-1}$ ; these loops on  $S$  do not intersect except at their common  
 327 basepoint  $b$ , which is now renamed to  $b'$  for global consistency of notation, as  
 328 announced at the beginning of Section 3.

## 329 5 Finding an embedded system of loops

330 We want to find a collection of geodesic loops on a hyperbolic surface  $S$ , all based  
 331 in a single point and disjoint otherwise, and such that the complementary region  
 332 of the loops is a convex hyperbolic polygon. What we show is that in fact we  
 333 can retain the choice of topological loops  $\beta'_0, \dots, \beta'_{2g-1}$  made in Section 4 by  
 334 moving the basepoint appropriately to ensure that their geodesic realizations  
 335 satisfy the desired properties.

336 Consider the set of topological loops  $\beta'_0, \dots, \beta'_{2g-1}$  all based at point  $b'$  con-  
 337 structed in the previous section. We choose a pair that intersects minimally

338 exactly once which, up to reordering, we can suppose are  $\beta'_0$  and  $\beta'_1$ . For future  
 339 reference we set  $L_0 := \max\{\ell(\beta'_0), \ell(\beta'_1)\}$ , where  $\ell$  denotes the length.

340 **Remark 4.** *We can fix any loop to be  $\beta'_0$  and find a loop  $\beta'_1$  intersecting it exactly*  
 341 *once. Indeed, the set  $\beta'_0, \dots, \beta'_{2g-1}$  contains curves that pairwise intersect at*  
 342 *most once, and are all non-separating and thus homologically non-trivial. As it*  
 343 *generates homotopy, it also generates homology and in particular every curve*  
 344 *must be intersected by at least one other curve. As they can intersect at most*  
 345 *once, they intersect exactly once.*

346 We begin by taking the unique geodesic loops, based in  $b'$ , in the free ho-  
 347 motopy classes of  $\beta'_0$  and  $\beta'_1$ , and we replace the curves with these geodesic  
 348 representatives (we keep the same notation for convenience). Now we further  
 349 consider the unique simple closed geodesic representatives in the free homotopy  
 350 class of  $\beta'_0$  and  $\beta'_1$ , which we denote  $\beta''_0$  and  $\beta''_1$ , respectively. By hypothesis,  
 351 they intersect in a single point  $b''$ , which will be our new basepoint.

352 We now define a path between  $b''$  and  $b'$  as follows. We consider a single lift  
 353  $\widetilde{\beta}'_0$  of  $\beta'_0$ . Its endpoints both correspond to distinct lifts of  $b'$  which are related  
 354 by a unique translation  $g_0$  in  $\Gamma$ . The copies of  $\widetilde{\beta}'_0$  by iterates of  $g_0$  form a broken  
 355 geodesic line with the same end points at infinity as the geodesic axis of  $g_0$ . This  
 356 singular geodesic, which we denote  $\hat{\beta}'_0$ , separates  $\mathbb{H}^2$  into two half-spaces, only  
 357 one of which is convex. We now choose an endpoint of  $\widetilde{\beta}'_0$  and consider a lift of  
 358  $\beta''_1$  that lies in the convex half-space. This lift we denote by  $\widetilde{\beta}'_1$  and, as before,  
 359 we consider the corresponding translation  $g_1$  in  $\Gamma$  and its geodesic axis and its  
 360 corresponding singular geodesic  $\hat{\beta}'_1$ . Now, we obtain  $\widetilde{b}''$  as the intersection of  
 361 the axes of  $g_0$  and  $g_1$ . We consider the unique geodesic path  $\widetilde{c}$  between  $\widetilde{b}'$  and  
 362  $\widetilde{b}''$  and its projection  $c$  on  $S$ . We first observe that we can control the length of  
 363 this path  $c$ :

364 **Lemma 5.**  $\ell(c) < 2L_0$ .

365 *Proof.* We observe that the axis of  $g_0$  must lie in an  $R$  neighborhood of  $\hat{\beta}'_0$  where  
 366  $R < \ell(\beta'_1)$ . In particular, the axis of  $g_1$  intersects  $\widetilde{\beta}'_0$ . Similarly, the axis of  $g_0$   
 367 intersects  $\widetilde{\beta}'_1$ . Now the proof essentially follows from drawing a picture of the  
 368 above situation in  $\mathbb{H}^2$  (see Figure 3).

By following an arc of  $\widetilde{\beta}'_0$  from  $\widetilde{b}'$  and then a segment of length  $\ell(\widetilde{\beta}'_1)$  on the  
 axis  $g_1$ , we obtain a path between  $\widetilde{b}'$  to  $\widetilde{b}''$ . As such, we have

$$\ell(\widetilde{c}) < \ell(\widetilde{\beta}'_0) + \ell(\widetilde{\beta}'_1)$$

and so by passing to the surface

$$\ell(c) < 2L_0.$$

369

□

370 Observe that for  $i = 0, 1$ ,  $\beta''_i$ , based in  $b''$ , is freely homotopic to  $c^{-1} \cdot \beta'_i \cdot c$   
 371 and that there is a homeomorphism of  $S$ , isotopic to the identity, that takes  $b'$   
 372 to  $b''$  and that sends (the homotopy class of)  $\beta'_i$  to  $\beta''_i$ . This homeomorphism is  
 373 often referred to as the point pushing map (see for instance [13, Section 4.2] for  
 374 details).

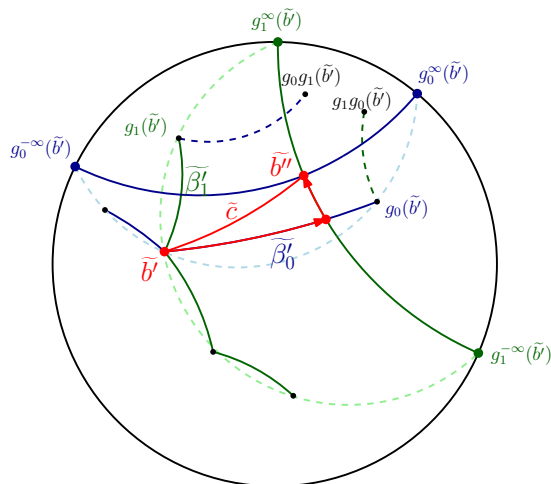


Figure 3:  $c$  is homotopic on  $S$  to the projection of the concatenation of the red arc of  $\tilde{\beta}'_0$  from  $\tilde{b}'$  and the red segment of the axis of  $g_1$ .

375 We can apply this same homeomorphism to the remaining curves. For  $i =$   
 376  $0, \dots, 2g - 1$  we set the homotopy class of loop  $\beta''_i$  to be:

$$[\beta''_i]_{b''} = [c^{-1} \cdot \beta'_i \cdot c]_{b''}. \quad (5.1)$$

377 As we have just moved the basepoint by a homeomorphism, the homotopy  
 378 classes  $[\beta''_i]_{b''}$  all have simple representatives and can be realized disjointly out-  
 379 side of  $b''$ . The following lemma implies that their unique geodesic representa-  
 380 tives enjoy this same property. It is well known to specialists, but we include a  
 381 proof sketch for completeness.

382 **Lemma 6.** *Let  $\Sigma$  be a hyperbolic surface with piecewise-geodesic boundary such*  
 383 *that the interior angles on the singular points  $s_0, \dots, s_{k-1}$  of the boundary are*  
 384 *cone points of angle  $\leq \pi$ . If  $[\alpha]_{p_i, q_i}, [\alpha']_{p_j, q_j}$  are simple homotopy classes of*  
 385 *paths (with endpoints  $p_i, p_j, q_i, q_j$  in the set  $s_0, \dots, s_{k-1}$ ), and disjoint except*  
 386 *for possibly in their endpoints, then the unique geodesic representatives are also*  
 387 *simple and disjoint.*

388 *Sketch of proof.* We consider  $\tilde{\Sigma}$ , the universal cover of  $\Sigma$ , which we view as a  
 389 (geodesically convex) subset of  $\mathbb{H}^2$ . We lift  $\partial\Sigma$  to  $\tilde{\Sigma}$  and representatives of  $[\alpha]_{p_i, q_i}$   
 390 and  $[\alpha']_{p_j, q_j}$ , which are simple and disjoint, to the universal cover. Observe that  
 391 being simple and disjoint is equivalent to *all* individual lifts in  $\mathbb{H}^2$  being simple  
 392 and pairwise disjoint. Now take two individual lifts of either  $\alpha$  or  $\alpha'$ , and their  
 393 unique geodesic representatives. We will see that they are also disjoint. Note  
 394 that in general, given two simple disjoint paths in the hyperbolic (or Euclidean)  
 395 plane, the unique geodesics between their endpoints might intersect (as already  
 396 mentioned in Section 2.1). However:

397 Observation: *Let  $C \subset \mathbb{H}^2$  be a convex with non-empty boundary, and  $p_0, q_0, p_1, q_1 \in$*   
 398  *$\partial C$ . Let  $\alpha_1 : [0, 1] \rightarrow C$  and  $\alpha_2 : [0, 1] \rightarrow C$  be simple paths, disjoint in their*  
 399 *interior, with  $\alpha_0(0) = p_0, \alpha_0(1) = q_0$  and  $\alpha_1(0) = p_1, \alpha_1(1) = q_1$ . Then the*  
 400 *unique geodesic between  $p_0$  and  $q_0$  and the unique geodesic between  $p_1$  and  $q_1$*   
 401 *are disjoint in their interior as well.*

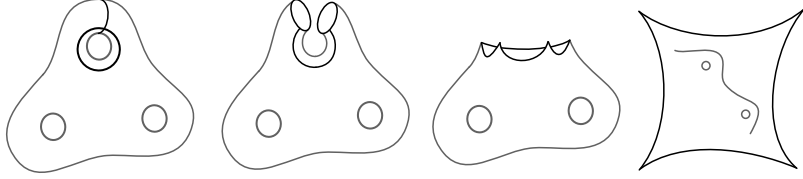


Figure 4: A visualization of the cutting along  $\beta_0''$  and  $\beta_1''$ .

402 A key point is that, thanks to the angle condition on the cone points,  $\tilde{\Sigma}$  is  
 403 a convex region of  $\mathbb{H}^2$ . (This is just a slightly more sophisticated observation  
 404 than the elementary fact that a polygon with all interior angles less than  $\pi$  is  
 405 convex.) The observation now implies that the lifts of geodesics corresponding  
 406 to  $\alpha$  and  $\alpha'$  are disjoint in their interior if and only if there are representatives  
 407 of  $[\alpha]_{p_i, q_i}$  and  $[\alpha']_{p_j, q_j}$  that are, too, which, by hypothesis, is the case.  $\square$

408 We can now apply Lemma 6 to the geodesic representatives of  $[\beta_i'']_{b''}$ . For  
 409 simplicity we denote by  $\beta_i''$  the unique geodesic loop in the corresponding ho-  
 410 motopy class.

411 **Theorem 7.** *Let  $\beta_0'', \dots, \beta_{2g-1}''$  be a set of topological loops based in  $b''$  that cuts*  
 412 *a surface  $S$  into a disk. Assume that  $\beta_0''$  and  $\beta_1''$  are closed geodesics. Then,*  
 413 *the geodesic loops homotopic to  $\beta_0'', \dots, \beta_{2g-1}''$  are simple and pairwise disjoint*  
 414 *in their interiors. Furthermore, by cutting  $S$  along those geodesics and lifting*  
 415 *to  $\mathbb{H}^2$ , one obtains a convex hyperbolic polygon with  $4g$  edges.*

416 *Proof.* As  $\beta_0''$  and  $\beta_1''$  are closed geodesics, they form 4 angles in  $b''$ , and the  
 417 opposite ones are equal. These angles thus satisfy  $2\theta + 2\theta' = 2\pi$  so in particular  
 418 both  $\theta$  and  $\theta'$  are strictly less than  $\pi$ . Thus by cutting along  $\beta_0''$  and  $\beta_1''$ , we  
 419 obtain a genus  $g - 1$  surface with a boundary consisting of 4 geodesic segments,  
 420 and with 4 cone point singularities of angles  $< \pi$  (see Figure 4)).

421 We now proceed inductively for  $i \geq 3$  and consider the unique geodesic path  
 422  $\beta_i''$ , which by virtue of Lemma 6, has disjoint interior from the previous geodesic  
 423 segments. Furthermore, as each segment further splits the angles, the angles  
 424 are all less than  $\pi$ .

425 The end result is a polygon with all interior angles less than  $\pi$  which, by  
 426 elementary hyperbolic geometry, is convex.  $\square$

427 **Proposition 8.** *Let  $S$  be hyperbolic of genus  $g$  and  $\Pi'$  a topological fundamental*  
 428 *polygon of  $S$  with  $4g$  edges and side pairings as described at the end of Section 4.*  
 429 *A convex fundamental polygon  $\Pi''$  with its side pairing and whose vertices project*  
 430 *to a single vertex on  $S$ , can be constructed in  $O(g)$  time. The perimeter of  $\Pi''$*   
 431 *has length  $O(gL')$ , where  $L'$  denotes the total geodesic length of the sides of  $\Pi'$ .*

432 *Proof.* We need to compute the output convex polygon  $\Pi''$  i.e.,  $4g$  lifts of  $b''$  and  
 433  $2g$  side pairings  $\gamma_0'', \dots, \gamma_{2g-1}''$ . As homotopy classes of  $\beta_i''$  and  $\beta_i'$  are conjugates  
 434 for  $i = 0, \dots, 2g - 1$  (Equation 5.1), the side pairing  $\gamma_i''$  is equal to  $\gamma_i'$  for each  
 435  $i$ .

436 The key point here is the computation of a lift of  $b''$ . The first step consists  
 437 in finding the loops  $\beta_0'$  and  $\beta_1'$  satisfying  $i(\beta_0', \beta_1') = 1$ . As shown in Remark 4,  
 438 we can choose any loop for  $\beta_0'$ . We also fix  $b_0'$  to be an endpoint of one of the  
 439 two paired sides of  $\Pi'$  that are lifts of  $\beta_0'$ . We compute the ordered sequence

440 of loops around  $b'$  as in Lemma 2, in  $O(g)$  operations; recall that each loop  
 441  $\beta'_0, \dots, \beta'_{2g-1}$  appears twice in the sequence (Section 2.2). We take as  $\beta'_1$  one of  
 442 the loops that alternate with  $\beta'_0$  in the sequence, and choose for  $\widetilde{\beta}'_1$  one of its  
 443 two lifts that are incident to  $\widetilde{b}'_0$ .

444 The second step consists in finding the free geodesics in the homotopy classes  
 445 of  $\beta'_0$  and  $\beta'_1$ , respectively. In  $\mathfrak{t}$ , we find the word  $g_0$  on  $\{\gamma'_0, \dots, \gamma'_{2g-1}\}$  repre-  
 446 senting the translation that sends  $\widetilde{b}'_0$  to the other endpoint of  $\widetilde{\beta}'_0$  (see Figure 5).  
 447 The sequences  $g_0^n(\widetilde{b}'_0)$  and  $g_0^{-n}(\widetilde{b}'_0)$  converge in  $\mathbb{C}$  to two points on the unit circle:  
 448 these points are the two (infinite in  $\mathbb{H}^2$ ) fixed points of the translation  $g_0$ , i.e.,  
 449 the two solutions of equation  $g_0(z) = z$  in  $\mathbb{C}$ . The axis of  $g_0$ , i.e., the geodesic  
 450 between these two points, projects onto  $S$  to the free geodesic in  $[\beta'_0]$ .

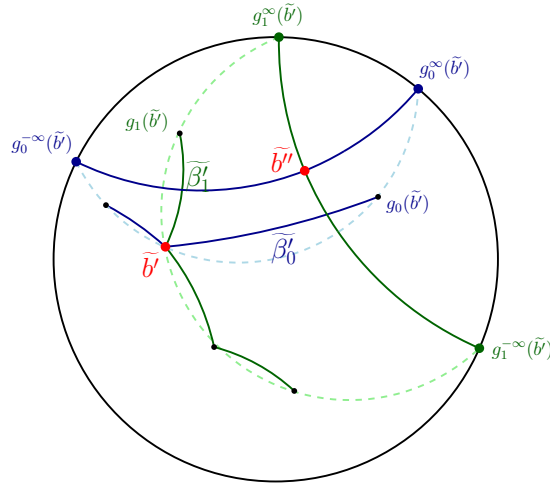


Figure 5: The computation of  $b''$  from  $b'$ .

451 We repeat the same process with  $\widetilde{\beta}'_1$  and find the geodesic in  $\mathbb{H}^2$  that projects  
 452 to the free geodesic in  $[\beta'_1]$ . The point  $\widetilde{b}''_0$  comes as the intersection point of the  
 453 two geodesics in  $\mathbb{H}^2$ . We now define  $\widetilde{\beta}''_0$  as the geodesic segment between  $\widetilde{b}''_0$  and  
 454  $w_0(\widetilde{b}''_0)$ , and  $\widetilde{\beta}''_1$  in the same way. This step is performed in constant time.

We can now compute the  $4g$  lifts of  $b''$  that are the vertices of  $\Pi''$  by applying  
 the elements of  $\mathfrak{t}$  to  $\widetilde{b}''_0$ . This last step has complexity  $O(g)$ . Additionally, we  
 have

$$\ell(\beta''_i) \leq \ell(\beta'_i) + 2\ell(c) = 5 \cdot (\max_j \ell(\beta'_j))$$

455 for each  $i = 1, \dots, 2g - 1$  (by Lemma 5) and thus the perimeter of  $\Pi''$  is  $O(g)$   
 456 times bigger than the perimeter  $\Pi'$ .  $\square$

## 457 6 Finding a Dirichlet domain from an embedded 458 system of loops

459 We first summarize what we have obtained so far. We started with a polygon  
 460  $\Pi$  of perimeter  $L$  and we obtained a convex polygon  $\Pi''$  of total length  $O(g^2L)$ .

461 Additionally, all vertices of  $\Pi''$  project on a single vertex  $b''$  on  $S$ . This con-  
 462 struction has complexity  $O(g^3)$  by Propositions 3 and 8. At this point, it is easy  
 463 to compute a Dirichlet domain. Indeed, we can now triangulate  $\Pi''$  easily since  
 464 it is convex and, thus, we obtain a geometric triangulation  $T$ , on to which the  
 465 Delaunay flip algorithm can be applied [11]. The complexity of this algorithm  
 466 depends on the diameter of  $T$ , for which the perimeter of  $\Pi''$  is an upper bound.

467 The output of the flip algorithm is a Delaunay triangulation  $DT$  of  $S$  with  
 468 the single vertex  $b''$  computed in Section 5. To obtain a Dirichlet domain from  
 469  $DT$ , we just have to compute the triangles of  $\widetilde{DT}$  incident to a lift  $\widetilde{b''}$  of  $b''$   
 470 and their dual: we compute the circumcenter of each triangle to obtain the  
 471 vertices of the Dirichlet domain and we put a geodesic between vertices that  
 472 correspond to adjacent triangles around  $b''$ . This step is also clearly done in  
 473  $O(g)$  operations. Putting all together we obtain the following theorem:

474 **Theorem 9.** *Let  $S$  be a closed orientable hyperbolic surface of genus  $g$  given*  
 475 *by a fundamental polygon of perimeter  $L$  and side pairings. A Dirichlet domain*  
 476 *of  $S$  can be computed in time  $O(f(g^2L) + g^3)$  where  $f(\Delta)$  is the complexity of*  
 477 *the flip algorithm for a triangulation of diameter  $\Delta$  with a single vertex.*

478 Using the best known bound for the flip algorithm so far, we obtain Theom-  
 479 rem 1 stated in the introduction as a corollary. Note that the constant in the  
 480  $O()$  depends on the metric on  $S$ . However, there are experimental and theoret-  
 481 ical insights suggesting that the actual complexity of the flip algorithm may be  
 482 much better [9].

## 483 References

- 484 [1] N.L. Balazs and A. Voros. Chaos on the pseudosphere. *Physics Reports*,  
 485 143(3):109–240, 1986. doi:10.1016/0370-1573(86)90159-6.
- 486 [2] Alan F. Beardon. *The Geometry of Discrete Groups*. Springer-Verlag, 1983.
- 487 [3] Mikhail Bogdanov, Olivier Devillers, and Monique Teillaud. Hyperbolic  
 488 Delaunay complexes and Voronoi diagrams made practical. *Journal of*  
 489 *Computational Geometry*, 5:56–85, 2014. doi:10.20382/jocg.v5i1a4.
- 490 [4] Mikhail Bogdanov, Monique Teillaud, and Gert Vegter. Delaunay trian-  
 491 gulations on orientable surfaces of low genus. In Sándor Fekete and Anna  
 492 Lubiw, editors, *32nd International Symposium on Computational Geometry*  
 493 *(SoCG 2016)*, volume 51 of *Leibniz International Proceedings in Informat-*  
 494 *ics (LIPIcs)*, pages 20:1–20:17, Dagstuhl, Germany, 2016. Schloss Dagstuhl  
 495 – Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.SoCG.2016.20.
- 496 [5] Peter Buser. *Geometry and spectra of compact Riemann surfaces*. Modern  
 497 Birkhäuser classics. Birkhäuser, Boston, Mass., 2nd edition, 2010.
- 498 [6] H. S. M. Coxeter and W. O. J. Moser. *Generators and Relations for Discrete*  
 499 *Groups*. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1957.
- 500 [7] Jason DeBlois. The centered dual and the maximal injectivity radius of  
 501 hyperbolic surfaces. *Geometry and Topology*, 19(2):953–1014, 2015. doi:  
 502 10.2140/gt.2015.19.953.



- 503 [8] Jason DeBlois. The Delaunay tessellation in hyperbolic space. *Mathematical Proceedings of the Cambridge Philosophical Society*, 164(1):15–46, 2018.  
504 doi:10.1017/S0305004116000827.  
505
- 506 [9] Vincent Despré, Loïc Dubois, Benedikt Kolbe, and Monique Teillaud. Experimental analysis of Delaunay flip algorithms on genus two hyperbolic  
507 surfaces. Preprint, INRIA, May 2021. URL: <https://hal.inria.fr/hal-03462834/>.  
508  
509
- 510 [10] Vincent Despré, Benedikt Kolbe, and Monique Teillaud. Representing  
511 infinite hyperbolic periodic Delaunay triangulations using finitely many  
512 Dirichlet domains. Preprint, INRIA, July 2021. URL: <https://hal.archives-ouvertes.fr/hal-03045921>.  
513
- 514 [11] Vincent Despré, Jean-Marc Schlenker, and Monique Teillaud. Flipping  
515 geometric triangulations on hyperbolic surfaces. In Sergio Cabello and  
516 Danny Z. Chen, editors, *36th International Symposium on Computational  
517 Geometry (SoCG 2020)*, volume 164 of *Leibniz International Proceedings in  
518 Informatics (LIPIcs)*, pages 35:1–35:16, Dagstuhl, Germany, 2020. Schloss  
519 Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SocG.  
520 2020.35.
- 521 [12] Matthijs Ebbens, Hugo Parlier, and Gert Vegter. Minimal Delaunay tri-  
522 angulations of hyperbolic surfaces. In Kevin Buchin and Éric Colin de  
523 Verdière, editors, *37th International Symposium on Computational Geom-  
524 etry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in In-  
525 formatics (LIPIcs)*, pages 31:1–31:16, Dagstuhl, Germany, 2021. Schloss  
526 Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SocG.  
527 2021.31.
- 528 [13] Benson Farb and Dan Margalit. *A Primer on Mapping Class Groups (PMS-  
529 49)*. Princeton University Press, 2012. URL: [http://www.jstor.org/  
530 stable/j.ctt7rkjw](http://www.jstor.org/stable/j.ctt7rkjw).
- 531 [14] Iordan Iordanov and Monique Teillaud. Implementing Delaunay triangula-  
532 tions of the Bolza surface. In *33rd International Symposium on Computa-  
533 tional Geometry (SoCG 2017)*, pages 44:1–44:15, Brisbane, Australia, July  
534 2017. doi:10.4230/LIPIcs.SocG.2017.44.
- 535 [15] Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins  
536 University Press, Baltimore, 2001.
- 537 [16] Nikolai C Passler, Xiang Ni, Guangwei Hu, Joseph R Matson, Giulia  
538 Carini, Martin Wolf, Mathias Schubert, Andrea Alù, Joshua D Cald-  
539 well, Thomas G Folland, et al. Hyperbolic shear polaritons in low-  
540 symmetry crystals. *Nature*, 602(7898):595–600, 2022. doi:10.1038/  
541 s41586-021-04328-y.
- 542 [17] John Voight. Computing fundamental domains for Fuchsian groups. *Journal de Théorie des Nombres de Bordeaux*, 21(2):467–489, 2009. doi:  
543 10.5802/jtnb.683.  
544